Adaptive Dynamic Programming for Finite-Horizon Optimal Control of Discrete-Time Nonlinear Systems with ε-Error Bound

Fei-Yue Wang, *Fellow, IEEE*, Ning Jin, *Student Member, IEEE*, Derong Liu, *Fellow, IEEE*, and Qinglai Wei

***Abstract*— In this paper, we study the finite-horizon optimal control problem for discrete-time nonlinear systems using the adaptive dynamic programming (ADP) approach. The idea is to use an iterative ADP algorithm to obtain the optimal control law which makes the performance index function close to the greatest lower bound of all performance indices within an ε-error bound. The optimal number of control steps can also be obtained by the proposed ADP algorithms. A convergence analysis of the proposed ADP algorithms in terms of performance index function and control policy is made. In order to facilitate the implementation of the iterative ADP algorithms, neural networks are used for approximating the performance index function, computing the optimal control policy, and modeling the nonlinear system. Finally, two simulation examples are employed to illustrate the applicability of the proposed method.**

***Index Terms*— Adaptive critic designs, adaptive dynamic programming, approximate dynamic programming, learning control, neural control, neural dynamic programming, optimal control, reinforcement learning.**

I. Introduction

THE optimal control problem of nonlinear systems has always been the key focus of control fields in the past several decades [1]–[15]. Traditional optimal control approaches are mostly implemented in infinite time horizon [2], [5], [9], [11], [13], [16], [17]. However, most real-world systems need to be effectively controlled within finite time horizon (finite-horizon for brief), such as stabilized or tracked to a desired trajectory in a finite duration of time. The design of finite-horizon optimal controllers faces a major obstacle in comparison to the infinite horizon one. An infinite horizon optimal controller generally obtains an asymptotic result for the controlled systems [9], [11]. That is, the system will not be stabilized or tracked until the time reaches infinity, while for finite-horizon optimal control problems the system must be stabilized or tracked to a desired trajectory in a finite duration of time [1], [8], [12], [14], [15]. Furthermore, in the case of discrete-time systems, the determination of the number of optimal control steps is necessary for finite-horizon optimal control problems, while for the infinite horizon optimal control problems the number of optimal control steps is infinity in general. The finite-horizon control problem has been addressed by many researchers [18]–[23]. But most of the existing methods consider only the stability problems of systems under finite-horizon controllers [18], [20], [22], [23]. Due to the lack of methodology and the fact that the number of control steps is difficult to determine, the optimal controller design of finite-horizon problems still presents a major challenge to control engineers. This motivates our present research.

As is known, dynamic programming is very useful in solving optimal control problems. However, due to the "curse of dimensionality" [24], it is often computationally untenable to run dynamic programming to obtain the optimal solution. The adaptive/approximate dynamic programming (ADP) algorithms were proposed in [25] and [26] as a way to solve optimal control problems forward in time. There are several synonyms used for ADP including "adaptive critic designs" [27]–[29], "adaptive dynamic programming" [30]–[32], "approximate dynamic programming" [26], [33]–[35], "neural dynamic programming" [36], "neuro-dynamic programming" [37], and "reinforcement learning" [38]. In recent years, ADP and related research have gained much attention from researchers [27], [28], [31], [33]–[36], [39]–[57]. In [29] and [26], ADP approaches were classified into several main schemes: heuristic dynamic programming (HDP), action-dependent HDP, also known as Q-learning [58], dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP.

Saridis and Wang [10], [52], [59] studied the optimal control problem for a class of nonlinear stochastic systems and presented the corresponding Hamilton-Jacobi-Bellman (HJB) equation for stochastic control problems. Al-Tamimi *et al.* [27] proposed a greedy HDP iteration algorithm to solve the discrete-time HJB (DTHJB) equation of the optimal control problem for discrete-time nonlinear systems. Though great progress has been made for ADP in the optimal control

Manuscript received April 16, 2010; revised August 20, 2010; accepted August 24, 2010. Date of publication September 27, 2010; date of current version January 4, 2011. This work was supported in part by the Natural Science Foundation (NSF) China under Grant 60573078, Grant 60621001, Grant 60728307, Grant 60904037, Grant 60921061, and Grant 70890084, by the MOST 973 Project 2006CB705500 and Project 2006CB705506, by the Beijing Natural Science Foundation under Grant 4102061, and by the NSF under Grant ECS-0529292 and Grant ECCS-0621694. The acting Editor-in-Chief who handled the review of this paper was Frank L. Lewis.

F. Y. Wang and Q. Wei are with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: feiyue.wang@ia.ac.cn; qinglai.wei@ia.ac.cn).

N. Jin is with the Department of Electrical and Computer Engineering, University of Illinois, Chicago, IL 60607 USA (e-mail: njin@uic.edu).

D. Liu is with the Department of Electrical and Computer Engineering, University of Illinois, Chicago, IL 60607 USA. He is also with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: dliu@ece.uic.edu).

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/TNN.2010.2076370

field, most ADP methods are based on infinite horizon, such as [16], [27], [33], [36], [37], [43]–[45], [53], [56] and [57]. Only [60] and [61] discussed how to solve the finite-horizon optimal control problems based on ADP and backpropagation-through-time algorithms.

In this paper, we will develop a new ADP scheme for finite-horizon optimal control problems. We will study the optimal control problems with an ε-error bound using ADP algorithms. First, the HJB equation for finite-horizon optimal control of discrete-time systems is derived. In order to solve this HJB equation, a new iterative ADP algorithm is developed with convergence and optimality proofs. Second, the difficulties of obtaining the optimal solution using the iterative ADP algorithm is presented and then the ε-optimal control algorithm is derived based on the iterative ADP algorithms. Next, it will be shown that the ε-optimal control algorithm can obtain suboptimal control solutions within a fixed finite number of control steps that make the performance index function converge to its optimal value with an ε-error. Furthermore, in order to facilitate the implementation of the iterative ADP algorithms, we use neural networks to obtain the iterative performance index function and the optimal control policy. Finally, an ε-optimal state feedback controller is obtained for finite-horizon optimal control problems.

This paper is organized as follows. In Section II, the problem statement is presented. In Section III, the iterative ADP algorithm for finite-horizon optimal control problem is derived. The convergence property and optimality property are also proved in this section. In Section IV, the ε-optimal control algorithm is developed, the properties of the algorithm are also proved in this section. In Section V, two examples are given to demonstrate the effectiveness of the proposed control scheme. Finally, in Section VI, the conclusion is drawn.

II. Problem Statement

In this paper, we will study deterministic discrete-time systems

$$x_{k+1} = F(x_k, u_k), \qquad k = 0, 1, 2, \ldots \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state and $u_k \in \mathbb{R}^m$ is the control vector. Let x_0 be the initial state. The system function $F(x_k, u_k)$ is continuous for $\forall\, x_k, u_k$ and $F(0, 0) = 0$. Hence, $x = 0$ is an equilibrium state of system (1) under the control $u = 0$. The performance index function for state x_0 under the control sequence $\underline{u}_0^{N-1} = (u_0, u_1, \ldots, u_{N-1})$ is defined as

$$J\left(x_0, \underline{u}_0^{N-1}\right) = \sum_{i=0}^{N-1} U(x_i, u_i) \tag{2}$$

where U is the utility function, $U(0, 0) = 0$, and $U(x_i, u_i) \geq 0$ for $\forall\, x_i, u_i$.

The sequence $\underline{u}_0^{N-1}$ defined above is a finite sequence of controls. Using this sequence of controls, system (1) gives a trajectory starting from x_0: $x_1 = F(x_0, u_0)$, $x_2 = F(x_1, u_1)$, $\ldots$, $x_N = F(x_{N-1}, u_{N-1})$. We call the number of elements in the control sequence $\underline{u}_0^{N-1}$ the length of $\underline{u}_0^{N-1}$ and denote it as $\left|\underline{u}_0^{N-1}\right|$. Then, $\left|\underline{u}_0^{N-1}\right| = N$. The length of the associated trajectory $\underline{x}_0^N = (x_0, x_1, \ldots, x_N)$ is $N + 1$. We denote the final state of the trajectory as $x^{(f)}\left(x_0, \underline{u}_0^{N-1}\right)$, i.e., $x^{(f)}\left(x_0, \underline{u}_0^{N-1}\right) = x_N$. Then, for $\forall\, k \geq 0$, the finite control sequence starting at k can be written as $\underline{u}_k^{k+i-1} = (u_k, u_{k+1}, \ldots, u_{k+i-1})$, where $i \geq 1$ is the length of the control sequence. The final state can be written as $x^{(f)}\left(x_k, \underline{u}_k^{k+i-1}\right) = x_{k+i}$.

We note that the performance index function defined in (2) does not have the term associated with the final state since in this paper we specify the final state $x_N = F(x_{N-1}, u_{N-1})$ to be at the origin, i.e., $x_N = x^{(f)} = 0$. For the present finite-horizon optimal control problems, the feedback controller $u_k = u(x_k)$ must not only drive the system state to zero within finite number of time steps but also guarantee the performance index function (2) to be finite, i.e., $\underline{u}_k^{N-1} = (u(x_k), u(x_{k+1}), \ldots, u(x_{N-1}))$ must be a finite-horizon admissible control sequence, where $N > k$ is a finite integer.

Definition 2.1: A control sequence $\underline{u}_k^{N-1}$ is said to be finite-horizon admissible for a state $x_k \in \mathbb{R}^n$, if $x^{(f)}\left(x_k, \underline{u}_k^{N-1}\right) = 0$ and $J\left(x_k, \underline{u}_k^{N-1}\right)$ is finite, where $N > k$ is a finite integer. ∎

A state x_k is said to be finite-horizon controllable (controllable for brief) if there is a finite-horizon admissible control sequence associated with this state.

Let $\underline{u}_k$ be an arbitrary finite-horizon admissible control sequence starting at k and let

$$\mathfrak{A}_{x_k} = \left\{\underline{u}_k \colon x^{(f)}\left(x_k, \underline{u}_k\right) = 0\right\}$$

be the set of all finite-horizon admissible control sequences of x_k. Let

$$\mathfrak{A}_{x_k}^{(i)} = \left\{\underline{u}_k^{k+i-1} \colon x^{(f)}\left(x_k, \underline{u}_k^{k+i-1}\right) = 0,\ \left|\underline{u}_k^{k+i-1}\right| = i\right\}$$

be the set of all finite-horizon admissible control sequences of x_k with length i. Then, $\mathfrak{A}_{x_k} = \cup_{1 \leq i < \infty} \mathfrak{A}_{x_k}^{(i)}$. By this notation, a state x_k is controllable if and only if $\mathfrak{A}_{x_k} \neq \emptyset$.

For any given system state x_k, the objective of the present finite-horizon optimal control problem is to find a finite-horizon admissible control sequence $\underline{u}_k^{N-1} \in \mathfrak{A}_{x_k}^{(N-k)} \subseteq \mathfrak{A}_{x_k}$ to minimize the performance index $J\left(x_k, \underline{u}_k^{N-1}\right)$. The control sequence $\underline{u}_k^{N-1}$ has finite length. However, before it is determined, we do not know its length, which means that the length of the control sequence $\left|\underline{u}_k^{N-1}\right| = N - k$ is unspecified. This kind of optimal control problems has been called finite-horizon problems with unspecified terminal time [1] (but in the present case, with fixed terminal state $x^{(f)} = 0$).

Define the optimal performance index function as

$$J^*(x_k) = \inf_{\underline{u}_k} \left\{J(x_k, \underline{u}_k) \colon \underline{u}_k \in \mathfrak{A}_{x_k}\right\}. \tag{3}$$

Then, according to Bellman's principle of optimality [24], $J^*(x_k)$ satisfies the DTHJB equation

$$J^*(x_k) = \min_{u_k} \left\{U(x_k, u_k) + J^*(F(x_k, u_k))\right\}. \tag{4}$$

Now, define the law of optimal control sequence starting at k by

$$\underline{u}^*(x_k) = \arg\inf_{\underline{u}_k} \left\{J(x_k, \underline{u}_k) \colon \underline{u}_k \in \mathfrak{A}_{x_k}\right\}$$

and define the law of optimal control vector by

$$u^*(x_k) = \arg\min_{u_k} \left\{U(x_k, u_k) + J^*(F(x_k, u_k))\right\}.$$

In other words, $\underline{u}^*(x_k) = \underline{u}_k^*$ and $u^*(x_k) = u_k^*$. Hence, we have

$$J^*(x_k) = U\left(x_k, u_k^*\right) + J^*\left(F\left(x_k, u_k^*\right)\right).$$

III. Properties of the Iterative ADP Algorithm

In this section, a new iterative ADP algorithm is developed to obtain the finite-horizon optimal controller for nonlinear systems. The goal of the present iterative ADP algorithm is to construct an optimal control policy $u^*(x_k)$, $k = 0, 1, \ldots$, which drives the system from an arbitrary initial state x_0 to the singularity 0 within finite time, and simultaneously minimizes the performance index function. Convergence proofs will also be given to show that the performance index function will indeed converge to the optimum.

A. Derivation

We first consider the case where, for any state x_k, there exists a control vector u_k such that $F(x_k, u_k) = 0$, i.e., we can control the state of system (1) to zero in one step from any initial state. For the case where $F(x_k, u_k) = 0$ does not hold, we will discuss and solve the problem later in this paper.

In the iterative ADP algorithm, the performance index function and control policy are updated by recursive iterations, with the iteration index number i increasing from 0 and with the initial performance index function $V_0(x) = 0$ for $\forall\, x \in \mathbb{R}^n$.

The performance index function for $i = 1$ is computed as

$$\begin{aligned} V_1(x_k) &= \min_{u_k} \{U(x_k, u_k) + V_0(F(x_k, u_k))\} \\ &\qquad \text{s.t.} \quad F(x_k, u_k) = 0 \\ &= \min_{u_k} U(x_k, u_k) \quad \text{s.t.} \quad F(x_k, u_k) = 0 \\ &= U(x_k, u_k^*(x_k)) \end{aligned} \tag{5}$$

where $V_0(F(x_k, u_k)) = 0$ and $F(x_k, u_k^*(x_k)) = 0$. The control vector $v_1(x_k)$ for $i = 1$ is chosen as $v_1(x_k) = u_k^*(x_k)$. Therefore, (5) can also be written as

$$\begin{aligned} V_1(x_k) &= \min_{u_k} U(x_k, u_k) \quad \text{s.t.} \quad F(x_k, u_k) = 0 \\ &= U(x_k, v_1(x_k)) \end{aligned} \tag{6}$$

where

$$v_1(x_k) = \arg\min_{u_k} U(x_k, u_k) \quad \text{s.t.} \quad F(x_k, u_k) = 0. \tag{7}$$

For $i = 2, 3, 4, \ldots$, the iterative ADP algorithm will be implemented as follows:

$$\begin{aligned} V_i(x_k) &= \min_{u_k} \{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\} \\ &= U(x_k, v_i(x_k)) + V_{i-1}(F(x_k, v_i(x_k))) \end{aligned} \tag{8}$$

where

$$\begin{aligned} v_i(x_k) &= \arg\min_{u_k} \{U(x_k, u_k) + V_{i-1}(x_{k+1})\} \\ &= \arg\min_{u_k} \{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\}. \end{aligned} \tag{9}$$

Equations (6)–(9) form the iterative ADP algorithm.

Remark 3.1: Equations (6)–(9) in the iterative ADP algorithm are similar to the HJB equation (4), but they are not the same. There are at least two obvious differences.

1) For any finite time k, if x_k is the state at k, then the optimal performance index function in HJB (4) is unique, i.e., $J^*(x_k)$, while in the iteration ADP equation (6)–(9), the performance index function is different for each iteration index i, i.e., $V_i(x_k) \neq V_j(x_k)$ for $\forall\, i \neq j$, in general.
2) For any finite time k, if x_k is the state at k, then the optimal control law obtained by HJB (4) possesses the unique optimal control expression, i.e., $u_k^* = u^*(x_k)$, while the control law solved by the iterative ADP algorithm (6)–(9) is different from each other for each iteration index i, i.e., $v_i(x_k) \neq v_j(x_k)$ for $\forall i \neq j$, in general. ∎

Remark 3.2: According to (2) and (8), we have

$$V_{i+1}(x_k) = \min_{\underline{u}_k^{k+i}} \left\{ J\left(x_k, \underline{u}_k^{k+i}\right) : \underline{u}_k^{k+i} \in \mathfrak{A}_{x_k}^{(i+1)} \right\}. \tag{10}$$

Since

$$\begin{aligned} V_{i+1}(x_k) &= \min_{u_k} \{U(x_k, u_k) + V_i(x_{k+1})\} \\ &= \min_{u_k} \Bigg\{ U(x_k, u_k) + \min_{u_{k+1}} \Big\{ U(x_{k+1}, u_{k+1}) \\ &\quad + \min_{u_{k+2}} \big\{ U(x_{k+2}, u_{k+2}) + \cdots \\ &\quad + \min_{u_{k+i-1}} \{ U(x_{k+i-1}, u_{k+i-1}) \\ &\quad + V_1(x_{k+i}) \} \cdots \big\} \Big\} \Bigg\} \end{aligned}$$

where

$$\begin{aligned} V_1(x_{k+i}) &= \min_{u_{k+i}} U(x_{k+i}, u_{k+i}) \\ &\qquad \text{s.t.} \quad F(x_{k+i}, u_{k+i}) = 0 \end{aligned}$$

we obtain

$$\begin{aligned} V_{i+1}(x_k) &= \min_{\underline{u}_k^{k+i}} \{U(x_k, u_k) + U(x_{k+1}, u_{k+1}) \\ &\qquad + \cdots + U(x_{k+i}, u_{k+i})\} \\ &\qquad \text{s.t.} \quad F(x_{k+i}, u_{k+i}) = 0 \\ &= \min_{\underline{u}_k^{k+i}} \left\{ J\left(x_k, \underline{u}_k^{k+i}\right) : \underline{u}_k^{k+i} \in \mathfrak{A}_{x_k}^{(i+1)} \right\}. \end{aligned}$$

Using the notation in (9), we can also write

$$V_{i+1}(x_k) = \sum_{j=0}^{i} U\left(x_{k+j}, v_{i+1-j}(x_{k+j})\right). \tag{11}$$

B. Properties

In the above, we can see that the performance index function $J^*(x_k)$ solved by HJB equation (4) is replaced by a sequence of iterative performance index functions $V_i(x_k)$ and the optimal control law $u^*(x_k)$ is replaced by a sequence of iterative control law $v_i(x_k)$, where $i \geq 1$ is the index of iteration. We can prove that $J^*(x_k)$ defined in (3) is the limit of $V_i(x_k)$ as $i \to \infty$.

Theorem 3.1: Let x_k be an arbitrary state vector. Suppose that $\mathfrak{A}_{x_k}^{(1)} \neq \emptyset$. Then, the performance index function $V_i(x_k)$

obtained by (6)–(9) is a monotonically nonincreasing sequence for $\forall i \geq 1$, i.e., $V_{i+1}(x_k) \leq V_i(x_k)$ for $\forall i \geq 1$.

Proof: We prove this by mathematical induction. First, we let $i = 1$. Then, we have $V_1(x_k)$ given as in (6) and the finite-horizon admissible control sequence is $\underline{\hat{u}}_k^k = (v_1(x_k))$.

Next, we show that there exists a finite-horizon admissible control sequence $\underline{\hat{u}}_k^{k+1}$ with length 2 such that $J\left(x_k, \underline{\hat{u}}_k^{k+1}\right) = V_1(x_k)$. The trajectory starting from x_k under the control of $\underline{\hat{u}}_k^k = (v_1(x_k))$ is $x_{k+1} = F(x_k, v_1(x_k)) = 0$. Then, we create a new control sequence $\underline{\hat{u}}_k^{k+1}$ by adding a 0 to the end of sequence $\underline{\hat{u}}_k^k$ to obtain the control sequence $\underline{\hat{u}}_k^{k+1} = \left(\underline{\hat{u}}_k^k, 0\right)$. Obviously, $\left|\underline{\hat{u}}_k^{k+1}\right| = 2$. The state trajectory under the control of $\underline{\hat{u}}_k^{k+1}$ is $x_{k+1} = F(x_k, v_1(x_k)) = 0$ and $x_{k+2} = F(x_{k+1}, \hat{u}_{k+1})$, where $\hat{u}_{k+1} = 0$. Since $x_{k+1} = 0$ and $F(0, 0) = 0$, we have $x_{k+2} = 0$. So, $\underline{\hat{u}}_k^{k+1}$ is a finite-horizon admissible control sequence. Furthermore

$$\begin{aligned} J(x_k, \underline{\hat{u}}_k^{k+1}) &= U(x_k, v_1(x_k)) + U(x_{k+1}, \hat{u}_{k+1}) \\ &= U(x_k, v_1(x_k)) \\ &= V_1(x_k) \end{aligned}$$

since $U(x_{k+1}, \hat{u}_{k+1}) = U(0, 0) = 0$. On the other hand, according to Remark 3.2, we have

$$V_2(x_k) = \min_{\underline{u}_k^{k+1}} \left\{ J\left(x_k, \underline{u}_k^{k+1}\right) : \underline{u}_k^{k+1} \in \mathfrak{A}_{x_k}^{(2)} \right\}.$$

Then, we obtain

$$\begin{aligned} V_2(x_k) &= \min_{\underline{u}_k^{k+1}} \left\{ J\left(x_k, \underline{u}_k^{k+1}\right) : \underline{u}_k^{k+1} \in \mathfrak{A}_{x_k}^{(2)} \right\} \\ &\leq J\left(x_k, \underline{\hat{u}}_k^{k+1}\right) \\ &= V_1(x_k). \end{aligned} \tag{12}$$

Therefore, the theorem holds for $i = 1$.

Assume that the theorem holds for any $i = q$, where $q > 1$. From (11), we have

$$V_q(x_k) = \sum_{j=0}^{q-1} U\left(x_{k+j}, v_{q-j}(x_{k+j})\right).$$

The corresponding finite-horizon admissible control sequence is $\underline{\hat{u}}_k^{k+q-1} = \left\{v_q(x_k), v_{q-1}(x_{k+1}), \ldots, v_1(x_{k+q-1})\right\}$.

For $i = q + 1$, we create a control sequence $\underline{\hat{u}}_k^{k+q} = \left\{v_q(x_k), v_{q-1}(x_{k+1}), \ldots, v_1(x_{k+q-1}), 0\right\}$ with length $q + 1$. Then, the state trajectory under the control of $\underline{\hat{u}}_k^{k+q}$ is x_k, $x_{k+1} = F(x_k, v_q(x_k))$, $x_{k+2} = F(x_{k+1}, v_{q-1}(x_{k+1}))$, …, $x_{k+q} = F(x_{k+q-1}, v_1(x_{k+q-1})) = 0$, $x_{k+q+1} = F(x_{k+q}, 0) = 0$. So, $\underline{\hat{u}}_k^{k+q}$ is a finite-horizon admissible control sequence. The performance index function under this control sequence is

$$\begin{aligned} J(x_k, \underline{\hat{u}}_k^{k+q}) &= U(x_k, v_q(x_k)) + U(x_{k+1}, v_{q-1}(x_{k+1})) \\ &\quad + \cdots + U(x_{k+q-1}, v_1(x_{k+q-1})) + U(x_{k+q}, 0) \\ &= \sum_{j=0}^{q-1} U\left(x_{k+j}, v_{q-j}(x_{k+j})\right) \\ &= V_q(x_k) \end{aligned}$$

since $U(x_{k+q}, 0) = U(0, 0) = 0$.

On the other hand, we have

$$V_{q+1}(x_k) = \min_{\underline{u}_k^{k+q}} \left\{ J\left(x_k, \underline{u}_k^{k+q}\right) : \underline{u}_k^{k+q} \in \mathfrak{A}_{x_k}^{(q+1)} \right\}.$$

Thus, we obtain

$$\begin{aligned} V_{q+1}(x_k) &= \min_{\underline{u}_k^{k+q}} \left\{ J\left(x_k, \underline{u}_k^{k+q}\right) : \underline{u}_k^{k+q} \in \mathfrak{A}_{x_k}^{(q+1)} \right\} \\ &\leq J\left(x_k, \underline{\hat{u}}_k^{k+q}\right) \\ &= V_q(x_k) \end{aligned}$$

which completes the proof. ■

From Theorem 3.1, we know that the performance index function $V_i(x_k) \geq 0$ is a monotonically nonincreasing sequence and is bounded below for iteration index $i = 1, 2, \ldots$. Now, we can derive the following theorem.

Theorem 3.2: Let x_k be an arbitrary state vector. Define the performance index function $V_\infty(x_k)$ as the limit of the iterative function $V_i(x_k)$

$$V_\infty(x_k) = \lim_{i \to \infty} V_i(x_k). \tag{13}$$

Then, we have

$$V_\infty(x_k) = \min_{u_k} \{U(x_k, u_k) + V_\infty(x_{k+1})\}.$$

Proof: Let $\eta_k = \eta(x_k)$ be any admissible control vector. According to Theorem 3.1, for $\forall i$, we have

$$V_\infty(x_k) \leq V_{i+1}(x_k) \leq U(x_k, \eta_k) + V_i(x_{k+1}).$$

Let $i \to \infty$, we have

$$V_\infty(x_k) \leq U(x_k, \eta_k) + V_\infty(x_{k+1})$$

which is true for $\forall \eta_k$. Therefore

$$V_\infty(x_k) \leq \min_{u_k} \{U(x_k, u_k) + V_\infty(x_{k+1})\}. \tag{14}$$

Let $\varepsilon > 0$ be an arbitrary positive number. Since $V_i(x_k)$ is nonincreasing for $i \geq 1$ and $\lim_{i \to \infty} V_i(x_k) = V_\infty(x_k)$, there exists a positive integer p such that

$$V_p(x_k) - \varepsilon \leq V_\infty(x_k) \leq V_p(x_k).$$

From (8), we have

$$\begin{aligned} V_p(x_k) &= \min_{u_k} \{U(x_k, u_k) + V_{p-1}(F(x_k, u_k))\} \\ &= U(x_k, v_p(x_k)) + V_{p-1}(F(x_k, v_p(x_k))). \end{aligned}$$

Hence,

$$\begin{aligned} V_\infty(x_k) &\geq U(x_k, v_p(x_k)) + V_{p-1}(F(x_k, v_p(x_k))) - \varepsilon \\ &\geq U(x_k, v_p(x_k)) + V_\infty(F(x_k, v_p(x_k))) - \varepsilon \\ &\geq \min_{u_k} \{U(x_k, u_k) + V_\infty(x_{k+1})\} - \varepsilon. \end{aligned}$$

Since ε is arbitrary, we have

$$V_\infty(x_k) \geq \min_{u_k} \{U(x_k, u_k) + V_\infty(x_{k+1})\}. \tag{15}$$

Combining (14) and (15), we prove the theorem. ■

Next, we will prove that the iterative performance index function $V_i(x_k)$ converges to the optimal performance index function $J^*(x_k)$ as $i \to \infty$.

Theorem 3.3: Let $V_\infty(x_k)$ be defined in (13). If the system state x_k is controllable, then we have the performance index function $V_\infty(x_k)$ equal to the optimal performance index function $J^*(x_k)$

$$\lim_{i\to\infty} V_i(x_k) = J^*(x_k)$$

where $V_i(x_k)$ is defined in (8).

Proof: According to (3) and (10), we have

$$J^*(x_k) \le \min_{\underline{u}_k^{k+i-1}} \left\{ J(x_k, \underline{u}_k^{k+i-1}) : \underline{u}_k^{k+i-1} \in \mathfrak{A}_{x_k}^{(i)} \right\} = V_i(x_k).$$

Then, let $i \to \infty$, we obtain

$$J^*(x_k) \le V_\infty(x_k). \tag{16}$$

Next, we show that

$$V_\infty(x_k) \le J^*(x_k). \tag{17}$$

For any $\omega > 0$, by the definition of $J^*(x_k)$ in (3), there exists $\underline{\eta}_k \in \mathfrak{A}_{x_k}$ such that

$$J(x_k, \underline{\eta}_k) \le J^*(x_k) + \omega. \tag{18}$$

Suppose that $|\underline{\eta}_k| = p$. Then $\underline{\eta}_k \in \mathfrak{A}_{x_k}^{(p)}$. So, by Theorem 3.1 and (10), we have

$$\begin{aligned} V_\infty(x_k) &\le V_p(x_k) \\ &= \min_{\underline{u}_k^{k+p-1}} \left\{ J(x_k, \underline{u}_k^{k+p-1}) : \underline{u}_k^{k+p-1} \in \mathfrak{A}_{x_k}^{(p)} \right\} \\ &\le J(x_k, \underline{\eta}_k) \\ &\le J^*(x_k) + \omega. \end{aligned}$$

Since ω is chosen arbitrarily, we know that (17) is true. Therefore, from (16) and (17), we prove the theorem. ∎

We can now present the following corollary.

Corollary 3.1: Let the performance index function $V_i(x_k)$ be defined by (8). If the system state x_k is controllable, then the iterative control law $v_i(x_k)$ converges to the optimal control law $u^*(x_k)$, i.e., $\lim_{i\to\infty} v_i(x_k) = u^*(x_k)$. ∎

Remark 3.3: Generally speaking, for the finite-horizon optimal control problems, the optimal performance index function depends not only on state x_k but also on the time left (see [60], [61]). For the finite-horizon optimal control problems with unspecified terminal time, we have proved that the iterative performance index functions converge to the optimal as the iterative index i reaches infinity. Then, the time left is negligible and we say that the optimal performance index function $V(x_k)$ is only a function of the state x_k, which is like the case of infinite-horizon optimal control problems. ∎

By Theorem 3.3 and Corollary 3.1, we know that if x_k is controllable, then, as $i \to \infty$, the iterative performance index function $V_i(x_k)$ converges to the optimal performance index function $J^*(x_k)$ and the iterative control law $v_i(x_k)$ also converges to the optimal control law $u^*(x_k)$. So, it is important to note that for controllable state x_k, the iterative performance index functions $V_i(x_k)$ are well defined for all i under the iterative control law $v_i(x_k)$.

Let $\mathcal{T}_0 = \{0\}$. For $i = 1, 2, \ldots$, define

$$\mathcal{T}_i = \{x_k \in \mathbb{R}^n \mid \exists\, u_k \in \mathbb{R}^m \text{ s.t. } F(x_k, u_k) \in \mathcal{T}_{i-1}\}. \tag{19}$$

Next, we prove the following theorem.

Theorem 3.4: Let $\mathcal{T}_0 = \{0\}$ and $\mathcal{T}_i$ be defined in (19). Then, for $i = 0, 1, \ldots$, we have $\mathcal{T}_i \subseteq \mathcal{T}_{i+1}$.

Proof: We prove the theorem by mathematical induction. First, let $i = 0$. Since $\mathcal{T}_0 = \{0\}$ and $F(0, 0) = 0$, we know that $0 \in \mathcal{T}_1$. Hence, $\mathcal{T}_0 \subseteq \mathcal{T}_1$.

Next, assume that $\mathcal{T}_{i-1} \subseteq \mathcal{T}_i$ holds. Now, if $x_k \in \mathcal{T}_i$, we have $F(x_k, \eta_{i-1}(x_k)) \in \mathcal{T}_{i-1}$ for some $\eta_{i-1}(x_k)$. Hence, $F(x_k, \eta_{i-1}(x_k)) \in \mathcal{T}_i$ by the assumption of $\mathcal{T}_{i-1} \subseteq \mathcal{T}_i$. So, $x_k \in \mathcal{T}_{i+1}$ by (19). Thus, $\mathcal{T}_i \subseteq \mathcal{T}_{i+1}$, which proves the theorem. ∎

According to Theorem 3.4, we have

$$\{0\} = \mathcal{T}_0 \subseteq \mathcal{T}_1 \subseteq \cdots \subseteq \mathcal{T}_{i-1} \subseteq \mathcal{T}_i \subseteq \cdots.$$

We can see that by introducing the sets $\mathcal{T}_i$, $i = 0, 1, \ldots$, the state x_k can be classified correspondingly. According to Theorem 3.4, the properties of the ADP algorithm can be derived in the following theorem.

Theorem 3.5:

1) For any i, $x_k \in \mathcal{T}_i \Leftrightarrow \mathfrak{A}_{x_k}^{(i)} \neq \emptyset \Leftrightarrow V_i(x_k)$ is defined at x_k.
2) Let $\mathcal{T}_\infty = \cup_{i=1}^\infty \mathcal{T}_i$. Then, $x_k \in \mathcal{T}_\infty \Leftrightarrow \mathfrak{A}_{x_k} \neq \emptyset \Leftrightarrow J^*(x_k)$ is defined at $x_k \Leftrightarrow x_k$ is controllable.
3) If $V_i(x_k)$ is defined at x_k, then $V_j(x_k)$ is defined at x_k for every $j \ge i$.
4) $J^*(x_k)$ is defined at x_k if and only if there exists an i such that $V_i(x_k)$ is defined at x_k. ∎

IV. ε-Optimal Control Algorithm

In the previous section, we proved that the iterative performance index function $V_i(x_k)$ converges to the optimal performance index function $J^*(x_k)$ and $J^*(x_k) = \min_{\underline{u}_k}\{J(x_k, \underline{u}_k),\ \underline{u} \in \mathfrak{A}_{x_k}\}$ satisfies the Bellman's equation (4) for any controllable state $x_k \in \mathcal{T}_\infty$.

To obtain the optimal performance index function $J^*(x_k)$, a natural strategy is to run the iterative ADP algorithm (6)–(9) until $i \to \infty$. But unfortunately, it is not practical to do so. In many cases, we cannot find the equality $J^*(x_k) = V_i(x_k)$ for any finite i. That is, for any admissible control sequence $\underline{u}_k$ with finite length, the performance index starting from x_k under the control of $\underline{u}_k$ will be larger than, not equal to, $J^*(x_k)$. On the other hand, by running the iterative ADP algorithm (6)–(9), we can obtain a control vector $v_\infty(x_k)$ and then construct a control sequence $\underline{u}_\infty(x_k) = (v_\infty(x_k), v_\infty(x_{k+1}), \ldots, v_\infty(x_{k+i}), \ldots)$, where $x_{k+1} = F(x_k, v_\infty(x_k)), \ldots, x_{k+i} = F(x_{k+i-1}, v_\infty(x_{k+i-1}))$, ... In general, $\underline{u}_\infty(x_k)$ has infinite length. That is, the controller $v_\infty(x_k)$ cannot control the state to reach the target in finite number of steps. To overcome this difficulty, a new ε-optimal control method using iterative ADP algorithm will be developed in this section.

A. ε-Optimal Control Method

In this section, we will introduce our method of iterative ADP with the consideration of the length of control sequences. For different x_k, we will consider different length i for the

optimal control sequence. For a given error bound $\varepsilon > 0$, the number i will be chosen so that the error between $J^*(x_k)$ and $V_i(x_k)$ is within the bound.

Let $\varepsilon > 0$ be any small number and $x_k \in \mathcal{T}_\infty$ be any controllable state. Let the performance index function $V_i(x_k)$ be defined by (8) and $J^*(x_k)$ be the optimal performance index function. According to Theorem 3.3, given $\varepsilon > 0$, there exists a finite i such that

$$|V_i(x_k) - J^*(x_k)| \leq \varepsilon. \tag{20}$$

We can now give the following definition.

Definition 4.1: Let $x_k \in \mathcal{T}_\infty$ be a controllable state vector. Let $\varepsilon > 0$ be a small positive number. The approximate length of optimal control sequence with respect to ε is defined as

$$K_\varepsilon(x_k) = \min\{i : |V_i(x_k) - J^*(x_k)| \leq \varepsilon\}. \tag{21}$$

Given a small positive number ε, for any state vector x_k, the number $K_\varepsilon(x_k)$ gives a suitable length of control sequence for optimal control starting from x_k. For $x_k \in \mathcal{T}_\infty$, since $\lim_{i\to\infty} V_i(x_k) = J^*(x_k)$, we can always find i such that (20) is satisfied. Therefore, $\{i : |V_i(x_k) - J^*(x_k)| \leq \varepsilon\} \neq \emptyset$ and $K_\varepsilon(x_k)$ is well defined.

We can see that an error ε between $V_i(x_k)$ and $J^*(x_k)$ is introduced into the iterative ADP algorithm, which makes the performance index function $V_i(x_k)$ converge within finite number of iteration steps. In this part, we will show that the corresponding control is also an effective control that drives the performance index function to within error bound ε from its optimal.

From Definition 4.1, we can see that all the states x_k that satisfy (21) can be classified into one set. Motivated by the definition in (19), we can further classify this set using the following definition.

Definition 4.2: Let ε be a positive number. Define $\mathcal{T}_0^{(\varepsilon)} = \{0\}$ and for $i = 1, 2, \ldots$, define

$$\mathcal{T}_i^{(\varepsilon)} = \{x_k \in \mathcal{T}_\infty : K_\varepsilon(x_k) \leq i\}. \quad \blacksquare$$

Accordingly, when $x_k \in \mathcal{T}_i^{(\varepsilon)}$, to find the optimal control sequence which has performance index less than or equal to $J^*(x_k) + \varepsilon$, one only needs to consider the control sequences $\underline{u}_k$ with length $|\underline{u}_k| \leq i$. The sets $\mathcal{T}_i^{(\varepsilon)}$ have the following properties.

Theorem 4.1: Let $\varepsilon > 0$ and $i = 0, 1, \ldots$. Then:

1) $x_k \in \mathcal{T}_i^{(\varepsilon)}$ if and only if $V_i(x_k) \leq J^*(x_k) + \varepsilon$;
2) $\mathcal{T}_i^{(\varepsilon)} \subseteq \mathcal{T}_i$;
3) $\mathcal{T}_i^{(\varepsilon)} \subseteq \mathcal{T}_{i+1}^{(\varepsilon)}$;
4) $\cup_i \mathcal{T}_i^{(\varepsilon)} = \mathcal{T}_\infty$;
5) If $\varepsilon > \delta > 0$, then $\mathcal{T}_i^{(\varepsilon)} \supseteq \mathcal{T}_i^{(\delta)}$.

Proof:

1) Let $x_k \in \mathcal{T}_i^{(\varepsilon)}$. By Definition 4.2, $K_\varepsilon(x_k) \leq i$. Let $j = K_\varepsilon(x_k)$. Then, $j \leq i$ and by Definition 4.1, $|V_j(x_k) - J^*(x_k)| \leq \varepsilon$. So, $V_j(x_k) \leq J^*(x_k) + \varepsilon$. By Theorem 3.1, $V_i(x_k) \leq V_j(x_k) \leq J^*(x_k) + \varepsilon$. On the other hand, if $V_i(x_k) \leq J^*(x_k) + \varepsilon$, then $|V_i(x_k) - J^*(x_k)| \leq \varepsilon$. So, $K_\varepsilon(x_k) = \min\{j : |V_j(x_k) - J^*(x_k)| \leq \varepsilon\} \leq i$, which implies that $x_k \in \mathcal{T}_i^{(\varepsilon)}$.
2) If $x_k \in \mathcal{T}_i^{(\varepsilon)}$, $K_\varepsilon(x_k) \leq i$ and $|V_i(x_k) - J^*(x_k)| \leq \varepsilon$. So, $V_i(x_k)$ is defined at x_k. According to Theorem 3.5 1), we have $x_k \in \mathcal{T}_i$. Hence, $\mathcal{T}_i^{(\varepsilon)} \subseteq \mathcal{T}_i$.
3) If $x_k \in \mathcal{T}_i^{(\varepsilon)}$, $K_\varepsilon(x_k) \leq i < i + 1$. So, $x_k \in \mathcal{T}_{i+1}^{(\varepsilon)}$. Thus, $\mathcal{T}_i^{(\varepsilon)} \subseteq \mathcal{T}_{i+1}^{(\varepsilon)}$.
4) Obviously, $\cup_i \mathcal{T}_i^{(\varepsilon)} \subseteq \mathcal{T}_\infty$ since $\mathcal{T}_i^{(\varepsilon)}$ are subsets of $\mathcal{T}_\infty$. For any $x_k \in \mathcal{T}_\infty$, let $p = K_\varepsilon(x_k)$. Then, $x_k \in \mathcal{T}_p^{(\varepsilon)}$. So, $x_k \in \cup_i \mathcal{T}_i^{(\varepsilon)}$. Hence, $\mathcal{T}_\infty \subseteq \cup_i \mathcal{T}_i^{(\varepsilon)} \subseteq \mathcal{T}_\infty$, and we obtain, $\cup_i \mathcal{T}_i^{(\varepsilon)} = \mathcal{T}_\infty$.
5) If $x_k \in \mathcal{T}_i^{(\delta)}$, $V_i(x_k) \leq J^*(x_k) + \delta$ by part 1) of this theorem. Clearly, $V_i(x_k) \leq J^*(x_k) + \varepsilon$ since $\delta < \varepsilon$. This implies that $x_k \in \mathcal{T}_i^{(\varepsilon)}$. Therefore, $\mathcal{T}_i^{(\varepsilon)} \supseteq \mathcal{T}_i^{(\delta)}$.

$\blacksquare$

According to Theorem 4.1 1), $\mathcal{T}_i^{(\varepsilon)}$ is just the region where $V_i(x_k)$ is close to $J^*(x_k)$ with error less than ε. This region is a subset of $\mathcal{T}_i$ according to Theorem 4.1 2). As stated in Theorem 4.1 3), when i is large, the set $\mathcal{T}_i^{(\varepsilon)}$ is also large. That means, when i is large, we have a large region where we can use $V_i(x_k)$ as the approximation of $J^*(x_k)$ under certain error. On the other hand, we claim that if x_k is far away from the origin, we have to choose a long control sequence to approximate the optimal control sequence. Theorem 4.1 4) means that for every controllable state $x_k \in \mathcal{T}_\infty$, we can always find a suitable control sequence with length i to approximate the optimal control. The size of the set $\mathcal{T}_i^{(\varepsilon)}$ depends on the value of ε. A smaller value of ε gives a smaller set $\mathcal{T}_i^{(\varepsilon)}$, which is indicated by Theorem 4.1 5).

Let $x_k \in \mathcal{T}_\infty$ be an arbitrary controllable state. If $x_k \in \mathcal{T}_i^{(\varepsilon)}$, the iterative performance index function satisfies (20) under the control $v_i(x_k)$, we call this control the ε-optimal control and denote it as $\mu_\varepsilon^*(x_k)$

$$\mu_\varepsilon^*(x_k) = v_i(x_k) = \arg\min_{u_k} \{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\}. \tag{22}$$

We have the following corollary.

Corollary 4.1: Let $\mu_\varepsilon^*(x_k)$ be expressed in (22), which makes the performance index function satisfy (20) for $x_k \in \mathcal{T}_i^{(\varepsilon)}$. Then, for any $x_k' \in \mathcal{T}_i^{(\varepsilon)}$, $\mu_\varepsilon^*(x_k')$ guarantees

$$|V_i(x_k') - J^*(x_k')| \leq \varepsilon. \tag{23}$$

Proof: The corollary can be proved by contradiction. Assume that the conclusion is not true. Then, the inequality (23) is false under the control $\mu_\varepsilon^*(\cdot)$ for some $x_k'' \in \mathcal{T}_i^{(\varepsilon)}$.

As $\mu_\varepsilon^*(x_k)$ makes the performance index function satisfy (20) for $x_k \in \mathcal{T}_i^{(\varepsilon)}$, we have $K_\varepsilon(x_k) \leq i$. Using the ε-optimal control law $\mu_\varepsilon^*(\cdot)$ at the state x_k'', according to the assumption, we have $|V_i(x_k'') - J^*(x_k'')| > \varepsilon$. Then, $K_\varepsilon(x_k'') > i$ and $x_k'' \notin \mathcal{T}_i^{(\varepsilon)}$. It is in contradiction with the assumption $x_k'' \in \mathcal{T}_i^{(\varepsilon)}$. Therefore, the assumption is false and (23) holds for any $x_k' \in \mathcal{T}_i^{(\varepsilon)}$. $\blacksquare$

Remark 4.1: Corollary 4.1 is very important for neural network implementation of the iterative ADP algorithm. It shows that we do not need to obtain the optimal control law by searching the entire subset $\mathcal{T}_i^{(\varepsilon)}$. Instead, we can just find one point of $\mathcal{T}_i^{(\varepsilon)}$, i.e., $x_k \in \mathcal{T}_i^{(\varepsilon)}$, to obtain the ε-optimal control

$\mu_\varepsilon^*(x_k)$ which will be effective for any other state $x_k' \in \mathcal{T}_i^{(\varepsilon)}$. This property not only makes the computational complexity much reduced but also makes the optimal control law easily obtained using neural networks. ∎

Theorem 4.2: Let $x_k \in \mathcal{T}_i^{(\varepsilon)}$ and let $\mu_\varepsilon^*(x_k)$ be expressed in (22). Then, $F(x_k, \mu_\varepsilon^*(x_k)) \in \mathcal{T}_{i-1}^{(\varepsilon)}$. In other words, if $K_\varepsilon(x_k) = i$, then $K_\varepsilon(F(x_k, \mu_\varepsilon^*(x_k))) \le i - 1$.

Proof: Since $x_k \in \mathcal{T}_i^{(\varepsilon)}$, by Theorem 4.1 1), we know that

$$V_i(x_k) \le J^*(x_k) + \varepsilon. \tag{24}$$

According to (8) and (22), we have

$$V_i(x_k) = U(x_k, \mu_\varepsilon^*(x_k)) + V_{i-1}(F(x_k, \mu_\varepsilon^*(x_k))). \tag{25}$$

Combining (24) and (25), we have

$$\begin{aligned} V_{i-1}(F(x_k, \mu_\varepsilon^*(x_k))) &= V_i(x_k) - U(x_k, \mu_\varepsilon^*(x_k)) \\ &\le J^*(x_k) + \varepsilon - U(x_k, \mu_\varepsilon^*(x_k)). \end{aligned} \tag{26}$$

On the other hand, we have

$$J^*(x_k) \le U(x_k, \mu_\varepsilon^*(x_k)) + J^*(F(x, \mu_\varepsilon^*(x_k))). \tag{27}$$

Putting (27) into (26), we obtain

$$V_{i-1}(F(x_k, \mu_\varepsilon^*(x_k))) \le J^*(F(x_k, \mu_\varepsilon^*(x_k))) + \varepsilon.$$

By Theorem 4.1 1), we have

$$F(x_k, \mu_\varepsilon^*(x_k)) \in \mathcal{T}_{i-1}^{(\varepsilon)}. \tag{28}$$

So, if $K_\varepsilon(x_k) = i$, we know that $x_k \in \mathcal{T}_i^{(\varepsilon)}$ and $F(x, \mu_\varepsilon^*(x_k)) \in \mathcal{T}_{i-1}^{(\varepsilon)}$ according to (28). Therefore, we have

$$K_\varepsilon(F(x_k, \mu_\varepsilon^*(x_k))) \le i - 1$$

which proves the theorem. ∎

Remark 4.2: From Theorem 4.2, we can see that the parameter $K_\varepsilon(x_k)$ gives an important property of the finite-horizon ADP algorithm. It not only gives an optimal condition of the iteration process, but also gives an optimal number of control steps for the finite-horizon ADP algorithm. For example, if $|V_i(x_k) - J^*(x_k)| \le \varepsilon$ for small ε, then we have $V_i(x_k) \approx J^*(x_k)$. According to Theorem 4.2, we can get $N = k + i$, where N is the number of control steps to drive the system to zero. The whole control sequence $\underline{u}_0^{N-1}$ may not be ε-optimal, but the control sequence $\underline{u}_k^{N-1}$ is ε-optimal control sequence. If $k = 0$, we have $N = K_\varepsilon(x_0) = i$. Under this condition, we say that the iteration index $K_\varepsilon(x_0)$ denotes the number of ε-optimal control steps. ∎

Corollary 4.2: Let $\mu_\varepsilon^*(x_k)$ be expressed in (22), which makes the performance index function satisfy (20) for $x_k \in \mathcal{T}_i^{(\varepsilon)}$. Then, for any $x_k' \in \mathcal{T}_j^{(\varepsilon)}$, where $0 \le j \le i$, $\mu_\varepsilon^*(x_k')$ guarantees

$$|V_i(x_k') - J^*(x_k')| \le \varepsilon. \tag{29}$$

Proof: The proof is similar to Corollary 4.1 and is omitted here. ∎

Remark 4.3: Corollary 4.2 shows that the ε-optimal control $\mu_\varepsilon^*(x_k)$ obtained for $\forall\, x_k \in \mathcal{T}_i^{(\varepsilon)}$ is effective for any state $x_k' \in \mathcal{T}_j^{(\epsilon)}$, where $0 \le j \le i$. This means that for $\forall\, x_k' \in \mathcal{T}_j^{(\varepsilon)}$, $0 \le j \le i$, we can use a same ε-optimal control $\mu_\varepsilon^*(x_k')$ to control the system. ∎

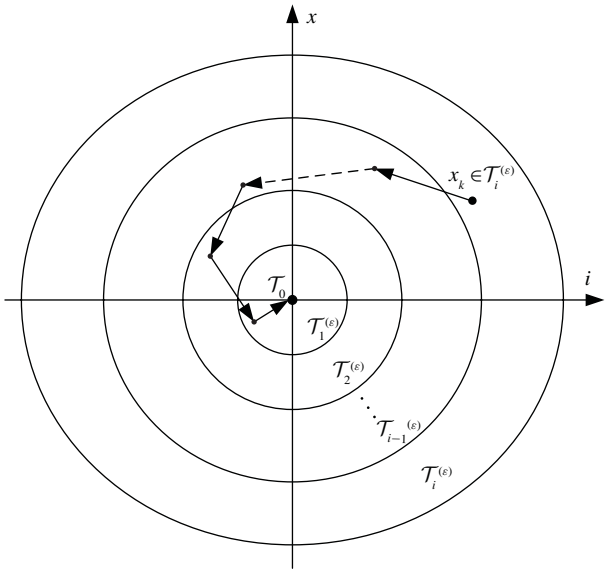

Fig. 1. Control process of the controllable sate $x_k \in \mathcal{T}_i^{(\varepsilon)}$ using iterative ADP algorithm.

B. ε-Optimal Control Algorithm

According to Theorem 4.1 3) and Corollary 4.1, the ε-optimal control $\mu_\varepsilon^*(x_k)$ obtained for an $x_k \in \mathcal{T}_i^{(\varepsilon)}$ is effective for any state $x_k' \in \mathcal{T}_{i-1}^{(\varepsilon)}$ (which is also stated in Corollary 4.2). That is to say, in order to obtain effective ε-optimal control, the iterative ADP algorithm only needs to run at some state $x_k \in \mathcal{T}_\infty$. In order to obtain an effective ε-optimal control law $\mu_\varepsilon^*(x_k)$, we should choose the state $x_k \in \mathcal{T}_i^{(\varepsilon)} \backslash \mathcal{T}_{i-1}^{(\varepsilon)}$ for each i to run the iterative ADP algorithm. The control process using iterative ADP algorithm is illustrated in Fig. 1.

From the iterative ADP algorithm (6)–(9), we can see that for any state $x_k \in \mathbb{R}^n$, there exits a control $u_k \in \mathbb{R}^m$ that drives the system to zero in one step. In other words, for $\forall\, x_k \in \mathbb{R}^n$, there exists a control $u_k \in \mathbb{R}^m$ such that $x_{k+1} = F(x_k, u_k) = 0$ holds. A large class of systems possesses this property, for example, all linear systems of the type $x_{k+1} = Ax_k + Bu_k$ when B is invertible and the affine nonlinear systems with the type $x_{k+1} = f(x_k) + g(x_k)u_k$ when the inverse of $g(x_k)$ exists. But there are also other classes of systems for which there does not exist any control $u_k \in \mathbb{R}^m$ that drives the state to zero in one step for some $x_k \in \mathbb{R}^n$, i.e., $\exists x_k \in \mathbb{R}^n$ such that $F(x_k, u_k) = 0$ is not possible for $\forall\, u_k \in \mathbb{R}^m$. In the following part, we will discuss the situation where $F(x_k, u_k) \ne 0$ for some $x_k \in \mathbb{R}^m$.

Since x_k is controllable, there exists a finite-horizon admissible control sequence $\underline{u}_k^{k+i-1} = \{u_k, u_{k+1}, \ldots, u_{k+i-1}\} \in \mathfrak{A}_{x_k}^{(i)}$ that makes $x^{(f)}\big(x_k, \underline{u}_k^{k+i-1}\big) = x_{k+i} = 0$. Let $N = k + i$ be the terminal time. Assume that for $k+1, k+2, \ldots, N-1$, the optimal control sequence $\underline{u}_{k+1}^{(N-1)*} = \{u_{k+1}^*, u_{k+2}^*, \ldots, u_{N-1}^*\} \in \mathfrak{A}_{x_{k+1}}^{(N-k-1)}$ has been determined. Denote the performance index function for x_{k+1} as $J\big(x_{k+1}, \underline{u}_{k+1}^{(N-1)*}\big) = V_0(x_{k+1})$. Now, we use the iterative ADP algorithm to determine the optimal control sequence for the state x_k.

The performance index function for $i = 1$ is computed as

$$V_1(x_k) = U(x_k, v_1(x_k)) + V_0(F(x_k, v_1(x_k))) \tag{30}$$

where

$$v_1(x_k) = \arg\min_{u_k}\{U(x_k, u_k) + V_0(F(x_k, u_k))\}. \tag{31}$$

Note that the initial condition used in the above expression is the performance index function V_0, which is obtained previously for x_{k+1} and now applied at $F(x_k, u_k)$. For $i = 2, 3, 4, \ldots$, the iterative ADP algorithm will be implemented as follows:

$$V_i(x_k) = U(x_k, v_i(x_k)) + V_{i-1}(F(x_k, v_i(x_k))) \tag{32}$$

where

$$v_i(x_k) = \arg\min_{u_k}\{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\}. \tag{33}$$

Theorem 4.3: Let x_k be an arbitrary controllable state vector. Then, the performance index function $V_i(x_k)$ obtained by (30)–(33) is a monotonically nonincreasing sequence for $\forall\, i \geq 0$, i.e., $V_{i+1}(x_k) \leq V_i(x_k)$ for $\forall\, i \geq 0$.

Proof: It can easily be proved following the proof of Theorem 3.1, and the proof is omitted here. ■

Theorem 4.4: Let the performance index function $V_i(x_k)$ be defined by (32). If the system state x_k is controllable, then the performance index function $V_i(x_k)$ obtained by (30)–(33) converges to the optimal performance index function $J^*(x_k)$ as $i \to \infty$

$$\lim_{i\to\infty} V_i(x_k) = J^*(x_k).$$

Proof: This theorem can be proved following similar steps to the proof of Theorem 3.3 and the proof is omitted here. ■

Remark 4.4: We can see that the iterative ADP algorithm (30)–(33) is an expansion from of the previous one (6)–(9). So, the properties of the iterative ADP algorithm (6)–(9) is also effective for the current one (30)–(33). But there also exist differences. From Theorem 3.1, we can see that $V_{i+1}(x_k) \leq V_i(x_k)$ for all $i \geq 1$, which means that $V_1(x_k) = \max\{V_i(x_k) : i = 0, 1, \ldots\}$, while Theorem 4.3 shows that $V_{i+1}(x_k) \leq V_i(x_k)$ for all $i \geq 0$, which means that $V_0(x_k) = \max\{V_i(x_k) : i = 0, 1, \ldots\}$. This difference is caused by the difference of the initial conditions of the two iterative ADP algorithms. In the previous iterative ADP algorithm (6)–(9), it begins with the initial performance index function $V_0(x_k) = 0$ since $F(x_k, u_k) = 0$ can be solved, while in the current iterative ADP algorithm (30)–(33), it begins with the performance index function V_0 for the state x_{k+1} which is determined previously. This also causes the difference between the proof of Theorems 3.1 and 3.3 and the corresponding results in Theorems 4.3 and 4.4. But the difference of the initial conditions of the iterative performance index function does not affect the convergence property of the two iterative ADP algorithms. ■

For the iterative ADP algorithm, the optimal criterion (20) is very difficult to verify because the optimal performance index function $J^*(x_k)$ is unknown in general. So, an equivalent criterion is established to replace (20).

If $|V_i(x_k) - J^*(x_k)| \leq \varepsilon$ holds, we have $V_i(x_k) \leq J^*(x_k) + \varepsilon$ and $J^*(x_k) \leq V_{i+1}(x_k) \leq V_i(x_k)$. These imply that

$$0 \leq V_i(x_k) - V_{i+1}(x_k) \leq \varepsilon \tag{34}$$

or

$$|V_i(x_k) - V_{i+1}(x_k)| \leq \varepsilon.$$

On the other hand, according to Theorem 4.4, $|V_i(x_k) - V_{i+1}(x_k)| \to 0$ implies that $V_i(x_k) \to J^*(x_k)$. Therefore, for any given small ε, if $|V_i(x_k) - V_{i+1}(x_k)| \leq \varepsilon$ holds, we have $|V_i(x_k) - J^*(x_k)| \leq \varepsilon$ holds if i is sufficiently large.

We will use (34) as the optimal criterion instead of the optimal criterion (20).

Let $\widehat{\underline{u}}_0^{K-1} = (u_0, u_1, \ldots, u_{K-1})$ be an arbitrary finite-horizon admissible control sequence and the corresponding state sequence be $\widehat{\underline{x}}_0^{K} = (x_0, x_1, \ldots, x_K)$ where $x_K = 0$.

We can see that the initial control sequence $\widehat{\underline{u}}_0^{K-1} = (u_0, u_1, \ldots, u_{K-1})$ may not be optimal, which means that the initial number of control steps K may not be optimal. So, the iterative ADP algorithm must complete two kinds of optimization: one is to optimize the number of control steps; and the other is to optimize the control law. In the following, we will show how the number of control steps and the control law are both optimized in the iterative ADP algorithm simultaneously.

For the state x_{K-1}, we have $F(x_{K-1}, u_{K-1}) = 0$. Then, we run the iterative ADP algorithm (6)–(9) at x_{K-1} as follows. The performance index function for $i = 1$ is computed as

$$\begin{aligned} V_1^1(x_{K-1}) &= \min_{u_{K-1}}\{U(x_{K-1}, u_{K-1}) + V_0(F(x_{K-1}, u_{K-1}))\} \\ &\qquad \text{s.t.} \quad F(x_{K-1}, u_{K-1}) = 0 \\ &= U(x_{K-1}, v_1^1(x_{K-1})) \end{aligned} \tag{35}$$

where

$$\begin{aligned} v_1^1(x_{K-1}) = \arg\min_{u_{K-1}} & U(x_{K-1}, u_{K-1}) \\ \text{s.t.} \quad & F(x_{K-1}, u_{K-1}) = 0 \end{aligned} \tag{36}$$

and $V_0(F(x_{K-1}, u_{K-1})) = 0$. The iterative ADP algorithm will be implemented as follows for $i = 2, 3, 4, \ldots$

$$\begin{aligned} V_i^1(x_{K-1}) = U\left(x_{K-1}, v_i^1(x_{K-1})\right) \\ + V_{i-1}^1\left(F\left(x_{K-1}, v_i^1(x_{K-1})\right)\right) \end{aligned} \tag{37}$$

where

$$\begin{aligned} v_i^1(x_{K-1}) = \arg\min_{u_{K-1}} \Big\{ & U(x_{K-1}, u_{K-1}) \\ & + V_{i-1}^1\left(F(x_{K-1}, u_{K-1})\right) \Big\} \end{aligned} \tag{38}$$

until the inequality

$$\left| V_{l_1}^1(x_{K-1}) - V_{l_1+1}^1(x_{K-1}) \right| \leq \varepsilon \tag{39}$$

is satisfied for $l_1 > 0$. This means that $x_{K-1} \in \mathcal{T}_{l_1}^{(\varepsilon)}$ and the optimal number of control steps is $K_\varepsilon(x_{K-1}) = l_1$.

Considering x_{K-2}, we have $F(x_{K-2}, u_{K-2}) = x_{K-1}$. Put x_{K-2} into (39). If $\left| V_{l_1}^1(x_{K-2}) - V_{l_1+1}^1(x_{K-2}) \right| \leq \varepsilon$ holds, then

according to Theorem 4.1 1), we know that $x_{K-2} \in \mathcal{T}_{l_1}^{(\varepsilon)}$. Otherwise, if $x_{K-2} \notin \mathcal{T}_{l_1}^{(\varepsilon)}$, we will run the iterative ADP algorithm as follows. Using the performance index function $V_{l_1}^1$ as the initial condition, we compute for $i = 1$

$$V_1^2(x_{K-2}) = U\left(x_{K-2}, v_1^2(x_{K-2})\right) + V_{l_1}^1\left(F\left(x_{K-2}, v_1^2(x_{K-2})\right)\right) \tag{40}$$

where

$$v_1^2(x_{K-2}) = \arg\min_{u_{K-2}} \left\{ U x_{K-2}, u_{K-2}) + V_{l_1}^1(F(x_{K-2}, u_{K-2}))\right\}. \tag{41}$$

The iterative ADP algorithm will be implemented as follows for $i = 2, 3, 4, \ldots$

$$V_i^2(x_{K-2}) = U\left(x_{K-2}, v_i^2(x_{K-2})\right) + V_{i-1}^2\left(F\left(x_{K-2}, v_i^2(x_{K-2})\right)\right) \tag{42}$$

where

$$v_i^2(x_{K-2}) = \arg\min_{u_{K-2}} \left\{ U(x_{K-2}, u_{K-2}) + V_{i-1}^2(F(x_{K-2}, u_{K-2}))\right\} \tag{43}$$

until the inequality

$$\left| V_{l_2}^2(x_{K-2}) - V_{l_2+1}^2(x_{K-2}) \right| \le \varepsilon \tag{44}$$

is satisfied for $l_2 > 0$. We can then obtain that $x_{K-2} \in \mathcal{T}_{l_2}^{(\varepsilon)}$ and the optimal number of control steps is $K_\varepsilon(x_{K-2}) = l_2$.

Next, assume that $j \ge 2$ and $x_{K-j+1} \in \mathcal{T}_{l_{j-1}}^{(\varepsilon)}$

$$\left| V_{l_{j-1}}^{j-1}(x_{K-j+1}) - V_{l_{j-1}+1}^{j-1}(x_{K-j+1}) \right| \le \varepsilon \tag{45}$$

holds. Considering x_{K-j}, we have $F(x_{K-j}, u_{K-j}) = x_{K-j+1}$. Putting x_{K-j} into (45) and, if

$$\left| V_{l_{j-1}}^{j-1}(x_{K-j}) - V_{l_{j-1}+1}^{j-1}(x_{K-j}) \right| \le \varepsilon \tag{46}$$

holds, then we know that $x_{K-j} \in \mathcal{T}_{l_{j-1}}^{(\varepsilon)}$. Otherwise, if $x_{K-j} \notin \mathcal{T}_{l_{j-1}}^{(\varepsilon)}$, then we run the iterative ADP algorithm as follows. Using the performance index function $V_{l_{j-1}}^{j-1}$ as the initial condition, we compute for $i = 1$

$$V_1^j(x_{K-j}) = U\left(x_{K-j}, v_1^j(x_{K-j})\right) + V_{l_{j-1}}^{j-1}\left(F\left(x_{K-j}, v_1^j(x_{K-j})\right)\right) \tag{47}$$

where

$$v_1^j(x_{K-j}) = \arg\min_{u_{K-j}} \left\{ U(x_{K-j}, u_{K-j}) + V_{l_{j-1}}^{j-1}(F(x_{K-j}, u_{K-j}))\right\}. \tag{48}$$

The iterative ADP algorithm will be implemented as follows for $i = 2, 3, 4, \ldots$

$$V_i^j(x_{K-j}) = U\left(x_{K-j}, v_i^j(x_{K-j})\right) + V_{i-1}^j\left(F\left(x_{K-j}, v_i^j(x_{K-j})\right)\right) \tag{49}$$

where

$$v_i^j(x_{K-j}) = \arg\min_{u_{K-j}} \left\{ U(x_{K-j}, u_{K-j}) + V_{i-1}^j(F(x_{K-j}, u_{K-j}))\right\} \tag{50}$$

until the inequality

$$\left| V_{l_j}^j(x_{K-j}) - V_{l_j+1}^j(x_{K-j}) \right| \le \varepsilon \tag{51}$$

is satisfied for $l_j > 0$. We can then obtain that $x_{K-j} \in \mathcal{T}_{l_j}^{(\varepsilon)}$ and the optimal number of control steps is $K_\varepsilon(x_{K-j}) = l_j$.

Finally, considering x_0, we have $F(x_0, u_0) = x_1$. If

$$\left| V_{l_{K-1}}^{K-1}(x_0) - V_{l_{K-1}+1}^{K-1}(x_0) \right| \le \varepsilon$$

holds, then we know that $x_0 \in \mathcal{T}_{l_{K-1}}^{(\varepsilon)}$. Otherwise, if $x_0 \notin \mathcal{T}_{l_{K-1}}^{(\varepsilon)}$, then we run the iterative ADP algorithm as follows, Using the performance index function $V_{l_{K-1}}^{K-1}$ as the initial condition, we compute for $i = 1$

$$V_1^K(x_0) = U\left(x_0, v_1^K(x_0)\right) + V_{l_{K-1}}^{K-1}\left(F\left(x_0, v_1^K(x_0)\right)\right) \tag{52}$$

where

$$v_1^K(x_0) = \arg\min_{u_0} \left\{ U(x_0, u_0) + V_{l_{K-1}}^{K-1}(F(x_0, u_0))\right\}. \tag{53}$$

The iterative ADP algorithm will be implemented as follows for $i = 2, 3, 4, \ldots$

$$V_i^K(x_0) = U\left(x_0, v_i^K(x_0)\right) + V_{i-1}^K\left(F\left(x_0, v_i^K(x_0)\right)\right) \tag{54}$$

where

$$v_i^K(x_0) = \arg\min_{u_0} \left\{ U(x_0, u_0) + V_{i-1}^K(F(x_0, u_0))\right\} \tag{55}$$

until the inequality

$$\left| V_{l_K}^K(x_0) - V_{l_K+1}^K(x_0) \right| \le \varepsilon \tag{56}$$

is satisfied for $l_K > 0$. Therefore, we can obtain that $x_0 \in \mathcal{T}_{l_K}^{(\varepsilon)}$ and the optimal number of control steps is $K_\varepsilon(x_0) = l_K$.

Starting from the initial state x_0, the optimal number of control steps is l_K according to our ADP algorithm.

Remark 4.5: For the case where there are some $x_k \in \mathbb{R}^n$, there does not exit a control $u_k \in \mathbb{R}^m$ that drives the system to zero in one step, and the computational complexity of the iterative ADP algorithm is much related to the original finite-horizon admissible control sequence $\underline{\widehat{u}}_0^{K-1}$. First, we repeat the iterative ADP algorithm at x_{K-1}, x_{K-2}, ..., x_1, x_0, respectively. It is related to the control steps K of $\underline{\widehat{u}}_0^{K-1}$. If K is large, it means that $\underline{\widehat{u}}_0^{K-1}$ takes a large number of control steps to drive the initial state x_0 to zero and then the number of times needed to repeat the iterative ADP algorithm will be large. Second, the computational complexity is also related to the quality of control results of $\underline{\widehat{u}}_0^{K-1}$. If $\underline{\widehat{u}}_0^{K-1}$ is close to the optimal control sequence $\underline{u}_0^{(N-1)*}$, then it will take less computation to make (51) hold for each j.

C. Summary of the ε-Optimal Control Algorithm

Now, we summarize the iterative ADP algorithm as follows.

Step 1: Choose an error bound ε and choose randomly an array of initial states x_0.

Step 2: Obtain an initial finite-horizon admissible control sequence $\widehat{\underline{u}}_0^{K-1} = (u_0, u_1, \ldots, u_{K-1})$ and obtain the corresponding state sequence $\widehat{\underline{x}}_0^{K} = (x_0, x_1, \ldots, x_K)$, where $x_K = 0$.

Step 3: For the state x_{K-1} with $F(x_{K-1}, u_{K-1}) = 0$, run the iterative ADP algorithm (35)–(38) at x_{K-1} until (39) holds.

Step 4: Record $V_{l_1}^1(x_{K-1})$, $v_{l_1}^1(x_{K-1})$ and $K_\varepsilon(x_{K-1}) = l_1$.

Step 5: For $j = 2, 3, \ldots, K$, if for x_{K-j}, the inequality (46) holds, go to Step 7; otherwise, go to Step 6.

Step 6: Using the performance index function $V_{l_{j-1}}^{j-1}$ as the initial condition, run the iterative ADP algorithm (47)–(50) until (51) is satisfied.

Step 7: If $j = K$, then we have obtained the optimal performance index function $V^*(x_0) = V_{l_K}^K(x_0)$, the law of the optimal control sequence $u^*(x_0) = v_{l_K}^K(x_0)$ and the number of optimal control steps $K_\varepsilon(x_0) = l_K$; otherwise, set $j = j + 1$, and go to Step 5.

Step 8: Stop.

V. Simulation Study

To evaluate the performance of our iterative ADP algorithm, we choose two examples with quadratic utility functions for numerical experiments.

Example 5.1: Our first example is chosen from [57]. We consider the following nonlinear system:

$$x_{k+1} = f(x_k) + g(x_k)u_k$$

where $x_k = [x_{1k} \quad x_{2k}]^T$ and $u_k = [u_{1k} \quad u_{2k}]^T$ are the state and control variables, respectively. The system functions are given as

$$f(x_k) = \begin{bmatrix} 0.2x_{1k}\exp(x_{2k}^2) \\ 0.3x_{2k}^3 \end{bmatrix}, \qquad g(x_k) = \begin{bmatrix} -0.2 & 0 \\ 0 & -0.2 \end{bmatrix}.$$

The initial state is $x_0 = [1 \quad -1]^T$. The performance index function is in quadratic form with finite-time horizon expressed as

$$J\left(x_0, \underline{u}_0^{N-1}\right) = \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k\right)$$

where the matrix $Q = R = I$ and I denotes the identity matrix with suitable dimensions.

The error bound of the iterative ADP is chosen as $\varepsilon = 10^{-5}$. Neural networks are used to implement the iterative ADP algorithm and the neural network structure can be seen in [32] and [57]. The critic network and the action network are chosen as three-layer backpropagation (BP) neural networks with the structures of 2–8–1 and 2–8–2, respectively. The model network is also chosen as a three-layer BP neural network with the structure of 4–8–2. The critic network is used to approximate the iterative performance index functions, which are expressed by (35), (37), (40), (42), (47), (49), (52), and (54). The action network is used to approximate the optimal control laws, which are expressed by (36), (38), (41), (43), (48), (50), (53), and (55). The training rules of the neural networks can be seen in [50]. For each iterative step, the critic network and the action network are trained for 1000 iteration steps using the learning rate of $\alpha = 0.05$ so that the neural network training error becomes less than 10^{-8}. Enough iteration steps should be implemented to guarantee the iterative performance index functions and the control law to converge sufficiently. We let the algorithm run for 15 iterative steps to obtain the optimal performance index function and optimal control law. The convergence curve of the performance index function is shown in Fig. 2(a). Then, we apply the optimal control law to the system for $T_f = 10$ time steps and obtain the following results. The ε-optimal control trajectories are shown in Fig. 2(b) and the corresponding state curves are shown in Fig. 2(c) and (d).

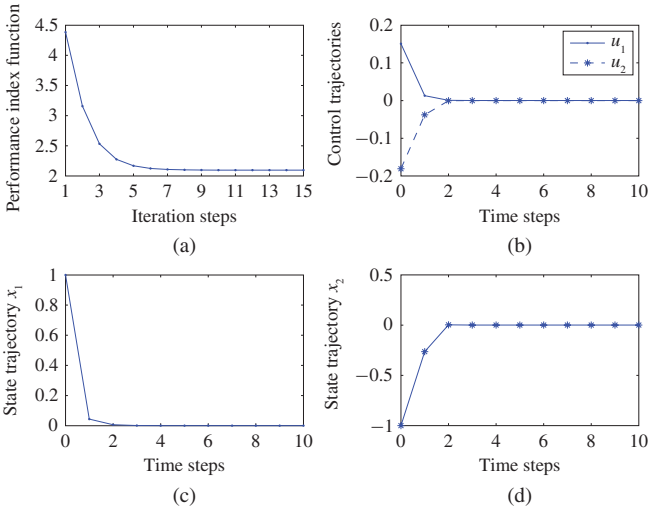

Fig. 2. Simulation results for Example 1. (a) Convergence of performance index function. (b) ε-optimal control vectors. (c) and (d) Corresponding state trajectories.

After seven steps of iteration, we have $|V_6(x_0) - V_7(x_0)| \leq 10^{-5} = \varepsilon$. Then, we obtain the optimal number of control steps $K_\varepsilon(x_0) = 6$. We can see that after six time steps, the state variable becomes $x_6 = [0.912 \times 10^{-6}, \ 0.903 \times 10^{-7}]^T$. The entire computation process takes about 10 s before satisfactory results are obtained. ■

Example 5.2: The second example is chosen from [62] with some modifications. We consider the following system:

$$x_{k+1} = F(x_k, u_k) = x_k + \sin\left(0.1x_k^2 + u_k\right) \tag{57}$$

where $x_k, u_k \in \mathbb{R}$, and $k = 0, 1, 2, \ldots$. The performance index function is defined as in Example 5.1 with $Q = R = 1$. The initial state is $x_0 = 1.5$. Since $F(0, 0) = 0$, $x_k = 0$ is an equilibrium state of system (57). But since $(\partial F(x_k, u_k)/\partial x_k)(0, 0) = 1$, system (57) is marginally stable at $x_k = 0$ and the equilibrium $x_k = 0$ is not attractive.

We can see that for the fixed initial state x_0, there does not exist a control $u_0 \in \mathbb{R}$ that makes $x_1 = F(x_0, u_0) = 0$. The error bound of the iterative ADP algorithm is chosen as $\varepsilon = 10^{-4}$. The critic network, the action network, and the model network are chosen as three-layer BP neural networks with the structures of 1–3–1, 1–3–1, and 2–4–1, respectively.

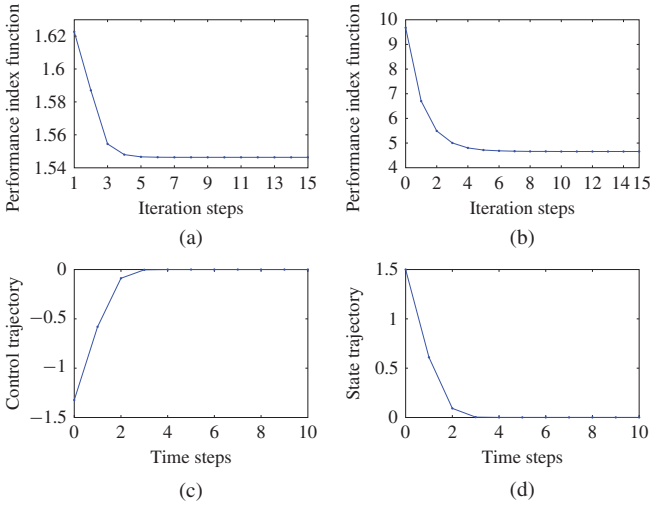

Fig. 3. Simulation results for Case 1 of Example 2. (a) Convergence of performance index function at $x_k = 0.8$. (b) Convergence of performance index function at $x_k = 1.5$. (c) ε-optimal control trajectory. (d) Corresponding state trajectory.

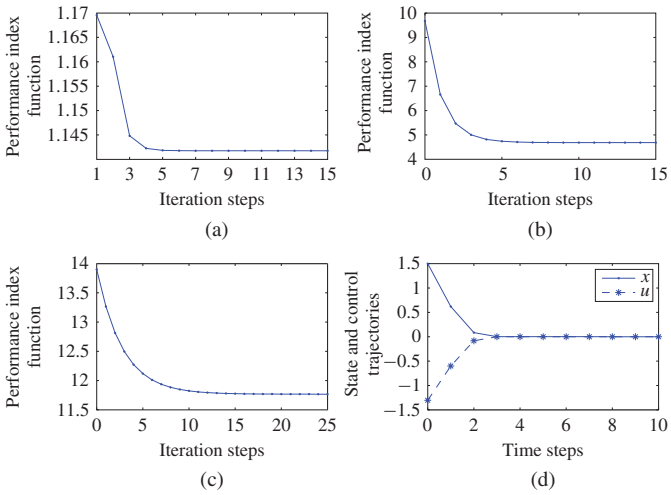

Fig. 4. Simulation results for Case 2 of Example 2. (a) Convergence of performance index function at $x_k = 0.7$. (b) Convergence of performance index function at $x_k = 1.2$. (c) Convergence of performance index function at $x_k = 1.49$. (d) ε-optimal control trajectory and the corresponding state trajectory.

According to (57), the control can be expressed by

$$u_k = -0.1x_k^2 + \sin^{-1}(x_{k+1} - x_k) + 2\lambda\pi \tag{58}$$

where $\lambda = 0, \pm 1, \pm 2, \ldots$.

To show the effectiveness of our algorithm, we choose two initial finite-horizon admissible control sequences.

Case 1: The control sequence is $\underline{\hat{u}}_0^1 = (-0.225 - \sin^{-1}(0.7), -0.064 - \sin^{-1}(0.8))$ and the corresponding state sequence is $\underline{\hat{x}}_0^2 = (1.5, 0.8, 0)$.

For the initial finite-horizon admissible control sequences in this case, run the iterative ADP algorithm at the states 0.8 and 1.5, respectively. For each iterative step, the critic network and the action network are trained for 1000 iteration steps using the learning rate of $\alpha = 0.05$ so that the neural network training accuracy of 10^{-8} is reached. After the algorithm runs for 15 iterative steps, we obtain the performance index function trajectories shown in Fig. 3(a) and (b), respectively. The ε-optimal control and state trajectories are shown in Fig. 3(c) and (d), respectively, for 10 time steps. We obtain $K_\varepsilon(0.8) = 5$ and $K_\varepsilon(1.5) = 8$.

Case 2: The control sequence is $\underline{\hat{u}}_0^3 = (-0.225 - \sin^{-1}(0.01), 2\pi - 2.2201 - \sin^{-1}(0.29), -0.144 - \sin^{-1}(0.5), -\sin^{-1}(0.7))$ and the corresponding state sequence is $\underline{\hat{x}}_0^4 = (1.5, 1.49, 1.2, 0.7, 0)$.

For the initial finite-horizon admissible control sequence in this case, run the iterative ADP algorithm at the states 0.7, 1.2, and 1.49, respectively. For each iterative step, the critic network and the action network are also trained for 1000 iteration steps using the learning rate of $\alpha = 0.05$ so that the neural network training accuracy of 10^{-8} is reached. Then, we obtain the performance index function trajectories shown in Fig. 4(a)–(c), respectively. We have $K_\varepsilon(0.7) = 4$, $K_\varepsilon(1.2) = 6$, and $K_\varepsilon(1.49) = 8$.

After 25 steps of iteration, the performance index function $V_i(x_k)$ is convergent sufficiently at $x_k = 1.49$, with $V_8^3(1.49)$ as the performance index function. For the state $x_k = 1.5$, we have $|V_8^3(1.5) - V_9^3(1.5)| = 0.52424 \times 10^{-7} < \varepsilon$. Therefore, the optimal performance index function at $x_k = 1.5$ is $V_8^3(1.5)$, and thus we have $x_k = 1.5 \in \mathcal{T}_8^{(\varepsilon)}$ and $K_\varepsilon(1.5) = 8$. The whole computation process takes about 20 s and then satisfactory results are obtained.

Then we apply the optimal control law to the system for $T_f = 10$ time steps. The ε-optimal control and state trajectories are shown in Fig. 4(d).

We can see that the ε-optimal control trajectory in Fig. 4(d) is the same as the one in Fig. 3(c). The corresponding state trajectory in Fig. 4(d) is the same as the one in Fig. 3(d). Therefore, the optimal control law is not dependent on the initial control law. The initial control sequence $\underline{\hat{u}}_0^{K-1}$ can arbitrarily be chosen as long as it is finite-horizon admissible. ∎

Remark 5.1: If the number of control steps of the initial admissible control sequence is larger than the number of control steps of the optimal control sequence, then we will have some of the states in the initial sequence to possess the same number of optimal control steps. For example, in Case 2 of Example 2, we see that the two states $x = 1.49$ and $x = 1.5$ possess the same number of optimal control steps, i.e., $K_\varepsilon(1.49) = K_\varepsilon(1.5) = 8$. Thus, we say that the control $u = -0.225 - \sin^{-1}(0.01)$ that makes $x = 1.5$ run to $x = 1.49$ is an unnecessary control step. After the unnecessary control steps are identified and removed, the number of control steps will reduce to the optimal number of control steps, and thus the initial admissible control sequence does not affect the final optimal control results. ∎

VI. Conclusion

In this paper, we developed an effective iterative ADP algorithm for finite-horizon ε-optimal control of discrete-time nonlinear systems. Convergence of the performance index function for the iterative ADP algorithm was proved, and the ε-optimal number of control steps could also be obtained. Neural networks were used to implement the iterative ADP algorithm. Finally, two simulation examples were given to illustrate the performance of the proposed algorithm.

REFERENCES

[1] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*. New York: Wiley, 1975.

[2] T. Cimen and S. P. Banks, "Nonlinear optimal tracking control with application to super-tankers for autopilot design," *Automatica*, vol. 40, no. 11, pp. 1845–1863, Nov. 2004.

[3] N. Fukushima, M. S. Arslan, and I. Hagiwara, "An optimal control method based on the energy flow equation," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 4, pp. 866–875, Jul. 2009.

[4] H. Ichihara, "Optimal control for polynomial systems using matrix sum of squares relaxations," *IEEE Trans. Autom. Control*, vol. 54, no. 5, pp. 1048–1053, May 2009.

[5] S. Keerthi and E. Gilbert, "Optimal infinite-horizon control and the stabilization of linear discrete-time systems: State-control constraints and nonquadratic cost functions," *IEEE Trans. Autom. Control*, vol. 31, no. 3, pp. 264–266, Mar. 1986.

[6] I. Kioskeridis and C. Mademlis, "A unified approach for four-quadrant optimal controlled switched reluctance machine drives with smooth transition between control operations," *IEEE Trans. Autom. Control*, vol. 24, no. 1, pp. 301–306, Jan. 2009.

[7] J. Mao and C. G. Cassandras, "Optimal control of multi-stage discrete event systems with real-time constraints," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 108–123, Jan. 2009.

[8] I. Necoara, E. C. Kerrigan, B. D. Schutter, and T. Boom, "Finite-horizon min-max control of max-plus-linear systems," *IEEE Trans. Autom. Control*, vol. 52, no. 6, pp. 1088–1093, Jun. 2007.

[9] T. Parisini and R. Zoppoli, "Neural approximations for infinite-horizon optimal control of nonlinear stochastic systems," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1388–1408, Nov. 1998.

[10] G. N. Saridis and F. Y. Wang, "Suboptimal control of nonlinear stochastic systems," *Control-Theory Adv. Technol.*, vol. 10, no. 4, pp. 847–871, Dec. 1994.

[11] C. Seatzu, D. Corona, A. Giua, and A. Bemporad, "Optimal control of continuous-time switched affine systems," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 726–741, May 2006.

[12] K. Uchida and M. Fujita, "Finite horizon H_∞ control problems with terminal penalties," *IEEE Trans. Autom. Control*, vol. 37, no. 11, pp. 1762–1767, Nov. 1992.

[13] E. Yaz, "Infinite horizon quadratic optimal control of a class of nonlinear stochastic systems," *IEEE Trans. Autom. Control*, vol. 34, no. 11, pp. 1176–1180, Nov. 1989.

[14] F. Yang, Z. Wang, G. Feng, and X. Liu, "Robust filtering with randomly varying sensor delay: The finite-horizon case," *IEEE Trans. Circuits Syst. I*, vol. 56, no. 3, pp. 664–672, Mar. 2009.

[15] E. Zattoni, "Structural invariant subspaces of singular Hamiltonian systems and nonrecursive solutions of finite-horizon optimal control problems," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1279–1284, Jun. 2008.

[16] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Boston, MA: Athena Scientific, 2003.

[17] J. Doyle, K. Zhou, K. Glover, and B. Bodenheimer, "Mixed H_2 and H_∞ performance objectives II: Optimal control," *IEEE Trans. Autom. Control*, vol. 39, no. 8, pp. 1575–1587, Aug. 1994.

[18] L. Blackmore, S. Rajamanoharan, and B. C. Williams, "Active estimation for jump Markov linear systems," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2223–2236, Nov. 2008.

[19] O. L. V. Costa and E. F. Tuesta, "Finite horizon quadratic optimal control and a separation principle for Markovian jump linear systems," *IEEE Trans. Autom. Control*, vol. 48, no. 10, pp. 1836–1842, Oct. 2003.

[20] P. J. Goulart, E. C. Kerrigan, and T. Alamo, "Control of constrained discrete-time systems with bounded ℓ_2 gain," *IEEE Trans. Autom. Control*, vol. 54, no. 5, pp. 1105–1111, May 2009.

[21] J. H. Park, H. W. Yoo, S. Han, and W. H. Kwon, "Receding horizon controls for input-delayed systems," *IEEE Trans. Autom. Control*, vol. 53, no. 7, pp. 1746–1752, Aug. 2008.

[22] A. Zadorojniy and A. Shwartz, "Robustness of policies in constrained Markov decision processes," *IEEE Trans. Autom. Control*, vol. 51, no. 4, pp. 635–638, Apr. 2006.

[23] H. Zhang, L. Xie, and G. Duan, "H_∞ control of discrete-time systems with multiple input delays," *IEEE Trans. Autom. Control*, vol. 52, no. 2, pp. 271–283, Feb. 2007.

[24] R. E. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.

[25] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control*, W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge, MA: MIT Press, 1991, pp. 67–95.

[26] 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: Reinhold, 1992, ch. 13.

[27] A. Al-Tamimi, M. Abu-Khalaf, and F. L. Lewis, "Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 37, no. 1, pp. 240–247, Feb. 2007.

[28] S. N. Balakrishnan and V. Biega, "Adaptive-critic-based neural networks for aircraft optimal control," *J. Guidance, Control, Dynamics*, vol. 19, no. 4, pp. 893–898, Jul.-Aug. 1996.

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

[30] 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, Jun. 2009.

[31] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev.*, vol. 32, no. 2, pp. 140–153, May 2002.

[32] 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.

[33] 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., Part B: Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[34] S. Ferrari, J. E. Steck, and R. Chandramohan, "Adaptive feedback control by constrained approximate dynamic programming," *, IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 982–987, Aug. 2008.

[35] J. Seiffertt, S. Sanyal, and D. C. Wunsch, "Hamilton-Jacobi-Bellman equations and approximate dynamic programming on time scales," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 918–923, Aug. 2008.

[36] R. Enns and J. Si, "Helicopter trimming and tracking control using direct neural dynamic programming," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 929–939, Jul. 2003.

[37] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

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

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

[40] Z. Chen and S. Jagannathan, "Generalized Hamilton-Jacobi-Bellman formulation-based neural network control of affine nonlinear discrete-time systems," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 90–106, Jan. 2008.

[41] T. Hanselmann, L. Noakes, and A. Zaknich, "Continuous-time adaptive critics," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 631–647, May 2007.

[42] G. G. Lendaris, "A retrospective on adaptive dynamic programming for control," in *Proc. Int. Joint Conf. Neural Netw.*, Atlanta, GA, Jun. 2009, pp. 14–19.

[43] B. Li and J. Si, "Robust dynamic programming for discounted infinite-horizon Markov decision processes with uncertain stationary transition matrices," in *Proc. IEEE Symp. Approx. Dyn. Program. Reinforcement Learn.*, Honolulu, HI, Apr. 2007, pp. 96–102.

[44] D. Liu, X. Xiong, and Y. Zhang, "Action-dependent adaptive critic designs," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2. Washington D.C., Jul. 2001, pp. 990–995.

[45] D. Liu and H. Zhang, "A neural dynamic programming approach for learning control of failure avoidance problems," *Int. J. Intell. Control Syst.*, vol. 10, no. 1, pp. 21–32, Mar. 2005.

[46] D. Liu, Y. Zhang, and H. Zhang, "A self-learning call admission control scheme for CDMA cellular networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 5, pp. 1219–1228, Sep. 2005.

[47] C. Lu, J. Si, and X. Xie, "Direct heuristic dynamic programming for damping oscillations in a large power system," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 1008–1013, Aug. 2008.

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

[49] P. Shih, B. C. Kaul, S. Jagannathan, and J. A. Drallmeier, "Reinforcement-learning-based dual-control methodology for complex nonlinear discrete-time systems with application to spark engine EGR operation," *IEEE Trans. Neural Netw.*, vol. 19, no. 8, pp. 1369–1388, Aug. 2008.

[50] 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.
[51] A. H. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Trans. Neural Netw.*, vol. 19, no. 2, pp. 230–244, Feb. 2008.
[52] F. Y. Wang and G. N. Saridis, "Suboptimal control for nonlinear stochastic systems," in *Proc. 31st IEEE Conf. Decis. Control*, Tucson, AZ, Dec. 1992, pp. 1856–1861.
[53] Q. L. Wei, H. G. Zhang, J. Dai, "Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1839–1848, Mar. 2009.
[54] P. J. Werbos, "Using ADP to understand and replicate brain intelligence: The next level design," in *Proc. IEEE Symp. Approx. Dyn. Program. Reinforcement Learn.*, Honolulu, HI, Apr. 2007, pp. 209–216.
[55] 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.
[56] H. G. Zhang, Y. H. Luo, and D. Liu, "Neural network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraint," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sep. 2009.
[57] H. G. Zhang, Q. L. Wei, and Y. H. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 937–942, Aug. 2008.
[58] C. Watkins, "Learning from delayed rewards," Ph.D. thesis, Dept. Comput. Sci., Cambridge Univ., Cambridge, U.K., 1989.
[59] F. Y. Wang and G. N. Saridis, "On successive approximation of optimal control of stochastic dynamic systems," in *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, M. Dror, P. Lécuyer, and F. Szidarovszky, Eds. Boston, MA: Kluwer, 2002, pp. 333–386.
[60] D. Han and S. N. Balakrishnan, "State-constrained agile missile control with adaptive-critic-based neural networks," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 4, pp. 481–489, Jul. 2002.
[61] E. S. Plumer, "Optimal control of terminal processes using neural networks," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 408–418, Mar. 1996.
[62] N. Jin, D. Liu, T. Huang, and Z. Pang, "Discrete-time adaptive dynamic programming using wavelet basis function neural networks," in *Proc. IEEE Symp. Approx. Dyn. Program. Reinforcement Learn.*, Honolulu, HI, Apr. 2007, pp. 135–142.

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, in 1990.

He joined the University of Arizona, Tuscon, in 1990, and became a Professor and Director of the Robotics and Automation Laboratory and the Program for Advanced Research in Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center at the Chinese Academy of Sciences (CAS), Beijing, China, with the support of the Outstanding Overseas Chinese Talents Program. Since 2002, he has been the Director of the Key Laboratory of Complex Systems and Intelligence Science at CAS. Currently, he is a Vice-President of the Institute of Automation, CAS. His current research interests include social computing, web science, complex systems, and intelligent control.

Dr. Wang is a member of Sigma Xi and an elected Fellow of the International Council on Systems Engineering, the International Federation of Automatic Control, the American Society of Mechanical Engineers (ASME), and the American Association for the Advancement of Science. He was the 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. Currently, he is the Editor-in-Chief of IEEE INTELLIGENT SYSTEMS and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He has served as Chair of more than 20 IEEE, the Association for Computing Machinery, the Institute for Operations Research and the Management Sciences, and ASME conferences. He was the President of the IEEE Intelligent Transportation Systems Society from 2005 to 2007, the Chinese Association for Science and Technology, Pittsburg, PA, in 2005, and the American Zhu Kezhen Education Foundation from 2007 to 2008. Currently, he is the Vice-President of the ACM China Council and Vice-President/Secretary-General of the Chinese Association of Automation. In 2007, he received the National Prize in Natural Sciences of China and was awarded the Outstanding Scientist Award by ACM for his work in intelligent control and social computing.

Ning Jin (S'06) received the Ph.D. degree in electrical and computer engineering from the University of Illinois, Chicago, in 2005.

He was an Associate Professor in the Department of Mathematics at Nanjing Normal University, Nanjing, China. From 2002 to 2005, he was a Visiting Scholar in the Department of Mathematics, Statistics, and Computer Science, University of Illinois. His current research interests include optimal control and dynamic programming, artificial intelligence, pattern recognition, neural networks, and wavelet analysis.

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, in 1994.

He was a Staff Fellow with the General Motors Research and Development Center, Warren, MI, from 1993 to 1995. He was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, from 1995 to 1999. He joined the University of Illinois, Chicago, in 1999, and became a Full Professor of electrical and computer engineering and of computer science in 2006. He was selected for the "100 Talents Program" by the Chinese Academy of Sciences in 2008.

Dr. Liu has been an Associate Editor of several IEEE publications. Currently, he is the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS and an Associate Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY. He received the Michael J. Birck Fellowship from the University of Notre Dame in 1990, the Harvey N. Davis Distinguished Teaching Award from the Stevens Institute of Technology in 1997, the Faculty Early Career Development Award from the National Science Foundation in 1999, the University Scholar Award from the University of Illinois in 2006, and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China in 2008.

Qinglai Wei received the B.S. degree in automation, the M.S. degree in control theory and control engineering, and the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2002, 2005, and 2008, respectively.

He is currently a Post-Doctoral Fellow with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include neural-networks-based control, nonlinear control, adaptive dynamic programming, and their industrial applications.