J Control Theory Appl 2011 9 (3) 381–390
DOI 10.1007/s11768-011-0181-5

Finite horizon optimal control of discrete-time nonlinear systems with unfixed initial state using adaptive dynamic programming

Qinglai WEI, Derong LIU

Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

Abstract: In this paper, we aim to solve the finite horizon optimal control problem for a class of discrete-time nonlinear systems with unfixed initial state using adaptive dynamic programming (ADP) approach. A new ϵ-optimal control algorithm based on the iterative ADP approach is proposed which makes the performance index function converge iteratively to the greatest lower bound of all performance indices within an error according to ϵ within finite time. The optimal number of control steps can also be obtained by the proposed ϵ-optimal control algorithm for the situation where the initial state of the system is unfixed. Neural networks are used to approximate the performance index function and compute the optimal control policy, respectively, for facilitating the implementation of the ϵ-optimal control algorithm. Finally, a simulation example is given to show the results of the proposed method.

Keywords: Adaptive dynamic programming; Unfixed initial state; Optimal control; Finite time; Neural networks

1 Introduction

The optimal control problem of dynamical systems has been the focus of many papers during the last several decades. In many theoretical discussions, controllers are generally designed to make the controlled systems stabilized or tracked within infinite time horizon [1–9]. That is, the system cannot really be stabilized or tracked until the time reaches infinity. However, in real world, most of the control systems should be stabilized to zero or tracked to a desired trajectory within finite time. Due to the lack of methodology and the fact that the number of control steps is difficult to determine, the controller design of finite horizon problems still presents a challenge to control engineers. On the other hand, the optimal control schemes should be effective for different initial states, i.e., for a set of initial states. However, many optimal control scheme is proposed only for the situations of fixed initial state [10–20]. To overcome these difficulties, a new method must be proposed for finite horizon optimal control problems with unfixed initial state. This motivates our research.

Dynamic programming is a very useful tool in solving optimization and optimal control problems [21–23]. The optimality principle can be expressed by a Hamilton-Jacobi-Bellman (HJB) equation. However, it is often computationally untenable to run real dynamic programming due to the 'curse of dimensionality' [21]. Adaptive/approximate dynamic programming (ADP) algorithms have gained much attention from researchers [19, 20, 24–34]. An ADP algorithm was proposed in [35] as a powerful methodology to solve optimal control problems forward in time. In [36] and [37], ADP approaches were classified into several main schemes: heuristic dynamic programming (HDP), dual heuristic programming (DHP), action dependent heuristic dynamic programming (ADHDP), also known as Q-learning, and action dependent dual heuristic programming (ADDHP), globalized-DHP (GDHP) and ADGDHP.

Though ADP algorithms have made great progress in the optimal control field, to the best of our knowledge, only [38] discussed finite horizon optimal control problem with fixed initial state. While if the initial state is unfixed, such as the initial state belongs to an initial state set, then [38] is invalid. There are at least two reasons. First, it cannot be decided which initial point should be used as the computation initial point. If the iterative ADP algorithm in [38] is implemented for each state point, the computation burden is extremely large to obtain the optimal control, because the elements in the initial state set may be infinite. Even if the iterative ADP algorithm is implemented for each state point, we still do not know which control law is optimal one according to [38].

Therefore, how to solve the optimal control problem for finite horizon systems with unfixed initial state based on ADP algorithms is still an open problem. In this paper, for the first time, we will show how to find an approximate optimal control that makes the performance index function converge to the greatest lower bound of all performance indices within an error according to ϵ (called ϵ-error bound for brief). It is also shown that the corresponding approximate optimal control (called ϵ-optimal control) can make the performance index function converge to the ϵ-error bound within finite steps. First, the HJB equation for finite horizon discrete-time systems is derived. In order to solve this HJB equation, a new iterative adaptive dynamic programming algorithm is developed with convergence proof. Sec-

Received 1 August 2010; revised 9 April 2011.
This work was partly supported by the National Natural Science Foundation of China (No.60904037, 60921061, 61034002), and the Beijing Natural Science Foundation (No. 4102061).

ond, it will show that the iterative ADP algorithm can obtain the approximate optimal control that makes the performance index function converge to an ϵ-error bound from its optimal under the situation where the initial state is unfixed. Furthermore, in order to facilitate the implementation of the iterative ADP algorithm, we show how to introduce neural networks to obtain the iterative performance index function. This in turn results in an ϵ-optimal state feedback controller suitable for finite horizon problems.

2 Problem statement

In this paper, we consider the following discrete-time nonlinear systems:

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

where $x_k \in \mathbb{R}^n$ is the state, and $u_k \in \mathbb{R}^m$ is the control vector. Let $x_0 \in \Omega_0$ be the initial state, where $\Omega_0 \subset \mathbb{R}^n$ is the initial state set. We assume that for any initial state x_k, there exists a control u_k that makes $F(x_k, u_k) = 0$. Let the system function $F(x_k, u_k)$ be continuous for $\forall\, x_k, u_k$, and $F(0, 0) = 0$. Hence, $x_k = 0$ is an equilibrium state of system (1) under the control $u_k = 0$. 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

$$J(x_0, \underline{u}_0^{N-1}) = \sum_{k=0}^{N-1} U(x_k, u_k), \tag{2}$$

where $U(x_k, u_k) \geqslant 0$, for $\forall\, x_k, u_k$, is the utility function.

Let $\underline{u}_0^{N-1} = (u_0, u_1, \ldots, u_{N-1})$ be a finite sequence of controls. We call the number of elements in the control sequence $\underline{u}_0^{N-1}$ the length of $\underline{u}_0^{N-1}$. Then, $|\underline{u}_0^{N-1}| = N$. We denote the final state of the trajectory as $x^{(\mathrm{f})}(x_0, \underline{u}_0^{N-1})$, i.e., $x^{(\mathrm{f})}(x_0, \underline{u}_0^{N-1}) = x_N$. Then, for $\forall k \geqslant 0$, the finite control sequence can be written as

$$\underline{u}_k^{k+i-1} = (u_k, u_{k+1}, \ldots, u_{k+i-1}),$$

where $i \geqslant 1$. Then, the final state can be written as $x^{(\mathrm{f})}(x_k, \underline{u}_k^{k+i-1})$, where $x^{(\mathrm{f})}(x_k, \underline{u}_k^{k+i-1}) = x_{k+i}$.

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

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

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 [21], $J^*(x_k)$ satisfies the discrete-time HJB 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

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

3 Properties of the iterative adaptive dynamic programming 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.

3.1 Derivation of the iterative ADP algorithm

In the iterative ADP algorithm, the performance index function and control policy are updated by recurrent iterations, with the iteration index number i increasing from 0 and with the initial performance index function $V_0(x_k) = 0$.

The performance index function $V_1(x_k)$ is computed as

$$V_1(x_k) = \min_{u_k} U(x_k, u_k) = U(x_k, v_0(x_k)), \quad \text{s.t. } F(x_k, u_k) = 0, \tag{5}$$

where

$$v_0(x_k) = \arg\min_{u_k} U(x_k, u_k) \ \text{ s.t. } \ F(x_k, u_k) = 0. \tag{6}$$

For $i = 1, 2, 3, \ldots$, the iterative ADP algorithm will be implemented as follows:

$$\begin{aligned} V_{i+1}(x_k) &= \min_{u_k} \left\{ U(x_k, u_k) + V_i(F(x_k, u_k)) \right\} \\ &= U(x_k, v_i(x_k)) + V_i(F(x_k, v_i(x_k))), \end{aligned} \tag{7}$$

where

$$\begin{aligned} v_i(x_k) &= \arg\min_{u_k} \left\{ U(x_k, u_k) + V_i(x_{k+1}) \right\} \\ &= \arg\min_{u_k} \left\{ U(x_k, u_k) + V_i(F(x_k, u_k)) \right\}. \end{aligned} \tag{8}$$

Equations (5)–(8) are called the iterative ADP algorithm.

3.2 Properties

In the above, we can see that the performance index function $J^*(x_k)$ solved by HJB equation (4) is replaced with a sequence of performance index functions $V_i(x_k)$, and the optimal control law $u^*(x_k)$ is replaced with a sequence of control law $v_i(x_k)$, where $i \geqslant 1$ 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 1 Let x_k be an arbitrary state vector. Suppose that $\mathfrak{A}_{x_k}^{(1)} \neq \varnothing$. Then, the performance index function $V_i(x_k)$ obtained by (5)–(8) is a monotonically nonincreasing sequence for $\forall i \geqslant 1$, i.e., $V_{i+1}(x_k) \leqslant V_i(x_k)$ for $\forall i \geqslant 1$.

Proof The details can be seen in [38].

From Theorem 1, we know that the performance index function $V_i(x_k) \geqslant 0$ is a monotonically nonincreasing sequence with lower bound for iteration index $i = 1, 2, \ldots$. Then, there exists a limit of iterative performance index function $V_i(x_k)$. Define the performance index function $V_\infty(x_k)$ as the limit of the iterative function $V_i(x_k)$, i.e.,

$$V_\infty(x_k) = \lim_{i \to \infty} V_i(x_k). \tag{9}$$

Now, we can derive the following theorem.

Theorem 2 Let the performance index function $V_i(x_k)$ be defined by (7). If the system state x_k is controllable, then the performance index function $V_i(x_k)$ converges to the optimal performance index function $J^*(x_k)$ as $i \to \infty$, i.e.,

$$V_i(x_k) \to J^*(x_k). \tag{10}$$

Proof According to (3), we have

$$\begin{aligned} J^*(x_k) &\leqslant \min_{\underline{u}_k^{k+i-1}} \{J(x_k, \underline{u}_k^{k+i-1}) \colon \underline{u}_k^{k+i-1} \in \mathfrak{A}_{x_k}^{(i)}\} \\ &= V_i(x_k). \end{aligned}$$

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

$$J^*(x_k) \leqslant V_\infty(x_k). \tag{11}$$

Next, we show that

$$V_\infty(x_k) \leqslant J^*(x_k). \tag{12}$$

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

$$J(x_k, \underline{\eta}_k) \leqslant J^*(x_k) + \omega. \tag{13}$$

Suppose that $|\underline{\eta}_k| = p$. Then, $\underline{\eta}_k \in \mathfrak{A}_{x_k}^{(p)}$. Therefore, by Theorem 1,

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

Since ω is chosen arbitrarily, we know that (12) is true. Therefore, from (11) and (12), we prove the theorem.

We can now present the following corollary.

Corollary 1 Let the performance index function $V_i(x_k)$ be defined by (7). If the system state x_k is controllable, then the iterative control law $u_i(x_k)$ converges to the optimal control law $u^*(x_k)$, i.e., $\lim\limits_{i\to\infty} u_i(x_k) = u^*(x_k)$.

4 The 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 $J^*(x_k)$ as $i \to \infty$. This means that to obtain the optimal performance index function $J^*(x_k)$, we should run the iterative ADP algorithm (5)–(8) for $i \to \infty$. However unfortunately, it is not achievable to run for infinite number of times. To overcome this difficulty, a new ϵ-optimal control algorithm is established in this section.

4.1 The derivation of the ϵ-optimal control algorithm

In this subsection, we will introduce our method for iterative ADP with the consideration of the length of control sequences. For different x_k, we will use different i for the length of optimal control sequence. For a given error bound according to $\epsilon > 0$, the number i will be chosen so that the error between $J^*(x_k)$ and $V_i(x_k)$ is bounded with a bound according to ϵ.

Theorem 3 Let $\epsilon > 0$ be any small number and x_k 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. Then, there exists a finite i that makes

$$|V_i(x_k) - J^*(x_k)| \leqslant \epsilon \tag{14}$$

hold.

Proof The proof follows the definition in (9) and Theorem 2.

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

$$\mathcal{T}_i = \{x_k \in \mathbb{R}^n \mid \exists\, u_k \in \mathbb{R}^m \text{ s.t. } F(x_k, u_k) \in \mathcal{T}_{i-1}\}. \tag{15}$$

According to Theorem 3, we can make the following definition.

Definition 1 Let $x_k \in \mathcal{T}_\infty$ be a controllable state vector. Let $\epsilon > 0$ be a small positive number. The approximate length of optimal control with respect to ϵ is defined as

$$K_\epsilon(x_k) = \min\{i \colon |V_i(x_k) - J^*(x_k)| \leqslant \epsilon\}. \tag{16}$$

Given a small positive number ϵ, for any state vector x_k, the number $K_\epsilon(x_k)$ gives a suitable length of control sequence for optimal control starting from x_k. For $x_k \in \mathcal{T}_\infty$, since $\lim\limits_{i\to\infty} V_i(x_k) = J^*(x_k)$, we can always find i such that

$$|V_i(x_k) - J^*(x_k)| \leqslant \epsilon. \tag{17}$$

Therefore, $\{i \colon |V_i(x_k) - J^*(x_k)| \leqslant \epsilon\} \neq \varnothing$, and $K_\epsilon(x_k)$ is well defined.

4.2 Properties of the ϵ-optimal control algorithm

We can see that an error ϵ between $J^*(x_k)$ and $V_i(x_k)$ is introduced into the iterative ADP algorithm, which makes the performance index function $V_i(x_k)$ converge within finite iteration step i. In this section, we will show that the corresponding control is also an effective control that makes the performance index function reach the optimum within error bound ϵ.

Theorem 4 Let $x_k \in \mathcal{T}_\infty$ be arbitrary controllable state. If for arbitrary small positive number ϵ, the iterative performance index function satisfies $|V_i(x_k) - J^*(x_k)| \leqslant \epsilon$ under the control $\mu_\epsilon^*(x_k)$, where $\mu_\epsilon^*(x_k)$ satisfies

$$\mu_\epsilon^*(x_k) = \arg\min_{u_k} \left\{U(x_k, u_k) + V_i(F(x_k, u_k))\right\}. \tag{18}$$

Then, the ϵ-optimal control sequence

$$\underline{\mu}_\epsilon^*(x_k) = (\mu_\epsilon^*(x_k), \mu_\epsilon^*(x_{k+1}), \ldots, \mu_\epsilon^*(x_{k+i}), \ldots) \tag{19}$$

is an admissible control sequence for system (1).

Proof According to (7) and (18), we have

$$V_i(x_{k+1}) - V_{i+1}(x_k) = -U(x_k, \mu_\epsilon^*(x_k)). \tag{20}$$

As $|V_i(x_k) - J^*(x_k)| \leqslant \epsilon$, according to Theorem 1, we have $J^*(x_k) \leqslant V_{i+1}(x_k) \leqslant V_i(x_k) \leqslant J^*(x_k) + \epsilon$. Then, there exists a $0 \leqslant \delta \leqslant \epsilon$ that satisfies

$$V_i(x_{k+1}) - V_i(x_k) = -U(x_k, \mu_\epsilon^*(x_k)) + \delta. \tag{21}$$

As ϵ is small, let $\epsilon \to 0$, then we have $\delta \to 0$. Hence, we can get $-U(x_k, \mu_\epsilon^*(x_k)) + \delta \leqslant 0$. Then, we can obtain

$$V_i(x_{k+1}) - V_i(x_k) \leqslant 0, \tag{22}$$

which implies that $\mu_\epsilon^*(x_k)$ is a stable controller. On the other hand, it is obvious that $V_i(x_k) \leqslant J^*(x_k) + \epsilon$ is finite. According to the definition of admissible control sequence [19, 32], we have that $\underline{\mu}_\epsilon^*(x_k)$ is an admissible control sequence. The proof is completed.

Remark 1 Theorem 4 shows that in the iterative ADP algorithm (5)–(8), the positive number ϵ should be small enough to obtain the effective approximate optimal control $\mu_\epsilon^*(x_k)$. However when ϵ is smaller, the iterative ADP algorithm has to run for longer time to obtain the higher precision which leads to heavier computational burden.

Then, we note the following corollary.

Corollary 2 If for $\forall i > 0$ and arbitrary system state $x_k \in \mathcal{T}_\infty$, the control $\mu_\epsilon^*(x_k)$ makes $|V_i(x_k) - J^*(x_k)| < \epsilon$ hold, then $|V_{i+1}(x_k) - J^*(x_k)| < \epsilon$ holds.

Definition 2 Let $x_k \in \mathcal{T}_\infty$ be a controllable state vector and ϵ be a positive number. For $i = 1, 2, \ldots$, define the set

$$\mathcal{T}_i^{(\epsilon)} = \{x_k \in \mathcal{T}_\infty \colon K_\epsilon(x_k) \leqslant i\}. \tag{23}$$

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

Theorem 5 Let $\epsilon > 0$ and $i = 1, 2, \ldots$. Then,

i) $x_k \in \mathcal{T}_i^{(\epsilon)}$ if and only if $V_i(x_k) \leqslant J^*(x_k) + \epsilon$.

ii) $\mathcal{T}_i^{(\epsilon)} \subseteq \mathcal{T}_i$.

iii) $\mathcal{T}_i^{(\epsilon)} \subseteq \mathcal{T}_{i+1}^{(\epsilon)}$.

iv) $\bigcup\limits_i \mathcal{T}_i^{(\epsilon)} = \mathcal{T}_\infty$.

v) If $\epsilon > \delta > 0$, then $\mathcal{T}_i^{(\epsilon)} \supseteq \mathcal{T}_i^{(\delta)}$.

Proof i) Let $x_k \in \mathcal{T}_i^{(\epsilon)}$. By Definition 2, $K_\epsilon(x_k) \leqslant i$. Let $j = K_\epsilon(x_k)$. Then, $j \leqslant i$ and by Definition 1, $|V_j(x_k) - J^*(x_k)| \leqslant \epsilon$. Therefore, $V_j(x_k) \leqslant J^*(x_k) + \epsilon$. By Theorem 1, $V_i(x_k) \leqslant V_j(x_k) \leqslant J^*(x_k) + \epsilon$. On the other hand, if $V_i(x_k) \leqslant J^*(x_k) + \epsilon$, then $|V_i(x_k) - J^*(x_k)| \leqslant \epsilon$. Therefore, $K_\epsilon(x_k) = \min\{j \colon |V_j(x_k) - J^*(x_k)| \leqslant \epsilon\} \leqslant i$. Therefore, $x_k \in \mathcal{T}_i^{(\epsilon)}$, and vice versa.

ii) When $x_k \in \mathcal{T}_i^{(\epsilon)}$, $K_\epsilon(x_k) \leqslant i$. Therefore, $V_i(x_k)$ has definition at x_k. Thus, $x_k \in \mathcal{T}_i$. Hence, $\mathcal{T}_i^{(\epsilon)} \subseteq \mathcal{T}_i$.

iii) When $x_k \in \mathcal{T}_i^{(\epsilon)}$, $K_\epsilon(x_k) \leqslant i < i + 1$. Therefore, $x_k \in \mathcal{T}_{i+1}^{(\epsilon)}$. Thus, $\mathcal{T}_i^{(\epsilon)} \subseteq \mathcal{T}_{i+1}^{(\epsilon)}$.

iv) Obviously, $\bigcup\limits_i \mathcal{T}_i^{(\epsilon)} \subseteq \mathcal{T}_\infty$ since $\mathcal{T}_i^{(\epsilon)}$ are subsets of $\mathcal{T}_\infty$. For any $x_k \in \mathcal{T}_\infty$, let $p = K_\epsilon(x_k)$. Then, $x_k \in \mathcal{T}_p^{(\epsilon)}$. So $x_k \in \bigcup\limits_i \mathcal{T}_i^{(\epsilon)}$. Hence, $\mathcal{T}_\infty \subseteq \bigcup\limits_i \mathcal{T}_i^{(\epsilon)} \subseteq \mathcal{T}_\infty$, and thus, $\bigcup\limits_i \mathcal{T}_i^{(\epsilon)} = \mathcal{T}_\infty$.

v) When $x_k \in \mathcal{T}_i^{(\delta)}$, $V_i(x_k) \leqslant J^*(x_k) + \delta$ by part i) of this theorem. Clearly, $V_i(x_k) \leqslant J^*(x_k) + \epsilon$ since $\delta < \epsilon$. This implies that $x_k \in \mathcal{T}_i^{(\epsilon)}$. Therefore, $\mathcal{T}_i^{(\epsilon)} \supseteq \mathcal{T}_i^{(\delta)}$.

We have the following corollary.

Corollary 3 Let $\epsilon > 0$ and $i = 0, 1, 2, \ldots$. If $x_k \in \mathcal{T}_i^{(\epsilon)}$, then for any $x_k' \in \mathcal{T}_i^{(\epsilon)}$, the following inequality

$$|V_i(x_k') - J^*(x_k')| \leqslant \epsilon \tag{24}$$

holds.

Proof The corollary can be easily proved by contradiction. Assume that the conclusion is false. Then, for some $x_k' \in \mathcal{T}_i^{(\epsilon)}$, we have

$$V_i(x_k') > J^*(x_k') + \epsilon. \tag{25}$$

Therefore, we have

$$K_\epsilon(x_k') = \min\{j \colon |V_j(x_k') - J^*(x_k')| \leqslant \epsilon\} > i. \tag{26}$$

Then, according to Definition 2, we have

$$x_k' \notin \mathcal{T}_i^{(\epsilon)}, \tag{27}$$

which is contradicting with the assumption of $x_k' \in \mathcal{T}_i^{(\epsilon)}$, so the conclusion holds.

Corollary 4 For $i = 0, 1, \ldots$, let $\mu_\epsilon^*(x_k)$ be expressed as in (18), which makes the performance index function (14) hold for $x_k \in \mathcal{T}_i^{(\epsilon)}$. Then, for $x_k' \in \mathcal{T}_i^{(\epsilon)}$, $\mu_\epsilon^*(x_k')$ makes

$$|V_i(x_k') - J^*(x_k')| \leqslant \epsilon \tag{28}$$

hold.

Proof The corollary can also be proved by contradiction.

According to Theorem 5 i), $\mathcal{T}_i^{(\epsilon)}$ 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 5 ii). As stated in Theorem 5 iii), when i is large, the set $\mathcal{T}_i^{(\epsilon)}$ is also large. This means that 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 from the origin, we have to choose a long control sequence to approximate the optimal control. Theorem 5 iv) means that for every controllable state $x_k \in \mathcal{T}_\infty$, we can always find a suitable length i of control sequence to approximate the optimal control. The size of the set $\mathcal{T}_i^{(\epsilon)}$ depends on the value of ϵ. Smaller value of ϵ gives smaller set $\mathcal{T}_i^{(\epsilon)}$, see Theorem 5 v).

Theorem 6 For $i = 0, 1, \ldots$, if we let $x_k \in \mathcal{T}_{i+1}^{(\epsilon)}$ and let $\mu_\epsilon^*(x_k)$ be expressed in (18), then $F(x_k, \mu_\epsilon^*(x_k)) \in \mathcal{T}_i^{(\epsilon)}$. In other words, if $K_\epsilon(x_k) = i + 1$, then

$$K_\epsilon(F(x, \mu_\epsilon^*(x_k))) \leqslant i.$$

Proof Since $x_k \in \mathcal{T}_{i+1}^{(\epsilon)}$, by Theorem 5 i), we know that

$$V_{i+1}(x_k) \leqslant J^*(x_k) + \epsilon. \tag{29}$$

According to the expression of $\mu_\epsilon^*(x_k)$ in (18), we have

$$V_{i+1}(x_k) = U(x_k, \mu_\epsilon^*(x_k)) + V_i(F(x_k, \mu_\epsilon^*(x_k))). \tag{30}$$

From (29) and (30), we have

$$\begin{aligned} V_i(F(x_k, \mu_\epsilon^*(x_k))) &= V_{i+1}(x_k) - U(x_k, \mu_\epsilon^*(x_k)) \\ &\leqslant J^*(x_k) + \epsilon - U(x_k, \mu_\epsilon^*(x_k)). \end{aligned} \tag{31}$$

On the other hand, we have

$$J^*(x_k) \leqslant U(x_k, \mu_\epsilon^*(x_k)) + J^*(F(x, \mu_\epsilon^*(x_k))). \tag{32}$$

Putting (32) into (31), we obtain

$$V_i(F(x_k, \mu_\epsilon^*(x_k))) \leqslant J^*(F(x_k, \mu_\epsilon^*(x_k))) + \epsilon. \tag{33}$$

By Theorem 5, we have

$$F(x_k, \mu_\epsilon^*(x_k)) \in \mathcal{T}_i^{(\epsilon)}. \tag{34}$$

Therefore, if $K_\epsilon(x_k) = i + 1$, we know that $x_k \in \mathcal{T}_{i+1}^{(\epsilon)}$ and $F(x, \mu_\epsilon^*(x_k)) \in \mathcal{T}_i^{(\epsilon)}$ according to (34). Therefore, we have

$$K_\epsilon(F(x_k, \mu_\epsilon^*(x_k))) \leqslant i. \tag{35}$$

The theorem is proved.

Corollary 5 For $i = 0, 1, \ldots$, let $\mu_\epsilon^*(x_k)$ be expressed in (18) where the performance index function $|V_{i+1}(x_k) - J^*(x_k)| \leqslant \epsilon$. Then, for any $x_k' \in \mathcal{T}_i^{(\epsilon)}$, the following inequality

$$|V_i(x_k') - J^*(x_k')| \leqslant \epsilon \tag{36}$$

holds.

Proof $|V_{i+1}(x_k) - J^*(x_k)| \leqslant \epsilon$ implies $x_k \in \mathcal{T}_{i+1}^{(\epsilon)}$. According to Theorem 6, we have

$$K_\epsilon(F(x_k, \mu_\epsilon^*(x_k))) \leqslant i, \tag{37}$$

which means that

$$|V_i(F(x_k, \mu_\epsilon^*(x_k))) - J^*(F(x_k, \mu_\epsilon^*(x_k)))| \leqslant \epsilon \tag{38}$$

and

$$F(x_k, \mu_\epsilon^*(x_k)) \in \mathcal{T}_i^{(\epsilon)}. \tag{39}$$

Then, according to Corollary 3, for any $x_k' \in \mathcal{T}_i^{(\epsilon)}$, we have

$$|V_i(x_k') - J^*(x_k')| \leqslant \epsilon \tag{40}$$

hold.

The proof is completed.

Now, we look back at the optimal control problem with respect to performance index function. If the initial state x_0 is fixed, we will show that if we choose x_0 to run the iterative index function, we can obtain the most effective ϵ-optimal control.

Theorem 7 Let x_0 be the fixed initial state, $\mu_\epsilon^*(x_0)$ satisfies (17) and (18) at $k = 0$. If x_k, $k = 0, 1, \ldots, N$, is the state under the control law $\mu_\epsilon^*(x_k)$, then we have $|V_i(x_k) - J^*(x_k)| \leqslant \epsilon$ for any k.

Proof For system (1) with respect to the performance index function (2), we have $x_0 \in \mathcal{T}_N^{(\epsilon)}$ and $K_\epsilon(x_0) = N$. Then, for small ϵ, there exists an ϵ-optimal control sequence:

$$\underline{\mu}_\epsilon^*(x_0) = (\mu_\epsilon^*(x_0), \mu_\epsilon^*(x_1), \ldots, \mu_\epsilon^*(x_{N-1})), \tag{41}$$

which stabilizes system (1) within finite time N and minimizes the performance index function (2). Then, obviously, we have $x_N \in \mathcal{T}_0$, $x_{N-1} \in \mathcal{T}_1^{(\epsilon)}, \ldots, x_0 \in \mathcal{T}_N^{(\epsilon)}$, where $0 = \mathcal{T}_0 \subseteq \mathcal{T}_1^{(\epsilon)} \subseteq \ldots \subseteq \mathcal{T}_N^{(\epsilon)}$. Therefore, according to Theorem 6 and Corollary 5, we have that the ϵ-optimal control law μ_ϵ^* obtained by the initial state x_0 using the iterative ADP algorithm is effective for the states $x_1, x_2, \ldots, x_N$.

The proof is completed.

We can see that if we choose x_0 to run the iterative index function, we can obtain the most effective ϵ-optimal control. If the initial state x_0 is unfixed, then we do not know which one should be used to implement the iterative ADP algorithm. In the next section, we will solve this problem.

4.3 The ϵ-optimal control algorithm for unfixed initial state

For a lot of practical nonlinear systems, the initial state x_0 cannot be fixed. Instead, the initial state belongs to a set, and we define the initial state set as Ω_0, where $\Omega_0 \subseteq \mathbb{R}^n$. Then, we have $x_0 \in \Omega_0$. For this case, if we only choose one state $x_0^{(i)} \in \Omega_0$ to run the iterative ADP algorithm and get corresponding ϵ-optimal control μ_ϵ^*, then the ϵ-optimal control μ_ϵ^* may not be ϵ-optimal for all $x_0 \in \Omega_0$ because there may exist a state $x_0^{(j)} \in \Omega_0$ such that $x_0^{(i)} \in \mathcal{T}_i^{(\epsilon)}$, while $x_0^{(j)} \in \mathcal{T}_j^{(\epsilon)} \backslash \mathcal{T}_i^{(\epsilon)}$, where $j > i$.

If we let

$$I = \max\{i \colon x_0 \in \mathcal{T}_i^{(\epsilon)} \text{ s.t. } x_0 \in \Omega_0\}. \tag{42}$$

then according to Corollary 5, we should find the initial state $x_0 \in \mathcal{T}_I^{(\epsilon)}$ to obtain the most effective ϵ-optimal control. Thus, the next job is to obtain the state $x_0 \in \mathcal{T}_I^{(\epsilon)}$. According to the grid method, we divide the state set Ω_0 into P subsets that are expressed as $\Omega_0^{(1)}$, $\Omega_0^{(2)}$, …, $\Omega_0^{(P)}$, where $\Omega_0 = \bigcup_{j=1}^{P} \Omega_0^{(j)}$ and $P > 0$ is a positive integer number. Then, for $j = 1, 2, \ldots, P$, let

$$X_0 = (x_0^{(1)}, x_0^{(2)}, \ldots, x_0^{(P)}), \tag{43}$$

where $x_0^{(j)}$ satisfies $x_0^{(j)} \in \Omega_0^{(j)}$. Then, according to Theorem 7, for every $x_0^{(j)}, j = 1, 2, \ldots, P$, we run the iterative ADP algorithm (5)–(8) at $x_k = x_0^{(j)}$, and then, we can obtain $x_0^{(j)} \in \mathcal{T}_{i_j}^{(\epsilon)}$. The corresponding performance index function and control can be expressed as $V_{i_j}(x_0^{(j)})$ and $\mu_\epsilon^*(x_0^{(j)})$, respectively. As the initial state set Ω_0 is divided into P subsets, then according to (43), we can redefine I as

$$I = \max\{\, i_j \colon x_0^{(j)} \in \mathcal{T}_{i_j}^{(\epsilon)},\ j = 1, 2, \ldots, P \text{ s.t. } x_0^{(j)} \in \Omega_0^{(j)}, \Omega_0 = \bigcup_{j=1}^{P} \Omega_0^{(j)}\}. \tag{44}$$

Without loss of generality, we can let $x_0^{(\bar{j})} \in \mathcal{T}_I^{(\epsilon)}$, where $1 \leqslant \bar{j} \leqslant P$. Then, the corresponding performance index function $V_I(x_0^{(\bar{j})})$ is the ϵ-optimal performance index function, and $\mu_\epsilon^*(x_0^{(\bar{j})})$ is the ϵ-optimal control, respectively.

For example, let $x_k \in \mathbb{R}$ be the state of system (1) and the initial state set be $\Omega_0 = \{x_0 : a \leqslant x_0 \leqslant b\}$, where $a, b \in \mathbb{R}$. According to the grid method, let $h > 0$ be the grid size, and $P = (b - a)/h$. Then, we have $\Omega_0^{(j)} = \{x_0 : a + (j-1)h \leqslant x_0 < a + jh\}$, where $j = 1, 2, \ldots, P$. Therefore, we can choose $x_0^{(j)} = a + (j-1)h$, where $x_0^{(j)}$ satisfies $x_0^{(j)} \in \Omega_0^{(j)}$. Then, run the iterative ADP algorithm (5)–(8) at $x_k = x_0^{(j)}$ for $j = 1, 2, \ldots, P$, and we can obtain $x_0^{(\bar{j})} \in \mathcal{T}_I^{(\epsilon)}$ where I satisfies (44). The corresponding control law μ_ϵ^* is the effective ϵ-optimal control law.

While if the initial set Ω_0 is a convex set, we can obtain more simple results.

Theorem 8 Let $\Omega_0 \subseteq \mathbb{R}^n$ be the initial state set and the initial state $x_0 \in \Omega_0$. If Ω_0 is a convex set on $\mathbb{R}^n$, then $x_0^{(I)}$ is a boundary point of Ω_0 where I is defined in (44).

Proof The theorem can be proved by contradiction. Assume that $x_0^{(I)}$ is a interior point of Ω_0. Without loss of generality, let the point be $x_\mathrm{a} = x_0^{(I)}$. Make a beeline between the origin and x_a. Let the point of intersection between the beeline and the boundary of the set Ω_0 be x_b. Let the point of intersection between the extended line and the boundary of the set Ω_0 be x_c. The situation of x_a, x_b and x_c is shown in Fig. 1.

As $x_0^{(I)}$ is an interior point of convex set Ω_0, according to the property of convex set, for $\forall x_0^{(j)} \in \Omega_0$, $j = 0, 1, \ldots$, there exists a positive real number $0 \leqslant \lambda \leqslant 1$ that makes

$$x_0^{(j_\mathrm{a})} = \lambda\, x_0^{(j_\mathrm{b})} + (1 - \lambda) x_0^{(j_\mathrm{c})} \tag{45}$$

hold, where j_a, j_b and j_c are nonnegative integer numbers.

If let $x_\mathrm{a} = x_0^{(I)} = x_0^{(j_\mathrm{a})}$, $x_\mathrm{b} = x_0^{(j_\mathrm{b})}$, and $x_\mathrm{c} = x_0^{(j_\mathrm{c})}$, then we have

$$x_\mathrm{a} = \lambda\, x_\mathrm{b} + (1 - \lambda) x_\mathrm{c}. \tag{46}$$

If we assume that $x_\mathrm{a} \in \mathcal{T}_\mathrm{a}^{(\epsilon)}$, $x_\mathrm{b} \in \mathcal{T}_\mathrm{b}^{(\epsilon)}$, and $x_\mathrm{c} \in \mathcal{T}_\mathrm{c}^{(\epsilon)}$, then we have

$$\mathcal{T}_\mathrm{c}^{(\epsilon)} \subseteq \mathcal{T}_\mathrm{a}^{(\epsilon)} \tag{47}$$

because $x_{\rm a} = x_0^{(I)}$, where I is expressed in (44). Then, we can obtain

$$I = K_\epsilon(F(x_{\rm a}, \mu_\epsilon^*(x_{\rm a}))) \geqslant K_\epsilon(F(x_{\rm c}, \mu_\epsilon^*(x_{\rm c}))) = c. \quad (48)$$

On the other hand, as $x_{\rm c}$ is the point of intersection between the extended beeline and the boundary of the set Ω_0, obviously, the point $x_{\rm c}$ is farther from the origin that can also be seen in Fig. 1. Therefore, we have

$$K_\epsilon(F(x_{\rm a}, \mu_\epsilon^*(x_{\rm a}))) \leqslant K_\epsilon(F(x_{\rm c}, \mu_\epsilon^*(x_{\rm c}))) = c, \quad (49)$$

which is a contradiction to (48). Therefore, $x_0^{(I)}$ cannot be expressed as (45). Then the assumption is false, and therefore, $x_0^{(I)}$ must be the boundary point of the set Ω_0.

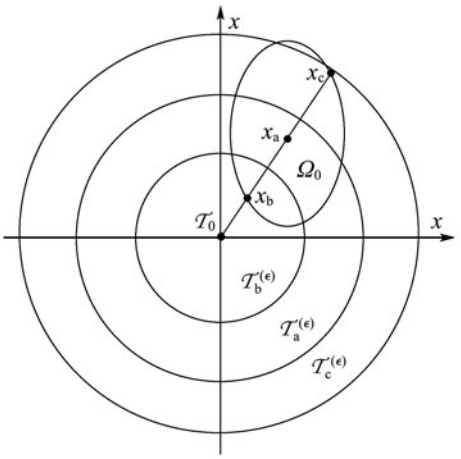

Fig. 1 The situation outline of the sates $x_{\rm a}$, $x_{\rm b}$ and $x_{\rm c}$.

Remark 2 Theorem 8 gives an important property obtaining the optimal control law. It means that if the initial set Ω_0 is convex, it is not necessary to search all the state points of the set. Instead, it only requires to search the boundary of the set, and therefore, the computational burden is much reduced.

4.4 The expressions of the ϵ-optimal control algorithm

Given the preparations, we now summarize the iterative ADP algorithms as follows:

Step 1 Give the initial state space Ω_0, the max iterative number $i_{\max}$, and the computation precision ϵ.

Step 2 Grid the state set Ω_0 into P subsets, which are expressed as $\Omega_0^{(1)}$, $\Omega_0^{(2)}$, $\ldots$, $\Omega_0^{(P)}$, where $\Omega_0 = \bigcup_{j=1}^{P} \Omega_0^{(j)}$ and $P > 0$ is a positive integer number.

Step 3 For $j = 1, 2, \ldots, P$, let X_0 be expressed as (43), and $x_0^{(j)}$ satisfies $x_0^{(j)} \in \Omega_0^{(j)}$.

Step 4 Set $i = 0$, choose the state point as $x_0 = x_k$ and $V_0(\cdot) = 0$. Set $K_\epsilon(x_0) = 0$.

Step 5 Compute $v_0(x_0)$ as in (6) and compute $V_1(x_0)$ as in (5).

Step 6 Set $i = i + 1$ and $K_\epsilon(x_0) = i$.

Step 7 Compute $v_i(x_0)$ as in (7) subjected to (1), and compute $V_i(x_0)$ as in (8).

Step 8 If $|V_{i+1}(x_0) - V_i(x_0)| \leqslant \epsilon$, then $x_0 \in \mathcal{T}_i^{(\epsilon)}$, $K_\epsilon(x_0) = i$, and go to Step 10. V_i is the ϵ-optimal performance index function $\mu_\epsilon^*(x_0) = u_i(x_0)$ is the ϵ-optimal control, and the optimal control step number $K_\epsilon(x_0) = i$. Otherwise, go to Step 9.

Step 9 If $i > i_{\max}$, then ϵ-optimal control with finite time does not exist and go to Step 12. Otherwise, go to Step 6.

Step 10 If $j < P$, then let I be expressed as (44); we get $x_0^{(j)} \in \mathcal{T}_I^{(\epsilon)}$ and $K_\epsilon(x_0^{(j)}) = I$; else goto Step 12.

Step 11 Record the corresponding performance index function $V_I(x_0^{(j)})$, and the control law $\mu_\epsilon^*(x_0^{(j)})$. Let $j = j + 1$, goto Step 3.

Step 12 Stop.

5 Neural network implementation for the ϵ-optimal control scheme

In the case of linear systems, the performance index function is quadratic and the control policy is linear. In the nonlinear case, this is not necessarily true, and therefore, we use neural networks to approximate u_k and $V_i(x_k)$.

Assume that the number of hidden layer neurons is denoted by l, the weight matrix between the input layer and hidden layer is denoted by V, and the weight matrix between the hidden layer and output layer is denoted by W. Then, the output of three-layer NN is represented by

$$\hat{F}(X, V, W) = W^{\rm T}\sigma(V^{\rm T}X), \quad (50)$$

where $\sigma(V^{\rm T}X) \in \mathbb{R}^l, [\sigma(z)]_i = \dfrac{{\rm e}^{z_i} - {\rm e}^{-z_i}}{{\rm e}^{z_i} + {\rm e}^{-z_i}}$, $i = 1, \ldots, l$ are the activation function.

The NN estimation error can be expressed by

$$F(X) = F(X, V^*, W^*) + \varepsilon(X), \quad (51)$$

where V^*, W^* are the ideal weight parameters, and $\varepsilon(X)$ is the reconstruction error.

Here, there are two networks, which are critic network and action network, respectively. Both neural networks are chosen as three-layer feedforward network. The whole structure diagram is shown in Fig. 2.

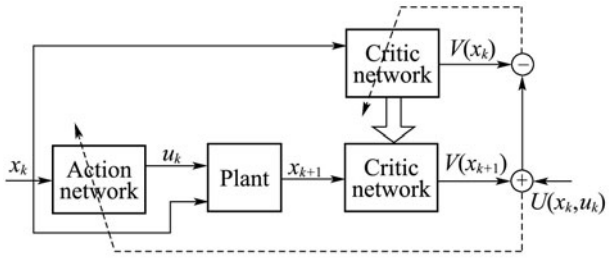

Fig. 2 The structure diagram of the algorithm.

5.1 The critic network

The critic network is used to approximate the performance index function $V_i(x_k)$. The output of the critic network is denoted as

$$\hat{V}_i(x_k) = W_{ci}^{\rm T}\sigma(V_{ci}^{\rm T}x_k). \quad (52)$$

The target function can be written as

$$V_{i+1}(x_k) = x_k^{\rm T}Qx_k + \hat{u}_i^{\rm T}(x_k)R\hat{u}_i(x_k) + \hat{V}_i(x_{k+1}). \quad (53)$$

Then, we define the error function for the critic network as

$$e_{ci}(k) = \hat{V}_{i+1}(x_k) - V_{i+1}(x_k). \quad (54)$$

The objective function to be minimized in the critic network

is

$$E_{ci}(k)=\frac{1}{2}e_{ci}^2(k). \tag{55}$$

Therefore, the gradient-based weight update rule for the critic network is given by

$$w_{c(i+1)}(k)=w_{ci}(k)+\Delta w_{ci}(k), \tag{56}$$

$$\Delta w_{ci}(k)=\alpha_c\left[-\frac{\partial E_{ci}(k)}{\partial w_{ci}(k)}\right], \tag{57}$$

$$\frac{\partial E_{ci}(k)}{\partial w_{ci}(k)}=\frac{\partial E_{ci}(k)}{\partial \hat{V}_i(x_k)}\frac{\partial \hat{V}_i(x_k)}{\partial w_{ci}(k)}, \tag{58}$$

where $\alpha_c>0$ is the learning rate of critic network, and $w_c(k)$ is the weight vector of the critic network.

5.2 The action network

In the action network, the state error x_k is used as an input to create the optimal control difference as the output of the network. The output can be formulated as

$$\hat{v}_i(x_k)=W_{ai}^{\mathrm{T}}\sigma(V_{ai}^{\mathrm{T}}x_k). \tag{59}$$

The target of the output of the action network is given by (7). Thus, we can define the output error of the action network as

$$e_{ai}(k)=\hat{v}_i(x_k)-v_i(x_k). \tag{60}$$

The weights of the action network are updated to minimize the following performance error measure:

$$E_{ai}(k)=\frac{1}{2}e_{ai}^2(k). \tag{61}$$

The weights updating algorithm is similar to that of for the critic network. By the gradient descent rule, we can obtain

$$w_{a(i+1)}(k)=w_{ai}(k)+\Delta w_{ai}(k), \tag{62}$$

$$\Delta w_{ai}(k)=\beta_a\left[-\frac{\partial E_{ai}(k)}{\partial w_{ai}(k)}\right], \tag{63}$$

$$\frac{\partial E_{ai}(k)}{\partial w_{ai}(k)}=\frac{\partial E_{ai}(k)}{\partial e_{ai}(k)}\frac{\partial e_{ai}(k)}{\partial v_i(k)}\frac{\partial v_i(k)}{\partial w_{ai}(k)}, \tag{64}$$

where $\beta_a>0$ is the learning rate of action network.

6 Simulation study

To evaluate the performance of our iterative ADP algorithm, an example is chosen with quadratic utility functions for numerical experiments. The program is written in MATLAB and running on a Dell Optiplex 960 PC with Windows XP and Pentium IV processor.

Our example is chosen as the same example, as in [19]. We consider the following affine nonlinear system:

$$x_{k+1}=f(x_k)+g(x_k)u_k, \tag{65}$$

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

$$f(x_k)=\begin{bmatrix}0.2x_{1k}\exp(x_{2k}^2)\\ 0.3x_{2k}^3\end{bmatrix},$$

$$g(x_k)=\begin{bmatrix}-0.2 & 0\\ 0 & -0.2\end{bmatrix}.$$

The initial state is not a fixed point but belongs to a convex set Ω_0. The initial set is expressed as

$$\Omega_0=\{\,x_0=[x_{10}x_{20}]^{\mathrm{T}}\,|\,-1\leqslant x_{10}\leqslant 1.5,\ -1\leqslant x_{20}\leqslant 1\}. \tag{66}$$

The performance index function is in quadratic form with finite-time horizon that is expressed as

$$V(x_0)=\sum_{k=0}^{N-1}\left(x_k^{\mathrm{T}}Qx_k+u_k^{\mathrm{T}}Ru_k\right), \tag{67}$$

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

The computation error of the iterative ADP is given as $\epsilon=10^{-5}$. The neural network structure of the algorithm is shown in Fig. 2. The critic network and the action network are chosen as three-layer BP neural networks with the structure 2-8-1 and 2-8-2, respectively.

As the initial state set Ω_0 is convex, we can just search the boundary to obtain the optimal control law. We start at the point $(-1,-1)$, and the search step is 0.1 for each direction. We illustrate the convergence of performance index functions at eight points, which are

$$x_{\mathrm{A}}(-1,-1),\ \ x_{\mathrm{B}}(-1,1),\ \ x_{\mathrm{C}}(1.5,-1),\ \ x_{\mathrm{D}}(1.5,1),$$
$$x_{\mathrm{E}}(1,-0.5),\ \ x_{\mathrm{F}}(-1,0.5),\ \ x_{\mathrm{G}}(1.5,-0.5),\ \ x_{\mathrm{H}}(1,1).$$

The corresponding convergence trajectories are V^{A}, V^{B}, V^{C}, V^{D}, V^{E}, V^{F}, V^{G}, and V^{H} which are showed in Fig. 3, respectively.

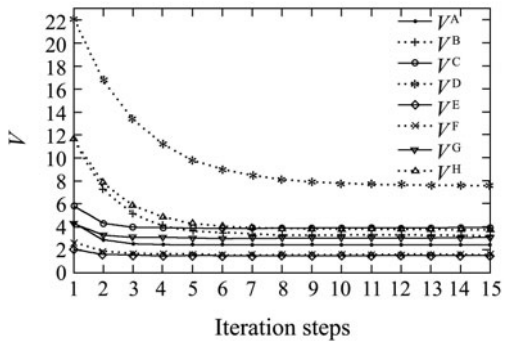

Fig. 3 The convergence of performance index functions

In Fig. 3, we can obtain

$$x_{\mathrm{A}}\in\mathcal{T}_4^{(\epsilon)},\ \ x_{\mathrm{B}}\in\mathcal{T}_6^{(\epsilon)},\ \ x_{\mathrm{C}}\in\mathcal{T}_4^{(\epsilon)},\ \ x_{\mathrm{D}}\in\mathcal{T}_7^{(\epsilon)},$$
$$x_{\mathrm{E}}\in\mathcal{T}_4^{(\epsilon)},\ \ x_{\mathrm{F}}\in\mathcal{T}_4^{(\epsilon)},\ \ x_{\mathrm{G}}\in\mathcal{T}_3^{(\epsilon)},\ \ x_{\mathrm{H}}\in\mathcal{T}_5^{(\epsilon)}.$$

Then, we have $I=7$, and $\mu_\epsilon^*(x_{\mathrm{D}})$ is the optimal control. To show the effectiveness of the optimal control, we arbitrarily choose five state points in Ω_0, such as

$$x_\alpha(-0.5,-0.5),\ \ x_\beta(1.5,1),\ \ x_\gamma(1,-1),$$
$$x_\delta(-1,1),\ \ x_\varepsilon(0.5,0.5).$$

Applying the optimal control law of $\mu_\epsilon^*(x_{\mathrm{D}})$ to the five state points, we obtain results shown in Figs. 4∼8.

The weights of the critic neural network can be expressed as

$$V_{\mathrm{c}}=[-0.0372,\ 0.6074;\ 0.5399,\ -0.0652;$$
$$0.5512,\ 0.0886;\ 0.2510,\ \ 0.2409;$$
$$0.5296,\ 0.4954;\ 0.6183,\ -0.1049;$$
$$0.2063,\ 0.3888;\ 0.4510,\ \ 0.1227],$$

$$W_{\mathrm{c}}=[0.7618,\ 0.8297,\ 1.0755,\ 0.9531,$$
$$1.2863,\ 0.9689,\ 0.5965,\ 0.8996].$$

The weights of the action neural network can be expressed as

$$\begin{aligned} V_{\rm a} = [&-0.1837, -0.2141; -0.1703, -0.1331;\\ &0.1387, \quad 0.5740; \quad 0.0102, \quad 0.0995;\\ &-0.2379, \quad 0.3453; \quad 0.3856, -0.2240;\\ &0.4053, -0.3282; -0.2559, \quad 0.2549;\\ &-0.2840, -0.3082; -0.0920, \quad 0.0409], \end{aligned}$$

$$\begin{aligned} W_{\rm a} = [&0.1506, \quad 0.4024, \quad 0.3322, -0.3552,\\ &0.2701, -0.0712, -0.1209, \quad 0.3973,\\ &-0.0935, -0.3311; -0.0751, -0.3773,\\ &-0.5238, -0.3431, \quad 0.1041, \quad 0.2021,\\ &0.2531, \quad 0.0186, -0.3601, \quad 0.1069]. \end{aligned}$$

Using the greedy HDP algorithm in [19], we can obtain $J^*(x_k) = 2.69671478308$ for $x = [-1 \quad 1]^{\rm T}$. Using the ϵ-optimal control that obtained by the proposed iterative ADP algorithm in this paper, we can obtain that $V_\epsilon(x_k) = 2.69671495855$. We have $V_\epsilon(x_k) - J^*(x_k) = 1.7547 \times 10^{-7}$. This illustrates the effectiveness of the ϵ-optimal control algorithm proposed in the paper.

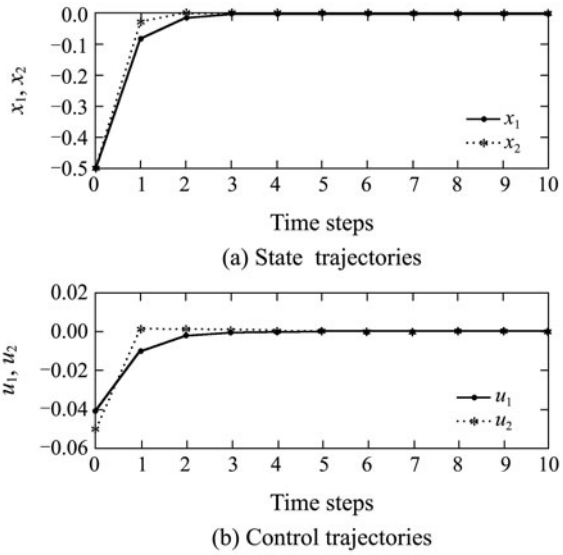

Fig. 4 Simulation results for the state $x_\alpha(-0.5, -0.5)$.

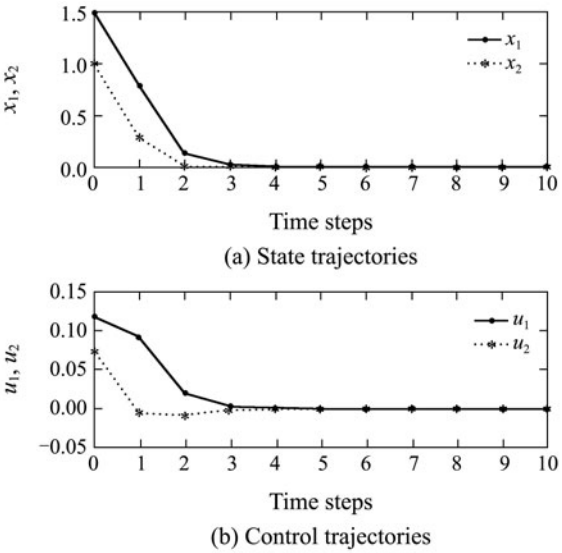

Fig. 5 Simulation results for the state $x_\beta(1.5, 1)$.

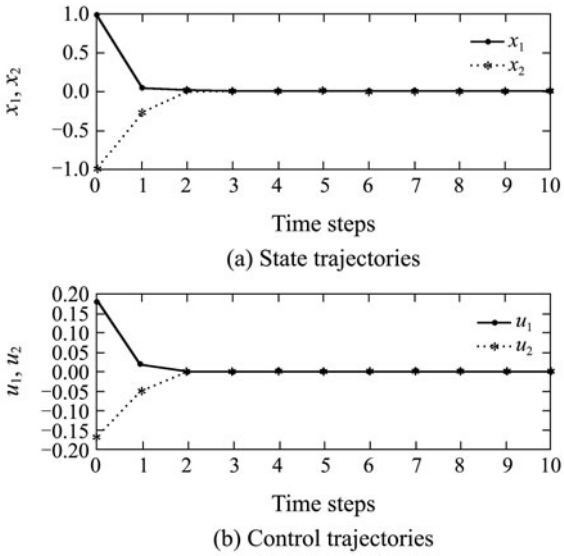

Fig. 6 Simulation results for the state $x_\gamma(1, -1)$.

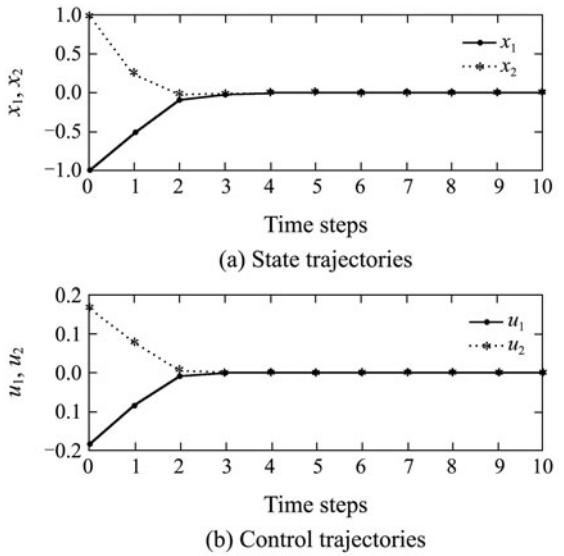

Fig. 7 Simulation results for the state $x_\delta(-1, 1)$.

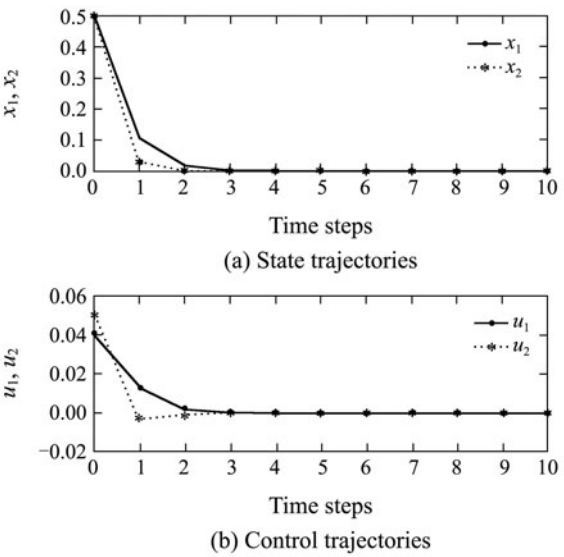

Fig. 8 Simulation results for the state $x_\varepsilon(0.5, 0.5)$.

7 Conclusions

In this paper, we solved the finite horizon optimal control problem for a class of discrete-time nonlinear systems using an adaptive dynamic programming (ADP) approach. The idea is to use an ADP technique to obtain the optimal control law iteratively, which makes the performance index function reach the optimum within finite time. The optimal number of control steps can also be obtained by the proposed ADP approach. Convergence analysis of the performance index function is proved. Neural networks are used to approximate the performance index function and compute the optimal control policy, respectively, for facilitating the implementation of the iterative ADP algorithm. A simulation example is given to illustrate the performance of the proposed method.

References

[1] H. G. Zhang, Y. Wang, D. Liu. Delay-dependent guaranteed cost control for uncertain stochastic fuzzy systems with multiple time delays. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 2008, 38(1): 125 – 140.

[2] M. R. Hsu, W. H. Ho, J. H. Chou. Stable and quadratic optimal control for T-S fuzzy-model-based time-delay control systems. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 2008, 38(4): 933 – 944.

[3] P. J. Goulart, E. C. Kerrigan, T. Alamo. Control of constrained discrete-time systems with bounded L_2 gain. *IEEE Transactions on Automatic Control*, 2009, 54(5): 1105 – 1111.

[4] J. H. Park, H. W. Yoo, S. Han, et al. Receding horizon controls for input-delayed systems. *IEEE Transactions on Automatic Control*, 2008, 53(7): 1746 – 1752.

[5] H. Zhang, L. Xie, G. Duan. H_∞ control of discrete-time systems with multiple input delays. *IEEE Transactions on Automatic Control*, 2007, 52(2): 271 – 283.

[6] H. Zhang, M. Li, J. Yang, et al. Fuzzy model-based robust networked control for a class of nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 2009, 39(2): 437 – 447.

[7] Z. Wang, H. Zhang, W. Yu. Robust stability of Cohen-Grossberg neural networks via state transmission matrix. *IEEE Transactions on Neural Networks*, 2009, 20(1): 169 – 174.

[8] L. Blackmore, S. Rajamanoharan, B. C. Williams. Active estimation for jump Markov linear systems. *IEEE Transactions on Automatic Control*, 2008, 53(10): 2223 – 2236.

[9] T. Cimen, S. P. Banks. Nonlinear optimal tracking control with application to super-tankers for autopilot design. *Automatica*, 2004, 40(11): 1845 – 1863.

[10] J. Azzato, J. B. Krawczyk. Applying a finite-horizon numerical optimization method to a periodic optimal control problem. *Automatica*, 2008, 44(6): 1642 – 1651.

[11] A. Ferrantea, G. Marrob, L. Ntogramatzidisb. A parametrization of the solutions of the finite-horizon LQ problem with general cost and boundary conditions. *Automatica*, 2005, 41(8): 1359 – 1366.

[12] N. Fukushima, M. S. Arslan, I. Hagiwara. An optimal control method based on the energy flow equation. *IEEE Transactions on Control Systems Technology*, 2009, 17(4): 866 – 875.

[13] I. Kioskeridis, C. Mademlis. A unified approach for four-quadrant optimal controlled switched reluctance machine drives with smooth transition between control operations. *IEEE Transactions on Automatic Control*, 2009, 24(1): 301 – 306.

[14] G. N. Saridis, F. Y. Wang. Suboptimal control of nonlinear stochastic systems. *Control-Theory and Advanced Technology*, 1994, 10(4): 847 – 871.

[15] A. Zadorojniy, A. Shwartz. Robustness of policies in constrained Markov decision processes. *IEEE Transactions on Automatic Control*, 2006, 51(4): 635 – 638.

[16] E. Zattoni. Structural invariant subspaces of singular hamiltonian systems and nonrecursive solutions of finite-horizon optimal control problems. *IEEE Transactions on Automatic Control*, 2008, 53(5): 1279 – 1284.

[17] H. Ichihara. Optimal control for polynomial systems using matrix sum of squares relaxations. *IEEE Transactions on Automatic Control*, 2009, 54(5): 1048 – 1053.

[18] J. Mao, C. G. Cassandras. Optimal control of multi-stage discrete event systems with real-time constraints. *IEEE Transactions on Automatic Control*, 2009, 54(1): 108 – 123.

[19] H. Zhang, Q. Wei, Y. Luo. A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 2008, 38(4): 937 – 942.

[20] H. Zhang, Y. Luo, D. Liu. The RBF neural network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraint. *IEEE Transactions on Neural Networks*, 2009, 20(9): 1490 – 1503.

[21] R. E. Bellman. *Dynamic Programming*. Princeton: Princeton University Press, 1957.

[22] V. G. Boltyanskii. *Optimal Control of Discrete Systems*. New York: John Wiley & Sons, 1978.

[23] A. E. Bryson, Y. C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. New York: Hemisphere Publishing Co., 1975.

[24] Q. Wei, H. Zhang, J. Dai. Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions. *Neurocomputing*, 2009, 72(7/9): 1839 – 1848.

[25] D. Liu, X. Xiong, Y. Zhang. Action-dependent adaptive critic designs. *International Joint Conference on Neural Networks*. New York: IEEE, 2001: 990 – 995.

[26] J. Si, Y. Wang. On-line learning control by association and reinforcement. *IEEE Transactions on Neural Networks*, 2001, 12(2): 264 – 276.

[27] J. J. Murray, C. J. Cox, G. G. Lendaris, et al. Adaptive dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Cybernetics*, 2002, 32(2): 140 – 153.

[28] D. Liu, H. Zhang. A neural dynamic programming approach for learning control of failure avoidance problems. *International Journal of Intelligence Control and Systems*, 2005, 10(1): 21 – 32.

[29] T. Landelius. *Reinforcement Learning and Distributed Local Model Synthesis*. Sweden: Linkoping University, 1997.

[30] D. Liu, Y. Zhang, H. Zhang. A self-learning call admission control scheme for CDMA cellular networks. *IEEE Transactions on Neural Networks*, 2005, 16(5): 1219 – 1228.

[31] N. Jin, D. Liu, T. Huang, et al. Discrete-time adaptive dynamic programming using wavelet basis function neural networks. *Proceedings of the IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning*. New York: IEEE, 2007: 135 – 142.

[32] A. Al-Tamimi, F. L. Lewis. Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. *Proceedings of the IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning*, New York: IEEE, 2007: 38 – 43.

[33] A. Al-Tamimi, M. Abu-Khalaf, F. L. Lewis. Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 2007, 37(1): 240 – 247.

[34] H. Zhang, Q. Wei, D. Liu. An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. *Automatica*, 2011, 47(1): 207 – 214.

[35] P. J. Werbos. A menu of designs for reinforcement learning over time. W. T. Miller, R. S. Sutton, P. J. Werbos, eds. *Neural Networks for Control*, Cambridge: MIT Press, 1991: 67 – 95.

[36] D. V. Prokhorov, D. C. Wunsch. Adaptive critic designs. *IEEE Transactions on Neural Networks*, 1997, 8(5): 997 – 1007.

[37] P. J. Werbos. Approximate dynamic programming for real-time control and neural modeling. D. A. White, D. A. Sofge, eds. *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, New York: Van Nostrand Reinhold, 1992: 493 – 525

[38] F. Wang, N. Jin, D. Liu, et al. Adaptive dynamic programming for finite horizon optimal control of discrete-time nonlinear systems with ε-error bound. *IEEE Trasctions on Neural Networks*, 2011, 22(1): 24 – 36.

Qinglai WEI received his B.S. degree in Automation, M.S. degree in Control Theory and Control Engineering, and Ph.D. degree in Control Theory and Control Engineering, from the Northeastern University, Shenyang, China, in 2002, 2005, and 2008, respectively. He is currently a postdoctoral fellow with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include neural-networks-based control, nonlinear control, adaptive dynamic programming, and their industrial applications. E-mail: qinglaiwei@gmail.com.

Derong LIU received his Ph.D. degree in Electrical Engineering from the University of Notre Dame, Notre Dame, IN, in 1994. Dr. Liu was a staff fellow with 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 at Chicago in 1999, where he 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.

Currently, Dr. Liu is the editor-in-chief of the IEEE Transactions on Neural Networks and an associate editor of several other journals. He received the Michael J. Birck Fellowship from the University of Notre Dame (1990), the Harvey N. Davis Distinguished Teaching Award from Stevens Institute of Technology (1997), the Faculty Early Career Development (CAREER) Award from the National Science Foundation (1999), the University Scholar Award from University of Illinois (2006), and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China (2008). E-mail: derongliu@gmail.com.