Neural Comput & Applic (2013) 23:1851–1863
DOI 10.1007/s00521-012-1188-7

ISNN2012

Dual iterative adaptive dynamic programming for a class of discrete-time nonlinear systems with time-delays

Qinglai Wei · Ding Wang · Dehua Zhang

Received: 30 April 2012 / Accepted: 13 September 2012 / Published online: 17 October 2012

Abstract In this paper, a new dual iterative adaptive dynamic programming (ADP) algorithm is developed to solve optimal control problems for a class of nonlinear systems with time-delays in state and control variables. The idea is to use the dynamic programming theory to solve the expressions of the optimal performance index function and control. Then, the dual iterative ADP algorithm is introduced to obtain the optimal solutions iteratively, where in each iteration, the performance index function and the system states are both updated. Convergence analysis is presented to prove the performance index function to reach the optimum by the proposed method. Neural networks are used to approximate the performance index function and compute the optimal control policy, respectively, for facilitating the implementation of the dual iterative ADP algorithm. Simulation examples are given to demonstrate the validity of the proposed optimal control scheme.

Keywords Adaptive dynamic programming · Approximate dynamic programming · Adaptive critic designs · Optimal control · Time-delay · Nonlinear systems

1 Introduction

Strictly speaking, time-delays exist in most practical control systems, which mainly result from the time taken in the online data acquisition of sensors, the time taken in the processing of the sensory data, the time taken by the actuator to produce the required control force, and so on. Time-delays may result in degradation of the control efficiency even instability of the control systems [21]. So, there have been many studies on the control systems with time-delays in various research fields such as power systems control, chemical process control, and networked control [10–12, 15, 26, 28, 48]. The optimal control problem with time-delays has been the key focus in the control field in the last several decades [4, 25, 42]. As the systems with time-delays are generally infinite-dimensional systems [21], the optimal control problem with time-delays generates some of the most challenging problems in control engineering. Lots of analysis and applications are limited to simple cases: linear systems with only state delays [5, 8], or the linear systems with only control delays [4, 22]. For nonlinear case, traditional method is to adopt fuzzy method or robust method which transforms the nonlinear time-delay system to a linear one [44, 48]. For the systems with time-delays both in states and controls, the optimal controller contains the delayed state and control information which makes the analysis of the system very difficult. Till now, it is still a open problem for the optimal control problem for the system with time-delays both in states and controls [25]. This motivates our research.

As is well known, dynamic programming is a very useful tool in solving the optimal control problems. However, due to the "curse of dimensionality" [6], it is often computationally untenable to run dynamic programming to obtain the optimal solution. Although there are some intelligent control methods to overcome the curse of dimensionality [17–19, 30, 33, 41, 43], most of them are only considered as the stability of the system.

The adaptive dynamic programming (ADP) algorithm was proposed by [37, 39] as a way to solve optimal control problems forward-in-time. In [38], ADP approaches were

Q. Wei (✉) · D. Wang · D. Zhang
State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, People's Republic of China
e-mail: qinglaiwei@gmail.com

classified into four main schemes: heuristic dynamic programming (HDP), dual heuristic dynamic programming (DHP), action-dependent heuristic dynamic programming (ADHDP), also known as Q-learning [35], and action-dependent dual heuristic dynamic programming (ADDHP). In [24], another two ADP schemes known as globalized-DHP (GDHP) and ADGDHP were developed. In [23], a convergent ADP algorithm was developed for stabilizing the continuous-time nonlinear systems. In [45], a greedy HDP iteration algorithm to solve the discrete-time Hamilton–Jacobi–Bellman (HJB) equation of the optimal control problem for general nonlinear discrete-time systems was proposed, which does not require an initially stable policy. It was proved in [3] that the greedy HDP iteration algorithm is convergent. Though in recent years, ADP has been further studied by many researchers [1, 2, 7, 9, 13, 16, 20, 31, 32, 34, 36, 40, 46], most of discussions are focus on the optimal problems without time-delays. Only in [29, 47], state delays were considered to obtain the optimal control by ADP. To the best of our knowledge, there have been no results discussing how to use ADP to solve the optimal control problems for nonlinear systems with time-delays in both state and control variables.

In this paper, it is the first time that the optimal control problem for a class of nonlinear system with time-delays in both state and control variables is solved by a dual iterative ADP algorithm. The significance of the algorithm is that in each iteration, the performance index function and the iterative states both update according to the iterative control policy, where the infinite-dimensional controller is effectively avoided. Next, it will show that the dual iterative ADP algorithm can obtain the optimal control that makes the performance index function converge to the optimum. Furthermore, in order to facilitate the implementation of the dual iterative ADP algorithm, we show how to introduce neural networks to obtain the iterative performance index functions.

2 Problem statement

Basically, we consider the following discrete-time affine nonlinear system with multiple time-delays in state and control variables

$$\begin{aligned} x(k+1) = & f(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m)) \\ & + g_0(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m)) \\ & \times u(k-\tau_0) \\ & + g_1(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m)) \\ & \times u(k-\tau_1) \\ & \ldots \\ & + g_n(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m)) \\ & \times u(k-\tau_n) \end{aligned} \tag{1}$$

where $\sigma_0, \ldots, \sigma_m$ are the state delays and $\tau_0 = 0, \tau_1, \ldots, \tau_n$ are the control delays. Set $\sigma_0 = 0$ and $\forall \sigma_i, i = 1, \ldots, m$, is positive integer number. Set $\tau_0 = 0$ and $\forall \tau_i, i = 1, 2, \ldots, n$, is also positive integer number. Without loss of generality, let $\sigma_0 \le \sigma_1 \le \cdots \le \sigma_m$ and $\tau_0 \le \tau_1 \le \cdots \le \tau_n$. Here, $x(k) \in \Re^n$ is the state variable and $u(k) \in \Re^m$ denotes the control variable. The initial condition is given by $x(s) = \phi(s), s \in \{-\sigma_m, -\sigma_m + 1, \ldots, -1, 0\}$ and $u(r) = 0$ for $r < 0$. Assume that $f, g_0, g_1, \ldots, g_n$ are all Lipschitz continuous functions and the system (1) is controllable in $\Re^n$. In this paper, we mainly discuss how to design an optimal feedback controller for discrete-time nonlinear systems with multiple time-delays (1). Therefore, it is desired to find a sequence of control $\underline{u}_k = \{u(k), u(k+1), \ldots\}$ to minimize the following generalized quadratic performance index function

$$V(x(0), \underline{u}_0) = \sum_{k=0}^{\infty} \left\{ X^T(k) Q X(k) + U^T(k) R U(k) \right\} \tag{2}$$

where $X(k) = [x^T(k-\sigma_0) \cdots x^T(k-\sigma_m)]^T$ and $U(k) = [u^T(k-\tau_0) \cdots u^T(k-\tau_n)]^T$. Let $Q > 0$ and $R > 0$ be both positive definite matrices with suitable dimensions. Let $l(X(k), U(k)) = X^T(k) Q X(k) + U^T(k) R U(k)$ be the utility function.

Let $V^*(x^*(k))$ denote the optimal performance index function which satisfies

$$V^*(x^*(k)) = \min_{\underline{u}_k} V(x(k), \underline{u}_k) = V(x^*(k), \underline{u}_k^*) \tag{3}$$

where $\underline{u}_k^*$ is the optimal control sequence and $x^*(k)$ is the resultant optimal state at time k under the optimal control sequence $\underline{u}_k^*$.

For the convenience, we let

$$F(X(k)) = f(x(k-\sigma_0), \ldots, x(k-\sigma_m)) \tag{4}$$

and

$$\begin{aligned} G(X(k)) = & [g_0(x(k-\sigma_0), \ldots, x(k-\sigma_m)), \\ & g_1(x(k-\sigma_0), \ldots, x(k-\sigma_m)), \\ & \ldots, \\ & g_n(x(k-\sigma_0), \ldots, x(k-\sigma_m))], \end{aligned} \tag{5}$$

and then (1) can be expressed as

$$x(k+1) = F(X(k)) + G(X(k)) U(k). \tag{6}$$

According to the Bellman's optimal principle, we can get the following HJB equation

$$\begin{aligned} V^*(x^*(k)) = & \min_{U(k)} \{ X^T(k) Q X(k) + U^T(k) R U(k) \\ & + V^*(x(k+1)) \} \\ = & X^{*T}(k) Q X^*(k) + U^{*T}(k) R U^*(k) \\ & + V^*(x^*(k+1)) \end{aligned} \tag{7}$$

where $U^*(k) = [u^*(k-\tau_0)u^*(k-\tau_1)\cdots u^*(k-\tau_n)]^T$ and $X^*(k) = [x^*(k-\sigma_0)x^*(k-\sigma_1)\cdots x^*(k-\sigma_m)]^T$ is the resultant optimal state sequence of $U^*(k)$.

By a sequence of transformations, we can see that the term with time-delays has been avoided in the HJB equation (7). But, it should be pointed out that (7) is different from the HJB equation of delay free systems. One obvious difference is that in (7), for $\forall\ k$, we need a sequence of optimal control, i.e., $\{u^*(k-\tau_n), u^*(k-\tau_{n-1}), \ldots, u^*(k-\tau_1), u^*(k)\}$, to obtain the optimal performance index function $V^*(x^*(k))$. While for the HJB equation of system without time-delays, we only need a optimal control at time k, i.e., $u(k)$, to obtain the optimal performance index function. Second, the optimal controls in the sequence $\{u^*(k-\tau_n), u^*(k-\tau_{n-1}), \ldots, u^*(k-\tau_1), u^*(k)\}$ of $U^*(k)$ couple with each other, while for the HJB equation of system without time-delays, the optimal control of current time $u^*(k)$ does not couple with the one of previous time. So, we say that time-delays do not be avoided in the HJB equation (7) essentially. The HJB equation with time-delays is more complex than the one without time-delays, and it is nearly impossible to obtain the optimal control sequence $\{u^*(k-\tau_n), u^*(k-\tau_{n-1}), \ldots, u^*(k-\tau_1), u^*(k)\}$ by solving (7) directly. To overcome the difficulties, a new dual iterative ADP algorithm is developed in this paper.

3 Dual iterative adaptive dynamic programming algorithm

In this section, dual iterative ADP approach is developed to obtain the optimal control for nonlinear system with time-delays. The goal of the proposed dual iterative ADP method is to use adaptive critic design technique to adaptively construct an optimal control sequence $\underline{u}_k^*$, which takes an arbitrary initial state $x(0)$ to the singularity 0 and simultaneously makes the performance index function reach the optimum $V^*(x^*(k))$ with convergence proofs.

3.1 Derivation of the iterative ADP algorithm

By solving the HJB equation (7), we can obtain the optimal control sequence $U^*(k)$ expressed as

$$\begin{aligned} U^*(k) &= \arg\min_{U(k)}\{X^T(k)QX(k) + U^T(k)RU(k) \\ &\qquad + V^*(X(k+1))\} \\ &= -\frac{1}{2}R^{-1}G^T(X^*(k))\frac{\partial V^*(x^*(k+1))}{\partial X^*(k+1)}. \end{aligned} \tag{8}$$

As the optimal performance index function $V^*(x^*(k))$ is unknown, it is nearly impossible to obtain the optimal control sequence from (8). So, an iterative index i is introduced in the algorithm.

In the dual iterative ADP algorithm, the iterative performance index functions are updated by recurrent iteration. The iterative control policies and the corresponding state are also updated by recurrent iteration, with the iteration number i increasing from 0 to ∞. First, set the iteration index $i = 0$. We start with initial performance index $V^{(0)}(\cdot) = 0$. Then, for $\forall k \geq 0$, the iterative control sequence $U^{(0)}(k)$ can be computed as follows:

$$U^{(0)}(k) = \arg\min_{U(k)}\{X^T(k)QX(k) + U^T(k)RU(k)\} \tag{9}$$

Then we update the performance index function as

$$\begin{aligned} V^{(1)}(x^{(0)}(k)) &= \min_{U(k)}\{X^T(k)QX(k)+U^T(k)RU(k)\} \\ &= X^{(0)T}(k)QX^{(0)}(k)+U^{(0)T}(k)RU^{(0)}(k) \end{aligned} \tag{10}$$

where $U^{(0)}(k) = [u^{(0)T}(k-\tau_0), \ldots, u^{(0)T}(k-\tau_n)]^T$ and $X^{(0)}(k) = [x^{(0)T}(k-\sigma_0), \ldots, x^{(0)T}(k-\sigma_m)]^T$.

We can see that before we obtain the iterative performance index function $V^{(1)}(x^{(0)}(k))$, we can obtain $U^{(0)}(k)$ from (9). It should be noticed that $U^{(0)}(k)$ contains a sequence of control $\{u^{(0)}(k-\tau_0), u^{(0)}(k-\tau_1), \ldots, u^{(0)}(k-\tau_n)\}$ which couples with each other. So, for $\forall\ k$, we cannot obtain $U^{(0)}(k)$ directly by solving (9). To overcome this difficulty, we start the algorithm at $j = 0$. Then, we have $u(j-\tau_1) = u(k-\tau_j) = \cdots = u(j-\tau_n) = 0$ and $x(j-\sigma_1) = \phi(j-\sigma_1), \ldots, x(j-\sigma_m) = \phi(j-\sigma_m)$. Then, we can easily obtain $U^{(0)}(0)$ and $V^{(1)}(x^{(0)}(0))$ from (9) and (10), respectively, and then, we can obtain $x^{(0)}(1)$ from (6). Then, start the algorithm at $j = 1$. As $U^{(0)}(0)$ and $x^{(0)}(1)$ both are known, we then can obtain $U^{(0)}(1)$ and $V^{(1)}(x^{(0)}(1))$ from (9) and (10), respectively. Then, after running the algorithm at $j = 0, 1, \ldots, k-1$, we can obtain a iterative control sequence expressed as $\{u^{(0)}(0), u^{(0)}(1), \ldots, u^{(0)}(k-1)\}$ and a iterative state sequence expressed as $\{x^{(0)}(0), x^{(0)}(1), \ldots, x^{(0)}(k)\}$. Then, running the algorithm at $j = k$, we can obtain $U^{(0)}(k)$ and $V^{(1)}(x^{(0)}(k))$ from (9) and (10), respectively.

For the iterative index $i = 1, 2, \ldots$ and $\forall k \geq 0$, the iterative ADP algorithm update the iterative control $U^{(i)}(k)$ by

$$\begin{aligned} U^{(i)}(k) = \arg\min_{U(k)}\{&X^T(k)QX(k) + U^T(k)RU(k). \\ &+V^{(i)}(\mathcal{X}^{(i-1)}(k+1))\}. \end{aligned} \tag{11}$$

Then we update the performance index function by

$$\begin{aligned} V^{(i+1)}(x^{(i)}(k)) &= \min_{U(k)}\{X^T(k)QX(k) + U^T(k)RU(k) \\ &\qquad + V^{(i)}(\mathcal{X}^{(i-1)}(k+1))\} \\ &= X^{(i)T}(k)QX^{(i)}(k) + U^{(i)T}(k)RU^{(i)}(k) \\ &\quad + V^{(i)}(x^{(i-1)}(k+1)), \end{aligned} \tag{12}$$

where $U^{(i)}(k) = [u^{(i)T}(k-\tau_0)\cdots u^{(i)T}(k-\tau_n)]^T$. Let $V^{(i)}(\mathcal{X}^{(i-1)}(k+1))$ be expressed as

$$\begin{aligned} &V^{(i)}(\mathcal{X}^{(i-1)}(k+1)) \\ &= V^{(i)}(f(x^{(i-1)}(k-\sigma_0),\ldots,x^{(i-1)}(k-\sigma_m)) \\ &\quad + g_0(x^{(i-1)}(k-\sigma_0),\ldots,x^{(i-1)}(k-\sigma_m))u(k-\tau_0) \\ &\quad + g_1(x^{(i-1)}(k-\sigma_0),\ldots,x^{(i-1)}(k-\sigma_m))u(k-\tau_1) \\ &\quad \cdots \\ &\quad + g_n(x^{(i-1)}(k-\sigma_0),\ldots,x^{(i-1)}(k-\sigma_m))u(k-\tau_n)) \end{aligned} \tag{13}$$

and $V^{(i)}(x^{(i-1)}(k+1))$ be expressed as

$$\begin{aligned} &V^{(i)}(x^{(i-1)}(k+1)) \\ &= V^{(i)}(f(x^{(i-1)}(k-\sigma_0),\ldots,x^{(i-1)}(k-\sigma_m)) \\ &\quad + g_0(x^{(i-1)}(k-\sigma_0),\ldots,x^{(i-1)}(k-\sigma_m))u^{(i)}(k-\tau_0) \\ &\quad + g_1(x^{(i-1)}(k-\sigma_0),\ldots,x^{(i-1)}(k-\sigma_m))u^{(i)}(k-\tau_1) \\ &\quad \cdots \\ &\quad + g_n(x^{(i-1)}(k-\sigma_0),\ldots,x^{(i-1)}(k-\sigma_m))u^{(i)}(k-\tau_n)). \end{aligned} \tag{14}$$

Following the idea of the iterative algorithm for $i = 0$, for the iterative index $i = 1, 2, \ldots$, the iterative ADP algorithm firstly runs at $j = 0$ and then obtain $U^{(i)}(0)$ and $V^{(i+1)}(x^{(i)}(0))$ from (11) and (12), respectively, and then we can obtain $x^{(i)}(1)$ from (6). Then, start the algorithm at $j = 1$. As $U^{(i)}(0)$ and $x^{(i)}(1)$ both are known, we then can obtain $U^{(i)}(1)$ and $V^{(i+1)}(x^{(i)}(1))$ from (11) and (12), respectively. Then, after running the algorithm at $j = 0, 1, \ldots, k-1$, we can obtain an iterative control sequence expressed as $\{u^{(i)}(0), u^{(i)}(1), \ldots, u^{(i)}(k-1)\}$ and an iterative state sequence expressed as $\{x^{(i)}(0), x^{(i)}(1), \ldots, x^{(i)}(k)\}$. Then, running the algorithm at $j = k$ and we can obtain $U^{(i)}(k)$ and $V^{(i+1)}(x^{(i)}(k))$ from (11) and (12), respectively.

There is an important property that we must point out. The iterative ADP algorithm from (9) to (12) is different from the value iteration algorithm in [3]. In [3], the state vector $x(k)$ is arbitrarily chosen in the state space. While in this paper, as the system (1) is a time-delay system, we can see that the state vectors $x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m)$ and control vectors $u(k-\tau_0), u(k-\tau_1), \ldots, u(k-\tau_n)$ couple with each other. So, it is impossible to arbitrarily chosen the expansion state sequence $X(k)$ in the state space. On the other side, for the value iteration algorithm in [3], the state $x(k)$ is fixed for any iteration index $i = 0, 1, \ldots$ and the iterative performance index functions are updated according to the optimality conditions. While in this paper, the iterative performance index functions are updated as the iterative index i increase from 0 to ∞. Simultaneously, the iterative state sequence $X^{(i)}(k)$ also updates when the iterative index i increase from 0 to ∞. So, for different iteration index, such as $i \neq j$, we have iterative state sequence $X^{(i)}(k) \neq X^{(j)}(k)$. To obtain the iterative state sequence $X^{(i)}(k)$, the iterative algorithm has to be implemented at $j = 0, 1, \ldots, k-1$, respectively. So, the value iteration algorithm in [3] is invalid for the nonlinear system with time-delays (1). The dual iterative ADP algorithm proposed in this paper is called performance index function and state iterative ADP algorithm (dual iterative ADP algorithm, for brief).

Remark 1 Generally speaking, the optimal control problem for the systems with time-delays belongs to the infinite-dimensional control problem [21]. The design of the infinite-dimensional controller and the analysis of the system are both very difficult. While from the dual iterative ADP algorithm proposed in this paper, we can see that for $\forall i = 0, 1, \ldots$, the iterative control $U^{(i)}(k)$ is updated only using the previous iteration information which has been obtained. So, the infinite-dimensional controller in [21] is effectively avoided.

Remark 2 For $i = 0, 1, \ldots$ and $j = 0, 1, \ldots, k$, as the performance index functions and the system states both update according to the iterative control sequence $U^{(i)}(j)$, this means the iterative state sequence $\{x^{(i)}(j-\sigma_0), \ldots, x^{(i)}(j-\sigma_m)\}$ in $X^{(i)}(j)$ and the iterative control sequence $\{u^{(i)}(j-\tau_0), u^{(i)}(j-\tau_1), \ldots, u^{(i)}(j-\tau_n)\}$ in $U^{(i)}(j)$ both need to update. While according to the initial condition of the system (1), we have $x(s) = \phi(s), s \in \{-\sigma, -\sigma+1, \ldots, -1, 0\}$ and $u(r) = 0$ for $r < 0$. This means if the implementation time $k \leq \max\{\sigma_m, \tau_n\}$, there may exist one or more state or control variables which belong to the initial condition and cannot be updated. Therefore, to implement the dual iterative ADP algorithm effectively, we must let the implementation time $k > \max\{\sigma_m, \tau_n\}$.

Remark 3 During the process of the dual iterative ADP algorithm, as the iterative states update from $j = 0$ to k, the nonlinear system with time-delays transforms frequently. For example, when $\sigma_h < j \leq \sigma_l$, where $h, l = 0, 1, \ldots, m$ and $h < l$, we can see that the state sequence $\{x(\sigma_l), x(\sigma_{l+1}), \ldots, x(\sigma_m)\}$ belongs to the initial state and is uncontrollable. When $\tau_h \leq j < \tau_l$, where $h, l = 0, 1, \ldots, n$ and $h < l$, then, we have the controls $u(k-\tau_l) = u(k-\tau_{l+1}) = \cdots = u(k-\tau_n) = 0$ which means that these controls are invalid. If we combine the two sequence $\{\sigma_h\}, h = 0, 1, \ldots, m$ and $\{\tau_l\}, l = 0, 1, \ldots, n$ into one sequence $\{\lambda_0, \lambda_1, \ldots, \lambda_{m+n}\}$ where $\lambda_0 = 0, \lambda_{m+n} = \max\{\sigma_m, \tau_n\}$ and $\lambda_h \leq \lambda_l, h, l = 0, 1, \ldots, \max\{\sigma_m, \tau_n\}$ then, we can say that the iterative performance index functions $V^{(i+1)}(x(k))$ and the law of the iterative controls $U^{(i)}(k)$ will change at $k = \lambda_j, j = 0, 1, \ldots, \max\{\sigma_m, \tau_n\}$. For $k > \max\{\sigma_m, \tau_n\}$, we can see that all the states are controllable and all the controls

are effective to the system. Therefore, for the dual iterative ADP algorithm (11) and (12), all the iterative performance index functions and the law of the iterative control at $k = \lambda_j, j = 0, 1, \ldots, \max\{\sigma_m, \tau_n\}$ should be recorded.

Remark 4 The proposed dual iterative ADP algorithm is also effective for the nonlinear systems with state delays or control delays. In Sect. 5, we will give an example to show the effectiveness. For the nonlinear systems without time-delays, we can see that the current control $u(k)$ does not couple with the time-delayed control. So, the state iteration is not necessary. If we omit the state iteration and fix the state in the dual iterative ADP algorithm, then the iterative algorithm reduces to the value iteration algorithm proposed in [3].

From (9) to (12), it can be seen that during the iteration process, the control policies for different iteration steps have different control laws. For $U^{(i)}(\cdot), i = 0, 1, \ldots$ that obtained by (11), we run the delayed system (1) to obtain the corresponding state trajectory $x^{(i)}(1), x^{(i)}(2), \ldots, x^{(i)}(k)$. After ith iteration, the control laws sequence can be expressed as $\{U^{(0)}(\cdot), U^{(1)}(\cdot), \ldots, U^{(i)}(\cdot)\}$, which are different from each other. For the infinite-horizon problem, however, both the optimal performance index function and the optimal control law is unique. Therefore, it is necessary to show that the iterative performance index function $V^{(i+1)}(x^{(i)}(k))$ will converge when the iteration number $i \to \infty$ under the iterative control $U^{(i)}(k)$ and it will be proved in the following subsection.

3.2 Properties of the iterative ADP method

In this subsection, we focus on the proof of convergence of the dual ADP iterative algorithm between (9) to (12), with the performance index $V^{(i+1)}(x^{(i)}(k)) \to V^*(x^*(k)), \forall k$.

Theorem 1 *Let $\widetilde{U}^{(i)}(k), k = 0, 1\ldots$ be an arbitrary control and $U^{(i)}(k)$ is expressed as (11). Define $V^{(i+1)}(x^{(i)}(k))$ by (12) and $\Lambda^{(i+1)}(\tilde{x}^{(i)}(k))$ by*

$$\begin{aligned}\Lambda^{(i+1)}(\tilde{x}^{(i)}(k)) &= \widetilde{X}^{(i)T}(k)Q\widetilde{X}^{(i)}(k) + \widetilde{U}^{(i)T}(k)R\widetilde{U}^{(i)}(k)\\ &\quad + \Lambda^{(i)}(\tilde{x}^{(i-1)}(k+1))\end{aligned} \tag{15}$$

where $\Lambda^{(i)}(\tilde{x}^{(i-1)}(k+1))$ is expressed by

$$\begin{aligned}&\Lambda^{(i)}(\tilde{x}^{(i)}(k+1))\\ &= \Lambda^{(i)}(f(\tilde{x}^{(i-1)}(k-\sigma_0), \ldots, \tilde{x}^{(i-1)}(k-\sigma_m))\\ &\quad + g_0(\tilde{x}^{(i-1)}(k-\sigma_0), \ldots, \tilde{x}^{(i-1)}(k-\sigma_m))\tilde{u}^{(i)}(k-\tau_0)\\ &\quad + g_1(\tilde{x}^{(i-1)}(k-\sigma_0), \ldots, \tilde{x}^{(i-1)}(k-\sigma_m))\tilde{u}^{(i)}(k-\tau_1)\\ &\quad \cdots\\ &\quad + g_n(\tilde{x}^{(i-1)}(k-\sigma_0), \ldots, \tilde{x}^{(i-1)}(k-\sigma_m))\tilde{u}^{(i)}(k-\tau_n)).\end{aligned} \tag{16}$$

If for $\forall x(k), V^{(0)}(\cdot) = \Lambda^{(0)}(\cdot) = 0$, then $V^{(i+1)}(x^{(i)}(k)) \le \Lambda^{(i+1)}(\tilde{x}^{(i)}(k)), \forall i$.

Proof From the expression of $\Lambda^{(i+1)}(\tilde{x}^{(i)}(k))$, we can also derive the expression $\Lambda^{(i)}(\tilde{x}^{(i-1)}(k+1))$ as

$$\begin{aligned}\Lambda^{(i)}(\tilde{x}^{(i-1)}(k+1)) &= \tilde{X}^{(i-1)T}(k+1)Q\tilde{X}^{(i-1)}(k+1)\\ &\quad + \tilde{U}^{(i-1)T}(k+1)R\tilde{U}^{(i-1)}(k+1)\\ &\quad + \Lambda^{(i-1)}(\tilde{x}^{(i-2)}(k+2)).\end{aligned} \tag{17}$$

Then (15) can be written as

$$\begin{aligned}\Lambda^{(i+1)}(\tilde{x}^{(i)}(k)) &= \tilde{X}^{(i)T}(k)Q\tilde{X}^{(i)}(k) + \tilde{U}^{(i)T}(k)R\tilde{U}^{(i)}(k)\\ &\quad \tilde{X}^{(i-1)T}(k+1)Q\tilde{X}^{(i-1)}(k+1)\\ &\quad + \tilde{U}^{(i-1)T}(k+1)R\tilde{U}^{(i-1)}(k+1)\\ &\quad + \Lambda^{(i-1)}(\tilde{x}^{(i-2)}(k+2)).\end{aligned} \tag{18}$$

Using the idea of iteration, we have

$$\begin{aligned}\Lambda^{(i+1)}(\tilde{x}^{(i)}(k)) &= \tilde{X}^{(i)T}(k)Q\tilde{X}^{(i)}(k) + \tilde{U}^{(i)T}(k)R\tilde{U}^{(i)}(k)\\ &\quad + \tilde{X}^{(i-1)T}(k+1)Q\tilde{X}^{(i-1)}(k+1)\\ &\quad + \tilde{U}^{(i-1)T}(k+1)R\tilde{U}^{(i-1)}(k+1)\\ &\quad \vdots\\ &\quad + \tilde{X}^{(0)T}(k+i)Q\tilde{X}^{(0)}(k+i)\\ &\quad + \tilde{U}^{(0)T}(k+i)R\tilde{U}^{(0)}(k+i).\end{aligned} \tag{19}$$

For $j = 0, 1, \ldots, i$, let

$$\begin{aligned}\tilde{L}(k+j) &= \tilde{X}^{(i-j)T}(k+j)Q\tilde{X}^{(i-j)}(k+j)\\ &\quad + \tilde{U}^{(i-j)T}(k+j)R\tilde{U}^{(i-j)}(k+j).\end{aligned} \tag{20}$$

Then (15) can be written as

$$\Lambda^{(i+1)}(\tilde{x}^{(i)}(k)) = \sum_{j=0}^{i}\tilde{L}(k+j). \tag{21}$$

On the other side, according to (12), $V^{(i+1)}(x^{(i)}(k))$ can be expressed as

$$\begin{aligned}&V^{(i+1)}(x^{(i)}(k))\\ &= \min_{U(k)}\{X^T(k)QX(k) + U^T(k)RU(k)\\ &\quad + \min_{U(k+1)}\{X^T(k+1)QX(k+1) + U^T(k+1)RU(k+1)\\ &\quad + \cdots\\ &\quad + \min_{U(k+i)}\{X^T(k+i)QX(k+i) + U^T(k+i)RU(k+i)\}\}\}\\ &= \min_{U(k),U(k+1),\ldots,U(k+i)}\left\{\sum_{j=0}^{i}L(k+j)\right\}.\end{aligned} \tag{22}$$

Then we have

$$V^{(i+1)}(x^{(i)}(k)) \le \Lambda^{(i+1)}(\tilde{x}^{(i)}(k)). \tag{23}$$

In order to prove the convergence of the performance index function, the following definition is necessary.

Definition 1 A control sequence $\underline{u}_k = \{u(k), u(k+1), \ldots\}$ is defined to be an admissible control sequence with respect to (2), if $\underline{u}_k$ stabilizes (1) and for $\forall$ $x(k)$, $V(x(k))$ is finite.

Then we have the following corollary.

Corollary 1 *Let the iterative performance index function $V^{(i+1)}(x^{(i)}(k))$ be defined by (12). If the system (1) is controllable, then there is an upper bound Y such that $0 \le V^{(i+1)}(x^{(i)}(k)) \le Y$, $\forall i$.*

Proof Let $\{\bar{u}^{(i)}(k)\}$ be any admissible control sequence. Then, for $i = 0, 1, \ldots$, we have $\bar{U}^{(i)}(k) = [\bar{u}^{(i)T}(k-\tau_0), \ldots, \bar{u}^{(i)T}(k-\tau_n)]^T$ is admissible. Define a new sequence $P^{(i)}(\bar{x}(k))$ as follows:

$$\begin{aligned} P^{(i+1)}(\bar{x}^{(i)}(k)) &= \bar{X}^{(i)T}(k)Q\bar{X}^{(i)}(k) + \bar{U}^{(i)T}(k)R\bar{U}^{(i)}(k) \\ &\quad + P^{(i)}(\bar{x}^{(i-1)}(k+1)) \end{aligned} \tag{24}$$

with $P^{(0)}(\cdot) = V^{(0)}(\cdot) = 0$, where $P^{(i)}(\bar{x}^{(i-1)}(k+1))$ is expressed by

$$\begin{aligned} &P^{(i)}(\bar{x}^{(i-1)}(k+1)) = P^{(i)}(f(\bar{x}^{(i-1)}(k-\sigma_0), \ldots, \bar{x}^{(i-1)}(k-\sigma_m)) \\ &\quad + g_0(\bar{x}^{(i-1)}(k-\sigma_0), \ldots, \bar{x}^{(i-1)}(k-\sigma_m))\bar{u}^{(i)}(k-\tau_0) \\ &\quad + g_1(\bar{x}^{(i-1)}(k-\sigma_0), \ldots, \bar{x}^{(i-1)}(k-\sigma_m))\bar{u}^{(i)}(k-\tau_1) \\ &\quad \cdots \\ &\quad + g_n(\bar{x}^{(i-1)}(k-\sigma_0), \ldots, \bar{x}^{(i-1)}(k-\sigma_m))\bar{u}^{(i)}(k-\tau_n)). \end{aligned} \tag{25}$$

From the iteration idea (17–21), we have

$$P^{(i+1)}(\bar{x}^{(i)}(k)) = \sum_{j=0}^{i} \bar{L}(k+j), \tag{26}$$

where

$$\begin{aligned} \bar{L}(k+j) &= \bar{X}^{(i-j)T}(k+j)Q\bar{X}^{(i-j)}(k+j) \\ &\quad + \bar{U}^{(i-j)T}(k+j)R\bar{U}^{(i-j)}(k+j). \end{aligned} \tag{27}$$

Noting that the control input $\{\bar{U}^{(i)}(k)\}$ is an admissible control sequence, we can obtain

$$\forall i : P^{(i+1)}(\bar{x}^{(i)}(k)) = \sum_{j=0}^{i-1} \bar{L}(k+j) \le \sum_{j=0}^{\infty} \bar{L}(k+j) \le Y. \tag{28}$$

From Lemma 1, we have

$$\forall i : V^{(i+1)}(x^{(i)}(k)) \le P^{(i+1)}(\bar{x}^{(i)}(k)) \le Y. \tag{29}$$

With Theorem 1 and Corollary 1, the following main theorem can be derived.

Theorem 2 *Define the iterative performance index function $V^{(i+1)}(x^{(i)}(k))$ as (12), with $V^{(0)}(\cdot) = 0$. If the system (1) is controllable, then $V^{(i+1)}(x^{(i)}(k))$ is a nondecreasing convergent sequence as $i \to \infty$.*

Proof For the convenience of analysis, define a new sequence $\Phi^{(i+1)}(x^{(i+1)}(k))$ as follows:

$$\begin{aligned} \Phi^{(i+1)}(x^{(i+1)}(k)) &= X^{(i+1)T}(k)QX^{(i+1)}(k) \\ &\quad + U^{(i+1)T}(k)RU^{(i+1)}(k) \\ &\quad + \Phi^{(i)}(x^{(i)}(k+1)) \end{aligned} \tag{30}$$

where $\Phi^{(i)}(x^{(i)}(k+1))$ is expressed by

$$\begin{aligned} &\Phi^{(i)}(x^{(i)}(k+1)) \\ &= \Phi^{(i)}(f(x^{(i)}(k-\sigma_0), \ldots, x^{(i)}(k-\sigma_m)) \\ &\quad + g_0(x^{(i)}(k-\sigma_0), \ldots, x^{(i)}(k-\sigma_m))u^{(i+1)}(k-\tau_0) \\ &\quad + g_1(x^{(i)}(k-\sigma_0), \ldots, x^{(i)}(k-\sigma_m))u^{(i+1)}(k-\tau_1) \\ &\quad \cdots \\ &\quad + g_n(x^{(i)}(k-\sigma_0), \ldots, x^{(i)}(k-\sigma_m))u^{(i+1)}(k-\tau_n)) \end{aligned} \tag{31}$$

with $\Phi^{(0)}(\cdot) = V^{(0)}(\cdot) = 0$ and $V^{(i+1)}(x^{(i)}(k))$ is updated by (12). In the following part, we prove $\Phi^{(i+1)}(x^{(i)}(k)) \le V^{(i+1)}(x^{(i)}(k))$ by mathematical induction.

First, we prove it holds for $i = 0$. Noting that

$$\begin{aligned} &V^{(1)}(x^{(0)}(k)) - \Phi^{(0)}(x^{(0)}(k)) \\ &= X^{(0)T}(k)QX^{(0)}(k) + U^{(0)T}(k)RU^{(0)}(k) \\ &\quad \ge 0, \end{aligned} \tag{32}$$

where the equal sign holds if and only if $X^{(0)}(k) = U^{(0)}(k) = 0$. Thus for $i = 0$, we can get

$$V^{(1)}(x^{(0)}(k)) \ge \Phi^{(0)}(x^{(0)}(k)). \tag{33}$$

Second, we assume it holds for $i-1$, i.e., $V^{(i)}(x^{(i-1)}(k)) - \Phi^{(i-1)}(x^{(i-1)}(k)) \ge 0, \forall x^{(i-1)}(k)$. Then, for i, because

$$\begin{aligned} \Phi^{(i)}(x^{(i)}(k)) &= X^{(i)T}(k)QX^{(i)}(k) \\ &\quad + U^{(i)T}(k)RU^{(i)}(k) \\ &\quad + \Phi^{(i-1)}(x^{(i-1)}(k+1)) \end{aligned} \tag{34}$$

and

$$\begin{aligned} V^{(i+1)}(x^{(i)}(k)) &= X^{(i)T}(k)QX^{(i)}(k) + U^{(i)T}(k)RU^{(i)}(k) \\ &\quad + V^{(i)}(x^{(i-1)}(k+1)). \end{aligned} \tag{35}$$

Thus, we can obtain

$$\begin{aligned} &V^{(i+1)}(x^{(i)}(k)) - \Phi^{(i)}(x^{(i)}(k)) \\ &\quad = V^{(i)}(x^{(i-1)}(k)) - \Phi^{(i-1)}(x^{(i-1)}(k)) \\ &\quad \geq 0, \end{aligned} \tag{36}$$

i.e.,

$$\Phi^{(i)}(x^{(i)}(k)) \leq V^{(i+1)}(x^{(i)}(k)). \tag{37}$$

Therefore, the mathematical induction proof is completed. Next, we will prove that $V^{(i)}(x^{(i-1)}(k)) \leq \Phi^{(i)}(x^{(i)}(k))$.

According to (15), let $\tilde{U}^{(i)}(k) = U^{(i+1)}(k)$, then we can get the corresponding state $\tilde{X}^{(i)}(k) = X^{(i+1)}(k)$. Hence we have $\Lambda^{(i+1)}(x^{(i+1)}(k)) = \Phi^{(i+1)}(x^{(i+1)}(k))$. According to Lemma 1, we have $V^{(i+1)}(x^{(i)}(k)) \leq \Lambda^{(i+1)}(x^{(i+1)}(k)) = \Phi^{(i+1)}(x^{(i+1)}(k))$. Replace i by $i-1$, and then we can obtain

$$V^{(i)}(x^{(i-1)}(k)) \leq \Phi^{(i)}(x^{(i)}(k)) \leq V^{(i+1)}(x^{(i)}(k)). \tag{38}$$

According to (29), we have $V^{(i+1)}(x^{(i)}(k))$ bounded. Hence, we conclude that $V^{(i+1)}(x^{(i)}(k))$ a nondecreasing convergent sequence as $i \to \infty$.

From Theorem 2, we know that the performance index function $V^{(i)}(x^{(i-1)}(k)) \geq 0$ is a monotonically nonincreasing convergent sequence. Then, we can define the performance index function $V^{(\infty)}(x^{(\infty)}(k))$ as the limit of the iterative function $V^{(i)}(x^{(i-1)}(k))$, i.e.,

$$V^{(\infty)}(x^{(\infty)}(k)) = \lim_{i\to\infty} V^{(i)}(x^{(i-1)}(k)). \tag{39}$$

Then we have the following corollary.

Corollary 2 *If the system (1) is controllable and the iterative performance index function $V^{(i+1)}(x^{(i)}(k))$ is convergent to $V^{(\infty)}(x^{(\infty)}(k))$ as $i \to \infty$, then we have the iterative control sequence $U^{(i)}(k)$ is also convergent, i.e.,*

$$U^{(\infty)}(k) = \lim_{i\to\infty} U^{(i)}(k). \tag{40}$$

Now, we can derive the following theorem.

Theorem 3 *If we let $V^{(\infty)}(x^{(\infty)}(k))$ be defined as (39) and $U^{(\infty)}(k)$ be defined as (40), then we have*

$$\begin{aligned} &V^{(\infty)}(x^{(\infty)}(k)) \\ &\quad = X^{(\infty)T}(k)QX^{(\infty)}(k) + U^{(\infty)T}(k)QU^{(\infty)}(k) \\ &\qquad + V^{(\infty)}(x^{(\infty)}(k+1)) \\ &\quad = \min_{U(k)}\{X^T(k)QX(k) + U^T(k)QU(k) \\ &\qquad + V^{(\infty)}(x(k+1))\}, \end{aligned} \tag{41}$$

where $x^{(\infty)}(k)$ is the corresponding state under the control sequence $U^{(\infty)}(k)$.

Proof According to Theorem 2 and (12), we have

$$\begin{aligned} V^{(\infty)}(x^{(\infty)}(k)) &\geq V^{(i+1)}(x^{(i)}(k)) \\ &= X^{(i)T}(k)QX^{(i)}(k) + U^{(i)T}(k)RU^{(i)}(k) \\ &\quad + V^{(i)}(x^{(i-1)}(k+1)) \\ &= \min_{U(k)}\{X^T(k)QX(k) + U^T(k)RU(k). \\ &\qquad + V^{(i)}(\mathcal{X}^{(i-1)}(k+1))\}, \end{aligned} \tag{42}$$

where $\mathcal{X}^{(i-1)}(k+1)$ is expressed in (12).

Let $i \to \infty$, and we have

$$\begin{aligned} V^{(\infty)}(x^{(\infty)}(k)) &\geq X^{(\infty)T}(k)QX^{(\infty)}(k) + U^{(\infty)T}(k)RU^{(\infty)}(k) \\ &\quad + V^{(\infty)}(x^{(\infty)}(k+1)) \\ &= \min_{U(k)}\{X^T(k)QX(k) + U^T(k)RU(k). \\ &\quad + V^{(\infty)}(\mathcal{X}^{(\infty)}(k+1))\}. \end{aligned} \tag{43}$$

Since $V^{(i)}(x^{(i-1)}(k))$ is nonincreasing for $i \geq 1$ and $V^{(\infty)}(x^{(\infty)}(k)) = \lim_{i\to\infty} V^{(i)}(x^{(i-1)}(k))$, for an arbitrary positive number $\varepsilon > 0$, there exists a positive integer p such that

$$V^{(p)}(x^{(p-1)}(k)) \leq V^{(\infty)}(x^{(\infty)}(k)) \leq V^{(p)}(x^{(p-1)}(k)) + \varepsilon.$$

From (12), we have

$$\begin{aligned} V^{(p)}(x^{(p-1)}(k)) &= X^{(p-1)T}(k)QX^{(p-1)}(k) \\ &\quad + U^{(p-1)T}(k)RU^{(p-1)}(k) \\ &\quad + V^{(p-1)}(x^{(p-2)}(k)). \end{aligned} \tag{44}$$

Hence,

$$\begin{aligned} &V^{(\infty)}(x^{(\infty)}(k)) \\ &\quad \leq X^{(p-1)T}(k)QX^{(p-1)}(k) + U^{(p-1)T}(k)RU^{(p-1)}(k) \\ &\qquad + V^{(p-1)}(x^{(p-2)}(k+1)) + \varepsilon \\ &\quad \leq X^{(p-1)T}(k)QX^{(p-1)}(k) + U^{(p-1)T}(k)RU^{(p-1)}(k) \\ &\qquad + V^{(\infty)}(x^{(\infty)}(k+1)) + \varepsilon \\ &\quad \leq \min_{U(k)}\{X^T(k)QX(k) + U^T(k)RU(k) \\ &\quad + V^{(\infty)}(\mathcal{X}^{(\infty)}(k+1))\} + \varepsilon. \end{aligned} \tag{45}$$

Since ε is arbitrary, we have

$$\begin{aligned} V^{(\infty)}(x^{(\infty)}(k)) \leq \min_{U(k)}\{&X^T(k)QX(k) + U^T(k)RU(k) \\ &+ V^{(\infty)}(\mathcal{X}^{(\infty)}(k+1))\}. \end{aligned} \tag{46}$$

Combining (43) and (46), we have

$$V^{(\infty)}(x^{(\infty)}(k)) = \min_{U(k)}\{X^T(k)QX(k) + U^T(k)RU(k) + V^{(\infty)}(\mathcal{X}^{(\infty)}(k+1))\}. \quad (47)$$

As $\{x^{(\infty)}(k)\}, k \geq 1$ is the state resultant sequence of the control sequence $\{u^{(\infty)}(k)\}, k \geq 0$, we have

$$\begin{aligned} &V^{(\infty)}(x^{(\infty)}(k)) \\ &= \min_{U(k)}\{X^T(k)QX(k) + U^T(k)RU(k) \\ &\quad + V^{(\infty)}(\mathcal{X}^{(\infty)}(k+1))\} \\ &= \min_{U(k)}\{X^T(k)QX(k) + U^T(k)QU(k) \\ &\quad + V^{(\infty)}(x(k+1))\} \\ &= X^{(\infty)T}(k)QX^{(\infty)}(k) + U^{(\infty)T}(k)QU^{(\infty)}(k) \\ &\quad + V^{(\infty)}(x^{(\infty)}(k+1)) \end{aligned} \quad (48)$$

which proves the result.

Theorem 4 *Let $V^{(i+1)}(x^{(i)}(k))$ be defined by (12). If the system state $x(k)$ is controllable, then we have the limit of the iterative performance index function $V^{(\infty)}(x^{(\infty)}(k))$ equals to the optimal performance index function $V^*(x^*(k))$, i.e.,*

$$V^{(i+1)}(x^{(i)}(k)) \to V^*(x^*(k)) \quad (49)$$

as $i \to \infty$.

Proof As

$$V^*(x^*(k)) = V^*(x^*(k)) = \min_{\underline{u}_k} V(x(k), \underline{u}_k), \quad (50)$$

we have

$$V^*(x^*(k)) \leq V^{(i+1)}(x^{(i)}(k)). \quad (51)$$

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

$$V^*(x^*(k)) \leq V^{(\infty)}(x^{(\infty)}(k)) \quad (52)$$

On the other side, according to Theorem 1 and Corollary 1, for any control sequence $\{\mu(k)\}$, we have

$$V^{(q+1)}(x^{(q)}(k)) \leq \Gamma^{(q+1)}(x^{(q)}(k)) = \sum_{j=0}^{q} \widetilde{L}(k+j) \quad (53)$$

where

$$\begin{aligned} \widetilde{L}(k+j) &= \widetilde{X}^{(q-j)T}(k+j)Q\widetilde{X}^{(q-j)}(k+j) \\ &\quad + \Pi^{(q-j)T}(k+j)R\Pi^{(q-j)}(k+j) \end{aligned} \quad (54)$$

and $\Pi^{(q-j)}(k) = \left[\mu^{(q-j)T}(k-\tau_0), \ldots, \mu^{(q-j)T}(k-\tau_n)\right]^T$. $\widetilde{X}^{(q-j)T}(k)$ is the resultant state of the control $\Pi^{(q-j)T}(k)$. Let $q \to \infty$, and we have

$$V^{(\infty)}(x^{(\infty)}(k)) \leq \Gamma^{(\infty)}(x^{(\infty)}(k)) = \sum_{j=0}^{\infty} \widetilde{L}(k+j). \quad (55)$$

As $\{\mu(k)\}$ is any control sequence, we have

$$\begin{aligned} V^{(\infty)}(x^{(\infty)}(k)) &\leq \min_{\underline{\mu}_k}\left\{\sum_{j=0}^{\infty} \widetilde{L}(k+j)\right\} \\ &= V^*(x^*(k)). \end{aligned} \quad (56)$$

Combining (52) and (56), we have

$$V^{(\infty)}(x^{(\infty)}(k)) = V^*(x^*(k)). \quad (57)$$

The proof is completed. □

3.3 Summary of the dual iterative ADP algorithm

Now, we summarize the dual iterative ADP algorithm as follows.

Step 1 Choose a random array of initial states x_0 and choose the time point $k > \max\{\sigma_m, \tau_n\}$. Set a computation precision ε.

Step 2 Let the iteration index $i = 0$ and $V^{(0)} = 0$. Implement the dual iterative ADP algorithm (9–10) at $j = 0, 1, \ldots, k$. Obtain the iterative control sequence $\{u^{(0)}(0), u^{(0)}(1), \ldots, u^{(0)}(k-1)\}$ and corresponding iterative state sequence $\{x^{(0)}(0), x^{(0)}(1), \ldots, x^{(0)}(k)\}$.

Step 3 Record the iterative performance index functions $V^{(1)}(x^{(0)}(j))$ at $j = 0, 1, \ldots, k$.

Step 4 For $i = 1, 2, \ldots$, implement the algorithm at $j = 0, 1, \ldots, k-1$. Obtain the iterative control sequence $\{u^{(i)}(0), u^{(i)}(1), \ldots, u^{(i)}(k-1)\}$ and the iterative state sequence $\{x^{(i)}(0), x^{(i)}(1), \ldots, x^{(i)}(k)\}$.

Step 5 Record the iterative performance index functions $V^{(i+1)}(x^{(i)}(j))$ at $j = 0, 1, \ldots, k$.

Step 6 If $|V^{(i+1)}(x^{(i)}(k)) - V^{(i)}(x^{(i-1)}(k))| \leq \varepsilon$, goto Step 7; else let $i = i + 1$ and goto Step 4.

Step 7 Stop.

4 Neural network implementation for the control scheme with time-delays

In this subsection, we will present the realization of the dual iterative ADP algorithm using neural networks. Assume the number of hidden layer neurons is denoted by l, the weight matrix between the input layer and hidden layer is denoted by V, the weight matrix between the hidden layer and output layer is denoted by W, then the output of three-layer neural network is represented by:

$$\hat{F}(X,V,W) = W^T\sigma(V^TX) \tag{58}$$

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

The NN estimation error can be expressed by

$$F(X) = F(X,V^*,W^*) + \varepsilon(X) \tag{59}$$

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

Here, there are two parts of neural networks, which are critic network, and action network, respectively. All the neural networks are chosen as three-layer feed-forward network. The whole structure diagram is shown in Fig. 1. The utility term in the figure denotes $l(X(k), U(k)) = X^T(k)QX(k) + U^T(k)RU(k)$.

The details of training the neural networks can be seen in [27] and omitted here.

5 Simulation study

In this section, two examples are provided to demonstrate the effectiveness of the optimal control scheme proposed in this paper.

5.1 Optimal control for nonlinear system with time-delay in state variables

For the first example, we will show that the proposed dual iterative ADP algorithm is effective for nonlinear systems with time-delay in the state variables. The system is chosen as the example in [14] with some modifications. We consider the following affine nonlinear system

$$x(k+1) = f(x(k), x(k-\sigma)) + g(x(k), x(k-\sigma))u(k) \tag{60}$$

where

$$f(x(k), x(k-\sigma)) = \begin{bmatrix} x_2(k) + \sin(x_1(k-\sigma)) \\ (1-x_1^2(k))x_2(k) - x_1(k) + x_1(k-\sigma)x_2(k-\sigma) \end{bmatrix},$$

$$g(x(k), x(k-\sigma)) = \begin{bmatrix} (1+x_1^2(k)+x_2^2(k)) & 0 \\ 0 & (1+x_1^2(k)+x_2^2(k)) \end{bmatrix}.$$

The initial conditions are the same as the example in [14]. Let the time-delay in state variables $\sigma = 2$, $x(s) = [-0.3, 1]^T$ for $s = -2, -1, 0$. The performance index function is defined as (2), where $Q = R = I$. We choose three-layer neural networks as the critic network and the action network with the structures 2-10-1 and 4-10-2, respectively. The initial weights of the critic network and the action network are both set to be random in $[-0.5, 0.5]$. We implement the algorithm at the time instant $k = 4$. The algorithm iterates for $i = 200$ times to guarantee the convergence of the algorithm. In each iteration step, the critic network and the action network are trained for 500 steps so that the given accuracy $\varepsilon = 10^{-6}$ is reached. In the training process, the learning rate $\alpha_c = \beta_a = 0.01$.

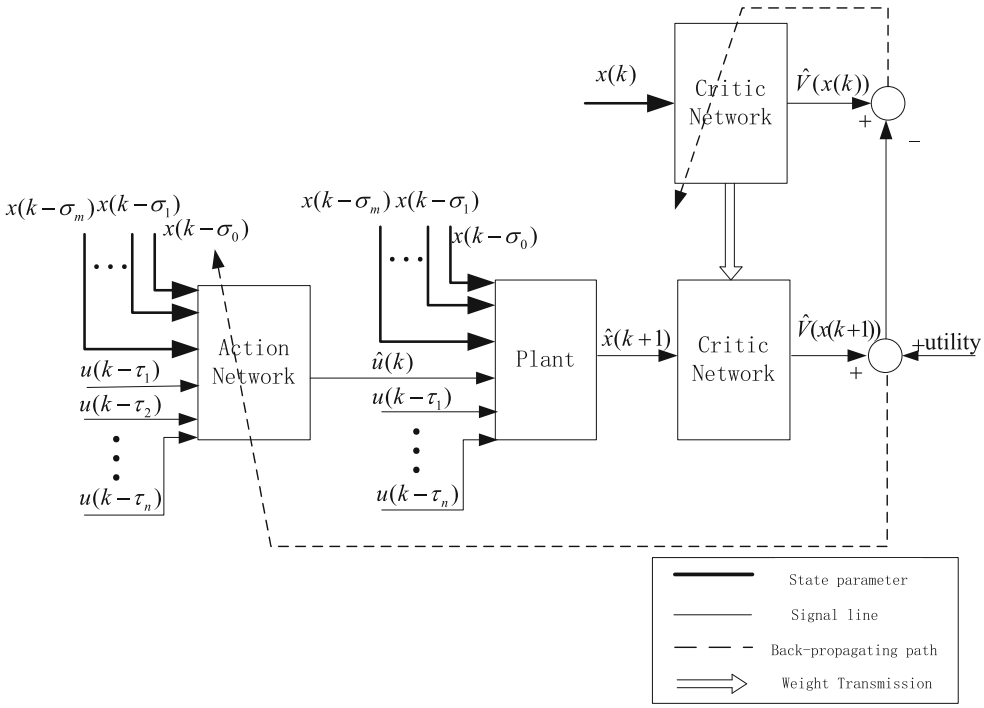

Fig. 1 The structure diagram of the algorithm

For each iteration index, we should implement the dual iterative ADP algorithm at $k = 0, 1, 2, 3$ to obtain the state sequence $[x^{(i)}(0), x^{(i)}(1), \ldots, x^{(i)}(4)]$. The trajectories of $x^{(i)}(4), i = 0, 1, \ldots, 200$ are shown in Fig. 2. The convergence curve of the performance index function at $k = 4$ is shown in Fig. 3. Then, we apply the optimal control to the system for $T_f = 30$ time steps and obtain the following results. The state trajectories are given as Fig. 4, and the corresponding control curves are given as Fig. 5.

Remark 5 From Fig. 2, we can see that for each iteration, the iterative states at $k = 4$ are different, i.e., $x^{(i)}(4) \neq x^{(j)}(4)$ for $i \neq j$. This is an obvious difference between the dual iterative ADP algorithm and the value iteration algorithm in [2]. While we can see that the performance index functions are still a nondecreasing trajectory as i increasing just like the value iteration algorithm in [2]. So, we can say that value iteration algorithm is a special case of the dual iterative ADP algorithm proposed in this paper.

5.2 Optimal control for nonlinear system with time-delays in state and control variables

In the second example, we will show that the proposed dual iterative ADP algorithm is effective for nonlinear systems with time-delays in the state and control variables. We introduce the control delays in the system of Example 1. Then, we consider the following affine nonlinear system

$$x(k+1) = f(x(k), x(k-\sigma)) + g_0(x(k), x(k-\sigma))u(k) + g_1(x(k), x(k-\sigma))u(k-\tau) \tag{61}$$

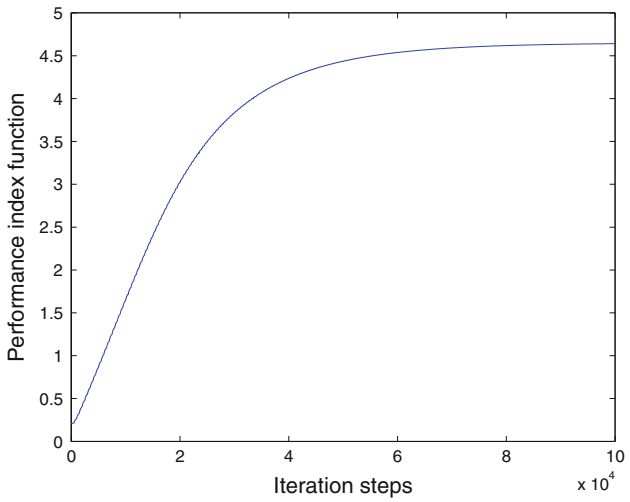

Fig. 3 The convergence of performance index function

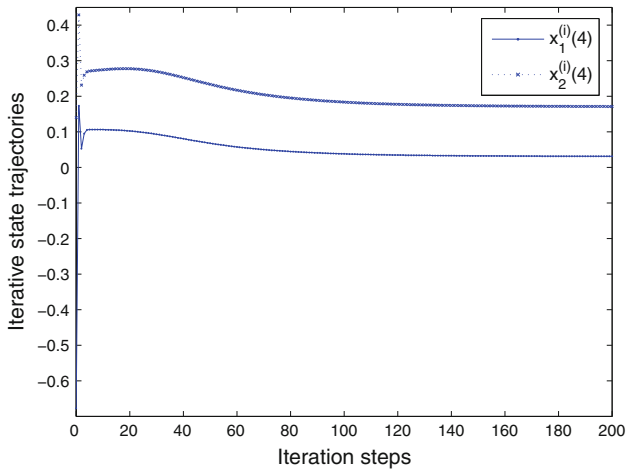

Fig. 2 The convergence of states at $k = 4$

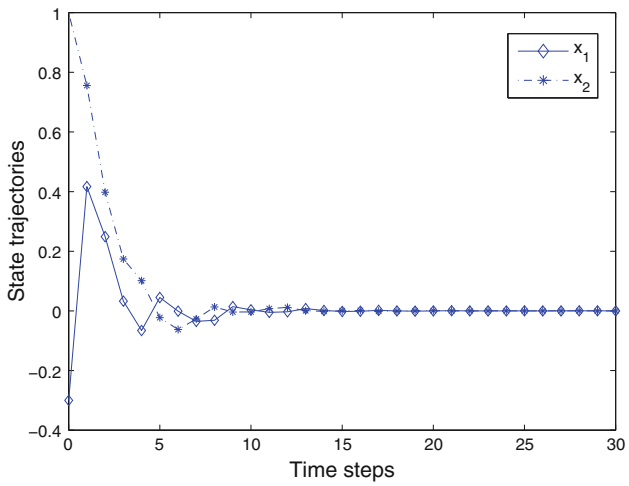

Fig. 4 The optimal state trajectories

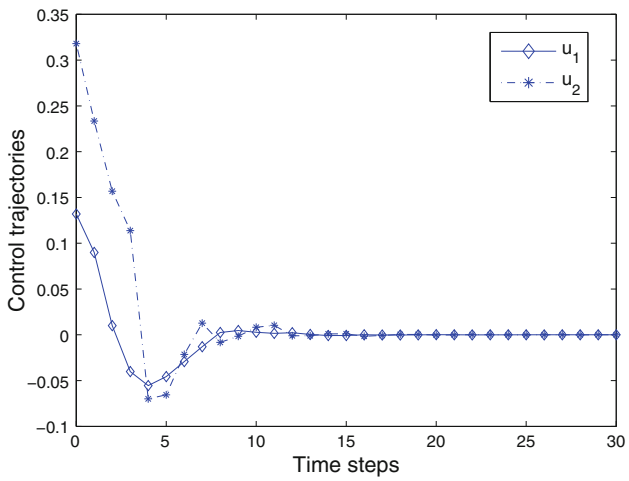

Fig. 5 The optimal control trajectories

where

$$
\begin{aligned}
&f(x(k), x(k-\sigma)) \\
&= \begin{bmatrix} x_2(k) + \sin(x_1(k-\sigma)) \\ (1 - x_1^2(k))x_2(k) - x_1(k) + x_1(k-\sigma)x_2(k-\sigma) \end{bmatrix}, \\
&g_0(x(k), x(k-\sigma)) \\
&= \begin{bmatrix} (1 + x_1^2(k) + x_2^2(k)) & 0 \\ 0 & (1 + x_1^2(k) + x_2^2(k)) \end{bmatrix}, \\
&g_1(x(k), x(k-\sigma)) \\
&= \begin{bmatrix} 0.3(1 + x_1^2(k) + x_2^2(k)) & 0 \\ 0 & 0.2(1 + x_1^2(k) + x_2^2(k)) \end{bmatrix}.
\end{aligned} \tag{62}
$$

The initial conditions are the same as the ones in Example 1. Let the time-delay of control variables $\tau = 1$, $x(s) = [-0.3, 1]^T$ for $s = -2, -1, 0$. We also choose three-layer neural networks as the critic network and the action network with the structures 2-10-1 and 6-10-2, respectively. The initial weights of the action network, critic network, and model network are all set to be random in $[-0.5, 0.5]^T$. We implement the algorithm at the time instant $k = 4$. The algorithm iterates for $i = 200$ times to guarantee the convergence of the algorithm. In each iteration step, the critic network and the action network are trained for 500 steps so that the given accuracy $\varepsilon = 10^{-6}$ is reached. In the training process, the learning rate $\alpha_c = \beta_a = 0.01$. For each iteration index, we implement the dual iterative ADP algorithm at $k = 0, 1, 2, 3$ and obtain the state sequence $[x^{(i)}(0), x^{(i)}(1), \ldots, x^{(i)}(4)]$. The trajectories of $x^{(i)}(4), i = 0, 1, \ldots, 200$ are shown in Fig. 6. The convergence curve of the performance index function at $k = 4$ is shown in Fig. 7. Then, we apply the optimal control to the system for $T_f = 50$ time steps and obtain the following results. The state trajectories are given as Fig. 8, and the corresponding control curves are given as Fig. 9.

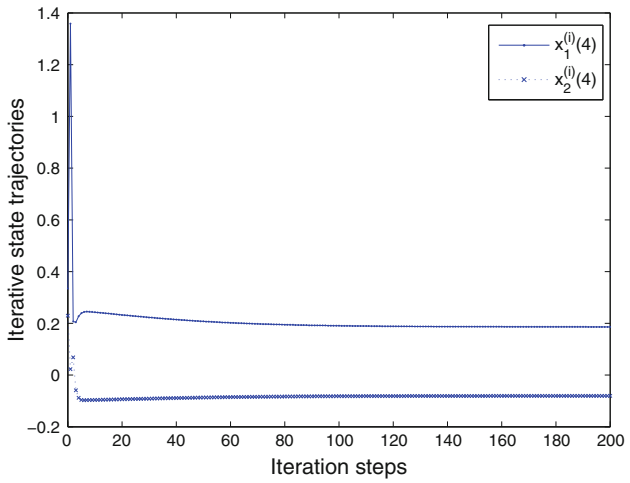

Fig. 6 The convergence of states at $k = 4$

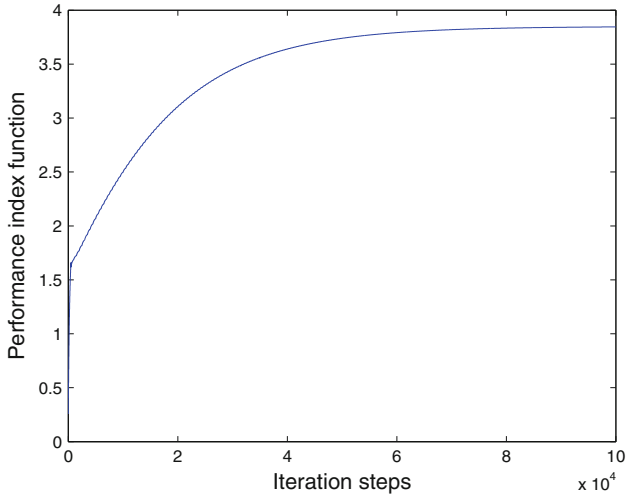

Fig. 7 The convergence of performance index function

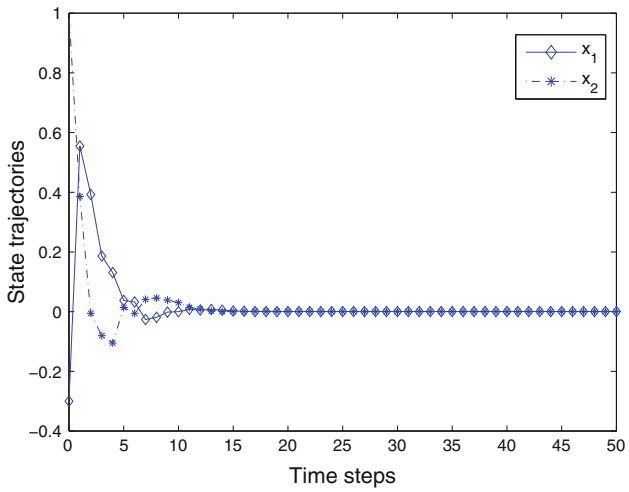

Fig. 8 The optimal state trajectories

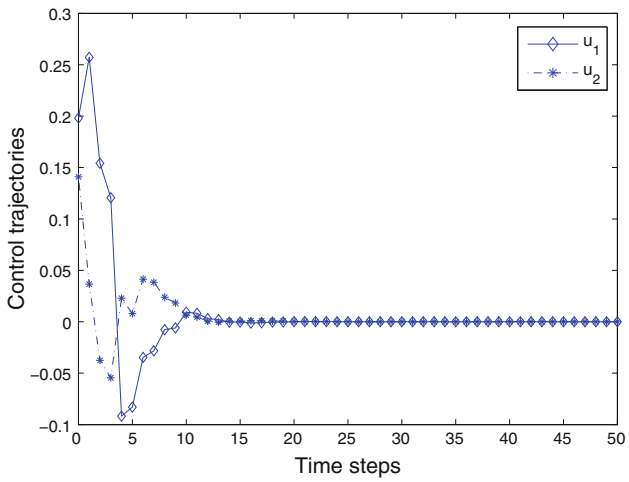

Fig. 9 The optimal control trajectories

6 Conclusion

In this paper, we propose an effective dual iterative algorithm to find the infinite-horizon optimal controller for a class of discrete-time nonlinear systems with time-delays in state and control variables using adaptive dynamic programming. Performance index functions and system state both are updated in each iteration to reach the optimal solution of the optimal problem. Convergence analysis of the performance index function for the dual iterative ADP algorithm is proved to guarantee the performance index function to reach the optimum. Neural networks are used to implement the dual iterative ADP algorithm. Finally, two simulation examples are given to illustrate the performance of the proposed algorithm.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grants 60904037, 60921061, and 61034002, in part by Beijing Natural Science Foundation under Grant 4102061, and in part by China Postdoctoral Science Foundation under Grant 201104162.

References

1. Abu-Khalaf M, Lewis FL (2005) Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. Automatica 41(5):779–791
2. Al-Tamimi A, Abu-Khalaf M, Lewis FL (2007) Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. IEEE Trans Syst Cybern Part B Cybern 37(1):240–247
3. Al-Tamimi A, Lewis FL, Abu-Khalaf M (2008) Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. IEEE Trans Syst Man Cybern Part B Cybern 38(4):943–949
4. Basin M, Rodriguez-Gonzalez J (2006) Optimal control for linear systems with multiple time delays in control input. IEEE Trans Autom Control 51(1):91–97
5. Basin M, Rodriguez-Gonzaleza J, Fridman L (2007) Optimal and robust control for linear state-delay systems. J Franklin Inst 344(7):830–845
6. Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton, NJ
7. Busoniu L, Ernst D, Schutter BD, Babuska R (2010) Approximate dynamic programming with a fuzzy parameterization. Automatica 46(5):804–814
8. Gao H, Sun W, Shi P (2010) Robust sampled-data H_∞ control for vehicle active suspension systems. IEEE Trans Control Syst Technol 18(1):238–245
9. Chen Z, Jagannathan S (2008) Generalized Hamilton-Jacobi-Bellman formulation-based neural network control of affine nonlinear discretetime systems. IEEE Trans Neural Netw 19(1):90–106
10. Chiasson J (2007) Applications of time delay systems. Springer, Berlin
11. Halpin SM, Harley KA, Jones RA, Taylor LY (2008) Slope-permissive under-voltage load shed relay for delayed voltage recovery mitigation. IEEE Trans Power Syst 23(3):1211–1216
12. Han M, Han B, Xi J, Hirasawa K (2006) Universal learning network and its application for nonlinear system with long time delay. Comput Chem Eng 31(1):13–20
13. Hanselmann T, Noakes L, Zaknich A (2007) Continuous-time adaptive critics. IEEE Trans Neural Netw 18(3):631–647
14. Ho DWC, Li J, Niu Y (2005) Adaptive neural control for a class of nonlinearly parametric time-delay systems. IEEE Trans Neural Netw 16(3):625–635
15. Huang X, Ma M (2008) Optimal scheduling for minimum delay in passive star coupled WDM optical networks. IEEE Trans Commun 56(8):1324–1330
16. Lewis FL, Vrabie D (2009) Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits Syst Mag 9(3):32–50
17. Li T, Tong SC, Feng G (2010) A novel robust adaptive-fuzzy-tracking control for a class of nonlinear multi-input/multi-output systems. IEEE Trans Fuzzy Syst 18(1):150–160
18. Li T, Wang D, Feng G, Tong SC (2010) A DSC approach to robust adaptive NN tracking control for strict-feedback nonlinear systems. IEEE Trans Syst Man Cybern Part B Cybern 40(3):915–927
19. Li T, Feng , Wang D, Tong S (2010) Neural-network-based simple adaptive control of uncertain multi-input multi-output non-linear systems. IET Control Theory Appl 4(9):1543–1557
20. Liu D, Zhang Y, Zhang H (2005) A self-learning call admission control scheme for CDMA cellular networks. IEEE Trans Neural Netw 16(5):1219–1228
21. Malek-Zavarei M, Jashmidi M (1987) Time-delay systems: analysis, optimization and applications. North-Holland, Amsterdam
22. Pindyck RS (1992) The distrete-time tracking problem with a time delay in the control. IEEE Trans Autom Control 17(6):397–398
23. Murray JJ, Cox CJ, Lendaris GG, Saeks R (2002) Adaptive dynamic programming. IEEE Trans Syst Man Cybern Part C Appl Rev 32(2):140–153
24. Prokhorov DV, Wunsch DC (1997) Adaptive critic designs. IEEE Trans Neural Netw 8(5):997–1007
25. Richard JP (2003) Time-delay systems: an overview of some recent advances and open problems. Automatica 39(10):1667–1694
26. Schenato L (2008) Optimal estimation in networked control systems subject to random delay and packet drop. IEEE Trans Autom Control 53(5):1311–1317
27. Si J, Wang YT (2001) On-line learning control by association and reinforcement. IEEE Trans Neural Netw 12(2):264–276
28. Silva GJ (2005) PID Controllers for time-delay systems. Birkhuser, Boston, MA
29. Song R, Zhang H, Luo Y, Wei Q (2010) Optimal control laws for time-delay systems with saturating actuators based on heuristic dynamic programming. Neurocomputing 73(16–18):3020–3027
30. Sun Q, Li Z, Yang J, Luo Y (2010) Load distribution model and voltage static profile of Smart Grid. J Central S Univ Technol 17(4):824–829
31. Vamvoudakis KG, Lewis FL (2010) Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. Automatica 46(5):878–888
32. Wang D, Liu D, Wei Q (2012) Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach. Neurocomputing 78(1):14–22
33. Wang FY, Jin N, Liu D, Wei Q (2011) Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ϵ-error bound. IEEE Trans Neural Netw 22(1):24–36
34. Wang FY, Zhang H, Liu D (2009) Adaptive dynamic programming: an introduction. IEEE Comput Intell Mag 4(2):39–47
35. Watkins C (1989) Learning from delayed rewards. Ph.D. Thesis. Cambridge University, Cambridge

36. Wei Q, Zhang H, Dai J (2009) Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions. Neurocomputing 72(7–9):1839–1848
37. Werbos PJ (1991) A menu of designs for reinforcement learning over time. In: Miller WT, Sutton RS, Werbos PJ (eds) Neural networks for control. MIT Press, Cambridge, pp 67–95
38. Werbos PJ (1992) Approximate dynamic programming for real-time control and neural modeling. In: White DA, Sofge DA (eds) Handbook of intelligent control: neural, fuzzy, and adaptive approaches ch. 13.. Van Nostrand Reinhold, New York
39. Widrow B, Gupta N, Maitra S (1973) Punish/reward: learning with a critic in adaptive threshold systems. IEEE Trans Syst Man Cybern 3:455–465
40. Yadav V, Padhi R, Balakrishnan SN (2007) Robust/optimal temperature profile control of a high-speed aerospace vehicle using neural networks. IEEE Trans Neural Netw 18(4):1115–1128
41. Yang Y, Feng G, Ren J (2004) A combined backstepping and small-gain approach to robust adaptive fuzzy control for strict-feedback nonlinear systems. IEEE Trans Syst Man Cybern Part A Syst Humans 34(3):406–420
42. Zhang H, Basin MV, Skliar M (2007) Itô-Volterra optimal state estimation with continuous, multirate, randomly sampled, and delayed measurements. IEEE Trans Autom Control 52(3):401–416
43. Zhang H, Quan Y (2001) Modeling, identification and control of a class of nonlinear system. IEEE Trans Fuzzy Syst 9(2):349–354
44. Zhang H, Wang Y, Liu D (2008) Delay-dependent guaranteed cost control for uncertain stochastic fuzzy systems with multiple time delays. IEEE Trans Syst Man Cybern Part B Cybern 38(1):125–140
45. Zhang H, Wei Q, Luo Y (2008) 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 38(4):937–942
46. Zhang H, Wei Q, Liu D (2011) An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. Automatica 47(1):207–214
47. Zhang H, Song R, Wei Q, Zhang T (2011) Optimal tracking control for a class of nonlinear discrete-time systems with time delays based on heuristic dynamic programming. IEEE Trans Neural Netw 22(12):1851–1862
48. Zhang H, Yang D, Chai T (2007) Guaranteed cost networked control for T-S fuzzy systems with time delay. IEEE Trans Syst Man Cybern Part C Appl Rev 37(2):160–172