Neurocomputing 156 (2015) 166–175

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Nearly finite-horizon optimal control for a class of nonaffine time-delay nonlinear systems based on adaptive dynamic programming

Ruizhuo Song [a], Qinglai Wei [b],*, Qiuye Sun [c]

[a] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
[b] The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[c] School of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110004, China

ARTICLE INFO

Article history:
Received 15 September 2014
Received in revised form
5 December 2014
Accepted 24 December 2014
Communicated by H. Zhang
Available online 5 January 2015

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

ABSTRACT

In this paper, a novel adaptive dynamic programming (ADP) algorithm is developed to solve the nearly optimal finite-horizon control problem for a class of deterministic nonaffine nonlinear time-delay systems. The idea is to use ADP technique to obtain the nearly optimal control which makes the optimal performance index function close to the greatest lower bound of all performance index functions within finite time. The proposed algorithm contains two cases with respective different initial iterations. In the first case, there exists control policy which makes arbitrary state of the system reach to zero in one time step. In the second case, there exists a control sequence which makes the system reach to zero in multiple time steps. The state updating is used to determine the optimal state. Convergence analysis of the performance index function is given. Furthermore, the relationship between the iteration steps and the length of the control sequence is presented. Two neural networks are used to approximate the performance index function and compute the optimal control policy for facilitating the implementation of ADP iteration algorithm. At last, two examples are used to demonstrate the effectiveness of the proposed ADP iteration algorithm.

1. Introduction

Time-delay phenomenons are often encountered in physical and biological systems, and require special attention in engineering applications [1]. Transportation systems, communication systems, chemical processing systems, metallurgical processing systems and power systems are examples of time-delay systems. Delays may result in degradation in the control efficiency even instability of the control systems [2]. So there have been many works about systems with time delays in various research areas such as electrical, chemical engineering and networked control [3]. In the past few decades, the stabilization and the control of time-delay systems have always been the key focus in the control field [4–7]. Furthermore, there are many researchers who studied the controllability of linear time-delay systems [8–10]. They proposed some related theorems to judge the controllability of the linear time-delay systems. In addition, the optimal control problem is often encountered in industrial production. So the investigation of the optimal control for time-delay systems is significant. In [11] Chyung has pointed out the disadvantages of discrete time-delay system written as an extended system by increasing the dimension method to deal with the optimal control problem. So some direct methods for linear time-delay systems were presented in [11,12]. While for nonlinear time-delay system, due to the complexity of systems, the optimal control problem is rarely researched. This motivated our research interest.

As is well known, dynamic programming is very useful in solving the optimal control problems [13–15]. But it is often computationally untenable to run dynamic programming [16]. In the early 1970s, Werbos set up the basic strategy for ADP [17] to overcome the "curse of dimensionality" of dynamic programming. In [18], Werbos classified ADP approaches into four main schemes: heuristic dynamic programming (HDP), dual heuristic dynamic programming (DHP), action dependent heuristic dynamic programming (ADHDP), and action dependent dual heuristic dynamic programming (ADDHP). In recent years, ADP algorithms have made great progress [19–24]. In [25], an iteration ADP scheme with convergence proof was proposed for solving the optimal control problem of nonlinear discrete-time systems. In [26], an optimal tracking controller was proposed for a class of nonlinear discrete-time systems with time delays based on a novel HDP algorithm. In [27], a ADP-based optimal control is developed for complex-valued systems. Note that most of the results of the present study are about the infinite-horizon optimal control. The system cannot be really stabilized or tracked until the time

[*] Corresponding author.

http://dx.doi.org/10.1016/j.neucom.2014.12.066
0925-2312/© 2014 Elsevier B.V. All rights reserved.

reaches infinity. While for finite-horizon control problems, the system must be stabilized to zero or tracked to a desired trajectory within finite time. The controller design of finite-horizon problems still presents a challenge to control engineers as the lack of methodology and the control step is difficult to determine. Few results relate to the finite-horizon optimal control based on ADP algorithm. As we know that [28] solved the finite-horizon optimal control problem for a class of discrete-time nonlinear systems using ADP algorithm. But the method in [28] cannot be used in nonlinear time-delay systems. As the delay states in time-delay systems are coupling with each other. The state of current time k is decided by the states before k and the control law, while the control law is not known before it is obtained. So based on the research results in [28], we proposed a new ADP algorithm to solve the nearly finite-horizon optimal control problem for discrete time-delay systems through the framework of Hamilton–Jacobi–Bellman (HJB) equation.

In this paper the optimal controller is designed based on the original time-delay systems, directly. The state updating method is proposed to determine the optimal state of the time-delay system. For finite-horizon optimal control, the system can reach to zero when the final running step N is finite. But it is impossible in practice. So the results in this paper are in the sense of an error bound. The main contributions of this paper can be summarized as follows:

1. The finite-horizon optimal control for deterministic discrete nonaffine time-delay systems is studied based on the ADP algorithm.
2. The state updating is used to determine the optimal states of HJB equation.
3. The relationship between the iteration steps and the length of the control sequence is given.

This paper is organized as follows. In Section 2, the problem formulation is presented. In Section 3, the nearly finite-horizon optimal control scheme is developed based on the iteration ADP algorithm and the convergence proof is given. In Section 4, two examples are given to demonstrate the effectiveness of the proposed control scheme. In Section 5, the conclusion is drawn.

2. Problem statement

Consider a class of deterministic nonaffine time-delay nonlinear systems

$$\begin{aligned}&x(t+1)=F(x(t),x(t-h_1),x(t-h_2),\ldots,x(t-h_l),u(t)),\\&x(t)=\chi(t),\ \ -h_l\le t\le 0\end{aligned}\tag{1}$$

where $x(t)\in\mathfrak{R}^n$ is the state and $x(t-h_1),\ldots,x(t-h_l)\in\mathfrak{R}^n$ are time-delay states. $u(t)\in\mathfrak{R}^m$ is the system input. $\chi(t)$ is the initial state, h_i, $i=1,2,\ldots,l$, is the time delay, set $0<h_1<h_2<\ldots<h_l$, and they are nonnegative integer numbers. $F(x(t),x(t-h_1),x(t-h_2),\ldots,x(t-h_l),u(t))$ is the known function. $F(0,0,\ldots,0)=0$.

For any time step k, the performance index function for state $x(k)$ under the control sequence $U(k,N+k-1)=(u(k),u(k+1),\ldots,u(N+k-1))$ is defined as

$$J(x(k),U(k,N+k-1))=\sum_{j=k}^{N+k-1}\{x^T(j)Qx(j)+u^T(j)Ru(j)\},\tag{2}$$

where Q and R are positive definite constant matrixes.

In this paper, we focus on solving the nearly finite-horizon optimal control problem for system (1). The feedback control $u(k)$ must not only stabilize the system within finite time steps but also guarantee the performance index function (2) to be finite. So the control sequence $U(k,N+k-1)=(u(k),u(k+1),\ldots,u(N+k-1))$ must be admissible.

Definition 1. N time steps control sequence: for any time step k, we define the N time steps control sequence $U(k,N+k-1)=(u(k),u(k+1),\ldots,u(N+k-1))$. The length of $U(k,N+k-1)$ is N.

Definition 2. Final state: we define final state $x_f=x_f(x(k),U(k,N+k-1))$, i.e., $x_f=x(N+k)$.

Definition 3. Admissible control sequence: an N time steps control sequence is said to be admissible for $x(k)$, if the final state $x_f(x(k),U(k,N+k-1))=0$ and $J(x(k),U(k,N+k-1))$ is finite.

Remark 1. Definitions 1 and 2 are used to state conveniently the admissible control sequence, i.e. Definition 3, which is necessary for the theorems of this paper.

Remark 2. It is important to point out that the length of control sequence N cannot be designated in advance. It is calculated by the proposed algorithm. If we calculate that the length of optimal control sequence is L at time step k, then we consider that the optimal control sequence length at time step k is $N=L$.

According to the theory of dynamics programming [29], the optimal performance index function is defined as

$$J^*(x(k))=\inf_{U(k,N+k-1)}J(x(k),U(k,N+k-1))\tag{3}$$

$$J^*(x(k))=\inf_{u(k)}\{x^T(k)Qx(k)+u^T(k)Ru(k)+J^*(x(k+1))\},\tag{4}$$

and the optimal control policy is

$$u^*(k)=\arg\inf_{u(k)}\{x^T(k)Qx(k)+u^T(k)Ru(k)+J^*(x(k+1))\},\tag{5}$$

so the state under the optimal control policy is

$$x^*(t+1)=F(x^*(t),x^*(t-h_1),\ldots,x^*(t-h_l),u^*(t)),\quad t=0,1,\ldots,k,\ldots,\tag{6}$$

and then, the HJB equation is written as

$$\begin{aligned}J^*(x^*(k))&=J(x^*(k),U^*(k,N+k-1))\\&=(x^*(k))^TQx^*(k)+(u^*(k))^TRu^*(k)+J^*(x^*(k+1)).\end{aligned}\tag{7}$$

Remark 3. From Remark 2, we can see that the length N of the optimal control sequence is unknown finite number and cannot be designated in advance. So we can say that if at time step k, the length of the optimal control sequence is N, then at time step $k+1$, the length of the optimal control sequence is $N-1$. Therefore, the HJB equation (7) is established.

In the following, we will give an explanation about the validity of Eq. (3). First, we define $U^*(k,N+k-1)=(u^*(k),u^*(k+1),\ldots,u^*(N+k-1))$, i.e.

$$U^*(k,N+k-1)=\arg\inf_{U(k,N+k-1)}J(x(k),U(k,N+k-1)).\tag{8}$$

Then we have

$$\begin{aligned}J^*(x(k))&=\inf_{U(k,N+k-1)}J(x(k),U(k,N+k-1))\\&=J(x(k),U^*(k,N+k-1)).\end{aligned}\tag{9}$$

Then according to (2), we can get

$$\begin{aligned}J^*(x(k))&=\sum_{j=k}^{N+k-1}\{x^T(j)Qx(j)+(u^*(j))^TRu^*(j)\}\\&=x^T(k)Qx(k)+(u^*(k))^TRu^*(k)+\cdots+x^T(N+k-1)Qx(N+k-1)\\&\quad+(u^*(N+k-1))^TRu^*(N+k-1).\end{aligned}\tag{10}$$

Eq. (10) can be written as

$$\begin{aligned}J^*(x(k))&=x^T(k)Qx(k)+(u^*(k))^TRu^*(k)+\cdots+x^T(N+k-2)Qx(N+k-2)\\&\quad+(u^*(N+k-2))^TRu^*(N+k-2)\\&\quad+\inf_{u(N+k-1)}\{x^T(N+k-1)Qx(N+k-1)\\&\quad+u^T(N+k-1)Ru(N+k-1)\}.\end{aligned}\tag{11}$$

We also obtain

$$\begin{aligned}J^*(x(k)) = {} & x^T(k)Qx(k)+(u^*(k))^T Ru^*(k)\\ & +\cdots+\inf_{u(N+k-2)}\{x^T(N+k-2)Qx(N+k-2)\\ & +u^T(N+k-2)Ru(N+k-2)\\ & +\inf_{u(N+k-1)}\{x^T(N+k-1)Qx(N+k-1)\\ & +u^T(N+k-1)Ru(N+k-1)\}\}. \end{aligned} \tag{12}$$

So we have

$$\begin{aligned}J^*(x(k)) = {} & \inf_{u(k)}\{x^T(k)Qx(k)+u^T(k)Ru(k)\\ & +\cdots+\inf_{u(N+k-2)}\{x^T(N+k-2)Qx(N+k-2)\\ & +u^T(N+k-2)Ru(N+k-2)\\ & +\inf_{u(N+k-1)}\{x^T(N+k-1)Qx(N+k-1)\\ & +u^T(N+k-1)Ru(N+k-1)\}\}\cdots\}. \end{aligned} \tag{13}$$

Thus according to (9), Eq. (10) is expressed as

$$\begin{aligned}J^*(x(k)) & = \inf_{u(k)}\Big\{x^T(k)Qx(k)+u^T(k)Ru(k)\\ & \quad +\inf_{U(k+1,N+k-1)} J(x(k+1),U(k+1,N+k-1))\Big\}\\ & = \inf_{u(k)}\Big\{x^T(k)Qx(k)+u^T(k)Ru(k)\\ & \quad +J^*(x(k+1))\Big\}. \end{aligned} \tag{14}$$

Therefore, Eqs. (3) and (4) are established.

In the following part we will give a novel iteration ADP algorithm to get the nearly optimal solution.

3. The iteration ADP algorithm and its convergence

3.1. The novel ADP iteration algorithm

In this subsection we will give the novel iteration ADP algorithm in detail. For the state $x(k)$ of system (1), there exists two cases. Case 1: $\exists U(k,k)$ which makes $x(k+1)=0$. Case 2: $\exists U(k,k+m)$, $m>0$, which makes $x(k+m+1)=0$. In the following part, we will discuss the two cases, respectively.

Case 1: There exists $U(k,k)=(\beta(k))$ which makes $x(k+1)=0$ for system (1). We set the optimal control sequence $U^*(k+1,k+1)=(0)$. The states of the system are driven by a given initial state $\chi(t)$, $-h_l \le t \le 0$ and the initial control policy $\beta(t)$. We set $V^{[0]}(x(k+1)) = J(x(k+1),U^*(k+1,k+1))=0$, $\forall x(k+1)$, then for time step k, we have the following iterations:

$$\begin{aligned}& u^{[1]}(k) = \arg\inf_{u(k)}\{x^T(k)Qx(k)+u^T(k)Ru(k)+V^{[0]}(x(k+1))\},\\ & \text{s.t.}\quad F(x(k),x(k-h_1),x(k-h_2),\ldots,x(k-h_l),u(k))=0\end{aligned} \tag{15}$$

and

$$V^{[1]}(x^{[1]}(k)) = (x^{[1]}(k))^T Qx^{[1]}(k)+(u^{[1]}(k))^T Ru^{[1]}(k)+V^{[0]}(x^{[0]}(k+1)), \tag{16}$$

where the states in (16) are obtained as follows:

$$\begin{aligned}& x^{[1]}(t+1) = F(x^{[1]}(t),x^{[1]}(t-h_1),x^{[1]}(t-h_2),\ldots,x^{[1]}(t-h_l),u^{[1]}(t)),\\ & \qquad t=0,1,\ldots,k-1\end{aligned} \tag{17}$$

and

$$\begin{aligned}& x^{[0]}(t+1) = F(x^{[1]}(t),x^{[1]}(t-h_1),x^{[1]}(t-h_2),\ldots,x^{[1]}(t-h_l),u^{[1]}(t)),\\ & \qquad t=k,k+1,\cdots\end{aligned} \tag{18}$$

For the iteration step i = 1, 2, …we have the iterations as follows:

$$u^{[i+1]}(k) = \arg\inf_{u(k)}\{x^T(k)Qx(k)+u^T(k)Ru(k)+V^{[i]}(x(k+1))\}, \tag{19}$$

and

$$\begin{aligned}V^{[i+1]}(x^{[i+1]}(k)) = {} & (x^{[i+1]}(k))^T Qx^{[i+1]}(k)+(u^{[i+1]}(k))^T Ru^{[i+1]}(k)\\ & +V^{[i]}(x^{[i]}(k+1)),\end{aligned} \tag{20}$$

where $V^{[i]}(x(k+1))$ in (19) is obtained as follows:

$$\begin{aligned}V^{[i]}(x(k+1)) = {} & \arg\inf_{u(k+1)}\{x^T(k+1)Qx(k+1)+u^T(k+1)Ru(k+1)\\ & +V^{[i-1]}(x(k+2))\},\end{aligned} \tag{21}$$

and the states in (20) are updated as follows:

$$\begin{aligned}& x^{[i+1]}(t+1) = F(x^{[i+1]}(t),x^{[i+1]}(t-h_1),x^{[i+1]}(t-h_2),\ldots,\\ & x^{[i+1]}(t-h_l),u^{[i+1]}(t)),\quad t=0,1,\ldots,k-1\end{aligned} \tag{22}$$

and

$$\begin{aligned}& x^{[i]}(t+1) = F(x^{[i+1]}(t),x^{[i+1]}(t-h_1),x^{[i+1]}(t-h_2),\ldots,\\ & \quad x^{[i+1]}(t-h_l),u^{[i+1]}(t)),\quad t=k,k+1,\cdots\end{aligned} \tag{23}$$

Case 2: There exists finite-horizon admissible control sequence $U(k,k+m)=(\beta(k),\ldots,\beta(k+m))$ which makes $x_f(x(k),U(k,k+m))=0$. We suppose that for $x(k+1)$, there exists optimal control sequence $U^*(k+1,k+j-1)=(u^*(k+1),u^*(k+2),\ldots,u^*(k+j-1))$. For time step k, the iteration ADP algorithm between

$$u^{[1]}(k) = \arg\inf_{u(k)}\{x^T(k)Qx(k)+u^T(k)Ru(k)+V^{[0]}(x(k+1))\}, \tag{24}$$

and

$$V^{[1]}(x^{[1]}(k)) = (x^{[1]}(k))^T Qx^{[1]}(k)+(u^{[1]}(k))^T Ru^{[1]}(k)+V^{[0]}(x^{[0]}(k+1)), \tag{25}$$

where $\forall x(k+1)$, $V^{[0]}(x(k+1))$ in (24) is obtained as

$$\begin{aligned}V^{[0]}(x(k+1)) & = J(x(k+1),U^*(k+1,k+j-1)\\ & = J^*(x(k+1)).\end{aligned} \tag{26}$$

In (25), $V^{[0]}(x^{[0]}(k+1))$ is obtained by the similar equation (26). The states in (25) are obtained as

$$\begin{aligned}& x^{[1]}(t+1) = F(x^{[1]}(t),x^{[1]}(t-h_1),x^{[1]}(t-h_2),\ldots,\\ & x^{[1]}(t-h_l),u^{[1]}(t)),\quad t=0,1,\ldots,k-1,\end{aligned} \tag{27}$$

and

$$\begin{aligned}& x^{[0]}(t+1) = F(x^{[1]}(t),x^{[1]}(t-h_1),x^{[1]}(t-h_2),\ldots,\\ & \quad x^{[1]}(t-h_l),u^{[1]}(t)),\quad t=k,k+1,\cdots\end{aligned} \tag{28}$$

For the iteration step $i=1,2,\ldots$, the iteration algorithm will be implemented as follows:

$$u^{[i+1]}(k) = \arg\inf_{u(k)}\{x^T(k)Qx(k)+u^T(k)Ru(k)+V^{[i]}(x(k+1))\}, \tag{29}$$

and

$$\begin{aligned}V^{[i+1]}(x^{[i+1]}(k)) = {} & (x^{[i+1]}(k))^T Qx^{[i+1]}(k)+(u^{[i+1]}(k))^T Ru^{[i+1]}(k)\\ & +V^{[i]}(x^{[i]}(k+1)),\end{aligned} \tag{30}$$

where $V^{[i]}(x(k+1))$ in (29) is updated as

$$\begin{aligned}V^{[i]}(x(k+1)) = {} & \inf_{u(k+1)}\{x^T(k+1)Qx(k+1)+u^T(k+1)Ru(k+1)\\ & +V^{[i-1]}(x(k+2))\},\end{aligned} \tag{31}$$

and the states in (30) are obtained as

$$\begin{aligned}& x^{[i+1]}(t+1) = F(x^{[i+1]}(t),x^{[i+1]}(t-h_1),x^{[i+1]}(t-h_2),\ldots,\\ & x^{[i+1]}(t-h_l),u^{[i+1]}(t)),\quad t=0,1,\ldots,k-1\end{aligned} \tag{32}$$

and

$$\begin{aligned}& x^{[i]}(t+1) = F(x^{[i+1]}(t),x^{[i+1]}(t-h_1),x^{[i+1]}(t-h_2),\ldots,\\ & x^{[i+1]}(t-h_l),u^{[i+1]}(t)),\quad t=k,k+1,\ldots\end{aligned} \tag{33}$$

This completes the iteration algorithm. From the two cases we can see that, if $V^{[0]}=0$ in (25), then Case 1 is a special one of Case 2. In the following, the algorithms are summarized as follows.

Remark 4. For the state $x(k)$ of system (1), which is driven by the fixed initial states $\chi(t), -h_l \le t \le 0$. If there exists a control sequence $U(k,k)=(\beta(k))$, which makes $x(k+1)=0$ hold, then we will use Case 1 of the algorithm to obtain the optimal control. Otherwise, i.e., there does not exist $U(k,k)$, which makes $x(k+1)=0$ hold. But there is a control sequence $U(k,k+m)=(\beta(k),\ldots,\beta(k+m))$ which makes $x_f(x(k),U(k,k+m))=0$, then we will adopt Case 2 of the algorithm. The detailed implementation process of the second algorithm is as follows.

For system (1), there exists arbitrary finite-horizon admissible control sequence $U(k,k+m)=(\beta(k),\ldots,\beta(k+m))$ and the corresponding state sequence $(x(k+1),\ldots,x(k+m),x(k+m+1))$ in which $x(k+m+1)=0$. It is clearly that $U(k,k+m)$ may not be optimal one. Which means two points: (1) the length $m+1$ of control sequence $U(k,k+m)$ may not be optimal. (2) The law of control sequence $U(k,k+m)$ may not be optimal. So it is necessary to use the proposed algorithm to obtain the optimal one.

We start to discuss the proposed algorithm from the state $x(k+m)$ now. Obviously, the situation of $x(k+m)$ is belongs to Case 1, so the optimal control for $x(k+m)$ can be obtained by Case 1 of the proposed algorithm. Although the state $x(k+m)$ can reach to zero in one step, the optimal control step number may be more than one, this property can be seen in Corollary 1. Next, we can obtain the optimal control for $x(k+m-1)$ according to Case 2 of the proposed algorithm. Continue this process, until the optimal control of state $x(k)$ is obtained. From [28] we known that if the optimal control length of state $x(k+m_1+1)$ is the same as the one of $x(k+m_1)$, then we say that the two states $x(k+m_1)$ and $x(k+m_1+1)$ are in the same circular region. The finite-horizon optimal control for the two states are same. The detailed analysis can be seen in [28].

Remark 5. The proposed algorithm is novel and different from the algorithms in [28,30–32].

(a) Reference [28] is about the nonlinear systems without delays. While this paper considers the time-delay systems. So the algorithm in [28] cannot be used to deal with optimal control problem of time-delay systems.
(b) To overcome the difficulty of time delay, the state updating is used to determine the optimal states of time-delay system. So in this paper, besides the performance index function iteration and control iteration, the state updating is necessary, which is the lacking one in reference [28,30,32].
(c) The algorithm in [31] is about the multi-objective optimal control, which considers the infinite-horizon situation. While in this paper, the finite-horizon optimal control method is presented.

Algorithm 1. ADP algorithm.

Initialization:

Compute $u^{[1]}(k)$ and $V^{[1]}(x^{[1]}(k))$ by (15) and (16) in Case 1, or by (24) and (25) in Case 2;

Update:

Update $u^{[i+1]}(k)$ and $V^{[i+1]}(x^{[i+1]}(k))$ by (19) and (20) in Case 1, or by (29) and (30) in Case 2.

3.2. Convergence analysis of the improved iteration algorithm

In the above subsection, the novel algorithm for finite-horizon time-delay nonlinear systems has been proposed in detail. In the following part, we will prove that the algorithm is convergent and the limitation of the sequence of performance index functions $V^{[i+1]}(x^{[i+1]}(k))$ satisfies the HJB equation (7).

Theorem 1. *For system (1), the states of the system are driven by a given initial state $\chi(t), -h_l \le t \le 0$, and the initial finite-horizon admissible control policy $\beta(t)$. The iteration algorithm is as in (15)–(33). For time step k, $\forall x(k)$ and $U(k,k+i)$, we define*

$$\begin{aligned}\Lambda^{[i+1]}(x(k),U(k,k+i)) = {} & x^T(k)Qx(k)+u^T(k)Ru(k)\\ & +x^T(k+1)Qx(k+1)+u^T(k+1)Ru(k+1)\\ & +\cdots+x^T(k+i)Qx(k+i)+u^T(k+i)Ru(k+i)\\ & +V^{[0]}(x(k+i+1)),\end{aligned} \tag{34}$$

where $V^{[0]}(x(k+i+1))$ as in (26) and $V^{[i+1]}(x(k))$ is updated as (31). Then we have

$$V^{[i+1]}(x(k)) = \inf_{U(k,k+i)} \Lambda^{[i+1]}(x(k),U(k,k+i)). \tag{35}$$

Proof. From (31) we have

$$\begin{aligned}V^{[i+1]}(x(k)) = {} & \inf_{u(k)}\{x^T(k)Qx(k)+u^T(k)Ru(k)\\ & +\inf_{u(k+1)}\{x^T(k+1)Qx(k+1)+u^T(k+1)Ru(k+1)\\ & +\cdots+\inf_{u(k+i)}\{x^T(k+i)Qx(k+i)+u^T(k+i)Ru(k+i)\}\\ & +V^{[0]}(x(k+i+1))\}\cdots\}\}.\end{aligned} \tag{36}$$

So we can further obtain

$$\begin{aligned}V^{[i+1]}(x(k)) = {} & \inf_{U(k,k+i)}\{x^T(k)Qx(k)+u^T(k)Ru(k)\\ & +x^T(k+1)Qx(k+1)+u^T(k+1)Ru(k+1)\\ & +\cdots+x^T(k+i)Qx(k+i)+u^T(k+i)Ru(k+i)\\ & +V^{[0]}(x(k+i+1))\},\end{aligned} \tag{37}$$

Thus we can have

$$V^{[i+1]}(x(k)) = \inf_{U(k,k+i)} \Lambda^{[i+1]}(x(k),U(k,k+i)).\square \tag{38}$$

Based on Theorem 1, we give the monotonicity theorem about the sequence of performance index functions $V^{[i+1]}(x^{[i+1]}(k))$, $\forall x^{[i+1]}(k)$.

Theorem 2. *For system (1), let the iteration algorithm be as in (15)–(33). Then we have $V^{[i+1]}(x^{[i]}(k)) \le V^{[i]}(x^{[i]}(k))$, $\forall i>0$, for Case 1; $V^{[i+1]}(x^{[i]}(k)) \le V^{[i]}(x^{[i]}(k))$, $\forall i \ge 0$, for Case 2.*

Proof. We first give the proof for Case 2. Define $\hat{U}(k,k+i)=(u^{[i]}(k),\ldots,u^{[1]}(k+i-1),u^*(k+i))$, then according to the definition of $\Lambda^{[i+1]}(x(k),\hat{U}(k,k+i))$ in (34), we have

$$\begin{aligned}\Lambda^{[i+1]}(x(k),\hat{U}(k,k+i)) = {} & x^T(k)Qx(k)+(u^{[i]}(k))^TRu^{[i]}(k)\\ & +\cdots+x^T(k+i-1)Qx(k+i-1)\\ & +(u^{[1]}(k+i-1))^TRu^{[1]}(k+i-1)\\ & +x^T(k+i)Qx(k+i)+(u^*(k+i))^TRu^*(k+i)\\ & +V^{[0]}(x(k+i+1)).\square\end{aligned} \tag{39}$$

From (26) and (4), we get

$$\begin{aligned}V^{[0]}(x(k+i)) &= J^*(x(k+i))\\ &= x^T(k+i)Qx(k+i)+(u^*(k+i))^TRu^*(k+i)\\ &\quad +J^*(x(k+i+1))\\ &= x^T(k+i)Qx(k+i)+(u^*(k+i))^TRu^*(k+i)\\ &\quad +V^{[0]}(x(k+i+1)).\end{aligned} \tag{40}$$

On the other side, from (31), we haveget

$$V^{[i]}(x(k)) = x^T(k)Qx(k) + (u^{[i]}(k))^T Ru^{[i]}(k) \\ + \cdots + x^T(k+i-1)Qx(k+i-1) + (u^{[1]}(k+i-1))^T Ru^{[1]}(k+i-1) \\ + V^{[0]}(x(k+i)). \tag{41}$$

So according to (40), we obtainget

$$\Lambda^{[i+1]}(x(k), \hat{U}(k,k+i)) = V^{[i]}(x(k)). \tag{42}$$

From Theorem 1, we can get

$$V^{[i+1]}(x(k)) \le \Lambda^{[i+1]}(x(k), \hat{U}(k,k+i)). \tag{43}$$

So we have $\forall x(k)$

$$V^{[i+1]}(x(k)) \le V^{[i]}(x(k)), \tag{44}$$

i.e., for $x^{[i]}(k)$

$$V^{[i+1]}(x^{[i]}(k)) \le V^{[i]}(x^{[i]}(k)). \tag{45}$$

For Case 1, we set $V^{[0]} = 0$, the proof is similar with Case 2.

From Theorem 2, we can conclude that the performance index function $\{V^{[i+1]}(x(k))\}$ is a monotonically nonincreasing sequence. As the performance index function is positive definite, so we can say that the performance index function is convergent. Thus we define $V^{\infty}(x(k)) = \lim_{i\to\infty} V^{[i+1]}(x(k))$, $u^{\infty}(k) = \lim_{i\to\infty} u^{[i+1]}(k)$ and $x^{\infty}(k)$ is the state under $u^{\infty}(k)$. In the following, we give a theorem to indicate that $V^{\infty}(x^{\infty}(k))$ satisfies HJB equation.

Theorem 3. *For system (1), the iteration algorithm is as in (15)–(33). Then we haveget*

$$V^{\infty}(x^{\infty}(k)) = (x^{\infty}(k))^T Qx^{\infty}(k) + (u^{\infty}(k))^T Ru^{\infty}(k) + V^{\infty}(x^{\infty}(k+1)). \tag{46}$$

Proof. Let ϵ be an arbitrary positive number. Since $V^{[i+1]}(x(k))$ is nonincreasing and $V^{\infty}(x(k)) = \lim_{i\to\infty} V^{[i+1]}(x(k))$, there exists a positive integer p such thatget

$$V^{[p]}(x(k)) - \epsilon \le V^{\infty}(x(k)) \le V^{[p]}(x(k)). \tag{47}$$

So we have

$$V^{\infty}(x(k)) \ge \inf_{u(k)} \{x^T(k)Qx(k) + u^T(k)Ru(k) + V^{[p-1]}(x(k+1))\} - \epsilon. \tag{48}$$

According to Theorem 2, we have

$$V^{\infty}(x(k)) \ge \inf_{u(k)} \{x^T(k)Qx(k) + u^T(k)Ru(k) + V^{\infty}(x(k+1))\} - \epsilon \tag{49}$$

hold. Since ϵ is arbitrary, we have

$$V^{\infty}(x(k)) \ge \inf_{u(k)} \{x^T(k)Qx(k) + u^T(k)Ru(k) + V^{\infty}(x(k+1))\}. \tag{50}$$

On the other side, according to Theorem 2, we haveget

$$V^{\infty}(x(k)) \le V^{[i+1]}(x(k)) \\ = \inf_{u(k)} \{x^T(k)Qx(k) + u^T(k)Ru(k) + V^{[i]}(x(k+1))\}. \tag{51}$$

Let $i \to \infty$, we haveget

$$V^{\infty}(x(k)) \le \inf_{u(k)} \{x^T(k)Qx(k) + u^T(k)Ru(k) + V^{\infty}(x(k+1))\}. \tag{52}$$

So from (50) and (52), we can getget

$$V^{\infty}(x(k)) = \inf_{u(k)} \{x^T(k)Qx(k) + u^T(k)Ru(k) + V^{\infty}(x(k+1))\}, \forall x(k). \tag{53}$$

According to (29), we obtain $u^{\infty}(k)$. From (32) and (33), we have the corresponding state $x^{\infty}(k)$, thus the following expression:

$$V^{\infty}(x^{\infty}(k)) = (x^{\infty}(k))^T Qx^{\infty}(k) + (u^{\infty}(k))^T Ru^{\infty}(k) + V^{\infty}(x^{\infty}(k+1)) \tag{54}$$

holds, which completes the proof. □

So we can say that $V^{\infty}(x^{\infty}(k)) = J^*(x^*(k))$. Until now, we have proven that for $\forall k$, the iteration algorithm converges to the optimal performance index function when the iteration index $i \to \infty$. For finite-horizon optimal control problem of time-delay systems, another aspect is the length N of the optimal control sequence. In this paper, the specific value of N is not known, but we can analyze the relationship between the iteration index i and the terminal time N.

Theorem 4. *Let the iteration algorithm be in (24)–(33). If $V^{[0]}(x(k+i+1)) = J(x(k+i+1), U^*(k+i+1, k+i+j-1)), \forall x(k+i+1)$, then the state at time step k of system (1) can reach to zero in $N = i+j$ steps for Case 2.*

Proof. For Case 2 of the iteration algorithm, we have

$$V^{[i+1]}(x^{[i+1]}(k)) = (x^{[i+1]}(k))^T Qx^{[i+1]}(k) + (u^{[i+1]}(k))^T Ru^{[i+1]}(k) \\ + (x^{[i]}(k+1))^T Qx^{[i]}(k+1) + (u^{[i]}(k+1))^T Ru^{[i]}(k+1) \\ + \cdots \\ + (x^{[1]}(k+i))^T Qx^{[1]}(k+i) + (u^{[1]}(k+i))^T Ru^{[1]}(k+i) \\ + V^{[0]}(x^{[0]}(k+i+1)).\square \tag{55}$$

According to [28], we can see that the optimal control sequence for $x^{[i+1]}(k)$ is $U^*(k,k+i) = (u^{[i+1]}(k), u^{[i]}(k+1), \ldots, u^{[1]}(k+i))$. As we have $V^{[0]}(x^{[0]}(k+i+1)) = J(x^{[0]}(k+i+1), U^*(k+i+1, k+i+j-1))$, so we can obtain $N = i+j$.

For Case 1 of the proposed iteration algorithm, we have the following corollary.

Corollary 1. *Let the iteration algorithm be in (15)–(23). Then for system (1), the state at time step k can reach to zero in $N = i+1$ steps for Case 1.*

Proof. For Case 1, we have

$$V^{[i+1]}(x^{[i+1]}(k)) = (x^{[i+1]}(k))^T Qx^{[i+1]}(k) + (u^{[i+1]}(k))^T Ru^{[i+1]}(k) \\ + (x^{[i]}(k+1))^T Qx^{[i]}(k+1) + (u^{[i]}(k+1))^T Ru^{[i]}(k+1) \\ + \cdots \\ + (x^{[1]}(k+i))^T Qx^{[1]}(k+i) + (u^{[1]}(k+i))^T Ru^{[1]}(k+i) \\ + V^{[0]}(x^{[0]}(k+i+1)) \\ = J(x^{[i+1]}(k), U(k,k+i)), \tag{56}$$

where $U^*(k,k+i) = (u^{[i+1]}(k), \ldots, u^{[1]}(k+i))$, and each element of $U^*(k,k+i)$ is obtained from (29). According to Case 1, $x^{[0]}(k+i+1) = 0$. So the state at time step k can reach to zero in $N = i+1$ steps. □

We can see that for time step k the optimal controller is obtained when $i \to \infty$, which induces the time steps $N \to \infty$ according to Theorem 4 and Corollary 1. In this paper, we want to get the nearly optimal performance index function within finite N time steps. The following corollary is used to prove the existences of nearly optimal performance index function and nearly optimal control.

Corollary 2. *For system (1), the iteration algorithm is as in (15)–(33), then $\forall \varepsilon > 0$, $\exists I \in \mathcal{N}$, $\forall i > I$, we have*

$$\left| V^{[i+1]}(x^{[i+1]}(k)) - J^*(x^*(k)) \right| \le \varepsilon. \tag{57}$$

Proof. From Theorems 2 and 3, we can see that $\lim_{i\to\infty} V^{[i]}(x^{[i]}(k)) = J^*(x^*(k))$, then from the limitation definition, the conclusion is obtained easily. □

So we can say that $V^{[i]}(x^{[i]}(k))$ is the nearly optimal performance index function in the sense of ε, the corresponding nearly optimal control is defined as follows:

$$u_\varepsilon(k) = \arg\inf_{u(k)}\{x^T(k)Qx(k)+u^T(k)Ru(k)+V^{[i]}(x(k+1))\}, \tag{58}$$

Remark 6. From Theorem 4 and Corollary 1, we can see that the length of the control sequence N is dependent on the iteration step. In addition, from Corollary 2, we know that the iteration step is dependent on ε. So it is concluded that the length of the control sequence N is dependent on ε.

From (57), we can see that the inequality is hard to satisfy. So in practice, we adopt the following standard to substitute (57):

$$\left|V^{[i+1]}(x^{[i+1]}(k)) - V^{[i]}(x^{[i]}(k))\right| \le \varepsilon. \tag{59}$$

3.3. Neural network implementation of the iteration ADP algorithm

The nonlinear optimal control solution relies on solving the HJB equation, and the exact solution of which is generally impossible to be obtained for nonlinear time-delay system. So we employ neural networks for approximations of $u^{[i]}(k)$ and $J^{[i+1]}(x(k))$ in this section.

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

$$\hat{F}(X, W, \hat{W}) = W^T\sigma(\hat{W}^T X), \tag{60}$$

where $\sigma(\hat{W}^T X) \in R^l, [\sigma(z)]_i = (e^{z_i} - e^{-z_i})/(e^{z_i}+e^{-z_i}), i = 1, \ldots l$, are the activation function. The gradient descent rule is adopted for the weight update rules of each neural network.

Here, there are two networks, which are critic network and action network. Both neural networks are chosen as three-layer back-propagation (BP) neural networks. The whole structure diagram is shown in Fig. 1.

3.3.1. The critic network

The critic network is used to approximate the performance index function $V^{[i+1]}(x(k))$. The output of the critic network is denoted as follows:

$$\hat{V}^{[i+1]}(x(k)) = (w_c^{[i+1]})^T\sigma((v_c^{[i+1]})^T x(k)). \tag{61}$$

The target function can be written as follows:

$$V^{[i+1]}(x(k)) = x^T(k)Qx(k) + (\hat{u}^{[i+1]}(k))^T R\hat{u}^{[i+1]}(k) + \hat{V}^{[i]}(x(k+1)). \tag{62}$$

Then we define the error function for the critic network as follows:

$$e_c^{[i+1]}(k) = \hat{V}^{[i+1]}(x(k)) - V^{[i+1]}(x(k)). \tag{63}$$

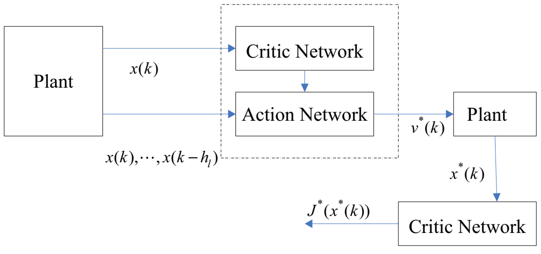


Fig. 1. The structure diagram of the algorithm.

The objective function to be minimized in the critic network is

$$E_c^{[i+1]}(k) = \frac{1}{2}(e_c^{[i+1]}(k))^2. \tag{64}$$

So the gradient-based weights update rule for the critic network is given by

$$\begin{aligned} w_c^{[i+2]}(k) &= w_c^{[i+1]}(k) + \Delta w_c^{[i+1]}(k),\\ v_c^{[i+2]}(k) &= v_c^{[i+1]}(k) + \Delta v_c^{[i+1]}(k), \end{aligned} \tag{65}$$

where

$$\begin{aligned} \Delta w_c^{[i+1]}(k) &= -\alpha_c \frac{\partial E_c^{[i+1]}(k)}{\partial w_c^{[i+1]}(k)},\\ \Delta v_c^{[i+1]}(k) &= -\alpha_c \frac{\partial E_c^{[i+1]}(k)}{\partial v_c^{[i+1]}(k)}, \end{aligned} \tag{66}$$

and the learning rate α_c of critic network is positive number.

3.3.2. The action network

In the action network the states $x(k), \ldots, x(k-h_l)$ are used as inputs to create the optimal control, $\hat{u}^{[i]}(k)$ as the output of the network. The output can be formulated as follows:

$$\hat{u}^{[i]}(k) = (w_a^{[i]})^T\sigma((v_a^{[i]})^T Y(k)), \tag{67}$$

where $Y(k) = [x^T(k), \ldots, x^T(k-h_l)]^T$.

We define the output error of the action network as follows:

$$e_a^{[i]}(k) = \hat{u}^{[i]}(k) - u^{[i]}(k). \tag{68}$$

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

$$E_a^{[i]}(k) = \frac{1}{2}(e_a^{[i]}(k))^T e_a^{[i]}(k). \tag{69}$$

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

$$\begin{aligned} w_a^{[i+1]}(k) &= w_a^{[i]}(k) + \Delta w_a^{[i]}(k),\\ v_a^{[i+1]}(k) &= v_a^{[i]}(k) + \Delta v_a^{[i]}(k), \end{aligned} \tag{70}$$

where

$$\begin{aligned} \Delta w_a^{[i]}(k) &= -\alpha_a \frac{\partial E_a^{[i]}(k)}{\partial w_a^{[i]}(k)},\\ \Delta v_a^{[i]}(k) &= -\alpha_a \frac{\partial E_a^{[i]}(k)}{\partial v_a^{[i]}(k)}, \end{aligned} \tag{71}$$

and the learning rate α_a of action network is the positive number.

In the next section, we will give the simulation study to explain the proposed iteration algorithm in details.

4. Simulation study

4.1. Example 1

We take the example in [28] with modification

$$x(t+1) = x(t-2) + \sin(0.1x^2(t) + u(t)). \tag{72}$$

We give the initial states as $\chi_1(-2) = \chi_1(-1) = \chi_1(0) = 1.5$, and the initial control policy as $\beta(t) = \sin^{-1}(x(t+1) - x(t-2)) - 0.1x^2(t)$. We implement the proposed algorithm at the time instant k=3.

First, according to the initial control policy $\beta(t) = \sin^{-1}(x(t+1) - x(t-2)) - 0.1x^2(t)$ of system (72), we give fist group of state data: $x(1) = 0.8, x(2) = 0.7, x(3) = 0.5, x(4) = 0$. We also can get the second group of state data: $x(1) = 1.4$, $x(2) = 1.2$, $x(3) = 1.1$, $x(4) = 0.8$, $x(5) = 0.7$, $x(6) = 0.5$, $x(7) = 0$. Obviously, for the first sequences of states we can get the optimal controller by Case 1 of the proposed algorithm. For the second one, the optimal controller can be obtained by Case 2 of the proposed algorithm, and the optimal control sequence $U^o(k+1,k+j+1)$ can be obtained in the first group of state data. We select $Q=R=1$.

The three-layer BP neural networks are used to approach the critic network and the action network with the structure $2-8-1$ and $6-8-1$. The iteration times of the weights updating for two neural networks are 200. The initial weights are chosen randomly from $(-0.1,0.1)$, and the learning rates are $\alpha_a=\alpha_c=0.05$. The performance index trajectories for the first and the second state data group are shown in Figs. 2 and 3, respectively. According to Theorem 2, for the first state group, the performance index is decreasing as $i>0$. For the second state group, the performance index is decreasing as $i\geq 0$. The state trajectory and the control trajectory of the second state data are shown in Figs. 4 and 5. From the figures, we can see that the system is asymptotically stable. The simulation study shows the new iteration ADP algorithm is very feasible.

4.2. Example 2

For demonstrating the effectiveness of the proposed iteration algorithm in this paper, we give a more substantial application. Consider the ball and beam experiment. A ball is placed on a beam as shown in Fig. 6.

The beam angle α can be expressed in terms of the servo gear angle θ as $\alpha \approx (2d/L)\theta$. The equation of motion for the ball is given as follows:

$$\left(\frac{M}{R^2}+m\right)\ddot{r}+mg\sin\alpha-mr(\dot{\alpha})^2=0, \tag{73}$$

where r is the ball position coordinate. The mass of the ball m = 0.1 kg, the radius of the ball R = 0.015 m, the radius of the lever gear d = 0.03 m, the length of the beam L = 1.0 m and the ball's moment of inertia $M=10^{-5}$ kg m^2. Given the time step Δh, let $r(t)=r(t\Delta h)$, $\alpha(t)=\alpha(t\Delta h)$ and $\theta(t)=\theta(t\Delta h)$, then Eq. (73) is discretized as

$$\begin{cases} x(t+1)=x(t)+y(t)-A\sin\left(\dfrac{2d}{L}\theta(t)\right)+Bx(t)(\theta(t)-z(t))^2 \\ y(t+1)=y(t)-A\sin\left(\dfrac{2d}{L}\theta(t)\right)+Bx(t)(\theta(t)-z(t))^2 \\ z(t+1)=\theta(t), \end{cases} \tag{74}$$

where $A=mg\Delta h^2R^2/(M+mR^2)$ and $B=4d^2mR^2/(L^2(M+mR^2))$. The state $X(t)=(x(t),y(t),z(t))^T$, in which $x(t)=r(t)$, $y(t)=r(t)-r(t-1)$ and $z(t)=\theta(t-1)$. The control input is $u(t)=\theta(t)$. For the convenience

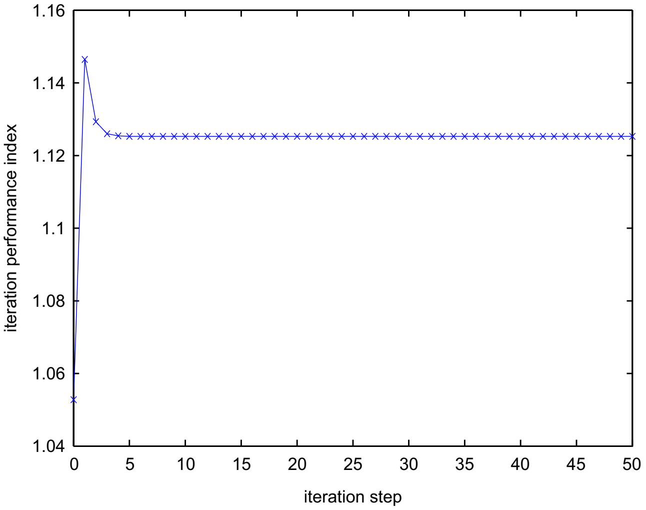


Fig. 2. The performance trajectory for $x(3)=0.5$.

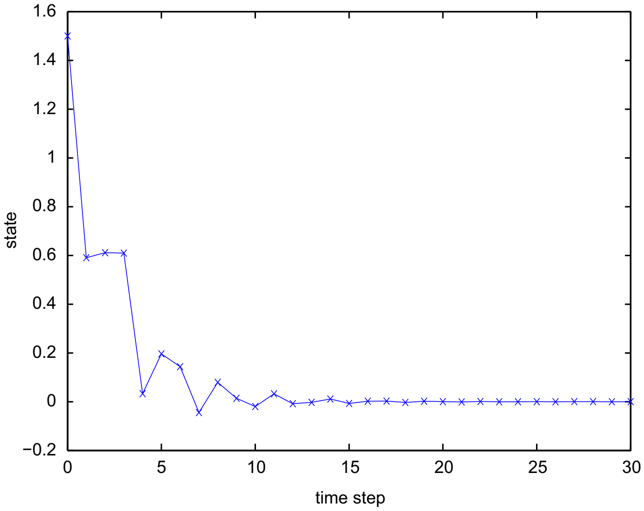


Fig. 4. The state trajectory using the second state data group.

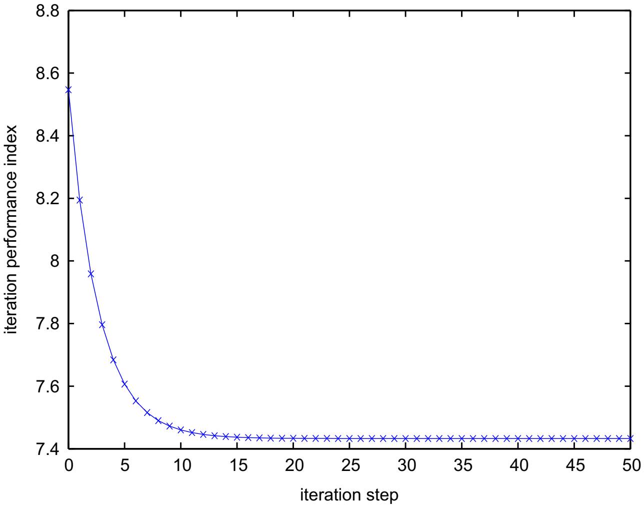


Fig. 3. The performance trajectory for $x(3)=1.1$.

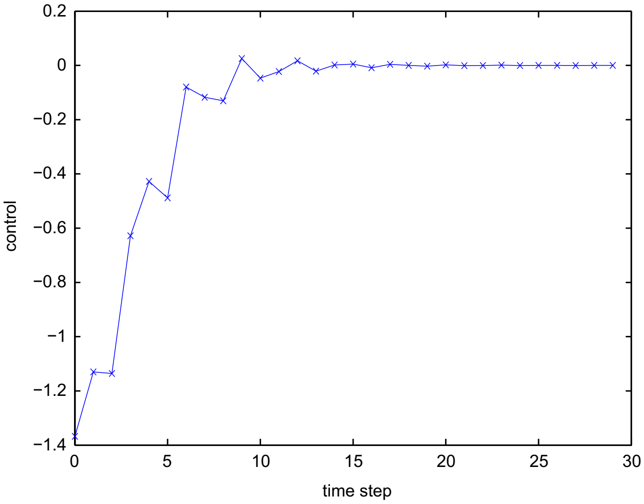


Fig. 5. The control trajectory using the second state data group.

of analysis, system (74) is rewritten as follows:

$$\begin{cases} x(t+1)=x(t-2)+y(t)-A\sin\left(\dfrac{2d}{L}\theta(t)\right)+Bx(t)(\theta(t)-z(t))^2 \\ y(t+1)=y(t)-A\sin\left(\dfrac{2d}{L}\theta(t)\right)+Bx(t)(\theta(t)-z(t-2))^2 \\ z(t+1)=\theta(t). \end{cases} \quad (75)$$

In this paper, Δh is selected as 0.1, the states of time-delay system (75) are $X(1)=[1.0027, 0.0098, 1]^T$, $X(2)=[0.0000, 0.0057, 1.0012]^T$, $X(3)=[1.0000, 0.0016, 1.0000]^T$, $X(4)=[1.0002, -0.0025, 0.9994]^T$ and $X(5)=[0,0,0]^T$. The initial states are $\chi(-2)=[0.9929, 0.0221, 1.0000]^T$, $\chi(-1)=[-0.0057, 0.0180, 1.0000]^T$ and $\chi(0)=[0.9984, 0.0139, 1.0000]^T$. The initial control sequence is (1.0000, 1.0012, 1.0000, 0.9994, 0.0000). Obviously, the initial control sequence and states are not the optimal ones, so the proposed algorithm in this paper is adopt to obtain the optimal solution. We select $Q=R=1$. The iteration times of the weights updating for two neural networks are 200. The initial weights of critic network are chosen randomly from $(-0.1, 0.1)$, the initial weights of action network are chosen randomly from $[-2,2]$, and the learning rates are $\alpha_a=\alpha_c=0.001$. For the state $X(4)=[1.0002, -0.0025, 0.9994]^T$. For the state $X(1)=[1.0027, 0.0098, 1]^T$. Obviously, for the state $X(4)$ we can get the optimal controller by Case 1 of the proposed algorithm. For the state $X(1)$, the optimal controller can be obtained by Case 2 of the proposed algorithm. Then we obtain the performance index function trajectories of the two states as shown in Figs. 7 and 8, which satisfy Theorem 2, i.e., for the state $X(4)$, the performance index is

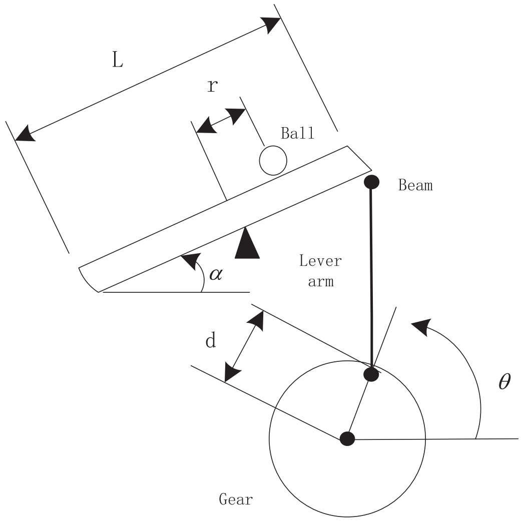


Fig. 6. Ball and beam experiment.

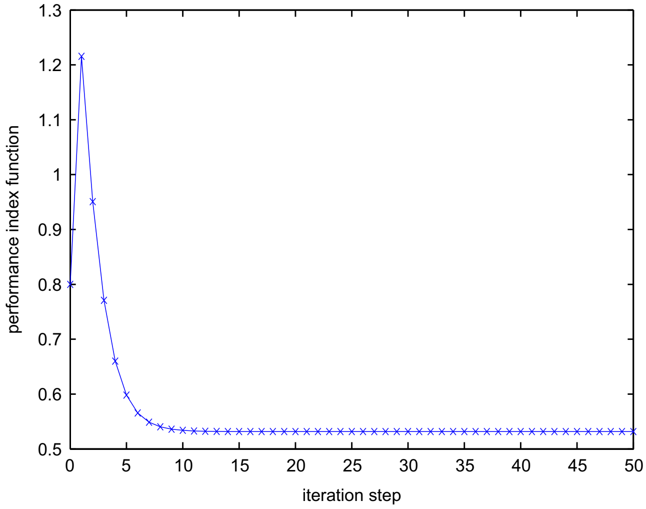


Fig. 7. The performance trajectory for $X(4)$.

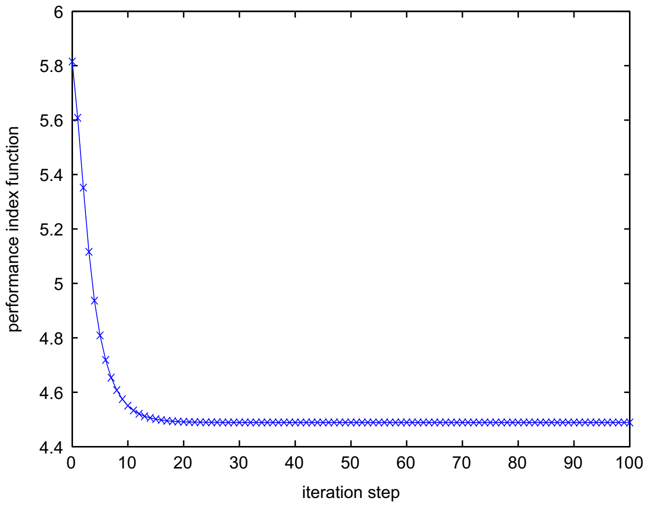


Fig. 8. The performance trajectory for $X(1)$.

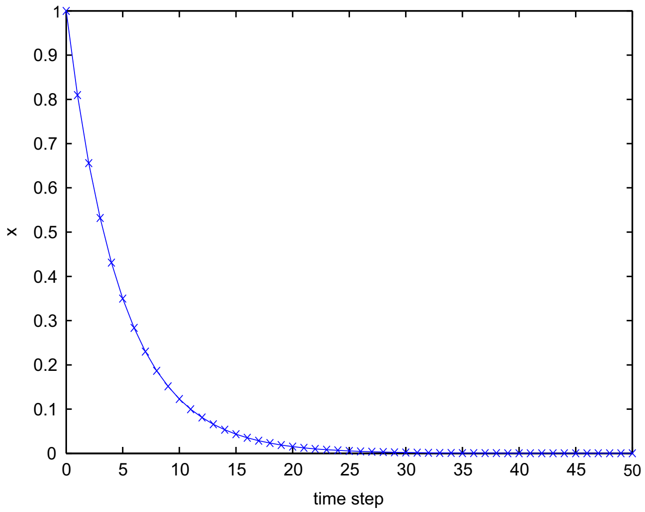


Fig. 9. The state trajectory of $x(t)$.

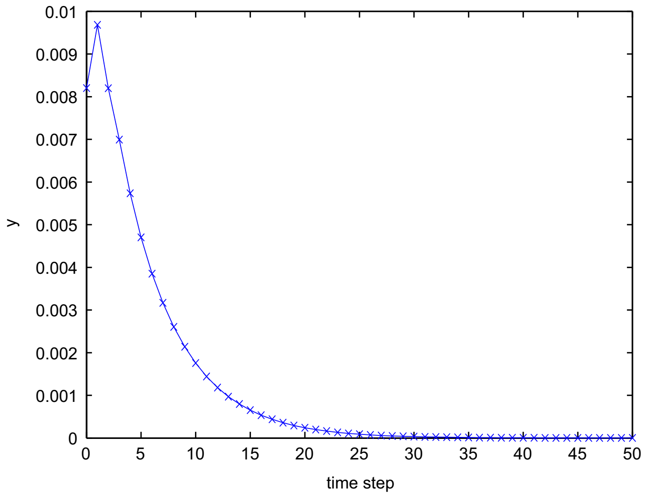


Fig. 10. The state trajectory of $y(t)$.

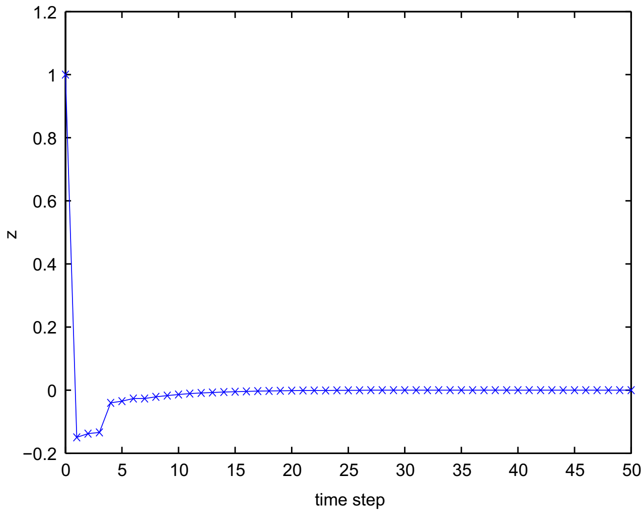


Fig. 11. The state trajectory of $z(t)$.

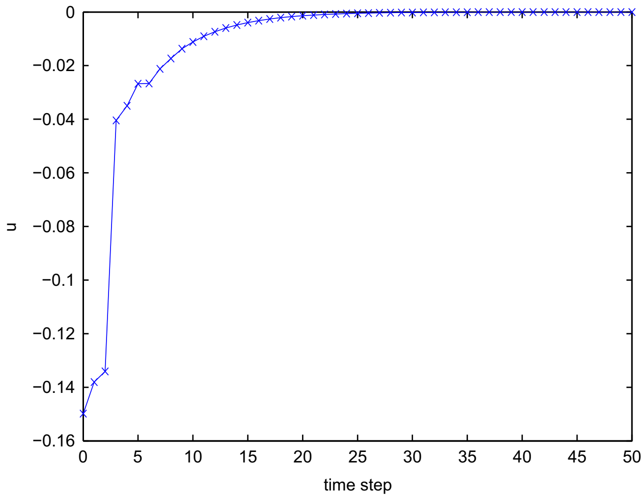


Fig. 12. The control trajectory.

decreasing as $i>0$, for the state $X(1)$, the performance index is decreasing as $i \geq 0$. The state trajectories and the control trajectory of state $X(1)$ are shown in Figs. 9–12. From the figures, we can see that the states of the system are asymptotically stable. Based on the above analysis, we can conclude that the proposed iteration ADP algorithm is satisfactory.

5. Conclusion

This paper proposed a novel ADP algorithm to deal with the nearly finite-horizon optimal control for a class of deterministic nonaffine time-delay nonlinear systems. For determining the optimal state, the state updating was contained in the novel ADP algorithm. The results of theorems showed the proposed iteration algorithm was convergent. Moreover, the relationship between the iteration steps and time steps was given. The simulation study has demonstrated the effectiveness of the proposed control algorithm.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61304079 and 61374105, in part by the Beijing Natural Science Foundation under Grants 4132078 and 4143065, in part by the China Postdoctoral Science Foundation under Grant 2013M530527, and in part by Fundamental Research Funds for the Central Universities under Grant FRF-TP-14-119A2, and in part by the Open Research Project from SKLMCCS under Grant 20120106.

References

[1] S. Niculescu, Delay Effects on Stability: A Robust Control Approach, Springer, Berlin, 2001.

[2] K. Gu, V. Kharitonov, J. Chen, Stability of Time-Delay Systems, Birkhäuser, Boston, 2003.

[3] R. Song, H. Zhang, Y. Luo, Q. Wei, Optimal control laws for time-delay systems with saturating actuators based on heuristic dynamic programming, Neurocomputing 73 (16-18) (2010) 3020–3027.

[4] J. Huang, F. Lewis, Neural-network predictive control for nonlinear dynamic systems with time-delay, IEEE Trans. Neural Netw. 14 (2) (2003) 377–389.

[5] C. Chen, G. Wen, Y. Liu, F. Wang, Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks, IEEE Trans. Neural Netw. Learn. Syst. 25 (6) (2014) 1217–1226.

[6] Y. Shen, J. Wang, Robustness analysis of global exponential stability of recurrent neural networks in the presence of time delays and random disturbances, IEEE Trans. Neural Netw. 23 (1) (2012) 87–96.

[7] R. Song, H. Zhang, Y. Luo, Q. Wei, Optimal control laws for time-delay systems with saturating actuators based on heuristic dynamic programming, Neurocomputing 73 (16–18) (2010) 3020–3027.

[8] D. Chyung, On the controllability of linear systems with delay in control, IEEE Trans. Autom. Control 15 (2) (1970) 255–257.

[9] D. Chyung, Controllability of linear systems with multiple delays in control, IEEE Trans. Autom. Control 15 (6) (1970) 694–695.

[10] V. Phat, Controllability of discrete-time systems with multiple delays on controls and states, Int. J. Control 49 (5) (1989) 1645–1654.

[11] D. Chyung, Discrete optimal systems with time delay, IEEE Trans. Autom. Control 13 (1) (1968) 117.

[12] D. Chyung, E. Lee, Linear optimal systems with time delays, SIAM J. Control. 4 (November (3)) (1966).

[13] Q. Wei, D. Liu, Neural-network-based adaptive optimal tracking control scheme for discrete-time nonlinear systems with approximation errors, Neurocomputing 149 (Part A 3) (2014) 106–115.

[14] S. Li, M. Fairbank, C. Johnson, D. Wunsch, E. Alonso, J. Proano, Artificial neural networks for control of a grid-connected rectifier/inverter under disturbance, dynamic and power converter switching conditions, IEEE Trans. Neural Netw. Learn. Syst. 25 (4) (2013) 738–750.

[15] P. He, S. Jagannathan, Reinforcement learning neural-network-based controller for nonlinear discrete-time systems with input constraints, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 37 (Apr. (2)) (2007) 425–436.

[16] R. Bellman, Dynamic Programming, Princeton University Press, Princeton, NJ, 1957.

[17] P. Werbos, Advanced forecasting methods for global crisis warning and models of intelligence, in: General Systems Yearbook, vol. 22, 1977, pp. 25–38.

[18] P. Werbos, Approximate dynamic programming for real-time control and neural modeling, in: D.A. White, D.A. Sofge (Eds.), Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches, Van Nostrand Reinhold, New York, 1992, Chapter 13.

[19] J. Zhang, H. Zhang, Y. Luo, T. Feng, Model-free optimal control design for a class of linear discrete-time systems with multiple delays using adaptive dynamic programming, Neurocomputing 135 (5) (2014) 163–170.

[20] D. Zhao, Z. Hu, Z. Xia, C. Alippi, Y. Zhu, D. Wang, Full-range adaptive cruise control based on supervised adaptive dynamic programming, Neurocomputing 125 (11) (2014) 57–67.

[21] R. Song, W. Xiao, H. Zhang, Multi-objective optimal control for a class of unknown nonlinear systems based on finite-approximation-error ADP algorithm, Neurocomputing 119 (7) (2013) 212–221.

[22] Y. Luo, Q. Sun, H. Zhang, L. Cui, Adaptive critic design-based robust neural network control for nonlinear distributed parameter systems with unknown dynamics, Neurocomputing 148 (19) (2015) 200–208.

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

[24] 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 Trans. Syst. Man Cybern. Part B: Cybern. 38 (Aug. (4)) (2008) 937–942.

[25] A. Al-Tamimi, F. Lewis, Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof, in: IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, Honolulu, April 2007, pp. 38–43.

[26] H. Zhang, R. Song, Q. Wei, T. Zhang, 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) (2011) 1851–1862.

[27] R. Song, W. Xiao, H. Zhang, C. Sun, Adaptive dynamic programming for a class of complex-valued nonlinear systems, IEEE Trans. Neural Netw. Learn. Syst. 25 (9) (2014) 1733–1739.

[28] F. Wang, N. Jin, D. Liu, Q. Wei, Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ϵ-error bound, IEEE Trans. Neural Netw. 22 (2011) 24–36.

[29] M. Manu, J. Mohammad, Time-delay Systems Analysis, Optimization and Applications, North-Holland, New York, USA, 1987.

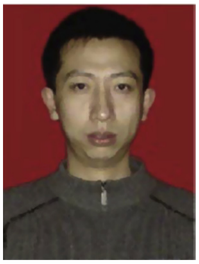
[30] X. Lin, N. Cao, Y. Lin, Optimal control for a class of nonlinear systems with state delay based on adaptive dynamic programming with ϵ-error bound, in: 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, 2013, pp. 177–182.

[31] R. Song, W. Xiao, Q. Wei, Multi-objective optimal control for a class of nonlinear time-delay systems via adaptive dynamic programming, Soft Comput. 17 (11) (2013) 2109–2115.

[32] Q. Wei, H. Zhang, D. Liu, Y. Zhao, An optimal control scheme for a class of discrete-time nonlinear systems with time delays using adaptive dynamic programming, Acta Autom. Sin. 36 (1) (2010) 121–129.

Ruizhuo Song received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2012. She is currently an associate professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing. Her research interests include optimal control, neural-networks-based control, nonlinear control, wireless sensor networks, adaptive dynamic programming and their industrial application.

Qinglai Wei received the B.S. degree in automation, the M.S. degree in control theory and control engineering, and the Ph.D. degree in control theory and control engineering, from the Northeastern University, Shenyang, China, in 2002, 2005, and 2008, respectively. From 2009 to 2011, he was a postdoctoral fellow with Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently an associate professor with The State Key Laboratory of Management and Control for Complex Systems. His research interests include neural-networks-based control, adaptive dynamic programming, optimal control, smart grid, nonlinear system and their industrial applications.



Qiuye Sun received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2007. His main research interests are analysis and diagnosis of uncertain information in power distribution network, optimization analysis technology of power distribution network and network control of distributed generation system. His research of the application of rough set on the massive interactive fault diagnosis of electric power systems was embodied first-class prize of Liaoning Technology Invention Prize and first-class prize of Shenyang Technology Invention Prize.