An Optimal Control Scheme for a Class of Discrete-Time Nonlinear Systems with Time Delays Using Adaptive Dynamic Programming

WEI Qing-Lai[1] ZHANG Hua-Guang[2] LIU De-Rong[1] ZHAO Yan[3]

Abstract In this paper, an optimal control scheme for a class of nonlinear systems with time delays in state and control variables with respect to a quadratic performance index function is proposed using a new iterative adaptive dynamic programming (ADP) algorithm. By introducing a delay matrix function, the explicit expression of the optimal control is obtained using the dynamic programming theory and the optimal control can iteratively be obtained using the adaptive critic technique. 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, compute the optimal control policy, solve delay matrix function and model the nonlinear system, respectively, for facilitating the implementation of the iterative ADP algorithm. Two examples are given to demonstrate the validity of the proposed optimal control scheme.

Key words Adaptive dynamic programming, approximate dynamic programming, time delay, optimal control, nonlinear system, neural networks

The optimal control problem of nonlinear systems has always been the key focus in the control field in the last several decades. Coupled with this is the fact that nothing can happen instantaneously, as is so often presumed in many mathematical models. So strictly speaking, time delays exnist in the most practical control systems. Time delays may result in degradation in the control efficiency even instability of the control systems. So there have been many studies on the control systems with time delay in various research fields such as electrical, chemical engineering and networked control[1,2]. The optimal control problem for the time-delay systems always attracts much attention of the researchers and many results have been obtained[3−5]. In general, the optimal control for the time-delay systems is an infinite-dimensional control problem[3], which is very difficult to solve. So lots of analysis and applications are limited to a very simple case: the linear systems with only state delays[6]. For nonlinear case with state delays, the traditional method is to adopt fuzzy method and robust method which transform the nonlinear time-delay systems to a linear one [7]. For the systems with time delays both in states and controls, it is still an open problem [4,5]. The main difficulty lies in the formulation of the optimal controller which must use the information of the delayed control term so as to obtain an efficient control. This makes the analysis of the system much more difficult, and there is no method strictly facing this problem even in the linear cases, much or less for the nonlinear cases. This motivates our research.

Adaptive dynamic programming (ADP), combining adaptive critic and reinforcement learning into dynamic programming[8,9], is a powerful tool in solving the optimal control problems and attached much attention by many researchers and groups in recent years, such as $[10-16]$. However, most of the results focus on the optimal control problems without delays. To the best of our knowledge, there are no results discussing how to use ADP to solve the time-delay optimal control problems. In this paper, it is the first time that the time-delay optimal control problem is solved by the iterative ADP algorithm. By introducing a delay matrixn function, we can obtain the explicit expression of the optimal control function. The optimal control can iteratively be obtained using the proposed iterative ADP algorithm which avoids the infinite-dimensional computation. Also, it is proved that the performance index function converges to the optimum using the proposed iterative ADP algorithm.

This paper is organized as follows. Section 1 presents the preliminaries. In Section 2, the time-delay optimal control scheme is proposed based on iterative ADP algorithm. In Section 3, the neural network implementation for the control scheme is discussed. In Section 4, two examples are given to demonstrate the effectiveness of the proposed control scheme. The conclusion is drawn in Section 5.

1 Preliminaries

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

$$\begin{aligned}\boldsymbol{x}(k+1) =& f(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma)) + g_0(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma))\boldsymbol{u}(k)\\ &+ g_1(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma))\boldsymbol{u}(k-\tau) \qquad (1)\end{aligned}$$

with the initial condition given by $\boldsymbol{x}(s)=\phi(s)$, $s=-\sigma,-\sigma+1,\dots,0$, where $\boldsymbol{x}(k)\in\Re^{\rm n}$ is the state vector, $f\colon \Re^{\rm n}\times\Re^{\rm n}\to\Re^{\rm n}$ and $g_0,g_1\colon\Re^{\rm n}\times\Re^{\rm n}\to\Re^{\rm n\times m}$ are differentiable functions and the control $\boldsymbol{u}(k)\in\Re^{\rm m}$. The state and control delays σ and τ are both nonnegative integral number. Assume that $f(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma))+g_0(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma))\boldsymbol{u}(k)+g_1(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma))\boldsymbol{u}(k-\tau)$ is Lipschitz continuous on a set Ω in $\Re^{\rm n}$ containing the origin, and that the system (1) is controllable in the sense that there exists a bounded control on Ω that asymptotically stabilizes the system. In this paper, we mainly discuss how to design an optimal state feedback controller for this class of delayed discrete-time systems. Therefore, it is desired to find the optimal control $\boldsymbol{u}(\boldsymbol{x})$ satisfying $\boldsymbol{u}(\boldsymbol{x}(k))=\boldsymbol{u}(k)$ to minimize the generalized performance functional as follows

$$\begin{aligned}V(\boldsymbol{x}(0),\boldsymbol{u}) =& \sum_{k=0}^{\infty}\Big(\boldsymbol{x}^{\rm T}(k)Q_0\boldsymbol{x}(k)+2\boldsymbol{x}^{\rm T}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+\boldsymbol{x}^{\rm T}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma)+\boldsymbol{u}^{\rm T}(k)R_0\boldsymbol{u}(k)\end{aligned}$$

Received September 5, 2008; in revised form January 10, 2009
Supported in part by the National Natural Science Foundation of P. R. China (60621001),(60534010), (60572070), (60774048), and (60728307), by the Program for Changjiang Scholars and Innovative Research Groups of P. R. China (60728307),(4031002), and by the National High Technology Research and Development Program of P. R. China (2006AA04Z183)

1. The Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China 2.The School of Information Science and Engineering, Northeastern University, Shenyang 110004, P. R. China 3.Department of Automatic Control Engineering, Shenyang Institute of Engineering, Shenyang 110136

DOI: 10.3724/SP.J.1004.2008.xxxxx

$$+2\boldsymbol{u}^{\mathrm{T}}(k)R_1\boldsymbol{u}(k-\tau)+\boldsymbol{u}^{\mathrm{T}}(k-\tau)R_2\boldsymbol{u}(k-\tau)\Big) \tag{2}$$

where $\begin{bmatrix} Q_0 & Q_1 \\ Q_1^{\mathrm{T}} & Q_2 \end{bmatrix} \geq 0$ and $\begin{bmatrix} R_0 & R_1 \\ R_1^{\mathrm{T}} & R_2 \end{bmatrix} > 0$ and $l(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma),\boldsymbol{u}(k),\boldsymbol{u}(k-\tau)) = \boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma) + \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \boldsymbol{u}^{\mathrm{T}}(k)R_0\boldsymbol{u}(i) + 2\boldsymbol{u}^{\mathrm{T}}(k)R_1\boldsymbol{u}(k-\tau) + \boldsymbol{u}^{\mathrm{T}}(k-\tau)R_2\boldsymbol{u}(k-\tau)$ is the utility function. Let $V^*(\boldsymbol{x})$ denote the optimal performance index function which satisfies

$$V^*(\boldsymbol{x}) = \min_{u} V(\boldsymbol{x},\boldsymbol{u}). \tag{3}$$

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

$$\begin{aligned} V^*(\boldsymbol{x}(k)) = \min_{\boldsymbol{u}(k)} \Big\{ & \boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma) \\ & + \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \boldsymbol{u}^{\mathrm{T}}(k)R_0\boldsymbol{u}(k) \\ & + 2\boldsymbol{u}^{\mathrm{T}}(k)R_1\boldsymbol{u}(k-\tau) + \boldsymbol{u}^{\mathrm{T}}(k-\tau)R_2\boldsymbol{u}(k-\tau) \\ & + V^*(\boldsymbol{x}(k+1)) \Big\}. \end{aligned} \tag{4}$$

For optimal control problem, the state feedback control $\boldsymbol{u}(\boldsymbol{x})$ must not only stabilize the system on Ω but also guarantee that (2) is finite, i.e., $\boldsymbol{u}(\boldsymbol{x})$ must be admissible[17].

Definition 1 *A control $\boldsymbol{u}(\boldsymbol{x})$ is defined to be admissible with respect to (4) on Ω if $\boldsymbol{u}(\boldsymbol{x})$ is continuous on Ω, $\boldsymbol{u}(0) = 0$, $\boldsymbol{u}(\boldsymbol{x})$ stabilizes (1) on Ω, and $\forall \boldsymbol{x}(0) \in \Omega$, $V(\boldsymbol{x}(0))$ is finite.*

2 Properties of the Iterative ADP Approach

Noting that the nonlinear delayed system (1) is infinite-dimensional[3], the control variable $\boldsymbol{u}(k)$ couples with $\boldsymbol{u}(k-\tau)$. It is nearly impossible to obtain the expression of the optimal control by solving the HJB equation (4). To overcome the difficulty, a new iterative algorithm is proposed in this paper. The following lemma is necessary to apply the algorithm.

Lemma 1 *For the delayed nonlinear system (1) with respect to the performance index function (2), if there exists a control $\boldsymbol{u}(k) \neq 0$ at time point k, then there exists a bounded matrix function $M(k)$ that makes*

$$\boldsymbol{u}(k-\tau) = M(k)\boldsymbol{u}(k) \tag{5}$$

hold for $\mathbb{j} = 0,1,\ldots,n$.

Proof. As $\boldsymbol{u}(k)$ and $\boldsymbol{u}(k-\tau_{\mathbb{j}})$, $\mathbb{j} = 0,1,\ldots,n$ are bounded real vector, then we can construct a function that satisfies

$$\boldsymbol{u}(k-\tau) = h(\boldsymbol{u}(k)) \tag{6}$$

where $\mathbb{j} = 0,1,\ldots,n$. Then using the method of undetermined coefficients, let $M(\boldsymbol{u}(k))$ satisfy

$$h(\boldsymbol{u}(k)) = M(\boldsymbol{u}(k))\boldsymbol{u}(k). \tag{7}$$

Then we can obtain $M(\boldsymbol{u}(k))$ expressed as

$$M(\boldsymbol{u}(k)) = h(\boldsymbol{u}(k))\boldsymbol{u}^{\mathrm{T}}(k)\left(\boldsymbol{u}(k)\boldsymbol{u}^{\mathrm{T}}(k)\right)^{-1} \tag{8}$$

where $\left(\boldsymbol{u}(k)\boldsymbol{u}^{\mathrm{T}}(k)\right)^{-1}$ means the generalized inverse matrix of $\left(\boldsymbol{u}(k)\boldsymbol{u}^{\mathrm{T}}(k)\right)$. On the other side, $\boldsymbol{u}(k)$ and $\boldsymbol{u}(k-\tau)$ are both bounded real vector, then we have $h(\boldsymbol{u}(k))$ and $\left(\boldsymbol{u}(k)\boldsymbol{u}^{\mathrm{T}}(k)\right)^{-1}$ are bounded. So $M(k) = M(\boldsymbol{u}(k))$ is the solution. □

According to Lemma 1, the HJB equation becomes

$$\begin{aligned} V^*(\boldsymbol{x}(k)) = & \boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma) \\ & + \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) \\ & + \boldsymbol{u}^{*\mathrm{T}}(k)R_0\boldsymbol{u}^*(k) + 2\boldsymbol{u}^{*\mathrm{T}}(k)R_1M^*(k)\boldsymbol{u}^*(k) \\ & + \boldsymbol{u}^{*\mathrm{T}}(k)M^{*\mathrm{T}}(k)R_2M^*(k)\boldsymbol{u}^*(k) \\ & + V^*(\boldsymbol{x}(k+1)) \end{aligned} \tag{9}$$

where $\boldsymbol{u}^*(k)$ is the optimal control and $\boldsymbol{u}^*(k-\tau) = M^*(k)\boldsymbol{u}^*(k)$.

2.1 Derivation of the Iterative ADP Algorithm

According to the Bellman's principle of optimality, we can obtain the optimal control by differentiating the HJB equation (9) with respect to control u. Then we can obtain the optimal control $\boldsymbol{u}^*(k)$ formulated as

$$\begin{aligned} \boldsymbol{u}^*(k) = & -\frac{1}{2}\left(R_0 + 2R_1M^*(k) + M^{*\mathrm{T}}(k)R_2M^*(k)\right)^{-1} \\ & \times \big(g_0\left(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma)\right) \\ & + g_1\left(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma)\right)M^*(k)\big)^{\mathrm{T}}\frac{\partial V^*(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}. \end{aligned} \tag{10}$$

In equation (10), the inverse of the term $\left(R_0 + 2R_1M^*(k) + M^{*T}(k)R_2M^*(k)\right)$ should exist and a proof is presented in the Appendix to guarantee the existence of the inverse.

From equation (10), the explicit optimal control expression $\boldsymbol{u}^*$ is obtained by solving the HJB equation (9). We can see that the optimal control $\boldsymbol{u}^*$ depends on M^* and $V^*(x)$ where $V^*(x)$ is a solution of the HJB equation (9). While how to solve the HJB equation is still open and there is currently no method for rigorously seeking for this performance index function of this delayed optimal control problem. Furthermore, the optimal delay matrix function M^* is also unknown which makes the optimal control $\boldsymbol{u}^*$ more difficult to obtain. So an iterative index i is introduced into the ADP approach to obtain the optimal control iteratively.

Firstly, for $i = 0,1,\ldots$, let

$$\boldsymbol{u}^{(i+1)}(k-\tau) = M^{(i)}(k)\boldsymbol{u}^{(i+1)}(k) \tag{11}$$

where $M^{(0)}(k) = I$ and $\boldsymbol{u}^{(0)}(k-\tau) = M^{(0)}(k)\boldsymbol{u}^{(0)}(k)$. We start with initial performance index $V^{(0)}(\boldsymbol{x}(k)) = 0$, and the control $\boldsymbol{u}^{(0)}(k)$ can be computed as follows

$$\boldsymbol{u}^{(0)}(\boldsymbol{x}(k)) = \arg\min_{u}\left\{\Gamma^0 + V^{(0)}(\boldsymbol{x}(k+1))\right\}, \tag{12}$$

where

$$\begin{aligned} \Gamma^0 = & \boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma) \\ & + \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \boldsymbol{u}^{(0)T}(k)R_0\boldsymbol{u}^{(0)}(k) \\ & + 2\boldsymbol{u}^{(0)T}(k)R_1M^{(0)}(k)\boldsymbol{u}^{(0)}(k) \\ & + \boldsymbol{u}^{(0)T}(k)M^{(0)T}(k)R_2M^{(0)}(k)\boldsymbol{u}^{(0)}(k). \end{aligned}$$

Then the performance index function is updated as

$$V^{(1)}(\boldsymbol{x}(k)) = \Gamma^0 + V^{(0)}(\boldsymbol{x}(k+1)). \tag{13}$$

Thus for $i = 1, 2, \ldots$, the iterative ADP can be used to implement the iteration between

$$\begin{aligned}\boldsymbol{u}^{(i)}(\boldsymbol{x}(k)) =& \arg\min_{u}\left\{\Gamma^{(i)} + V^{(i)}(\boldsymbol{x}(k+1))\right\}\\ =& -\frac{1}{2}\Big(R_0 + 2R_1M^{(i-1)}(k)\\ &+M^{(i-1)T}(k)R_2M^{(i-1)}(k)\Big)^{-1}\big(g_0(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma))\\ &+g_1(\boldsymbol{x}(k),\boldsymbol{x}(k-\sigma))M^{(i-1)}(k)\big)^{\mathrm{T}}\frac{\partial V^{(i)}(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)},\end{aligned} \tag{14}$$

where

$$\begin{aligned}\Gamma^{(i)} =&\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+ \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \boldsymbol{u}^{(i)T}(k)R_0\boldsymbol{u}^{(i)}(k)\\ &+ 2\boldsymbol{u}^{(i)T}(k)R_1M^{(i-1)}(k)\boldsymbol{u}^{(i)}(k)\\ &+ \boldsymbol{u}^{(i)T}(k)M^{(i-1)T}(k)R_2M^{(i-1)}(k)\boldsymbol{u}^{(i)}(k),\end{aligned}$$

and

$$V^{(i+1)}(\boldsymbol{x}(k)) = \Gamma^{(i)} + V^{(i)}(\boldsymbol{x}(k+1)). \tag{15}$$

Then the optimal control can be obtained iteratively. From (14) and (15), it can be seen that during the iteration process, the control actions for different control steps obey different control laws. After the iteration number of i, the obtained control laws sequence is $(\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(i)})$. For the infinite-horizon problem, 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 $V^{(i)}(\boldsymbol{x}(k))$ will converge when the iteration number $i \to \infty$ under the iterative control $\boldsymbol{u}^{(i)}(k)$ and it will be proved in the following subsection.

2.2 Properties of the Iterative ADP Algorithm

In this subsection, we focus on the proof of convergence of the iteration between (14) and (15), with the performance index $V^{(i)}(\boldsymbol{x}(k)) \to V^{*}(\boldsymbol{x}(k)))$, $\forall k$.

Lemma 2 [17] *Let $\tilde{\boldsymbol{u}}^{(i)}(k), k = 0, 1 \ldots$ be any sequence of control, and $\boldsymbol{u}^{(i)}(k)$ is expressed as (14). Define $V^{(i+1)}(\boldsymbol{x}(k))$ as (15) and $\Lambda^{(i+1)}(\boldsymbol{x}(k))$ as*

$$\begin{aligned}\Lambda^{(i+1)}(\boldsymbol{x}(k)) =&\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+ \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \tilde{\boldsymbol{u}}^{(i)T}(k)R_0\tilde{\boldsymbol{u}}^{(i)}(k)\\ &+ 2\tilde{\boldsymbol{u}}^{(i)T}(k)R_1M^{(i-1)}(k)\tilde{\boldsymbol{u}}^{(i)}(k)\\ &+ \tilde{\boldsymbol{u}}^{(i)T}(k)M^{(i-1)T}(k)R_2M^{(i-1)}(k)\tilde{\boldsymbol{u}}^{(i)}(k)\\ &+ \Lambda^{(i)}(\boldsymbol{x}(k+1)).\end{aligned} \tag{16}$$

If $V^{(0)}(\boldsymbol{x}(k)) = \Lambda^{(0)}(\boldsymbol{x}(k)) = 0$, then $V^{(i)}(\boldsymbol{x}(k)) \leq \Lambda^{(i)}(\boldsymbol{x}(k))$, $\forall i$.

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

Theorem 1 *Let the performance index function $V^{(i)}(\boldsymbol{x}(k))$ be defined by (15). If $\boldsymbol{x}(k)$ for the system (1) is controllable, then there exists an upper bound Y such that $0 \leq V^{(i)}(\boldsymbol{x}(k)) \leq Y, \forall i$.*

Proof. As the system (1) is Lipschitz, $M^{(i)}(k)$ is a bounded matrix for $i = 0, 1, \ldots$. Define a delay matrix function $\bar{M}(k)$ which makes

$$\begin{aligned}&\chi^{\mathrm{T}}\left(R_0 + 2R_1\bar{M}(k) + \bar{M}^{\mathrm{T}}(k)R_2\bar{M}(k)\right)\chi - \chi^{\mathrm{T}}\big(R_0\\ &+2R_1M^{(i)}(k) + M^{(i)T}(k)R_2M^{(i)}(k)\big)\chi \geq 0\end{aligned} \tag{17}$$

hold for $\forall i$, where χ is any nonzero m-dimensional vector. Let $\bar{\boldsymbol{u}}(k), k = 0, 1 \ldots$ be any admissible control input. Define a new sequence $P^{(i)}(\boldsymbol{x}(k))$ as follows:

$$\begin{aligned}P^{(i+1)}(\boldsymbol{x}(k)) =&\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+ \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \bar{\boldsymbol{u}}^{\mathrm{T}}(k)R_0\bar{\boldsymbol{u}}(k)\\ &+ 2\bar{\boldsymbol{u}}^{\mathrm{T}}(k)R_1\bar{M}(k)\bar{\boldsymbol{u}}(k)\\ &+ \bar{\boldsymbol{u}}^{\mathrm{T}}(k)\bar{M}^{\mathrm{T}}(k)R_2\bar{M}(k)\bar{\boldsymbol{u}}(k)+P^{(i)}(\boldsymbol{x}(k+1))\end{aligned} \tag{18}$$

where let $P^{(0)}(\boldsymbol{x}(k)) = V^{(0)}(\boldsymbol{x}(k)) = 0$ and $\bar{\boldsymbol{u}}(k-\tau) = \bar{M}(k)\bar{\boldsymbol{u}}(k)$. $V^{(i)}(\boldsymbol{x}(k))$ is updated by (15). Thus we can obtain

$$\begin{aligned}P^{(i+1)}(\boldsymbol{x}(k)) - P^{(i)}(\boldsymbol{x}(k)) &= P^{(i)}(\boldsymbol{x}(k+1)) - P^{(i-1)}(\boldsymbol{x}(k+1))\\ &\;\;\vdots\\ &= P^{(1)}(\boldsymbol{x}(k+i)) - P^{(0)}(\boldsymbol{x}(k+i)).\end{aligned} \tag{19}$$

Because $P^{(0)}(\boldsymbol{x}(k+i)) = 0$, we have

$$\begin{aligned}P^{(i+1)}(\boldsymbol{x}(k)) =&P^{(1)}(\boldsymbol{x}(k+i)) + P^{(i)}(\boldsymbol{x}(k))\\ =&\sum_{j=0}^{i} P^{(1)}(\boldsymbol{x}(k+j)).\end{aligned} \tag{20}$$

According to (18), (20) can be rewritten as

$$P^{(i+1)}(\boldsymbol{x}(k)) = \sum_{j=0}^{i} \Xi(k+j) \leq \sum_{j=0}^{\infty} \Xi(k+j) \tag{21}$$

where

$$\begin{aligned}\Xi(k+j) =& \boldsymbol{x}^{\mathrm{T}}(k+j)Q_0\boldsymbol{x}(k+j)\\ &+2\boldsymbol{x}^{\mathrm{T}}(k+j)Q_1\boldsymbol{x}(k+j-\sigma)\\ &+\boldsymbol{x}^{\mathrm{T}}(k+j-\sigma)Q_2\boldsymbol{x}(k+j-\sigma)\\ &+ \bar{\boldsymbol{u}}^{\mathrm{T}}(k+j)R_0\bar{\boldsymbol{u}}(k+j)\\ &+ 2\bar{\boldsymbol{u}}^{\mathrm{T}}(k+j)R_1\bar{M}(k+j)\bar{\boldsymbol{u}}(k+j)\\ &+\bar{\boldsymbol{u}}^{\mathrm{T}}(k+j)\bar{M}^{\mathrm{T}}(k+j)R_2\bar{M}(k+j)\boldsymbol{u}(k+j).\end{aligned}$$

Noting that the control input $\bar{\boldsymbol{u}}(k), k = 0, 1, \ldots$ is an admissible control, we can obtain

$$\forall i : P^{(i+1)}(\boldsymbol{x}(k)) \leq \sum_{j=0}^{\infty} P^{(1)}(\boldsymbol{x}(k+j)) \leq Y. \tag{22}$$

From Lemma 1, we have

$$\forall i : V^{(i+1)}(\boldsymbol{x}(k)) \leq P^{(i+1)}(\boldsymbol{x}(k)) \leq Y. \tag{23}$$

□

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

Theorem 2 *Define the performance index function $V^{(i)}(\boldsymbol{x}(k))$ as (15), with $V^{(0)}(\boldsymbol{x}(k)) = 0$. If $\boldsymbol{x}(k)$ for the system (1) is controllable, then $V^{(i)}(\boldsymbol{x}(k))$ is a nondecreasing sequence that is $V^{(i)}(\boldsymbol{x}(k)) \leq V^{(i+1)}(\boldsymbol{x}(k))$ and $V^{(i)}(\boldsymbol{x}(k))$ is convergent as $i \to \infty$.*

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

$$\begin{aligned}\Phi^{(i+1)}(\boldsymbol{x}(k)) =&\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+\boldsymbol{x}^{T}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma)+\boldsymbol{u}^{(i+1)T}(k)R_0\boldsymbol{u}^{(i+1)}(k)\\ &+2\boldsymbol{u}^{(i+1)T}(k)R_1M^{(i)}(k)\boldsymbol{u}^{(i+1)}(k)\\ &+\boldsymbol{u}^{(i+1)T}(k)M^{(i)T}(k)R_2M^{(i)}(k)\boldsymbol{u}^{(i+1)}(k)\\ &+\Phi^{(i)}(\boldsymbol{x}(k+1)) \end{aligned} \tag{24}$$

with $\boldsymbol{u}^{(i)}(k)$ obtained by (14) and $\Phi_0(\boldsymbol{x}(k)) = V_0(\boldsymbol{x}(k)) = 0$. $V^{(i)}(\boldsymbol{x}(k))$ is updated by (15).

In the following part, we prove $\Phi^{(i)}(\boldsymbol{x}(k)) \leq V^{(i+1)}(\boldsymbol{x}(k))$ by mathematical induction.

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

$$\begin{aligned}V^{(1)}(\boldsymbol{x}(k))-\Phi^{(0)}(\boldsymbol{x}(k)) =&\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+\boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma)\\ \geq& \ 0. \end{aligned} \tag{25}$$

Thus for $i = 0$, we can get

$$V^{(1)}(\boldsymbol{x}(k)) \geq \Phi^{(0)}(\boldsymbol{x}(k)). \tag{26}$$

Second, we assume it holds for $i-1$, i.e. $V^{(i)}(\boldsymbol{x}(k)) - \Phi^{(i-1)}(\boldsymbol{x}(k)) \geq 0$, $\forall \boldsymbol{x}(k)$. Then, for i, from (15) and (24), we can obtain

$$\begin{aligned}V^{(i+1)}(\boldsymbol{x}(k))-\Phi^{(i)}(\boldsymbol{x}(k)) =&V^{(i)}(\boldsymbol{x}(k+1))-\Phi^{(i-1)}(\boldsymbol{x}(k+1))\\ \geq&0, \end{aligned} \tag{27}$$

i.e.,

$$\Phi^{(i)}(\boldsymbol{x}(k)) \leq V^{(i+1)}(\boldsymbol{x}(k)). \tag{28}$$

Therefore, the mathematical induction proof is completed.

Moreover, from Lemma 1, we know that $V^{(i)}(\boldsymbol{x}(k)) \leq \Phi^{(i)}(\boldsymbol{x}(k))$ and therefore we can obtain

$$V^{(i)}(\boldsymbol{x}(k)) \leq \Phi^{(i)}(\boldsymbol{x}(k)) \leq V^{(i+1)}(\boldsymbol{x}(k)) \tag{29}$$

which proves that $V^{(i)}(\boldsymbol{x}(k))$ is a nondecreasing sequence bounded by (23). Hence, we conclude that $V^{(i)}(\boldsymbol{x}(k))$ a nondecreasing convergent sequence as $i \to \infty$. □

We note the obvious corollary.

Corollary 1 *If Theorem 2 holds, then the delay matrix function $M^{(i)}(k)$ is a convergent sequence, as $i \to \infty$.*

According to Corollary 1, we define

$$M^{(\infty)}(k) = \lim_{i\to\infty} M^{(i)}(k). \tag{30}$$

Next we will prove that the performance index function sequence $V^{(i)}(\boldsymbol{x}(k))$ converges to $V^{*}(\boldsymbol{x}(k))$ as $i \to \infty$. As $V^{(i)}(\boldsymbol{x}(k))$ is a convergent sequence as $i \to \infty$, we define

$$V^{(\infty)}(\boldsymbol{x}(k)) = \lim_{i\to\infty} V^{(i)}(\boldsymbol{x}(k)). \tag{31}$$

Let $\bar{\boldsymbol{u}}_l$ be the lth admissible control, similar to the proof of Theorem 1, we can construct the performance index function sequence $P_l^{(i)}(\boldsymbol{x})$ as follows

$$\begin{aligned}P_l^{(i+1)}(\boldsymbol{x}(k)) =&\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+\boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma)\\ &+\bar{\boldsymbol{u}}_l^{\mathrm{T}}(k)R_0\bar{\boldsymbol{u}}_l(k) + 2\bar{\boldsymbol{u}}_l(k)R_1M^{(\infty)}(k)\bar{\boldsymbol{u}}_l(k)\\ &+\bar{\boldsymbol{u}}_l(k)M^{(\infty)T}(k)R_2M^{(\infty)}(k)\bar{\boldsymbol{u}}_l(k)\\ &+P_l^{(i)}(\boldsymbol{x}(k+1)), \end{aligned} \tag{32}$$

with $P_l^{(0)}(\cdot) = 0$ and $\bar{\boldsymbol{u}}_l(k) = M^{(\infty)}(k)\bar{\boldsymbol{u}}_l(k-\tau)$. According to Theorem 1, we have

$$\begin{aligned}P_l^{(i+1)}(\boldsymbol{x}(k)) =&\sum_{j=0}^{i}\Big(\boldsymbol{x}^{\mathrm{T}}(k+j)Q_0\boldsymbol{x}(k+j)\\ &+2\boldsymbol{x}^{\mathrm{T}}(k+j)Q_1\boldsymbol{x}(k+j-\sigma)\\ &+\boldsymbol{x}^{\mathrm{T}}(k+j-\sigma)Q_2\boldsymbol{x}(k+j-\sigma)\\ &+\bar{\boldsymbol{u}}_l^{\mathrm{T}}(k+j)R_0\bar{\boldsymbol{u}}_l(k+j)\\ &+2\bar{\boldsymbol{u}}_l^{\mathrm{T}}(k+j)R_1M^{(\infty)}(k+j)\bar{\boldsymbol{u}}_l(k+j)\\ &+\bar{\boldsymbol{u}}_l^{\mathrm{T}}(k+j)M^{(\infty)T}(k+j)R_2\\ &\times M^{(\infty)}(k+j)\bar{\boldsymbol{u}}_l(k+j)\Big) \end{aligned} \tag{33}$$

Let

$$P_l^{(\infty)}(\boldsymbol{x}(k)) = \lim_{i\to\infty} P_l^{(i+1)}(\boldsymbol{x}(k)) \tag{34}$$

So we have

$$P_l^{(i)}(\boldsymbol{x}(k)) \leq P_l^{(\infty)}(\boldsymbol{x}(k)). \tag{35}$$

Theorem 3 *Define $P_l^{(\infty)}(\boldsymbol{x}(k))$ as in (34), define the performance index function $V^{(i)}(\boldsymbol{x}(k))$ as in (15) with $V^{(0)}(\cdot) = 0$. For any state vector $\boldsymbol{x}(k)$, define $V^{*}(\boldsymbol{x}(k)) = \min\limits_{l}\left\{P_l^{(\infty)}(\boldsymbol{x}(k))\right\}$ starting from $\boldsymbol{x}(k)$ for all admissible control sequences. Then we can conclude that $V^{*}(\boldsymbol{x}(k))$ is the limit of the performance index function $V^{(i)}(\boldsymbol{x}(k))$ as $i \to \infty$.*

Proof. For any l, there exists an upper bound Y_l such that

$$P_l^{(i+1)}(\boldsymbol{x}(k)) \leq P_l^{(\infty)}(\boldsymbol{x}(k)) \leq Y_l \tag{36}$$

According to (23), for $\forall l$, we have

$$V^{(\infty)}(\boldsymbol{x}(k)) \leq P_l^{(\infty)}(\boldsymbol{x}(k)) \leq Y_l. \tag{37}$$

Since $V^{*}(\boldsymbol{x}(k)) = \min\limits_{l}\left\{P_l^{(\infty)}(\boldsymbol{x}(k))\right\}$, for any $\epsilon > 0$, there exists an admissible control $\bar{\boldsymbol{u}}_K$ where K is a nonnegative number such that the associated performance index function satisfies $P_K^{(\infty)}(\boldsymbol{x}(k)) \leq V^{*}(\boldsymbol{x}(k))+\epsilon$. According to (23), we have $V^{(\infty)}(\boldsymbol{x}(k)) \leq P_l^{(\infty)}(\boldsymbol{x}(k))$ for any l. Thus we can obtain $V^{(\infty)}(\boldsymbol{x}(k)) \leq P_K^{(\infty)}(\boldsymbol{x}(k)) \leq V^{*}(\boldsymbol{x}(k)) + \epsilon$. Noting that ϵ is chosen arbitrarily, we have

$$V^{(\infty)}(\boldsymbol{x}(k)) \leq V^{*}(\boldsymbol{x}(k)). \tag{38}$$

On the other hand, since $V^{(i)}(\boldsymbol{x}(k))$ is bounded for $\forall i$, according to the definition of admissible control, the control sequence associated with the performance index function $V^{(\infty)}(\boldsymbol{x}(k))$ must be an admissible control, i.e., there exists an admissible control $\bar{\boldsymbol{u}}_N^{(i)}$ such that $V^{(\infty)}(\boldsymbol{x}(k)) = P_N^{(\infty)}(\boldsymbol{x}(k))$. Combining with the definition $V^*(\boldsymbol{x}(k)) = \min\limits_l \left\{P_l^{(\infty)}(\boldsymbol{x}(k))\right\}$, we can obtain

$$V^{(\infty)}(\boldsymbol{x}(k)) \geq V^*(\boldsymbol{x}(k)). \tag{39}$$

Therefore, combining (38) and (39), we can conclude that

$$V^{(\infty)}(\boldsymbol{x}(k)) = \lim_{i\to\infty} V^{(i)}(\boldsymbol{x}(k)) = V^*(\boldsymbol{x}(k)), \tag{40}$$

i.e., $V^*(\boldsymbol{x}(k))$ is the limit of the performance index function $V^{(i)}(\boldsymbol{x}(k))$, as $i \to \infty$. □

Based on Theorem 3, we will prove that the performance index function $V^*(\boldsymbol{x}(k))$ satisfies the principle of optimality, which shows that $V^{(i)}(\boldsymbol{x}(k))$ can reach the optimum as $i \to \infty$.

Theorem 4 *For any state vector $\boldsymbol{x}(k)$, the "optimal" performance index function $V^*(\boldsymbol{x}(k))$ satisfies $V^*(\boldsymbol{x}(k)) = \min\limits_{\boldsymbol{u}(k)}\{\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k)+2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)+\boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \boldsymbol{u}^{\mathrm{T}}(k)R_0u(k) + 2\boldsymbol{u}^{\mathrm{T}}(k)R_1M(k)u(k)$ $+\boldsymbol{u}^{\mathrm{T}}(k)M(k)R_2M(k)u(k)+V^*(\boldsymbol{x}(k+1))\}$ where $u(k-\tau) = M(k)u(k)$.*

Proof. For any $\boldsymbol{u}(k)$ and i, based on Bellman's optimality principle, we have

$$V^{(i)}(\boldsymbol{x}(k)) \leq \Upsilon^{(i-1)} + V^{(i-1)}(\boldsymbol{x}(k+1)), \tag{41}$$

where

$$\begin{aligned}\Upsilon^{(i-1)} =&\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+ \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma)\\ &+ \boldsymbol{u}^{\mathrm{T}}(k)R_0u(k) + 2\boldsymbol{u}^{\mathrm{T}}(k)R_1M^{(i-1)}(k)u(k)\\ &+ \boldsymbol{u}^{\mathrm{T}}(k)M^{(i-1)T}(k)R_2M^{(i-1)}(k)u(k).\end{aligned}$$

As $V^{(i)}(\boldsymbol{x}(k)) \leq V^{(i+1)}(\boldsymbol{x}(k)) \leq V^{(\infty)}(\boldsymbol{x}(k))$ and $V^{(\infty)}(\boldsymbol{x}(k)) = V^*(\boldsymbol{x}(k))$, we can obtain

$$V^{(i)}(\boldsymbol{x}(k)) \leq \Upsilon^{(i-1)} + V^*(\boldsymbol{x}(k+1)). \tag{42}$$

Let $i \to \infty$, we have

$$V^*(\boldsymbol{x}(k)) \leq \Upsilon^{(\infty)} + V^*(\boldsymbol{x}(k+1)). \tag{43}$$

Since $\boldsymbol{u}(k)$ in the above equation is chosen arbitrarily, the following equation holds

$$V^*(\boldsymbol{x}(k)) \leq \min_{\boldsymbol{u}(k)}\left\{\Upsilon^{(\infty)} + V^*(\boldsymbol{x}(k+1))\right\}. \tag{44}$$

On the other hand, for any i, the performance index function satisfies

$$V^{(i)}(\boldsymbol{x}(k)) = \Omega^{(i-1)} + V^{(i-1)}(\boldsymbol{x}(k+1)), \tag{45}$$

where

$$\begin{aligned}\Omega^{(i-1)} =&\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+ \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma)+\boldsymbol{u}^{(i-1)T}(k)R_0\boldsymbol{u}^{(i-1)}(k)\\ &+ 2\boldsymbol{u}^{(i)T}(k)R_1M^{(i-2)}(k)\boldsymbol{u}^{(i-1)T}(k)\\ &+ \boldsymbol{u}^{(i-1)T}(k)M^{(i-2)T}(k)R_2M^{(i-2)}(k)\boldsymbol{u}^{(i-1)T}(k).\end{aligned}$$

Combining with $V^{(i)}(\boldsymbol{x}(k)) \leq V^*(\boldsymbol{x}(k)), \forall i$, we have

$$V^*(\boldsymbol{x}(k)) \geq \Omega^{(i-1)} + V^{(i-1)}(\boldsymbol{x}(k+1)). \tag{46}$$

Let $i \to \infty$, and then

$$\begin{aligned}V^*(\boldsymbol{x}(k)) \geq& \lim_{i\to\infty}\left\{\Omega^{(i-1)} + V^{(i-1)}(\boldsymbol{x}(k+1))\right\}\\ \geq& \min_{\boldsymbol{u}(k)}\left\{\Omega^{(\infty)} + V^*(\boldsymbol{x}(k+1))\right\}.\end{aligned} \tag{47}$$

Combining (44) with (47), we have

$$\begin{aligned}V^*(\boldsymbol{x}(k)) =&\min_{\boldsymbol{u}(k)}\{\Omega^{(\infty)} + V^*(\boldsymbol{x}(k+1))\}\\ =&\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+ \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \boldsymbol{u}^{*\mathrm{T}}(k)R_0\boldsymbol{u}^*(k)\\ &+ 2\boldsymbol{u}^{*\mathrm{T}}(k)R_1M^{(\infty)}(k)\boldsymbol{u}^*(k)\\ &+ \boldsymbol{u}^{*\mathrm{T}}(k)M^{(\infty)T}(k)R_2M^{(\infty)}(k)\boldsymbol{u}^*(k)\\ &+ V^*(\boldsymbol{x}(k+1)).\end{aligned} \tag{48}$$

Thus we have that $\boldsymbol{u}^{(i)}(k) \to \boldsymbol{u}^*(k)$ as $i \to \infty$ so does $\boldsymbol{u}^{(i)}(k-\tau)$. On the other hand, we also have $M^{(i)}(k) \to M^{(\infty)}(k)$ and $\boldsymbol{u}^{(i)}(k-\tau) = M^{(i-1)}(k)\boldsymbol{u}^{(i)}(k)$. Let $i \to \infty$, we get

$$\boldsymbol{u}^*(k-\tau) = M^{(\infty)}(k)\boldsymbol{u}^*(k). \tag{49}$$

Therefore, we have $M^{(\infty)}(k) = M^*(k)$ and (48) can be written as

$$\begin{aligned}V^*(\boldsymbol{x}(k)) =& \boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma)\\ &+ \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \boldsymbol{u}^{*\mathrm{T}}(k)R_0\boldsymbol{u}^*(k)\\ &+ 2\boldsymbol{u}^{*\mathrm{T}}(k)R_1M^*(k)\boldsymbol{u}^*(k)\\ &+ \boldsymbol{u}^{*\mathrm{T}}(k)M^{*\mathrm{T}}(k)R_2M^*(k)\boldsymbol{u}^*(k)\\ &+ V^*(\boldsymbol{x}(k+1))\end{aligned} \tag{50}$$

where $\boldsymbol{u}^*(k-\tau) = M^*(k)\boldsymbol{u}^*(k)$. □

Therefore, we can conclude that the performance index function $V^{(i)}(\boldsymbol{x}(k))$ converges to the optimum $V^*(\boldsymbol{x}(k))$ as $i \to \infty$.

2.3 The Implementation of Iterative ADP Algorithm

Given the above preparation, we may formulate the desired iterative ADP approach for nonlinear systems with delays.

1. Give initial state $\boldsymbol{x}(s) = \phi(s)$, $s = -\sigma, -\sigma+1, \ldots, 0$, initial control $\boldsymbol{u}(\rho)$, $\rho = 0, 1, \ldots, k-1$; give $i_{\max}$, computation accuracy ε.

2. Set the iterative step $i = 0$, $M^{(0)}(k) = I$, $V^{(0)}(\cdot) = 0$.

3. Compute $\boldsymbol{u}^{(0)}(k)$ by (12) and the performance index function $V^{(1)}(\boldsymbol{x}(k))$ by (13).

4. For the iterative step $i \geq 1$, compute $\boldsymbol{u}^{(i)}(k)$ by (14).

5. Compute the performance index function $V^{(i)}(\boldsymbol{x}(k))$ by (15).

6. If

$$\left[V^{(i)}(\boldsymbol{x}(k)) - V^{(i-1)}(\boldsymbol{x}(k))\right]^2 < \varepsilon, \tag{51}$$

go to Step 9; otherwise, go to Step 7.

7. If $i > i_{\max}$, go to Step 9; otherwise, compute $M^{(i)}(k)$ by

$$M^{(i)}(k)=\boldsymbol{u}^{(i)}(k-\tau)\boldsymbol{u}^{(i)T}(k)\left(\boldsymbol{u}^{(i)}(k)\boldsymbol{u}^{(i)T}(k)\right)^{-1}. \tag{52}$$

8. Set $i = i + 1$ and go to Step 4.

9. Stop.

In (52) of the above algorithm, the term $\left(\boldsymbol{u}^{(i)}(k)\boldsymbol{u}^{(i)T}(k)\right)^{-1}$ can be obtained by the Moore-Penrose pseudoinverse technique to compute the delay matrix function $M^{(i)}(k)$. There are another two methods to compute $M^{(i)}(k)$. One choice is to introduce a small zero-mean Gaussian noise with variances γ^2 denoted by $\delta(0, \gamma^2)$ into the control $\boldsymbol{u}(k-\tau)$ (see [18], for detail).

The other choice is to use a neural network to approximate delay matrix function $M^{(i)}(k)$. In this paper, we use the neural network approximation method and the details will be shown in the next section.

3 Neural network implementation

In the case of linear systems the performance index function is quadratic and the control policy is linear. In the nonlinear case, this is not necessarily true and therefore we use neural networks to approximate $\boldsymbol{u}^{(i)}(k)$ and $V^{(i)}(\boldsymbol{x}(k))$.

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 NN is represented by:

$$\hat{F}(\boldsymbol{X}, V, W) = W^{\mathrm{T}}\sigma(V^{\mathrm{T}}\boldsymbol{X}) \tag{53}$$

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

The NN estimation error can be expressed by

$$F(\boldsymbol{X}) = F(\boldsymbol{X}, V^*, W^*) + \varepsilon(\boldsymbol{X}) \tag{54}$$

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

Here, there are four neural networks, which are critic network, model network, action network and delay matrix function network (M network) respectively. All the neural networks are chosen as three-layer feedforward network. The whole structure diagram is shown in Fig.1. The utility term in the figure denotes $\boldsymbol{x}^{\mathrm{T}}(k)Q_0\boldsymbol{x}(k) + 2\boldsymbol{x}^{\mathrm{T}}(k)Q_1\boldsymbol{x}(k-\sigma) + \boldsymbol{x}^{\mathrm{T}}(k-\sigma)Q_2\boldsymbol{x}(k-\sigma) + \boldsymbol{u}^{\mathrm{T}}(k)R_0\boldsymbol{u}(k) + 2\boldsymbol{u}^{\mathrm{T}}(k)R_1\boldsymbol{u}(k-\tau) + \boldsymbol{u}^{\mathrm{T}}(k-\tau)R_2\boldsymbol{u}(k-\tau)$.

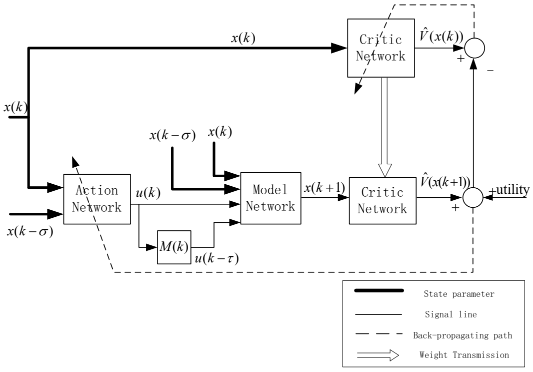

Fig. 1 The structure diagram of the algorithm

3.1 The Model Network

The model network is to approximate the system dynamic and it should be trained before the implementation of the iterative ADP algorithm. The update rule of the model network is adopted as gradient decent method. The training process is simple and general. The details can be seen in [13, 19] and it is omitted here.

After the model network is trained, its weights are kept unchanged.

3.2 the M network

The M network is to approximate the delay matrix function $M(k)$. The output of the M network is denoted as

$$\hat{\boldsymbol{u}}(k-\tau) = W_M^{\mathrm{T}}\sigma(V_M^{\mathrm{T}}\boldsymbol{u}(k)). \tag{55}$$

We define the error function of the model network as

$$e_M(k) = \hat{\boldsymbol{u}}(k-\tau) - \boldsymbol{u}(k-\tau). \tag{56}$$

Define the performance error measure as:

$$E_M(k) = \frac{1}{2}e_M^2(k). \tag{57}$$

Then the gradient-based weight updating rule for the critic network can be described by

$$w_M(k+1) = w_M(k) + \Delta w_M(k), \tag{58}$$

$$\Delta w_M(k) = \alpha_M\left[-\frac{\partial E_M(k)}{\partial w_M(k)}\right] \tag{59}$$

where α_M is the learning rate of the M network.

3.3 The critic network

The critic network is used to approximate the performance index function $V^{(i)}(\boldsymbol{x}(k))$. The output of the critic network is denoted as

$$\hat{V}^{(i)}(\boldsymbol{x}(k)) = W_{ci}^{\mathrm{T}}\sigma(V_{ci}^{\mathrm{T}}\boldsymbol{z}(k)). \tag{60}$$

The target function can be written as

$$V^{(i+1)}(\boldsymbol{x}(k)) = \Gamma^{(i)} + \hat{V}^{(i)}(\boldsymbol{x}(k+1)). \tag{61}$$

Then we define the error function for the critic network as

$$e_{ci}(k) = \hat{V}^{(i+1)}(\boldsymbol{x}(k)) - V^{(i+1)}(\boldsymbol{x}(k)). \tag{62}$$

And the objective function to be minimized in the critic network is

$$E_{ci}(k) = \frac{1}{2}e_{ci}^2(k). \tag{63}$$

So the gradient-based weight updating rule for the critic network is given by

$$w_{c(i+1)}(k) = w_{ci}(k) + \Delta w_{ci}(k), \tag{64}$$

$$\Delta w_{ci}(k) = \alpha_c \left[-\frac{\partial E_{ci}(k)}{\partial w_{ci}(k)} \right], \tag{65}$$

$$\frac{\partial E_{ci}(k)}{\partial w_{ci}(k)} = \frac{\partial E_{ci}(k)}{\partial \hat{V}^{(i)}(\boldsymbol{x}(k))} \frac{\partial \hat{V}^{(i)}(\boldsymbol{x}(k))}{\partial w_{ci}(k)} \tag{66}$$

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

3.4 The Action Network

In the action network the state $\boldsymbol{x}(k)$ is used as input to create the optimal control as the output of the network. The output can be formulated as

$$\hat{\boldsymbol{u}}^{(i)}(k) = W_{ai}^{\mathrm{T}} \sigma(V_{ai}^{\mathrm{T}} \boldsymbol{x}(k)). \tag{67}$$

And the target of the output of the action network is given by (14). So we can define the output error of the action network as

$$e_{ai}(k) = \hat{\boldsymbol{u}}^{(i)}(k) - \boldsymbol{u}^{(i)}(k) \tag{68}$$

where $\boldsymbol{u}^{(i)}(k)$ is the target function which can be described by

$$\begin{aligned} \boldsymbol{u}^{(i)}(k) = & -\frac{1}{2}\Big(R_0 + 2R_1 M^{(i-1)}(k) + M^{(i-1)T}(k) R_2 M^{(i-1)}(k)\Big)^{-1} \\ & \times \Big(g_0\left(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma)\right) \\ & + g_1\left(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma)\right) M^{(i-1)}(k)\Big)^T \frac{\partial \hat{V}^{(i)}(\boldsymbol{x}(k+1))}{\partial \boldsymbol{x}(k+1)}. \end{aligned}$$

As $\boldsymbol{u}^{(i)}(k-\tau) = M^{(i-1)}(k)\boldsymbol{u}^{(i)}(k)$, we have $\dfrac{\partial \boldsymbol{u}^{(i)}(k-\tau)}{\partial \boldsymbol{u}^{(i)}(k)} = M^{(i-1)}(k)$. Then according to (55), $M^{(i-1)}(k)$ can be expressed as

$$M_{\mathbb{i}\mathbb{j}}^{(i-1)}(k) = V_{M\mathbb{i}}^{\mathrm{T}} \left[1 - \Big(\sigma(V_M^{\mathrm{T}} \boldsymbol{u}(k)) \Big)_{\mathbb{i}}^2 \right] W_{M\mathbb{j}} \tag{69}$$

for $\mathbb{i}, \mathbb{j} = 1, 2, \ldots, m$. $M_{\mathbb{i}\mathbb{j}}^{(i-1)}(k)$ denotes the element of row $\mathbb{i}$, column $\mathbb{j}$ of matrix $M^{(i-1)}(k)$; $V_{M\mathbb{i}}$ and $W_{M\mathbb{j}}$ mean the column $\mathbb{i}$ and column $\mathbb{j}$ of the weight matrices V_M and W_M, respectively; $\big(\sigma(V_M^{\mathrm{T}}\boldsymbol{u}(k))\big)_{\mathbb{i}}$ is the $\mathbb{i}_{\mathtt{th}}$ element of the vector $\sigma(V_M^{\mathrm{T}}\boldsymbol{u}(k))$.

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

$$E_{ai}(k) = \frac{1}{2} e_{ai}^2(k). \tag{70}$$

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

$$w_{a(i+1)}(k) = w_{ai}(k) + \Delta w_{ai}(k), \tag{71}$$

$$\Delta w_{ai}(k) = \beta_a \left[-\frac{\partial E_{ai}(k)}{\partial w_{ai}(k)} \right], \tag{72}$$

$$\frac{\partial E_{ai}(k)}{\partial w_{ai}(k)} = \frac{\partial E_{ai}(k)}{\partial e_{ai}(k)} \frac{\partial e_{ai}(k)}{\partial \boldsymbol{u}^{(i)}(k)} \frac{\partial \boldsymbol{u}^{(i)}(k)}{\partial w_{ai}(k)} \tag{73}$$

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

4 Simulation

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

4.1 optimal control for state delayed system

For the first example, the nonlinear system is the modification of the example 1 in [13] which introduces state delays into the system.

Consider the following affine nonlinear system

$$\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma)) + g(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma))\boldsymbol{u}(k) \tag{74}$$

where $\boldsymbol{x}(k) = \big[\boldsymbol{x}_1(k)\ \boldsymbol{x}_2(k)\big]^{\mathrm{T}}$, $\boldsymbol{u}(k) = \big[u_1(k)\ u_2(k)\big]^{\mathrm{T}}$, and $f(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma)) = \begin{bmatrix} \boldsymbol{x}_1(k)\exp(\boldsymbol{x}_2^3(k))\boldsymbol{x}_2(k-2) \\ \boldsymbol{x}_2^3(k)\boldsymbol{x}_1(k-2) \end{bmatrix}$, $g(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma)) = \begin{bmatrix} -0.2 & 0 \\ 0 & -0.2 \end{bmatrix}$ The time delay in the state is $\sigma = 2$ and the initial condition is $\boldsymbol{x}(k) = [1\ {-1}]^{\mathrm{T}}$ for $-2 \le k \le 0$. The performance index function is defined as (2) where $Q_0 = Q_2 = R_0 = I$ and $Q_1 = R_1 = R_2 = 0$.

And we implement the algorithm at the time instant $k = 5$. We choose three-layer neural networks as the critic network, the action network and the model network with the structure 4-10-2, 2-10-1 and 6-10-2 respectively. The initial weights of action network, critic network and model network are all set to be random in $[-0.5, 0.5]$. It should be mentioned that the model network should be trained first. For the given initial state, we train the model network for 3000 steps under the learning rate $\alpha_m = 0.05$. After the training of the model network completed, the weights keep unchanged. Then the critic network and the action network are trained for 3000 steps so that the given accuracy $\varepsilon = 10^{-6}$ is reached. In the training process, the learning rate $\beta_a = \alpha_c = 0.05$. The convergence curve of the performance index function is shown in Fig.2. 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.3 and the corresponding control curves are given as Fig.4.

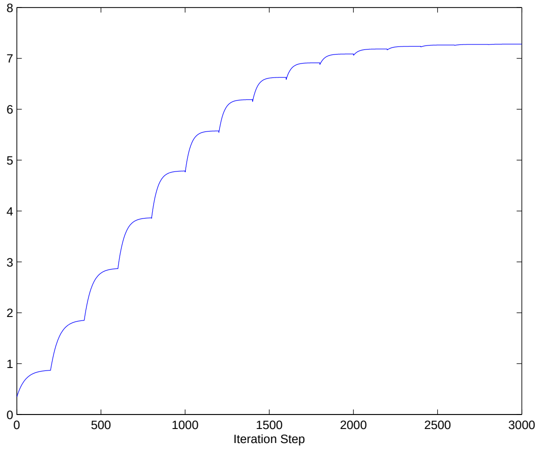

Fig. 2 The convergence of performance index function

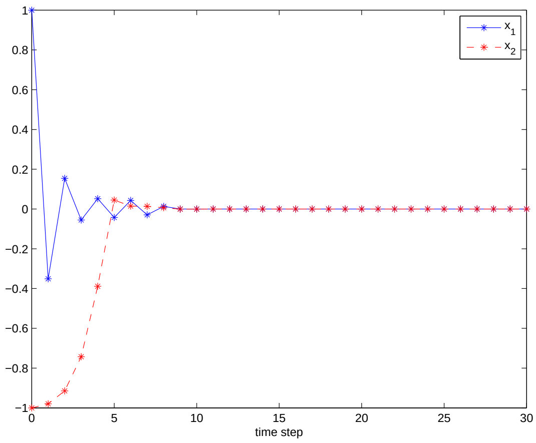

Fig. 3 The state variables trajectories

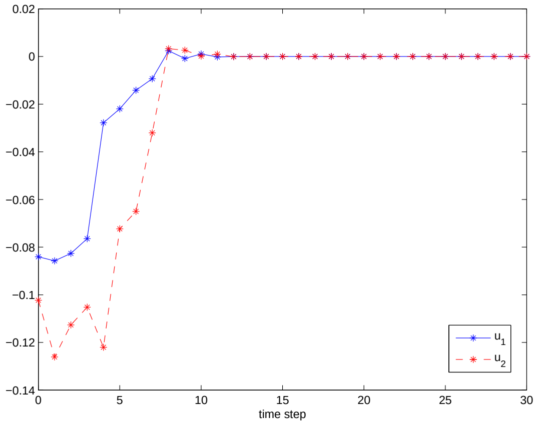

Fig. 4 The optimal control trajectories

4.2 optimal control for nonlinear system with state and control delays

For the second example, the control time delay is added into the system of example 1 and the system becomes

$$\begin{aligned}\boldsymbol{x}(k+1) =& f(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma)) + g_0(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma))\boldsymbol{u}(k)\\ &+ g_1(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma))\boldsymbol{u}(k-\tau) \qquad (75)\end{aligned}$$

where $\boldsymbol{x}(k) = \begin{bmatrix}\boldsymbol{x}_1(k) \ \boldsymbol{x}_2(k)\end{bmatrix}^{\mathrm{T}}$, $\boldsymbol{u}(k) = \begin{bmatrix}u_1(k) \ u_2(k)\end{bmatrix}^{\mathrm{T}}$, and $f(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma))$ is the same to example 1, $g_0(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma)) = g_1(\boldsymbol{x}(k), \boldsymbol{x}(k-\sigma)) = \begin{bmatrix}-0.2 & 0\\ 0 & -0.2\end{bmatrix}$. The state time delay $\sigma = 2$ and the control time delay $\tau = 1$. The initial condition is $\boldsymbol{x}(k) = [-1 \, -1]^{\mathrm{T}}$ and $\boldsymbol{u}(k) = 0$ for $-2 \leq k \leq 0$. The performance index function is defined as (2) where $Q_0 = Q_2 = R_0 = R_2 = I$ and $Q_1 = R_1 = 0$.

We also implement the algorithm at the time instant $k = 5$. We choose three-layer neural networks as the critic network, the action network, the model network and the M network with the structure 4-10-2, 2-10-1, 8-10-2 and 2-8-2 respectively. All th other parameters are set the same as example 1. The initial weights of action network, critic network, model network and the M network are all set to be random in $[-0.5, 0.5]$. For the given initial state, we train the model network for 4000 steps. After the training of the model network completed, the weights keep unchanged. Then the critic network, the action network and the M network are trained for 3000 steps to reach the given accuracy $\varepsilon = 10^{-6}$. The convergence curve of the performance index function is shown in Fig.5. 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.6 and the corresponding control curves are given as Fig.7.

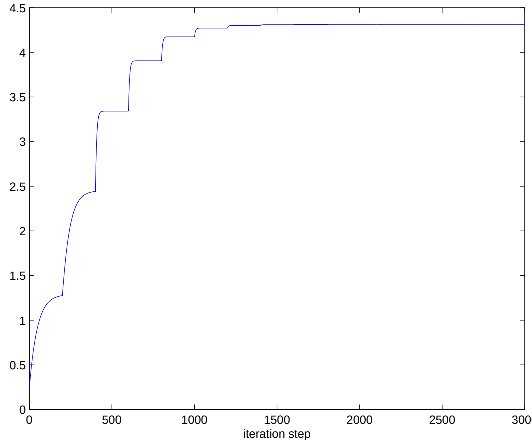

Fig. 5 The convergence of performance index function

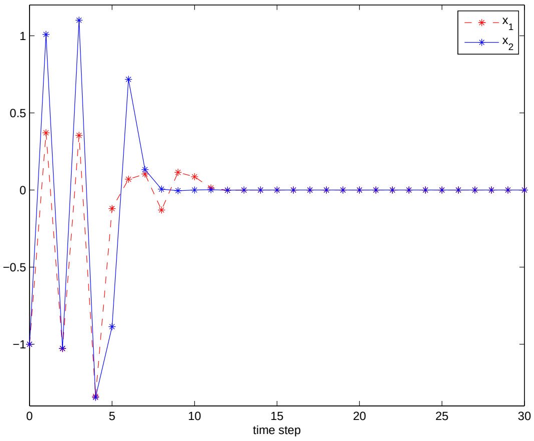

Fig. 6 The state variables trajectories

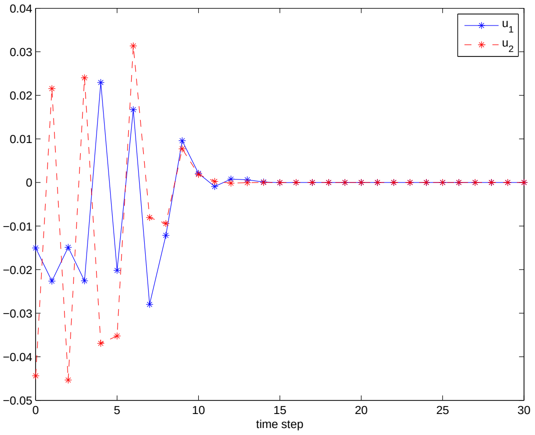

Fig. 7 The optimal control trajectories

5 Conclusion

In this paper, we propose an effective algorithm to find the optimal infinite-time controller for a class of discrete-time nonlinear systems with time delays in state and control variables. Introducing a delay matrix function, the explicit expression of the optimal control is obtained. Then the iterative ADP algorithm is implemented to deal with the time delay problem with rigorous convergence analysis. Four neural networks are used as parametric structures to approximate the performance index function, compute the optimal control policy, model the unknown system and solve delay matrix function respectively, i.e. the critic network, the action network, the model network and the M network. The simulation studies have successfully demonstrated the upstanding performance of the proposed time-delay optimal control scheme for various discrete-time nonlinear systems.

6 Appendix

Lemma 3 *If* $\begin{bmatrix} R_0 & R_1 \\ R_1^{\mathrm{T}} & R_2 \end{bmatrix}$ *is a positive definite matrix where* $R_0, R_1, R_2 \in \Re^{n\times n}$, *then for any nonsingular matrix* $M \in \Re^{n\times n}$, $\begin{bmatrix} R_0 & R_1 M \\ M^{\mathrm{T}} R_1^{\mathrm{T}} & M^{\mathrm{T}} R_2 M \end{bmatrix} > 0$.

Proof. Since $\begin{bmatrix} R_0 & R_1 \\ R_1^{\mathrm{T}} & R_2 \end{bmatrix}$ is positive definite matrix, according to Schur complement[20], we have

$$R_2 - R_1^{\mathrm{T}} R_0^{-1} R_1 > 0. \tag{76}$$

As the matrix $M \in \Re^{n\times n}$ is nonsingular, let M^{-1} denote the inverse matrix of M, and then (76) can be written as

$$R_2 - R_1^{\mathrm{T}} M^{\mathrm{T}} M^{-\mathrm{T}} R_0^{-1} M^{-1} M R_1 > 0 \tag{77}$$

where $M^{-\mathrm{T}} = (M^{-1})^{\mathrm{T}}$. Again, using Schur complement, we can obtain

$$\begin{bmatrix} R_0 & R_1 M \\ M^{\mathrm{T}} R_1^{\mathrm{T}} & M^{\mathrm{T}} R_2 M \end{bmatrix} > 0. \tag{78}$$

□

References

1 Zhang H G, Yang D D, Chai T Y. Guaranteed cost networked control for T-S fuzzy systems with time delay. *IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews*, 2007, **37**(2): 160–172

2 Kong S L, Zhang H S, Zhang Z S, Zhang C H. Joint Predictive Control of Power and Rate for Wireless Networks. *Acta Automatica Sinica*, 2007, **33**(7): 761–764

3 Malek-Zavarei M, Jashmidi M. *Time-Delay Systems: Analysis, Optimization and Applications*. North-Holland, Amsterdam: The Netherlands, 1987. 80–96

4 Basin M, Rodriguez-Gonzalez J. Optimal control for linear systems with multiple time delays in control input. *IEEE Transactions on Automatic Control*, 2006, **51**(1): 91–97

5 Richard J P. Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 2003, **39**(10): 1667–1694

6 Wu Z G, Zhou W N. Delay-dependent Robust Stabilization for Uncertain Singular Systems with State Delay. *Acta Automatica Sinica*, 2007, **33**(7): 714–718

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

8 Hou Z G,Wu C P. A dynamic programming neural network for large-scale optimization problems.*Acta Automatica Sinica*, 2005, **25**(1): 46–51

9 Bellman R E. *Dynamic Programming*, Princeton, New Jersey: Princeton University Press, 1957. 150–155

10 Tang H, Yuan J B, Lu Y, Cheng W J. Performance potential-based neuro-dynamic programming for SMDPs. *Acta Automatica Sinica*, 2005, **31**(4): 642–645

11 Werbos P J. Approximate dynamic programming for real-time control and neural modeling. *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches,* D.A. White and D.A. Sofge, Ed., New York: Van Nostrand Reinhold, 1992, chapter 13

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

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

14 Liu D, Javaherian H, Kovalenko O, Huang T. Adaptive critic learning techniques for engine torque and airCfuel ratio control. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 2008, **38**(4): 988–993

15 Liu D R. Approximate dynamic programming for self-learning control. *Acta Automatica Sinica*, 2005, 31(1): 13–18

16 Ray S, Venayagamoorthy G K, Chaudhuri B, Majumder R. Comparison of adaptive critic-based and classical wide-area controllers for power systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 2008, **38**(4): 1002–1007

17 Al-Tamimi A, Lewis F L, Abu-Khalaf M. Discrete-time nonlinear HJB solution using approximate dynamic programming– convergence proof. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 2008, **38**(4): 943–949

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

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

20 Boyd S P, Ghaoui L E, Feron E, Balakrishnam V. Linear Matrix Inequalities in System and Control Theory. Philadelphia, PA: SIAM, 1994

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

ZHANG Hua-Guang Received his Ph. D. degree from Southeast University, in 1991. He is currently as a Full Professor and Ph. D. advisor. He was nominated as the Changjiang Scholar by China Education Ministry in 2005, and was awarded National Excellent Postdoctoral Research Fellow Award in 2005, China. His current research interests are Fuzzy System Theory, Fuzzy Control, Neural Network-Based Control, Adaptive Control, Chaotic Control, Complex Industry Process Automation, Electric Power System Automation, Motor Driving System Automation.

LIU De-Rong received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1994. In 1999, he joined the University of Illinois at Chicago, Chicago, where he is currently a Full Professor of electrical and computer engineering and of computer science and, since 2005, has been the Director of Graduate Studies in the Department of Electrical and Computer Engineering. In 2008, he was selected into the "100 Talents Program" by the Chinese Academy of Sciences, Beijing.

ZHAO Yan received the PH. D. degree from Northeastern University in 2008. He is a lecturer in Department of Automatic Control Engineering, Shenyang Institute of Engineering. His research interests are intelligent control and applications.