Optimal Tracking Control for a Class of Nonlinear Discrete-Time Systems with Time Delays Based on Heuristic Dynamic Programming

Huaguang Zhang, *Senior Member, IEEE*, Ruizhuo Song, *Member, IEEE*, Qinglai Wei, *Member, IEEE*, and Tieyan Zhang, *Member, IEEE*

***Abstract*— In this paper, a novel heuristic dynamic programming (HDP) iteration algorithm is proposed to solve the optimal tracking control problem for a class of nonlinear discrete-time systems with time delays. The novel algorithm contains state updating, control policy iteration, and performance index iteration. To get the optimal states, the states are also updated. Furthermore, the "backward iteration" is applied to state updating. Two neural networks are used to approximate the performance index function and compute the optimal control policy for facilitating the implementation of HDP iteration algorithm. At last, we present two examples to demonstrate the effectiveness of the proposed HDP iteration algorithm.**

***Index Terms*— Adaptive dynamic programming, approximate dynamic programming, heuristic dynamic programming iteration, optimal control, time delays.**

I. Introduction

TIME delay often occurs in the transmission between different parts of systems. Transportation systems, communication systems, chemical processing systems, metallurgical processing systems, and power systems are examples of time delay systems [1]. So the investigation of time delay systems is significant [2]–[7]. In addition, the tracking control problem is often encountered in industrial production. It has been the focus of many researchers for many years [8]–[11]. In [12], the optimal tracking control problem was studied based on dynamic output feedback for linear systems with time delays in state and control. The optimal output tracking control problem was discussed for a class of nonlinear systems with time delay in [13], and the optimal output tracking control problem was transformed into a sequence of linear inhomogeneous two-point boundary value problems including adjoint vector differential equations. In [14], multilayer neural networks were used to design an optimal tracking neurocontroller for discrete-time nonlinear dynamic systems with quadratic cost function. However, the optimal tracking control problem for nonlinear discrete-time systems with time delays through the framework of Hamilton–Jacobi–Bellman (HJB) equation is still very difficult to be handled.

As is well known, dynamic programming is very useful in solving the optimal control problems. However, due to the "curse of dimensionality" of dynamic programming [15], [16], adaptive/approximate dynamic programming (ADP) algorithms have been paid much attention in order to obtain approximate solutions of the HJB equation effectively. ADP was proposed by Werbos for discrete-time dynamical systems [17]. In recent years, ADP algorithms were further developed by Lewis [18]–[20], Powell [21], Jagannathan [22], [23], Murray [24], Si [25]–[27], Liu [28], [29], and so on. In [30], Werbos classified ADP approaches into four main schemes: heuristic dynamic programming (HDP), dual heuristic dynamic programming (DHP), action-dependent HDP, and action-dependent DHP. HDP is the most basic and widely applied structure of ADP [31], [32]. Moreover, some recent research trends within the field of ADP and optimal feedback control based on ADP were introduced in [33] and [34]. In [35], Bradtke, Ydestie, and Barto presented stability and convergence results for dynamic programming-based reinforcement learning applied to linear quadratic regulation. Furthermore, a greedy iteration HDP scheme with convergence proof was proposed for solving the optimal control problem for nonlinear discrete-time systems in [36]. In [37], a new type of performance index of optimal tracking problem was defined, and the infinite-time optimal tracking control problem for a class of discrete-time nonlinear systems using HDP iteration algorithm was solved. Additionally, the near-optimal control problem for a class of nonlinear discrete-time systems with control constraints was solved by iteration ADP algorithm in [38].

In spite of significant progress on HDP algorithm in the optimal control field, within the radius of our knowledge, it is still an open problem about how to solve the optimal tracking control problem for nonlinear systems with time delays based

Manuscript received June 29, 2010; revised September 27, 2011; accepted October 5, 2011. Date of publication November 1, 2011; date of current version December 1, 2011. This work was supported in part by the National Natural Science Foundation of China under Grant 50977008, Grant 60821063, Grant 61034005, and Grant 60972164, and the National Basic Research Program of China under Grant 2009CB320601.

H. Zhang is with the School of Information Science and Engineering, Northeastern University, Shenyang 110004, China. He is also with the Key Laboratory of Integrated Automation of Process Industry (Northeastern University) of the National Education Ministry, Shenyang 110004, China (e-mail: hgzhang@ieee.org).

R. Song is with the School of Information Science and Engineering, Northeastern University, Shenyang 110004, China (e-mail: ruizhuosong@163.com).

Q. Wei is with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: qinglaiwei@gmail.com).

T. Zhang is with the Department of Electrical Engineering, Shenyang Institute of Engineering, Shenyang 110136, China (e-mail: zty@sie.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2172628

on HDP algorithm. In this paper, this open problem will be explicitly figured out. First, the HJB equation for discrete time delay system is derived which is based on state error and control error. In order to solve this HJB equation, a novel iteration HDP algorithm containing state updating, control policy iteration, and performance index function iteration is proposed. We also give the convergence proof for the novel iteration HDP algorithm. At last, two neural networks are used, the critic neural network and the action neural network are devoted to approximate the performance index function and the corresponding control policy, respectively. The main contributions of this paper can be summarized as follows.

1) It is the first time to solve the optimal tracking control problem for nonlinear systems with time delays using HDP algorithm.
2) For the novel HDP algorithm, in order to get the optimal states, the state is also updated according to the control policy.
3) In the state update, we adopt "backward iteration." For the $(i+1)^{th}$ performance index function iteration, the state at time step $k+1$ is updated according to the states before time step $k+1$ in the i^{th} iteration and the control policy in the i^{th} iteration.

This paper is organized as follows. In Section II, we present the problem formulation. In Section III, the optimal tracking control scheme is developed based on iteration HDP algorithm and the convergence proof is given. In Section IV, the neural network implementation for the tracking control scheme is discussed. In Section V, two examples are given to demonstrate the effectiveness of the proposed control scheme. In Section VI, the conclusion is drawn.

II. Problem Formulation

Consider a class of discrete-time affine nonlinear systems with time delays

$$\begin{aligned} x(k+1) &= f(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m)) \\ &\quad + g(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m))u(k) \\ x(k) &= \varepsilon_1(k), \quad -\sigma_m \le k \le 0 \end{aligned} \tag{1}$$

where $x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m) \in \Re^n$, and $x(k-\sigma_1), \ldots, x(k-\sigma_m)$ are states of time delays. $f(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m)) \in \Re^n$, $g(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m)) \in \Re^{n\times m}$ and the input $u(k) \in \Re^m$. $\varepsilon_1(k)$ is the initial state, σ_i is the time delay, set $0 = \sigma_0 < \sigma_1 < \cdots < \sigma_m$, and they are nonnegative integer numbers. $f(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m))$, and $g(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m))$ are known functions, and $g(x(k-\sigma_0), x(k-\sigma_1), \ldots, x(k-\sigma_m))$ is analytic. System (1) is controllable and reachable in $\Omega \in \Re^n$ [39].

Definition 1 (Controllability) [40]: System (1) is controllable from $(x(k_0), k_0)$ to $(0, k_f)$, if for some control $u(k)$, $k_0 \le k \le k_f$, the solution of (1) with $x(k_0)$ is such that $x(k_f) = 0$, where k_f is terminal time.

Definition 2 (Reachability) [41]: A state $x(k_f)$ is reachable at time $k_f, k_f > 0$, if there exists control $u(k), k_0 \le k \le k_f$ that leads the state trajectory from $x(k_0) = 0$ to $x(k_f) = x_f$.

In this paper, we define the state error as follows:

$$e(k) = x(k) - \eta(k) \tag{2}$$

where the reference orbit $\eta(k)$ is generated by the n-dimensional autonomous system as follows:

$$\begin{aligned} \eta(k+1) &= S(\eta(k)) \\ \eta(0) &= \varepsilon_2(k), \qquad -\sigma_m \le k \le 0 \end{aligned} \tag{3}$$

in which $\eta(k) \in \Re^n$, $S(\eta(k)) \in \Re^n$ and ε_2 is the initial state, $\varepsilon_2(-\sigma_m) = \cdots = \varepsilon_2(-\sigma_1) = 0$.

Noticing that the objective in this paper is to design an optimal state feedback control policy $u(k)$ based on any given $\varepsilon_1(k)(-\sigma_m \le k \le 0)$ and initial control policy $\beta(k)$, which not only renders the state $x(k)$ asymptotically tracking the reference orbit, i.e., $e(k)$ asymptotically approaches zero, but also minimizes the performance index function as follows:

$$J(e(k), v(k)) = \sum_{t=k}^{\infty} \left\{ e^T(t) Q e(t) + v^T(t) R v(t) \right\} \tag{4}$$

where Q and R are symmetric and positive-definite matrices. We divide $u(k)$ into two parts, i.e., $v(k)$ and $u_s(k)$. So we have

$$v(k) = u(k) - u_s(k) \tag{5}$$

where $u_s(k)$ denotes the steady control input corresponding to the desired trajectory $\eta(k)$. In fact, $v(k)$ is the error between actual control $u(k)$ of (1) and the steady control $u_s(k)$.

We can see that the problem of solving the optimal tracking control policy $u(k)$ of (1) is converted into solving the optimal control policy $v(k)$. In the following part, we will discuss how to design $v(k)$.

III. Iteration HDP Algorithm and Its Convergence

In this section, we focus on designing the optimal control policy to handle the optimal tracking problem. We first give the representation of the steady control input $u_s(k)$. Inspired by paper [37], we define the steady control as follows:

$$\begin{aligned} u_s(k) &= g^{-1}(\eta(k-\sigma_0), \ldots, \eta(k-\sigma_m)) \\ &\quad \times (\eta(k+1) - f(\eta(k-\sigma_0), \ldots, \eta(k-\sigma_m))) \end{aligned} \tag{6}$$

where $g^{-1}(\cdot)$ denotes the inversion of $g(\cdot)$.

Remark 1: If $g(\cdot)$ is invertible, then u_s can be obtained directly by (6). So u_s exists, when $g(\cdot)$ is invertible. If $g(\cdot)$ is noninvertible, u_s also exists. Because of the numerical solution of $g^{-1}(\cdot)$ can be solved at least by one of the three methods as follows.

1) Moore-Penrose Pseudoinverse Technique [42]: The Moore-Penrose pseudoinverse is a matrix G of the same dimensions as g satisfying four conditions: $gGg = g$, $GgG = G$, Gg is Hermitian, gG is Hermitian.

In [42], it is proved that the matrix G exists and unique, if $g \neq 0$. In this paper, $g \neq 0$ obviously, because we supposed that (1) is controllable and reachable. So we can say that $g^{-1} = G$. Furthermore, in MATLAB 7.5, Moore-Penrose

pseudoinverse of g can be obtained by the MATLAB function $G = pinv(g)$.

2) Least Square Method [43]: As $g(\cdot)g^{-1}(\cdot) = I$, we have

$$g^{-1}(\cdot) = \left(g^T(\cdot)g(\cdot)\right)^{-1} g^T(\cdot). \tag{7}$$

Introducing $E(0, r^2) \in \Re^{n\times m}$ into $g(\cdot)$, then $\bar{g}(\cdot)$ can be expressed as follows:

$$\bar{g}(\cdot) = g(\cdot) + E\left(0, r^2\right) \tag{8}$$

where every element in $E(0, r^2)$ is zero-mean Gaussian noise. So we can get

$$g^{-1}(\cdot) = \left(\bar{g}^T(\cdot)\bar{g}(\cdot)\right)^{-1} g^{\mathrm{T}}(\cdot). \tag{9}$$

Then we sample K times, every time we take nm Gaussian points, so we have $E_i(0, r^2)$, $i \in \{1, 2, \ldots, K\}$, every element in E_i is a Gaussian sample point. Thus, we have $G(\cdot) = [\bar{g}_1, \ldots, \bar{g}_K]$, where $\bar{g}_i = g + E_i(0, r^2)$, $i \in \{1, 2, \ldots, K\}$.

So we can get

$$g^{-1}(\cdot) = \left(G^T(\cdot)G(\cdot)\right)^{-1} g^T(\cdot). \tag{10}$$

3) Neural Network Method [44]: First, we let the back-propagation (BP) neural network be expressed as $\hat{F}(X, V, W) = W^T\sigma(V^T X)$, where W and V are the weights of the neural network and σ is the activation function. Second, we use the output $\hat{x}(k+1)$ of BP neural network to approximate $x(k+1)$. Then we have $\hat{x}(k+1) = W^T\sigma(V^T X)$, where $X = [x(k-\sigma_0), \ldots, x(k-\sigma_m), u(k)]$. We have known that (1) is an affine nonlinear system. So we can get $g = (\partial\hat{x}(k+1)/\partial u)$. The equation $g^{-1} = (\partial u/\partial\hat{x}(k+1))$ can be established.

So we can see that u_s is existent and can be obtained by (6). In this paper, we adopt Moore-Penrose pseudoinverse technique to get $g^{-1}(\cdot)$ in simulation part.

According to (2) and (3), we can easily obtain the following system:

$$\begin{aligned} e(k+1) = {} & f(e(k-\sigma_0) + \eta(k-\sigma_0), \ldots, e(k-\sigma_m) \\ & + \eta(k-\sigma_m)) + g(e(k-\sigma_0) + \eta(k-\sigma_0), \\ & \ldots, e(k-\sigma_m) + \eta(k-\sigma_m)) \\ & \times (g^{-1}(\eta(k-\sigma_0), \ldots, \eta(k-\sigma_m)) \\ & \times (S(\eta(k)) - f(\eta(k-\sigma_0), \ldots, \eta(k-\sigma_m))) \\ & + \upsilon(k)) - S(\eta(k)), \\ e(k) = {} & \varepsilon(k), \qquad -\sigma_m \le k \le 0 \end{aligned} \tag{11}$$

where $\varepsilon(k) = \varepsilon_1(k) - \varepsilon_2(k)$.

So the aim for this paper is changed to get an optimal control policy not only making (11) asymptotically stable but also making the performance index function (4) minimal. To solve the optimal tracking problem in this paper, the following definition and assumption are required.

Definition 3 (Asymptotic Stability) [1]: An equilibrium state $e = 0$ for (11) is asymptotically stable if:

1) it is stable, i.e., given any positive numbers k_0 and ϵ, there exists $\delta > 0$, such that every solution of (11) satisfies $\max_{k_0\le k\le k_0+\sigma_m} |e(k)| \le \delta$ and $\max_{k_0\le k\le\infty} |e(k)| \le \epsilon$;
2) for each $k_0 > 0$ there is a $\delta > 0$ such that every solution of (11) satisfies $\max_{k_0\le k\le k_0+\sigma_m} |e(k)| \le \delta$ and $\lim_{k\to\infty} e(k) = 0$.

Assumption 1: Given (11), for the infinite-time horizon problem, there exists a control policy $\upsilon(k)$, which satisfies:

1) $\upsilon(k)$ is continuous on Ω, if $e(k-\sigma_0) = e(k-\sigma_1) = \cdots = e(k-\sigma_m) = 0$, then $\upsilon(k) = 0$;
2) $\upsilon(k)$ stabilizes (11);
3) $\forall e(-\sigma_0), e(-\sigma_1), e(-\sigma_m) \in \Re^n$, $J(e(0), \upsilon(0))$ is finite.

Actually, for nonlinear systems without time delays, if $\upsilon(k)$ satisfies Assumption 1, then we can say that $\upsilon(k)$ is an admissible control. The definition of admissible control can be seen in [36] and [45].

In the following section we focus on the design of $\upsilon(k)$.

A. Derivation of the Iteration HDP Algorithm

In (11), for time step k, $J^*(e(k))$ is used to denote the optimal performance index function, i.e., $J^*(e(k)) = \inf_{\upsilon(k)} J(e(k), \upsilon(k))$, and $\upsilon^*(k) = arg\inf_{\upsilon(k)} J(e(k), \upsilon(k))$. $u^*(k) = \upsilon^*(k) + u_s(k)$ is the optimal control for (1). Let $e^*(k) = x^*(k) - \eta(k)$, where $x^*(k)$ is used to denote the state under the action of the optimal tracking control policy $u^*(k)$.

According to Bellman's principle of optimality [1], $J^*(e(k))$ should satisfy the following HJB equation:

$$\begin{aligned} J^*(e(k)) = \inf_{\upsilon(k)} \Big\{ & e^T(k)Qe(k) + \upsilon^T(k)R\upsilon(k) \\ & + J^*(e(k+1)) \Big\} \end{aligned} \tag{12}$$

the optimal controller $\upsilon^*(k)$ should satisfy

$$\begin{aligned} \upsilon^*(k) = \arg\inf_{\upsilon(k)} \Big\{ & e^T(k)Qe(k) + \upsilon^T(k)R\upsilon(k) \\ & + J^*(e(k+1)) \Big\}. \end{aligned} \tag{13}$$

Here we define $e^*(k) = x^*(k) - \eta(k)$ and

$$\begin{aligned} x^*(k+1) = {} & f(x^*(k-\sigma_0), \ldots, x^*(k-\sigma_m)) \\ & + g(x^*(k-\sigma_0), \ldots, x^*(k-\sigma_m)) \\ & \times u^*(k), \quad k = 0, 1, 2, \ldots, \\ x^*(k) = {} & \varepsilon_1(k) \quad k = -\sigma_m, -\sigma_{m-1}, \ldots, -\sigma_0. \end{aligned} \tag{14}$$

Then the HJB equation is written as follows:

$$\begin{aligned} J^*(e^*(k)) = {} & (e^*(k))^T Qe^*(k) + (\upsilon^*(k))^T R\upsilon^*(k) \\ & + J^*(e^*(k+1)). \end{aligned} \tag{15}$$

Remark 2: Of course, one can reduce (1) to a system without time delay by defining a new $(\sigma_m + 1)n$-dimensional state vector $y(k) = (x(k), x(k-1), \ldots, x(k-\sigma_m))$. However, Chyung has pointed out that there are two major disadvantages of this method in [46]. First, the resulting new system is a $(\sigma_m + 1)n$-dimensional system increasing the dimension of the system by $(\sigma_m + 1)$ fold. Second, the new system may not be controllable even if the original system is controllable. This causes the set of attainability to have an empty interior which, in turn, introduces additional difficulties [46]–[48].

The following example is used to explain that the controllable condition of the expanded system is not the same as the original time delay system.

Example: We consider the following time delay system:

$$x(k+1) = A_0 x(k) + A_1 x(k-1) + Bu(k) \tag{16}$$

where $A_0 = \begin{bmatrix} 1.4 & 1.7 \\ 1.1 & -1 \end{bmatrix}$, $A_1 = \begin{bmatrix} 0 & -1 \\ -0.5 & 2 \end{bmatrix}$, and $B = \begin{bmatrix} 0.9 \\ 0.3 \end{bmatrix}$.

According to the controllable criterion of linear time delay system in [1], [49], and [50], we have

$$rank[B \quad G_1 B \quad G_2 B \quad G_3 B] = 2 \tag{17}$$

where $G_1 B = A_0 B$, $G_2 B = (A_0^2 + A_1)B$, and $G_3 B = (A_0^3 + A_1 A_0 + A_0 A_1)B$.

So the time delay system (16) is controllable. After that, we discuss the controllability of the expanded system.

We let $z(k) = x(k-1)$, (16) can be rewritten as follows:

$$\begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} A_0 & A_1 \\ I & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k). \tag{18}$$

Let $\bar{A} = \begin{bmatrix} A_0 & A_1 \\ I & 0 \end{bmatrix}$, $\bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$. According to the controllability rank criterion in [51], we calculate the rank of the controllability matrix

$$rank\left[\bar{B} \quad \bar{A}\bar{B} \quad \cdots \quad \bar{A}^{2n-1}\bar{B}\right] = 3 \neq 4. \tag{19}$$

So the expanded system is not controllable.

From the example, we can see that the expanded system (18) is not controllable, even if the original time delay system is controllable.

Therefore, in the following part, a direct method for time delay systems is proposed to get the optimal tracking control policy. For purposes of analysis, inspired by [52], the following system is used to design the iteration algorithm:

$$\begin{cases} e(k+1) = f(e(k-\sigma_0) + \eta(k-\sigma_0), \ldots, e(k-\sigma_m) \\ \quad + \eta(k-\sigma_m)) \\ \quad + g(e(k-\sigma_0) + \eta(k-\sigma_0), \ldots, e(k-\sigma_m) \\ \quad + \eta(k-\sigma_m))(g^{-1}(\eta(k-\sigma_0), \ldots, \eta(k-\sigma_m)) \\ \quad \times (S(\eta(k)) - f(\eta(k-\sigma_0), \ldots, \eta(k-\sigma_m))) \\ \quad + \upsilon(k)) - S(\eta(k)), \\ \eta(k+1) = S(\eta(k)). \end{cases} \tag{20}$$

The detailed iteration process is as follows.

First, we start with the initial performance index function $J^{[0]}(\cdot) = 0$ which is not necessary to be the optimal performance index function. Then for any given state $\varepsilon_1(k)(-\sigma_m \leq k \leq 0)$, and initial control $\beta(k)$ in (1), at any current time k, we start the iteration algorithm from $i = 0$ to find the control policy $v^{[0]}(k)$ as follows:

$$v^{[0]}(k) = \arg\inf_{\upsilon(k)} \left\{ e^T(k) Q e(k) + \upsilon^T(k) R \upsilon(k) \right\} \tag{21}$$

and the performance index function is updated as follows:

$$J^{[1]}(e(k)) = \inf_{\upsilon(k)} \left\{ e^T(k) Q e(k) + \upsilon^T(k) R \upsilon(k) \right\} \tag{22}$$

where $e(k) = x(k) - \eta(k)$ and

$$\begin{aligned} x(t+1) &= f(x(t-\sigma_0), \ldots, x(t-\sigma_m)) \\ &\quad + g(x(t-\sigma_0), \ldots, x(t-\sigma_m))\beta(t) \quad t > 0 \\ x(t) &= \varepsilon_1(t) \quad -\sigma_m \leq t \leq 0. \end{aligned} \tag{23}$$

We notice that the states are generated by the designed controller, so we further get the performance index function iteration as follows:

$$J^{[1]}\left(e^{[0]}(k)\right) = \left(e^{[0]}(k)\right)^T Q e^{[0]}(k) + \left(\upsilon^{[0]}(k)\right)^T R \upsilon^{[0]}(k). \tag{24}$$

For $i = 1, 2, \ldots$, the HDP iteration algorithm iterates as follows:

$$\begin{aligned} \upsilon^{[i]}(k) = \arg\inf_{\upsilon(k)} \Big\{ & e^T(k) Q e(k) + \upsilon^T(k) R \upsilon(k) \\ & + J^{[i]}(e(k+1)) \Big\} \end{aligned} \tag{25}$$

and

$$\begin{aligned} J^{[i+1]}(e(k)) = \inf_{\upsilon(k)} \Big\{ & e^T(k) Q e(k) + \upsilon^T(k) R \upsilon(k) \\ & + J^{[i]}(e(k+1)) \Big\}. \end{aligned} \tag{26}$$

Notice that, we emphasize the states in each iteration are regulated by the designed control policy, so we further obtain

$$\begin{aligned} J^{[i+1]}\left(e^{[i]}(k)\right) = & \left(e^{[i]}(k)\right)^T Q e^{[i]}(k) + \left(\upsilon^{[i]}(k)\right)^T R \upsilon^{[i]}(k) \\ & + J^{[i]}\left(e^{[i-1]}(k+1)\right) \end{aligned} \tag{27}$$

where $e^{[i]}(k) = x^{[i]}(k) - \eta(k)$, $i = 0, 1, 2, \ldots$, we update states as follows:

$$\begin{aligned} x^{[i]}(t+1) = & \begin{cases} f\left(x^{[i+1]}(t-\sigma_0), \ldots, x^{[i+1]}(t-\sigma_m)\right) \\ \quad + g\left(x^{[i+1]}(t-\sigma_0), \ldots, x^{[i+1]}(t-\sigma_m)\right) \\ \quad \times u^{[i+1]}(t), \quad t \geq k \\ f\left(x^{[i]}(t-\sigma_0), \ldots, x^{[i]}(t-\sigma_m)\right) \\ \quad + g\left(x^{[i]}(t-\sigma_0), \ldots, x^{[i]}(t-\sigma_m)\right) \\ \quad \times u^{[i]}(t), \quad 0 \leq t < k \end{cases} \\ x^{[i]}(t) = & \varepsilon_1(t), \quad -\sigma_m \leq t \leq 0 \end{aligned} \tag{28}$$

where $u^{[i]}(t) = \upsilon^{[i]}(t) + u_s(t)$.

Here, we should point out that the parameter i in $[\cdot]$ of $x^{[i]}(k)$, $J^{[i]}(e^{[i-1]}(k))$, and $\upsilon^{[i]}(k)$ is the iteration index, and t, k in $(\cdot)$ are the time step indices.

Remark 3: From (28) we have that the state $x^{[i]}(t+1), t \geq k$, is related to $x^{[i+1]}(t-\sigma_0), \ldots, x^{[i+1]}(t-\sigma_m)$ and $u^{[i+1]}(t)$. It reflects the "backward iteration" of state at $t, t \geq k$.

Remark 4: One important property we must point out is that the HDP algorithm proposed in this paper is different from the algorithm presented in [37].

1) The HDP algorithm presented in [37] deals with nonlinear system without time delays. While the HDP algorithm proposed in this paper deals with time delay nonlinear system.
2) For the HDP algorithm presented in [37], only the performance index function is updated according to the control policy iteration. While for the HDP algorithm proposed in this paper, besides the performance index function iteration, the state is also updated according to the control policy iteration.

Based on the analysis above, our algorithm is a novel HDP algorithm. It is the development of the HDP algorithm presented in [37].

In the following part, we present the convergence analysis of the iteration about (21)–(28).

B. Convergence Analysis of the Iteration HDP Algorithm

Lemma 1: Let $v^{[i]}(k)$ and $J^{[i+1]}(e^{[i]}(k))$ be expressed as (25) and (27), and let $\{\mu^{[i]}(k)\}$ be arbitrary sequence of control policy, $\Lambda^{[i+1]}$ be defined by

$$\Lambda^{[i+1]}\left(e^{[i]}(k)\right) = \left(e^{[i]}(k)\right)^T Qe^{[i]}(k) + \left(\mu^{[i]}(k)\right)^T R\mu^{[i]}(k) + \Lambda^{[i]}\left(e^{[i-1]}(k+1)\right), \quad i \geq 0. \tag{29}$$

Thus, if $J^{[0]} = \Lambda^{[0]} = 0$, then $J^{[i+1]}(e^{[i]}(k)) \leq \Lambda^{[i+1]}(e^{[i]}(k))$ $\forall e^{[i]}(k)$.

Proof: For given $\varepsilon_1(k)(-\sigma_m \leq k \leq 0)$ of (1), it can be straight forward from the fact that $J^{[i+1]}(e^{[i]}(k))$ is obtained by the control input $v^{[i]}(k)$, while $\Lambda^{[i+1]}(e^{[i]}(k))$ is a result of arbitrary control input. ■

Lemma 2: Let the sequence $J^{[i+1]}(e^{[i]}(k))$ be defined by (27), $v^{[i]}(k)$ is the control policy expressed as (25). There is an upper bound Y, such that $0 \leq J^{[i+1]}(e^{[i]}(k)) \leq Y$, $\forall e^{[i]}(k)$.

Proof: Let $\gamma^{[i]}(k)$ be any control policy that satisfies Assumption 1. Let $J^{[0]} = P^{[0]} = 0$, and $P^{[i+1]}(e^{[i]}(k))$ is defined as follows:

$$P^{[i+1]}\left(e^{[i]}(k)\right) = \left(e^{[i]}(k)\right)^T Qe^{[i]}(k) + \left(\gamma^{[i]}(k)\right)^T R\gamma^{[i]}(k) + P^{[i]}\left(e^{[i-1]}(k+1)\right). \tag{30}$$

From (30), we can obtain

$$P^{[i]}\left(e^{[i-1]}(k+1)\right) = \left(e^{[i-1]}(k+1)\right)^T Qe^{[i-1]}(k+1) + \left(\gamma^{[i-1]}(k+1)\right)^T R\gamma^{[i-1]}(k+1) + P^{[i-1]}\left(e^{[i-2]}(k+2)\right). \tag{31}$$

Thus we can get

$$\begin{aligned} P^{[i+1]}\left(e^{[i]}(k)\right) = {} & \left(e^{[i]}(k)\right)^T Qe^{[i]}(k) \\ & + \left(\gamma^{[i]}(k)\right)^T R\gamma^{[i]}(k) \\ & + \left(e^{[i-1]}(k+1)\right)^T Qe^{[i-1]}(k+1) \\ & + \left(\gamma^{[i-1]}(k+1)\right)^T R\gamma^{[i-1]}(k+1) \\ & + \cdots + \left(e^{[0]}(k+i)\right)^T Qe^{[0]}(k+i) \\ & + \left(\gamma^{[0]}(k+i)\right)^T R\gamma^{[0]}(k+i). \end{aligned} \tag{32}$$

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

$$L(k+j) = \left(e^{[i-j]}(k+j)\right)^T Qe^{[i-j]}(k+j) + \left(\gamma^{[i-j]}(k+j)\right)^T R\gamma^{[i-j]}(k+j). \tag{33}$$

Then (32) can further be written as follows:

$$P^{[i+1]}\left(e^{[i]}(k)\right) = \sum_{j=0}^{i} L(k+j). \tag{34}$$

Furthermore, we can see that

$$\forall i : P^{[i+1]}\left(e^{[i]}(k)\right) \leq \lim_{i\to\infty} \sum_{j=0}^{i} L(k+j). \tag{35}$$

Noting that $\{\gamma^{[i]}(k)\}$ satisfies Assumption 1, according to the third condition of Assumption 1, there exists an upper bound Y such that

$$\lim_{i\to\infty} \sum_{j=0}^{i} L(k+j) \leq Y. \tag{36}$$

From Lemma 1 we can obtain

$$\forall i : J^{[i+1]}\left(e^{[i]}(k)\right) \leq P^{[i+1]}\left(e^{[i]}(k)\right) \leq Y. \tag{37}$$

■

Theorem 1: For (1), the iteration algorithm is as in (21)–(28), then we have $J^{[i+1]}(e^{[i]}(k)) \geq J^{[i]}(e^{[i]}(k))$, $\forall e^{[i]}(k)$.

Proof: For convenience of analysis, define a new sequence $\{\Phi^{[i]}(e(k))\}$ as follows:

$$\Phi^{[i]}(e(k)) = e^T(k)Qe(k) + \left(v^{[i]}(k)\right)^T Rv^{[i]}(k) + \Phi^{[i-1]}(e(k+1)) \tag{38}$$

with $v^{[i]}(k)$ defined by (25), $\Phi^{[0]}(e(k)) = 0$.

In the following part, we will prove $\Phi^{[i]}(e(k)) \leq J^{[i+1]}(e(k))$ by mathematical induction.

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

$$J^{[1]}(e(k)) - \Phi^{[0]}(e(k)) = e^T(k)Qe(k) + \left(v^{[0]}(k)\right)^T Rv^{[0]}(k) \geq 0 \tag{39}$$

thus for $i = 0$, we have

$$J^{[1]}(e(k)) \geq \Phi^{[0]}(e(k)). \tag{40}$$

Second, we assume it holds for i, i.e., $J^{[i]}(e(k)) \geq \Phi^{[i-1]}(e(k))$, for $\forall e(k)$.

Then, from (26) and (38), we can get

$$\begin{aligned} & J^{[i+1]}(e(k)) - \Phi^{[i]}(e(k)) \\ & \quad = J^{[i]}(e(k+1)) - \Phi^{[i-1]}(e(k+1)) \geq 0 \end{aligned} \tag{41}$$

the following inequality holds:

$$\Phi^{[i]}(e(k)) \leq J^{[i+1]}(e(k)). \tag{42}$$

Therefore, (42) is proved by mathematical induction, for $\forall e(k)$.

On the other hand, from Lemma 1, we have $\forall e^{[i]}(k)$, $J^{[i+1]}(e^{[i]}(k)) \leq \Phi^{[i+1]}(e^{[i]}(k))$. So we have $J^{[i]}(e(k)) \leq \Phi^{[i]}(e(k))$.

Therefore $\forall e(k)$, we have

$$J^{[i]}(e(k)) \leq \Phi^{[i]}(e(k)) \leq J^{[i+1]}(e(k)). \tag{43}$$

So, for $\forall e^{[i]}(k)$, we have

$$J^{[i+1]}\left(e^{[i]}(k)\right) \geq J^{[i]}\left(e^{[i]}(k)\right). \tag{44}$$

■

We let $J^L(e^L(k)) = \lim_{i\to\infty} J^{[i+1]}(e^{[i]}(k))$. Accordingly, $v^L(k) = \lim_{i\to\infty} v^{[i]}(k)$ and $e^L(k) = \lim_{i\to\infty} e^{[i]}(k)$ is the corresponding states. Let $x^L(k) = e^L(k) + \eta(k)$ and $u^L(k) = v^L(k) + u_s(k)$.

In the following part, we will show that the performance index function $J^L(e^L(k))$ satisfies the corresponding HJB function, and it is the optimal performance index function.

Theorem 2: If the performance index function $J^{i+1}(e^i(k))$ is defined by (27). Let $J^L(e^L(k)) = \lim_{i\to\infty} J^{[i+1]}(e^{[i]}(k))$. Let $v^L(k) = \lim_{i\to\infty} v^{[i]}(k)$ and $e^L(k) = \lim_{i\to\infty} e^{[i]}(k)$ is the corresponding states. Then the following equation can be established:

$$J^L\left(e^L(k)\right) = \left(e^L(k)\right)^T Qe^L(k) + \left(v^L(k)\right)^T Rv^L(k) + J^L\left(e^L(k+1)\right). \quad (45)$$

Proof: First, according to Theorem 1, we have $J^{[i+1]}(e^{[i]}(k)) \geq J^{[i]}(e^{[i]}(k))$, $\forall e^{[i]}(k)$. So, we can obtain

$$J^{[i+1]}\left(e^{[i]}(k)\right) \geq \left(e^{[i]}(k)\right)^T Qe^{[i]}(k) + \left(v^{[i-1]}(k)\right)^T Rv^{[i-1]}(k) + J^{[i-1]}\left(e^{[i-1]}(k+1)\right). \quad (46)$$

Let $i \to \infty$, we have $v^L(k) = \lim_{i\to\infty} v^{[i]}(k)$ and

$$J^L\left(e^L(k)\right) \geq \left(e^L(k)\right)^T Qe^L(k) + \left(v^L(k)\right)^T Rv^L(k) + J^L\left(e^L(k+1)\right). \quad (47)$$

On the other hand, for any i and arbitrary control policy $\left\{\mu^{[i]}(k)\right\}$, let $\Lambda^{[i+1]}(e^{[i]}(k))$ be as (29). By Lemma 1, we have

$$J^{[i+1]}\left(e^{[i]}(k)\right) \leq \left(e^{[i]}(k)\right)^T Qe^{[i]}(k) + \left(\mu^{[i]}(k)\right)^T R\mu^{[i]}(k) + \Lambda^{[i]}\left(e^{[i-1]}(k+1)\right). \quad (48)$$

Since $\mu^{[i]}(k)$ in (48) is chosen arbitrarily. We let the control policy $\{\mu^{[i]}(k)\} = \{v^{[i]}(k)\}, \forall i$. So we can get

$$J^{[i+1]}\left(e^{[i]}(k)\right) \leq \left(e^{[i]}(k)\right)^T Qe^{[i]}(k) + (v^{[i]}(k))^T Rv^{[i]}(k) + J^{[i]}\left(e^{[i-1]}(k+1)\right). \quad (49)$$

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

$$J^L(e^L(k)) \leq \left(e^L(k)\right)^T Qe^L(k) + \left(v^L(k)\right)^T Rv^L(k) + J^L\left(e^L(k+1)\right). \quad (50)$$

Thus, combining (47) with (50), we have

$$J^L\left(e^L(k)\right) = \left(e^L(k)\right)^T Qe^L(k) + \left(v^L(k)\right)^T Rv^L(k) + J^L\left(e^L(k+1)\right). \quad (51)$$

■

Next, we give a theorem to demonstrate $J^L(e^L(k)) = J^*(e^*(k))$.

Theorem 3: Let the performance index function $J^{[i+1]}(e^{[i]}(k))$ be defined as (27) and $J^L(e^L(k)) = \lim_{i\to\infty} J^{[i+1]}(e^{[i]}(k))$. $J^*(e^*(k))$ is defined as in (15). Then we have $J^L(e^L(k)) = J^*(e^*(k))$.

Proof: According to the definition $J^*(e(k)) = \inf_{v(k)} J(e(k), v(k))$, we know that

$$J^{[i+1]}(e(k)) \geq J^*(e(k)). \quad (52)$$

So for $\forall e^{[i]}(k)$, we have

$$J^{[i+1]}(e^{[i]}(k)) \geq J^*\left(e^{[i]}(k)\right). \quad (53)$$

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

$$J^L\left(e^L(k)\right) \geq J^*\left(e^L(k)\right). \quad (54)$$

On the other hand, according to the definition $J^*(e(k)) = \inf_{v(k)} J(e(k), v(k))$, for any $\theta > 0$ there exists a sequence of control policy $\mu^{[i]}(k)$, such that the associated performance index function $\Lambda^{[i+1]}(e(k))$ similar as (29) satisfies $\Lambda^{[i+1]}(e(k)) \leq J^*(e(k)) + \theta$. From Lemma 1, we can get

$$J^{[i+1]}(e(k)) \leq \Lambda^{[i+1]}(e(k)) \leq J^*(e(k)) + \theta. \quad (55)$$

So we have

$$J^{[i+1]}\left(e^{[i]}(k)\right) \leq J^*\left(e^{[i]}(k)\right) + \theta. \quad (56)$$

Let $i \to \infty$, and then we can obtain

$$J^L\left(e^L(k)\right) \leq J^*\left(e^L(k)\right) + \theta. \quad (57)$$

Noting that θ is chosen arbitrarily, we have

$$J^L\left(e^L(k)\right) \leq J^*\left(e^L(k)\right). \quad (58)$$

From (54) and (58), we can get

$$J^L\left(e^L(k)\right) = J^*\left(e^L(k)\right). \quad (59)$$

From Theorem 2, we can see that $J^L(e^L(k))$ satisfies HJB equation, so we have $v^L(k) = v^*(k)$. From (14) and (28), we have $e^L(k) = e^*(k)$. Then we draw conclusion $J^L(e^L(k)) = J^*(e^*(k))$. ■

After that, we give a theorem to demonstrate the state error system (11) is asymptotically stable, i.e., the system state $x(k)$ follows $\eta(k)$ asymptotically.

The following lemma is necessary for the proof of stability property.

Lemma 3: Define the performance index function sequence $\left\{J^{[i+1]}(e^{[i]}(k))\right\}$ as (27) with $J^{[0]} = 0$, and the control policy sequence $\left\{v^{[i]}(k)\right\}$ as (25). Then we have that for $\forall i = 0, 1, \ldots$, the performance index function $J^{[i+1]}(e^{[i]}(k))$ is a positive definite function.

Proof: The Lemma can be proved by the following three steps.

1) Show That Zero Is an Equilibrium Point for (11): For the autonomous system of (11), let $e(k-\sigma_0) = e(k-\sigma_1) = \cdots = e(k-\sigma_m) = 0$, we have $e(k+1) = 0$. According to the first condition of Assumption 1, when $e(k-\sigma_0) = e(k-\sigma_1) = \cdots = e(k-\sigma_m) = 0$, we have $v(k) = 0$. So according to the definition of equilibrium state in [1], we can say that zero is an equilibrium point for (11).

2) Show That for $\forall i$, the Performance Index Function $J^{[i+1]}(e^{[i]}(k)) = 0$ at the Equilibrium Point: This conclusion can be proved by mathematical induction.

For $i = 0$, we have $J^{[0]} = 0$, and

$$J^{[1]}\left(e^{[0]}(k)\right) = \left(e^{[0]}(k)\right)^T Qe^{[0]}(k) + \left(v^{[0]}(k)\right)^T Rv^{[0]}(k). \quad (60)$$

Let $e^{[0]}(k-\sigma_0) = e^{[0]}(k-\sigma_1) = \cdots = e^{[0]}(k-\sigma_m) = 0$ at the equilibrium point, hence we have $v^{[0]}(k) = 0$ according to Assumption 1. Then we can get $J^{[1]}(e^{[0]}(k)) = 0$ at the equilibrium point.

Assume that for any i, $J^{[i]}(e^{[i-1]}(k+1)) = 0$ holds at the equilibrium point. Then for $i+1$, we have

$$\begin{aligned} J^{[i+1]}\left(e^{[i]}(k)\right) &= \left(e^{[i]}(k)\right)^T Qe^{[i]}(k) \\ &\quad + \left(v^{[i]}(k)\right)^T Rv^{[i]}(k) + J^{[i]}\left(e^{[i-1]}(k+1)\right) \\ &= \left(e^{[i]}(k)\right)^T Qe^{[i]}(k) \\ &\quad + \left(v^{[i]}(k)\right)^T Rv^{[i]}(k) \\ &\quad + J^{[i]}\Big(f\left(e^{[i]}(k-\sigma_0)+\eta(k-\sigma_0),\ldots,\right. \\ &\quad \left. e^{[i]}(k-\sigma_m)+\eta(k-\sigma_0)\right) \\ &\quad + g\left(e^{[i]}(k-\sigma_0)+\eta(k-\sigma_0),\ldots,\right. \\ &\quad \left. e^{[i]}(k-\sigma_m)+\eta(k-\sigma_m)\right) \\ &\quad \times \left(v^{[i]}(k)+u_s(k)\right) - S(\eta(k))\Big). \end{aligned} \quad (61)$$

Let $e^{[i]}(k-\sigma_0) = e^{[i]}(k-\sigma_1) = \cdots = e^{[i]}(k-\sigma_m) = 0$ at the equilibrium point. Then according to the first condition of Assumption 1, we have $v^{[i]}(k) = 0$. So we can obtain $J^{[i+1]}(e^{[i]}(k)) = 0$ at the equilibrium point.

3) Show That the Iteration Performance Index Function $J^{[i+1]}(e^{[i]}(k))$, $i = 0, 1, \ldots$, is a Positive Definite Function: From (27), we can get

$$\begin{aligned} J^{[i+1]}\left(e^{[i]}(k)\right) &= \left(e^{[i]}(k)\right)^T Qe^{[i]}(k) \\ &\quad + \left(v^{[i]}(k)\right)^T Rv^{[i]}(k) \\ &\quad + \left(e^{[i-1]}(k+1)\right)^T Qe^{[i-1]}(k+1) \\ &\quad + \left(v^{[i-1]}(k+1)\right)^T Rv^{[i-1]}(k+1) \\ &\quad + \cdots + \left(e^{[0]}(k+i)\right)^T Qe^{[0]}(k+i) \\ &\quad + \left(v^{[0]}(k+i)\right)^T Rv^{[0]}(k+i). \end{aligned} \quad (62)$$

Then we have $J^{[i+1]}(e^{[i]}(k)) > 0$ for $e^{[i]}(k) \neq 0, \forall i$. On the other hand, as $e^{[i]}(k) \to \infty$, we have $J^{[i+1]}(e^{[i]}(k)) \to \infty$. So we can say that the performance index function $J^{[i+1]}(e^{[i]}(k))$ is a positive definite function. ■

Now, we give the theorem of stability property.

Theorem 4: Let the optimal control $v^*(k)$ be expressed as (13) and the performance index function $J^*(e^*(k))$ be expressed as (15). Then we have optimal control $v^*(k)$ stabilizes (11) asymptotically.

Proof: We have known that $J^L(e^L(k))$ is a positive definite function. Furthermore, we have proved $J^L(e^L(k)) = J^*(e^*(k))$ in Theorem 3. So we have $J^*(e^*(k))$ is a positive definite function.

Furthermore, according to the HJB equation (15), we have

$$\begin{aligned} &J^*\left(e^*(k+1)\right) - J^*(e^*(k)) \\ &\quad = -\left\{(e^*(k))^T Qe^*(k) + \left(v^*(k)\right)^T Rv^*(k)\right\} \le 0. \end{aligned} \quad (63)$$

According to the definition of Lyapunov function [53], we have $J^*(e^*(k))$ is a Lyapunov function, which proves the conclusion. ■

Therefore, we can conclude that the limitation of $\left\{J^{[i+1]}(e^{[i]}(k))\right\}$ satisfies HJB equation and is the optimal one.

In following section, we give the implementation steps of the new iteration HDP algorithm.

C. Design Steps of Iteration HDP Algorithm

In this section, we give the detailed procedure of the design steps for the iteration HDP algorithm.

- Step 1: Given the reference orbit $\eta(k)$, and the current time k and maximal iteration step $i_{\max}$ for implementation of the HDP algorithm. We also give the initial control policy $\beta(k)$ and initial states $\varepsilon_1(k)(-\sigma_m \le k \le 0)$ of (1).
- Step 2: Set the iteration step $i = 0$. According to (21) and (22), we can get $v^{[0]}(k)$ and $J^{[1]}(e(k))$.
- Step 3: We can get the updated states $x^{[0]}(1), \ldots, x^{[0]}(k)$ from (28), and then we have $J^{[1]}(e^{[0]}(k))$ from (24).
- Step 4: Set the iteration step $i = 1$.
- Step 5: According to (25) and (26), we can get $v^{[i]}(k)$ and $J^{[i+1]}(e(k))$.
- Step 6: From (28), we can get the updated states $x^{[i]}(1), \ldots, x^{[i]}(k)$. From (27), we have $J^{[i+1]}(e^{[i]}(k))$.
- Step 7: If $i > i_{\max}$, go to Step 8, otherwise, set $i = i + 1$ and go to Step 5.
- Step 8: Stop.

In this paper, $i_{\max}$ means the maximal iteration step of the iteration HDP algorithm. If the maximal iteration step $i_{\max}$ is larger, then the computational accuracy is higher. But the larger iteration step $i_{\max}$ can bring more computational burden. For example, if we require high computational accuracy, then the maximal iteration step $i_{\max}$ should be large. If we want to get the result quickly, then we should decrease $i_{\max}$.

In this paper, if the proposed algorithm is convergent within $i_{\max}$ steps, then we say the algorithm succeeds and the optimal control is obtained. If the proposed algorithm is not convergent within $i_{\max}$ steps, we say that the optimal control cannot be obtained within $i_{\max}$ steps.

In the following section, we will give the neural network implementation of the iteration HDP algorithm for discrete time nonlinear system with time delays.

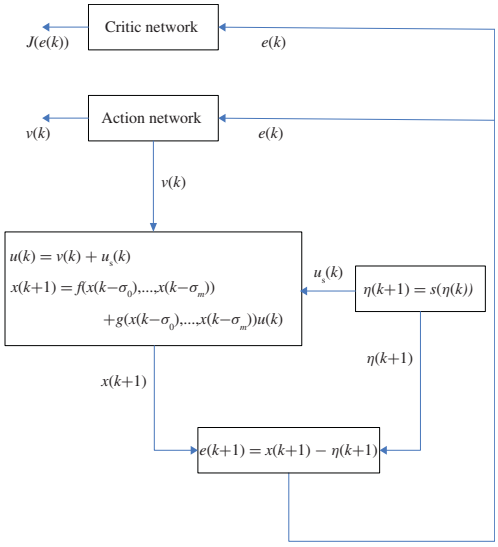

Fig. 1. Structure diagram of the algorithm.

IV. Neural Network Implementation of the Iteration HDP Algorithm

The nonlinear optimal control solution relies on solving the HJB equation, exact solution of which is generally impossible to be obtained for nonlinear time delay system. So we need to use parametric structures or neural networks to approximate both control policy and performance index function. Therefore, in order to implement the HDP iterations, we employ neural networks for approximations in this section.

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

$$\hat{F}(X, V, W) = W^T \sigma\left(V^T X\right) \tag{64}$$

where $\sigma(V^T X) \in R^l$ is the sigmoid function. The gradient descent rule is adopted for the weight update rules of each neural network [26].

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

A. Critic Network

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

$$\hat{J}^{[i+1]}(e(k)) = \left(w_c^{[i+1]}\right)^T \sigma\left((v_c^{[i+1]})^T e(k)\right). \tag{65}$$

The target function can be written as follows:

$$J^{[i+1]}(e(k)) = e^T(k) Q e(k) + \left(\hat{v}^{[i]}(k)\right)^T R \hat{v}^{[i]}(k) + \hat{J}^{[i]}(e(k+1)). \tag{66}$$

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

$$e_c^{[i+1]}(k) = \hat{J}^{[i+1]}(e(k)) - J^{[i+1]}(e(k)). \tag{67}$$

The objective function to be minimized in the critic network is

$$E_c^{[i+1]}(k) = \frac{1}{2}\left(e_c^{[i+1]}(k)\right)^2. \tag{68}$$

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

$$\begin{aligned} w_c^{[i+2]}(k) &= w_c^{[i+1]}(k) + \Delta w_c^{[i+1]}(k), \\ v_c^{[i+2]}(k) &= v_c^{[i+1]}(k) + \Delta v_c^{[i+1]}(k) \end{aligned} \tag{69}$$

where

$$\begin{aligned} \Delta w_c^{[i+1]}(k) &= -\alpha_c \frac{\partial E_c^{[i+1]}(k)}{\partial w_c^{[i+1]}(k)}, \\ \Delta v_c^{[i+1]}(k) &= -\alpha_c \frac{\partial E_c^{[i+1]}(k)}{\partial v_c^{[i+1]}(k)} \end{aligned} \tag{70}$$

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

B. Action Network

In the action network, the states $e(k-\sigma_0), \ldots, e(k-\sigma_m)$ are used as inputs to create the optimal control, $\hat{v}^{[i]}(k)$ as the output of the network. The output can be formulated as follows:

$$\hat{v}^{[i]}(k) = \left(w_a^{[i]}\right)^T \sigma\left((v_a^{[i]})^T Y(k)\right) \tag{71}$$

where $Y(k) = [e^T(k-\sigma_0), \ldots, e^T(k-\sigma_m)]^T$.

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

$$e_a^{[i]}(k) = \hat{v}^{[i]}(k) - v^{[i]}(k). \tag{72}$$

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

$$E_a^{[i]}(k) = \frac{1}{2}\left(e_a^{[i]}(k)\right)^2. \tag{73}$$

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

$$\begin{aligned} w_a^{[i+1]}(k) &= w_a^{[i]}(k) + \Delta w_a^{[i]}(k), \\ v_a^{[i+1]}(k) &= v_a^{[i]}(k) + \Delta v_a^{[i]}(k) \end{aligned} \tag{74}$$

where

$$\begin{aligned} \Delta w_a^{[i]}(k) &= -\alpha_a \frac{\partial E_a^{[i]}(k)}{\partial w_a^{[i]}(k)}, \\ \Delta v_a^{[i]}(k) &= -\alpha_a \frac{\partial E_a^{[i]}(k)}{\partial v_a^{[i]}(k)} \end{aligned} \tag{75}$$

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

V. Simulation Study

In this section, two examples are provided to demonstrate the effectiveness of the proposed iteration HDP algorithm in this paper. One is about tracking chaotic signal, and the other is about the situation that the function $g(\cdot)$ is noninvertible.

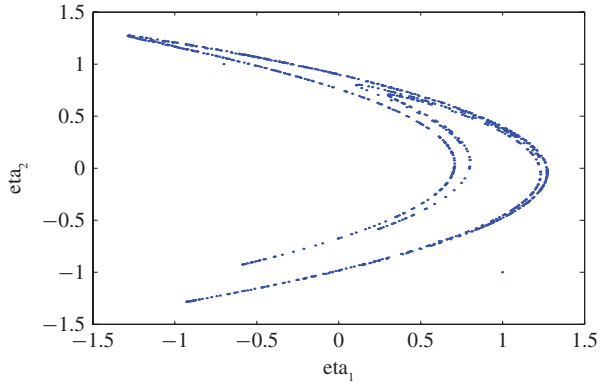

Fig. 2. Hénon chaos orbits.

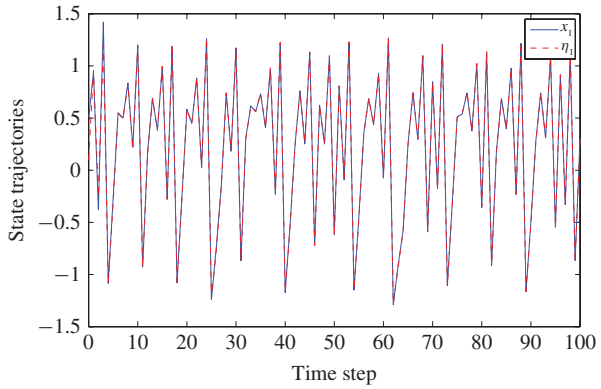

Fig. 3. State variable trajectory x_1 and desired trajectory η_1.

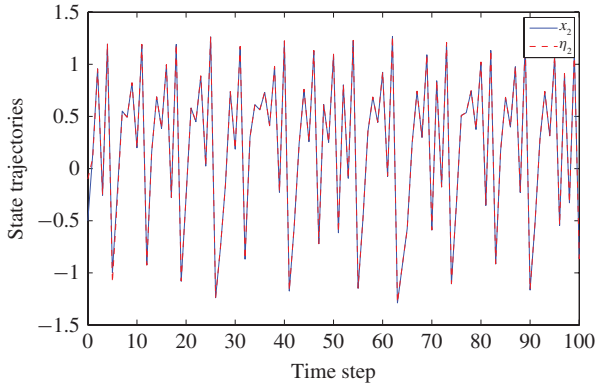

Fig. 4. State variable trajectory x_2 and desired trajectory η_2.

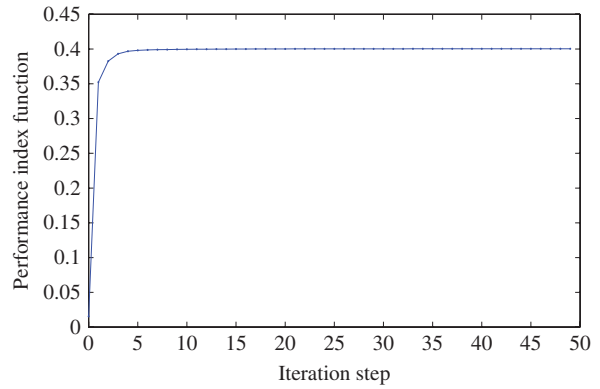

Fig. 5. Convergence of performance index.

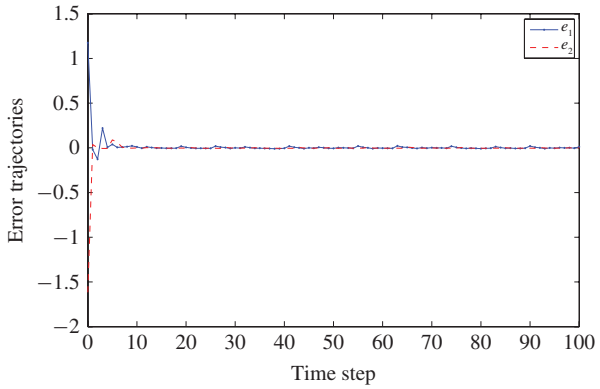

Fig. 6. Tracking error trajectories e_1 and e_2.

A. *Example* 1

Consider the following nonlinear time delay system which is the example in [54] and [37] with modification:

$$\begin{aligned} x(k+1) &= f(x(k), x(k-1), x(k-2)) \\ &\quad + g(x(k), x(k-1), x(k-2))u(k) \\ x(k) &= \varepsilon_1(k), \qquad -2 \le k \le 0 \end{aligned} \tag{76}$$

where

$$f(x(k), x(k-1), x(k-2)) = \begin{bmatrix} 0.2x_1(k)\exp\left(x_2(k)\right)^2 x_2(k-2) \\ 0.3\left(x_2(k)\right)^2 x_1(k-1) \end{bmatrix}$$

and

$$g(x(k), x(k-1), x(k-2)) = \begin{bmatrix} -0.2 & 0 \\ 0 & -0.2 \end{bmatrix}.$$

The desired signal is well-known Hénon mapping as follows:

$$\begin{bmatrix} \eta_1(k+1) \\ \eta_2(k+1) \end{bmatrix} = \begin{bmatrix} 1+b\times\eta_2(k)-a\times\eta_1^2(k) \\ \eta_1(k) \end{bmatrix} \tag{77}$$

where $a = 1.4$, $b = 0.3$, $\begin{bmatrix} \eta_1(0) \\ \eta_2(0) \end{bmatrix} = \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}$. The chaotic signal orbits are given as Fig. 2.

Based on the implementation of proposed HDP algorithm in Section III-C, we first give the initial states as $\varepsilon_1(-2) = \varepsilon_1(-1) = \varepsilon_1(0) = [\,0.5\ -0.5\,]^T$, and the initial control policy as $\beta(k) = -2x(k)$. The implementation of the algorithm is at the time instant $k = 3$. The maximal iteration step $i_{\max}$ is 50. We choose three-layer BP neural networks as the critic network and the action network with the structure $2-8-1$ and $6-8-2$, respectively. The iteration time of the weights updating for two neural networks is 100. The initial weights are chosen randomly from $[-0.1, 0.1]$, and the learning rate is $\alpha_a = \alpha_c = 0.05$. We select $Q = R = I_2$.

The state trajectories are given as Figs. 3 and 4. The solid lines in the two figures are the system states, and the dashed lines are the trajectories of the desired chaotic signal. From the two figures, we can see that the state trajectories of (76) follow the chaotic signal. From Lemma 2, Theorems 1 and 3, the performance index function sequence is bounded and

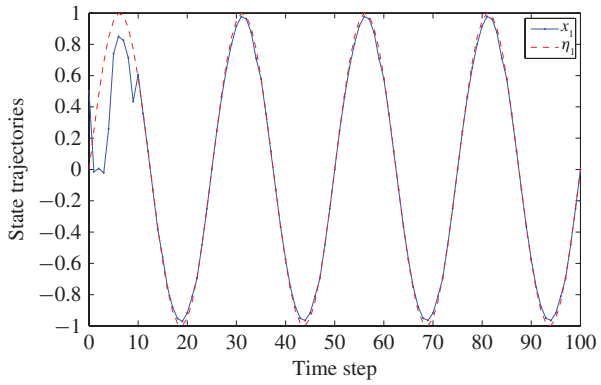

Fig. 7. State variable trajectory x_1 and desired trajectory η_1.

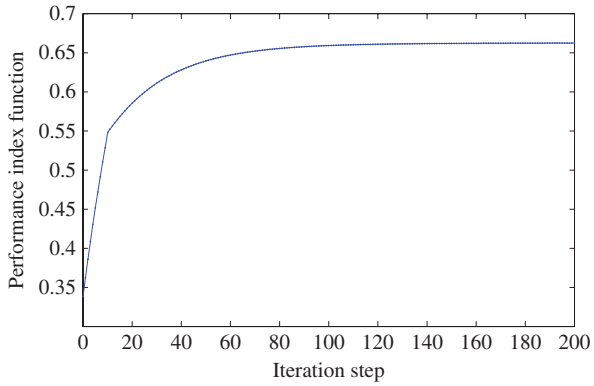

Fig. 9. Convergence of performance index.

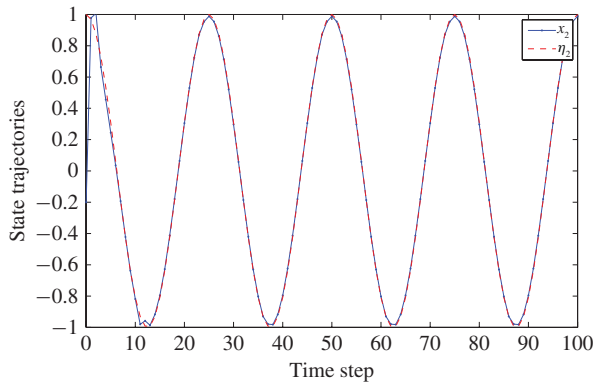

Fig. 8. State variable trajectory x_2 and desired trajectory η_2.

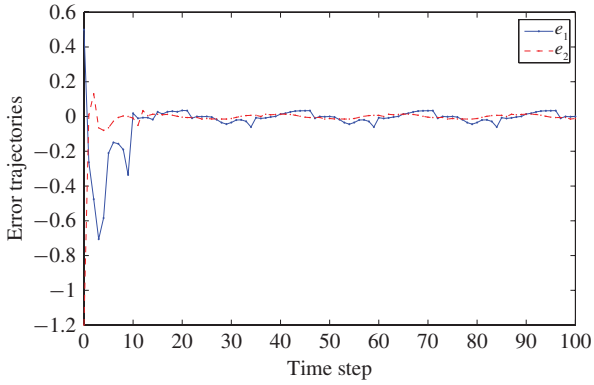

Fig. 10. Tracking error trajectories e_1 and e_2.

nondecreasing. Furthermore, it converges to the optimal performance index function as $i \to \infty$. The curve in Fig. 5 shows the properties of the performance index function sequence. According to Theorem 4, we know that (11) is asymptotically stable. The error trajectories between (76) and Hénon chaotic signal are presented in Fig. 6, and they converge to zero asymptotically. It is clear that the new HDP iteration algorithm is very feasible.

B. Example 2

In this subsection, we consider the following nonlinear time delay system:

$$\begin{aligned} x(k+1) &= f(x(k), x(k-1), x(k-2)) \\ &\quad + g(x(k), x(k-1), x(k-2))u(k) \\ x(k) &= \varepsilon_1(k), \qquad -2 \le k \le 0 \end{aligned} \tag{78}$$

where

$$f(x(k), x(k-1), x(k-2)) = \begin{bmatrix} 0.2x_1(k)\exp\left(x_2(k)\right)^2 x_2(k-2) \\ 0.3\left(x_2(k)\right)^2 x_1(k-1) \end{bmatrix}$$

and

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

From (78), we know that $g(\cdot)$ is not always invertible. Here Moore-Penrose pseudoinverse technique is used to obtain $g^{-1}(\cdot)$.

The desired orbit $\eta(k)$ is generated by the following exosystem:

$$\eta(k+1) = A\eta(k) \tag{79}$$

with

$$A = \begin{bmatrix} \cos wT & \sin wT \\ -\sin wT & \cos wT \end{bmatrix}, \quad \eta(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

where $T = 0.1$ s, $w = 0.8\pi$.

At first, we give the initial states as $\varepsilon_1(-2) = \varepsilon_1(-1) = \varepsilon_1(0) = [\,0.5 \;\; -0.2\,]^T$, and the initial control policy as $\beta(k) = -2x(k)$. We also implement the proposed HDP algorithm at the time instant $k = 3$. The maximal iteration step $i_{\max}$ is 60. The learning rate is $\alpha_a = \alpha_c = 0.01$, and other parameters in BP neural networks are the same as in Example 1. We select $Q = R = I_2$.

The trajectories of the system states are presented in Figs. 7 and 8. In the two figures, the solid lines are the system states, and the dashed lines are the desired trajectories. In addition, we give the curve of the performance index sequence in Fig. 9. It is bounded and convergent. It verifies Theorems 1 and 3 as well. The tracking errors show in Fig. 10, which converge to zero asymptotically. It is clear that the tracking performance is satisfactory, and the new iteration algorithm proposed in this paper is very effective.

VI. Conclusion

In this paper, we proposed an effective HDP algorithm to solve optimal tracking problem for a class of nonlinear discrete-time systems with time delays. First we defined a performance index for time delay systems. Then a novel iteration HDP algorithm has been developed to solve the optimal tracking control problem. Two neural networks have been used to facilitate the implementation of the iteration algorithm. Simulation examples have demonstrated the effectiveness of the proposed optimal tracking control algorithm.

References

[1] M. Z. Manu and J. Mohammad, *Time-Delay Systems Analysis, Optimization and Applications*. New York: North-Holland, 1987.

[2] D. H. Chyung, "On the controllability of linear systems with delay in control," *IEEE Trans. Autom. Control*, vol. 15, no. 2, pp. 255–257, Apr. 1972.

[3] D. H. Chyung, "Controllability of linear systems with multiple delays in control," *IEEE Trans. Autom. Control*, vol. 15, no. 6, pp. 694–695, Dec. 1970.

[4] H. Y. Shao and Q. L. Han, "New delay-dependent stability criteria for neural networks with two additive time-varying delay components," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 812–818, May 2011.

[5] J. P. Richard, "Time-delay systems: An overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, Oct. 2003.

[6] S. C. Tong, Y. M. Li, and H. G. Zhang, "Adaptive neural network decentralized backstepping output-feedback control for nonlinear large-scale systems with time delays," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1073–1086, Jul. 2011.

[7] V. N. Phat and H. Trinh, "Exponential stabilization of neural networks with various activation functions and mixed time-varying delays," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1180–1184, Jul. 2010.

[8] Y. J. Liu, C. L. P. Chen, G.-X. Wen, and S. C. Tong, "Adaptive neural output feedback tracking control for a class of uncertain discrete-time nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1162–1167, Jul. 2011.

[9] M. Wang, S. S. Ge, and K. S. Hong, "Approximation-based adaptive tracking control of pure-feedback nonlinear systems with multiple unknown time-varying delays," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1804–1816, Nov. 2010.

[10] W. S. Chen and L. C. Jiao, "Adaptive tracking for periodically time-varying and nonlinearly parameterized systems using multilayer neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 345–351, Feb. 2010.

[11] W. Y. Lan and J. Huang, "Neural-network-based approximate output regulation of discrete-time nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 18, no. 4, pp. 1196–1208, Jul. 2007.

[12] C. M. Zhang, G. Y. Tang, and S. Y. Han, "Approximate design of optimal tracking controller for systems with delayed state and control," in *Proc. IEEE Int. Conf. Control Autom.*, Christchurch, New Zealand, Dec. 2009, pp. 1168–1172.

[13] Y. D. Zhao, G. Y. Tang, and C. Li, "Optimal output tracking control for nonlinear time-delay systems," in *Proc. 6th World Congr. Intell. Control Autom.*, Dalian, China, Jun. 2006, pp. 757–761.

[14] Y. M. Park, M. S. Choi, and K. Y. Lee, "An optimal tracking neuro-controller for nonlinear dynamic systems," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1009–1110, Sep. 1996.

[15] J. Fu, H. B. He, and X. M. Zhou, "Adaptive learning and control for MIMO system based on adaptive dynamic programming," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1133–1148, Jul. 2011.

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

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

[18] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, Mar. 2007.

[19] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, Feb. 2009.

[20] K. G. Vamvoudakis and F. L. Lewis, "Online actor critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.

[21] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. New York: Wiley, 2009.

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

[23] P. He and S. Jagannathan, "Reinforcement learning neural-network-based controller for nonlinear discrete-time systems with input constraints," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 37, no. 2, pp. 425–436, Apr. 2007.

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

[25] B. H. Li and J. Si, "Approximate robust policy iteration using multilayer perceptron neural networks for discounted infinite-horizon Markov decision processes with uncertain correlated transition matrices," *IEEE Trans. Neural Netw.*, vol. 21, no. 8, pp. 1270–1280, Aug. 2010.

[26] J. Si and Y. T. Wang, "Online learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

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

[28] F. Y. Wang, N. Jin, D. R. Liu, and Q. L. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε-error bound," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.

[29] H. G. Zhang, Q. L. Wei, and D. R. Liu, "An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games," *Automatica*, vol. 47, no. 1, pp. 207–214, Jan. 2011.

[30] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand, 1992, ch. 13.

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

[32] Y. Zhao, S. D. Patek, and P. A. Beling, "Decentralized Bayesian search using approximate dynamic programming methods," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 970–975, Aug. 2008.

[33] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Sep. 2009.

[34] F. Y. Wang, H. G. Zhang, and D. R. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[35] S. J. Bradtke, B. E. Ydestie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proc. Amer. Control Conf.*, vol. 3. Baltimore, MD, Jun. 1994, pp. 3475–3479.

[36] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[37] H. G. Zhang, Q. L. Wei, and Y. H. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 937–942, Aug. 2008.

[38] H. G. Zhang, Y. H. Luo, and D. R. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sep. 2009.

[39] A. Isidori, *Nonlinear Control Systems II*. New York: Springer-Verlag, 2005.

[40] S. B. Gershwin and D. H. Jacobson, "A controllability theory for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 16, no. 1, pp. 37–46, Feb. 1971.

[41] A. Bicchi, A. Marigo, and B. Piccoli, "On the reachability of quantized control systems," *IEEE Trans. Autom. Control*, vol. 47, no. 4, pp. 546–563, Apr. 2002.

[42] A. Ben-Israel and T. N. E. Greville, *Generalized Inverse: Theory and Applications*, 2nd ed. New York: Springer-Verlag, 2002.

[43] A. Al-Tamimi, M. Abu-Khalaf, and F. L. Lewis, "Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 37, no. 1, pp. 240–247, Feb. 2007.
[44] Q. L. Wei, H. G. Zhang, D. R. Liu, and Y. Zhao, "An optimal control scheme for a class of discrete-time nonlinear systems with time delays using adaptive dynamic programming," *ACTA Autom. Sin.*, vol. 36, no. 1, pp. 121–129, Jan. 2010.
[45] R. Beard, "Improving the closed-loop performance of nonlinear systems," Ph.D. dissertation, Dept. Electr. Eng., Rensselaer Polytechnic Inst., Troy, NY, 1995.
[46] D. H. Chyung, "Discrete optimal systems with time delay," *IEEE Trans. Autom. Control*, vol. 13, no. 1, pp. 1–117, Feb. 1968.
[47] D. H. Chyung, "Discrete systems with time delay," presented at the 5th Annual Allerton Conference on Circuit and System Theory, Urbana, IL, Oct. 1967.
[48] D. H. Chyung and E. B. Lee, "Linear optimal systems with time delays," *SIAM J. Control*, vol. 4, no. 3, pp. 548–575, Nov. 1966.
[49] P. Yang, G. Xie, and L. Wang. *Controllability of Linear Discrete-Time Systems with Time-Delay in State and Control* [Online]. Available: http://dean.pku.edu.cn/bksky/1999tzlwj/4.pdf
[50] V. N. Phat, "Controllability of discrete-time systems with multiple delays on controls and states," *Int. J. Control*, vol. 49, no. 5, pp. 1645–1654, 1989.
[51] C. T. Chen, *Linear System Theory and Design*, 3rd ed. New York: Oxford Univ. Press, 1999.
[52] D. Wang, D. R. Liu, and Q. L. Wei, "Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach," *Neurocomputing*, to be published.
[53] X. Liao, L. Wang, and P. Yu, *Stability of Dynamical Systems*. Amsterdam, The Netherlands: Elsevier, 2007.
[54] A. Al-Tamimi and F. L. Lewis, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," in *Proc. IEEE Int. Symp. Approx. Dyn. Program. Reinforce. Learn.*, Honolulu, HI, Apr. 2007, pp. 38–43.

Huaguang Zhang (SM'04) received the B.S. and M.S. degrees in control engineering from the Northeast Dianli University of China, Jilin City, China, in 1982 and 1985, respectively, and the Ph.D. degree in thermal power engineering and automation from Southeast University, Nanjing, China, in 1991.

He joined the Department of Automatic Control, Northeastern University, Shenyang, China, in 1992, as a Post-Doctoral Fellow for two years. Since 1994, he has been a Professor and Head of the Institute of Electric Automation, School of Information Science and Engineering, Northeastern University. He has authored and co-authored over 200 journal and conference papers, four monographs, and co-invented on 20 patents. His current research interests include fuzzy controls, stochastic system controls, neural networks-based controls, nonlinear controls, and their applications.

Dr. Zhang is an Associate Editor of *Automatica*, the IEEE TRANSACTIONS ON FUZZY SYSTEMS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B, and *Neurocomputing*. He was awarded the Outstanding Youth Science Foundation Award from the National Natural Science Foundation Committee of China in 2003. He was named the Cheung Kong Scholar by the Education Ministry of China in 2005.

Ruizhuo Song (M'11) has been pursuing the Ph.D. degree with Northeastern University, Shenyang, China, since 2008.

Her current research interests include neural-networks-based controls, non-linear controls, fuzzy controls, adaptive dynamic programming, and their industrial applications.

Qinglai Wei (M'11) received the B.S. degree in automation, the M.S. degree in control theory and control engineering, and the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2002, 2005, and 2008, respectively.

He was a Post-Doctoral Fellow with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China, from 2009 to 2011. He is currently an Associate Researcher with the Institute of Automation. His current research interests include neural-networks-based controls, nonlinear controls, adaptive dynamic programming, and their industrial applications.

Tieyan Zhang (M'08) was born in 1962. He received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2007.

He is currently a Professor and President of the Shenyang Institute of Engineering, Shenyang. His current research interests include fuzzy controls, fault diagnosis on electric power systems, and stability analysis on smart grids.