Neurocomputing 78 (2012) 14–22

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach ☆

Ding Wang [a], Derong Liu [a,b,*], Qinglai Wei [a]

[a] State Key Laboratory of Intelligent Control and Management of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China
[b] Department of Electrical and Computer Engineering, University of Illinois, Chicago, IL 60607, USA

ARTICLE INFO

Available online 3 September 2011

Keywords:
Adaptive critic designs
Adaptive dynamic programming
Approximate dynamic programming
Finite-horizon optimal tracking control
Learning control
Neural networks
Reinforcement learning

ABSTRACT

In this paper, a finite-horizon neuro-optimal tracking control strategy for a class of discrete-time nonlinear systems is proposed. Through system transformation, the optimal tracking problem is converted into designing a finite-horizon optimal regulator for the tracking error dynamics. Then, with convergence analysis in terms of cost function and control law, the iterative adaptive dynamic programming (ADP) algorithm via heuristic dynamic programming (HDP) technique is introduced to obtain the finite-horizon optimal tracking controller which makes the cost function close to its optimal value within an ε-error bound. Three neural networks are used as parametric structures to implement the algorithm, which aims at approximating the cost function, the control law, and the error dynamics, respectively. Two simulation examples are included to complement the theoretical discussions.

1. Introduction

It is well known that the optimal tracking control problem has been the focus of control systems community for several decades since it is usually encountered in real world systems [1–3]. In the case of infinite-horizon optimal tracking control, the system will not be tracked until the time reaches infinity, while for the finite case, the system must be tracked to a reference trajectory in a finite duration of time. Since many limitations exist in traditional optimal tracking control approaches, such as plant inversion [2] and linearization [3], it is necessary to design direct optimal tracking control schemes for nonlinear systems. In this paper, we will study how to solve this problem through the framework of Hamilton–Jacobi–Bellman (HJB) [4] equation from optimal control theory. Unlike the open-loop optimal controller design for nonlinear systems, however, for closed-loop optimal feedback control, it is difficult to solve directly the time-varying HJB equation which involves solving either nonlinear partial difference or differential equations. Though dynamic programming (DP) has been an useful computational technique in solving optimal control problems for many years, it is often computationally untenable to run it to obtain the optimal solution due to the "curse of dimensionality" [5].

As Poggio and Girosi [6] stated, the problem of learning between input and output spaces is equivalent to that of synthesizing an associative memory that retrieves appropriate output when the input is present and generalizes when a new input is applied. With strong capabilities of self-learning and adaptivity, artificial neural networks (ANN or NN) are an effective tool for implementing intelligent control [7–10]. Besides, it has been used for universal function approximation in adaptive/approximate dynamic programming (ADP) algorithms, which were proposed in [9–11] as a method for solving optimal control problems forward-in-time. There are several synonyms used for ADP including "adaptive dynamic programming" [12–14], "approximate dynamic programming" [9,15,16], "neuro-dynamic programming" [17], "neural dynamic programming" (NDP) [18], "adaptive critic designs" [19], and "reinforcement learning" [15,20]. As an effective intelligent control method, ADP and the related research have gained much attention from researchers [9–19,21–35]. Very good surveys were given in Wang et al. [13], Lewis and Vrabie [14], and Balakrishnan et al. [25]. According to [9,19], ADP approaches were classified into several main schemes: heuristic dynamic programming (HDP), action-dependent HDP (ADHDP), also known as Q-learning [20], dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP. Al-Tamimi et al. [16] proposed a greedy HDP algorithm to solve the discrete-time HJB (DTHJB) equation for optimal control of nonlinear systems. Wang et al. [23] developed an ε-ADP algorithm for studying finite-horizon optimal control of discrete-time nonlinear systems.

☆This work was supported in part by the NSFC under Grants 60874043, 60904037, 60921061, and 61034002, by Beijing Natural Science Foundation under Grant 4102061, and by the NSF under Grant ECCS-1027602.

*Corresponding author at: State Key Laboratory of Intelligent Control and Management of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China. Tel.: +86 10 62557379, +1 312 355 4475; fax: +86 10 62650912, +1 312 966 6465.

E-mail addresses: ding.wang@ia.ac.cn (D. Wang), derong.liu@ia.ac.cn, dliu@ece.uic.edu (D. Liu), qinglai.wei@ia.ac.cn (Q. Wei).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.03.058

With the rapid development of NN technology and recently, the ADP method, various new strategies were devised to deal with the optimal tracking control problems. Park et al. [36] used the multilayer NN to design an optimal tracking neuro-controller for discrete-time nonlinear systems with quadratic cost function. Zhang et al. [32] gave a novel infinite-horizon optimal tracking control scheme for discrete-time nonlinear systems via greedy HDP algorithm. Dierks and Jagannathan [31] utilized the NDP technique to solve the HJB equation forward-in-time for optimal tracking control of affine nonlinear systems. However, to the best of our knowledge, there is still no result to solve the finite-horizon optimal tracking control problem for discrete-time nonlinear systems based on iterative ADP algorithm via HDP technique (iterative HDP algorithm for brief). In this paper, for the first time, we will provide an iterative ADP algorithm to design finite-horizon near-optimal tracking controller for a class of discrete-time nonlinear systems.

The rest of this paper is organized as follows. In Section 2, we present the problem statement, transform the finite-horizon optimal tracking control problem into an optimal regulation problem, and introduce the DTHJB equation for nonlinear systems. Section 3 starts by deriving the iterative ADP algorithm with convergence analysis, and then the finite-horizon optimal tracking control scheme is proposed which makes the cost function close to its optimal value within an ε-error bound. In Section 4, the NN implementation of the iterative ADP algorithm is presented. In Section 5, two examples are given to substantiate the theoretical results. Section 6 contains concluding remarks.

2. Problem statement

Consider the discrete-time nonlinear systems given by

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

where $x_k\in\mathbb{R}^n$ is the state, $u_p(x_k)\in\mathbb{R}^m$ is the control vector, $f(\cdot)$ and $g(\cdot)$ are differentiable in their argument with $f(0)=0$. Assume that $f+gu_p$ is Lipschitz continuous on a set Ω in $\mathbb{R}^n$ containing the origin, and that the system (1) is controllable in the sense that there exists a continuous control on Ω that asymptotically stabilizes the system. In the following part, $u_p(x_k)$ is denoted by u_{pk} for simplicity.

The objective for optimal tracking control problem is to determine optimal control law u_p^*, so as to make the nonlinear system (1) to track a reference (or desired) trajectory r_k in an optimal manner. Here, we assume that the reference trajectory r_k satisfies

$$r_{k+1}=\phi(r_k), \tag{2}$$

where $r_k\in\mathbb{R}^n$ and $\phi(r_k)\in\mathbb{R}^n$. Then, we define the tracking error as

$$e_k=x_k-r_k. \tag{3}$$

Inspired by the work of [31,32,36], we define the steady control corresponding to the reference trajectory r_k as

$$u_{dk}=g^{-1}(r_k)(\phi(r_k)-f(r_k)), \tag{4}$$

where $g^{-1}(r_k)g(r_k)=I_m$ and I_m is an $m\times m$ identity matrix.

By denoting

$$u_k=u_{pk}-u_{dk} \tag{5}$$

and using (1)–(4), we obtain

$$\begin{cases} e_{k+1}=f(e_k+r_k)+g(e_k+r_k)g^{-1}(r_k)(\phi(r_k),\\ \qquad -f(r_k))-\phi(r_k)+g(e_k+r_k)u_k\\ r_{k+1}=\phi(r_k) \end{cases} \tag{6}$$

as the new system. Note that in system (6), e_k and r_k are regarded as the system variables while u_k is seen as system input. The second equation of system (6) only gives the evolution of the reference trajectory which is not affected by the system input. Therefore, for simplicity, (6) can be rewritten as

$$e_{k+1}=F(e_k,u_k). \tag{7}$$

Now, let e_0 be an initial state of system (7) and define $\underline{u}_0^{N-1}=(u_0,u_1,\ldots,u_{N-1})$ be a control sequence with which the system (7) gives a trajectory starting from e_0: $e_1, e_2, \ldots, e_N$. We call the number of elements in the control sequence $\underline{u}_0^{N-1}$ the length of $\underline{u}_0^{N-1}$ and denote it as $|\underline{u}_0^{N-1}|$. Then, $|\underline{u}_0^{N-1}|=N$. The final state under the control sequence $\underline{u}_0^{N-1}$ is denoted as $e^{(f)}(e_0,\underline{u}_0^{N-1})=e_N$.

Definition 1. A nonlinear dynamical system is said to be stabilizable on a compact set $\Omega\in\mathbb{R}^n$, if for all initial conditions $e_0\in\Omega$, there exists a control sequence $\underline{u}_0^{N-1}=(u_0,u_1,\ldots,u_{N-1})$, $u_i\in\mathbb{R}^m$, $i=0,1,\ldots,N-1$, such that the state $e^{(f)}(e_0,\underline{u}_0^{N-1})=0$.

Let $\underline{u}_k^{N-1}=(u_k,u_{k+1},\ldots,u_{N-1})$ be the control sequence starting at k with length $N-k$. For finite-horizon optimal tracking control problem, it is desired to find the control sequence which minimizes the following cost function:

$$J(e_k,\underline{u}_k^{N-1})=\sum_{i=k}^{N-1}U(e_i,u_i), \tag{8}$$

where U is the utility function, $U(0,0)=0$, $U(e_i,u_i)\geq 0$ for $\forall e_i,u_i$. In this paper, the utility function is chosen as the quadratic form as follows:

$$U(e_i,u_i)=e_i^TQe_i+u_i^TRu_i.$$

This quadratic cost function can not only force the system state to follow the reference trajectory, but also force the system input to be close to the steady value in maintaining the state to its reference value. In fact, it can also be expressed as

$$U(e_i,u_i)=[e_i^T\ r_i^T]\begin{bmatrix}Q & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}e_i\\ r_i\end{bmatrix}+u_i^TRu_i,$$

when considered from the angle of system (6).

In this sense, the nonlinear tracking problem is converted into a regulation problem and the finite-horizon cost function for tracking is written in terms of e_k and u_k. Then, the problem of solving the finite-horizon optimal tracking control law u_p^* for system (1) is transformed into seeking the finite-horizon optimal control law u^* for system (7) with respect to (8). As a result, we will focus on how to design u^* in the following sections.

For finite-horizon optimal control problems, the designed feedback control must be finite-horizon admissible, which means it must not only stabilize the controlled system on Ω within finite number of time steps but also guarantee the cost function to be finite.

Definition 2. A control sequence $\underline{u}_k^{N-1}$ is said to be finite- horizon admissible for a state $e_k\in\mathbb{R}^n$ with respect to (8) on Ω if $\underline{u}_k^{N-1}$ is continuous on a compact set $\Omega_u\in\mathbb{R}^m$, $u(0)=0$, $e^{(f)}(e_k,\underline{u}_k^{N-1})=0$ and $J(e_k,\underline{u}_k^{N-1})$ is finite.

Let

$$\mathfrak{U}_{e_k}=\{\underline{u}_k: e^{(f)}(e_k,\underline{u}_k)=0\}$$

be the set of all finite-horizon admissible control sequences of e_k and

$$\mathfrak{U}_{e_k}^{(i)}=\{\underline{u}_k^{k+i-1}: e^{(f)}(e_k,\underline{u}_k^{k+i-1})=0, |\underline{u}_k^{k+i-1}|=i\}$$

be the set of all finite-horizon admissible control sequences of e_k with length i. Define the optimal cost function as

$$J^*(e_k)=\inf_{\underline{u}_k}\{J(e_k,\underline{u}_k): \underline{u}_k\in\mathfrak{U}_{e_k}\}. \tag{9}$$

Note that Eq. (8) can be written as

$$J(e_k,\underline{u}_k^{N-1}) = e_k^T Q e_k + u_k^T R u_k + \sum_{i=k+1}^{N-1} U(e_i,u_i) = e_k^T Q e_k + u_k^T R u_k + J(e_{k+1},\underline{u}_{k+1}^{N-1}). \quad (10)$$

Then, according to Bellman's optimality principle, it is known that the optimal cost function $J^*(e_k)$ satisfies the DTHJB equation

$$J^*(e_k) = \min_{u_k}\{e_k^T Q e_k + u_k^T R u_k + J^*(e_{k+1})\}. \quad (11)$$

The optimal control u^* satisfies the first-order necessary condition, which is given by the gradient of the right-hand side of (11) with respect to u_k. Then,

$$u^*(e_k) = -\frac{1}{2} R^{-1} g^T(e_k+r_k)\frac{\partial J^*(e_{k+1})}{\partial e_{k+1}}. \quad (12)$$

By substituting (12) into (11), the DTHJB equation becomes

$$J^*(e_k) = e_k^T Q e_k + \frac{1}{4}\left(\frac{\partial J^*(e_{k+1})}{\partial e_{k+1}}\right)^T g(e_k+r_k)R^{-1} \times g^T(e_k+r_k)\frac{\partial J^*(e_{k+1})}{\partial e_{k+1}} + J^*(e_{k+1}), \quad (13)$$

where $J^*(e_k)$ is the optimal cost function corresponding to the optimal control law $u^*(e_k)$. Since the above DTHJB equation cannot be solved exactly, we will present a novel algorithm to approximate the cost function iteratively in next section. Before that, we make the following assumption.

Assumption 1. For system (6), the inverse of the control coefficient matrix $g(e_k+r_k)$ exists.

Assumption 1 make sure that for given e_k and r_k, there exists an initial control u_k which can derive e_k to zero in one time step.

3. Finite-horizon neuro-optimal tracking control based on iterative ADP algorithm

Four subsections are included in this section. The iterative ADP algorithm is introduced in the first subsection, while in the second subsection, the corresponding convergence proof is developed. Then, the ε-optimal control algorithm and its design procedure are described in the third and fourth subsections, respectively.

3.1. Derivation of the iterative ADP algorithm

In this part, we present the iterative ADP algorithm, where the cost function and the control law are updated by recursive iterations.

First, we start with the initial cost function $V_0(\cdot)=0$, and then solve for the law of single control vector $v_0(e_k)$ as follows:

$$v_0(e_k) = \arg\min_{u_k}\{U(e_k,u_k)+V_0(e_{k+1})\} \quad \text{subject to } F(e_k,u_k)=0. \quad (14)$$

Once the control law $v_0(e_k)$ is determined, we update the cost function as

$$V_1(e_k) = \min_{u_k}\{U(e_k,u_k)+V_0(e_{k+1})\} = U(e_k,v_0(e_k)),$$

which can also be written as

$$V_1(e_k) = \min_{u_k} U(e_k,u_k) \text{ subject to } F(e_k,u_k)=0 = U(e_k,v_0(e_k)). \quad (15)$$

Then, for $i=1,2,\ldots$, the iterative algorithm can be implemented between the control law

$$v_i(e_k) = \arg\min_{u_k}\{U(e_k,u_k)+V_i(e_{k+1})\} = -\frac{1}{2}R^{-1}g^T(e_k+r_k)\frac{\partial V_i(e_{k+1})}{\partial e_{k+1}} \quad (16)$$

and the cost function

$$V_{i+1}(e_k) = \min_{u_k}\{U(e_k,u_k)+V_i(e_{k+1})\} = U(e_k,v_i(e_k))+V_i(F(e_k,v_i(e_k))). \quad (17)$$

Remark 1. In the iterative ADP algorithm (14)–(17), i is the iteration index of the control law and the cost function, while k is the time index of system's control and state trajectories. The cost function and control law are updated until they converge to the optimal ones.

Next, we will present a convergence proof of the iteration between (16) and (17) with the cost function $V_i \to J^*$ and the control law $v_i \to u^*$ as $i\to\infty$. Before that, we will see what $V_{i+1}(e_k)$ will be when it is expanded. According to (15) and (17), we can obtain

$$\begin{aligned} V_{i+1}(e_k) &= \min_{u_k}\{U(e_k,u_k)+V_i(e_{k+1})\} \\ &= \min_{\underline{u}_k^{k+1}}\{U(e_k,u_k)+U(e_{k+1},u_{k+1})+V_{i-1}(e_{k+2})\} \\ &\vdots \\ &= \min_{\underline{u}_k^{k+i-1}}\{U(e_k,u_k)+U(e_{k+1},u_{k+1}) \\ &\quad +\cdots+U(e_{k+i-1},u_{k+i-1})+V_1(e_{k+i})\}, \end{aligned} \quad (18)$$

where

$$V_1(e_{k+i}) = \min_{u_{k+i}} U(e_{k+i},u_{k+i}) \quad \text{subject to } F(e_{k+i},u_{k+i})=0. \quad (19)$$

Then, we have

$$\begin{aligned} V_{i+1}(e_k) &= \min_{\underline{u}_k^{k+i}}\sum_{j=0}^{i} U(e_{k+j},u_{k+j}) \\ &\text{subject to } F(e_{k+i},u_{k+i})=0 \\ &= \min_{\underline{u}_k^{k+i}}\left\{J(e_k,\underline{u}_k^{k+i}) : \underline{u}_k^{k+i}\in\mathfrak{U}_{e_k}^{(i+1)}\right\}, \end{aligned} \quad (20)$$

which can also be written as

$$V_{i+1}(e_k) = \sum_{j=0}^{i} U(e_{k+j},v_{i-j}(e_{k+j})) \quad (21)$$

when using the notation in (16). These equations will be useful in the convergence proof of the iterative ADP algorithm.

3.2. Convergence analysis of the iterative ADP algorithm

Theorem 1. *Suppose $\mathfrak{U}_{e_k}^{(1)}\neq\emptyset$. Then, the cost function sequence $\{V_i\}$ obtained by (14)–(17) is a monotonically nonincreasing sequence satisfying $V_{i+1}(e_k)\le V_i(e_k)$ for $\forall i\ge 1$, i.e., $V_1(e_k)=\max\{V_i(e_k): i=1,2,\ldots\}$.*

Proof. We prove this theorem by mathematical induction. First, we let $i=1$. The cost function $V_1(e_k)$ is given in (15) and the finite-horizon admissible control sequence is $\hat{\underline{u}}_k^k = (v_0(e_k))$. Now, we show that there exists a finite-horizon admissible control sequence $\hat{\underline{u}}_k^{k+1}$ with length 2 such that $J(e_k,\hat{\underline{u}}_k^{k+1}) = V_1(e_k)$. Let $\hat{\underline{u}}_k^{k+1} = (\hat{\underline{u}}_k^k, 0)$, then $|\hat{\underline{u}}_k^{k+1}| = 2$. Since $e_{k+1} = F(e_k,v_0(e_k)) = 0$ and $\hat{u}_{k+1}=0$, we have $e_{k+2} = F(e_{k+1},\hat{u}_{k+1}) = F(0,0) = 0$. Thus, $\hat{\underline{u}}_k^{k+1}$ is a finite-horizon admissible control sequence. Since $U(e_{k+1},\hat{u}_{k+1}) = U(0,0) = 0$,

we can obtain

$$J(e_k,\underline{\hat{u}}_k^{k+1}) = U(e_k,v_0(e_k)) + U(e_{k+1},\hat{u}_{k+1}) = U(e_k,v_0(e_k)) = V_1(e_k).$$

On the other hand, according to (20), we have

$$V_2(e_k) = \min_{\underline{u}_k^{k+1}}\{J(e_k,\underline{u}_k^{k+1}) : \underline{u}_k^{k+1} \in \mathfrak{U}_{e_k}^{(2)}\},$$

which reveals that

$$V_2(e_k) \le J(e_k,\underline{\hat{u}}_k^{k+1}) = V_1(e_k). \tag{22}$$

Therefore, the theorem holds for $i=1$.

Next, assume that the theorem holds for any $i=q$, where $q>1$. The current cost function can be expressed as

$$V_q(e_k) = \sum_{j=0}^{q-1} U(e_{k+j},v_{q-1-j}(e_{k+j})),$$

where $\underline{\hat{u}}_k^{k+q-1} = (v_{q-1}(e_k),v_{q-2}(e_{k+1}),\ldots,v_0(e_{k+q-1}))$ is the corresponding finite-horizon admissible control sequence.

Then, for $i=q+1$, we can construct a control sequence $\underline{\hat{u}}_k^{k+q} = (v_{q-1}(e_k),v_{q-2}(e_{k+1}),\ldots,v_0(e_{k+q-1}),0)$ with length $q+1$, under which the error trajectory is given as e_k, $e_{k+1} = F(e_k, v_{q-1}(e_k))$, $e_{k+2} = F(e_{k+1},v_{q-2}(e_{k+1})),\ldots,$ $e_{k+q} = F(e_{k+q-1}, v_0(e_{k+q-1})) = 0$, $e_{k+q+1} = F(e_{k+q},\hat{u}_{k+q}) = F(0,0) = 0$. This shows that $\underline{\hat{u}}_k^{k+q}$ is a finite-horizon admissible control sequence. As $U(e_{k+q},\hat{u}_{k+q}) = U(0,0) = 0$, we can acquire

$$\begin{aligned} J(e_k,\underline{\hat{u}}_k^{k+q}) &= U(e_k,v_{q-1}(e_k)) + U(e_{k+1},v_{q-2}(e_{k+1})) \\ &\quad + \cdots + U(e_{k+q-1},v_0(e_{k+q-1})) + U(e_{k+q},\hat{u}_{k+q}) \\ &= \sum_{j=0}^{q-1} U(e_{k+j},v_{q-1-j}(e_{k+j})) = V_q(e_k). \end{aligned}$$

On the other hand, according to (20), we have

$$V_{q+1}(e_k) = \min_{\underline{u}_k^{k+q}}\{J(e_k,\underline{u}_k^{k+q}) : \underline{u}_k^{k+q} \in \mathfrak{U}_{e_k}^{(q+1)}\},$$

which implies that

$$V_{q+1}(e_k) \le J(e_k,\underline{\hat{u}}_k^{k+q}) = V_q(e_k). \tag{23}$$

Accordingly, we complete the proof by mathematical induction. □

We have concluded that the cost function sequence $\{V_i(e_k)\}$ is a monotonically nonincreasing sequence which is bounded below, and therefore, its limit exists. Here, we denote it as $V_\infty(e_k)$, i.e., $\lim_{i\to\infty} V_i(e_k) = V_\infty(e_k)$. Next, let us consider what will happen when we make $i\to\infty$ in (17).

Theorem 2. *For any discrete time step k and tracking error e_k, the following equation holds*:

$$V_\infty(e_k) = \min_{u_k}\{U(e_k,u_k) + V_\infty(e_{k+1})\}. \tag{24}$$

Proof. For any admissible control $\tau_k = \tau(e_k)$ and i, according to Theorem 1 and (17), we have

$$V_\infty(e_k) \le V_{i+1}(e_k) = \min_{u_k}\{U(e_k,u_k) + V_i(e_{k+1})\} \le U(e_k,\tau_k) + V_i(e_{k+1}).$$

Let $i\to\infty$, we get

$$V_\infty(e_k) \le U(e_k,\tau_k) + V_\infty(e_{k+1}).$$

Note that in the above equation, τ_k is chosen arbitrarily. Thus, we can obtain

$$V_\infty(e_k) \le \min_{u_k}\{U(e_k,u_k) + V_\infty(e_{k+1})\}. \tag{25}$$

On the other hand, let $\delta>0$ be an arbitrary positive number. Then, there exists a positive integer l such that

$$V_l(e_k) - \delta \le V_\infty(e_k) \le V_l(e_k) \tag{26}$$

because $V_i(e_k)$ is nonincreasing for $i\ge 1$ with $V_\infty(e_k)$ as its limit. Besides, from (17), we can acquire

$$\begin{aligned} V_l(e_k) &= \min_{u_k}\{U(e_k,u_k) + V_{l-1}(e_{k+1})\} \\ &= U(e_k,v_{l-1}(e_k)) + V_{l-1}(F(e_k,v_{l-1}(e_k))). \end{aligned}$$

Combining with (26), we can obtain

$$\begin{aligned} V_\infty(e_k) &\ge U(e_k,v_{l-1}(e_k)) + V_{l-1}(F(e_k,v_{l-1}(e_k))) - \delta \\ &\ge U(e_k,v_{l-1}(e_k)) + V_\infty(F(e_k,v_{l-1}(e_k))) - \delta \\ &\ge \min_{u_k}\{U(e_k,u_k) + V_\infty(e_{k+1})\} - \delta, \end{aligned}$$

which reveals that

$$V_\infty(e_k) \ge \min_{u_k}\{U(e_k,u_k) + V_\infty(e_{k+1})\} \tag{27}$$

because of the arbitrariness of δ. Based on (25) and (27), we can conclude that (24) is true. □

Next, we will prove that the cost function sequence $\{V_i\}$ converges to the optimal cost function J^* as $i\to\infty$.

Theorem 3. *Define the cost function sequence $\{V_i\}$ as in (17) with $V_0(\cdot)=0$. If the system state e_k is controllable, then J^* is the limit of the cost function sequence $\{V_i\}$, i.e.,*

$$V_\infty(e_k) = J^*(e_k).$$

Proof. On one hand, in accordance with (9) and (20), we can acquire

$$\begin{aligned} J^*(e_k) &= \inf_{\underline{u}_k}\{J(e_k,\underline{u}_k) : \underline{u}_k \in \mathfrak{U}_{e_k}\} \\ &\le \min_{\underline{u}_k^{k+i-1}}\{J(e_k,\underline{u}_k^{k+i-1}) : \underline{u}_k^{k+i-1} \in \mathfrak{U}_{e_k}^{(i)}\} = V_i(e_k). \end{aligned}$$

Letting $i\to\infty$, we get

$$J^*(e_k) \le V_\infty(e_k). \tag{28}$$

On the other hand, according to the definition of $J^*(e_k)$, for any $\eta>0$, there exists an admissible control sequence $\underline{\sigma}_k \in \mathfrak{U}_{e_k}$ such that

$$J(e_k,\underline{\sigma}_k) \le J^*(e_k) + \eta. \tag{29}$$

Now, we suppose that $|\underline{\sigma}_k| = q$, which shows that $\underline{\sigma}_k \in \mathfrak{U}_{e_k}^{(q)}$. Then, we can obtain

$$\begin{aligned} V_\infty(e_k) \le V_q(e_k) &= \min_{\underline{u}_k^{k+q-1}}\{J(e_k,\underline{u}_k^{k+q-1}) : \underline{u}_k^{k+q-1} \in \mathfrak{U}_{e_k}^{(q)}\} \\ &\le J(e_k,\underline{\sigma}_k), \end{aligned}$$

using Theorem 1 and (20). Combining with (29), we get

$$V_\infty(e_k) \le J^*(e_k) + \eta.$$

Noticing that η is chosen arbitrarily in the above expression, we have

$$V_\infty(e_k) \le J^*(e_k). \tag{30}$$

Based on (28) and (30), we can conclude that $J^*(e_k)$ is the limit of the cost function sequence $\{V_i\}$ as $i\to\infty$, i.e., $V_\infty(e_k) = J^*(e_k)$. □

From Theorems 1–3, we can obtain that the cost function sequence $\{V_i(e_k)\}$ converges to the optimal cost function $J^*(e_k)$ of the DTHJB equation, i.e., $V_i\to J^*$ as $i\to\infty$. Then, according to (12) and (16), we can conclude the convergence of the corresponding control law sequence. Now, we present the following corollary.

Corollary 1. *Define the cost function sequence $\{V_i\}$ as in (17) with $V_0(\cdot)=0$, and the control law sequence $\{v_i\}$ as in (16). If the system state e_k is controllable, then the sequence $\{v_i\}$ converges to the*

optimal control law u^* as $i\to\infty$, i.e.,

$$\lim_{i\to\infty} v_i(e_k) = u^*(e_k).$$

3.3. The ε-optimal control algorithm

According to Theorems 1–3 and Corollary 1, we should run the iterative ADP algorithm (14)–(17) until $i\to\infty$ to obtain the optimal cost function $J^*(e_k)$, and then to get a control vector $v_\infty(e_k)$ based on which we can construct a control sequence $\underline{u}_\infty(e_k)=(v_\infty(e_k),v_\infty(e_{k+1}),\ldots,v_\infty(e_{k+i}),\ldots)$ to control the state to reach the target. Obviously, $\underline{u}_\infty(e_k)$ has infinite length. Though it is feasible in terms of theory, it is always not practical to do so because most real world systems need to be effectively controlled within finite-horizon. Therefore, in this section, we will propose a novel ε-optimal control strategy using the iterative ADP algorithm to deal with the problem. The idea is, for a given error bound $\varepsilon>0$, the iterative number i will be chosen so that the error between $V_i(e_k)$ and $J^*(e_k)$ is within the bound.

Let $\varepsilon>0$ be any small number, e_k be any controllable state, and $J^*(e_k)$ be the optimal value of the cost function sequence defined as in (17). From Theorem 3, it is clear that there exists a finite i such that

$$|V_i(e_k)-J^*(e_k)|\le\varepsilon. \tag{31}$$

The length of the optimal control sequence starting from e_k with respect to ε is defined as

$$K_\varepsilon(e_k)=\min\{i:|V_i(e_k)-J^*(e_k)|\le\varepsilon\}. \tag{32}$$

The corresponding control law

$$\begin{aligned} v_{i-1}(e_k) &= \arg\min_{u_k}\{U(e_k,u_k)+V_{i-1}(e_{k+1})\}\\ &= -\frac{1}{2}R^{-1}g^T(e_k+r_k)\frac{\partial V_{i-1}(e_{k+1})}{\partial e_{k+1}} \end{aligned} \tag{33}$$

is called the ε-optimal control and is denoted as $\mu_\varepsilon^*(e_k)$.

In this sense, we can see that an error ε between $V_i(e_k)$ and $J^*(e_k)$ is introduced into the iterative ADP algorithm, which makes the cost function sequence $\{V_i(e_k)\}$ converge in finite number of iteration steps.

However, the optimal criterion (31) is difficult to verify because the optimal cost function $J^*(e_k)$ is unknown in general. Consequently, we will use an equivalent criterion, i.e.,

$$|V_i(e_k)-V_{i+1}(e_k)|\le\varepsilon \tag{34}$$

to replace (31).

In fact, if $|V_i(e_k)-J^*(e_k)|\le\varepsilon$ holds, we have $V_i(e_k)\le J^*(e_k)+\varepsilon$. Combining with $J^*(e_k)\le V_{i+1}(e_k)\le V_i(e_k)$, we can find that

$$0\le V_i(e_k)-V_{i+1}(e_k)\le\varepsilon,$$

which means

$$|V_i(e_k)-V_{i+1}(e_k)|\le\varepsilon.$$

On the other hand, according to Theorem 3, $|V_i(e_k)-V_{i+1}(e_k)|\to0$ connotes that $V_i(e_k)\to J^*(e_k)$. As a result, if $|V_i(e_k)-V_{i+1}(e_k)|\le\varepsilon$ holds for any given small ε, we can derive the conclusion that $|V_i(e_k)-J^*(e_k)|\le\varepsilon$ holds if i is sufficiently large.

3.4. Design procedure of the finite-horizon optimal tracking control scheme using iterative ADP algorithm

In this section, we will give the detailed design procedure for the finite-horizon nonlinear optimal tracking control scheme using the iterative ADP algorithm.

Step 1 Specify an error bound ε for the given initial state x_0. Choose $i_{\max}$, the reference trajectory r_k, and the matrices Q and R.
Step 2 Compute e_k according to (2) and (3).
Step 3 Set $i=0$, $V_0(e_k)=0$. Obtain the initial finite-horizon admissible vector $v_0(e_k)$ by (14) and update the cost function $V_1(e_k)$ by (15).
Step 4 Set $i=i+1$.
Step 5 Compute $v_i(e_k)$ by (16) and the corresponding cost function $V_{i+1}(e_k)$ by (17).
Step 6 If $|V_i(e_k)-V_{i+1}(e_k)|\le\varepsilon$, then go to Step 8; otherwise, go to Step 7.
Step 7 If $i>i_{\max}$, then go to Step 8; otherwise, go to Step 4.
Step 8 Stop.

After the optimal control law $u^*(e_k)$ for system (6) is derived under the given error bound ε, we can compute the optimal tracking control input for original system (1) by

$$u_{pk}^*=u^*(e_k)+u_{dk}=u^*(e_k)+g^{-1}(r_k)(\phi(r_k)-f(r_k)). \tag{35}$$

In the following section, we will describe the implementation of the iterative ADP algorithm based on NNs in detail.

4. NN implementation of the iterative ADP algorithm via HDP technique

Now, we implement the iterative HDP algorithm in (14)–(17) using NNs. In the iterative HDP algorithm, there are three networks, which are model network, critic network and action network. All the networks are chosen as three-layer feedforward NNs. The inputs of the critic network and the action network are e_k, while the inputs of the model network are e_k and $\hat{v}_i(e_k)$. The structure diagram of the iterative HDP algorithm is shown in Fig. 1.

4.1. The model network

The purpose of designing the model network is to approximate the error d ynamics. We should train the model network before carrying out the iterative HDP algorithm. For given e_k and $\hat{v}_i(e_k)$, we can obtain the output of the model network as

$$\hat{e}_{k+1}=\omega_m^T\sigma(v_m^Tz_k), \tag{36}$$

where

$$z_k=[e_k^T\ \hat{v}_i^T(e_k)]^T.$$

We define the error function of the model network as

$$e_{mk}=\hat{e}_{k+1}-e_{k+1}. \tag{37}$$

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

$$E_{mk}=\tfrac{1}{2}e_{mk}^Te_{mk}. \tag{38}$$

Using the gradient-based adaptation rule, the weights can be updated as

$$\omega_m(j+1)=\omega_m(j)-\alpha_m\left[\frac{\partial E_{mk}}{\partial\omega_m(j)}\right], \tag{39}$$

$$v_m(j+1)=v_m(j)-\alpha_m\left[\frac{\partial E_{mk}}{\partial v_m(j)}\right], \tag{40}$$

where $\alpha_m>0$ is the learning rate of the model network, and j is the iterative step for updating the weight parameters.

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

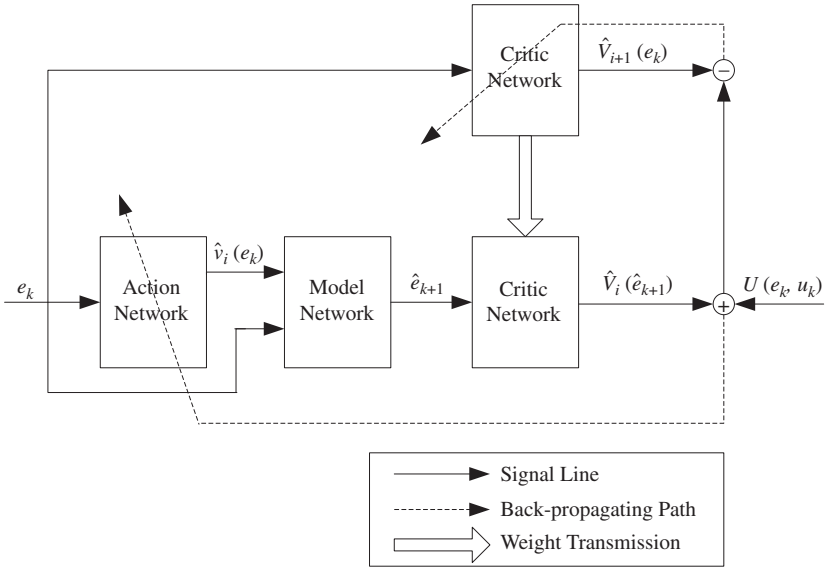


Fig. 1. The structure diagram of the iterative HDP algorithm.

4.2. The critic network

The critic network is used to approximate the cost function $V_i(e_k)$. The output of the critic network is denoted as

$$\hat{V}_i(e_k) = \omega_{ci}^T \sigma(v_{ci}^T e_k). \tag{41}$$

The target function can be written as

$$V_i(e_k) = e_k^T Q e_k + v_{i-1}^T(e_k) R v_{i-1}(e_k) + \hat{V}_{i-1}(\hat{e}_{k+1}). \tag{42}$$

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

$$e_{cik} = \hat{V}_i(e_k) - V_i(e_k). \tag{43}$$

The objective function to be minimized for the critic network is

$$E_{cik} = \tfrac{1}{2} e_{cik}^T e_{cik}. \tag{44}$$

The weight updating rule for training the critic network is also gradient-based adaptation given by

$$\omega_{ci}(j+1) = \omega_{ci}(j) - \alpha_c \left[\frac{\partial E_{cik}}{\partial \omega_{ci}(j)}\right], \tag{45}$$

$$v_{ci}(j+1) = v_{ci}(j) - \alpha_c \left[\frac{\partial E_{cik}}{\partial v_{ci}(j)}\right], \tag{46}$$

where $\alpha_c > 0$ is the learning rate of the critic network, and j is the inner-loop iterative step for updating the weight parameters.

4.3. The action network

In the action network, the state e_k is used as input to obtain the optimal control as the output of the network. The output can be formulated as

$$\hat{v}_i(e_k) = \omega_{ai}^T \sigma(v_{ai}^T e_k). \tag{47}$$

The target control input is given by

$$v_i(e_k) = -\frac{1}{2} R^{-1} g^T(e_k + r_k) \frac{\partial \hat{V}_i(\hat{e}_{k+1})}{\partial \hat{e}_{k+1}}. \tag{48}$$

The error function of the action network can be defined as

$$e_{aik} = \hat{v}_i(e_k) - v_i(e_k). \tag{49}$$

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

$$E_{aik} = \tfrac{1}{2} e_{aik}^T e_{aik}. \tag{50}$$

Similarly, the weight updating algorithm is

$$\omega_{ai}(j+1) = \omega_{ai}(j) - \alpha_a \left[\frac{\partial E_{aik}}{\partial \omega_{ai}(j)}\right], \tag{51}$$

$$v_{ai}(j+1) = v_{ai}(j) - \alpha_a \left[\frac{\partial E_{aik}}{\partial v_{ai}(j)}\right], \tag{52}$$

where $\alpha_a > 0$ is the learning rate of the action network, and j is the inner-loop iterative step for updating the weight parameters.

5. Simulation study

In this section, two simulation examples are provided to confirm the theoretical results.

5.1. Example 1

The first example is derived from [31] with some modifications. Consider the following nonlinear system:

$$x_{k+1} = f(x_k) + g(x_k) u_{pk}, \tag{53}$$

where $x_k = [x_{1k}\ x_{2k}]^T \in \mathbb{R}^2$ and $u_{pk} = [u_{p1k}\ u_{p2k}]^T \in \mathbb{R}^2$ are the state and control variables, respectively. The parameters of the cost function are chosen as $Q = 0.5I$ and $R = 2I$, where I denotes the identity matrix with suitable dimensions. The state of the controlled system is initialized to be $x_0 = [0.8\ -0.5]^T$. The system functions are given as

$$f(x_k) = \begin{bmatrix} \sin(0.5x_{2k}) x_{1k}^2 \\ \cos(1.4x_{2k}) \sin(0.9x_{1k}) \end{bmatrix},$$

$$g(x_k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The reference trajectory for the above system is selected as

$$r_k = \begin{bmatrix} \sin(0.25k) \\ \cos(0.25k) \end{bmatrix}.$$

We set the error bound of the iterative HDP algorithm as $\varepsilon = 10^{-5}$ and implement the algorithm at time instant $k = 0$. The initial control vector of system (6) can be computed as $v_0(e_0) = [0.64\ \sin(0.25) - \sin(0.72)\cos(0.7)]^T$, where $e_0 = [0.8\ -\ 0.5]^T$. Then, we choose three-layer feedforward NNs as model network, critic

network and action network with the structures 4–8–2, 2–8–1, 2–8–2, respectively. The initial weights of the three networks are all set to be random in $[-1,1]$. It should be mentioned that the model network should be trained first. We train the model network for 1000 steps using 500 data samples under the learning rate $\alpha_m = 0.1$. After the training of the model network is completed, the weights are kept unchanged. Then, we train the critic network and the action network for 20 iterations (i.e., for $i = 1,2,\ldots,20$) with each iteration of 2000 training steps to make sure the given error bound $\varepsilon = 10^{-5}$ is reached. In the training process, the learning rate $\alpha_c = \alpha_a = 0.05$. The convergence process of the cost function of the iterative HDP algorithm is shown in Fig. 2, for $k=0$. We can see that the iterative cost function sequence does converge to the optimal cost function quite rapidly, which indicates the effectiveness of the iterative HDP algorithm. Therefore, we have $|V_{19}(e_0)-V_{20}(e_0)| \le \varepsilon$, which means that the number of steps of the ε-optimal control is $K_\varepsilon(e_0) = 19$. Besides, the ε-optimal control law $\mu_\varepsilon^*(e_0)$ for system (6) can also be obtained during the iteration process.

Next, we compute the near-optimal tracking control law for original system (1) using (35) and apply it to the controlled system for 40 time steps. The obtained state curves are shown in

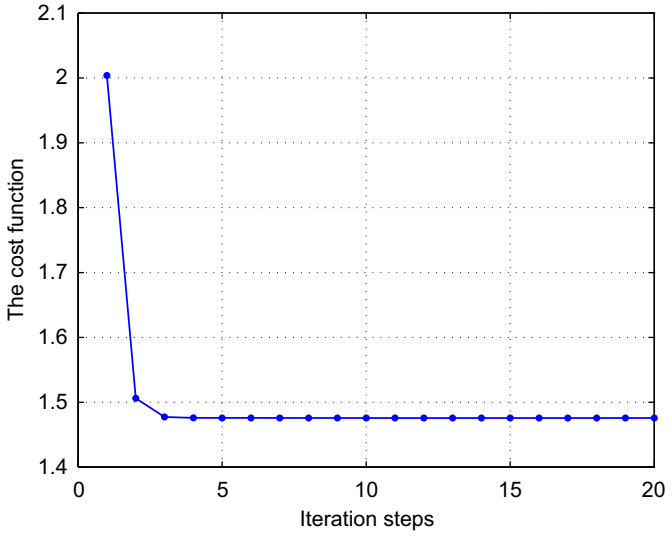


Fig. 2. The convergence process of the cost function.

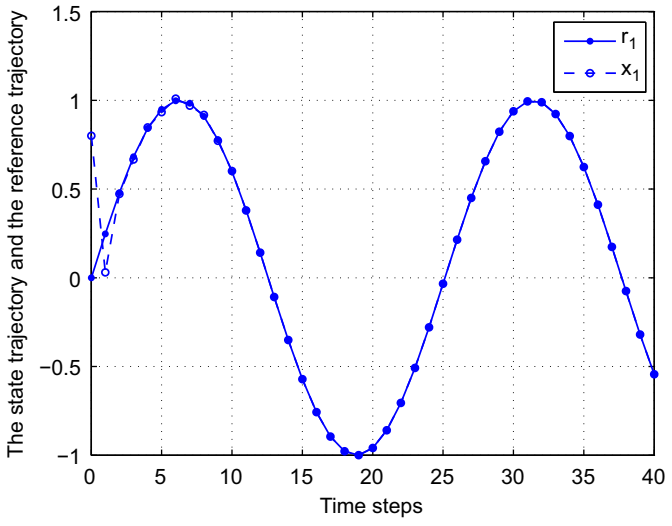


Fig. 3. The state trajectory x_1 and the reference trajectory r_1.

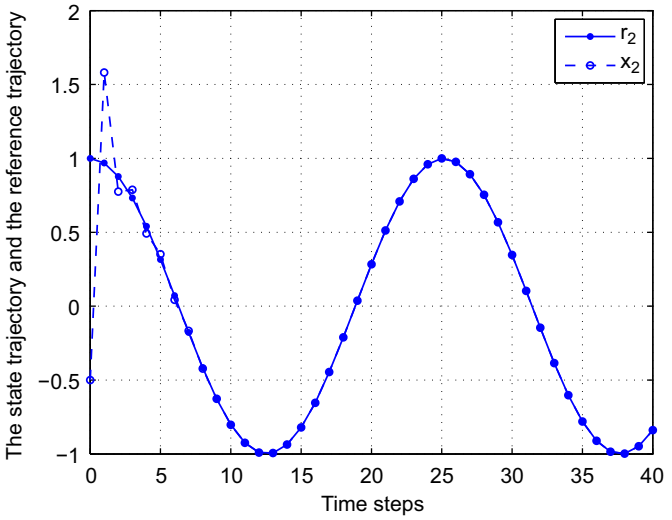


Fig. 4. The state trajectory x_2 and the reference trajectory r_2.

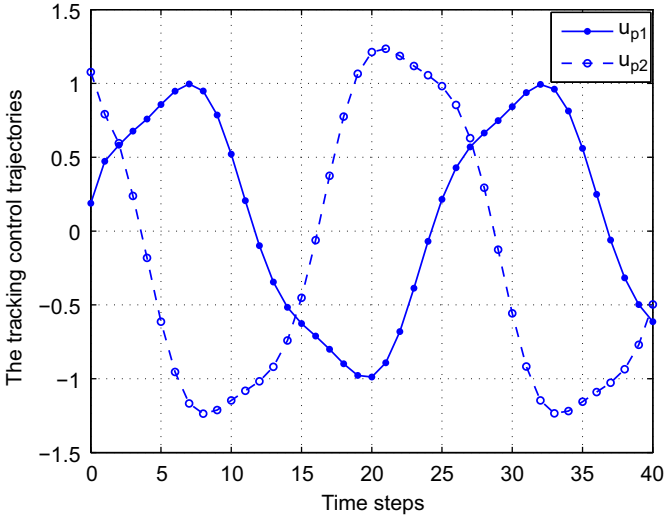


Fig. 5. The tracking control trajectories u_p.

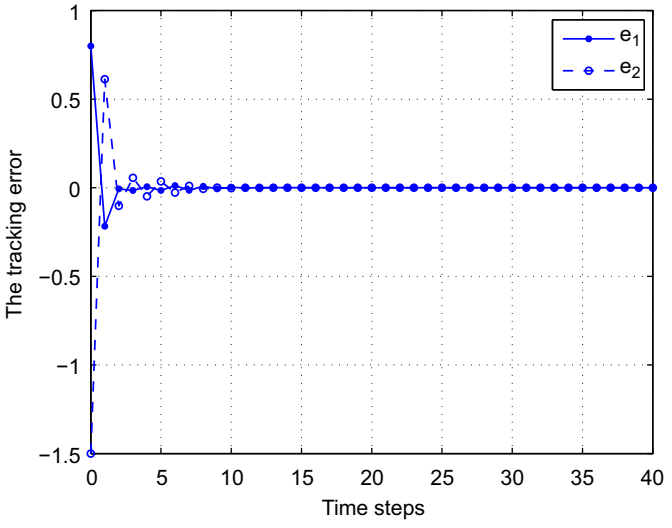


Fig. 6. The tracking error e.

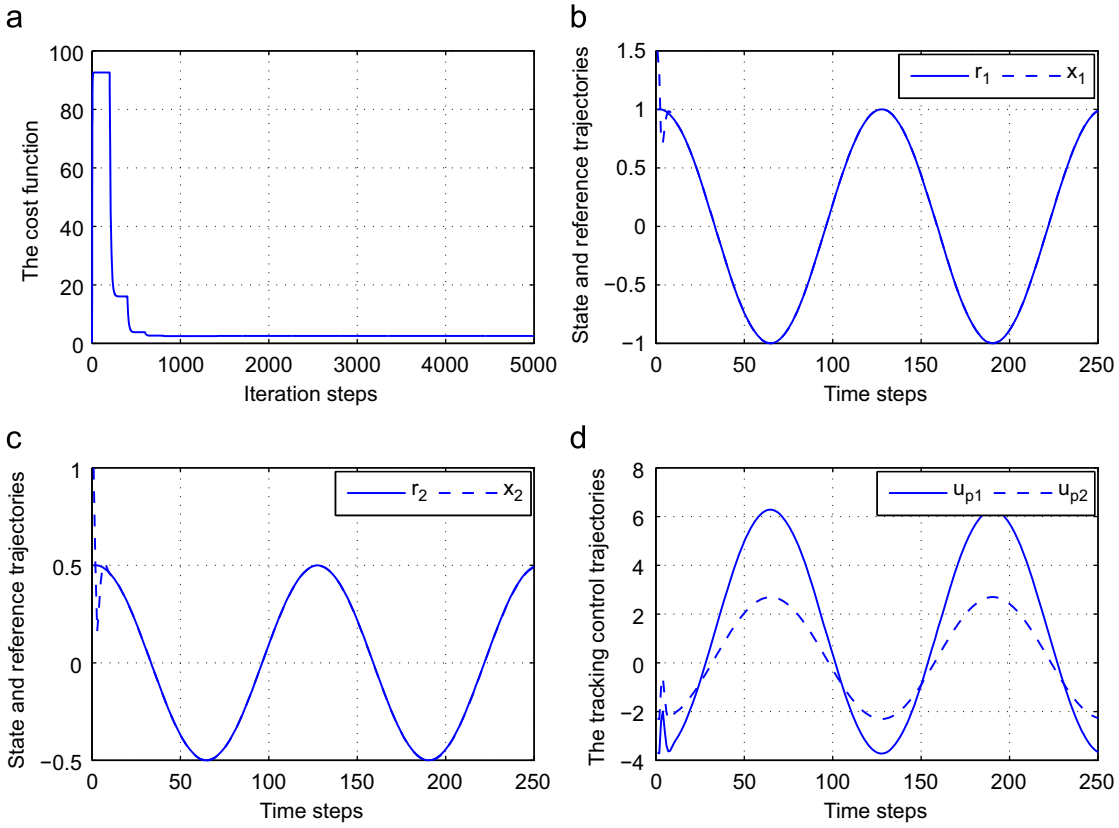


Fig. 7. Simulation results of Example 2.

Figs. 3 and 4, where the corresponding reference trajectories are also plotted to evaluate the tracking performance. The tracking control curves and the tracking errors are shown in Figs. 5 and 6, respectively. Besides, we can derive that the tracking error becomes $e_{19}=[0.2778\times10^{-5}-0.8793\times10^{-5}]^T$ after 19 time steps. These simulation results verify the excellent performance of the tracking controller developed by the iterative ADP algorithm.

5.2. Example 2

The second example is obtained from [32]. Consider the nonlinear discrete-time system described as (53) where

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

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

The desired trajectory is set to

$$r_k=\begin{bmatrix}\sin(k+0.5\pi)\\0.5\cos k\end{bmatrix}.$$

In the implementation of the iterative HDP algorithm, the initial weights and structures of three networks are set the same as Example 1. Then, for the given initial state $x_0=[1.5\ 1]^T$, we train the model network for 10 000 steps using 1000 data samples under the learning rate $\alpha_m=0.05$. Besides, the critic network and the action network are trained for 5000 iterations so that the given error bound $\varepsilon=10^{-6}$ is reached. The learning rate in the training process is also $\alpha_c=\alpha_a=0.05$.

The convergence process of the cost function of the iterative HDP algorithm is shown in Fig. 7(a), for $k=0$. Then, we apply the tracking control law to the system for 250 time steps and obtain the state and reference trajectories shown in Fig. 7(b) and (c). Besides, the tracking control curves are given in Fig. 7(d). It is clear from the simulation results that the iterative HDP algorithm proposed in this paper is very effective in solving the finite-horizon tracking control problems.

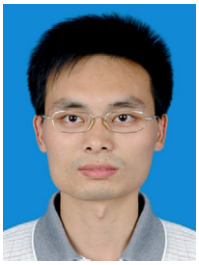
6. Conclusion

An effective method is proposed in this paper to design the finite-horizon near-optimal tracking controller for a class of discrete-time nonlinear systems. The iterative ADP algorithm is introduced to solve the cost function of the DTHJB equation with convergence analysis, which obtains a finite-horizon near-optimal tracking controller that makes the cost function close to its optimal value within an ε-error bound. Three NNs are used to approximate the cost function, the control law, and the nonlinear system, respectively. The simulation examples confirmed the validity of the tracking control approach. The strategy presented in this paper only be true of a class of affine nonlinear systems and requires complete knowledge of the system dynamics. Though there are many practical systems for which the approach can be applied, it is necessary to broaden its applicability for a more general class of nonlinear systems. Consequently, our future work includes studying the optimal tracking control problems for non-affine nonlinear systems and model-free systems.

References

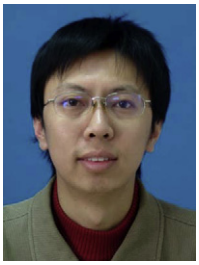
[1] L. Cui, H. Zhang, B. Chen, Q. Zhang, Asymptotic tracking control scheme for mechanical systems with external disturbances and friction, Neurocomputing 73 (2010) 1293–1302.

[2] S. Devasia, D. Chen, B. Paden, Nonlinear inversion-based output tracking, IEEE Trans. Autom. Control 41 (1996) 930–942.
[3] G. Tang, Y. Liu, Y. Zhang, Approximate optimal output tracking control for nonlinear discrete-time systems, Control Theory Appl. 27 (2010) 400–405.
[4] F.L. Lewis, V.L. Syrmos, Optimal Control, Wiley, New York, 1995.
[5] R.E. Bellman, Dynamic Programming, Princeton University Press, Princeton, NJ, 1957.
[6] T. Poggio, F. Girosi, Networks for approximation and learning, Proc. IEEE 78 (1990) 1481–1497.
[7] S. Jagannathan, Neural Network Control of Nonlinear Discrete-time Systems, CRC Press, Boca Raton, FL, 2006.
[8] W. Yu, Recent Advances in Intelligent Control Systems, Springer-Verlag, London, 2009.
[9] P.J. Werbos, Approximate dynamic programming for real-time control and neural modeling, in: White, D.A., Sofge, D.A. (Eds.), Handbook of Intelligent Control, Van Nostrand Reinhold, New York, 1992 (Chapter 13).
[10] P.J. Werbos, Intelligence in the brain: a theory of how it works and how to build it, Neural Networks 22 (2009) 200–212.
[11] P.J. Werbos, Advanced forecasting methods for global crisis warning and models of intelligence, General Syst. Yearb. 22 (1977) 25–38.
[12] J.J. Murray, C.J. Cox, G.G. Lendaris, R. Saeks, Adaptive dynamic programming, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 32 (2002) 140–153.
[13] F.Y. Wang, H. Zhang, D. Liu, Adaptive dynamic programming: an introduction, IEEE Comput. Intell. Mag. 4 (2009) 39–47.
[14] F.L. Lewis, D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, IEEE Circuits Syst. Mag. 9 (2009) 32–50.
[15] J. Si, A.G. Barto, W.B. Powell, D.C. Wunsch (Eds.), Handbook of Learning and Approximate Dynamic Programming, IEEE Press, Wiley, New York, 2004.
[16] A. Al-Tamimi, F.L. Lewis, M. Abu-Khalaf, Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof, IEEE Trans. Syst. Man Cybern. Part B Cybern. 38 (2008) 943–949.
[17] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, Belmont, MA, 1996.
[18] J. Si, Y.T. Wang, On-line learning control by association and reinforcement, IEEE Trans. Neural Networks 12 (2001) 264–276.
[19] D.V. Prokhorov, D.C. Wunsch, Adaptive critic designs, IEEE Trans. Neural Networks 8 (1997) 997–1007.
[20] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, The MIT Press, Cambridge, MA, 1998.
[21] D. Liu, X. Xiong, Y. Zhang, Action-dependent adaptive critic designs, in: Proceedings of International Joint Conference on Neural Networks, Washington, DC, July 2001, pp. 990–995.
[22] D. Liu, Y. Zhang, H. Zhang, A self-learning call admission control scheme for CDMA cellular networks, IEEE Trans. Neural Networks 16 (2005) 1219–1228.
[23] F.Y. Wang, N. Jin, D. Liu, Q. Wei, Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε-error bound, IEEE Trans. Neural Networks 22 (2011) 24–36.
[24] S.N. Balakrishnan, V. Biega, Adaptive-critic based neural networks for aircraft optimal control, J. Guidance Control Dyn. 19 (1996) 893–898.
[25] S.N. Balakrishnan, J. Ding, F.L. Lewis, Issues on stability of ADP feedback controllers for dynamic systems, IEEE Trans. Syst. Man Cybern. Part B Cybern. 38 (2008) 913–917.
[26] G.K. Venayagamoorthy, R.G. Harley, D.C. Wunsch, Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator, IEEE Trans. Neural Networks 13 (2002) 764–773.
[27] G.K. Venayagamoorthy, R.G. Harley, D.C. Wunsch, Implementation of adaptive critic-based neurocontrollers for turbogenerators in a multimachine power system, IEEE Trans. Neural Networks 14 (2003) 1047–1064.
[28] M. Abu-Khalaf, F.L. Lewis, Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach, Automatica 41 (2005) 779–791.
[29] T. Cheng, F.L. Lewis, M. Abu-Khalaf, A neural network solution for fixed-final time optimal control of nonlinear systems, Automatica 43 (2007) 482–490.
[30] H. Zhang, Y. Luo, D. Liu, Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints, IEEE Trans. Neural Networks 20 (2009) 1490–1503.
[31] T. Dierks, S. Jagannathan, Optimal tracking control of affine nonlinear discrete-time systems with unknown internal dynamics, in: Proceedings of Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, Shanghai, PR China, December 2009, pp. 6750–6755.
[32] H. Zhang, Q. Wei, Y. Luo, A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm, IEEE Trans. Syst. Man Cybern. Part B Cybern. 38 (2008) 937–942.
[33] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, F.L. Lewis, Adaptive optimal control for continuous-time linear systems based on policy iteration, Automatica 45 (2009) 477–484.
[34] K.G. Vamvoudakis, F.L. Lewis, Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (2010) 878–888.
[35] R. Song, H. Zhang, Y. Luo, Q. Wei, Optimal control laws for time-delay systems with saturating actuators based on heuristic dynamic programming, Neurocomputing 73 (2010) 3020–3027.
[36] Y.M. Park, M.S. Choi, K.Y. Lee, An optimal tracking neuro-controller for nonlinear dynamic systems, IEEE Trans. Neural Networks 7 (1996) 1099–1110.



Ding Wang received the B.S. degree in mathematics from Zhengzhou University of Light Industry, Zhengzhou, China, and the M.S. degree in operational research and cybernetics from Northeastern University, Shenyang, China, in 2007 and 2009, respectively. He is currently working toward the Ph.D. degree in the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include adaptive dynamic programming, neural networks, and intelligent control.

Derong Liu received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1994. Dr. Liu was a Staff Fellow with General Motors Research and Development Center, Warren, MI, from 1993 to 1995. He was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, from 1995 to 1999. He joined the University of Illinois at Chicago in 1999, where he became a Full Professor of Electrical and Computer Engineering and of Computer Science in 2006. He was selected for the "100 Talents Program" by the Chinese Academy of Sciences in 2008. Dr. Liu was an Associate Editor of the IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications (1997–1999), the IEEE Transactions on Signal Processing (2001–2003), the IEEE Transactions on Neural Networks (2004–2009), the IEEE Computational Intelligence Magazine (2006–2009), and the IEEE Circuits and Systems Magazine (2008–2009), and the Letters Editor of the IEEE Transactions on Neural Networks (2006–2008). Currently, he is the Editor-in-Chief of the IEEE Transactions on Neural Networks and an Associate Editor of the IEEE Transactions on Control Systems Technology. He received the Michael J. Birck Fellowship from the University of Notre Dame (1990), the Harvey N. Davis Distinguished Teaching Award from Stevens Institute of Technology (1997), the Faculty Early Career Development (CAREER) Award from the National Science Foundation (1999), the University Scholar Award from University of Illinois (2006), and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China (2008).



Qinglai Wei received the B.S. degree in automation, the M.S. degree in control theory and control engineering, and the Ph.D. degree in control theory and control engineering, from the Northeastern University, Shenyang, China, in 2002, 2005, and 2008, respectively. He is currently a postdoctoral fellow with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include neural-networks-based control, nonlinear control, adaptive dynamic programming, and their industrial applications.