Neurocomputing 149 (2015) 106–115

Contents lists available at ScienceDirect

Neurocomputing

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

Neural-network-based adaptive optimal tracking control scheme for discrete-time nonlinear systems with approximation errors

Qinglai Wei*, Derong Liu

The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 15 June 2013
Received in revised form
30 August 2013
Accepted 17 September 2013
Available online 1 August 2014

Keywords:
Adaptive dynamic programming
Adaptive critic designs
Approximate dynamic programming
Value iteration
Approximation errors
Optimal tracking control

ABSTRACT

In this paper, a new infinite horizon neural-network-based adaptive optimal tracking control scheme for discrete-time nonlinear systems is developed. The idea is to use iterative adaptive dynamic programming (ADP) algorithm to obtain the iterative tracking control law which makes the iterative performance index function reach the optimum. When the iterative tracking control law and iterative performance index function in each iteration cannot be accurately obtained, the convergence criteria of the iterative ADP algorithm are established according to the properties with finite approximation errors. If the convergence conditions are satisfied, it shows that the iterative performance index functions can converge to a finite neighborhood of the lowest bound of all performance index functions. Properties of the finite approximation errors for the iterative ADP algorithm are also analyzed. Neural networks are used to approximate the performance index function and compute the optimal control policy, respectively, for facilitating the implementation of the iterative ADP algorithm. Convergence properties of the neural network weights are proven. Finally, simulation results are given to illustrate the performance of the developed method.

1. Introduction

Optimal tracking control problems of nonlinear systems have always been the key focus in the control field in the latest several decades. Traditional optimal tracking control is mostly implemented by feedback linearization [1]. However, the controller designed by feedback linearization technique is only effective in the neighborhood of the equilibrium point. When the required operating range is large, the nonlinearities in the system cannot be properly compensated by using a linear model. Hence the control performance of feedback linearization technique is usually unsatisfied and the nonlinear controller design of the optimal tracking control is necessary. The difficulty for nonlinear optimal feedback control lies in solving the time-varying HJB equation which is usually too difficult to solve analytically. To overcome the difficulty, many approximation methods are proposed to obtain optimal tracking control law [2–5]. Among these approximate approaches, adaptive dynamic programming (ADP) algorithm, proposed by Werbos [6,7], has played an important role in seeking approximate solutions of dynamic programming problems as a way to solve the computational issue forward-in-time [8–14]. There are several synonyms used for ADP including "adaptive critic designs" [15], "adaptive dynamic programming" [16,17], "approximate dynamic programming" [18,19], "neural dynamic programming" [20], "neuro-dynamic programming" [21], and "reinforcement learning" [22]. In Werbos [19], ADP approaches were classified into four main schemes: Heuristic Dynamic Programming (HDP), Dual Heuristic Programming (DHP), Action Dependent HDP (ADHDP) (also known as Q-learning [23]), and Action Dependent DHP (ADDHP). In [15], two more ADP that are Globalized-DHP (GDHP) and ADGDHP were proposed. Iterative methods are also used in ADP to obtain the solution of HJB equation indirectly and have received lots of attentions [24–32].There are two main iterative ADP algorithms which are based on policy iteration and value iteration [33].

Policy iteration algorithm for optimal control of continuous-time systems with continuous state and action spaces was given in [34]. In [16], Murray et al. studied the deterministic continuous-time stabilizable systems where an iterative process was proposed to find the optimal control law by starting from an arbitrary admissible control law. In the policy iteration algorithms of ADP, to obtain the iterative performance index functions and iterative control laws, an initial admissible control law of the system is required. But, unfortunately, the admissible control law for nonlinear systems is also difficult to obtain. Thus, the initial conditions for the controller greatly limit the applications of the policy

* Corresponding author.
E-mail addresses: qinglai.wei@ia.ac.cn (Q. Wei), derong.liu@ia.ac.cn (D. Liu).

http://dx.doi.org/10.1016/j.neucom.2013.09.069
0925-2312/

iteration algorithms. Value iteration algorithm of optimal control for discrete-time nonlinear systems was given in [35]. In [18], Al-Tamimi et al. studied the deterministic discrete-time affine nonlinear systems

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

where x_k is the system state and u_k is the system control. Functions $f(x_k)$ and $g(x_k)$ denote system functions. In [18], the performance index function is defined by

$$J(x_k)=\sum_{j=k}^{\infty}(x_j^TQx_j+u_j^TRu_j), \tag{2}$$

where $Q\in\mathbb{R}^{n\times n}$ and $R\in\mathbb{R}^{m\times m}$ are positive definite matrices. In [18], a value iteration algorithm, which was referred to as HDP, was proposed for finding the optimal control law. It starts from $V_0(x_k)\equiv 0$, and then the iteration

$$\begin{cases} u_i(x_k)=-\dfrac{1}{2}R^{-1}g(x_k)^T\dfrac{\partial V_i(x_{k+1})}{\partial x_{k+1}}, \\ V_{i+1}(x_k)=x_k^TQx_k+u_i^T(x_k)Ru_i(x_k)+V_i(x_{k+1}), \end{cases} \tag{3}$$

is introduced for $i=0,1,2,\ldots$, where $x_{k+1}=f(x_k)+g(x_k)u_i(x_k)$. It was proven that $V_i(x_k)$ is nondecreasing and upper bounded, and hence converges to $J^*(x_k)$ as i increases to infinity. In 2008, Zhang et al. [36] applied value iteration of ADP to solve optimal tracking problems for nonlinear systems. Liu et al. [37] realized the value iteration of ADP by GDHP. Although iterative ADP algorithms attract more and more attentions [38–47], for most of the iterative ADP algorithms, the iterative control of each iteration is required to be accurately obtained. These iterative ADP algorithms can be called "accurate iterative ADP algorithms".

For most real-world control systems, however, the accurate iterative control laws in the iterative ADP algorithms cannot be obtained. As approximation structures are used to achieve the optimal control law and the performance index function, there must exist approximation errors between the approximation functions and the expected ones. This shows that the convergence properties in the accurate iterative ADP algorithms may be invalid for the iterative ADP with approximation errors. Till now, the discussion on the convergence properties of the iterative ADP algorithms with approximation errors is very little. Only in [41], based on iterative θ-ADP algorithm, an optimal regulation control scheme with approximation errors was proposed for discrete-time nonlinear systems, while in [41], the convergence of neural network weights is not analyzed. To the best of our knowledge, there are no discussions on the ADP algorithm for optimal tracking control problems with approximation errors.

In this paper, we will develop a new iterative ADP scheme for infinite horizon optimal tracking control problems. The main contribution of this paper is that the optimal tracking control problems with finite approximation errors are solved effectively using the present iterative ADP algorithms. A convergence analysis of the performance index function is developed and the least upper bound of the converged iterative performance index function is also presented. The convergence criteria are obtained. In order to facilitate the implementation of the iterative ADP algorithms, we use neural networks to obtain the iterative performance index function and the optimal tracking control policy, respectively. The convergence properties of the neural network weights are proven to guarantee the effectiveness of the neural network applications. Finally, simulation results are given to show the effectiveness of the developed iterative ADP algorithm.

The rest of this paper is organized as follows. In Section 2, the problem formulation is presented. In Section 3, the iterative ADP algorithm for the optimal tracking control problem is derived. The convergence criteria for the iterative ADP algorithm is also analyzed in this section. In Section 4, the neural network implementation with convergence proof for the optimal control scheme is discussed. In Section 5, numerical results and analysis are presented to demonstrate the effectiveness of the developed optimal control scheme. Finally, in Section 6, the conclusion is drawn and our future work will be declared.

2. Problem formulation

Consider a class of affine nonlinear systems of the form:

$$x(k+1)=f(x(k))+g(x(k))u(x(k)) \tag{4}$$

where $x(k)\in\Re^n, f(x(k))\in\Re^n, g(x(k))\in\Re^{n\times m}$, the input $u(k)\in\Re^m$ and $g(\cdot)$ has a generalized inverse. Here, assume that the system is controllable on $\Omega\subset\Re^n$. For infinite-time optimal tracking problem, the control objective is to design optimal control $u(x(k))$ for system (4) such that the state $x(k)$ track the specified desired trajectory $\eta(k)\in\Re^n$, $k=0,1,\ldots$. Define the tracking error as

$$z(k)=x(k)-\eta(k). \tag{5}$$

Define the following quadratic performance index:

$$J(z(0),\underline{u}_0)=\sum_{k=0}^{\infty}\{z^T(k)Qz(k)+(u(k)-u_e(k))^TR(u(k)-u_e(k))\} \tag{6}$$

where $Q\in\Re^{n\times n}$ and $R\in\Re^{m\times m}$ are positive definite matrices and $\underline{u}_0=(u(0),u(1),\ldots)$. Let

$$U(z(k),v(k))=z^T(k)Qz(k)+v^T(k)Rv(k)$$

be the utility function, where $v(k)=u(k)-u_e(k)$. Let $u_e(k)$ denote the expected control introduced for analytical purpose, which can be given as

$$u_e(k)=g^{-1}(\eta(k))(\eta(k+1)-f(\eta(k))) \tag{7}$$

where $g^{-1}(\eta(k))g(\eta(k))=I$ and $I\in\Re^{m\times m}$ is the identity matrix. Combining (4) and (5), we can get

$$\begin{aligned} z(k+1)=&f(z(k)+\eta(k))-\eta(k+1)+g(z(k)+\eta(k)) \\ &\times(v(k)+g^{-1}(\eta(k))(f(\eta(k))-\eta(k+1))). \end{aligned} \tag{8}$$

We will study optimal tracking control problems for (4). The goal of this paper is to find an optimal tracking control scheme which tracks the desired trajectory $\eta(k)$ and simultaneously minimizes the performance index function (6). The optimal performance index function is defined as

$$J^*(z(k))=\inf_{\underline{v}_k}\{J(z(k),\underline{v}_k)\}, \tag{9}$$

where $\underline{v}_k=(v(k),v(k+1),\ldots)$. According to Bellman's principle of optimality, $J^*(z(k))$ satisfies the discrete-time HJB equation:

$$J^*(z(k))=\inf_{v(k)}\{U(z(k),v(k))+J^*(F(z(k),v(k)))\}. \tag{10}$$

Then, the law of optimal single control vector can be expressed as

$$v^*(z(k))=\arg\inf_{v(k)}\{U(z(k),v(k))+J^*(z(k+1))\}. \tag{11}$$

Hence, the HJB equation (10) can be written as

$$J^*(z(k))=U(z(k),v^*(z(k)))+J^*(z(k+1)). \tag{12}$$

In [36], based on the greedy HDP iteration technique, the performance index and control policy are updated by iterations, with the iteration number i increasing from 0 to ∞. First, the initial performance index $V_0(z(k))\equiv 0$. Then, for $i=0,1,\ldots$, the control $v_i(k)$ and $V_{i+1}(z(k))$ are computed by the following two equations:

$$v_i(k)=\arg\min_{v(k)}\{z^T(k)Qz(k)+v^T(k)Rv(k)+V_i(z(k+1))\} \tag{13}$$

and

$$V_{i+1}(z(k)) = \min_{v(k)}\{z^T(k)Qz(k)+v^T(k)Rv(k)+V_i(z(k+1))\} = z^T(k)Qz(k)+v_i^T(z(k))Rv_i(z(k))+V_i(z(k+1)). \quad (14)$$

In [36], it was proven that the iterative performance index function $V_i(z(k))$ converges to $J^*(z(k))$, as $i\to\infty$. For the greedy HDP iteration algorithm, we can see that for $\forall i=0,1,\ldots$, the accurate iterative control law must be obtained in order to guarantee the convergence of the iterative performance index function. In the real-world implementation, however, for $\forall i=0,1,\ldots$, the accurate iterative control law $v_i(z(k))$ and the iterative performance index function $V_i(z(k))$ are generally impossible to obtain without any errors. In this situation, the convergence of the iterative performance index function and iterative control law may be invalid and the iterative ADP algorithm may even be divergent. To overcome this difficulty, a new ADP analysis method must be developed considering the approximation errors.

3. Iterative ADP algorithm for nonlinear optimal tracking control with finite approximation errors

3.1. Derivation of the iterative ADP algorithm with finite approximation errors

In the developed iterative ADP algorithm, the performance index function and control law are updated by iterations, with the iteration index i increasing from 0 to infinity. For $i=0$, let $V_0(z(k))=0$. The iterative control law $\hat{v}_0(z(k))$ can be computed as follows:

$$\hat{v}_0(z(k)) = \arg\min_{v(k)}\{z^T(k)Qz(k)+v^T(k)Rv(k)+\hat{V}_0(z(k+1))\}+\rho_0(z(k)) \quad (15)$$

where $V_0(z(k+1))=\hat{V}_0(z(k+1))$. The performance index function can be updated as

$$\hat{V}_1(z(k)) = z^T(k)Qz(k)+\hat{v}_0^T(z(k))R\hat{v}_0(z(k))+\hat{V}_0(z(k+1))+\pi_0(z(k)). \quad (16)$$

For $i=1,2,\ldots$, the iterative ADP algorithm will iterate between

$$\hat{v}_i(z(k)) = \arg\min_{v(k)}\{z^T(k)Qz(k)+v^T(k)Rv(k)+\hat{V}_i(z(k+1))\}+\rho_i(z(k)) \quad (17)$$

and

$$\hat{V}_{i+1}(z(k)) = \min_{v(k)}\{z^T(k)Qz(k)+v^T(k)Rv(k)+\hat{V}_i(z(k+1))\}+\pi_i(z(k)) = z^T(k)Qz(k)+\hat{v}_i^T(z(k))R\hat{v}_i(z(k))+\hat{V}_i(z(k+1))+\pi_i(z(k)). \quad (18)$$

3.2. Properties of the iterative ADP algorithm with finite approximation errors

From the iterative ADP algorithms (15)–(18), we can see that for $\forall i=0,1,\ldots$, there exists an approximation error between the iterative performance index functions $\hat{V}_i(z(k))$ and $V_i(z(k))$. As the accurate iterative control law $v_i(z(k))$ cannot be obtained which means the iterative performance index functions $V_i(z(k))$ cannot be obtained, the accurate value of each iterative error is unknown and nearly impossible to obtain. It makes the property analysis of the iterative performance index function $\hat{V}_i(z(k))$ and iterative control law $\hat{v}_i(z(k))$ very difficult. So, in this subsection, a new "error bound" analysis method is developed. The idea of the "error bound" analysis method is that for each iterative index $i=0,1\ldots$, the least upper bound of the iterative performance index functions $\hat{V}_i(z(k))$ is analyzed, which avoids to analyze the value of $\hat{V}_i(z(k))$ directly. Using the "error bound" method, it can be proven that the iterative performance index functions $\hat{V}_i(z(k))$ can uniformly converge to a finite neighborhood of optimal performance index function.

Define a new iterative performance index function as

$$\Gamma_i(z(k)) = \min_{v(k)\in\mathbb{R}^m}\{U(z(k),v(k))+\hat{V}_{i-1}(z(k+1))\} \quad (19)$$

where $\hat{V}_i(z(k))$ is defined in (18) and $v(k)$ can accurately be obtained in $\mathbb{R}^m$. Then, for $\forall i=0,1,\ldots$, there exists a finite constant $\sigma\ge 1$ that makes

$$\hat{V}_i(z(k)) \le \sigma\Gamma_i(z(k)) \quad (20)$$

hold uniformly. Hence, we can give the following theorem.

Theorem 1. *For $\forall\, i=0,1,\ldots$, let $\Gamma_i(z(k))$ be expressed as in (19) and $\hat{V}_i(z(k))$ be expressed as in (18). Let $\gamma<\infty$ and $1\le\delta<\infty$ be both constants that make*

$$J^*(z(k+1)) \le \gamma U(z(k),v(k)) \quad (21)$$

and

$$V_0(z(k)) \le \delta J^*(z(k)) \quad (22)$$

hold uniformly. If there exists $1\le\sigma<\infty$ that makes (20) hold uniformly, then we have

$$J^*(z(k)) \le \hat{V}_i(z(k)) \le \sigma\left(1+\sum_{j=1}^{i}\frac{\gamma^j\sigma^{j-1}(\sigma-1)}{(\gamma+1)^j}+\frac{\gamma^i\sigma^i(\delta-1)}{(\gamma+1)^i}\right)J^*(z(k)), \quad (23)$$

where we define $\sum_j^i(\cdot)=0$, for $\forall j>i$ and $i,j=0,1,\ldots$.

Proof. The theorem can be proven by mathematical induction. First, let $i=0$. Then, (23) becomes

$$J^*(z(k)) \le \hat{V}_0(z(k)) \le \sigma\delta J^*(z(k)). \quad (24)$$

As $\hat{V}_0(z(k))\le\delta J^*(z(k))$, then we can obtain $\hat{V}_0(z(k))\le\delta J^*(z(k))\le\sigma\delta J^*(z(k))$, which obtains (24). So, the conclusion holds for $i=0$.

Next, let $i=1$. We have

$$\begin{aligned}\Gamma_1(z(k)) &= \min_{v(k)}\left\{U(z(k),v(k))+\hat{V}_0(F(z(k),v(k)))\right\}\\ &\le \min_{v(k)}\{U(z(k),v(k))+\sigma\delta J^*(F(z(k),v(k)))\}\\ &\le \min_{v(k)}\left\{\left(1+\gamma\frac{\sigma\delta-1}{\gamma+1}\right)U(z(k),v(k))+\left(\sigma\delta-\frac{\sigma\delta-1}{\gamma+1}\right)J^*(F(z(k),v(k)))\right\}\\ &= \left(1+\frac{\gamma(\sigma-1)}{\gamma+1}+\frac{\gamma\sigma(\delta-1)}{\gamma+1}\right)J^*(z(k)).\end{aligned} \quad (25)$$

According to (20), we can obtain

$$\hat{V}_1(z(k)) \le \sigma\left(1+\frac{\gamma(\sigma-1)}{\gamma+1}+\frac{\gamma\sigma(\delta-1)}{\gamma+1}\right)J^*(z(k)), \quad (26)$$

which shows that (23) holds for $i=1$.

Assume that (23) holds for $i=l-1$, $l=1,2,\ldots$. Then, for $i=l$, we have

$$\begin{aligned}\Gamma_l(z(k)) &= \min_{v(k)}\left\{U(z(k),v(k))+\hat{V}_{l-1}(F(z(k+1)))\right\}\\ &\le \min_{v(k)}\left\{U(z(k),v(k))+\sigma\left(1+\sum_{j=1}^{l}\frac{\gamma^j\sigma^{j-1}(\sigma-1)}{(\gamma+1)^j}+\frac{\gamma^l\sigma^l(\delta-1)}{(\gamma+1)^i}\right)\times J^*(z(k))\right\}\end{aligned}$$

$$\leq \left(1+\sum_{j=1}^{l}\frac{\gamma^j\sigma^{j-1}(\sigma-1)}{(\gamma+1)^j}+\frac{\gamma^l\sigma^l(\delta-1)}{(\gamma+1)^l}\right) \times\min_{v(k)}\{U(z(k),v(k))+J^*(z(k+1))\} = \left(1+\sum_{j=1}^{l}\frac{\gamma^j\sigma^{j-1}(\sigma-1)}{(\gamma+1)^j}+\frac{\gamma^l\sigma^l(\delta-1)}{(\gamma+1)^l}\right)J^*(z(k)). \quad (27)$$

Then, according to (20), we can obtain (23) which proves the conclusion for $\forall\, i=0,1,\ldots$.

From (23), we can see that for an arbitrary finite i, there exists a bounded error between the iterative performance index function $\hat{V}_i(z(k))$ and the optimal performance index function $J^*(z(k))$. As $i\to\infty$, the bound of the approximation error may increase to infinity. Thus, in the following, we will give the convergence properties of the iterative ADP algorithms (15)–(18) using error bound method. □

Theorem 2. *Suppose that Theorem 1 holds for $\forall z(k)\in\mathbb{R}^n$. If for $\gamma<\infty$ and $\sigma\geq 1$, the inequality*

$$\sigma<\frac{\gamma+1}{\gamma} \quad (28)$$

holds, then as $i\to\infty$, the iterative performance index function $\hat{V}_i(z(k))$ in the iterative ADP algorithms (15)–(18) is uniformly convergent to a bounded neighborhood of the optimal performance index function $J^*(z(k))$, i.e.,

$$\lim_{i\to\infty}\hat{V}_i(z(k))=\hat{V}_\infty(z(k))\leq\sigma\left(1+\frac{\gamma(\sigma-1)}{1-\gamma(\sigma-1)}\right)J^*(z(k)). \quad (29)$$

Proof. According to (27) in Theorem 1, we can see that for $j=1,2,\ldots$, the sequence $\{\gamma^j\sigma^{j-1}(\sigma-1)/(\gamma+1)^j\}$ is a geometrical series. Then, (27) can be written as

$$\Gamma_i(z(k))\leq\left(1+\frac{\frac{\gamma(\sigma-1)}{\gamma+1}\left(1-\left(\frac{\gamma\sigma}{\gamma+1}\right)^i\right)}{1-\frac{\gamma\sigma}{\gamma+1}}+\frac{\gamma^i\sigma^i(\delta-1)}{(\gamma+1)^i}\right)J^*(z(k)). \quad (30)$$

As $i\to\infty$, if $1\leq\sigma<(\gamma+1)/\gamma$, then (30) becomes

$$\lim_{i\to\infty}\Gamma_i(z(k))=\Gamma_\infty(z(k))\leq\left(1+\frac{\gamma(\sigma-1)}{1-\gamma(\sigma-1)}\right)J^*(z(k)). \quad (31)$$

According to (20), letting $i\to\infty$, then we have

$$\hat{V}_\infty(z(k))\leq\sigma\Gamma_\infty(z(k)). \quad (32)$$

According to (31) and (32), we can obtain (29).

Remark 1. From (28), we can see that the condition is not easy for σ to satisfy. It requires a very accurate training result of critic neural networks as indicated by (20).

Corollary 1. *Suppose that Theorem 1 holds for $\forall z(k)\in\mathbb{R}^n$. If for $\gamma<\infty$ and $\sigma\geq 1$, the inequality (28) holds, then the iterative control law $\hat{v}_i(z(k))$ of the iterative ADP algorithms (15)–(18) is convergent, i.e.,*

$$\hat{v}_\infty(z(k))=\lim_{i\to\infty}\hat{v}_i(z(k))=\arg\min_{v(k)\in\mathfrak{A}}\{U(z(k),v(k))+\hat{V}_\infty(z_{k+1})\}. \quad (33)$$

3.3. Convergence criteria of the iterative ADP algorithm

In the previous subsection, we have discussed the convergence property of the iterative ADP algorithms (15)–(18). The convergence criterion is obtained by (28). From (28), we can see that if we obtain the parameters σ and γ, then the convergence criterion can be justified. However, the parameters σ and γ are difficult to achieve. First, σ is a uniform approximation error which satisfies (20) for $\forall i=0,1,\ldots$. This means that σ is unknown till all the σ_i is considered and this is impossible before $i\to\infty$. Second, if we want to obtain γ, then according to (21) and (22), we should solve J^* first. While we can see that J^* cannot be solved before the iterative ADP algorithms (15)–(18) is implemented for $i\to\infty$. Thus, we cannot justify the convergence of the iterative ADP algorithm by solving (21) and (22) directly. To overcome this difficulty, a new justification for the convergence criteria with approximation errors is developed to guarantee the convergence of the iterative ADP algorithm.

According to the definitions of iterative performance index functions $\hat{V}_i(z(k))$ and $\Gamma_i(z(k))$ in (18) and (19), for $\forall\, i=0,1,\ldots$, if we let σ_i satisfy

$$\hat{V}_i(z(k))\leq\sigma_i\Gamma_i(z(k)), \quad (34)$$

then we have $\sigma=\max\{\sigma_0,\sigma_1,\ldots\}$. On the other hand, for $\forall\, i=0,1,\ldots$, there exists an $\epsilon_i(z(k))$ that satisfies

$$\hat{V}_i(z(k))-\Gamma_i(z(k))\leq\epsilon_i(z(k)). \quad (35)$$

In (35), it is required that $\epsilon_i(z(k))\geq 0$ for $i=0,1,\ldots$. For the situation that $\epsilon_i(z(k))<0$, we have $\hat{V}_i(z(k))<\Gamma_i(z(k))$. Then, $\hat{V}_i(z(k))$ can be guaranteed to converge. Hence, for $\forall\sigma_i$, we can choose $\epsilon_i(z(k))$ that satisfies

$$\hat{V}_i(z(k))-\epsilon_i(z(k))\leq\frac{\hat{V}_i(z(k))}{\sigma_i}. \quad (36)$$

Next, we will develop an effective method to estimate the parameter γ. As $J^*(z(k+1))$ is unknown, we cannot estimate $J^*(z(k+1))$ before $i\to\infty$. Hence an indirect estimation method is developed. First, we introduce the definition of admissible control law.

Definition 1. A control law $u(z(k))$ is defined to be admissible with respect to (8) on Ω if $u(z(k))$ is continuous on Ω, $u(0)=0$, $u(z(k))$ stabilizes (8) on Ω, and for $\forall\, z(0)\in\Omega$, $J(z(0))$ is finite.

Lemma 1. *Let $\mu(z(k))$ be an arbitrary admissible control law of system (8). Let $J^*(z(k))$ be the optimal performance index function and let $\Phi(z(k))$ be the performance index function constructed by $\mu(z(k))$, which satisfies*

$$\Phi(z(k))=U(z(k),\mu(z(k)))+\Phi(z(k+1)). \quad (37)$$

Then, we have $J^*(z(k))\leq\Phi(z(k))$.

Proof. The conclusion is easy to obtain by mathematical induction and the proof is omitted here.

Lemma 1 shows that if we obtain an admissible control law $\mu(z(k))$, then the upper bound of the optimal performance index function can be estimated by $\Phi(z(k))$. In the following, we will give an effective algorithm to obtain the performance index function $\Phi(z(k))$ by repeating experiments using neural networks. The detailed procedure is expressed by the following algorithm.

Algorithm 1. Solve the performance index function $\Phi(z(k))$.

Step (i). Choose a semi-positive definite function $\Psi(z(k))\geq 0$. Initialize two neural networks (critic networks for brief) cnet1 and cnet2 with random weights. Let $\Phi_0(z(k))=\Psi(z(k))$. Give the max iteration of computation $i_{\max}$.

Step (ii). Establish a neural network (action network for brief) with random weights to generate an initial control law $\mu(z(k))$ with $\mu(z(k))=0$ for $z(k)=0$. Let $i=0$.

Step (iii). Train the critic network cnet1 to approximate $\Phi_1(x_k)$, where $\Phi_1(x_k)$ satisfies

$$\Phi_1(x_k)=U(x_k,\mu(x_k))+\Phi_0(x_{k+1}).$$

Step (iv). Copy cnet1 to cnet2, i.e., cnet2=cnet1.

Step (v). Let $i=i+1$. Use cnet2 to get $\Phi_i(x_{k+1})$ and train the critic network cnet1 to approximate $\Phi_{i+1}(x_k)$, where $\Phi_{i+1}(x_k)$ satisfies

$$\Phi_{i+1}(x_k)=U(x_k,\mu(x_k))+\Phi_i(x_{k+1}). \tag{38}$$

Step (vi). Use cnet1 to get $\Phi_{i+1}(x_k)$ and use cnet2 to get $\Phi_i(x_k)$. If $|\Phi_{i+1}(x_k)-\Phi_i(x_k)|<\varepsilon$, then goto Step (viii). Else goto next step.

Step (vii). If $i>i_{\max}$, then goto Step (ii). Else goto Step (iv).

Step (viii). Return $\mu(x_k)$ and let $v_0(x_k)=\mu(x_k)$.

According to Steps (i)–(viii), we can see that if $\Phi_i(z(k))$ is convergent as $i\to\infty$, then we say that we have obtained an effective performance index function $\Phi(z(k))\geq J^*(z(k))$. We now derive the following theorem to guarantee the effectiveness of the algorithm.

Theorem 3. *Let $\Psi(z(k))\geq 0$ be an arbitrary semi-positive definite function. Let $\mu(z(k))$ be an arbitrary control law for system (8) which satisfies $\mu(0)=0$. Define the iterative performance index function $\Phi_i(z(k))$ as (38), where $\Phi_0(z(k))=\Psi(z(k))$. Then $\mu(z(k))$ is an admissible control law if and only if the limit $\lim_{i\to\infty}\Phi(z(k))$ exists for $\forall z(k)\in\mathbb{R}^n$.*

Proof. We first prove the sufficiency of the statement. Assume that $\mu(z(k))$ is an admissible control law. According to (38), we can get

$$\Phi_{i+1}(z(k))=\sum_{j=0}^{i}U(z(k+j),\mu(z(k+j)))+\Psi(z(k)). \tag{39}$$

Let $i\to\infty$. We can obtain

$$\lim_{i\to\infty}\Phi_i(z(k))=\sum_{j=0}^{\infty}U(z(k+j),\mu(z(k+j)))+\Psi(z(k)). \tag{40}$$

If $\mu(z(k))$ is an admissible control law, then $\sum_{j=0}^{\infty}U(z(k+j),\mu(z(k+j)))$ is finite. As for an arbitrary finite $z(k)$, $\Psi(z(k))$ is finite, then for $\forall i=0,1,2,\ldots$, we have that $P_{i+1}(z(k))$ is finite. Hence $\lim_{i\to\infty}\Phi_i(z(k))$ is finite, which means $\Phi_{i+1}(z(k))=\Phi_i(z(k))$, as $i\to\infty$.

On the other hand, if the limit $\lim_{i\to\infty}\Phi(z(k))$ exits, according to (40), we can get that $\sum_{j=0}^{\infty}U(z(k+j),\mu(z(k+j)))$ is finite. Since the utility function $U(z(k),\mu(z(k)))$ is positive definite for $\forall z(k),\mu(z(k))$, then we can obtain $U(z(k+j),\mu(z(k+j)))\to 0$ as $j\to\infty$. As $\mu(z(k))=0$ for $z(k)=0$, we can get that $z(k)\to 0$ as $k\to\infty$, which means that system (8) is stable and $\mu(z(k))$ is an admissible control law. The necessity of the statement is proven and the proof is completed.

From Theorem 3, we can see that if the performance index function $\Phi_i(z(k))$ is convergent as $i\to\infty$, i.e.,

$$\lim_{i\to\infty}\Phi_i(z(k))=\Phi(z(k)), \tag{41}$$

then we can find a constant $\overline{\gamma}$ that satisfies

$$\Phi(z(k+1))\leq\overline{\gamma}U(z(k),v(k)), \tag{42}$$

where $z(k+1)$ is defined in (8). Next, we can give the convergence justification theorem of the iterative ADP algorithm.

Theorem 4. *Let the iterative performance index function $\hat{V}_i(z(k))$ and the iterative control $\hat{v}_i(z(k))$ be obtained by (15)–(18), respectively. Let $\overline{\gamma}$ satisfy (42) for $\forall z(k),v(k)$. For $\forall i=0,1,\ldots$, if the iterative approximation error $\epsilon_i(z(k))$ satisfies*

$$\epsilon_i(z(k))<\frac{\hat{V}_i(z(k))}{\overline{\gamma}+1} \tag{43}$$

then we have that the iterative performance index function $\hat{V}_i(z(k))$ in the iterative ADP algorithms (15)–(18) is convergent to a finite neighborhood of the optimal performance index function $J^*(z(k))$, as $i\to\infty$.

Proof. Let $\overline{\gamma}$ be expressed by (42). Then, we have

$$\overline{\gamma}\geq\frac{\Phi(z(k+1))}{U(z(k),v(k))}\geq\frac{J^*(z(k+1))}{U(z(k),v(k))}>\gamma. \tag{44}$$

If for $\forall i=0,1,\ldots,\sigma_i\leq(\overline{\gamma}+1)/\overline{\gamma}$ holds, then we have

$$\frac{\hat{V}_i(z(k))}{\hat{V}_i(z(k))-\epsilon_i(z(k))}\leq\frac{\overline{\gamma}+1}{\overline{\gamma}}<\frac{\gamma+1}{\gamma}. \tag{45}$$

According to (45), we can obtain (43) easily. On the other hand, according to (36), we have that

$$\max\left\{\frac{\hat{V}_i(z(k))}{\hat{V}_i(z(k))-\epsilon_i(z(k))}\right\}\leq\max\{\sigma_i\}=\sigma<\frac{\gamma+1}{\gamma}. \tag{46}$$

According to Theorem 2, we can draw the conclusion.

Theorem 5. *Let the iterative performance index function $\hat{V}_i(z(k))$ and the iterative control $\hat{v}_i(z(k))$ be obtained by (15)–(18), respectively. If Theorem 4 holds for $\forall i=0,1,\ldots$, and there exists a constant $\lambda_i(z(k))$ that makes*

$$\hat{V}_i(z(k))=\lambda_i(z(k))J^*(z(k)) \tag{47}$$

hold, where

$$0\leq\lambda_i(z(k))\leq\left(1+\sum_{j=1}^{i}\frac{\gamma^j\sigma^{j-1}(\sigma-1)}{(\gamma+1)^j}+\frac{\gamma^i\sigma^i(\delta-1)}{(\gamma+1)^i}\right),$$

then we have that $\epsilon_i(z(k))$ is a positive definite function of $z(k)$.

Proof. As Theorem 4 holds, we have for $\forall i=0,1,\ldots$, $\lambda_i(z(k))$ is finite and the limit of $\lambda_i(z(k))$ exists as $i\to\infty$. Next, let $z(k)=0$, we have $\hat{V}_i(0)=\lambda_i(0)J^*(0)=0$. On the other hand, if $z(k)\to\infty$, then we have $\hat{V}_i(z(k))\to\infty$. Hence we have that $\epsilon_i(z(k))$ is a positive definite function of $z(k)$. □

Remark 2. From Theorem 5 we can see that for different state variable $z(k)$, it requires different approximation error $\epsilon_i(z(k))$ to guarantee the convergence of the developed iterative ADP algorithm. From (47) we can see that if $\|z(k)\|$ is large, then the developed iterative ADP algorithm permits a large approximation error to be convergent and if $\|z(k)\|$ is small, then small approximation error is required to make the convergence of the iterative ADP algorithm.

3.4. The iterative ADP algorithm

Based on the above preparations, we now summarize the optimal tracking control scheme with approximation errors by iterative ADP algorithm in Algorithm 2.

Algorithm 2. Iterative ADP algorithm for optimal tracking control scheme with approximation errors.

Block 1: *Initialization*

Step 1(a). Choose randomly a vector of initial states, i.e., $X=(x^{(1)},x^{(2)},\ldots,\{x^{(p)})$, where p is a large integer.

Step 1(b). Give the desired trajectory $\eta(k)$.

Step 1(c). Choose an approximation precision ε.

Block 2: *Solving* $\overline{\gamma}$

Step 2(a). Solve the performance index function $\Phi(z(k))$ by Algorithm 1.

Step 2(b). Obtain $\overline{\gamma}$ that satisfies (42).

Block 3: *Iteration*

Step 3(a). Let $i=0$ and let the initial performance index function $\hat{V}_0(z(k))=0$.

Step 3(b). Estimate the approximation error $\epsilon_0(z(k))$ by (35).

Step 3(c). If $\epsilon_0(z(k))$ satisfies (43), then goto next step. Otherwise, reduce $\epsilon_0(z(k))$ by reducing $\rho_0(z(k))$ and $\pi_0(e(k))$. Goto Step 3(b).

Step 3(d). Let $i=i+1$.

Step 3(e). Compute $\hat{v}_i(z(k))$ by (17) and obtain $\hat{V}_{i+1}(z(k))$ by (18). Estimate the approximation error $\epsilon_i(z(k))$ by (35).

Step 3(f). If $\epsilon_i(z(k))$ satisfies (43), then, goto next step. Otherwise, reduce $\epsilon_i(z(k))$ by reducing $\rho_i(z(k))$ and $\pi_i(e(k))$. Goto Step 3(e).

Step 3(g). If $|\hat{V}_i(z(k))-\hat{V}_{i-1}(z(k))|\le\varepsilon$, then the optimal performance index function is obtained and goto Step 3(h); Else, goto Step 3(d).

Step 3(h). Return $\hat{v}_i(e(k))$ and $\hat{V}_i(e(k))$.

4. Implementation of the iterative ADP algorithm

In this paper, for $i=0,1,\ldots$, BP neural networks are used to approximate $v_i(x_k)$ and $V_{i+1}(x_k)$, respectively. Assume that the number of hidden layer neurons is denoted by ℓ, the weight matrix between the input layer and hidden layer is denoted by Y, the weight matrix between the hidden layer and output layer is denoted by W, then the output of three-layer NN is represented by

$$\hat{F}(X,Y,W)=W^T\sigma(Y^TX) \tag{48}$$

where $\sigma(Y^TX)\in R^{\ell}$, $[\sigma(z)]_i=(e^{z_i}-e^{-z_i})/(e^{z_i}+e^{-z_i}), i=1,\ldots\ell$, are the activation function. The target function of the neural network can be expressed by

$$F(X)=F(X,Y^*,W^*) \tag{49}$$

where Y^* and W^* are the ideal weight parameters.

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

4.1. The critic network

For $\forall i=0,1,\ldots$, the critic network is used to approximate the performance index function $V_{i+1}(z(k))$. The output of the critic network is denoted by

$$\hat{V}^j_{i+1}(z(k))=W^{jT}_{ci}\sigma(Y^T_cz(k))=W^{jT}_{ci}\sigma(Z_c(k)), \tag{50}$$

where $Z_c(k)=Y^T_cz(k)$ and $j=0,1,\ldots$. Let W^0_{ci} and Y_c be random weight matrices. The target function can be written as

$$V_{i+1}(z(k))=U(x_k,v_i(z(k)))+\hat{V}_i(z(k+1)). \tag{51}$$

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

$$e^j_{ci}(k)=\hat{V}^j_{i+1}(z(k))-V_{i+1}(z(k)). \tag{52}$$

The objective function to be minimized in the critic network training is

$$E^j_{ci}(k)=\tfrac{1}{2}(e^j_{ci}(k))^2.$$

The gradient-based weight update rule [8] can be applied here to train the critic network

$$\begin{aligned}W^{j+1}_{ci}(k)&=W^j_{ci}(k)+\Delta W^j_{ci}(k),\\&=W^j_{ci}(k)-\alpha_c\left[\frac{\partial E^j_{ci}(k)}{\partial\hat{V}^j_{i+1}(z(k))}\frac{\partial\hat{V}^j_{i+1}(z(k))}{\partial W^j_{ci}(k)}\right]\\&=W^j_{ci}(k)-\alpha_ce^j_{ci}(k)\sigma(Z_c(k)),\end{aligned} \tag{53}$$

where $\alpha_c>0$ is the learning rate of critic network. If the training precision is achieved, then we say that $V_{i+1}(z(k))$ can be approximated by the critic network.

4.2. The action network

In the action network, the state error $z(k)$ is used as input to create the optimal control law as the output of the network. The output can be formulated as

$$\hat{v}^j_i(z(k))=W^{jT}_{ai}\sigma(Y^T_az(k))=W^{jT}_{ai}\sigma(Z_a(k)),$$

where $Z_a(k)=Y^T_az(k)$ and $j=0,1,\ldots$. Let W^0_{ai} and Y_a be random weight matrices. The target of the output of the action network is given by $v_i(z(k))=\arg\min_{v(k)}\{U(z(k),v(k))+\hat{V}_i(z(k+1))\}$. So we can define the output error of the action network as

$$e^j_{ai}(k)=\hat{v}^j_i(z(k))-v_i(z(k)). \tag{54}$$

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

$$E^j_{ai}(k)=\tfrac{1}{2}(e^j_{ai}(k))^T(e^j_{ai}(k)).$$

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

$$\begin{aligned}W^{j+1}_{ai}(k)&=W^j_{ai}(k)+\Delta W^j_{ai}(k),\\&=W^j_{ai}(k)-\beta_a\left[\frac{\partial E^j_{ai}(k)}{\partial e^j_{ai}(k)}\frac{\partial e^j_{ai}(k)}{\partial\hat{v}^j_i(k)}\frac{\partial\hat{v}^j_i(k)}{\partial W^j_{ai}(k)}\right]\\&=W^j_{ai}(k)-\beta_a\sigma(Z_a(k))(e^j_{ai}(k))^T,\end{aligned} \tag{55}$$

where $\beta_a>0$ is the learning rate of action network. If the training precision is achieved, then we say that the iterative control law $v_i(z(k))$ can be approximated by the action network.

In this paper, to enhance the convergence speed of the neural networks, only one layer of neural network is updated during the training procedure. To guarantee the effectiveness of the neural network implementation, the convergence of the neural network weights is proven which makes the iterative performance index function and iterative control be approximated by the critic and action networks, respectively. The convergence property of the neural network weights is shown in the following theorem.

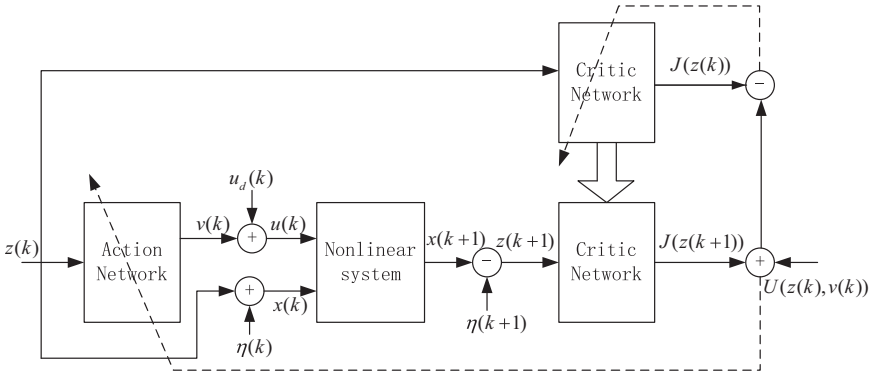

Fig. 1. The structure diagram of the algorithm.

Theorem 6. *Let the target performance index function and the target iterative control law be expressed by*

$$V_{i+1}(z(k)) = W_{ci}^{*T}\sigma(Z_c(k)), \tag{56}$$

and

$$v_i(x_k) = W_{ai}^{*T}\sigma(Z_a(k)), \tag{57}$$

respectively. Let the critic and action networks be trained by (53) and (55), respectively. Then, the critic weights $W_{ci}(k)$ and action network weights $W_{ai}(k)$ are asymptotically convergent to the optimal weights $W_{ci}^*(k)$ and $W_{ai}^*(k)$, respectively.

Proof. Let $\overline{W}_{ci}^j = W_{ci}^j - W_{ci}^*$ and $\overline{W}_{ai}^j = W_{ai}^j - W_{ai}^*$. From (53) and (55), we have

$$\overline{W}_{ci}^{j+1}(k) = \overline{W}_{ci}^j(k) - \alpha_c e_{ci}^j(k)\sigma(Z_c(k)),$$

and

$$\overline{W}_{ai}^{j+1}(k) = \overline{W}_{ai}^j(k) - \beta_a e_{ai}^j(k)\sigma(Z_a(k)).$$

Consider the following Lyapunov function candidate:

$$L(\overline{W}_{ci}^j, \overline{W}_{ai}^j) = \mathrm{tr}\{\overline{W}_{ci}^{jT}\overline{W}_{ci}^j + \overline{W}_{ai}^{jT}\overline{W}_{ai}^j\}. \tag{58}$$

Then, the difference of the Lyapunov function candidate (58) is given by

$$\begin{aligned}\Delta L(\overline{W}_{ci}^j, \overline{W}_{ai}^j) &= \mathrm{tr}\{\overline{W}_{ci}^{(j+1)T}\overline{W}_{ci}^{j+1} + \overline{W}_{ai}^{(j+1)T}\overline{W}_{ai}^{j+1}\} \\ &\quad - \mathrm{tr}\{\overline{W}_{ci}^{jT}\overline{W}_{ci}^j + \overline{W}_{ai}^{jT}\overline{W}_{ai}^j\} \\ &= \alpha_c \| e_{ci}^j(k)\|^2(-2+\alpha_c\|\sigma(Z_c(k))\|^2) \\ &\quad + \beta_a \| e_{ai}^j(k)\|^2(-2+\beta_a\|\sigma(Z_a(k))\|^2).\end{aligned}$$

According to the definition of $\sigma(\cdot)$ in (48), we know that $\|\sigma(Z_c(k))\|^2$ and $\|\sigma(Z_a(k))\|^2$ are both finite for $\forall\, Z_c(k), Z_a(k)$. Thus, if we choose the learning rates α_c and β_a that satisfy $\alpha_c \le 2/\|\sigma(Z_c(k))\|^2$ and $\beta_a \le 2/\|\sigma(Z_a(k))\|^2$, then we have $\Delta L(\overline{W}_{ci}^j, \overline{W}_{ai}^j) < 0$. The proof is completed.

5. Simulation study

Our example is chosen as the example in [36] with motifications. Consider the following affine nonlinear system:

$$x(k+1) = f(x(k)) + g(x(k))u(k) \tag{59}$$

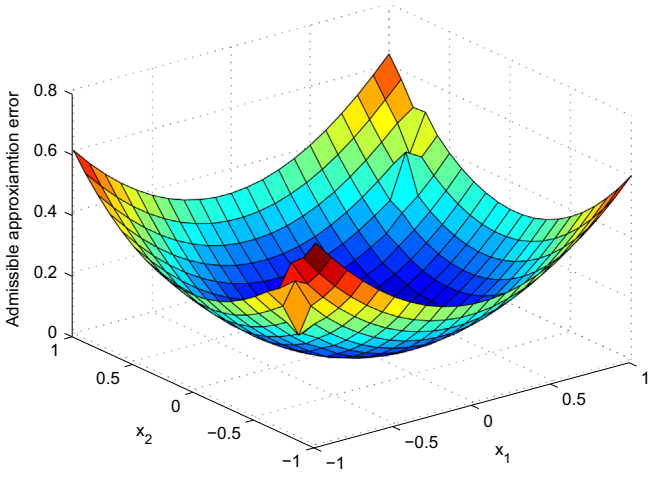

Fig. 2. The curve of the approximation errors.

where $x(k) = [x_1(k)\ x_2(k)]^T$ and $u(k) = [u_1(k)\ u_2(k)]^T$. Let the system function be expressed as

$$f(x(k)) = \begin{bmatrix} 0.2x_1(k)\exp(x_2^2(k)) \\ 0.3x_2^3(k) \end{bmatrix},$$

$$g(x(k)) = \begin{bmatrix} -x_1(k)x_2(k) & 0.1 \\ x_2(k) & -0.8x_1(k)x_2(k) \end{bmatrix}.$$

The desired trajectory is set to $\eta(k) = [\sin(k+\pi/2)\ 0.5\ \cos(k)]^T$. The performance index function is defined as in (6), where $Q = R = I \in \mathbb{R}^{2\times 2}$ and I denotes the identity matrix.

We use neural networks to implement the iterative ADP algorithm. The critic network and the action network are chosen as three-layer BP neural networks with the structures of 2–8–1 and 2–8–2, respectively. We choose a random array of state variable in $[-1,1]$ to train the neural networks. For each iterative step, the critic network and the action network are trained for 2000 steps under the learning rate $\alpha = 0.001$ so that the approximation error limit is reached. The iterative ADP algorithm runs for 50 iteration steps to guarantee the convergence of the iterative performance index function. According to Theorem 4, we know that if the approximation errors of the neural networks satisfy the inequality (43), then the iterative performance index functions of the iterative ADP algorithm is convergent to the finite neighborhood of the optimal performance index function. The curve of the approximation errors is displayed in Fig. 2.

Remark 3. It is an important property for the neural network implementation of the iterative ADP algorithm. From Theorem 5 we can see that to guarantee the convergence of the iterative performance index function, for different state variable $z(k)$, we should use different training precisions. For large $\|z(k)\|$, the approximation error of the neural networks can be large. As $\|z(k)\| \to 0$, the approximation error of the neural networks should also approach zero. This property can be seen from Fig. 2. As is known, the approximation of neural networks cannot be globally accurate with no approximation errors, so the implementation of the iterative ADP algorithm by neural networks may be invalid as $z(k) \to 0$. On the other hand, in the real-world, the neural networks are generally trained under a global uniform training precision or approximation error. Thus, it is required for the proposed iterative ADP algorithm that the approximation error is small to make the iterative performance index function converge for most of the state space. In the following, we will explain this property in detail.

We choose four different global training precisions of neural networks to justify the effectiveness of the developed iterative ADP algorithm. First, let the approximation error of the neural networks be $\epsilon = 10^{-6}$. The trajectory of the iterative performance index function is shown in Fig. 3(a). Second, let the approximation error of the neural networks be $\epsilon = 10^{-4}$. The trajectory of the iterative performance index function is shown in Fig. 3(b). Third, let the approximation error of the neural networks be $\epsilon = 10^{-3}$. The trajectory of the iterative performance index function is shown in Fig. 3(c). We can see that the iterative performance index function is not monotonically increasing convergent. We continue enlarging the approximation error of the neural networks to $\epsilon = 10^{-1}$. The trajectory of the iterative performance index function is shown in Fig. 3(d).

For approximation error $\epsilon = 10^{-6}$, implement the approximate optimal control for the system (59). Let the implementation time T_f=250 and the trajectories of the controls are displayed in Fig. 4(a) and the corresponding state trajectories are displayed in Fig. 5(a). For approximation error $\epsilon = 10^{-4}$, the trajectories of the controls are displayed in Fig. 4(b) and the corresponding state trajectories ar displayed in 5(b). When the approximation error

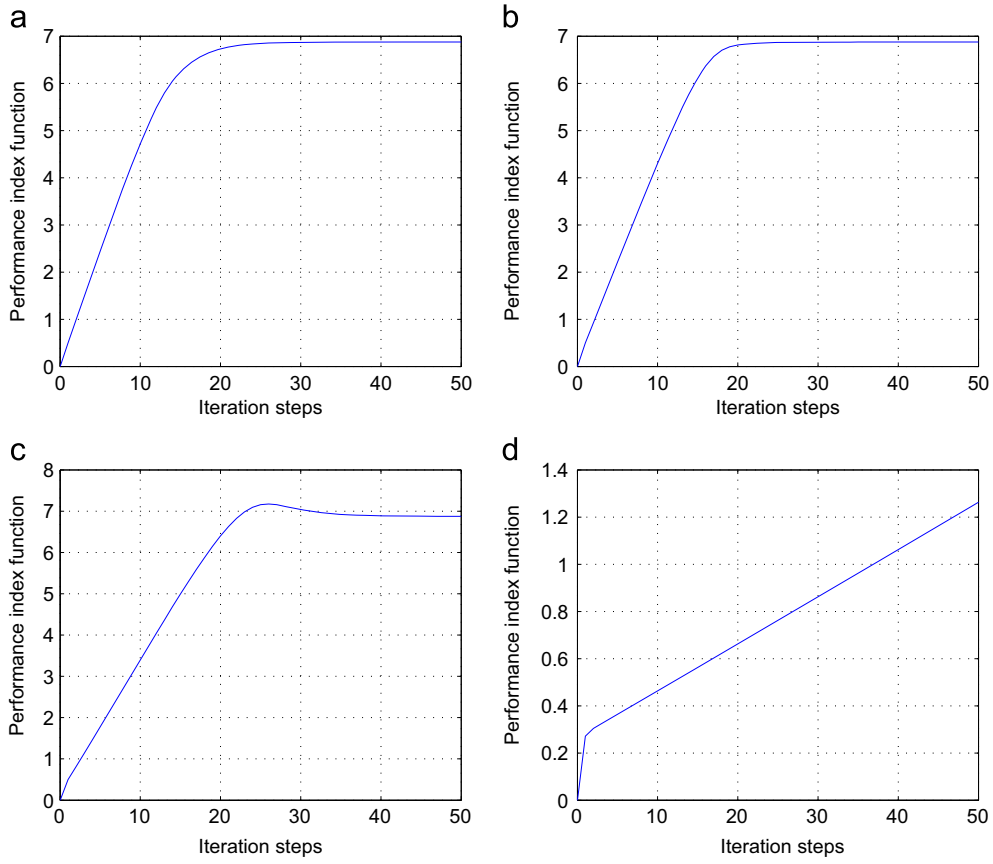

Fig. 3. Performance index functions. (a) $\sigma = 10^{-6}$. (b) $\sigma = 10^{-4}$. (c) $\sigma = 10^{-3}$. (d) $\sigma = 10^{-1}$.

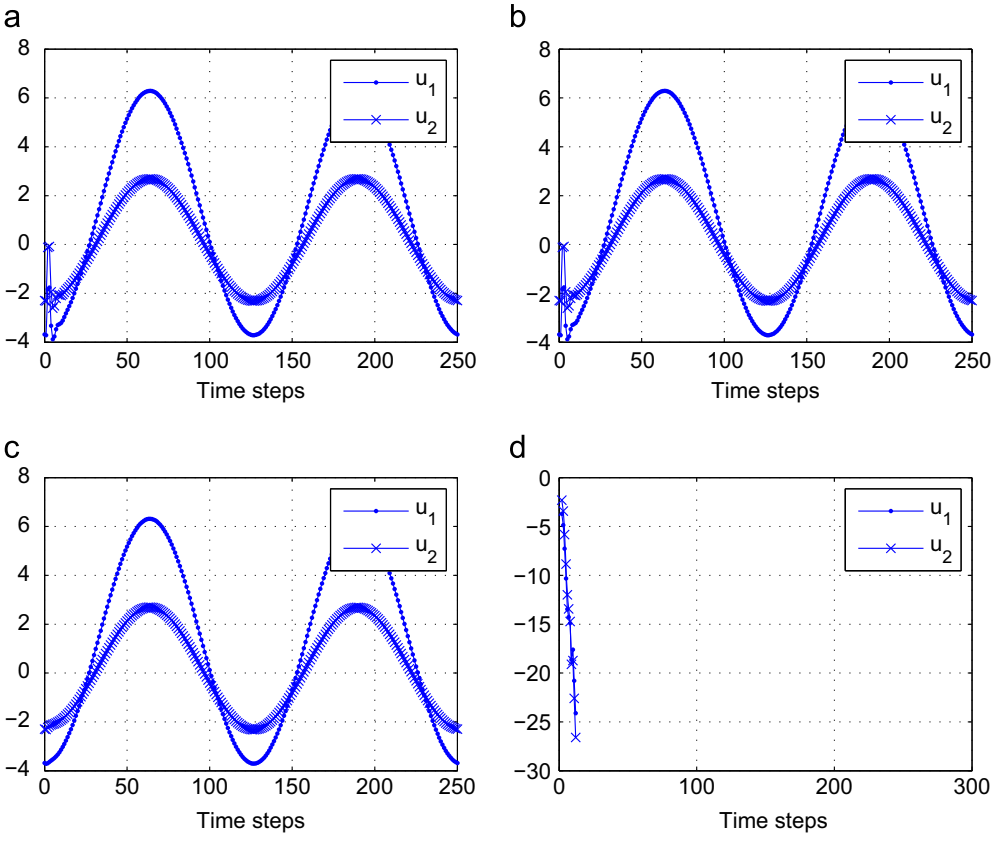

Fig. 4. Control trajectories. (a) $\sigma = 10^{-6}$. (b) $\sigma = 10^{-4}$. (c) $\sigma = 10^{-3}$. (d) $\sigma = 10^{-1}$.

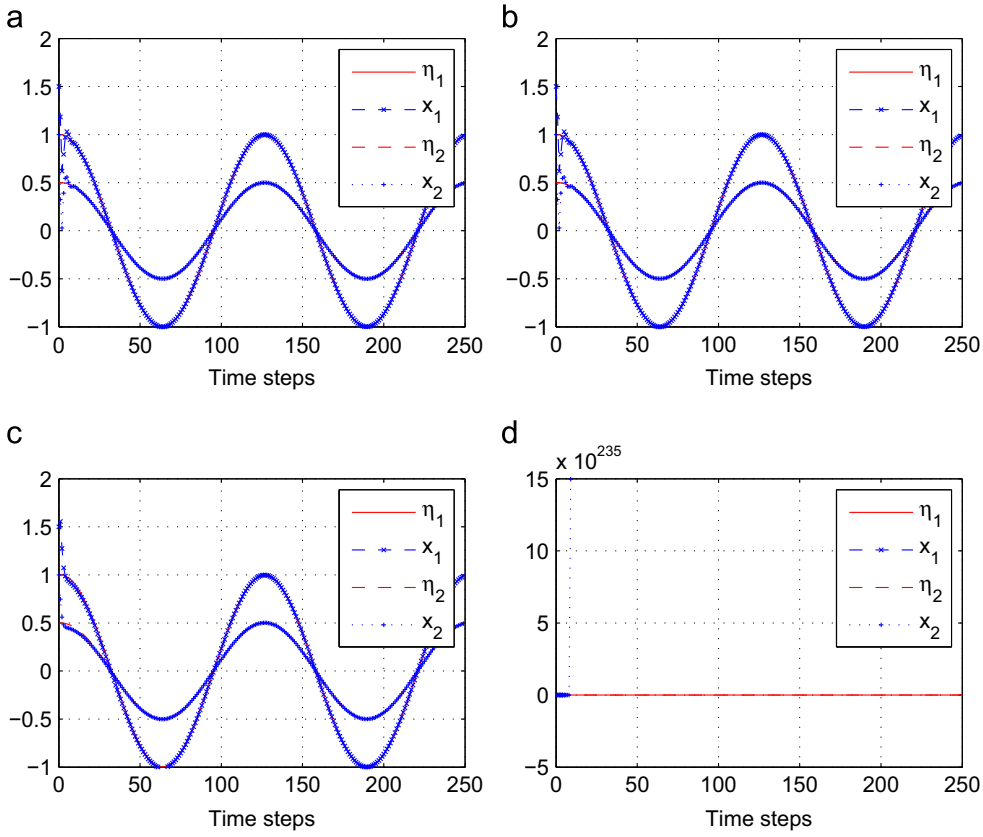

Fig. 5. State trajectories. (a) $\sigma = 10^{-6}$. (b) $\sigma = 10^{-4}$. (c) $\sigma = 10^{-3}$. (d) $\sigma = 10^{-1}$.

$\epsilon = 10^{-3}$, we can see that the iterative performance index functions is not monotone. The trajectories of the control and state are displayed in Fig. 4(c) and the corresponding state trajectories ar displayed in 5(c). We can see that for approximation errors $\epsilon = 10^{-6}$, $\epsilon = 10^{-4}$, $\epsilon = 10^{-3}$, the iterative control law can make the iterative performance index function convergent. In these situations, the obtained control law can make the system state track the desired one. When the approximation error $\epsilon = 10^{-1}$. Then, we can see that the iterative performance index functions is not convergent. In this situation, the control system is not stable and the trajectories of the control and state are displayed in Figs. 4(d) and 5(d), respectively. In this situation, we can see that the control system is not stable under the iterative control law.

6. Concluding remarks and future works

In this paper, an effective iterative ADP algorithm is developed to solve optimal tracking control problems for infinite horizon discrete-time nonlinear systems. The approximation errors of the neural networks are considered. The convergence criteria under the approximation errors are established which guarantees that the iterative performance index functions converge to a finite neighborhood of the optimal performance index function. Neural networks are employed to implement the developed iterative ADP algorithm and the convergence property of the neural networks are analyzed. Finally, simulation results are given to illustrate the performance of the developed algorithm.

We point out that the convergence property of the iterative index functions is presented in this paper considering the approximation errors while the stability property of the system under the iterative control laws is not discussed in this paper. Hence, the stability analysis of the iterative ADP algorithm will be investigated in our future work. On the other hand, in this paper, the approximation errors of the performance index functions and the iterative control laws are considered, while the accurate system function is necessary. In the real world, however, the accurate system model is also difficult to obtain. Thus the properties of iterative ADP algorithm considering the system and iteration errors is another future research direction.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 61034002, 61233001, 61273140, 61304086, and 61374105, in part by Beijing Natural Science Foundation under Grant 4132078, and in part by the Early Career Development Award of SKLMCCS.

References

[1] I.J. Ha, E.G. Gilbert, Robust tracking in nonlinear systems, IEEE Trans. Autom. Control 32 (1987) 763–771.
[2] 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.
[3] B. Miao, T. Li, W. Luo, A DSC and MLP based robust adaptive NN tracking control for underwater vehicle, Neurocomputing 111 (2013) 184–189.
[4] Y. Pan, Y. Zhou, T. Sun, M.J. Er, Composite adaptive fuzzy H_∞ tracking control of uncertain nonlinear systems, Neurocomputing 99 (2013) 15–24.
[5] X. Zhang, H. Zhang, Q. Sun, Y. Luo, Adaptive dynamic programming-based optimal control of unknown nonaffine nonlinear discrete-time systems with proof of convergence, Neurocomputing 91 (2012) 48–55.
[6] P.J. Werbos, Advanced forecasting methods for global crisis warning and models of intelligence, Gen. Syst. Yearb. 22 (1977) 25–38.
[7] P.J. Werbos, A menu of designs for reinforcement learning over time, in: W. T. Miller, R.S. Sutton, P.J. Werbos (Eds.), Neural Networks for Control, MIT Press, Cambridge, 1991, pp. 67–95.

[8] J. Si, Y.T. Wang, On-line learning control by association and reinforcement, IEEE Trans. Neural Netw. 12 (2001) 264–276.
[9] Q. Wei, H. Zhang, J. Dai, Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions, Neurocomputing 72 (2009) 1839–1848.
[10] X. Xu, Z. Hou, C. Lian, H. He, Online learning control using adaptive critic designs with sparse kernel machines, IEEE Trans. Neural Netw. Learn. Syst. 24 (2013) 762–775.
[11] Q. Wei, D. Liu, Numerical adaptive learning control scheme for discrete-time nonlinear systems, IET Control Theory Appl. 7 (2013) 1472–1486.
[12] D. Wang, D. Liu, Neuro-optimal control for a class of unknown nonlinear dynamic systems using SN-DHP technique, Neurocomputing 121 (2013) 218–225.
[13] D. Liu, Y. Huang, D. Wang, Q. Wei, Neural network observer-based optimal control for unknown nonlinear systems using adaptive dynamic programming, Int. J. Control 86 (2013) 1554–1566.
[14] D. Liu, Y. Zhang, H. Zhang, A self-learning call admission control scheme for CDMA cellular networks, IEEE Trans. Neural Netw. 16 (2005) 1219–1228.
[15] D.V. Prokhorov, D.C. Wunsch, Adaptive critic designs, IEEE Trans. Neural Netw. 8 (1997) 997–1007.
[16] 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.
[17] F. Wang, H. Zhang, D. Liu, Adaptive dynamic programming: an introduction, IEEE Comput. Intell. Mag. 4 (2009) 39–47.
[18] 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.
[19] P.J. Werbos, Approximate dynamic programming for real-time control and neural modeling, in: D.A. White, D.A. Sofge (Eds.), Handbook of Intelligent Control: Neural Fuzzy and Adaptive Approaches, New York, 1992 (Chapter 13).
[20] R. Enns, J. Si, Helicopter trimming and tracking control using direct neural dynamic programming, IEEE Trans. Neural Netw. 14 (2003) 929–939.
[21] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, MA, 1996.
[22] F.L. Lewis, V.G. Kyriakos, Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data, IEEE Trans. Syst. Man Cybern.–Part B: Cybern. 41 (2011) 14–25.
[23] C. Watkins, Learning from Delayed Rewards (Ph.D. Thesis), Cambridge University, Cambridge, England, 1989.
[24] R. Song, W. Xiao, H. Zhang, Multi-objective optimal control for a class of unknown nonlinear systems based on finite-approximation-error ADP algorithm, Neurocomputing 119 (2013) 212–221.
[25] F. Wang, N. Jin, D. Liu, Q. Wei, Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ϵ-error bound, IEEE Trans. Neural Netw. 22 (2011) 24–36.
[26] Q. Wei, D. Liu, An iterative ϵ-optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state, Neural Netw. 32 (2012) 236–244.
[27] H. Zhang, R. Song, Q. Wei, T. Zhang, Optimal tracking control for a class of nonlinear discrete-time systems with time delays based on heuristic dynamic programming, IEEE Trans. Neural Netw. 22 (2011) 1851–1862.
[28] H. Zhang, Q. Wei, D. Liu, An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games, Automatica 47 (2011) 207–214.
[29] H. Li, D. Liu, Optimal control for discrete-time affine nonlinear systems using general value iteration, IET Control Theory Appl. 6 (2012) 2725–2736.
[30] D. Wang, D. Liu, Q. Wei, D. Zhao, N. Jin, Optimal control of unknown nonlinear discrete-time systems based on adaptive dynamic programming approach, Automatica 48 (2012) 1825–1832.
[31] Q. Wei, H. Zhang, D. Liu, Y. Zhao, An optimal control scheme for a class of discrete-time nonlinear systems with time delays using adaptive dynamic programming, Acta Autom. Sin. 36 (2010) 121–129.
[32] 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) 16–18.
[33] F.L. Lewis, D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, IEEE Circuits Syst. Mag. 9 (2009) 32–50.
[34] 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.
[35] R. Beard, Improving the Closed-Loop Performance of Nonlinear Systems (Ph.D. Thesis) Rensselaer Polytechnic Institute, Troy, NY, 1995.
[36] 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.
[37] D. Liu, D. Wang, D. Zhao, Q. Wei, N. Jin, Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming, IEEE Trans. Autom. Sci. Eng. 9 (2012) 628–634.
[38] A. Heydari, S.N. Balakrishnan, Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics, IEEE Trans. Neural Netw. Learn. Syst. 24 (2013) 145–157.
[39] T. Huang, D. Liu, A self-learning scheme for residential energy system control and management, Neural Comput. Appl. 22 (2013) 259–269.
[40] D. Liu, H. Li, D. Wang, Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm, Neurocomputing 110 (2013) 92–100.
[41] D. Liu, Q. Wei, Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems, IEEE Trans. Cybern. 43 (2013) 779–789.
[42] Z. Ni, H. He, J. Wen, Adaptive learning in tracking control based on the dual critic network design, IEEE Trans. Neural Netw. Learn. Syst. 24 (2013) 913–928.
[43] D. Liu, D. Wang, X. Yang, An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs, Inf. Sci. 220 (2013) 331–342.
[44] D. Wang, D. Liu, Q. Wei, Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach, Neurocomputing 78 (2012) 14–22.
[45] D. Liu, H. Li, D. Wang, Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm, Neurocomputing 110 (2013) 92–100.
[46] D. Liu, Q. Wei, Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems, IEEE Trans. Cybern. 43 (2013) 779–789.
[47] D. Wang, D. Liu, D. Zhao, Y. Huang, D. Zhang, A neural-network-based iterative GDHP approach for solving a class of nonlinear optimal control problems with control constraints, Neural Comput. Appl. 22 (2013) 219–227.

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

Derong Liu (S'91–M'94–SM'96–F'05) received the Ph.D. degree in electrical engineering from the University of Notre Dame in 1994. He 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, and 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. He has published 14 books (six research monographs and eight edited volumes). Currently, he is the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems. 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 (2006–2009), and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China (2008). He is a Fellow of the IEEE and a Fellow of the INNS.