Neural Comput & Applic (2013) 22:219–227
DOI 10.1007/s00521-011-0707-2

ISNN 2011

A neural-network-based iterative GDHP approach for solving a class of nonlinear optimal control problems with control constraints

Ding Wang · Derong Liu · Dongbin Zhao · Yuzhu Huang · Dehua Zhang

Received: 12 February 2011 / Accepted: 5 July 2011 / Published online: 19 July 2011

Abstract In this paper, a novel neural-network-based iterative adaptive dynamic programming (ADP) algorithm is proposed. It aims at solving the optimal control problem of a class of nonlinear discrete-time systems with control constraints. By introducing a generalized nonquadratic functional, the iterative ADP algorithm through globalized dual heuristic programming technique is developed to design optimal controller with convergence analysis. Three neural networks are constructed as parametric structures to facilitate the implementation of the iterative algorithm. They are used for approximating at each iteration the cost function, the optimal control law, and the controlled nonlinear discrete-time system, respectively. A simulation example is also provided to verify the effectiveness of the control scheme in solving the constrained optimal control problem.

D. Wang (✉) · D. Liu · D. Zhao · Y. Huang · D. Zhang
Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, People's Republic of China
e-mail: ding.wang@ia.ac.cn

D. Liu
e-mail: derong.liu@ia.ac.cn; dliu@ece.uic.edu

D. Zhao
e-mail: dongbin.zhao@ia.ac.cn

Y. Huang
e-mail: yuzhu.huang@ia.ac.cn

D. Zhang
e-mail: dehua.zhang@ia.ac.cn

D. Liu
Department of Electrical and Computer Engineering, University of Illinois, Chicago, IL 60607, USA

Keywords Adaptive critic designs · Adaptive dynamic programming · Approximate dynamic programming · Neural dynamic programming · Neural networks · Optimal control · Reinforcement learning

1 Introduction

The classical control schemes work well for controlling linear, single input, single output systems, but they are unsuitable for controlling complex nonlinear, multiple input, multiple output systems, which are characteristic of numerous real-life control problems. As is known, optimal control theory has been used to solve many such nonlinear, multivariate problems in a variety of industrial settings, particularly in aerospace applications. However, it often requires solving the nonlinear Hamilton–Jacobi–Bellman (HJB) equation instead of the Riccati equation. For example, discrete-time HJB (DTHJB) equation is more difficult to work with than the Riccati equation because it involves solving nonlinear partial difference equations. Moreover, the control constraints are often confronted in practical problems, which results in a considerable difficulty in designing the optimal controller. Thus, the control of nonlinear systems with constraints has been the focus of many researchers for several decades. There are some methods for designing control laws considering the saturation phenomena [1–3]. Though the traditional dynamic programming (DP) approach has been a powerful technique for finding an optimal strategy of action over time in a constrained, nonlinear environment for many years, it is often computationally untenable to run it to obtain the optimal solution due to the fact that the cost grows drastically with the number of variables in the environment, i.e., the well-known "curse of dimensionality" [4].

In recent years, the ability of artificial neural networks (ANN or NN) to approximate arbitrary nonlinear functions plays a primary role in the use of such networks as components or subsystems in identifiers and controllers [5–9]. Besides, it has been used for universal function approximation in adaptive/approximate dynamic programming (ADP) algorithms, which were proposed in [7–9] as a method for solving optimal control problems forward-in-time. There are several synonyms used for ADP including "adaptive dynamic programming" [10–12], "approximate dynamic programming" [13], "neuro-dynamic programming" [14], "neural dynamic programming" [15, 16], "adaptive critic designs" (ACD) [17], and "reinforcement learning" [18, 19].

Research in ADP and the related reinforcement learning has gained much attention from scholars [7–17, 20–38]. According to Werbos [7] and Prokhorov and Wunsch [17], ADP approaches were classified into several main schemes: heuristic dynamic programming (HDP), action-dependent HDP (ADHDP), also known as Q-learning [18, 19], dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP. Using the adaptive-critic-based approach, Venayagamoorthy et al. [21] and Han and Balakrishnan [28] presented the neurocontrol for a turbogenerator and an agile missile system, respectively. Al-Tamimi et al. [29] proposed a greedy HDP iteration algorithm to solve the DTHJB equation of the optimal control problem for discrete-time nonlinear systems. Vrabie et al. [31] studied the continuous-time optimal control problem using ADP. Wang et al. [11] derived an ε-ADP algorithm for finite horizon discrete-time nonlinear systems. Abu-Khalaf and Lewis [33], Zhang et al. [34], and Song et al. [37] studied the near-optimal control of affine nonlinear systems with control constraints, respectively.

However, there is still no result for solving the optimal control problems for affine nonlinear discrete-time systems with control constraints through the GDHP technique. Incidentally, according to Prokhorov and Wunsch [17], the outputs of the critic network of the GDHP technique contain not only the cost function but also its derivatives. In addition, they stated that this is very important because the information associated with the cost function is as useful as the knowledge of its derivatives. It will show improved performance when using the iterative GDHP algorithm to tackle the constrained optimal control problems of nonlinear discrete-time systems. This paper deals with the problem based on iterative ADP algorithm via the GDHP technique (iterative GDHP algorithm for brief).

The rest of this paper is organized as follows. In Sect. 2, the DTHJB equation which includes nonquadratic functional is introduced for the constrained nonlinear discrete-time systems. Section 3 starts by developing the optimal control scheme based on iterative ADP algorithm with convergence analysis, and then the corresponding NN implementation of the iterative algorithm is presented using the GDHP technique. In Sect. 4, an example is presented to substantiate the derived theoretical results. Section 5 contains concluding remarks.

2 Problem statement

Consider the nonlinear discrete-time system given by

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

where $x_k \in \mathbb{R}^n$ is the state and $u(x_k) \in \mathbb{R}^m$ is the control vector, $f(\cdot)$ and $g(\cdot)$ are differentiable in their argument with $f(0) = 0$ and $g(0) = 0$. Assume that $f + gu$ 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. We denote $\Omega_u = \{u_k | u_k = [u_{1k}, u_{2k}, \ldots, u_{mk}]^T \in \mathbb{R}^m, |u_{ik}| \le \bar{u}_i, i = 1, 2, \ldots, m\}$, where $\bar{u}_i$ is the saturating bound for the ith actuator. Let $\bar{U} = \mathrm{diag}\{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m\}$ be the constant diagonal matrix.

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 $x_0 \in \Omega$, there exists a control sequence $u_0, u_1, \ldots, u_i \in \mathbb{R}^m, i = 0, 1, \ldots$, such that the state $x_k \to 0$ as $k \to \infty$.

The objective for general optimal control problems is to find the control law $u(x)$ which minimizes the infinite horizon cost function given by

$$J(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} U(x_i, u_i) \tag{2}$$

where U is the utility function, $U(0, 0) = 0$, $U(x_i, u_i) \ge 0$ for $\forall\ x_i, u_i$, and γ is the discount factor with $0 < \gamma \le 1$. The utility function can be written as

$$U(x_i, u_i) = x_i^T Q x_i + Y(u_i)$$

where $Y(u_i)$ is positive definite and can be chosen as quadratic form for the case of unconstrained problems.

Inspired by the work of Lyshevski [2, 3] and Abu-Khalaf [33], when dealing with bounded optimal control problems, we can employ a generalized nonquadratic functional

$$Y(u_i) = 2 \int_0^{u_i} \psi^{-T}(\bar{U}^{-1} s) \bar{U} R \mathrm{d}s \tag{3}$$

where $\psi^{-1}(u_i) = \left[\phi^{-1}(u_{1i}), \phi^{-1}(u_{2i}), \ldots, \phi^{-1}(u_{mi})\right]^T$, R is positive definite and assumed to be diagonal for simplicity of analysis, $s \in \mathbb{R}^m, \psi \in \mathbb{R}^m, \psi^{-T}$ denotes $(\psi^{-1})^T$, and $\phi(\cdot)$

is a bounded one-to-one function satisfying $|\phi(\cdot)| \leq 1$ and belonging to $C^p(p \geq 1)$ and $L_2(\Omega)$. Moreover, it is a monotonic odd function with its first derivative bounded by a constant M. The well-known hyperbolic tangent function $\phi(\cdot) = \tanh(\cdot)$ is one example of such function. Besides, it is important to note that $Y(u_i)$ is positive definite since $\phi^{-1}(\cdot)$ is a monotonic odd function and R is positive definite.

For optimal control problems, the designed control law must be admissible, which connotes that it must not only stabilize the system on Ω but also guarantee the cost function to be finite.

Definition 2 A control $u(x_k)$ is said to be admissible with respect to (2) on Ω if $u(x_k)$ is continuous on a compact set $\Omega_u \in \mathbb{R}^m, u(0) = 0, u$ stabilizes (1) on Ω, and $\forall x_0 \in \Omega, J(x_0)$ is finite.

Note that (2) can be written as

$$\begin{aligned} J(x_k) &= x_k^T Q x_k + Y(u_k) + \gamma \sum_{i=k+1}^{\infty} \gamma^{i-k-1} U(x_i, u_i) \\ &= x_k^T Q x_k + Y(u_k) + \gamma J(x_{k+1}). \end{aligned} \tag{4}$$

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

$$J^*(x_k) = \min_{u_k} \left\{ x_k^T Q x_k + 2 \int_0^{u_k} \psi^{-T}(\bar{U}^{-1} s) \bar{U} R \mathrm{d}s + \gamma J^*(x_{k+1}) \right\}. \tag{5}$$

Besides, the optimal control law u^* satisfies the first-order necessary condition, which is given by the gradient of the right-hand side of (5) with respect to u_k, i.e.,

$$\begin{aligned} & u^*(x_k) \\ &= \arg\min_{u_k} \left\{ x_k^T Q x_k + 2 \int_0^{u_k} \psi^{-T}(\bar{U}^{-1} s) \bar{U} R \mathrm{d}s + \gamma J^*(x_{k+1}) \right\} \\ &= \bar{U} \psi \left(-\frac{\gamma}{2} (\bar{U} R)^{-1} g^T(x_k) \frac{\partial J^*(x_{k+1})}{\partial x_{k+1}} \right). \end{aligned} \tag{6}$$

After substituting (6) into (5), the DTHJB equation can be expressed as

$$\begin{aligned} J^*(x_k) = x_k^T Q x_k + 2 & \int_0^{u^*(x_k)} \psi^{-T}(\bar{U}^{-1} s) \bar{U} R \mathrm{d}s \\ & + \gamma J^*(f(x_k) + g(x_k) u^*(x_k)) \end{aligned} \tag{7}$$

where $J^*(x_k)$ is the optimal cost function corresponding to the optimal control law $u^*(x_k)$. When dealing with the linear quadratic regulator (LQR) optimal control problems, this equation reduces to the Riccati equation which can be efficiently solved. However, in the general nonlinear case, the HJB equation cannot be solved exactly. Therefore, we will present a novel algorithm to approximate the cost function iteratively in the following section.

3 Derivation, convergence analysis, and the NN implementation of the iterative ADP algorithm

Three subsections are included in this section. In the first subsection, the iterative ADP algorithm is introduced. In the second subsection, the corresponding convergence proof of the iterative algorithm is presented. Then, in the third subsection, the implementation of the iterative ADP algorithm based on NN is described.

3.1 Derivation of the iterative ADP algorithm

Since direct solution of the HJB equation is computationally intensive, in this subsection, we develop an iterative ADP algorithm, based on Bellman's principle of optimality and the greedy iteration principle.

First, let the initial cost function $V_0(\cdot) = 0$. Then, we can derive the law of single control vector $v_0(x_k)$ using

$$v_0(x_k) = \arg\min_{u_k} \left\{ x_k^T Q x_k + Y(u_k) + \gamma V_0(x_{k+1}) \right\}. \tag{8}$$

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

$$\begin{aligned} V_1(x_k) &= \min_{u_k} \left\{ x_k^T Q x_k + Y(u_k) + \gamma V_0(x_{k+1}) \right\} \\ &= x_k^T Q x_k + Y(v_0(x_k)). \end{aligned} \tag{9}$$

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

$$\begin{aligned} v_i(x_k) &= \arg\min_{u_k} \left\{ x_k^T Q x_k + Y(u_k) + \gamma V_i(x_{k+1}) \right\} \\ &= \bar{U} \psi \left(-\frac{\gamma}{2} (\bar{U} R)^{-1} g^T(x_k) \frac{\partial V_i(x_{k+1})}{\partial x_{k+1}} \right) \end{aligned} \tag{10}$$

and the cost function

$$\begin{aligned} V_{i+1}(x_k) &= \min_{u_k} \left\{ x_k^T Q x_k + Y(u_k) + \gamma V_i(x_{k+1}) \right\} \\ &= x_k^T Q x_k + Y(v_i(x_k)) + \gamma V_i(f(x_k) + g(x_k) v_i(x_k)). \end{aligned} \tag{11}$$

In the above iterative algorithm, i is the iteration index of the cost function and the control law, while k is the time index. The cost function and control law are updated until they converge to the optimal ones. In the following part, we will present the convergence analysis of the iteration

between (10) and (11) with the cost function $V_i \to J^*$ and the control law $v_i \to u^*$ as $i \to \infty$.

3.2 Convergence analysis of the iterative ADP algorithm

Lemma 1 *Let $\{\mu_i\}$ be any arbitrary sequence of control laws and $\{v_i\}$ be the control laws as in (10). Define V_i as in (11) and Λ_i be*

$$\Lambda_{i+1}(x_k) = x_k^T Q x_k + Y(\mu_i(x_k)) + \gamma \Lambda_i(f(x_k) + g(x_k)\mu_i(x_k)). \tag{12}$$

If $V_0(x_k) = \Lambda_0(x_k) = 0$, then $V_i(x_k) \le \Lambda_i(x_k), \forall i$.

Proof It can easily be derived by noticing that V_{i+1} is the result of minimizing the right-hand side of (11) with respect to the control input u_k, while Λ_{i+1} is a result of an arbitrary control input. □

Lemma 2 *Let the sequence $\{V_i\}$ be defined as in (11). If the system is controllable, there is an upper bound B such that $0 \le V_i(x_k) \le B$, $\forall$ i.*

Proof Let $\eta(x_k)$ be any stabilizing and admissible control input, and let $V_0(\cdot) = Z_0(\cdot) = 0$, where V_i is updated as in (11) and Z_i is updated by

$$Z_{i+1}(x_k) = x_k^T Q x_k + Y(\eta(x_k)) + \gamma Z_i(x_{k+1}). \tag{13}$$

The difference of $Z_i(x_k)$ can be derived as follows:

$$\begin{aligned} Z_{i+1}(x_k) - Z_i(x_k) &= \gamma(Z_i(x_{k+1}) - Z_{i-1}(x_{k+1})) \\ &= \gamma^2(Z_{i-1}(x_{k+2}) - Z_{i-2}(x_{k+2})) \\ &= \gamma^3(Z_{i-2}(x_{k+3}) - Z_{i-3}(x_{k+3})) \\ &\vdots \\ &= \gamma^i(Z_1(x_{k+i}) - Z_0(x_{k+i})) \\ &= \gamma^i Z_1(x_{k+i}). \end{aligned} \tag{14}$$

Then, we can obtain that

$$\begin{aligned} Z_{i+1}(x_k) &= \gamma^i Z_1(x_{k+i}) + Z_i(x_k) \\ &= \gamma^i Z_1(x_{k+i}) + \gamma^{i-1} Z_1(x_{k+i-1}) + Z_{i-1}(x_k) \\ &= \gamma^i Z_1(x_{k+i}) + \gamma^{i-1} Z_1(x_{k+i-1}) \\ &\quad + \gamma^{i-2} Z_1(x_{k+i-2}) + Z_{i-2}(x_k) \\ &= \gamma^i Z_1(x_{k+i}) + \gamma^{i-1} Z_1(x_{k+i-1}) \\ &\quad + \gamma^{i-2} Z_1(x_{k+i-2}) + \cdots \\ &\quad + \gamma Z_1(x_{k+1}) + Z_1(x_k). \end{aligned} \tag{15}$$

It is clear that (15) can also be written as

$$\begin{aligned} Z_{i+1}(x_k) &= \sum_{j=0}^{i} \gamma^j Z_1(x_{k+j}) \\ &= \sum_{j=0}^{i} \gamma^j \left(x_{k+j}^T Q x_{k+j} + Y(\eta(x_{k+j})) \right) \\ &\le \sum_{j=0}^{\infty} \gamma^j \left(x_{k+j}^T Q x_{k+j} + Y(\eta(x_{k+j})) \right). \end{aligned} \tag{16}$$

Since $\eta(x_k)$ is a stabilizing and admissible control input, i.e., $x_k \to 0$ as $k \to \infty$, we have

$$Z_{i+1}(x_k) \le \sum_{j=0}^{\infty} \gamma^j Z_1(x_{k+j}) \le B, \ \forall i. \tag{17}$$

By using Lemma 1, we can further get

$$V_{i+1}(x_k) \le Z_{i+1}(x_k) \le B, \ \forall i. \tag{18}$$

Based on Lemmas 1 and 2, we now present our main theorems. □

Theorem 1 *Define the cost function sequence $\{V_i\}$ as in (11) with $V_0(\cdot) = 0$, and the control law sequence $\{v_i\}$ as in (10). Then, $\{V_i\}$ is a nondecreasing sequence satisfying $V_{i+1} \ge V_i$, $\forall i$.*

Proof Define a new sequence

$$\Phi_{i+1}(x_k) = x_k^T Q x_k + Y(v_{i+1}(x_k)) + \gamma \Phi_i(x_{k+1}) \tag{19}$$

with $\Phi_0(\cdot) = V_0(\cdot) = 0$. Let the control law sequence $\{v_i\}$ be defined as in (10), and the cost function sequence $\{V_i\}$ be updated by (11).

In the following part, we prove that $\Phi_i(x_k) \le V_{i+1}(x_k)$ by mathematical induction.

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

$$V_1(x_k) - \Phi_0(x_k) = x_k^T Q x_k + Y(v_0(x_k)) \ge 0,$$

we get

$$V_1(x_k) \ge \Phi_0(x_k). \tag{20}$$

Second, we assume that it holds for $i - 1$, i.e., $V_i(x_k) \ge \Phi_{i-1}(x_k), \forall x_k$. Then for i, according to (11) and (19), we get

$$V_{i+1}(x_k) - \Phi_i(x_k) = \gamma(V_i(x_{k+1}) - \Phi_{i-1}(x_{k+1})) \ge 0$$

i.e.,

$$V_{i+1}(x_k) \ge \Phi_i(x_k). \tag{21}$$

Thus, we complete the proof by mathematical induction.

Furthermore, from Lemma 1, we know that $V_i(x_k) \le \Phi_i(x_k)$. Therefore, we have

$$V_{i+1}(x_k) \ge \Phi_i(x_k) \ge V_i(x_k). \tag{22}$$

As a result, we can obtain the conclusion that $\{V_i\}$ is a monotonically nondecreasing sequence with an upper bound, and therefore, its limit exists. Here, we define it as $\lim_{i\to\infty} V_i(x_k) = V_\infty(x_k)$.

Next, we will prove that

$$V_\infty(x_k) = \min_{u_k}\left\{x_k^T Q x_k + Y(u_k) + \gamma V_\infty(x_{k+1})\right\}. \tag{23}$$

□

Theorem 2 *Define the cost function sequence* $\{V_i\}$ *as in* (11) *with* $V_0(\cdot) = 0$, *and the control law sequence* $\{v_i\}$ *as in* (10). *The sequence* $\{V_i\}$ *converges to the optimal cost function of the DTHJB equation* (5), *i.e.*, $V_i \to J^*$ *as* $i \to \infty$. *Meanwhile, the control law sequence also converges to the optimal control law* (6), *i.e.*, $v_i \to u^*$ *as* $i \to \infty$.

Proof For any u_k and i, according to (11), we can derive

$$V_i(x_k) \le x_k^T Q x_k + Y(u_k) + \gamma V_{i-1}(x_{k+1}).$$

Combining with

$$V_i(x_k) \le V_\infty(x_k),\ \forall i \tag{24}$$

which is obtained from (22), we have

$$V_i(x_k) \le x_k^T Q x_k + Y(u_k) + \gamma V_\infty(x_{k+1}),\ \forall i.$$

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

$$V_\infty(x_k) \le x_k^T Q x_k + Y(u_k) + \gamma V_\infty(x_{k+1}).$$

Note that in the above equation, u_k is chosen arbitrarily; thus, it implies that

$$V_\infty(x_k) \le \min_{u_k}\left\{x_k^T Q x_k + Y(u_k) + \gamma V_\infty(x_{k+1})\right\}. \tag{25}$$

On the other hand, since the cost function sequence satisfies

$$V_i(x_k) = \min_{u_k}\left\{x_k^T Q x_k + Y(u_k) + \gamma V_{i-1}(x_{k+1})\right\}$$

for any i, considering (24), we have

$$V_\infty(x_k) \ge \min_{u_k}\left\{x_k^T Q x_k + Y(u_k) + \gamma V_{i-1}(x_{k+1})\right\},\ \forall i.$$

Let $i \to \infty$, we can obtain that

$$V_\infty(x_k) \ge \min_{u_k}\left\{x_k^T Q x_k + Y(u_k) + \gamma V_\infty(x_{k+1})\right\}. \tag{26}$$

Based on (25) and (26), we can conclude that (23) is true.

We have just proved that the cost function $V_\infty(x_k)$ satisfies the DTHJB equation, and therefore, it is the optimal cost function of the DTHJB equation. Accordingly, we say that the cost function sequence converges to the optimal cost function of the DTHJB equation, i.e., $\lim_{i\to\infty} V_i(x_k) = J^*(x_k)$. Simultaneously, according to (6) and (10), we can conclude that the corresponding control law sequence also converges to the optimal one. □

3.3 NN implementation of the iterative ADP algorithm via GDHP technique

As is known, when the controlled system is linear and the cost function is quadratic, we can obtain a linear control law when solving the optimal control problems. However, in the nonlinear case, this is not necessarily true. Therefore, we need to use function approximation structure, such as NN, to approximate both the control law and the cost function.

Let the number of hidden layer neurons be denoted by l, the weight matrix between the input layer and hidden layer be denoted by v, and the weight matrix between the hidden layer and output layer be denoted by ω. Then, the output of three-layer NN is represented by

$$\hat{F}(X, v, \omega) = \omega^T \sigma\left(v^T X\right) \tag{27}$$

where $\sigma(v^T X) \in \mathbb{R}^l, [\sigma(z)]_i = (e^{z_i} - e^{-z_i})/(e^{z_i} + e^{-z_i}), i = 1, 2, \ldots, l$, are the activation function.

Now, we implement iterative GDHP algorithm in (10) and (11). In the iterative GDHP algorithm, there are three NNs, which are model network, critic network, and action network. All the networks are chosen as three-layer feed-forward neural networks. The inputs of the critic network and action network are x_k, and the inputs of the model network are x_k and $\hat{v}_i(x_k)$. The structural diagram of the proposed iterative GDHP algorithm is shown in Fig. 1, where

$$W = \left(\frac{\partial \hat{x}_{k+1}}{\partial x_k} + \frac{\partial \hat{x}_{k+1}}{\partial \hat{v}_i(x_k)}\frac{\partial \hat{v}_i(x_k)}{\partial x_k}\right)^T.$$

In order to avoid the requirement of knowing the system dynamics, we should train the model network before carrying out the iterative algorithm, which is in fact the system identification process. For given x_k and $\hat{v}_i(x_k)$, we can obtain the output of the model network as

$$\hat{x}_{k+1} = \omega_m^T \sigma\left(v_m^T\left[x_k^T\ \hat{v}_i^T(x_k)\right]^T\right). \tag{28}$$

We define the error function of the model network as

$$e_{\mathrm{mk}} = \hat{x}_{k+1} - x_{k+1}. \tag{29}$$

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

$$E_{\mathrm{mk}} = \frac{1}{2} e_{\mathrm{mk}}^T e_{\mathrm{mk}}. \tag{30}$$

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

$$\omega_m(j+1) = \omega_m(j) - \alpha_m\left[\frac{\partial E_{\mathrm{mk}}}{\partial \omega_m(j)}\right] \tag{31}$$

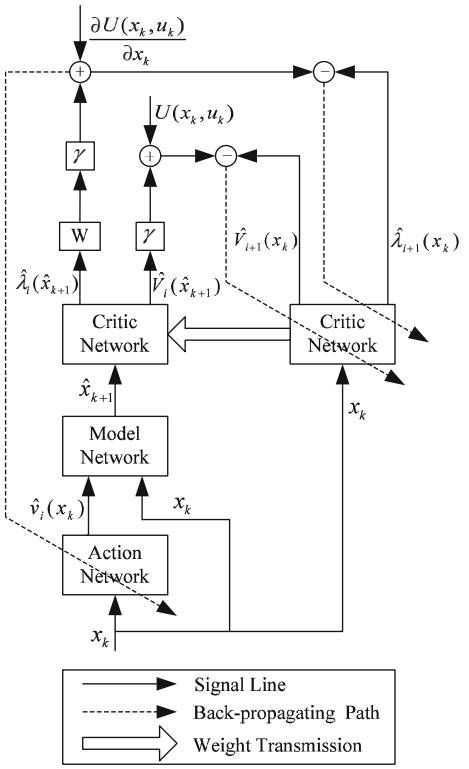

Fig. 1 The structure diagram of the iterative GDHP algorithm

$$v_m(j+1) = v_m(j) - \alpha_m\left[\frac{\partial E_{\mathrm{mk}}}{\partial v_m(j)}\right] \tag{32}$$

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.

The critic network is used to approximate both $V_i(x_k)$ and its derivative $\partial V_i(x_k)/\partial x_k$, which is denoted as $\lambda_i(x_k)$. The output of the critic network can be formulated as

$$\begin{bmatrix}\hat{V}_i(x_k)\\ \hat{\lambda}_i(x_k)\end{bmatrix} = \begin{bmatrix}\omega_{\mathrm{ci}}^{1T}\\ \omega_{\mathrm{ci}}^{2T}\end{bmatrix}\sigma\left(v_{\mathrm{ci}}^T x_k\right) = \omega_{\mathrm{ci}}^T\sigma\left(v_{\mathrm{ci}}^T x_k\right) \tag{33}$$

where

$$\omega_{\mathrm{ci}} = \left[\omega_{\mathrm{ci}}^1\ \omega_{\mathrm{ci}}^2\right]$$

i.e.,

$$\hat{V}_i(x_k) = \omega_{\mathrm{ci}}^{1T}\sigma\left(v_{\mathrm{ci}}^T x_k\right) \tag{34}$$

and

$$\hat{\lambda}_i(x_k) = \omega_{\mathrm{ci}}^{2T}\sigma\left(v_{\mathrm{ci}}^T x_k\right). \tag{35}$$

The target function can be written as

$$V_{i+1}(x_k) = x_k^T Q x_k + Y(v_i(x_k)) + \gamma\hat{V}_i(\hat{x}_{k+1}) \tag{36}$$

and

$$\begin{aligned}\lambda_{i+1}(x_k) &= \frac{\partial\left(x_k^T Q x_k + Y(v_i(x_k))\right)}{\partial x_k} + \gamma\frac{\partial\hat{V}_i(\hat{x}_{k+1})}{\partial x_k}\\ &= 2Qx_k + 2\left(\frac{\partial v_i(x_k)}{\partial x_k}\right)^T\bar{U}R\psi^{-1}(\bar{U}^{-1}v_i(x_k))\\ &\quad + \gamma\left(\frac{\partial\hat{x}_{k+1}}{\partial x_k} + \frac{\partial\hat{x}_{k+1}}{\partial\hat{v}_i(x_k)}\frac{\partial\hat{v}_i(x_k)}{\partial x_k}\right)^T\hat{\lambda}_i(\hat{x}_{k+1}).\end{aligned} \tag{37}$$

Then, we define error functions for the critic network as

$$e_{\mathrm{cik}}^1 = \hat{V}_i(x_k) - V_{i+1}(x_k) \tag{38}$$

and

$$e_{\mathrm{cik}}^2 = \hat{\lambda}_i(x_k) - \lambda_{i+1}(x_k). \tag{39}$$

The objective function to be minimized for the critic network is

$$E_{\mathrm{cik}} = (1-\theta)E_{\mathrm{cik}}^1 + \theta E_{\mathrm{cik}}^2 \tag{40}$$

where

$$E_{\mathrm{cik}}^1 = \frac{1}{2}e_{\mathrm{cik}}^{1T}e_{\mathrm{cik}}^1 \tag{41}$$

and

$$E_{\mathrm{cik}}^2 = \frac{1}{2}e_{\mathrm{cik}}^{2T}e_{\mathrm{cik}}^2. \tag{42}$$

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

$$\omega_{\mathrm{ci}}(j+1) = \omega_{\mathrm{ci}}(j) - \alpha_c\left[(1-\theta)\frac{\partial E_{\mathrm{cik}}^1}{\partial\omega_{\mathrm{ci}}(j)} + \theta\frac{\partial E_{\mathrm{cik}}^2}{\partial\omega_{\mathrm{ci}}(j)}\right] \tag{43}$$

$$v_{\mathrm{ci}}(j+1) = v_{\mathrm{ci}}(j) - \alpha_c\left[(1-\theta)\frac{\partial E_{\mathrm{cik}}^1}{\partial v_{\mathrm{ci}}(j)} + \theta\frac{\partial E_{\mathrm{cik}}^2}{\partial v_{\mathrm{ci}}(j)}\right] \tag{44}$$

where $\alpha_c > 0$ is the learning rate of the critic network j is the inner-loop iterative step for updating the weight parameters, and $0 \le \theta \le 1$ is a parameter that adjusts how HDP and DHP are combined in GDHP. When $\theta = 0$, the training of the critic network reduces to a pure HDP, while $\theta = 1$ reduces to a pure DHP.

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

$$\hat{v}_i(x_k) = \omega_{\mathrm{ai}}^T\sigma\left(v_{\mathrm{ai}}^T x_k\right). \tag{45}$$

The target control input is given by

$$v_i(x_k) = \bar{U}\psi\left(-\frac{\gamma}{2}(\bar{U}R)^{-1}g^T(x_k)\frac{\partial\hat{V}_i(\hat{x}_{k+1})}{\partial\hat{x}_{k+1}}\right). \tag{46}$$

The error function of the action network can be defined as

$$e_{\text{aik}} = \hat{v}_i(x_k) - v_i(x_k). \tag{47}$$

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

$$E_{\text{aik}} = \frac{1}{2} e_{\text{aik}}^T e_{\text{aik}}. \tag{48}$$

Similarly, the weight update algorithm is

$$\omega_{\text{ai}}(j+1) = \omega_{\text{ai}}(j) - \alpha_a \left[\frac{\partial E_{\text{aik}}}{\partial \omega_{\text{ai}}(j)} \right] \tag{49}$$

$$v_{\text{ai}}(j+1) = v_{\text{ai}}(j) - \alpha_a \left[\frac{\partial E_{\text{aik}}}{\partial v_{\text{ai}}(j)} \right] \tag{50}$$

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.

Remark 1 According to Theorem 2, $V_i \to J^*$ as $i \to \infty$. Since $\lambda_i(x_k) = \partial V_i(x_k)/\partial x_k$, we can conclude that the sequence $\{\lambda_i\}$ is also convergent with $\lambda_i \to \lambda^*$ as $i \to \infty$.

4 Simulation study

In this section, an example is carried out to demonstrate the effectiveness of the iterative GDHP algorithm in solving the constrained optimal control problems.

Consider the following nonlinear discrete-time system:

$$x_{k+1} = \begin{bmatrix} 0.2x_{1k}e^{x_{2k}^2} \\ 0.3x_{2k}^3 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.2 \end{bmatrix} u(x_k)$$

where $x_k = [x_{1k}\, x_{2k}]^T \in \mathbb{R}^2$ and $u_k \in \mathbb{R}$ are the state and control variables, respectively. It is desired to control the system with control constraint of $|u| \le 0.1$. The cost function is chosen as

$$J(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} \left\{ x_i^T Q x_i + 2 \int_0^{u_i} \tanh^{-T}(\bar{U}^{-1}s) \bar{U} R \mathrm{d}s \right\}$$

where Q and R are identity matrices with suitable dimensions.

In order to implement the iterative GDHP algorithm at time instant $k = 0$, we choose three-layer feedforward NNs as model network, critic network, and action network with the structures 3–8–2, 2–8–3, and 2–8–1, 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 1,000 time steps using 100 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, let discount factor $\gamma = 1$ and the adjusting parameter $\theta = 0.5$, we train the critic network and action network for 53 iterations (i.e., for $i = 1, 2, \ldots, 53$) with 2,000 training steps for each iteration to make sure the prespecified accuracy 10^{-6} is reached. In the training process, the learning rate $\alpha_c = \alpha_a = 0.05$. The convergence process of the cost function and its derivative of GDHP algorithm are shown in Fig. 2, for $k = 0$ and $x_0 = [2{-}1]^T$. We can see that the iterative cost function sequence does converge to the optimal cost function quite rapidly, which also indicates the validity of the iterative GDHP algorithm. Incidentally, the derivative of the cost function sequence is also convergent just like the statement in Remark 1.

Then, for the given initial state $x_0 = [2{-}1]^T$, we apply the optimal control law designed by the iterative GDHP algorithm to the controlled nonlinear system for 14 time

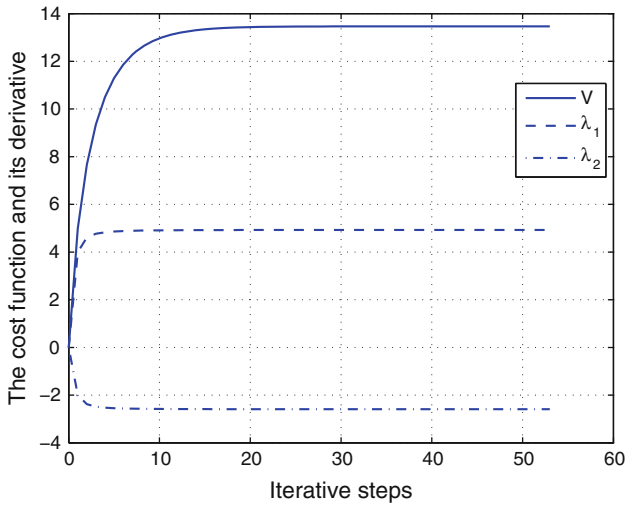

Fig. 2 The convergence process of the cost function and its derivative of the iterative GDHP algorithm

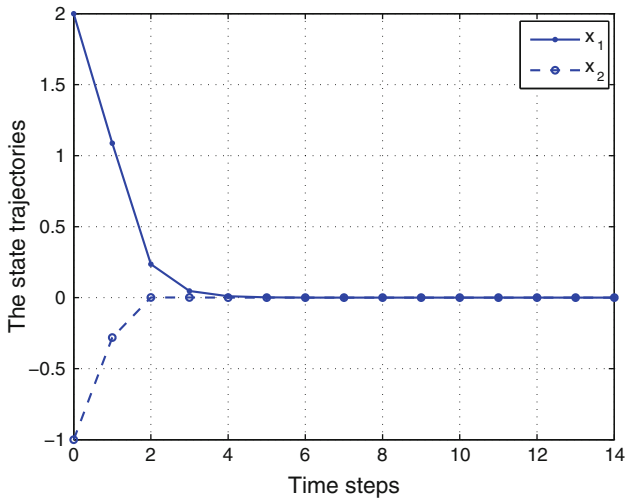

Fig. 3 The state trajectories x

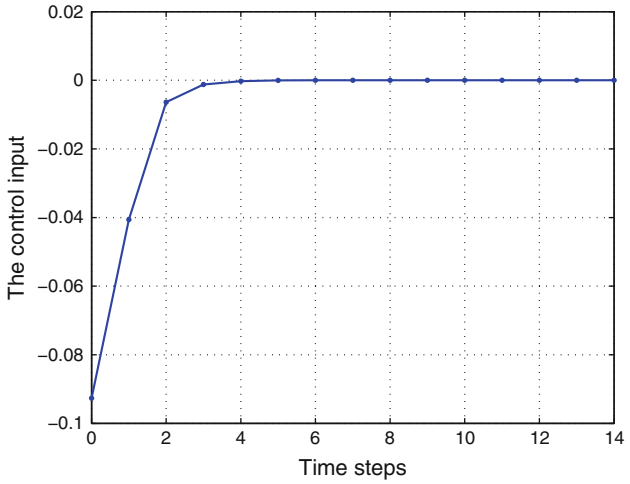

Fig. 4 The control input u

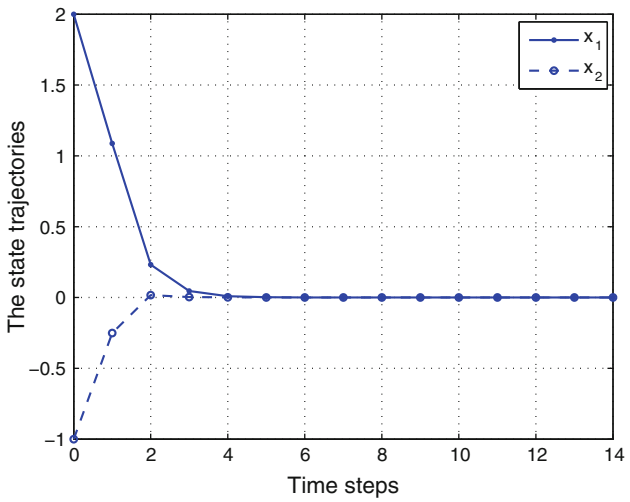

Fig. 5 The state trajectories x without considering the control constraint in controller design

steps and obtain the state trajectories as shown in Fig. 3. The corresponding control input is shown in Fig. 4. Moveover, in order to make comparison with the performance obtained by the controller without considering the actuator saturation, we also present the controller designed by the iterative GDHP algorithm regardless of the actuator saturation and apply it to the same controlled system. The state trajectories and the corresponding control input are shown in Figs. 5 and 6, respectively.

Now, we contrast the results obtained from the above two cases. When comparing Figs. 4 with 6, it can be seen that the restriction of actuator saturation has been overcome successfully in the former. However, in the latter, the control input has overrun the saturation bound, and therefore, is limited to the bounded value. It also should be mentioned that the difference between Figs. 3 and 5 lies in the tiny discrepancy of the response in different time steps. Even so,

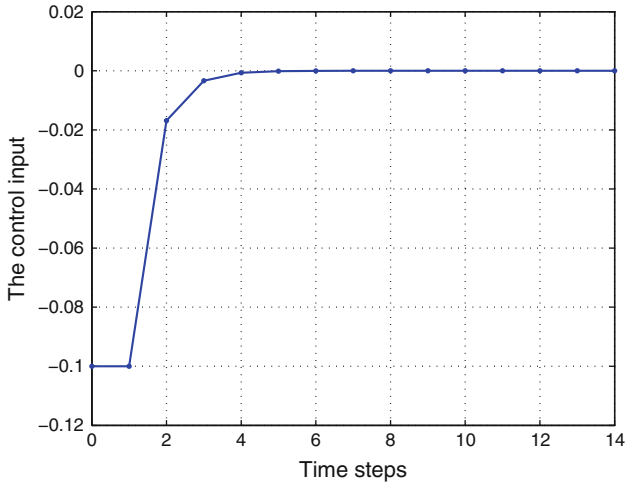

Fig. 6 The control input u without considering the control constraint in controller design

more attention should be given to the difference of the control curves when dealing with this kind of problems. In a word, the simulation results commendably verify the effectiveness of the proposed iterative GDHP algorithm.

5 Conclusions

By employing a generalized nonquadratic functional, an effective iterative algorithm is proposed in this paper to deal with the constrained optimal control problem for a class of nonlinear discrete-time systems. The iterative GDHP algorithm is developed for solving the cost function of the DTHJB equation with convergence analysis. Three NNs are used as parametric structures to approximate at each iteration the cost function, the control law, and the controlled nonlinear system, respectively. The simulation study demonstrated the validity of the present optimal control approach.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grants 60874043, 60904037, 60921061, and 61034002, by Beijing Natural Science Foundation under Grant 4102061, and by the National Science Foundation of USA under Grant ECCS-1027602.

References

1. Chen D, Yang J, Mohler RR (2008) On near optimal neural control of multiple-input nonlinear systems. Neural Comput Appl 17(4):327–337
2. Lyshevski SE (1996) Constrained optimization and control of nonlinear systems: new results in optimal control. In: Proceedings of the 35th IEEE conference on decision and control, Kobe, Japan, pp 541–546
3. Lyshevski SE (1998) Nonlinear discrete-time systems: constrained optimization and application of nonquadratic costs. In:

Proceedings of the American control conference, Philadelphia, pp 3699–3703
4. Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton
5. Jagannathan S (2006) Neural network control of nonlinear discrete-time systems. CRC Press, Boca Raton
6. Yu W (2009) Recent advances in intelligent control systems. Springer, London
7. Werbos PJ (1992) Approximate dynamic programming for real-time control and neural modeling. In: White DA, Sofge DA (eds) Handbook of intelligent control: neural, fuzzy, and adaptive approaches. Van Nostrand Reinhold, New York, pp 493–525
8. Werbos PJ (2008) ADP: The key direction for future research in intelligent control and understanding brain intelligence. IEEE Trans Syst Man Cybern B Cybern 38(4):898–900
9. Werbos PJ (2009) Intelligence in the brain: a theory of how it works and how to build it. Neural Netw 22(3):200–212
10. Murray JJ, Cox CJ, Lendaris GG, Saeks R (2002) Adaptive dynamic programming. IEEE Trans Syst Man Cybern C Appl Rev 32(2):140–153
11. Wang FY, Zhang H, Liu D (2009) Adaptive dynamic programming: an introduction. IEEE Comput Intell Mag 4(2):39–47
12. Lewis FL, Vrabie D (2009) Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits Syst Mag 9(3):32–50
13. Si J, Barto AG, Powell WB, Wunsch DC (2004) Handbook of learning and approximate dynamic programming. IEEE Press/ Wiley, New York
14. Bertsekas DP, Tsitsiklis JN (1996) Neuro-dynamic programming. Athena Scientific, Belmont
15. Si J, Wang YT (2001) On-line learning control by association and reinforcement. IEEE Trans Neural Netw 12(2):264–276
16. Liu D, Zhang H (2005) A neural dynamic programming approach for learning control of failure avoidance problems. Int J Intell Control Syst 10(1):21–32
17. Prokhorov DV, Wunsch DC (1997) Adaptive critic designs. IEEE Trans Neural Netw 8(5):997–1007
18. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. The MIT Press, Cambridge
19. Hagen ST, Krose B (2003) Neural Q-learning. Neural Comput Appl 12(2):81–88
20. Liu D, Xiong X, Zhang Y (2001) Action-dependent adaptive critic designs. In: Proceedings of the international joint conference on neural networks, Washington, vol 2, pp 990–995
21. Venayagamoorthy GK, Harley RG, Wunsch DC (2002) Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator. IEEE Trans Neural Netw 13(3):764–773
22. Venayagamoorthy GK, Harley RG, Wunsch DC (2003) Implementation of adaptive critic-based neurocontrollers for turbogenerators in a multimachine power system. IEEE Trans Neural Netw 14(5):1047–1064
23. Yen GG, Delima PG (2005) Improving the performance of globalized dual heuristic programming for fault tolerant control through an online learning supervisor. IEEE Trans Autom Sci Eng 2(2):121–131
24. Jagannathan S, He P (2008) Neural-network-based state feedback control of a nonlinear discrete-time system in nonstrict feedback form. IEEE Trans Neural Netw 19(12):2073–2087
25. Cheng T, Lewis FL, Abu-Khalaf M (2007) A neural network solution for fixed-final time optimal control of nonlinear systems. Automatica 43(3):482–490
26. Balakrishnan SN, Biega V (1996) Adaptive-critic based neural networks for aircraft optimal control. J Guid Control Dyn 19(4): 893–898
27. Balakrishnan SN, Ding J, Lewis FL (2008) Issues on stability of ADP feedback controllers for dynamic systems. IEEE Trans Syst Man Cybern B Cybern 38(4):913–917
28. Han D, Balakrishnan SN (2002) State-constrained agile missile control with adaptive critic-based neural networks. IEEE Trans Control Syst Technol 10(4):481–489
29. Al-Tamimi A, Lewis FL, Abu-Khalaf M (2008) Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. IEEE Trans Syst Man Cybern B Cybern 38(4):943–949
30. Zhang H, Wei Q, Luo Y (2008) 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 B Cybern 38(4):937–942
31. Vrabie D, Pastravanu O, Abu-Khalaf M, Lewis FL (2009) Adaptive optimal control for continuous-time linear systems based on policy iteration. Automatica 45(2):477–484
32. Liu D, Jin N (2008) ε-adaptive dynamic programming for discrete-time systems. In: Proceedings of the international joint conference on neural networks, Hong Kong, pp 1417–1424
33. Abu-Khalaf M, Lewis FL (2005) Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. Automatica 41(5):779–791
34. Zhang H, Luo Y, Liu D (2009) Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. IEEE Trans Neural Netw 20(9): 1490–1503
35. Vamvoudakis KG, Lewis FL (2010) Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. Automatica 46(5):878–888
36. Zhang H, Wei Q, Liu D (2011) An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. Automatica 47(1):207–214
37. Song R, Zhang H, Luo Y, Wei Q (2010) Optimal control laws for time-delay systems with saturating actuators based on heuristic dynamic programming. Neurocomputing 73(16–18):3020–3027
38. Ma J, Yang T, Hou ZG, Tan M, Liu D (2008) Neurodynamic programming: a case study of the traveling salesman problem. Neural Comput Appl 17(4):347–355