

# Neural-Network-Based Optimal Control for a Class of Nonlinear Discrete-Time Systems With Control Constraints Using the Iterative GDHP Algorithm

Derong Liu, Ding Wang, and Dongbin Zhao

**Abstract**—In this paper, a neural-network-based optimal control scheme for a class of nonlinear discrete-time systems with control constraints is proposed. The iterative adaptive dynamic programming (ADP) algorithm via globalized dual heuristic programming (GDHP) technique is developed to design the optimal controller with convergence proof. Three neural networks are used to facilitate the implementation of the iterative algorithm, which will approximate at each iteration the cost function, the optimal control law, and the controlled nonlinear discrete-time system, respectively. A simulation study is carried out to demonstrate the effectiveness of the present approach in dealing with the nonlinear constrained optimal control problem.

## I. INTRODUCTION

THOUGH classical control schemes work well for controlling linear, single input, single output systems, they are unsuitable for controlling the 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, the 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], [2]. 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, but it is often computationally untenable to run it to obtain the optimal solution. This is because the cost can grow drastically with the number of variables in the environment, which is known as the “curse of dimensionality” [3].

Derong Liu, Ding Wang, and Dongbin Zhao are with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China (email: {derong.liu, ding.wang, dongbin.zhao}@ia.ac.cn).

This work was supported in part by the NSFC under grants 60904037, 60921061, and 61034002, by Beijing Natural Science Foundation under grant 4102061, and by K. C. Wong Education Foundation, Hong Kong.

It is well known that 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 [4]–[8]. Besides, it has been used for universal function approximation in adaptive/approximate dynamic programming (ADP) algorithms, which were proposed in [6]–[8] as a method to solve optimal control problems forward-in-time. There are several synonyms used for ADP including “adaptive dynamic programming” [9], “approximate dynamic programming” [10], “neuro-dynamic programming” [11], “neural dynamic programming” [12], “adaptive critic designs” (ACD) [13], and “reinforcement learning” (RL) [14].

Recently, research in ADP and the related RL have gained much attention from various scholars [6]–[37]. According to [6] and [13], ADP approaches were classified into several main schemes: heuristic dynamic programming (HDP), action-dependent HDP (ADHDP), also known as Q-learning [14], dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP. Using the adaptive-critic-based approach, Venayagamoorthy et al. [18], Han and Balakrishnan [26] presented the neuro-control for a turbogenerator and an agile missile system, respectively. Al-Tamimi et al. [28] proposed a greedy HDP iteration algorithm to solve the DTHJB equation of the optimal control problem for discrete-time nonlinear systems. Vrabie et al. [30] studied the continuous-time optimal control problem using ADP. Liu and Jin [31] derived an  $\varepsilon$ -ADP algorithm for finite horizon discrete-time nonlinear systems. Abu-Khalaf and Lewis [32], Zhang et al. [33] studied the near-optimal control of affine nonlinear systems with control constraints, respectively.

It should be noted that the above results are often obtained through either HDP or DHP technique. Moreover, the GDHP technique has great superiority in combining the two techniques, which is essential in the case of approximating both the cost function and its derivatives. However, there is still no result to solve the optimal control problems for affine nonlinear discrete-time systems with control constraints through the GDHP technique. This motivates our research. In this paper, we will deal with the problem based on the iterative ADP algorithm via GDHP technique (iterative GDHP algorithm for brief).

The rest of this paper is organized as follows. In Section II, the DTHJB equation which includes nonquadratic functional is introduced for the constrained nonlinear discrete-time systems. Section III 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 Section IV, a simulation example is presented to substantiate the derived theoretical results. Section V contains concluding remarks.

## II. PROBLEM STATEMENT

Consider the nonlinear discrete-time system given by

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

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  $\Omega$  in  $\mathbb{R}^n$  containing the origin, and that the system (1) is controllable in the sense that there exists a continuous control on  $\Omega$  that asymptotically stabilizes the system. We denote  $\Omega_u = \{u_k | u_k = [u_{1k}, u_{2k}, \dots, u_{mk}]^T \in \mathbb{R}^m, |u_{ik}| \leq \bar{u}_i, i = 1, 2, \dots, m\}$ , where  $\bar{u}_i$  is the saturating bound for the  $i$ th actuator. Let  $\bar{U} = \text{diag}\{\bar{u}_1, \bar{u}_2, \dots, \bar{u}_m\}$  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  $u_0, u_1, \dots, u_i \in \mathbb{R}^m, i = 0, 1, \dots$ , such that the state  $x_k \rightarrow 0$  as  $k \rightarrow \infty$ .

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

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

where  $U$  is the utility function,  $U(0, 0) = 0$ ,  $U(x_i, u_i) \geq 0$  for  $\forall x_i, u_i$ , and  $\gamma$  is the discount factor with  $0 < \gamma \leq 1$ . 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 [1][2][32], when dealing with the bounded optimal control problems, we can employ a generalized non-quadratic functional

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

where  $\psi^{-1}(u_i) = [\phi^{-1}(u_{1i}), \phi^{-1}(u_{2i}), \dots, \phi^{-1}(u_{mi})]^T$ ,  $R$  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 it must not only stabilize

the system on  $\Omega$  but also guarantee that the cost function is finite.

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

Note that equation (2) can be written as

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

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

$$\begin{aligned} J^*(x_k) &= \min_{u_k} \left\{ x_k^T Q x_k + 2 \int_0^{u_k} \psi^{-T}(\bar{U}^{-1}s) \bar{U} R ds \right. \\ &\quad \left. + \gamma J^*(x_{k+1}) \right\}. \end{aligned} \quad (5)$$

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.,

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

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

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

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.

## III. DERIVATION, CONVERGENCE ANALYSIS, AND THE NEURAL NETWORK IMPLEMENTATION OF THE ITERATIVE ADAPTIVE DYNAMIC PROGRAMMING ALGORITHM

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

### A. 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

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

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

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

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

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

and the cost function

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

In the above iterative algorithm,  $i$  is the iteration index of the control law and the cost function, 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  $V_i \rightarrow J^*$  and the control law  $v_i \rightarrow u^*$  as  $i \rightarrow \infty$ .

### B. 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  $\Lambda_i$  be

$$\begin{aligned} \Lambda_{i+1}(x_k) &= x_k^T Q x_k + Y(\mu_i(x_k)) + \gamma \Lambda_i(f(x_k) \\ &\quad + g(x_k)\mu_i(x_k)). \end{aligned} \quad (12)$$

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

**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  $\Lambda_{i+1}$  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  $0 \leq V_i(x_k) \leq B$ ,  $\forall i$ .

**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

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

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

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

Then, we can obtain that

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

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

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

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

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

By using Lemma 1, we can further get

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

■

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  $V_{i+1} \geq V_i$ ,  $\forall i$ .

**Proof:** Define a new sequence

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

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  $\Phi_i(x_k) \leq V_{i+1}(x_k)$  by mathematical induction.

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

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

we get

$$V_1(x_k) \geq \Phi_0(x_k). \quad (20)$$

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

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

i.e.,

$$V_{i+1}(x_k) \geq \Phi_i(x_k). \quad (21)$$

Thus, we complete the proof by mathematical induction.

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

$$V_{i+1}(x_k) \geq \Phi_i(x_k) \geq V_i(x_k). \quad (22)$$

■

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  $\lim_{i \rightarrow \infty} V_i(x_k) = V_\infty(x_k)$ . In the following part, we will proof that

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

**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.,  $V_i \rightarrow J^*$  as  $i \rightarrow \infty$ . Meanwhile, the control law sequence also converges to the optimal control law (6), i.e.,  $v_i \rightarrow u^*$  as  $i \rightarrow \infty$ .

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

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

Combining with

$$V_i(x_k) \leq V_\infty(x_k), \quad \forall i \quad (24)$$

which is obtained from (22), we have

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

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

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

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

$$V_\infty(x_k) \leq \min_{u_k} \{x_k^T Q x_k + Y(u_k) + \gamma V_\infty(x_{k+1})\}. \quad (25)$$

On the other hand, since the cost function sequence satisfies

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

for any  $i$ , considering (24), we have

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

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

$$V_\infty(x_k) \geq \min_{u_k} \{x_k^T Q x_k + Y(u_k) + \gamma V_\infty(x_{k+1})\}. \quad (26)$$

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.,  $\lim_{i \rightarrow \infty} V_i(x_k) = J^*(x_k)$ . Simultaneously, according to (6) and (10), we can conclude that the corresponding control law sequence also converges to the optimal one. ■

### C. 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 in 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  $\nu$ , and the weight matrix between the hidden layer and output layer be denoted by  $\omega$ . Then the output of three-layer NN is represented by

$$\hat{F}(X, \nu, \omega) = \omega^T \sigma(\nu^T X) \quad (27)$$

where  $\sigma(\nu^T X) \in \mathbb{R}^l$ ,  $[\sigma(z)]_i = (e^{z_i} - e^{-z_i})/(e^{z_i} + e^{-z_i})$ ,  $i = 1, 2, \dots, l$ , 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 chose as three-layer feedforward 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 structure 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

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

We define the error function of the model network as

$$e_{mk} = \hat{x}_{k+1} - x_{k+1}. \quad (29)$$

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

$$E_{mk} = \frac{1}{2} e_{mk}^T e_{mk}. \quad (30)$$

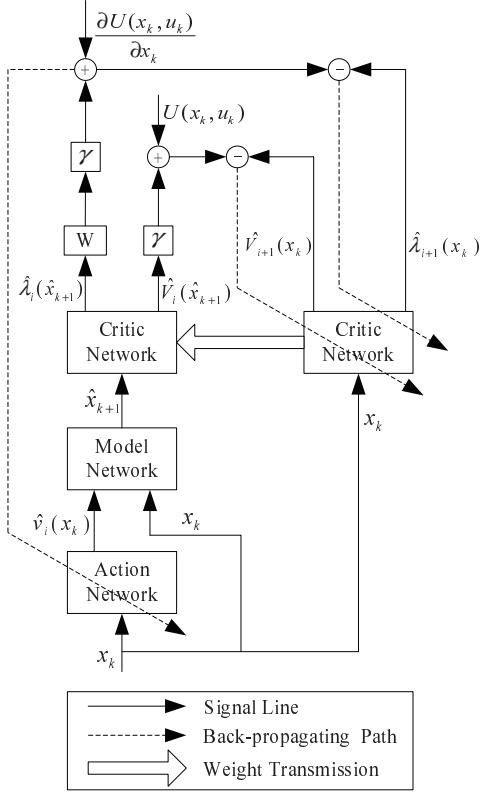


Fig. 1. The structure diagram of the iterative GDHP algorithm

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

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

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

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

$$\begin{bmatrix} \hat{V}_i(x_k) \\ \hat{\lambda}_i(x_k) \end{bmatrix} = \begin{bmatrix} \omega_{ci}^{1T} \\ \omega_{ci}^{2T} \end{bmatrix} \sigma(\nu_{ci}^T x_k) = \omega_{ci}^T \sigma(\nu_{ci}^T x_k) \quad (33)$$

where

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

i.e.,

$$\hat{V}_i(x_k) = \omega_{ci}^{1T} \sigma(\nu_{ci}^T x_k) \quad (34)$$

and

$$\hat{\lambda}_i(x_k) = \omega_{ci}^{2T} \sigma(\nu_{ci}^T x_k). \quad (35)$$

The target function can be written as

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

and

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

Then, we define error functions for the critic network as

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

and

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

The objective function to be minimized for the critic network is

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

where

$$E_{cik}^1 = \frac{1}{2} e_{cik}^{1T} e_{cik}^1 \quad (41)$$

and

$$E_{cik}^2 = \frac{1}{2} e_{cik}^{2T} e_{cik}^2. \quad (42)$$

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

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

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

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  $0 \leq \theta \leq 1$  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  $\theta = 1$  does the same for 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

$$\hat{v}_i(x_k) = \omega_{ai}^T \sigma(\nu_{ai}^T x_k). \quad (45)$$

The target control input is given by

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

The error function of the action network can be defined as

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

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

$$E_{aik} = \frac{1}{2} e_{aik}^T e_{aik}. \quad (48)$$

Similarly, the weight update algorithm is

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

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

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,  $V_i \rightarrow J^*$  as  $i \rightarrow \infty$ . 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  $\lambda_i \rightarrow \lambda^*$  as  $i \rightarrow \infty$ .

#### IV. 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  $|u| \leq 0.1$ . The cost function is chosen as

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

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 NN as model network, critic network and action network with the structures 3–8–2, 2–8–3, 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 1000 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  $i = 1, 2, \dots, 53$ ) with 2000 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. It should be mentioned that the parameters often have apparent impact on our algorithm. This lies in that the smaller discount factor can bring on quicker convergence of the cost function sequence, while too small learning rates may slower the implementation process of our iterative algorithm.

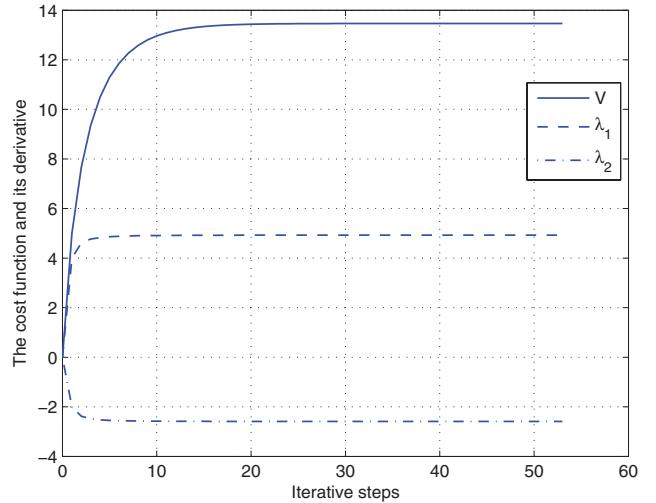


Fig. 2. The convergence process of the cost function and its derivative of the iterative GDHP algorithm

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 steps, and obtain the state trajectories as shown in Fig. 3. The corresponding control input is shown in Fig. 4. Moreover, 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 Fig. 5 and Fig. 6, respectively. It can clearly be seen from the simulation results that the restriction of actuator saturation has been overcome successfully, which verifies the effectiveness of the proposed iterative GDHP algorithm.

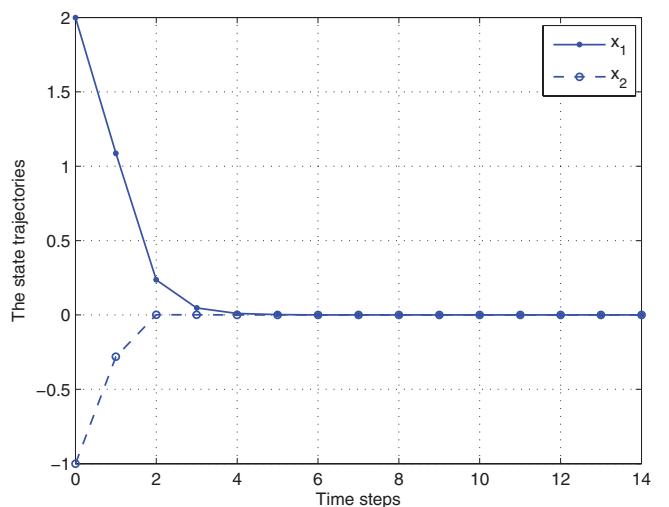


Fig. 3. The state trajectories  $x$

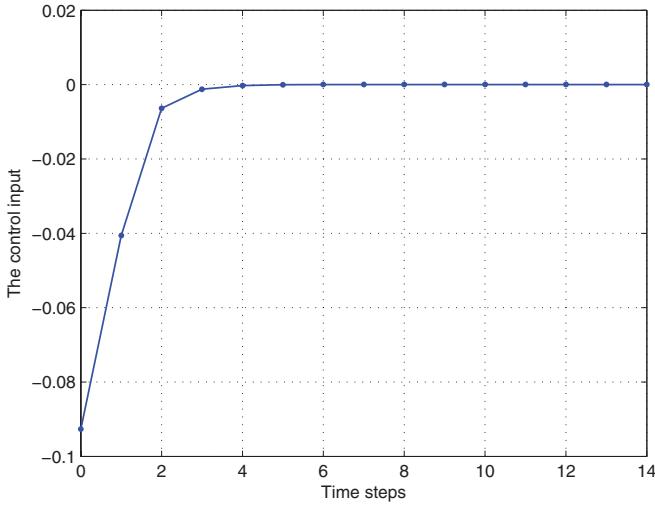


Fig. 4. The control input  $u$

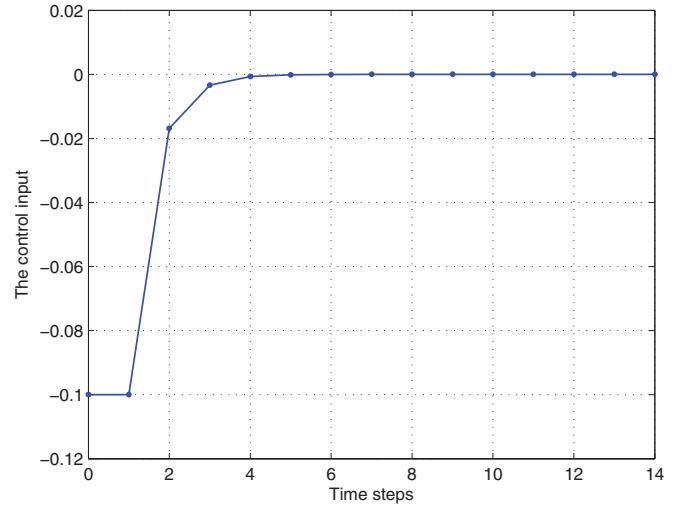


Fig. 6. The control input  $u$  without considering the control constraint

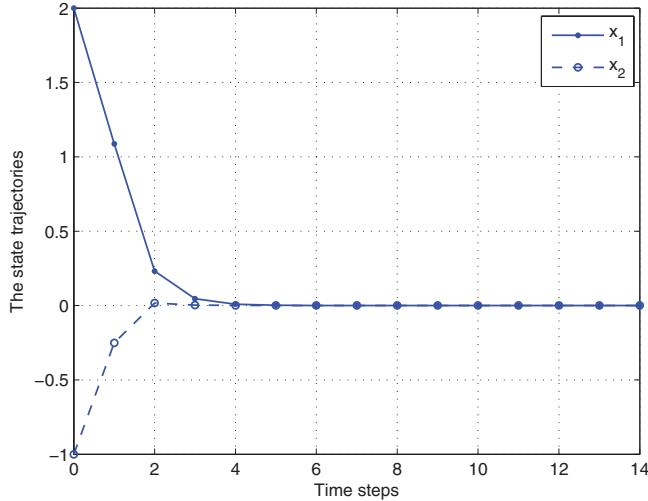


Fig. 5. The state trajectories  $x$  without considering the control constraint

## V. CONCLUSIONS

Since it is computationally expensive to use DP to solve the nonlinear optimal control problems, the ADP method, which combines RL, NN and DP, has become an important methodology in the area of computational intelligence. Werbos [8] indicated that ADP is one of the core technologies to potentially achieve the brain-like general-purpose intelligence and bridge the gap between theoretic study and challenging real-world engineering applications. Consequently, extensive efforts have been devoted to ADP related research and tremendous progresses have been achieved through HDP or DHP technique. On the other hand, Prokhorov and Wunsch [13] suggested that the outputs of the critic network of the GDHP technique contain not only the cost function but also

its derivative. 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 derivative. Therefore, it will bring outstanding performance when using the iterative GDHP algorithm to tackle the constrained optimal control problems of nonlinear discrete-time systems. This is the initial motivation of our research.

In this paper, an effective iterative algorithm is proposed to deal with the near optimal control for a class of nonlinear discrete-time systems with control constraints. The iterative GDHP algorithm is introduced to solve 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 nonlinear system, respectively. The detailed adaption and tuning of parameters in all NNs are also presented. The simulation study demonstrated the validity of the proposed optimal control approach. Simulation results also show that the restriction of actuator saturation can be overcome successfully by using the presented iterative GDHP algorithm.

Incidentally, it should be noticed that the strategy proposed in this paper only be true for a class of affine nonlinear systems. Needless to say, it is necessary to broaden its applicability to a more general classes of nonlinear systems. In other words, our future work will focus on solving the constrained optimal control problems for more general nonlinear systems.

## REFERENCES

- [1] S. E. Lyshevski, "Constrained optimization and control of nonlinear systems: new results in optimal control," in *Joint 35th IEEE Conference on Decision and Control*, Kobe, Japan, Dec. 1996, pp. 541–546.
- [2] S. E. Lyshevski, "Nonlinear discrete-time systems: constrained optimization and application of nonquadratic costs," in *Proc. Amer. Control Conf.*, Philadelphia, PA, Jun. 1998, pp. 3699–3703.
- [3] R. E. Bellman, *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

- [4] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-time Systems*. CRC Press, Boca Raton, FL, 2006.
- [5] W. Yu, *Recent Advances in Intelligent Control Systems*. Springer-Verlag, London, 2009.
- [6] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control*, D. A. White and D. A. Sofge, Eds. Van Nostrand Reinhold, New York, 1992, ch. 13.
- [7] P. J. Werbos, "ADP: The key direction for future research in intelligent control and understanding brain intelligence," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 898–900, Aug. 2008.
- [8] P. J. Werbos, "Intelligence in the brain: a theory of how it works and how to build it," *Neural Networks*, vol. 22, no. 3, pp. 200–212, Apr. 2009.
- [9] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 2, pp. 140–153, May 2002.
- [10] J. Si, A. G. Barto, W. B. Powell, and D. C. Wunsch, Eds., *Handbook of Learning and Approximate Dynamic Programming*. IEEE Press/Wiley, New York, 2004.
- [11] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.
- [12] J. Si and Y. T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.
- [13] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sept. 1997.
- [14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.
- [15] F. Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: an introduction," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 39–47, May 2009.
- [16] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, July 2009.
- [17] D. Liu, X. Xiong, and Y. Zhang, "Action-dependent adaptive critic designs," in *Proc. International Joint Conference on Neural Networks*, Washington, DC, July 2001, vol. 2, pp. 990–995.
- [18] G. K. Venayagamoorthy, R. G. Harley, and D. C. Wunsch, "Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 764–773, May 2002.
- [19] G. K. Venayagamoorthy, R. G. Harley, and D. C. Wunsch, "Implementation of adaptive critic-based neurocontrollers for turbogenerators in a multimachine power system," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1047–1064, Sept. 2003.
- [20] J. W. Park, R. G. Harley, and G. K. Venayagamoorthy, "Adaptive-critic-based optimal neurocontrol for synchronous generators in a power system using MLP/RBF neural networks," *IEEE Trans. Ind. Appl.*, vol. 39, no. 5, pp. 1529–1540, Sept./Oct. 2003.
- [21] G. G. Yen and P. G. Delima, "Improving the performance of globalized dual heuristic programming for fault tolerant control through an online learning supervisor," *IEEE Trans. Automation Science and Engineering*, vol. 2, no. 2, pp. 121–131, Apr. 2005.
- [22] T. Cheng, F. L. Lewis, and M. Abu-Khalaf, "A neural network solution for fixed-final time optimal control of nonlinear systems," *Automatica*, vol. 43, no. 3, pp. 482–490, Mar. 2007.
- [23] S. N. Balakrishnan and V. Biega, "Adaptive-critic based neural networks for aircraft optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 4, pp. 893–898, July–Aug. 1996.
- [24] S. N. Balakrishnan, J. Ding, and F. L. Lewis, "Issues on stability of ADP feedback controllers for dynamic systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 913–917, Aug. 2008.
- [25] R. Padhi, S. N. Balakrishnan, and T. Randolph, "Adaptive-critic based optimal neuro control synthesis for distributed parameter systems," *Automatica*, vol. 37, no. 8, pp. 1223–1234, Aug. 2001.
- [26] D. Han and S. N. Balakrishnan, "State-constrained agile missile control with adaptive critic-based neural networks," *IEEE Trans. Control Systems Technology*, vol. 10, no. 4, pp. 481–489, July 2002.
- [27] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan, "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Networks*, vol. 19, no. 10, pp. 1648–1660, Dec. 2006.
- [28] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.
- [29] H. Zhang, Q. Wei, and 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. B, Cybern.*, vol. 38, no. 4, pp. 937–942, Aug. 2008.
- [30] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, Feb. 2009.
- [31] D. Liu and N. Jin, " $\varepsilon$ -adaptive dynamic programming for discrete-time systems," in *Proc. International Joint Conference on Neural Networks*, Hong Kong, June 2008, pp. 1417–1424.
- [32] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, 779–791, May 2005.
- [33] H. Zhang, Y. Luo, and D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sept. 2009.
- [34] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.
- [35] Z. Sun, X. Chen, and Z. He, "Adaptive critic designs for energy minimization of portable video communication devices," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 27–37, Jan. 2010.
- [36] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 14–25, Feb. 2011.
- [37] F. Y. Wang, N. Jin, D. Liu, and Q. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with  $\varepsilon$ -error bound," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.