Neurocomputing 121 (2013) 218–225

Contents lists available at ScienceDirect

Neurocomputing

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

Neuro-optimal control for a class of unknown nonlinear dynamic systems using SN-DHP technique ☆

Ding Wang, 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 4 April 2012
Received in revised form
15 February 2013
Accepted 24 April 2013
Communicated by M.-J. Er
Available online 15 May 2013

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

ABSTRACT

In this paper, the adaptive dynamic programming (ADP) approach is utilized to design a neural-network-based optimal controller for a class of unknown discrete-time nonlinear systems with quadratic cost function. To begin with, a neural network identifier is constructed to learn the unknown dynamic system with stability proof. Then, the iterative ADP algorithm is developed to handle the nonlinear optimal control problem with convergence analysis. Moreover, the single network dual heuristic dynamic programming (SN-DHP) technique, which eliminates the use of action network, is introduced to implement the iterative ADP algorithm. Finally, two simulation examples are included to illustrate the effectiveness of the present approach.

1. Introduction

The researches of nonlinear systems have been the focus of control field for many decades and various topics are concerned, such as nonlinear adaptive control [1–3], nonlinear optimal control [4–7], etc. Among that, the nonlinear optimal control problem often requires solving the nonlinear Hamilton–Jacobi–Bellman (HJB) equation [4–7]. However, the discrete-time HJB (DTHJB) equation is more difficult to deal with than the Riccati equation because it involves solving nonlinear partial difference equations. Though dynamic programming (DP) has been an useful computational technique in solving optimal control problems for many years, it is often computationally untenable to run it to obtain the optimal solution, due to the well-known "curse of dimensionality" [8]. What is worse, the backward direction of the search process precludes the use of DP in practical control tasks.

In Ref. [9], Poggio and Girosi indicated that the problem of learning between input and output spaces is equivalent to that of synthesizing an associative memory that retrieves appropriate output when the input is presented and generalizes when a new input is applied. With strong capabilities of self-learning and adaptivity, artificial neural networks (ANN or NN) have become an effective tool to implement intelligent control [10–13]. Accordingly, they are always used for universal function approximation in adaptive/approximate dynamic programming (ADP) algorithms, which were proposed in [11–13] as a method to solve optimal control problem forward-in-time. There are several synonyms used for ADP, including "approximate dynamic programming" [12], "neuro-dynamic programming" [14], "adaptive critic design" [15], "neural dynamic programming" [16], "adaptive dynamic programming" [17,18], and "reinforcement learning" [19].

As an emerging and promising intelligent control method, in recent years, ADP and the related research have gained much attention from researchers [12–37]. According to [12,15], ADP approaches were classified into several main schemes: heuristic dynamic programming (HDP), action-dependent HDP (ADHDP), also known as Q-learning [22], dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP. Padhi et al. [23] presented the single network adaptive critic (SNAC) architecture, by eliminating the use of one NN, which results in a simpler architecture, less computational load and elimination of the approximation error associated with the eliminated NN. Liu and Jin [27] developed an ε−ADP algorithm for finite horizon discrete-time nonlinear systems. Al-Tamimi et al. [30] proposed a greedy iterative HDP algorithm to solve the DTHJB equation of the optimal control problem for discrete-time nonlinear systems. Zhang et al. [31] studied the near-optimal control for a class of discrete-time affine nonlinear systems with control constraints

☆This work was supported in part by the National Natural Science Foundation of China under Grants 61034002, 61233001, and 61273140.

* Corresponding author. Tel.: +86 10 62557379; fax: +86 10 62650912.
E-mail addresses: ding.wang@ia.ac.cn (D. Wang), derong.liu@ia.ac.cn, derongliu@gmail.com (D. Liu).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.04.006

using DHP scheme. Abu-Khalaf and Lewis [32] and Vrabie and Lewis [33] studied the continuous-time optimal control problem using ADP approach. Additionally, in order to control the unknown system, Kim and Lewis [37] gave a model-free H_∞ control design scheme for unknown linear discrete-time system via Q-learning, which was expressed in the form of linear matrix inequality. Campi and Savaresi [38] proposed a virtual reference feedback tuning approach which was in fact a data-based method.

Note that in most existing structures of iterative ADP algorithm, like DHP [31] and GDHP [36], up to three networks have to be built. The complicated architecture inevitability leads to the increase of computational burden. The SNAC technique is simple to implement, but it is not combined with the iterative ADP algorithm, not to mention that the system dynamics is required when applying it to design optimal controller. To summarize, there is no result on optimal control of nonlinear system with unknown dynamics using iterative ADP algorithm and SNAC technique. Therefore, in this paper, we study the model-free optimal control of a class of unknown nonlinear systems by using single network DHP (SN-DHP) technique. First of all, an NN model is constructed as an identifier to learn the unknown nonlinear system. Then, based on the identification results, the iterative ADP algorithm is introduced to solve the DTHJB equation with convergence analysis. Furthermore, the optimal controller is designed through the SN-DHP approach.

The main contributions of our work are as follows:

(1) We combine the SNAC technique with the iterative ADP algorithm, and then present the convergence analysis of the designed optimal controller for affine nonlinear systems. The SN-DHP technique is established with model network and critic network. This is different from SNAC, where only critic network is designed. In addition, it is also different from the traditional ADP techniques, where model network, critic network and action network are constructed.
(2) By performing identification process, the iterative ADP algorithm, based on three-layer feedforward NNs and gradient-based adaptation rule, is applicable to deal with the optimal control problem of unknown nonlinear systems. However, the SNAC technique requires knowing the system dynamics.

The rest of the paper is organized as follows. In Section 2, preliminaries of the nonlinear optimal control is introduced. In Section 3, we design an NN identifier for the unknown controlled system with stability proof. In Section 4, the optimal control scheme based on the learned system knowledge and iterative ADP algorithm is developed with convergence analysis. In Section 5, we describe the SN-DHP technique and the implementation process of the iterative algorithm. In Section 6, two examples are given to demonstrate the effectiveness of the proposed control scheme. In Section 7, concluding remarks are given.

2. Background in nonlinear optimal control

In this paper, we consider the nonlinear discrete-time system described 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 vector and $u(x_k) \in \mathbb{R}^m$ is the control vector, $f(\cdot)$ and $g(\cdot)$ are differentiable in their arguments with $f(0)=0$. 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 law on Ω that asymptotically stabilizes the system.

For infinite horizon optimal control problem, it is desired to find the control law $u(x)$ which can minimize the cost function given by

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

where U is the utility function, $U(0,0)=0$, and $U(x_i,u_i)\geq 0$ for $\forall x_i, u_i$. In this paper, the utility function is chosen as the quadratic form as $U(x_i,u_i) = x_i^T Q x_i + u_i^T R u_i$. In addition, the designed feedback control must not only stabilize the system on Ω but also guarantee that (2) is finite, i.e., the control must be admissible. Thus, we give the following definition.

Definition 1. A control $u(x)$ is defined to be admissible with respect to (2) on Ω if $u(x)$ 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.

According to Bellman's optimality principle, the optimal cost function

$$J^*(x_k) = \min_{u_k, u_{k+1}, \ldots, u_\infty} \sum_{i=k}^{\infty} U(x_i, u_i) \tag{3}$$

can be rewritten as

$$J^*(x_k) = \min_{u_k} \left\{ U(x_k, u_k) + \min_{u_{k+1}, \ldots, u_\infty} \sum_{i=k+1}^{\infty} U(x_i, u_i) \right\}. \tag{4}$$

In other words, $J^*(x_k)$ satisfies the DTHJB equation

$$J^*(x_k) = \min_{u_k} \{ U(x_k, u_k) + J^*(x_{k+1}) \}. \tag{5}$$

The corresponding optimal control law u^* is

$$u^*(x_k) = \arg\min_{u_k} \{ U(x_k, u_k) + J^*(x_{k+1}) \}. \tag{6}$$

Unlike the optimal control of linear system, for nonlinear optimal control problems, the DTHJB equation (5) cannot be solved exactly. Next, we will study how to handle this problem using the iterative ADP algorithm.

3. System identification strategy of the controlled plant using neural networks

In this paper, we assume that stable controls are utilized in the identification process.

In this section, a three-layer NN is constructed to identify the unknown system dynamics. Let the number of hidden layer neurons be denoted by l, the ideal weight matrix between the input layer and hidden layer be denoted by ν_m^*, and the ideal weight matrix between the hidden layer and output layer be denoted by ω_m^*. According to the universal approximation property [10] of NN, the system dynamics (1) has an NN representation on a compact set S, which can be written as

$$x_{k+1} = \omega_m^{*T} \sigma(\nu_m^{*T} z_k) + \varepsilon_k. \tag{7}$$

In (7), $z_k = [x_k^T u_k^T]^T$ is the NN input, ε_k is the bounded NN functional approximation error, and $[\sigma(\xi)]_i = (e^{\xi_i} - e^{-\xi_i})/(e^{\xi_i} + e^{-\xi_i})$, $i = 1, 2, \ldots, l$ are the activation functions. Let $\bar{z}_k = \nu_m^{*T} z_k$, $\bar{z}_k \in \mathbb{R}^l$. In addition, the NN activation functions are bounded such that $\|\sigma(\bar{z}_k)\| \leq \sigma_M$ for a constant σ_M.

According to [34], we define the NN system identification scheme as

$$\hat{x}_{k+1} = \omega_m^T(k)\sigma(\bar{z}_k) - r_k \tag{8}$$

where $\hat{x}_k$ is the estimated system state vector, r_k is the robust term, and $\omega_m(k)$ is the estimation of the constant ideal weight matrix ω_m^*.

We denote $\tilde{x}_k = \hat{x}_k - x_k$ as the system identification error. Then, combining (7) and (8), we can obtain the identification error

dynamics as

$$\tilde{x}_{k+1} = \tilde{\omega}_m^T(k)\sigma(\bar{z}_k) - r_k - \varepsilon_k \tag{9}$$

where $\tilde{\omega}_m(k) = \omega_m(k) - \omega_m^*$. Define the robust term as a function of the identification error $\tilde{x}_k$ and an additional tunable parameter $\beta(k) \in \mathbb{R}$, i.e.,

$$r_k = \frac{\beta(k)\tilde{x}_k}{\tilde{x}_k^T\tilde{x}_k + C} \tag{10}$$

where $C > 0$ is a constant. Denote β^* as the constant ideal value of the parameter $\beta(k)$ and let $\tilde{\beta}(k) = \beta(k) - \beta^*$, $\psi_k = \tilde{\omega}_m^T(k)\sigma(\bar{z}_k)$, and $\varphi_k = \tilde{\beta}(k)\tilde{x}_k/(\tilde{x}_k^T\tilde{x}_k + C)$. Then, the system dynamics (9) can be rewritten as

$$\tilde{x}_{k+1} = \psi_k - \varphi_k - \frac{\beta^*\tilde{x}_k}{\tilde{x}_k^T\tilde{x}_k + C} - \varepsilon_k. \tag{11}$$

The parameters in the system identification process are updated to minimize the following performance measure:

$$E_{k+1} = \tfrac{1}{2}\tilde{x}_{k+1}^T\tilde{x}_{k+1}. \tag{12}$$

Using the gradient-based adaptation rule, the NN weight and tunable parameter can be updated as

$$\omega_m(k+1) = \omega_m(k) - \alpha_m\left[\frac{\partial E_{k+1}}{\partial \omega_m(k)}\right] = \omega_m(k) - \alpha_m\sigma(\bar{z}_k)\tilde{x}_{k+1}^T \tag{13}$$

$$\beta(k+1) = \beta(k) - \alpha_r\left[\frac{\partial E_{k+1}}{\partial \beta(k)}\right] = \beta(k) + \alpha_r\frac{\tilde{x}_{k+1}^T\tilde{x}_k}{\tilde{x}_k^T\tilde{x}_k + C} \tag{14}$$

where $\alpha_m > 0$ and $\alpha_r > 0$ are learning rates.

Before presenting the asymptotic stability proof of the state estimation error $\tilde{x}_k$, we now give the following assumption, which has been used in [10,34] and is considered mild in comparison with the approximation error being bounded by a known constant.

Assumption 1 (*cf. Dierks et al. [34]*). The NN approximation error term ε_k is assumed to be upper bounded by a function of the state estimation error $\tilde{x}_k$, i.e.,

$$\varepsilon_k^T\varepsilon_k \le \varepsilon_{Mk} = \delta\tilde{x}_k^T\tilde{x}_k \tag{15}$$

where δ is a bounded constant value such that $\|\delta\| \le \delta_M$.

Theorem 1. *Let the identification scheme (8) be used to identify the nonlinear system (1), and let the parameter update law given in (13) and (14) be used for tuning the NN weights and the robust term, respectively. Then, the state estimation error $\tilde{x}_k$ is asymptotically stable while the parameter estimation error $\tilde{\omega}_m(k)$ and $\tilde{\beta}(k)$ are bounded.*

Proof. Consider the positive definite Lyapunov function candidate defined as

$$L_k = \tilde{x}_k^T\tilde{x}_k + \frac{\tilde{\beta}^2(k)}{\alpha_r} + \frac{1}{\alpha_m}\mathrm{tr}\{\tilde{\omega}_m^T(k)\tilde{\omega}_m(k)\}. \tag{16}$$

In the following proof process, we denote $C_k = \tilde{x}_k^T\tilde{x}_k + C$ for brief. By taking the first difference of the Lyapunov function (16), and considering $\|\sigma(\bar{z}_k)\| \le \sigma_M$ and (15), we can find that

$$\begin{aligned}\Delta L_k = L_{k+1} - L_k \le & -(1 - 4\alpha_m\sigma_M^2 - 4\alpha_r)(\|\psi_k\|^2 + \|\varphi_k\|^2) \\ & -(1 - 2\delta_M - 2\delta_M^2 - 4\alpha_m\delta_M\sigma_M^2 - 4\alpha_m\delta_M^2\sigma_M^2 \\ & -4\alpha_r\delta_M - 4\alpha_r\delta_M^2)\|\tilde{x}_k\|^2 + 2\|\psi_k\|\|\varphi_k\|.\end{aligned} \tag{17}$$

Then, we define $\theta_1\overline{\psi}_k = \psi_k$, $\theta_2\overline{\varphi}_k = \varphi_k$, where θ_1 and θ_2 are constants. After selecting the parameters as $\alpha_m\sigma_M^2 = \alpha_r$, $\theta_1\theta_2 = \rho$, $8\alpha_m\sigma_M^2 \le \theta_1^2$, and applying the Cauchy–Schwarz inequality, (17) becomes

$$\begin{aligned}\Delta L_k \le & -\left(1 - \theta_1^2 - \frac{\rho}{\theta_1^2}\right)\|\psi_k\|^2 - \left(1 - \theta_1^2 - \frac{\theta_1^2}{\rho}\right)\|\varphi_k\|^2 \\ & -(1 - 2\delta_M - 2\delta_M^2 - \delta_M\theta_1^2 - \delta_M^2\theta_1^2)\|\tilde{x}_k\|^2 \\ = & -\left(1 - \theta_1^2 - \frac{\rho}{\theta_1^2}\right)\|\tilde{\omega}_m^T(k)\sigma(\bar{z}_k)\|^2 \\ & -\left(1 - \theta_1^2 - \frac{\theta_1^2}{\rho}\right)\|\tilde{\beta}(k)\|^2\left\|\frac{\tilde{x}_k}{C_k}\right\|^2 \\ & -(1 - 2\delta_M - 2\delta_M^2 - \delta_M\theta_1^2 - \delta_M^2\theta_1^2)\|\tilde{x}_k\|^2.\end{aligned} \tag{18}$$

According to (18), we can conclude that $\Delta L_k \le 0$ provided that $0 < \delta_M \le (\sqrt{3} - 1)/2$, $0 < \rho < 1/4$, and $\tau_1 < \theta_1 < \min\{\tau_2, \tau_3, \tau_4\}$, where

$$\tau_1 = \sqrt{\frac{1 - \sqrt{1 - 4\rho}}{2}}$$

$$\tau_2 = \sqrt{\frac{1 - 2\delta_M - 2\delta_M^2}{\delta_M + \delta_M^2}}$$

$$\tau_3 = \sqrt{\frac{\rho}{1 + \rho}}$$

$$\tau_4 = \sqrt{\frac{1 + \sqrt{1 - 4\rho}}{2}}.$$

As long as the parameters are selected as above, $\Delta L_k \le 0$, which shows stability in the sense of Lyapunov. Therefore, $\tilde{x}_k$, $\tilde{\omega}_m(k)$, and $\tilde{\beta}(k)$ are bounded, provided that $\tilde{x}_0$, $\tilde{\omega}_m(0)$, and $\tilde{\beta}(0)$ are bounded in the compact set S. Furthermore, by summing both sides of (18) to infinity and taking the absolute value, we can obtain

$$\begin{aligned}\sum_{k=0}^{\infty}\Bigg\{ & \left(1 - \theta_1^2 - \frac{\rho}{\theta_1^2}\right)\|\tilde{\omega}_m^T(k)\sigma(\bar{z}_k)\|^2 + \left(1 - \theta_1^2 - \frac{\theta_1^2}{\rho}\right)\|\tilde{\beta}(k)\|^2\left\|\frac{\tilde{x}_k}{C_k}\right\|^2 \\ & +(1 - 2\delta_M - 2\delta_M^2 - \delta_M\theta_1^2 - \delta_M^2\theta_1^2)\|\tilde{x}_k\|^2\Bigg\} \\ \le & \left|\sum_{k=0}^{\infty}\Delta L_k\right| = \left|\lim_{k\to\infty} L_k - L_0\right| < \infty.\end{aligned} \tag{19}$$

From (19), we can conclude that $\|\tilde{x}_k\| \to 0$ as $k \to \infty$. □

In light of Theorem 1, the NN system identification error can approach zero after a sufficiently long learning session. Besides, from (10), the robust term approaches zero as well. Hence, we have

$$x_{k+1} = \omega_m^T(k)\sigma(\bar{z}_k). \tag{20}$$

Taking the partial derivative of both sides of (20) with respect to u_k, we can obtain the estimate of $g(x_k)$ as

$$\hat{g}(x_k) = \frac{\partial(\omega_m^T(k)\sigma(\bar{z}_k))}{\partial u_k} = \omega_m^T(k)\sigma'(\bar{z}_k)\nu_m^{*T}\Theta \tag{21}$$

where

$$\sigma'(\bar{z}_k) = \frac{\partial\sigma(\bar{z}_k)}{\partial\bar{z}_k}$$

$$\Theta = \frac{\partial z_k}{\partial u_k} = \begin{bmatrix} 0_{n\times m} \\ I_m \end{bmatrix}$$

and I_m is an $m \times m$ identity matrix.

In nonlinear optimal control problem, in order to avoid the requirement of knowing the system dynamics, we can replace $g(x_k)$ with $\omega_m^T(k)\sigma'(\bar{z}_k)\nu_m^{*T}\Theta$ when we solve (6). Next, this result will be used in the derivation and implementation of the iterative ADP algorithm.

4. Derivation and convergence analysis of the iterative ADP algorithm

4.1. Derivation of the iterative ADP algorithm

In this section, we introduce the iterative ADP algorithm. First, we start with the initial cost function $V_0(\cdot)=0$ and obtain

$$v_0(x_k)=\arg\min_{u_k}\{U(x_k,u_k)+V_0(x_{k+1})\}. \tag{22}$$

Then, we derive

$$x_{k+1}=f(x_k)+g(x_k)v_0(x_k) \tag{23}$$

and

$$v_0(x_{k+1})=\arg\min_{u_{k+1}}\{U(x_{k+1},u_{k+1})+V_0(x_{k+2})\}. \tag{24}$$

Next, we update the cost function as

$$\begin{aligned}V_1(x_{k+1})&=\min_{u_{k+1}}\{U(x_{k+1},u_{k+1})+V_0(x_{k+2})\}\\&=U(x_{k+1},v_0(x_{k+1}))+V_0(x_{k+2}).\end{aligned} \tag{25}$$

Moreover, for $i=1,2,\ldots$ the algorithm iterates between

$$v_i(x_{k+1})=\arg\min_{u_{k+1}}\{U(x_{k+1},u_{k+1})+V_i(x_{k+2})\} \tag{26}$$

and

$$\begin{aligned}V_{i+1}(x_{k+1})&=\min_{u_{k+1}}\{U(x_{k+1},u_{k+1})+V_i(x_{k+2})\}\\&=U(x_{k+1},v_i(x_{k+1}))+V_i(x_{k+2}).\end{aligned} \tag{27}$$

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

4.2. Convergence analysis of the iterative ADP algorithm

Lemma 1 (*cf. Al-Tamimi et al. [30]*). *Let $\{\mu_i\}$ be an arbitrary sequence of control laws and $\{v_i\}$ be the sequence of control laws described in (26). Define V_i as in (27) and Λ_i as*

$$\Lambda_{i+1}(x_{k+1})=U(x_{k+1},\mu_i(x_{k+1}))+\Lambda_i(x_{k+2}). \tag{28}$$

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

Lemma 2. *Let the sequence of cost functions $\{V_i\}$ be defined as in (27). If the system is controllable, then there is an upper bound Y such that $0\le V_i(x_k)\le Y$, $\forall i$.*

Proof. Let $\eta(x_{k+1})$ be an admissible control input, and let $V_0(\cdot)=Z_0(\cdot)=0$, where V_i is updated as in (27) and Z_i is updated by

$$Z_{i+1}(x_{k+1})=U(x_{k+1},\eta(x_{k+1}))+Z_i(x_{k+2}). \tag{29}$$

Hence, the following equation is true

$$Z_{i+1}(x_{k+1})-Z_i(x_{k+1})=Z_i(x_{k+2})-Z_{i-1}(x_{k+2}). \tag{30}$$

When further expanding (30), we can get

$$Z_{i+1}(x_{k+1})-Z_i(x_{k+1})=Z_1(x_{k+i+1})-Z_0(x_{k+i+1}). \tag{31}$$

Thus, we have

$$Z_{i+1}(x_{k+1})=Z_1(x_{k+i+1})+Z_i(x_{k+1}). \tag{32}$$

By expanding (32), we can derive that

$$Z_{i+1}(x_{k+1})=\sum_{j=0}^{i}Z_1(x_{k+j+1})=\sum_{j=0}^{i}U(x_{k+j+1},\eta(x_{k+j+1})). \tag{33}$$

Since $\eta(x_{k+1})$ is an admissible control input, we have

$$Z_{i+1}(x_{k+1})\le\sum_{j=0}^{\infty}Z_1(x_{k+j+1})\le Y,\quad\forall i. \tag{34}$$

By using Lemma 1, we obtain

$$V_{i+1}(x_{k+1})\le Z_{i+1}(x_{k+1})\le Y,\quad\forall i. \tag{35}$$

and the proof is completed. □

Theorem 2. *Define the sequence $\{V_i\}$ as in (27) with $V_0(\cdot)=0$, and the sequence of control laws $\{v_i\}$ as in (26). Then, $\{V_i\}$ is a nondecreasing sequence satisfying $V_i\le V_{i+1}$, $\forall i$.*

Proof. Define a new sequence

$$\Phi_{i+1}(x_{k+1})=U(x_{k+1},v_{i+1}(x_{k+1}))+\Phi_i(x_{k+2}) \tag{36}$$

with $\Phi_0(\cdot)=V_0(\cdot)=0$. Next, we prove that $\Phi_i(x_{k+1})\le V_{i+1}(x_{k+1})$ by mathematical induction.

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

$$V_1(x_{k+1})-\Phi_0(x_{k+1})=U(x_{k+1},v_0(x_{k+1}))\ge 0, \tag{37}$$

we have

$$V_1(x_{k+1})\ge\Phi_0(x_{k+1}). \tag{38}$$

Second, we assume that it holds for $i-1$, i.e., $V_i(x_{k+1})\ge\Phi_{i-1}(x_{k+1})$, $\forall x_{k+1}$. Then, for i, by noticing

$$V_{i+1}(x_{k+1})=U(x_{k+1},v_i(x_{k+1}))+V_i(x_{k+2}) \tag{39}$$

and

$$\Phi_i(x_{k+1})=U(x_{k+1},v_i(x_{k+1}))+\Phi_{i-1}(x_{k+2}), \tag{40}$$

we can get

$$V_{i+1}(x_{k+1})-\Phi_i(x_{k+1})=V_i(x_{k+2})-\Phi_{i-1}(x_{k+2})\ge 0 \tag{41}$$

i.e.,

$$V_{i+1}(x_{k+1})\ge\Phi_i(x_{k+1}). \tag{42}$$

Furthermore, from Lemma 1, we know that $V_i(x_{k+1})\le\Phi_i(x_{k+1})$. Therefore, we have

$$V_{i+1}(x_{k+1})\ge\Phi_i(x_{k+1})\ge V_i(x_{k+1}) \tag{43}$$

and the proof is completed. □

Theorem 3. *Let the sequence of cost functions $\{V_i\}$ be defined as in (27) and $V_\infty(x_{k+1})$ as its limit. Then,*

$$V_\infty(x_{k+1})=\min_{u_{k+1}}\{U(x_{k+1},u_{k+1})+V_\infty(x_{k+2})\}. \tag{44}$$

Proof. On one hand, for any u_{k+1} and i, according to (27), we can derive

$$V_i(x_{k+1})\le U(x_{k+1},u_{k+1})+V_{i-1}(x_{k+2}). \tag{45}$$

Combining with

$$V_i(x_{k+1})\le V_\infty(x_{k+1}),\quad\forall i, \tag{46}$$

which can be derived from Theorem 2, we have

$$V_i(x_{k+1})\le U(x_{k+1},u_{k+1})+V_\infty(x_{k+2}),\quad\forall i. \tag{47}$$

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

$$V_\infty(x_{k+1})\le U(x_{k+1},u_{k+1})+V_\infty(x_{k+2}). \tag{48}$$

In (48), u_{k+1} is chosen arbitrarily. Therefore, it implies that

$$V_\infty(x_{k+1})\le\min_{u_{k+1}}\{U(x_{k+1},u_{k+1})+V_\infty(x_{k+2})\}. \tag{49}$$

On the other hand, since the cost function sequence satisfies

$$V_i(x_{k+1})=\min_{u_{k+1}}\{U(x_{k+1},u_{k+1})+V_{i-1}(x_{k+2})\} \tag{50}$$

for any i, considering (46), we have

$$V_\infty(x_{k+1})\ge\min_{u_{k+1}}\{U(x_{k+1},u_{k+1})+V_{i-1}(x_{k+2})\},\quad\forall i. \tag{51}$$

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

$$V_\infty(x_{k+1})\geq\min_{u_{k+1}}\{U(x_{k+1},u_{k+1})+V_\infty(x_{k+2})\}. \tag{52}$$

Based on (49) and (52), we can conclude that (44) is true. □

From the aforementioned results, we derive that the sequence of cost functions converges to the optimal cost function of the DTHJB equation, i.e., $V_i\to J^*$ as $i\to\infty$. Then, according to (6) and (26), we can conclude that the corresponding sequence of control laws also converges to the optimal one, i.e., $v_i\to u^*$ as $i\to\infty$.

5. Implementation of the iterative ADP algorithm using SN-DHP technique

In this section, we carry out the iterative ADP algorithm by using the SN-DHP technique. Its main difference from the traditional DHP technique lies in that only a critic network is utilized when building the actor-critic structure. In other words, we avoid the requirement of constructing the action network. Note that the model network is still needed to approximate the controlled plant.

5.1. Derivation of the SN-DHP technique

Now, we introduce the SN-DHP scheme. Define $\lambda(x_k)=\partial J(x_k)/\partial x_k$ and $\lambda^*(x_k)=\partial J^*(x_k)/\partial x_k$. Then,

$$\begin{aligned}\lambda^*(x_k)&=\frac{\partial U(x_k,u^*(x_k))}{\partial x_k}+\frac{\partial J^*(x_{k+1})}{\partial x_k}\\&=\frac{\partial U(x_k,u^*(x_k))}{\partial x_k}+\left(\frac{\partial u^*(x_k)}{\partial x_k}\right)^T\left[\frac{\partial U(x_k,u^*(x_k))}{\partial u^*(x_k)}\right.\\&\quad\left.+\left(\frac{\partial x_{k+1}}{\partial u^*(x_k)}\right)^T\frac{\partial J^*(x_{k+1})}{\partial x_{k+1}}\right]+\left(\frac{\partial x_{k+1}}{\partial x_k}\right)^T\frac{\partial J^*(x_{k+1})}{\partial x_{k+1}}\\&=2Qx_k+\left(\frac{\partial x_{k+1}}{\partial x_k}\right)^T\lambda^*(x_{k+1}).\end{aligned} \tag{53}$$

In the iteration process, we denote $\lambda_i(x_k)=\partial V_i(x_k)/\partial x_k$. Then, according to (27) and (53), we have

$$\lambda_{i+1}(x_{k+1})=2Qx_{k+1}+\left(\frac{\partial x_{k+2}}{\partial x_{k+1}}\right)^T\lambda_i(x_{k+2}). \tag{54}$$

Since the utility function is chosen as the quadratic form, we can solve the iterative control law from (26) as

$$v_i(x_{k+1})=-\tfrac{1}{2}R^{-1}\hat{g}^T(x_{k+1})\lambda_i(x_{k+2}). \tag{55}$$

Therefore, the iteration between (26) and (27) becomes (54) and (55).

Next, considering $x_{k+1}=f(x_k)+g(x_k)u(x_k)$, we denote $\bar{\lambda}_i(x_k)=\lambda_i(x_{k+1})$ and therefore, (54) becomes

$$\bar{\lambda}_{i+1}(x_k)=2Qx_{k+1}+\left(\frac{\partial x_{k+2}}{\partial x_{k+1}}\right)^T\bar{\lambda}_i(x_{k+1}). \tag{56}$$

Besides, the corresponding control input is

$$v_i(x_k)=-\tfrac{1}{2}R^{-1}\hat{g}^T(x_k)\bar{\lambda}_i(x_k). \tag{57}$$

In this sense, the iteration between (54) and (55) becomes (56) and (57). In addition, it is important to note that we can directly obtain the iterative control law $v_i(x_k)$ after deriving the $\bar{\lambda}_i(x_k)$.

5.2. Training the critic network

In the SN-DHP-based iterative ADP algorithm, both the model network and critic network are chosen as the three-layer feed forward NNs. The training of the model network is completed after the system identification process and its weights are kept unchanged. Then, according to Theorem 1, when given x_k and $v_i(x_k)$, we can compute x_{k+1} by using the model network. Hence, we avoid the requirement of knowing the system dynamics during the implementation of iterative ADP algorithm. The structure diagram of the iterative algorithm is shown in Fig. 1.

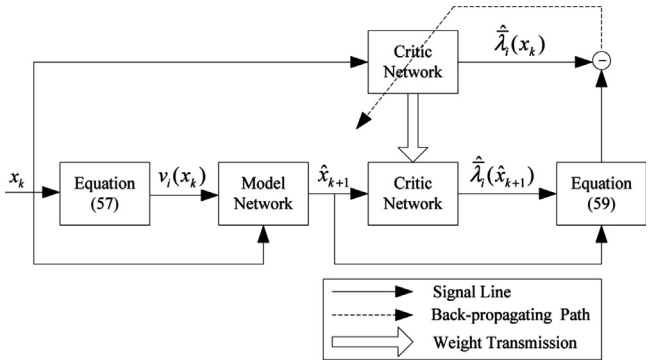


Fig. 1. The structure diagram of the SN-DHP-based iterative ADP algorithm.

The critic network is constructed to approximate $\bar{\lambda}_i(x_k)$ and its output can be formulated as

$$\hat{\bar{\lambda}}_i(x_k)=\omega_{ci}^T\sigma(\nu_{ci}^Tx_k). \tag{58}$$

The target function can be written as

$$\tilde{\bar{\lambda}}_{i+1}(x_k)=2Q\hat{x}_{k+1}+\left(\frac{\partial\hat{x}_{k+2}}{\partial\hat{x}_{k+1}}\right)^T\hat{\bar{\lambda}}_i(\hat{x}_{k+1}). \tag{59}$$

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

$$e_{cik}=\hat{\bar{\lambda}}_i(x_k)-\tilde{\bar{\lambda}}_{i+1}(x_k). \tag{60}$$

The objective function to be minimized in the critic network is

$$E_{cik}=\tfrac{1}{2}e_{cik}^Te_{cik}. \tag{61}$$

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

$$\omega_{ci}(j+1)=\omega_{ci}(j)-\alpha_c\left[\frac{\partial E_{cik}}{\partial\omega_{ci}(j)}\right] \tag{62}$$

$$\nu_{ci}(j+1)=\nu_{ci}(j)-\alpha_c\left[\frac{\partial E_{cik}}{\partial\nu_{ci}(j)}\right] \tag{63}$$

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

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

6. Simulation examples

In this section, two numerical examples are provided to demonstrate the effectiveness of the control scheme derived by the SN-DHP-based iterative ADP algorithm.

6.1. Example 1

Consider the following linear discrete-time system:

$$x_{k+1}=\begin{bmatrix}0&0.1&0\\0.3&-1&0\\0&0&0.5\end{bmatrix}x_k+\begin{bmatrix}0\\-1\\0.5\end{bmatrix}u_k \tag{64}$$

where $x_k=[x_{1k}\ x_{2k}\ x_{3k}]^T\in\mathbb{R}^3$, $u_k\in\mathbb{R}$. The open-loop poles are 0.0292, −1.0292 and 0.5000, so the system (64) is unstable. The utility function is chosen as $U(x_k,u_k)=x_k^TQx_k+u_k^TRu_k$, where $Q=2I$, $R=I$,

and I denotes the identity matrix with suitable dimensions. Note that this is a linear quadratic regulator (LQR) problem. By using the classical linear optimal control method, we can obtain the optimal feedback gain is $K=[0.1919\ -0.6465\ -0.1595]$.

Then, we implement the SN-DHP-based iterative ADP algorithm. We choose three-layer feedforward NN as model network and critic network with the structures 4–9–3 and 3–9–3, respectively. In the system identification process, the initial weights between the input layer and hidden layer, and the hidden layer and output layer are chosen randomly in $[-0.5, 0.5]$ and $[-0.1, 0.1]$, respectively. We apply the NN identification scheme for 100 steps under the learning rate $\alpha_m=0.05$ to make sure that the accuracy 10^{-8} is reached. After that, we finish the training of the model network and keep its weights unchanged. The initial weights of the critic network are set to be random in $[-0.1, 0.1]$. Then, we train the critic network for 20 training cycles with each cycle of 2000 steps. In the training process, the learning rate $\alpha_c=0.05$.

Next, for the given initial state $x_0=[0.5\ 1\ -1]^T$, we apply the control laws designed by the iterative ADP algorithm and by solving the LQR problem to system (64) for 20 time steps, respectively. The obtained state curves and control input are shown in Figs. 2–5. From the simulation results, we can see that the controller acquired by the iterative ADP algorithm can converge to the optimal controller derived by solving the LQR problem. These results also illustrate the validity of the proposed method.

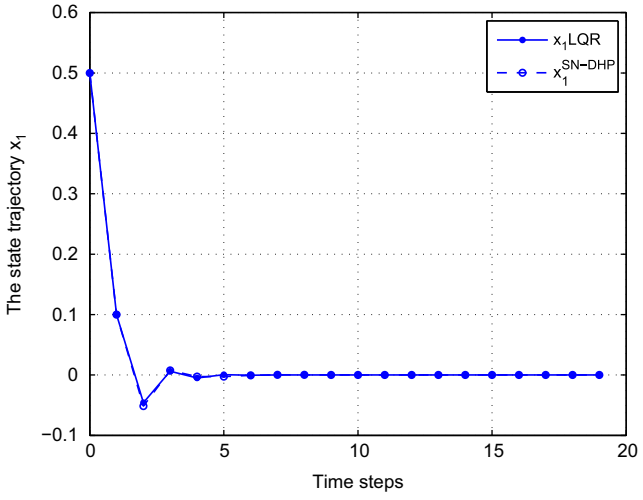


Fig. 2. The state trajectory x_1.

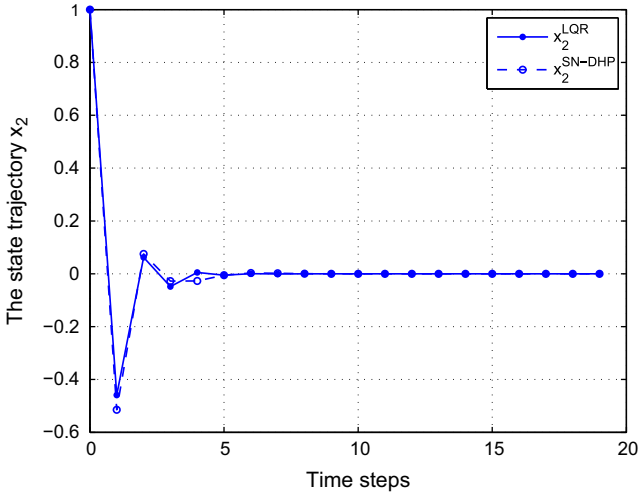


Fig. 3. The state trajectory x_2.

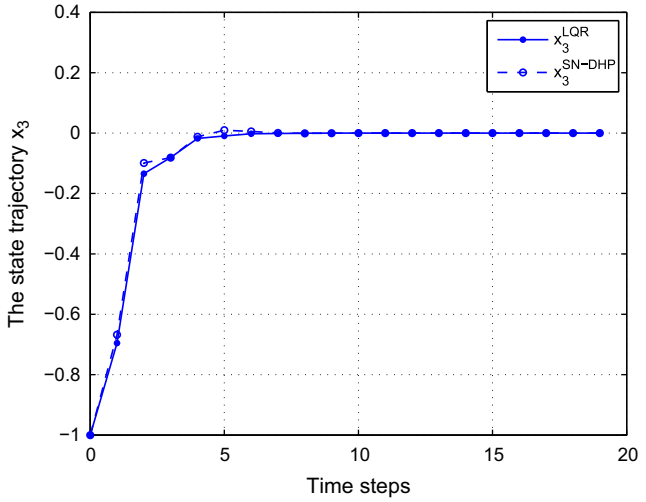


Fig. 4. The state trajectory x_3.

6.2. Example 2

Consider the nonlinear discrete-time system described by

$$x_{k+1}=\begin{bmatrix} x_{1k}x_{2k} \\ x_{1k}^2-\sin\ x_{2k}\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}u_k \tag{65}$$

where $x_k=[x_{1k}\ x_{2k}]^T\in\mathbb{R}^2$, $u_k\in\mathbb{R}$. The utility function is set the same as Example 1. It can be seen that $x_k=[0\ 0]^T$ is an equilibrium state of system (65). However, system (65) is marginally stable at this equilibrium, since the eigenvalues of

$$\left.\frac{\partial x_{k+1}}{\partial x_k}\right|_{(0,0)}=\begin{bmatrix}0 & 0\\ 0 & -1\end{bmatrix}$$

are 0 and −1.

When carrying out the SN-DHP-based iterative ADP algorithm, we choose three-layer feedforward NN as model network and critic network with the structures 3–8–2 and 2–8–2, respectively. First, we do the system identification process. The initial weights between the input layer and hidden layer, and the hidden layer and output layer are chosen randomly in $[-0.5, 0.5]$ and $[-0.1, 0.1]$, respectively. Here, we apply the NN identification scheme for 150 steps under the learning rate $\alpha_m=0.05$ to make sure that the accuracy 10^{-8} is reached. After finishing the training process, we keep the weights of model network unchanged. In addition, we set the initial weights of the critic network and its learning rate the same as Example 1. Then, we train the critic network for 36 training cycles with each cycle of 2000 steps, and then get the optimal control law.

In order to draw a comparison with the DHP [31] and GDHP [36] techniques, we also design the optimal controllers by using the DHP and GDHP based iterative ADP algorithms. Then, for the given initial state $x_0=[0.5\ 0.5]^T$, we apply the optimal control laws obtained by the three techniques to system (65) for 20 time steps, respectively. The derived state curves and the corresponding control input are shown in Figs. 6–8. It can be seen that the state and control trajectories generated by the SN-DHP, DHP, and GDHP techniques are very close to each other. This signifies that the SN-DHP technique is as good as the DHP and GDHP techniques in handling the nonlinear optimal control problem. Nevertheless, for the same problem, the SN-DHP-based iterative ADP algorithm takes about 182 s while the DHP and GDHP take greater than 290 s

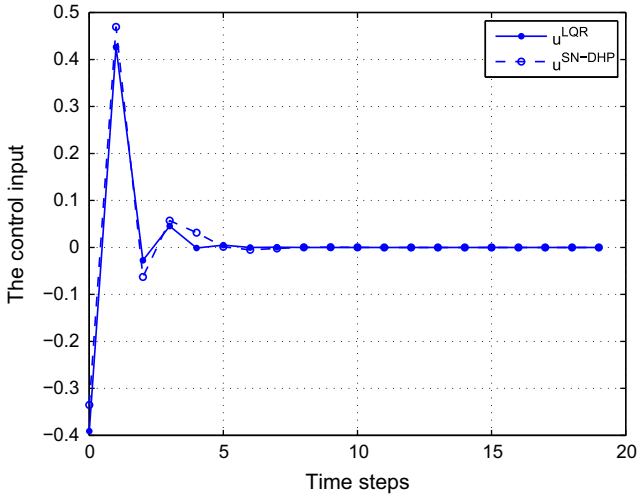


Fig. 5. The control input u.

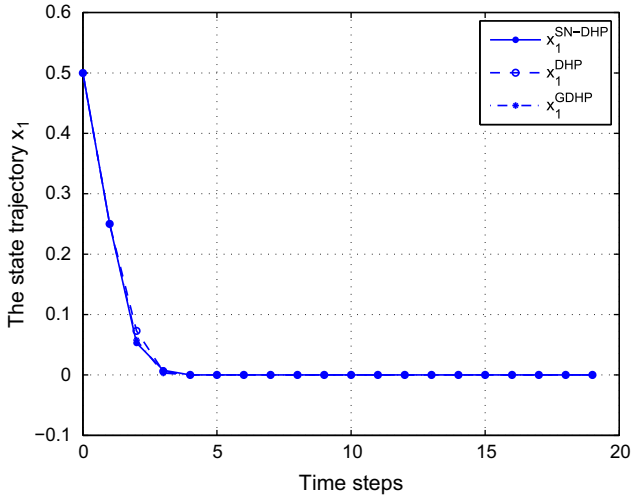


Fig. 6. The state trajectory x_1.

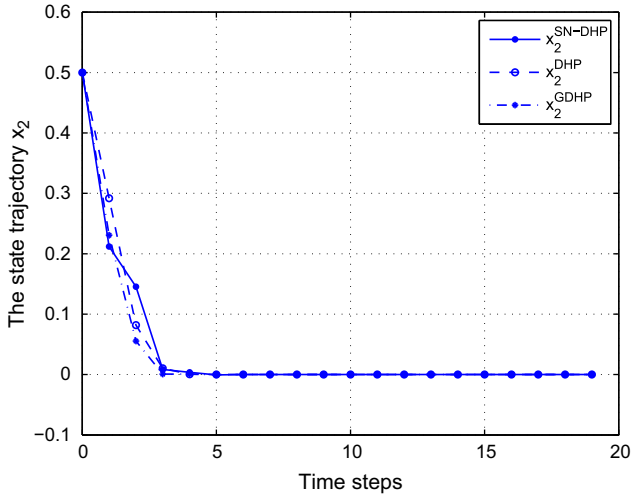


Fig. 7. The state trajectory x_2.

before satisfactory results are achieved. Therefore, the fact that the SN-DHP technique eliminates the use of action network has indeed reduced the computational complexity. The excellent control performance verifies the effectiveness of the SN-DHP technique.

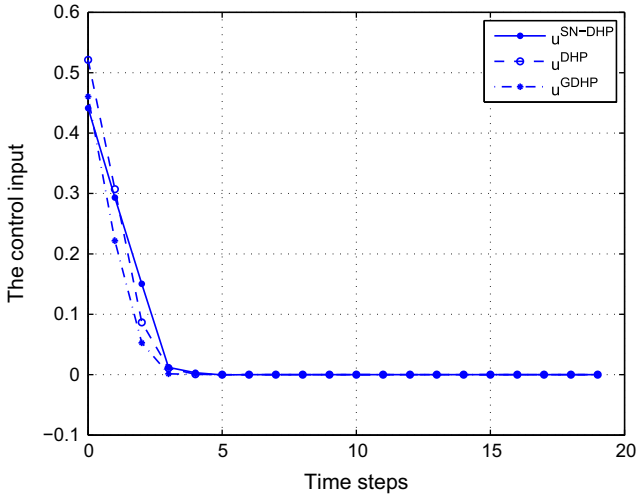


Fig. 8. The control input u.

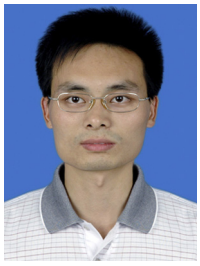
7. Conclusions

A novel SN-DHP-based technique is developed in this paper to find the near optimal controller for unknown affine nonlinear discrete-time systems with quadratic cost function. The iterative ADP algorithm is introduced to solve the cost function of the DTHJB equation with convergence analysis. Two NNs are used as parametric structures to approximate at each iteration the cost function and identify the unknown nonlinear system, respectively. Simulation studies demonstrated the validity of the control approach developed in this paper.

References

[1] O. Mohareri, R. Dhaouadi, A.B. Rad, Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks, Neurocomputing 88 (2012) 54–66.
[2] A. Boulkroune, M. M'Saad, M. Farza, Adaptive fuzzy tracking control for a class of MIMO nonaffine uncertain systems, Neurocomputing 93 (2012) 48–55.
[3] Y. Pan, Y. Zhou, T. Sun, M.J. Er, Composite adaptive fuzzy H_∞ tracking control of uncertain nonlinear systems, Neurocomputing 99 (2013) 15–24.
[4] Z. Xiong, J. Zhang, Modelling and optimal control of fed-batch processes using a novel control affine feedforward neural network, Neurocomputing 61 (2004) 317–337.
[5] R. Song, H. Zhang, Y. Luo, Q. Wei, Optimal control laws for time-delay systems with saturating actuators based on heuristic dynamic programming, Neurocomputing 73 (2010) 3020–3027.
[6] Y.J. Kim, M.T. Lim, Parallel optimal control for weakly coupled nonlinear systems using successive Galerkin approximation, IEEE Trans. Autom. Control 53 (2008) 1542–1547.
[7] 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. Sinica 36 (2010) 121–129.
[8] R.E. Bellman, Dynamic Programming, Princeton University Press, Princeton, NJ, 1957.
[9] T. Poggio, F. Girosi, Networks for approximation and learning, Proc. IEEE 78 (1990) 1481–1497.
[10] S. Jagannathan, Neural Network Control of Nonlinear Discrete-time Systems, CRC Press, Boca Raton, FL, 2006.
[11] P.J. Werbos, Advanced forecasting methods for global crisis warning and models of intelligence, Gen. Syst. Yearb. 22 (1977) 25–38.
[12] 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, Van Nostrand Reinhold, New York, 1992. (Chapter 13).
[13] P.J. Werbos, ADP: the key direction for future research in intelligent control and understanding brain intelligence, IEEE Trans. Syst. Man Cybern. 38 (2008) 898–900.
[14] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, Belmont, MA, 1996.

[15] D.V. Prokhorov, D.C. Wunsch, Adaptive critic designs, IEEE Trans. Neural Networks 8 (1997) 997–1007.
[16] J. Si, Y.T. Wang, On-line learning control by association and reinforcement, IEEE Trans. Neural Networks 12 (2001) 264–276.
[17] F.Y. Wang, H. Zhang, D. Liu, Adaptive dynamic programming: an introduction, IEEE Comput. Intell. Mag. 4 (2009) 39–47.
[18] F.Y. Wang, N. Jin, D. Liu, Q. Wei, Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε—error bound, IEEE Trans. Neural Networks 22 (2011) 24–36.
[19] J. Si, A.G. Barto, W.B. Powell, D.C. Wunsch (Eds.), Handbook of Learning and Approximate Dynamic Programming, IEEE Press/Wiley, New York, 2004.
[20] X. Zhang, H. Zhang, Q. Sun, Y. Luo, Adaptive dynamic programming-based optimal control of unknown nonaffine discrete-time systems with proof of convergence, Neurocomputing 91 (2012) 48–55.
[21] 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.
[22] C. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (1992) 279–292.
[23] R. Padhi, N. Unnikrishnan, X. Wang, S.N. Balakrishnan, A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems, Neural Networks 19 (2006) 1648–1660.
[24] G.G. Lendaris, A retrospective on adaptive dynamic programming for control, in: Proceedings of International Joint Conference on Neural Networks, June 2009, Atlanta, GA, 2009, pp. 1750–1757.
[25] D. Liu, Approximate dynamic programming for self-learning control, Acta Autom. Sinica 31 (2005) 13–18.
[26] D. Liu, Y. Zhang, H. Zhang, A self-learning call admission control scheme for CDMA cellular networks, IEEE Trans. Neural Networks 16 (2005) 1219–1228.
[27] D. Liu, N. Jin, ε—Adaptive dynamic programming for discrete-time systems, in: Proceedings of International Joint Conference on Neural Networks, June 2008, Hong Kong, 2008, pp. 1417–1424.
[28] Y. Luo, H. Zhang, N. Cao, B. Chen, Near-optimal stabilization for a class of nonlinear systems with control constraint based on single network greedy iterative DHP algorithm, Acta Autom. Sinica 35 (2009) 1436–1445.
[29] S.N. Balakrishnan, J. Ding, F.L. Lewis, Issues on stability of ADP feedback controllers for dynamic systems, IEEE Trans. Syst. Man Cybern. 38 (2008) 913–917.
[30] 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. 38 (2008) 943–949.
[31] H. Zhang, Y. Luo, D. Liu, Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints, IEEE Trans. Neural Networks 20 (2009) 1490–1503.
[32] 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.
[33] D. Vrabie, F.L. Lewis, Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems, Neural Networks 22 (2009) 237–246.
[34] T. Dierks, B.T. Thumati, S. Jagannathan, Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence, Neural Networks 22 (2009) 851–860.
[35] K.G. Vamvoudakis, F.L. Lewis, Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (2010) 878–888.
[36] 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.
[37] J.H. Kim, F.L. Lewis, Model-free H_∞ control design for unknown linear discrete-time systems via Q-learning with LMI, Automatica 46 (2010) 1320–1326.
[38] M.C. Campi, S.M. Savaresi, Direct nonlinear control design: the virtual reference feedback tuning approach, IEEE Trans. Autom. Control 51 (2006) 14–27.



Ding Wang received the B.S. degree in mathematics from Zhengzhou University of Light Industry, Zhengzhou, China, the M.S. degree in operations research and cybernetics from Northeastern University, Shenyang, China, and the Ph.D. degree in control theory and control engineering from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2007, 2009, and 2012, respectively. He is currently an assistant professor with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include adaptive dynamic programming, neural networks and learning systems, and intelligent control.

Derong Liu 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. Dr. Liu has been an Associate Editor of several IEEE publications. Currently, he is the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems, and an Associate Editor of the IEEE Transactions on Control Systems Technology. He was an elected AdCom member of the IEEE Computational Intelligence Society (2006–2008). He received 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.