Automatica 48 (2012) 1825–1832

Contents lists available at SciVerse ScienceDirect

Automatica

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

Brief paper

Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming☆

Ding Wang[a], Derong Liu[a,1], Qinglai Wei[a], Dongbin Zhao[a], Ning Jin[b]

[a] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China

[b] Department of Electrical and Computer Engineering, University of Illinois, Chicago, IL 60607, USA

ARTICLE INFO

Article history:
Received 31 October 2010
Received in revised form
22 January 2012
Accepted 8 February 2012
Available online 25 June 2012

Keywords:
Adaptive critic designs
Adaptive dynamic programming
Approximate dynamic programming
Globalized dual heuristic programming
Intelligent control
Neural network
Optimal control

ABSTRACT

An intelligent-optimal control scheme for unknown nonaffine nonlinear discrete-time systems with discount factor in the cost function is developed in this paper. The iterative adaptive dynamic programming algorithm is introduced to solve the optimal control problem with convergence analysis. Then, the implementation of the iterative algorithm via globalized dual heuristic programming technique is presented by using three neural networks, which will approximate at each iteration the cost function, the control law, and the unknown nonlinear system, respectively. In addition, two simulation examples are provided to verify the effectiveness of the developed optimal control approach.

1. Introduction

The main difference between optimal control of linear systems and nonlinear systems lies in that the latter often requires solving the nonlinear Hamilton–Jacobi–Bellman (HJB) equation instead of the Riccati equation (Abu-Khalaf & Lewis, 2005; Al-Tamimi, Lewis, & Abu-Khalaf, 2008; Primbs, Nevistic, & Doyle, 2000; Wang, Zhang, & Liu, 2009). For example, the discrete-time HJB (DTHJB) equation is more difficult to deal with than Riccati equation because it involves solving nonlinear partial difference equations. Although there were some methods that did not need to solve the HJB equation directly (e.g., Beard, Saridis, & Wen, 1997; Chen, Edgar, & Manousiouthakis, 2004), they were limited to handle some special classes of systems or they needed to perform very complex calculations. On the other hand, dynamic programming (DP) has been a useful technique in solving optimal control problems for many years (Bellman, 1957). However, it is often computationally untenable to run DP to obtain optimal solutions due to the "curse of dimensionality" (Bellman, 1957). Moreover, the backward direction of search precludes the application of DP in real-time control.

Artificial neural networks (ANN or NN) are an effective tool to implement intelligent control due to the properties of nonlinearity, adaptivity, self-learning, fault tolerance, and universal approximation of input–output mapping (Jagannathan, 2006; Werbos, 1992, 2008, 2009). Thus, it has been used for universal function approximation in adaptive/approximate dynamic programming (ADP) algorithms, which were proposed in Werbos (1992, 2008, 2009) as a method to solve optimal control problems forward-in-time. There are several synonyms used for ADP including "adaptive dynamic programming" (Lewis & Vrabie, 2009; Liu & Jin, 2008; Murray, Cox, Lendaris, & Saeks, 2002; Wang et al., 2009), "approximate dynamic programming" (Al-Tamimi et al., 2008; Werbos, 1992), "neuro-dynamic programming" (Bertsekas & Tsitsiklis, 1996), "neural dynamic programming" (Si & Wang, 2001), "adaptive critic designs" (Prokhorov & Wunsch, 1997), and "reinforcement learning" (Watkins & Dayan, 1992).

As an effective intelligent control method, in recent years, ADP and the related research have gained much attention from researchers (Balakrishnan & Biega, 1996; Balakrishnan, Ding, & Lewis, 2008; Dierks, Thumati, & Jagannathan, 2009; Jagannathan &

☆ This work was supported in part by the National Natural Science Foundation of China under Grants 60874043, 60904037, 60921061, and 61034002, and by Beijing Natural Science Foundation under Grant 4102061. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Gang Tao under the direction of Editor Miroslav Krstic.

E-mail addresses: ding.wang@ia.ac.cn (D. Wang), derong.liu@ia.ac.cn (D. Liu), qinglai.wei@ia.ac.cn (Q. Wei), dongbin.zhao@ia.ac.cn (D. Zhao), njin@uic.edu (N. Jin).

[1] Tel.: +86 10 62557379; fax: +86 10 62650912.

0005-1098/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2012.05.049

He, 2008; Vamvoudakis & Lewis, 2010; Venayagamoorthy, Harley, & Wunsch, 2002; Venayagamoorthy, Wunsch, & Harley, 2000; Vrabie & Lewis, 2009; Yen & Delima, 2005; Zhang, Luo, & Liu, 2009; Zhang, Wei, & Liu, 2011). According to Prokhorov and Wunsch (1997) and Werbos (1992), ADP approaches were classified into several main schemes: heuristic dynamic programming (HDP), action-dependent HDP (ADHDP; note the prefix "action-dependent" (AD) used hereafter), also known as Q-learning (Watkins & Dayan, 1992), dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP. Al-Tamimi et al. (2008) derived a significant result that applied the HDP iteration algorithm to solve the DTHJB equation of affine nonlinear discrete-time systems.

In this paper, we will tackle the optimal control problem for unknown nonlinear discrete-time systems using iterative ADP algorithm via GDHP technique (iterative GDHP algorithm for brief). Though great progress has been made for ADP in optimal control field, to the best of our knowledge, there is still no result to solve this problem by using the iterative GDHP algorithm. Additionally, the outputs of critic network of the GDHP technique contain not only the cost function but also its derivatives. This is different from HDP and DHP and is very important because the information associated with the cost function is as useful as the knowledge of its derivatives. Though the structure of the GDHP technique is somewhat complicated, it is expected to bring remarkable advantage when compared with simple ADP strategies. These motivate our research.

This paper is organized as follows. In Section 2, we present the formulation of the problem. In Section 3, we develop the optimal control scheme based on iterative ADP algorithm with convergence analysis, and then present the corresponding NN implementation of the iterative GDHP algorithm. In Section 4, two examples are given to demonstrate the effectiveness of the present control strategy. In Section 5, concluding remarks are given.

2. Problem statement

Here, we make the assumption that the state of the controlled system is available for measurement.

In this paper, we will study the nonlinear discrete-time systems described by

$$x_{k+1} = F(x_k, u_k), \quad k = 0, 1, 2, \dots, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state and $u_k = u(x_k) \in \mathbb{R}^m$ is the control vector. Let x_0 be the initial state. The system function $F(x_k, u_k)$ is continuous for $\forall x_k, u_k$ and $F(0, 0) = 0$. Hence, $x = 0$ is an equilibrium state of system (1) under the control $u = 0$.

Definition 1. A nonlinear dynamical system is said to be stabilizable on a compact set $\Omega \in \mathbb{R}^n$, if for all initial states $x_0 \in \Omega$, there exists a control sequence $u_0, u_1, \dots, u_i \in \mathbb{R}^m$, $i = 0, 1, \dots$, such that the state $x_k \to 0$ as $k \to \infty$.

It is desired to find the control law $u_k = u(x_k)$ which minimizes the infinite horizon cost function given by

$$J(x_k) = \sum_{p=k}^{\infty} \gamma^{p-k} U(x_p, u_p), \tag{2}$$

where U is the utility function, $U(0, 0) = 0$, $U(x_p, u_p) \ge 0$ for $\forall x_p, u_p$, and γ is the discount factor with $0 < \gamma \le 1$. In this paper, the utility function is chosen as the quadratic form $U(x_p, u_p) = x_p^T Q x_p + u_p^T R u_p$, where Q and R are positive definite matrices with suitable dimensions.

For optimal control problems, 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.

Definition 2. A control $u(x)$ is said 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.

Note that Eq. (2) can be written as

$$\begin{aligned} J(x_k) &= x_k^T Q x_k + u_k^T R u_k + \gamma \sum_{p=k+1}^{\infty} \gamma^{p-k-1} U(x_p, u_p) \\ &= x_k^T Q x_k + u_k^T R u_k + \gamma J(x_{k+1}). \end{aligned} \tag{3}$$

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

$$J^*(x_k) = \min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma J^*(x_{k+1}) \right\}. \tag{4}$$

Besides, the optimal control u^* can be expressed as

$$u^*(x_k) = \arg\min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma J^*(x_{k+1}) \right\}. \tag{5}$$

By substituting (5) into (4), the DTHJB equation becomes

$$J^*(x_k) = x_k^T Q x_k + u^{*T}(x_k) R u^*(x_k) + \gamma J^*(x_{k+1}). \tag{6}$$

It should be noticed that Definitions 1 and 2 are the same for linear systems. Moreover, when dealing with linear quadratic regulator problems, the DTHJB equation reduces to the Riccati equation which can be efficiently solved. For the general nonlinear case, however, it is considerably difficult to cope with the DTHJB equation directly. Therefore, we will develop an iterative ADP algorithm to solve it in the next section, based on Bellman's optimality principle and the greedy iteration approach.

3. Neuro-optimal control scheme based on iterative ADP algorithm via the GDHP technique

3.1. Derivation of the iterative algorithm

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

$$v_0(x_k) = \arg\min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_0(x_{k+1}) \right\}. \tag{7}$$

Then, we update the cost function as

$$V_1(x_k) = x_k^T Q x_k + v_0^T(x_k) R v_0(x_k). \tag{8}$$

Next, for $i = 1, 2, \dots$, the algorithm iterates between

$$v_i(x_k) = \arg\min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_i(x_{k+1}) \right\} \tag{9}$$

and

$$V_{i+1}(x_k) = x_k^T Q x_k + v_i^T(x_k) R v_i(x_k) + \gamma V_i(F(x_k, v_i(x_k))). \tag{10}$$

In the above recurrent iteration, 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 proof of the iteration between (9) and (10) 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 algorithm

The convergence analysis provided here is an extension of that given in Al-Tamimi et al. (2008).

Lemma 1. *Let $\{\mu_i\}$ be any arbitrary sequence of control laws and $\{\upsilon_i\}$ be the control laws as in (9). Define V_i as in (10) and define Λ_i as*

$$\Lambda_{i+1}(x_k) = x_k^T Q x_k + \mu_i^T(x_k) R \mu_i(x_k) + \gamma \Lambda_i(F(x_k, \mu_i(x_k))). \tag{11}$$

If $V_0(\cdot) = \Lambda_0(\cdot) = 0$, then $V_{i+1}(x) \le \Lambda_{i+1}(x)$, $\forall i$.

Proof. It can be derived by noticing that V_{i+1} is the result of minimizing the right-hand side of (10) 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 (10). If the system is controllable, there is an upper bound Y such that $0 \le V_i(x_k) \le Y$, $\forall i$.*

Proof. Let $\eta(x_k)$ be any admissible control input, and let $V_0(\cdot) = Z_0(\cdot) = 0$, where V_i is updated as in (10) and Z_i is updated by

$$Z_{i+1}(x_k) = x_k^T Q x_k + \eta^T(x_k) R \eta(x_k) + \gamma Z_i(x_{k+1}). \tag{12}$$

Noticing the difference

$$\begin{aligned} Z_{i+1}(x_k) - Z_i(x_k) &= \gamma(Z_i(x_{k+1}) - Z_{i-1}(x_{k+1})) \\ &= \gamma^2(Z_{i-1}(x_{k+2}) - Z_{i-2}(x_{k+2})) \\ &= \gamma^3(Z_{i-2}(x_{k+3}) - Z_{i-3}(x_{k+3})) \\ &\;\;\vdots \\ &= \gamma^i(Z_1(x_{k+i}) - Z_0(x_{k+i})) \\ &= \gamma^i Z_1(x_{k+i}), \end{aligned} \tag{13}$$

we can obtain

$$\begin{aligned} Z_{i+1}(x_k) &= \gamma^i Z_1(x_{k+i}) + Z_i(x_k) \\ &= \gamma^i Z_1(x_{k+i}) + \gamma^{i-1} Z_1(x_{k+i-1}) + Z_{i-1}(x_k) \\ &= \gamma^i Z_1(x_{k+i}) + \gamma^{i-1} Z_1(x_{k+i-1}) \\ &\quad + \gamma^{i-2} Z_1(x_{k+i-2}) + Z_{i-2}(x_k) \\ &= \gamma^i Z_1(x_{k+i}) + \gamma^{i-1} Z_1(x_{k+i-1}) \\ &\quad + \gamma^{i-2} Z_1(x_{k+i-2}) + \cdots + \gamma Z_1(x_{k+1}) + Z_1(x_k), \end{aligned} \tag{14}$$

and therefore,

$$\begin{aligned} Z_{i+1}(x_k) &= \sum_{j=0}^{i} \gamma^j Z_1(x_{k+j}) \\ &= \sum_{j=0}^{i} \gamma^j \left(x_{k+j}^T Q x_{k+j} + \eta^T(x_{k+j}) R \eta(x_{k+j}) \right) \\ &\le \sum_{j=0}^{\infty} \gamma^j \left(x_{k+j}^T Q x_{k+j} + \eta^T(x_{k+j}) R \eta(x_{k+j}) \right). \end{aligned} \tag{15}$$

Since $\eta(x_k)$ is an admissible control input, i.e., $x_k \to 0$ as $k \to \infty$, there exists a finite Y such that

$$Z_{i+1}(x_k) \le \sum_{j=0}^{\infty} \gamma^j Z_1(x_{k+j}) \le Y, \quad \forall i. \tag{16}$$

By using Lemma 1, we get

$$V_{i+1}(x_k) \le Z_{i+1}(x_k) \le Y, \quad \forall i, \tag{17}$$

and so the proof is completed. □

Based on Lemmas 1 and 2, we now present the convergence proof of the cost function sequence.

Theorem 1. *Define the sequence $\{V_i\}$ as in (10) with $V_0(\cdot) = 0$, and the control law sequence $\{\upsilon_i\}$ as in (9). Then, we can conclude that $\{V_i\}$ is a nondecreasing sequence satisfying $V_i \le V_{i+1}$, $\forall i$.*

Proof. Define a new sequence as

$$\Phi_{i+1}(x_k) = x_k^T Q x_k + \upsilon_{i+1}^T(x_k) R \upsilon_{i+1}(x_k) + \gamma \Phi_i(x_{k+1}) \tag{18}$$

with $\Phi_0(\cdot) = V_0(\cdot) = 0$. Now, we show that $\Phi_i(x_k) \le V_{i+1}(x_k)$.

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

$$V_1(x_k) - \Phi_0(x_k) = x_k^T Q x_k + \upsilon_0^T(x_k) R \upsilon_0(x_k) \ge 0, \tag{19}$$

we have

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

Second, we assume that it holds for $i-1$, i.e., $\Phi_{i-1}(x_k) \le V_i(x_k)$, $\forall x_k$. Then, for i, from (10) and (18), we get

$$V_{i+1}(x_k) - \Phi_i(x_k) = \gamma(V_i(x_{k+1}) - \Phi_{i-1}(x_{k+1})) \ge 0, \tag{21}$$

i.e.,

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

Thus, (22) is true for any i by mathematical induction.

Furthermore, according to Lemma 1, we know that $V_i(x_k) \le \Phi_i(x_k)$. Combining with (22), we have

$$V_i(x_k) \le \Phi_i(x_k) \le V_{i+1}(x_k), \tag{23}$$

which completes the proof. □

According to Lemma 2 and Theorem 1, we can obtain that $\{V_i\}$ is a monotonically nondecreasing sequence with an upper bound, and therefore, its limit exists. Here, we define it as $\lim_{i\to\infty} V_i(x_k) = V_\infty(x_k)$ and present the following theorem.

Theorem 2. *Let the cost function sequence $\{V_i\}$ be defined as in (10). Then, its limit satisfies*

$$V_\infty(x_k) = \min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_\infty(x_{k+1}) \right\}. \tag{24}$$

Proof. For any u_k and i, according to (10), we can derive

$$V_i(x_k) \le x_k^T Q x_k + u_k^T R u_k + \gamma V_{i-1}(x_{k+1}). \tag{25}$$

Combining with

$$V_i(x_k) \le V_\infty(x_k), \quad \forall i, \tag{26}$$

which is obtained from (23), we have

$$V_i(x_k) \le x_k^T Q x_k + u_k^T R u_k + \gamma V_\infty(x_{k+1}), \quad \forall i. \tag{27}$$

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

$$V_\infty(x_k) \le x_k^T Q x_k + u_k^T R u_k + \gamma V_\infty(x_{k+1}). \tag{28}$$

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

$$V_\infty(x_k) \le \min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_\infty(x_{k+1}) \right\}. \tag{29}$$

On the other hand, since the cost function sequence satisfies

$$V_i(x_k) = \min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_{i-1}(x_{k+1}) \right\} \tag{30}$$

for any i, considering (26), we have

$$V_\infty(x_k) \ge \min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_{i-1}(x_{k+1}) \right\}, \quad \forall i. \tag{31}$$

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

$$V_\infty(x_k) \ge \min_{u_k} \left\{ x_k^T Q x_k + u_k^T R u_k + \gamma V_\infty(x_{k+1}) \right\}. \tag{32}$$

Based on (29) and (32), we can conclude that (24) is true. □

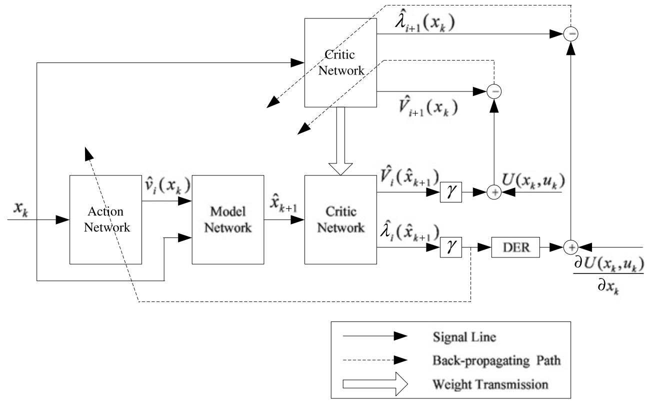


Fig. 1. The structure diagram of the iterative GDHP algorithm.

Remark 1. Let $\lim_{i\to\infty} v_i(x_k) = v_\infty(x_k)$. According to Theorem 2 and the relationship between (9) and (10), we have

$$V_\infty(x_k) = \min_{u_k}\left\{x_k^T Q x_k + u_k^T R u_k + \gamma V_\infty(x_{k+1})\right\}$$
$$= x_k^T Q x_k + v_\infty^T(x_k) R v_\infty(x_k) + \gamma V_\infty(F(x_k, v_\infty(x_k))), \quad (33)$$

where

$$v_\infty(x_k) = \arg\min_{u_k}\left\{x_k^T Q x_k + u_k^T R u_k + \gamma V_\infty(x_{k+1})\right\}. \quad (34)$$

Observing (33) and (34), and then (4) and (5), we can find that $V_\infty(x_k) = J^*(x_k)$ and $v_\infty(x_k) = u^*(x_k)$. In other words, $\lim_{i\to\infty} V_i(x_k) = J^*(x_k)$ and $\lim_{i\to\infty} v_i(x_k) = u^*(x_k)$.

3.3. NN implementation of the iterative algorithm

For carrying out the iterative ADP algorithm, we need to use a function approximation structure, such as NN, to approximate both $v_i(x_k)$ and $V_i(x_k)$.

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

$$\hat{F}(X, \nu, \omega) = \omega^T \sigma\left(\nu^T X\right), \quad (35)$$

where $\sigma\left(\nu^T X\right) \in \mathbb{R}^l$, $[\sigma(z)]_q = (e^{z_q} - e^{-z_q})/(e^{z_q} + e^{-z_q})$, $q = 1, 2, \ldots, l$, are the activation functions.

Now, we implement the iterative ADP algorithm via the GDHP technique. It consists of a model network, two critic networks and an action network, which are all chosen as three-layer feedforward NNs. The whole structure diagram is shown in Fig. 1, where

$$\mathrm{DER} = \left(\frac{\partial \hat{x}_{k+1}}{\partial x_k} + \frac{\partial \hat{x}_{k+1}}{\partial \hat{v}_i(x_k)} \frac{\partial \hat{v}_i(x_k)}{\partial x_k}\right)^T. \quad (36)$$

In order to avoid the requirement of knowing $F(x_k, u_k)$, we should train the model network before carrying out the main iterative process. For given x_k and $\hat{v}_i(x_k)$, we can obtain the output of the model network as

$$\hat{x}_{k+1} = \omega_m^T \sigma\left(\nu_m^T \left[x_k^T \ \hat{v}_i^T(x_k)\right]^T\right). \quad (37)$$

We define the error function of the model network as

$$e_{mk} = \hat{x}_{k+1} - x_{k+1}. \quad (38)$$

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

$$E_{mk} = \frac{1}{2} e_{mk}^T e_{mk}. \quad (39)$$

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

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

$$\nu_m(j+1) = \nu_m(j) - \alpha_m \left[\frac{\partial E_{mk}}{\partial \nu_m(j)}\right], \quad (41)$$

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

The weights of the model network are kept unchanged after the training process is finished.

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 input of critic network is x_k, while the output is given by

$$\begin{bmatrix}\hat{V}_i(x_k)\\ \hat{\lambda}_i(x_k)\end{bmatrix} = \begin{bmatrix}\omega_{ci}^{1T}\\ \omega_{ci}^{2T}\end{bmatrix} \sigma\left(\nu_{ci}^T x_k\right) = \omega_{ci}^T \sigma\left(\nu_{ci}^T x_k\right), \quad (42)$$

where $\omega_{ci} = \left[\omega_{ci}^1 \ \omega_{ci}^2\right]$. Note that the same weight matrix between input layer and hidden layer is used to approximate the cost function and its derivative. This framework can reduce the computational burden when compared to the case of constructing two separate critics. Hence, we have

$$\hat{V}_i(x_k) = \omega_{ci}^{1T} \sigma\left(\nu_{ci}^T x_k\right) \quad (43)$$

and

$$\hat{\lambda}_i(x_k) = \omega_{ci}^{2T} \sigma\left(\nu_{ci}^T x_k\right). \quad (44)$$

The target functions can be written as

$$V_{i+1}(x_k) = x_k^T Q x_k + v_i^T(x_k) R v_i(x_k) + \gamma \hat{V}_i(\hat{x}_{k+1}) \quad (45)$$

and

$$\lambda_{i+1}(x_k) = \frac{\partial\left(x_k^T Q x_k + v_i^T(x_k) R v_i(x_k)\right)}{\partial x_k} + \gamma \frac{\partial \hat{V}_i(\hat{x}_{k+1})}{\partial x_k}$$
$$= 2Qx_k + 2\left(\frac{\partial v_i(x_k)}{\partial x_k}\right)^T R v_i(x_k)$$
$$+ \gamma\left(\frac{\partial \hat{x}_{k+1}}{\partial x_k} + \frac{\partial \hat{x}_{k+1}}{\partial \hat{v}_i(x_k)} \frac{\partial \hat{v}_i(x_k)}{\partial x_k}\right)^T \hat{\lambda}_i(\hat{x}_{k+1}). \quad (46)$$

Note that Eq. (46) is simply the derivative form of (45), and therefore, the two are equivalent in principle. Then, the error functions can be defined as

$$e_{cik}^1 = \hat{V}_i(x_k) - V_{i+1}(x_k) \quad (47)$$

and

$$e_{cik}^2 = \hat{\lambda}_i(x_k) - \lambda_{i+1}(x_k). \quad (48)$$

Since the GDHP technique is a combination of HDP and DHP techniques, we choose the objective function to be minimized by the critic network as

$$E_{cik} = (1-\theta) E_{cik}^1 + \theta E_{cik}^2, \quad (49)$$

where

$$E_{cik}^1 = \frac{1}{2} e_{cik}^{1T} e_{cik}^1 \quad (50)$$

and

$$E_{cik}^2 = \frac{1}{2} e_{cik}^{2T} e_{cik}^2. \quad (51)$$

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

$$\omega_{ci}(j+1) = \omega_{ci}(j) - \alpha_c \left[(1-\theta) \frac{\partial E^1_{cik}}{\partial \omega_{ci}(j)} + \theta \frac{\partial E^2_{cik}}{\partial \omega_{ci}(j)} \right], \tag{52}$$

$$\nu_{ci}(j+1) = \nu_{ci}(j) - \alpha_c \left[(1-\theta) \frac{\partial E^1_{cik}}{\partial \nu_{ci}(j)} + \theta \frac{\partial E^2_{cik}}{\partial \nu_{ci}(j)} \right], \tag{53}$$

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 $\theta = 1$ does the same for DHP.

In the action network, x_k is used as the input and the output is

$$\hat{v}_i(x_k) = \omega^T_{ai} \sigma \left(\nu^T_{ai} x_k \right). \tag{54}$$

The target control input is given by

$$v_i(x_k) = \arg\min_{u_k} \left\{ x^T_k Q x_k + u^T_k R u_k + \gamma \hat{V}_i(\hat{x}_{k+1}) \right\}. \tag{55}$$

The error function of the action network can be defined as

$$e_{aik} = \hat{v}_i(x_k) - v_i(x_k). \tag{56}$$

The weights of the action network are updated to minimize

$$E_{aik} = \frac{1}{2} e^T_{aik} e_{aik}. \tag{57}$$

Similarly, the weight update algorithm is

$$\omega_{ai}(j+1) = \omega_{ai}(j) - \alpha_a \left[\frac{\partial E_{aik}}{\partial \omega_{ai}(j)} \right], \tag{58}$$

$$\nu_{ai}(j+1) = \nu_{ai}(j) - \alpha_a \left[\frac{\partial E_{aik}}{\partial \nu_{ai}(j)} \right], \tag{59}$$

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 2. According to Remark 1, $V_i \to J^*$ 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 $i \to \infty$.

Remark 3. Since we cannot implement the iteration until $i \to \infty$ in practical applications, we should run the algorithm with a prespecified accuracy ε to test the convergence of the cost function sequence. When $|V_{i+1}(x_k) - V_i(x_k)| < \varepsilon$, we consider the cost function sequence has converged sufficiently and stop running the iterative GDHP algorithm.

4. Simulation studies

In this section, two examples are provided to demonstrate the effectiveness of the iterative GDHP algorithm.

4.1. Example 1

Consider the following nonlinear system:

$$x_{k+1} = x_k + \sin(x_k + u_k), \tag{60}$$

where $x_k \in \mathbb{R}$, $u_k \in \mathbb{R}$, $k = 1, 2, \ldots$. The utility function is chosen as $U(x_k, u_k) = x^T_k x_k + u^T_k u_k$. It can be seen that $x_k = 0$ is an equilibrium state of system (60). However, the system is unstable at this equilibrium, since $(\partial x_{k+1}/\partial x_k)|_{(0,0)} = 2 > 1$.

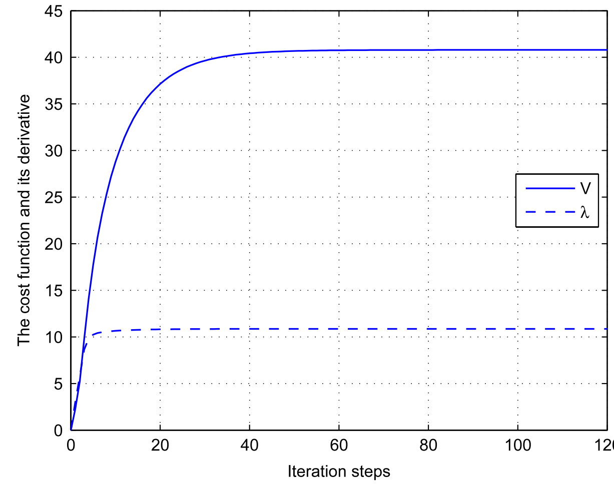


Fig. 2. The convergence processes of the cost function and its derivative of the iterative GDHP algorithm.

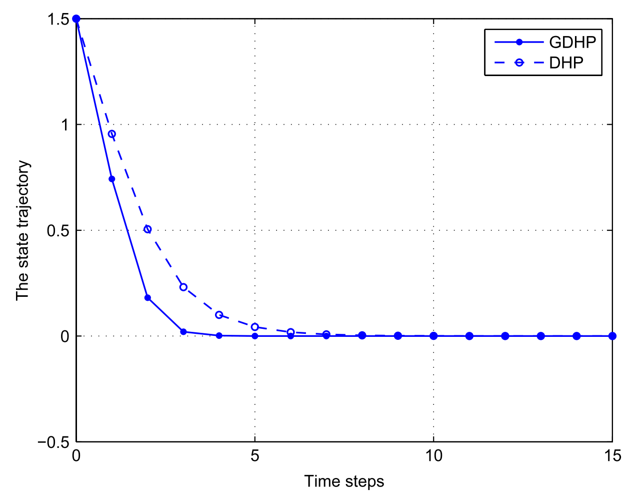


Fig. 3. The state trajectory x.

We choose three-layer feedforward NNs as model network, critic network and action network with structures 2–8–1, 1–8–2, and 1–8–1, respectively, and implement the algorithm at time instant $k = 0$. The initial weights of the three NNs are all set to be random in $[-1, 1]$. Note that the model network should be trained first. We train the model network for 100 time steps using 500 data samples under the learning rate $\alpha_m = 0.1$. After the model network is trained, its weights are kept unchanged. Then, let the discount factor $\gamma = 1$ and the adjusting parameter $\theta = 0.5$, we train the critic network and action network for 120 iterations (i.e., for $i = 1, 2, \ldots, 120$) with 2000 training epochs for each iteration to make sure the given accuracy $\varepsilon = 10^{-6}$ is reached. In the training process, the learning rate $\alpha_c = \alpha_a = 0.05$. The convergence processes of the cost function and its derivative of GDHP algorithm are shown in Fig. 2, for $k = 0$ and $x_0 = 1.5$. We can see that the iterative cost function sequence does converge to the optimal value quite rapidly, which also indicates the validity of the iterative GDHP algorithm. For the same problem, the iterative GDHP algorithm takes about 16 s while HDP takes about 117 s before satisfactory results are obtained.

Moreover, in order to make comparison with DHP algorithm, we also present the controller designed by DHP algorithm. Then, for given initial state $x_0 = 1.5$, we apply the optimal control laws designed by GDHP and DHP techniques to the system for 15 time steps, respectively, and obtain the state curves as shown in Fig. 3. The corresponding control curves are shown in Fig. 4. It can be seen

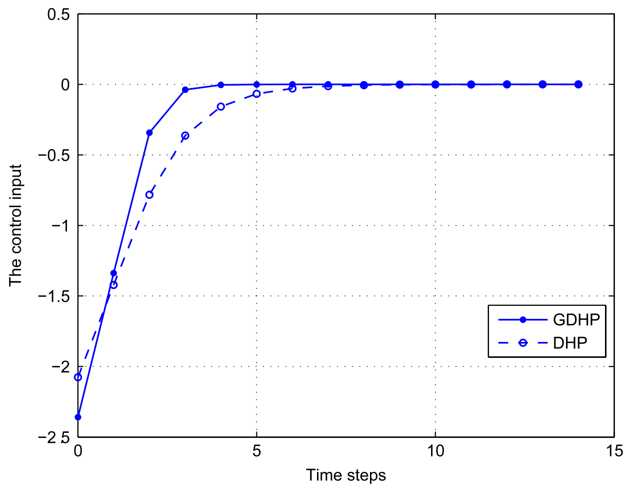


Fig. 4. The control input u.

from the simulation results that the controller derived by the GDHP algorithm has a better performance than the DHP algorithm.

To show the discount factor has evident impact on our iterative algorithm, in this case, we choose the discount factor $\gamma = 0.9$ and set the other parameters the same as above. Then, we train the critic network and action network for 80 iterations and find that the given accuracy $\varepsilon = 10^{-6}$ has been reached, which demonstrates that smaller discount factor can insure quicker convergence of the cost function sequence. Next, we will show the discrepancy of the state and control curves under different iterations to prove the usefulness of the iterative algorithm. For the same initial state $x_0 = 1.5$, we apply different control laws to the controlled plant for 15 time steps and obtain simulation results as follows. The state curves are shown in Fig. 5, and the corresponding control inputs are shown in Fig. 6. From the simulation results, we can see that the closed-loop system is divergent when using the control law obtained in the first iteration. However, the system responses become better and better as the iteration numbers increasing from 3 to 80. Besides, the responses basically remain unchanged when the iteration number is larger than 5, which verifies the effectiveness of the proposed iterative GDHP algorithm.

4.2. Example 2

Consider the nonlinear discrete-time system given by

$$x_{k+1} = \begin{bmatrix} -x_{1k}x_{2k} \\ 1.5x_{2k} + \sin(x_{2k}^2 + u_k) \end{bmatrix} \tag{61}$$

where $x_k = [x_{1k}\ x_{2k}]^T \in \mathbb{R}^2$, $u_k \in \mathbb{R}$, $k = 1, 2, \ldots$. The utility function is also set as $U(x_k, u_k) = x_k^T x_k + u_k^T u_k$.

In this example, we also choose three-layer feedforward NNs as the model network, the critic network and the action network, but with structures 3–8–2, 2–8–3, and 2–8–1, respectively. Here, we train the critic network and action network for 50 iterations while keeping the other parameters the same as the above example. The convergence processes of the cost function and its derivative of the iterative GDHP algorithm are shown in Fig. 7, which verify the theoretical conjectures of Theorems 1–2 and Remarks 1–2. Furthermore, for given initial state $x_{10} = 0.5$ and $x_{20} = -1$, we apply the optimal control law designed by the iterative GDHP algorithm to (61) for 25 time steps, and obtain the state curves and the corresponding control curves as shown in Figs. 8 and 9, respectively. These simulation results verify the excellent performance of the controller derived by the iterative GDHP algorithm.

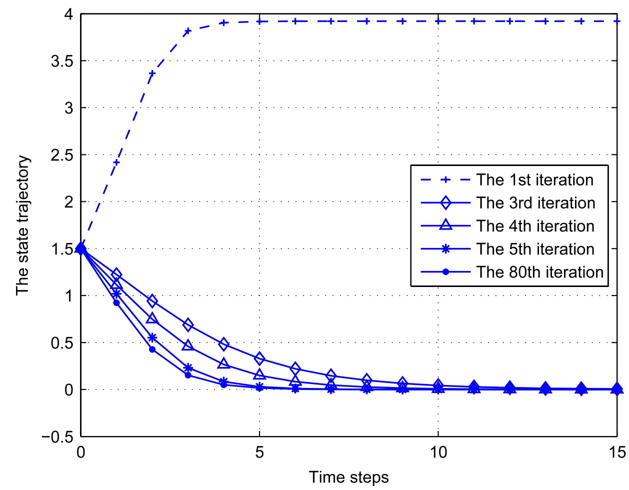


Fig. 5. The state trajectory x.

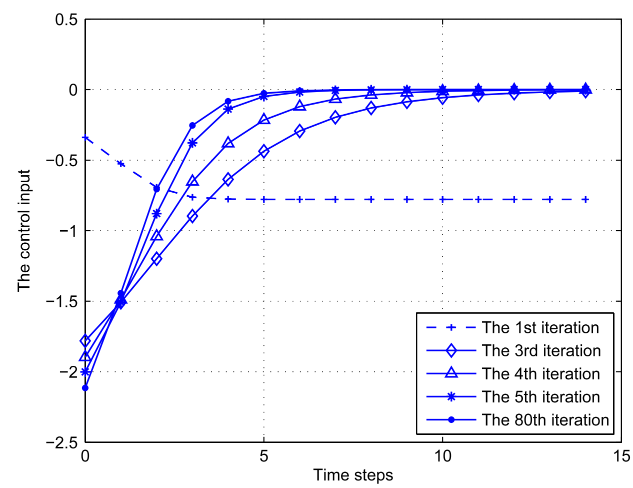


Fig. 6. The control input u.

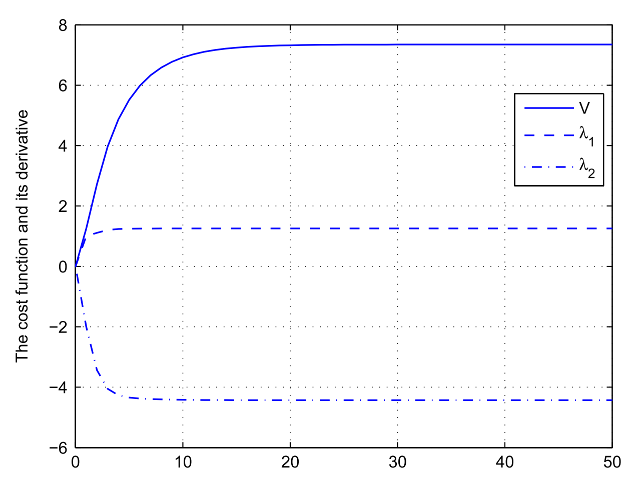


Fig. 7. The convergence processes of the cost function and its derivative of the iterative GDHP algorithm.

5. Conclusion

In this paper, an effective iterative ADP algorithm with convergence analysis is given to design the near optimal controller for unknown nonaffine nonlinear discrete-time systems with discount factor in the cost function. The GDHP technique is introduced to implement the algorithm. Three NNs are used as

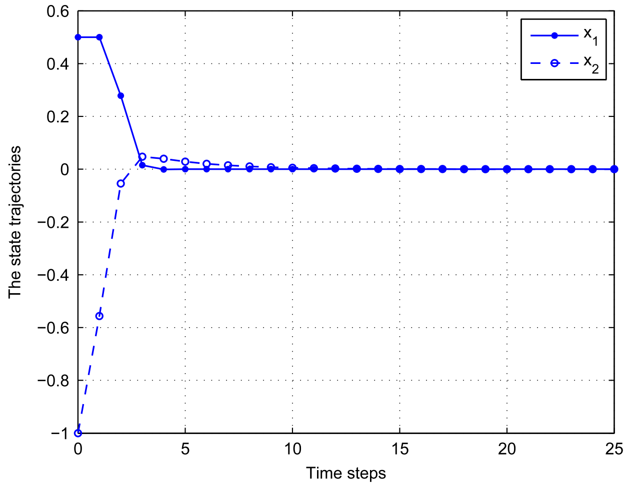


Fig. 8. The state trajectories x_1 and x_2.

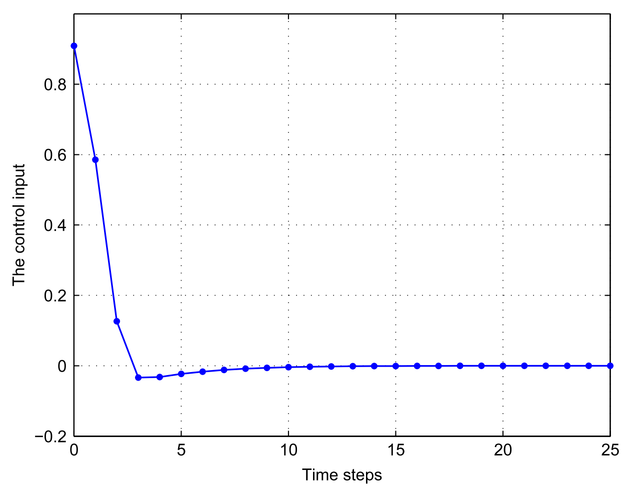


Fig. 9. The control input u.

parametric structures to approximate the cost function and its derivative, the control law and identify the unknown nonlinear system, respectively. The simulation studies demonstrated the validity of the proposed optimal control scheme.

References

Abu-Khalaf, M., & Lewis, F. L. (2005). Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica*, *41*(5), 779–791.

Al-Tamimi, A., Lewis, F. L., & Abu-Khalaf, M. (2008). Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, *38*(4), 943–949.

Balakrishnan, S. N., & Biega, V. (1996). Adaptive-critic based neural networks for aircraft optimal control. *Journal of Guidance, Control, and Dynamics*, *19*(4), 893–898.

Balakrishnan, S. N., Ding, J., & Lewis, F. L. (2008). Issues on stability of ADP feedback controllers for dynamic systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *38*(4), 913–917.

Beard, R. W., Saridis, G. N., & Wen, J. T. (1997). Galerkin approximation of the generalized Hamilton–Jacobi–Bellman equation. *Automatica*, *33*(12), 2159–2177.

Bellman, R. E. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Belmont, MA: Athena Scientific.

Chen, Y., Edgar, T., & Manousiouthakis, V. (2004). On infinite-time nonlinear quadratic optimal control. *Systems & Control Letters*, *51*(3–4), 259–268.

Dierks, T., Thumati, B. T., & Jagannathan, S. (2009). Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence. *Neural Networks*, *22*(5–6), 851–860.

Jagannathan, S. (2006). *Neural network control of nonlinear discrete-time systems*. Boca Raton, FL: CRC Press.

Jagannathan, S., & He, P. (2008). Neural-network-based state feedback control of a nonlinear discrete-time system in nonstrict feedback form. *IEEE Transactions on Neural Networks*, *19*(12), 2073–2087.

Lewis, F. L., & Vrabie, D. (2009). Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine*, *9*(3), 32–50.

Liu, D., & Jin, N. (2008). ε-adaptive dynamic programming for discrete-time systems. In *Proceedings of the international joint conference on neural networks* (pp. 1417–1424).

Murray, J. J., Cox, C. J., Lendaris, G. G., & Saeks, R. (2002). Adaptive dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*, *32*(2), 140–153.

Primbs, J. A., Nevistic, V., & Doyle, J. C. (2000). A receding horizon generalization of pointwise min-norm controllers. *IEEE Transactions on Automatic Control*, *45*(5), 898–909.

Prokhorov, D. V., & Wunsch, D. C. (1997). Adaptive critic designs. *IEEE Transactions on Neural Networks*, *8*(5), 997–1007.

Si, J., & Wang, Y. T. (2001). On-line learning control by association and reinforcement. *IEEE Transactions on Neural Networks*, *12*(2), 264–276.

Vamvoudakis, K. G., & Lewis, F. L. (2010). Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, *46*(5), 878–888.

Venayagamoorthy, G. K., Harley, R. G., & Wunsch, D. C. (2002). Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator. *IEEE Transactions on Neural Networks*, *13*(3), 764–773.

Venayagamoorthy, G. K., Wunsch, D. C., & Harley, R. G. (2000). Adaptive critic based neurocontroller for turbogenerators with global dual heuristic programming. In *Proceedings of the IEEE PES winter meeting* (pp. 291–294).

Vrabie, D., & Lewis, F. L. (2009). Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. *Neural Networks*, *22*(3), 237–246.

Wang, F. Y., Zhang, H., & Liu, D. (2009). Adaptive dynamic programming: an introduction. *IEEE Computational Intelligence Magazine*, *4*(2), 39–47.

Watkins, C., & Dayan, P. (1992). *Q*-learning. *Machine Learning*, *8*(3–4), 279–292.

Werbos, P. J. (1992). Approximate dynamic programming for real-time control and neural modeling. In D. A. White, & D. A. Sofge (Eds.), *Handbook of intelligent control: neural, fuzzy, and adaptive approaches*. New York: Van Nostrand Reinhold.

Werbos, P. J. (2008). ADP: the key direction for future research in intelligent control and understanding brain intelligence. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, *38*(4), 898–900.

Werbos, P. J. (2009). Intelligence in the brain: a theory of how it works and how to build it. *Neural Networks*, *22*(3), 200–212.

Yen, G. G., & Delima, P. G. (2005). Improving the performance of globalized dual heuristic programming for fault tolerant control through an online learning supervisor. *IEEE Transactions on Automation Science and Engineering*, *2*(2), 121–131.

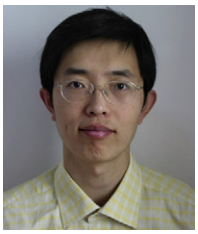
Zhang, H., Luo, Y., & Liu, D. (2009). Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. *IEEE Transactions on Neural Networks*, *20*(9), 1490–1503.

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.

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

Derong Liu received his Ph.D. degree in electrical engineering from the University of Notre Dame in 1994. Liu was a Staff Fellow with General Motors Research and Development Center, Warren, MI, from 1993 to 1995. He was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, from 1995 to 1999. He joined the University of Illinois at Chicago in 1999, 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 10 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 and several other journals. 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.

Qinglai Wei received his B.S. degree in automation, his M.S. degree in control theory and control engineering, and his Ph.D. degree in control theory and control engineering, from the Northeastern University, Shenyang, China, in 2002, 2005, and 2008, respectively. From 2009 to 2010, he was a post-doctoral fellow with the Institute of Automation, Chinese Academy of Sciences. 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 neural-networks-based control, nonlinear control, adaptive dynamic programming, and their industrial applications.



Dongbin Zhao received his B.S., M.S., and Ph.D. degrees in material processing engineering from the Harbin Institute of Technology, Harbin, China, in 1994, 1996, and 2000, respectively. Dr. Zhao was a post-doctoral fellow with Tsinghua University, Beijing, China, from May 2000 to January 2002. He is currently an Associate Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing. He has published one book and over 30 international journal papers. His current research interests include the area of computational intelligence, adaptive dynamic programming, robotics, intelligent transportation systems, and process simulation. Dr. Zhao has been a Senior Member of the Chinese Mechanical Engineering Society since 2006. He served as many international program committee member and the Finance Chair of the International Symposium on Neural Networks in 2011. He was Guest Editor of several international journals. He received the Second Award for Scientific Progress of National Defense from the Commission of Science Technology and Industry for National Defense of China in 1999, the First Award for Scientific Progress of Chinese Universities, Ministry of Education of China in 2001, the Third Award for Scientific and Technology Progress from the China Petroleum and Chemical Industry Association in 2009, and the First Award for Scientific and Technology Progress from the China Petroleum and Chemical Industry Association in 2010.

Ning Jin received his Ph.D. degree in electrical and computer engineering from the University of Illinois at Chicago in 2012. He was an Associate Professor in the Department of Mathematics at Nanjing Normal University. From 2002 to 2005, he was a visiting scholar in the Department of Mathematics, Statistics, and Computer Science at the University of Illinois at Chicago. He is currently an Instructor in the Department of Electrical and Computer Engineering at the University of Illinois at Chicago. His research interests include optimal control and dynamic programming, artificial intelligence, pattern recognition, neural networks, and wavelet analysis.