as far as the image processing time is over the capturing period. In general, current 30 fps cameras are good enough for the visual alignment applications. If a high-performance image processing system with the maximum computing power is available, the update rate for the centroid measurements in the Kalman filter can be increased as much. Then, the look-and-move period in Fig. 4(b) will get faster and the coarse-to-fine alignment would be more promising. Another consideration can be taken into the design of joint servos for the alignment stage to follow the error compensation trajectory. Although we have applied PID control for each active joint, a model-based control design to reflect the friction characteristics of the moving axes would be more effective in the precision control.

References

[1] A. K. Kanjilal, "Automatic mask alignment without a microscope," *IEEE Trans. Instrumen. Meas.*, vol. 44, no. 3, pp. 806–809, Jun. 1995.

[2] H. T. Kim, C. S. Song, and H. J. Yang, "2-step algorithm of automatic alignment in wafer dicing process," *Microelectron. Reliab.*, vol. 44, pp. 1165–1179, 2004.

[3] S. J. Kwon and J. Hwang, "Kinematics, pattern recognition, and motion control of mask-panel alignment system," *Control Eng. Practice*, vol. 19, pp. 883–892, 2011.

[4] C. S. Park and S. J. Kwon, "An efficient vision algorithm for fast and fine mask-panel alignment," in *Proc. SICE-ICASE Int. Joint Conf.*, Oct. 2006, pp. 1461–1465.

[5] S. J. Kwon and H. Jeong, "Observer based fine motion control of autonomous visual alignment systems," in *Proc. 2009 IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, Singapore, Jul. 14–17, 2009, pp. 1822–1827.

[6] L. Alvarez, J. Yi, R. Horowitz, and L. Olmos, "Dynamic friction model-based tire-road friction estimation and emergency braking control," *ASME J. Dynamic Syst., Meas., Control*, vol. 127, pp. 22–32, Mar. 2005.

[7] D. Chwa and J. Y. Choi, "Observer-based control for tail-controlled skid-to-turn missiles using a parametric affine model," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 1, pp. 167–175, Jan. 2004.

[8] A. Swarnakar, H. J. Marquez, and T. Chen, "A new scheme on robust observer-based control design for interconnected systems with application to an industrial utility boiler," *IEEE Trans. Control Syst.s Technol.*, vol. 16, no. 3, pp. 539–548, May 2008.

[9] A. T. Elfizy, G. M. Bone, and M. A. Elbestawi, "Design and control of a dual-stage feed drive," *Int. J. Mach. Tools Manuf.*, vol. 45, pp. 153–165, 2005.

[10] J. B. Morrell and J. K. Salisbury, "Parallel-coupled micro-macro actuators," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 773–791, 1998.

[11] T. Semba, T. Hirano, J. Hong, and L. Fan, "Dual-stage servo controller for HDD using MEMS microactuator," *IEEE Trans. Magn.*, vol. 35, no. 5, 1999.

[12] J. Wang, H. Zha, and R. Cipolla, "Coarse-to-fine vision-based localization by indexing scale-Invariant features," *IEEE Trans. Syst., Man, Cybern.-Part B: Cybern.*, vol. 36, no. 2, pp. 413–422, Apr. 2006.

[13] L. Ren, L. Wang, J. K. Mills, and D. Sun, "Vision-based 2-D automatic micrograsping using coarse-to-fine grasping strategy," *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, pp. 3324–3331, Sep. 2008.

[14] S. J. Ralis, B. Vikramaditya, and B. J. Nelson, "Micropositioning of a weakly calibrated microassembly system using coarse-to-fine visual servoing strategies," *IEEE Trans. Electron. Packag. Manuf.*, vol. 23, no. 2, pp. 123–131, Apr. 2000.

[15] M.-S. Choi and W.-K. Kim, "A novel two stage template matching method for rotation and illumination invariance," *Pattern Recognit.*, vol. 35, pp. 119–129, 2002.

[16] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using Matlab*. New York: Wiley, 2001.

[17] R. Qian, M. Sezan, and K. Matthews, "A robust real-time face tracking algorithm," in *Proc. IEEE Int Conf. Image Process. (ICIP 98)*, Oct. 1998, pp. 131–135.

[18] D. S. Jang, S. W. Jang, and H. I. Choi, "2D human body tracking with structural Kalman filter," *Pattern Recognit.*, vol. 35, pp. 2041–2049, 2002.

[19] M. Gharavi-Alkhansari, "A fast globally optimal algorithm for template matching using low-resolution pruning," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 526–533, Apr. 2001.

Neural-Network-Based Optimal Control for a Class of Unknown Discrete-Time Nonlinear Systems Using Globalized Dual Heuristic Programming

Derong Liu*, *Fellow, IEEE*, Ding Wang,
Dongbin Zhao, *Senior Member, IEEE*, Qinglai Wei, *Member, IEEE*,
and Ning Jin, *Student Member, IEEE*

***Abstract*—In this paper, a neuro-optimal control scheme for a class of unknown discrete-time nonlinear systems with discount factor in the cost function is developed. The iterative adaptive dynamic programming algorithm using globalized dual heuristic programming technique is introduced to obtain the optimal controller with convergence analysis in terms of cost function and control law. In order to carry out the iterative algorithm, a neural network is constructed first to identify the unknown controlled system. Then, based on the learned system model, two other neural networks are employed as parametric structures to facilitate the implementation of the iterative algorithm, which aims at approximating at each iteration the cost function and its derivatives and the control law, respectively. Finally, a simulation example is provided to verify the effectiveness of the proposed optimal control approach.**

***Note to Practitioners*—The increasing complexity of the real-world industry processes inevitably leads to the occurrence of nonlinearity and high dimensions, and their mathematical models are often difficult to build. How to design the optimal controller for nonlinear systems without the requirement of knowing the explicit model has become one of the main foci of control practitioners. However, this problem cannot be handled by only relying on the traditional dynamic programming technique because of the "curse of dimensionality". To make things worse, the backward direction of solving process of dynamic programming precludes its wide application in practice. Therefore, in this paper, the iterative adaptive dynamic programming algorithm is proposed to deal with the optimal control problem for a class of unknown nonlinear systems forward-in-time. Moreover, the detailed implementation of the iterative ADP algorithm through the globalized dual heuristic programming technique is also presented by using neural networks. Finally, the effectiveness of the control strategy is illustrated via simulation study.**

***Index Terms*—Adaptive dynamic programming, approximate dynamic programming, globalized dual heuristic programming, intelligent control, neural networks, optimal control.**

I. Introduction

As is known, nonlinear optimal control is a difficult and challenging area since it often requires solving the Hamilton–Jacobi–Bellman (HJB) equation instead of the Riccati equation. For ex-

Manuscript received January 09, 2012; accepted April 22, 2012. Date of publication May 22, 2012; date of current version June 28, 2012. This paper was recommended for publication by Associate Editor H. Tanner and Editor K. Lynch upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 60904037, Grant 60921061, and Grant 61034002, in part by the Beijing Natural Science Foundation under Grant 4102061, and in part by the China Postdoctoral Science Foundation under Grant 201104162. *Asterisk indicates corresponding author.*

*D. Liu is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: derong.liu@ia.ac.cn).

D. Wang, D. Zhao, and Q. Wei are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: ding.wang@ia.ac.cn; dongbin.zhao@ia.ac.cn; qinglai.wei@ia.ac.cn).

N. Jin is with the Department of Electrical and Computer Engineering, University of Illinois, Chicago, IL 60607 USA (e-mail: njin@uic.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2012.2198057

ample, the discrete-time HJB (DTHJB) equation is more difficult to solve than the Riccati equation because it involves dealing with nonlinear partial difference equations.

Though dynamic programming has been a useful technique in solving optimal control problems for many years, it is often computationally untenable to run it to obtain optimal solution, due to the "curse of dimensionality" [1]. Thus, based on dynamic programming and neural networks (NNs), adaptive/approximate dynamic programming (ADP) was proposed in [2]–[4] as a method to solve optimal control problems forward-in-time. There are several synonyms used for ADP, including "adaptive dynamic programming" [5]–[8], "approximate dynamic programming" [9], [10], "neuro-dynamic programming" [11], "neural dynamic programming" [12], "adaptive critic designs" [13], and "reinforcement learning" [14]–[16].

In recent years, ADP and related research have gained much attention from researchers [2]–[28]. In [2], Werbos defined "intelligence" as the general-purpose ability of brain to learn to maximize some kind of "utility function" over time, in a complex, unknown, and nonlinear environment. ADP is the only general-purpose scheme to learn to approximate optimal strategy of action in the general case. Therefore, it can be considered as one of the key methods to be able to design truly brain-like general-purpose intelligent systems. According to [4] and [13], ADP approaches were classified into several main schemes: heuristic dynamic programming (HDP), action-dependent HDP (ADHDP), also known as Q-learning, dual heuristic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP. Al-Tamimi *et al.* [10] derived a greedy HDP iteration algorithm to solve the DTHJB equation. Fu *et al.* [7] investigated the adaptive learning and control for multiple-input-multiple-output system based on ADP.

Since the mathematical models of most real-world plants are often difficult to build, how to design the optimal controller for nonlinear systems with unknown dynamics has become one of the main foci of control practitioners. However, there are still no results to solve the optimal control problems for unknown discrete-time nonlinear systems with discount factor in the cost function based on iterative ADP algorithm using GDHP technique (iterative GDHP algorithm for brief). In this paper, for the first time, we will solve these problems via the iterative GDHP algorithm.

The main contributions of this paper can be summarized as follows. (1) By introducing identification section, we generalize the iterative ADP algorithm to nonlinear optimal control problems with discount factor and unknown system dynamics. (2) We show more clearly that the limit of the cost function sequence equals to its optimal value. (3) When implementing the iterative algorithm, we make use of the GDHP technique in order to output the cost function and its derivative simultaneously and obtain more satisfactory results.

II. Problem Statement

In this paper, we study the discrete-time nonlinear systems 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, and $f(\cdot)$ 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. In the following part, $u(x_k)$ is denoted by u_k for simplicity.

Let x_0 be an initial state and define $\underline{u}_0^{N-1} = (u_0, u_1, \ldots, u_{N-1})$ be a control sequence with which the system (1) gives a trajectory starting from x_0: $x_1 = f(x_0) + g(x_0)u_0$, $x_2 = f(x_1) + g(x_1)u_1, \ldots, x_N = f(x_{N-1}) + g(x_{N-1})u_{N-1}$. We call the number of elements in the control sequence $\underline{u}_0^{N-1}$ the length of $\underline{u}_0^{N-1}$ and denote it as $\left|\underline{u}_0^{N-1}\right|$. Then, $\left|\underline{u}_0^{N-1}\right| = N$. The final state under control sequence $\underline{u}_0^{N-1}$ can be denoted as $x^{(f)}\left(x_0, \underline{u}_0^{N-1}\right) = x_N$. When the control sequence starting from u_0 has infinite length, we denote it as $\underline{u}_0^{\infty} = (u_0, u_1, \ldots)$. Then, the corresponding final state can be written as $x^{(f)}(x_0, \underline{u}_0^{\infty}) = \lim_{k\to\infty} x_k$.

Let $\underline{u}_k^{\infty} = (u_k, u_{k+1}, \ldots)$ be the control sequence starting at k. It is desired to find the control sequence $\underline{u}_k^{\infty}$ which minimizes the infinite horizon cost function given by

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

where U is the utility function, $U(0,0) = 0$, $U(x_i, u_i) \geq 0$ for $\forall x_i, u_i$, and γ is the discount factor with $0 < \gamma \leq 1$. The discount factor mirrors the fact that we are less concerned about costs acquired further into the future. Generally speaking, the utility function can be chosen as the quadratic form $U(x_i, u_i) = x_i^T Q x_i + u_i^T R u_i$.

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 1 (cf. [10]): A control sequence $\underline{u}_k^{\infty}$ is said to be admissible for a state $x_k \in \mathbb{R}^n$ with respect to (2) on Ω, if u is continuous on a compact set $\Omega_u \in \mathbb{R}^m$, $u(0) = 0$, $x^{(f)}(x_k, \underline{u}_k^{\infty}) = 0$ and $J(x_k, \underline{u}_k^{\infty})$ is finite.

Let $\mathsf{A}_{x_k} = \left\{\underline{u}_k^{\infty} : x^{(f)}(x_k, \underline{u}_k^{\infty}) = 0\right\}$ be the set of all infinite horizon admissible control sequences of x_k. Define the optimal cost function as

$$J^*(x_k) = \inf_{\underline{u}_k^{\infty}} \left\{ J(x_k, \underline{u}_k^{\infty}) : \underline{u}_k^{\infty} \in \mathsf{A}_{x_k} \right\}. \tag{3}$$

Note that (2) can be written as

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

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

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

The optimal control u^* is given by

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

In (5) and (6), $J^*(x_k)$ is the optimal cost function corresponding to the optimal control $u^*(x_k)$. When dealing with the linear quadratic regulator problems, the DTHJB equation reduces to the Riccati equation which can be solved efficiently. For general nonlinear problems, however, it is not the case.

III. Neuro-Optimal Control Scheme Based on the Iterative ADP Algorithm

A. NN Identification of the Unknown Nonlinear System

For the NN identifier, a three-layer NN is employed as the function approximation structure in this paper. Let the number of hidden layer neurons be denoted by l, the ideal weight matrix between input layer and hidden layer be denoted by ν_m^*, and the ideal weight matrix between hidden layer and output layer be denoted by ω_m^*. According to the universal approximation property of NN, the system dynamics (1) has an NN representation on a compact set S, which can be written as

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

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

During the system identification process, we keep the weight matrix between input layer and hidden layer constant while only tune the weight matrix between hidden layer and output layer. Hence, we define the NN identification scheme as

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

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

Denote $\tilde{x}_k = \hat{x}_k - x_k$ as the system identification error. Then, by combining (7) with (8), we can obtain the identification error dynamics as

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

where $\tilde{\omega}_m(k) = \omega_m(k) - \omega_m^*$. Let $\psi_k = \tilde{\omega}_m^T(k)\sigma(\bar{z}_k)$. Then, (9) can be rewritten as

$$\tilde{x}_{k+1} = \psi_k - \theta_k. \tag{10}$$

The weights in the system identification process are updated to minimize the performance measure $E_{k+1} = \tilde{x}_{k+1}^T \tilde{x}_{k+1}/2$. Using the gradient-based adaptation rule, the weights can be updated by

$$\begin{aligned}\omega_m(k+1) &= \omega_m(k) - \alpha_m \left[\frac{\partial E_{k+1}}{\partial \omega_m(k)}\right] \\ &= \omega_m(k) - \alpha_m \sigma(\bar{z}_k)\tilde{x}_{k+1}^T\end{aligned} \tag{11}$$

where $\alpha_m > 0$ is the NN learning rate.

Assumption 1: The NN approximation error term θ_k is assumed to be upper bounded by a function of the state estimation error $\tilde{x}_k$ such that $\theta_k^T \theta_k \le \theta_{Mk} = \delta \tilde{x}_k^T \tilde{x}_k$, where δ is the constant target value with δ_M as its upper bound, i.e., $\|\delta\| \le \delta_M$.

Theorem 1: Let the identification scheme (8) be used to identify the nonlinear system (1), and let the parameter update law (11) be used to tune the NN weights. Then, the state estimation error dynamics $\tilde{x}_k$ is asymptotically stable while the parameter estimation error $\tilde{\omega}_m(k)$ is bounded.

Proof: We consider the positive definite Lyapunov function candidate $L_k = \tilde{x}_k^T \tilde{x}_k + \mathrm{tr}\{\tilde{\omega}_m^T(k)\tilde{\omega}_m(k)\}/\alpha_m$. Taking its first difference, we can obtain

$$\begin{aligned}\Delta L_k \le &-\left(1 - 2\alpha_m \sigma_M^2\right)\|\psi_k\|^2 \\ &-\left(1 - \delta_M - 2\alpha_m \delta_M \sigma_M^2\right)\|\tilde{x}_k\|^2.\end{aligned} \tag{12}$$

Define $\alpha_m \le \rho^2/(2\sigma_M^2)$, then (12) becomes

$$\begin{aligned}\Delta L_k \le &-\left(1-\rho^2\right)\|\psi_k\|^2 - \left(1 - \delta_M - \delta_M\rho^2\right)\|\tilde{x}_k\|^2 \\ = &-\left(1-\rho^2\right)\left\|\tilde{\omega}_m^T(k)\sigma(\bar{z}_k)\right\|^2 \\ &-\left(1 - \delta_M - \delta_M\rho^2\right)\|\tilde{x}_k\|^2.\end{aligned} \tag{13}$$

From (13), we can conclude that $\Delta L_k \le 0$ if $0 < \delta_M < 1$ and

$$\max\left\{-\sqrt{\frac{1-\delta_M}{\delta_M}}, -1\right\} \le \rho \le \min\left\{\sqrt{\frac{1-\delta_M}{\delta_M}}, 1\right\}$$

where $\rho \ne 0$. As long as the parameters are selected as above, $\Delta L_k \le 0$ which shows stability in the sense of Lyapunov. Therefore, $\tilde{x}_k$ and $\tilde{\omega}_m(k)$ are bounded, provided $\tilde{x}_0$ and $\tilde{\omega}_m(0)$ are bounded in the compact set S. Furthermore, by summing both sides of (13) to infinity and taking account of $\Delta L_k \le 0$, we have

$$\left|\sum_{k=0}^{\infty} \Delta L_k\right| = \left|\lim_{k\to\infty} L_k - L_0\right| < \infty.$$

This implies that

$$\sum_{k=0}^{\infty}\left\{\left(1-\rho^2\right)\left\|\tilde{\omega}_m^T(k)\sigma(\bar{z}_k)\right\|^2 + \left(1-\delta_M-\delta_M\rho^2\right)\|\tilde{x}_k\|^2\right\} < \infty.$$

Hence, it can be concluded that the estimation error approaches zero, i.e., $\|\tilde{x}_k\| \to 0$ as $k \to \infty$. ■

Remark 1: According to Theorem 1, after a sufficient learning session, the NN identification error converges to zero, i.e., we have

$$f(x_k) + \hat{g}(x_k)u_k = \omega_m^T(k)\sigma(\bar{z}_k) \tag{14}$$

where $\hat{g}(x_k)$ denotes the estimated value of the control coefficient matrix $g(x_k)$. Taking the partial derivative of both sides of (14) with respect to u_k yields

$$\begin{aligned}\hat{g}(x_k) &= \frac{\partial\left(\omega_m^T(k)\sigma(\bar{z}_k)\right)}{\partial u_k} \\ &= \omega_m^T(k)\frac{\partial \sigma(\bar{z}_k)}{\partial \bar{z}_k}\nu_m^{*T}\begin{bmatrix} 0_{n\times m} \\ ----- \\ I_m \end{bmatrix}\end{aligned} \tag{15}$$

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

B. Derivation of the Iterative ADP Algorithm

In this section, we present the iterative ADP algorithm. First, we start with the initial cost function $V_0(\cdot) = 0$ and solve the initial control $v_0(x_k)$ as

$$v_0(x_k) = \arg\min_{u_k}\left\{x_k^T Q x_k + u_k^T R u_k + \gamma V_0(x_{k+1})\right\}. \tag{16}$$

Then, we update the cost function as

$$\begin{aligned} V_1(x_k) &= \min_{u_k}\left\{x_k^T Q x_k + u_k^T R u_k + \gamma V_0(x_{k+1})\right\} \\ &= x_k^T Q x_k + v_0^T(x_k) R v_0(x_k). \end{aligned} \tag{17}$$

Next, for $i = 1, 2, \ldots$, the iterative ADP algorithm is implemented between the control law

$$\begin{aligned} v_i(x_k) &= \arg\min_{u_k}\left\{x_k^T Q x_k + u_k^T R u_k + \gamma V_i(x_{k+1})\right\} \\ &= -\frac{\gamma}{2} R^{-1} \hat{g}^T(x_k) \frac{\partial V_i(x_{k+1})}{\partial x_{k+1}} \end{aligned} \tag{18}$$

and the cost function

$$\begin{aligned} V_{i+1}(x_k) &= \min_{u_k}\left\{x_k^T Q x_k + u_k^T R u_k + \gamma V_i(x_{k+1})\right\} \\ &= x_k^T Q x_k + v_i^T(x_k) R v_i(x_k) + \gamma V_i(x_{k+1}). \end{aligned} \tag{19}$$

In the above recurrent process, i is the iteration index of control law and cost function, while k is the time index of system's control and state trajectories. The cost function and control law are updated until they converge to the optimal ones. Next, we will provide the convergence proof of the iterative ADP algorithm to show that the cost function $V_i \to J^*$ and the control law $v_i \to u^*$ as $i \to \infty$.

C. Convergence Analysis of the Iterative ADP Algorithm

Before presenting our main theorem, we first give the following two lemmas.

Lemma 1 (cf. [23]): Let $\{v_i\}$ be the control law sequence described in (18) and $\{\mu_i\}$ be an arbitrary sequence of control laws. Define V_i as in (19) and Λ_i as

$$\Lambda_{i+1}(x_k) = x_k^T Q x_k + \mu_i^T(x_k) R \mu_i(x_k) + \gamma \Lambda_i(x_{k+1}). \tag{20}$$

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

Lemma 2 (cf. [23]): Let the cost function sequence $\{V_i\}$ be defined as in (19). If the system is controllable, then there is an upper bound Y such that $0 \le V_i(x_k) \le Y, \forall i$.

Then, we conclude that the cost function sequence $\{V_i\}$ is a monotonically nondecreasing one with an upper bound, and therefore, its limit exists. Define $\lim_{i\to\infty} V_i(x_k) = V_\infty(x_k)$. Hence, the following equation holds:

$$V_\infty(x_k) = \min_{u_k}\left\{x_k^T Q x_k + u_k^T R u_k + \gamma V_\infty(x_{k+1})\right\}. \tag{21}$$

These results can be obtained according to [23], only noting that the discount factor should be considered.

Next, we will prove that the cost function sequence converges to its optimal value.

Theorem 2: Define the cost function sequence $\{V_i\}$ as in (19) with $V_0(\cdot) = 0$. If the system state x_k is controllable, then J^* is the limit of the cost function sequence $\{V_i\}$, i.e., $\lim_{i\to\infty} V_i(x_k) = J^*(x_k)$.

Proof: Let $\left\{\eta_i^{(l)}\right\}$ be the lth admissible control law sequence. We construct the associated sequence $\left\{P_i^{(l)}(x_k)\right\}$ as follows:

$$P_{i+1}^{(l)}(x_k) = x_k^T Q x_k + \left(\eta_i^{(l)}(x_k)\right)^T R \eta_i^{(l)}(x_k) + \gamma P_i^{(l)}(x_{k+1}) \tag{22}$$

with $P_0^{(l)}(\cdot) = 0$. By expanding (22), we can obtain

$$P_{i+1}^{(l)}(x_k) = \sum_{j=0}^{i} \gamma^j U\left(x_{k+j}, \eta_{i-j}^{(l)}(x_{k+j})\right). \tag{23}$$

Using Lemmas 1 and 2, we have

$$V_{i+1}(x_k) \le P_{i+1}^{(l)}(x_k) \le Y_l, \ \forall l, i \tag{24}$$

where Y_l is the upper bound associated with the sequence $\left\{P_i^{(l)}(x_k)\right\}$. Denote $\lim_{i\to\infty} P_i^{(l)}(x_k) = P_\infty^{(l)}(x_k)$. Then, we can obtain

$$V_\infty(x_k) \le P_\infty^{(l)}(x_k) \le Y_l, \ \forall l. \tag{25}$$

Let the corresponding control sequence associated with (23) be

$$\begin{aligned} {}^{(l)}\underline{\hat{u}}_k^{k+i} &= \left({}^{(l)}\hat{u}_k, {}^{(l)}\hat{u}_{k+1}, \ldots, {}^{(l)}\hat{u}_{k+i}\right) \\ &= \left(\eta_i^{(l)}(x_k), \eta_{i-1}^{(l)}(x_{k+1}), \ldots, \eta_0^{(l)}(x_{k+i})\right). \end{aligned}$$

Then, we have

$$J\left(x_k, {}^{(l)}\underline{\hat{u}}_k^{k+i}\right) = \sum_{j=0}^{i} \gamma^j U\left(x_{k+j}, \eta_{i-j}^{(l)}(x_{k+j})\right) = P_{i+1}^{(l)}(x_k). \tag{26}$$

Letting $i \to \infty$, and denoting the admissible control sequence related to $P_\infty^{(l)}(x_k)$ with length ∞ as ${}^{(l)}\underline{\hat{u}}_k^\infty$, we get

$$J\left(x_k, {}^{(l)}\underline{\hat{u}}_k^\infty\right) = \sum_{j=0}^{\infty} \gamma^j U\left(x_{k+j}, {}^{(l)}\hat{u}_{k+j}\right) = P_\infty^{(l)}(x_k). \tag{27}$$

Then, according to the definition of $J^*(x_k)$ in (3), for any $\varepsilon > 0$, there exists a sequence of admissible control laws $\left\{\eta_i^{(M)}\right\}$ such that the associated cost function

$$J\left(x_k, {}^{(M)}\underline{\hat{u}}_k^\infty\right) = \sum_{j=0}^{\infty} \gamma^j U\left(x_{k+j}, {}^{(M)}\hat{u}_{k+j}\right) = P_\infty^{(M)}(x_k) \tag{28}$$

satisfies $J\left(x_k, {}^{(M)}\underline{\hat{u}}_k^\infty\right) \le J^*(x_k) + \varepsilon$. Combining with (25), we have

$$V_\infty(x_k) \le P_\infty^{(M)}(x_k) \le J^*(x_k) + \varepsilon. \tag{29}$$

Since ε is chosen arbitrarily, we get

$$V_\infty(x_k) \le J^*(x_k). \tag{30}$$

On the other hand, considering $V_{i+1}(x_k) \le P_{i+1}^{(l)}(x_k) \le Y_l, \forall l, i$, we can obtain $V_\infty(x_k) \le \inf_l\{Y_l\}$. According to the definition of admissible control law sequence, the control law sequence associated with the cost function $V_\infty(x_k)$ must be an admissible one. Hence, there exists a sequence of admissible control laws $\left\{\eta_i^{(N)}\right\}$ such that $V_\infty(x_k) = P_\infty^{(N)}(x_k)$. Combining with (27), we get $V_\infty(x_k) = J\left(x_k, {}^{(N)}\underline{\hat{u}}_k^\infty\right)$. Because $J^*(x_k)$ is the infimum of all cost functions associated with the admissible control sequences starting at k with length ∞, we can obtain

$$V_\infty(x_k) \ge J^*(x_k). \tag{31}$$

Based on (30) and (31), we can acquire that J^* is the limit of sequence $\{V_i\}$, i.e., $V_\infty(x_k) = J^*(x_k)$. ■

From the aforementioned conclusions, we derive that the limit of the cost function sequence $\{V_i\}$ satisfies the DTHJB equation, which can be seen in (21). In addition, based on Theorem 2, we get $V_\infty(x_k) = J^*(x_k)$. Therefore, we can obtain that the cost function sequence $\{V_i(x_k)\}$ converges to the optimal cost function $J^*(x_k)$ of the DTHJB equation, i.e., $V_i \to J^*$ as $i \to \infty$. Then, according to (6) and (18), we can conclude the convergence of the corresponding control law sequence, i.e., $\lim_{i\to\infty} v_i(x_k) = u^*(x_k)$.

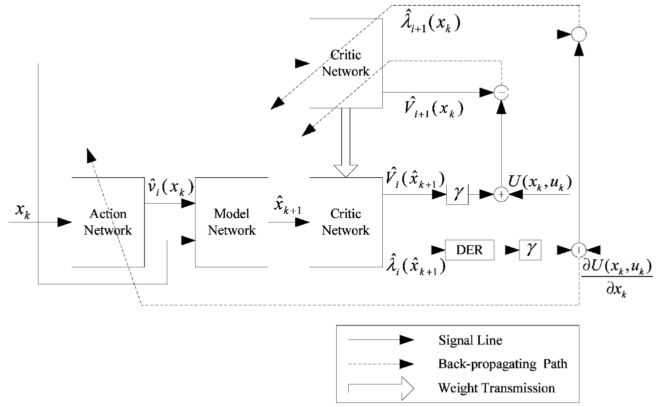

Fig. 1. The structural diagram of the iterative GDHP algorithm.

D. NN Implementation of the Iterative ADP Algorithm Using GDHP Technique

In the iterative GDHP algorithm, there are three NNs, which are model network, critic network, and action network. All the networks are chosen as three-layer feedforward NNs. The diagram of the whole structure is shown in Fig. 1, where

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

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 $\hat{v}_i(x_k)$, we can compute $\hat{x}_{k+1}$ by (8), i.e., $\hat{x}_{k+1} = \omega_m^T(k)\sigma(\nu_m^{*T}[x_k^T \ \hat{v}_i^T(x_k)]^T)$. As a result, we avoid the requirement of knowing $f(x_k)$ and $g(x_k)$ during the implementation process of the iterative GDHP algorithm.

Next, the learned NN system model will be used in the process of training critic network and action network.

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 named costate function and denoted as $\lambda_i(x_k)$. The output of the critic network is

$$\begin{bmatrix} \hat{V}_i(x_k) \\ \hat{\lambda}_i(x_k) \end{bmatrix} = \begin{bmatrix} \omega_{ci}^{1T} \\ \omega_{ci}^{2T} \end{bmatrix} \sigma\left(\nu_{ci}^T x_k\right) = \omega_{ci}^T \sigma\left(\nu_{ci}^T x_k\right) \tag{32}$$

where $\omega_{ci} = \left[\omega_{ci}^1 \ \ \omega_{ci}^2\right]$. Then, we have

$$\hat{V}_i(x_k) = \omega_{ci}^{1T} \sigma\left(\nu_{ci}^T x_k\right) \tag{33}$$

and

$$\hat{\lambda}_i(x_k) = \omega_{ci}^{2T} \sigma\left(\nu_{ci}^T x_k\right). \tag{34}$$

The target function can be written as

$$V_{i+1}(x_k) = x_k^T Q x_k + v_i^T(x_k) R v_i(x_k) + \gamma \hat{V}_i(\hat{x}_{k+1}) \tag{35}$$

and

$$\begin{aligned} \lambda_{i+1}(x_k) &= \frac{\partial\left(x_k^T Q x_k + v_i^T(x_k) R v_i(x_k)\right)}{\partial x_k} + \gamma \frac{\partial \hat{V}_i(\hat{x}_{k+1})}{\partial x_k} \\ &= 2Qx_k + 2\left(\frac{\partial v_i(x_k)}{\partial x_k}\right)^T R v_i(x_k) \\ &\quad + \gamma \left(\frac{\partial \hat{x}_{k+1}}{\partial x_k} + \frac{\partial \hat{x}_{k+1}}{\partial \hat{v}_i(x_k)} \frac{\partial \hat{v}_i(x_k)}{\partial x_k} \right)^T \hat{\lambda}_i(\hat{x}_{k+1}). \end{aligned} \tag{36}$$

We define the error function for training the critic network as

$$e_{cik}^1 = \hat{V}_i(x_k) - V_{i+1}(x_k) \tag{37}$$

and

$$e_{cik}^2 = \hat{\lambda}_i(x_k) - \lambda_{i+1}(x_k). \tag{38}$$

Then, the objective function to be minimized is $E_{cik} = (1-\beta)E_{cik}^1 + \beta E_{cik}^2$, where $E_{cik}^1 = e_{cik}^{1T} e_{cik}^1/2$ and $E_{cik}^2 = e_{cik}^{2T} e_{cik}^2/2$. The weight updating rule for training the critic network is also gradient-based adaptation given by

$$\omega_{ci}(j+1) = \omega_{ci}(j) - \alpha_c \left[(1-\beta)\frac{\partial E_{cik}^1}{\partial \omega_{ci}(j)} + \beta \frac{\partial E_{cik}^2}{\partial \omega_{ci}(j)} \right] \tag{39}$$

$$\nu_{ci}(j+1) = \nu_{ci}(j) - \alpha_c \left[(1-\beta)\frac{\partial E_{cik}^1}{\partial \nu_{ci}(j)} + \beta \frac{\partial E_{cik}^2}{\partial \nu_{ci}(j)} \right] \tag{40}$$

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

In the action network, the state x_k is used as input to obtain the control vector as its output, which can be expressed by

$$\hat{v}_i(x_k) = \omega_{ai}^T \sigma\left(\nu_{ai}^T x_k\right). \tag{41}$$

The target control input is given by

$$v_i(x_k) = -\frac{\gamma}{2} R^{-1} \hat{g}^T(x_k) \frac{\partial \hat{V}_i(\hat{x}_{k+1})}{\partial \hat{x}_{k+1}}. \tag{42}$$

The error function of the action network can be defined as

$$e_{aik} = \hat{v}_i(x_k) - v_i(x_k). \tag{43}$$

The weights of the action network are updated to minimize the performance measure $E_{aik} = e_{aik}^T e_{aik}/2$. Similarly, the weight updating algorithm is

$$\omega_{ai}(j+1) = \omega_{ai}(j) - \alpha_a \left[\frac{\partial E_{aik}}{\partial \omega_{ai}(j)} \right] \tag{44}$$

$$\nu_{ai}(j+1) = \nu_{ai}(j) - \alpha_a \left[\frac{\partial E_{aik}}{\partial \nu_{ai}(j)} \right] \tag{45}$$

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

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

IV. Simulation Study

Consider the following discrete-time nonlinear system:

$$x_{k+1} = \begin{bmatrix} -\sin(0.5x_{2k}) \\ -\cos(1.4x_{2k})\sin(0.9x_{1k}) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k$$

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. The cost function is chosen as $U(x_k, u_k) = x_k^T x_k + u_k^T u_k$.

We choose three-layer feedforward NNs as model network, critic network, and action network with the structures 3–8–2, 2–8–3, and 2–8–1, respectively. In the system identification process, the initial weights between input layer and hidden layer, and between hidden layer and output layer are chosen randomly in $[-0.5, 0.5]$ and

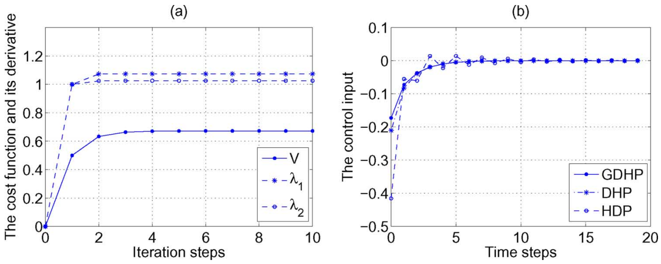

Fig. 2. Simulation results. (a) The convergence process of the cost function and its derivative of the iterative GDHP algorithm. (b) The control input $\boldsymbol{u}$.

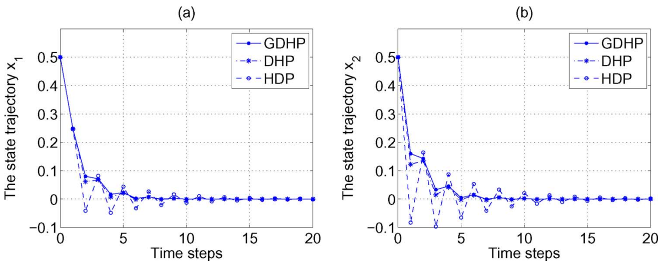

Fig. 3. Simulation results. (a) The state trajectory $\boldsymbol{x}_1$. (b) The state trajectory $\boldsymbol{x}_2$.

$[-0.1, 0.1]$, respectively. We apply the NN identification scheme for 100 steps under the learning rate $\alpha_m = 0.05$. The NN identifier can learn the unknown nonlinear system successfully. Then, we finish the training of the model network and keep its weights unchanged.

The initial weights of the critic network and action network are all set to be random in $[-0.1, 0.1]$. Then, let the discount factor $\gamma = 1$ and the adjusting parameter $\beta = 0.5$, we train the critic network and action network for 10 training cycles with each cycle of 2000 steps. In the training process, the learning rate $\alpha_c = \alpha_a = 0.05$. The convergence process of the cost function and its derivative of the iterative GDHP algorithm at time instant $k = 0$ is shown in Fig. 2(a). We can see that the iterative cost function sequence does converge to the optimal cost function quite quickly, which also indicates the effectiveness of the iterative GDHP algorithm. In addition, if we compare the results obtained by using different discount factors, we can find that smaller discount factor can insure quicker convergence of the cost function sequence.

Moreover, in order to make comparison with the iterative ADP algorithm using HDP and DHP technique (iterative HDP algorithm and iterative DHP algorithm for brief), we also present the controllers designed by iterative HDP algorithm and iterative DHP algorithm, respectively. Then, for given initial state $x_{10} = 0.5$ and $x_{20} = 0.5$, we apply the optimal control laws designed by iterative GDHP, HDP, and DHP algorithm to the controlled system for 20 time steps, respectively, and obtain the control curves are shown in Fig. 2(b). The corresponding state curves are shown in Fig. 3(a) and (b). It can be seen from the simulation results that the controller designed by the iterative GDHP algorithm has better performance than iterative HDP algorithm and iterative DHP algorithm. The most important property that the iterative GDHP algorithm is superior to the iterative DHP algorithm is that the former can show us directly the convergence process of the cost function sequence. Besides, the time that the iterative GDHP algorithm takes in the entire computation process is much less than that of HDP. For the same problem, the iterative GDHP algorithm takes about 26.6 s, while the iterative HDP algorithm takes about 61.3 s before satisfactory results are obtained.

V. Conclusion

An effective iterative algorithm is investigated in this paper to design the near optimal controller for a class of unknown discrete-time nonlinear systems with discount factor in the cost function. The NN-based GDHP technique is introduced for the purpose of implementing the iterative ADP algorithm. The simulation study demonstrates the validity of the derived optimal control scheme.

References

[1] R. E. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.

[2] 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.

[3] P. J. Werbos, "Using ADP to understand and replicate brain intelligence: The next level design," in *Proc. IEEE Symp. Approx. Dynamic Program. Reinforcement Learning*, Honolulu, HI, Apr. 2007, pp. 209–216.

[4] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand Reinhold, 1992, ch. 13.

[5] F. Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[6] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Jul. 2009.

[7] J. Fu, H. He, and X. Zhou, "Adaptive learning and control for MIMO system based on adaptive dynamic programming," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1133–1148, Jul. 2011.

[8] F. Y. Wang, N. Jin, D. Liu, and Q. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with $\boldsymbol{\varepsilon}$-error bound," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.

[9] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ: Wiley, 2007.

[10] 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.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 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, Sep. 1997.

[14] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-Time Systems*. Boca Raton, FL: CRC Press, 2006.

[15] P. He and S. Jagannathan, "Reinforcement learning neural-network-based controller for nonlinear discrete-time systems with input constraints," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 425–436, Apr. 2007.

[16] R. Ganesan, T. K. Das, and K. M. Ramachandran, "A multiresolution analysis-assisted reinforcement learning approach to run-by-run control," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 2, pp. 182–193, Apr. 2007.

[17] D. Liu, D. Wang, and D. Zhao, "Adaptive dynamic programming for optimal control of unknown nonlinear discrete-time systems," in *Proc. IEEE Symp. Adaptive Dynamic Program. Reinforcement Learning*, Paris, France, Apr. 2011, pp. 242–249.

[18] D. Liu, Y. Zhang, and H. Zhang, "A self-learning call admission control scheme for CDMA cellular networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 5, pp. 1219–1228, Sep. 2005.

[19] 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.

[20] 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. Autom. Sci. Eng.*, vol. 2, no. 2, pp. 121–131, Apr. 2005.

[21] 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, pp. 779–791, May 2005.

[22] 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.

[23] 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, Sep. 2009.

[24] D. Vrabie and F. L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, no. 3, pp. 237–246, Apr. 2009.

[25] T. Dierks, B. T. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Networks*, vol. 22, no. 5–6, pp. 851–860, Jul.-Aug. 2009.

[26] D. Wang, D. Liu, and Q. Wei, "Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach," *Neurocomputing*, vol. 78, no. 1, pp. 14–22, Feb. 2012.

[27] 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.

[28] W. S. Lin and J. W. Sheu, "Optimization of train regulation and energy usage of metro lines using an adaptive-optimal-control algorithm," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 4, pp. 855–864, Oct. 2011.