Policy Iteration Adaptive Dynamic Programming Algorithm for Discrete-Time Nonlinear Systems

Derong Liu, *Fellow, IEEE*, and Qinglai Wei, *Member, IEEE*

***Abstract*— This paper is concerned with a new discrete-time policy iteration adaptive dynamic programming (ADP) method for solving the infinite horizon optimal control problem of nonlinear systems. The idea is to use an iterative ADP technique to obtain the iterative control law, which optimizes the iterative performance index function. The main contribution of this paper is to analyze the convergence and stability properties of policy iteration method for discrete-time nonlinear systems for the first time. It shows that the iterative performance index function is nonincreasingly convergent to the optimal solution of the Hamilton–Jacobi–Bellman equation. It is also proven that any of the iterative control laws can stabilize the nonlinear systems. Neural networks are used to approximate the performance index function and compute the optimal control law, respectively, for facilitating the implementation of the iterative ADP algorithm, where the convergence of the weight matrices is analyzed. Finally, the numerical results and analysis are presented to illustrate the performance of the developed method.**

***Index Terms*— Adaptive critic designs, adaptive dynamic programming (ADP), approximate dynamic programming, discrete-time policy iteration, neural networks, neurodynamic programming, nonlinear systems, optimal control, reinforcement learning.**

I. Introduction

DYNAMIC programming is a very useful tool in solving optimal control problems. However, due to the curse of dimensionality [1], it is often computationally untenable to run dynamic programming for obtaining the optimal solution. The adaptive/approximate dynamic programming (ADP) algorithms were proposed in [2] and [3] as a way to solve optimal control problems forward in time and gained much attention from the researchers [4]–[12]. There are several synonyms used for ADP including adaptive critic designs [13]–[15], ADP [16]–[19], approximate dynamic programming [20], [21], neural dynamic programming [22], neurodynamic programming [23], and reinforcement learning [24]–[26]. In [14] and [21], ADP approaches were classified into several main schemes, which include heuristic dynamic programming (HDP), action-dependent HDP, also known as Q-learning [27], dual heuristic dynamic programming (DHP), action-dependent DHP, globalized DHP (GDHP), and action-dependent GDHP. Iterative methods are also used in ADP to obtain the solution of the Hamilton–Jacobi–Bellman (HJB) equation indirectly and have received more and more attention [28]–[35]. There are two main iterative ADP algorithms, which are value iteration ADP (value iteration for brief) algorithm and policy iteration ADP (policy iteration for brief) algorithm [36], [37].

Value iteration algorithm for optimal control of discrete-time nonlinear systems was given in [38]. In 2008, Al-Tamimi *et al.* [39] studied value iteration algorithm for deterministic discrete-time affine nonlinear systems, which was referred to as HDP and was proposed for finding the optimal control law. Starting from a zero initial performance index function, it is proven in [39] that the iterative performance index function is a nondecreasing sequence and bounded. When the iteration index increases to infinity, the iterative performance index function converges to the optimal performance index function, which satisfies the HJB equation. In 2008, Zhang *et al.* [40] applied value iteration ADP to solve optimal tracking control problems for nonlinear systems. Liu *et al*. [16] realized the value iteration ADP by GDHP. For the value iteration algorithm of ADP, the initial performance index function is required to be zero [16], [18], [39]–[42]. On the other hand, for value iteration algorithm, the stability of the system under the iterative control law cannot be guaranteed. This means that only the converged optimal control law can be used to control the nonlinear system, and all the iterative controls during the iteration procedure may be invalid. Therefore, the computation efficiency of the value iteration algorithm of ADP is very low. Till now, all the value iteration algorithms are implemented offline, which limit their practical applications.

Policy iteration algorithm for continuous-time systems was proposed in [17]. Murray *et al*. [17] proved that for continuous-time affine nonlinear systems, the iterative performance index function will converge to the optimum nonincreasingly and each of the iterative controls stabilizes the nonlinear systems using policy iteration algorithms. This is a great merit of policy iteration algorithm and hence achieved many applications for solving optimal control problems of nonlinear systems. In 2005, Abu-Khalaf and Lewis [43] proposed a policy iteration algorithm for continuous-time nonlinear system with control constraints. Zhang *et al*. [44] applied policy iteration algorithm to solve continuous-time nonlinear two-person zero-sum games.

Manuscript received August 2, 2012; accepted August 27, 2013. Date of publication September 25, 2013; date of current version February 14, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61034002, Grant 61233001, Grant 61273140, and Grant 61374105, in part by Beijing Natural Science Foundation under Grant 4132078, and in part by the Early Career Development Award of SKLMCCS. The acting Editor-in-Chief who handled the review of this paper was Ivo Bukovsky.

The authors are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (email: derong.liu@ia.ac.cn; qinglai.wei@ia.ac.cn).

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/TNNLS.2013.2281663

2162-237X © 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Vamvoudakis *et al.* [45] proposed a multiagent differential graphical games for continuous-time linear systems using policy iteration algorithms. Bhasin *et al.* [46] proposed an online actor-critic-identifier architecture to obtain the optimal control law for uncertain nonlinear systems by policy iteration algorithms. Till now, nearly all the online iterative ADP algorithms are policy iteration algorithms. However, almost all the discussions on the policy iteration algorithms are based on continuous-time control systems [44]–[47]. The discussions on policy iteration algorithms for discrete-time control systems are scarce. Only in [36] and [37] a brief sketch of policy iteration algorithm for discrete-time systems was displayed, whereas the stability and convergence properties were not discussed. To the best of our knowledge, there are still no discussions focused on the policy iteration algorithms for discrete-time systems, which motives our research.

In this paper, it is the first time that the properties of policy iteration algorithm of ADP for discrete-time nonlinear systems are analyzed. First, the policy iteration algorithm is introduced to find the optimal control law. Second, it will show that any of the iterative control laws can stabilize the nonlinear system. Third, it will show that the iterative control laws using policy iteration algorithm can make the iterative performance index functions converge to the optimal solution monotonically. Next, an effective method is developed to obtain the initial admissible control law by repeating experiments. Furthermore, to facilitate the implementation of the policy iteration algorithm, it will show how to use neural networks for implementing the developed policy iteration algorithm to obtain the iterative performance index functions and the iterative control laws. The weight convergence properties of the neural networks are also established. In numerical examples, comparisons will be displayed to show the effectiveness of the developed algorithm. For linear systems, the control results by the policy iteration algorithm will be compared with the ones by the algebraic Riccati equation (ARE) method to justify the correctness of the developed method. For nonlinear systems, the control results by the policy iteration algorithm will be compared with the ones by value iteration algorithm to show the effectiveness of the developed algorithm.

The rest of this paper is organized as follows. In Section II, the problem formulation is presented. In Section III, the policy iteration algorithm for discrete-time nonlinear systems is derived. The stability, convergence. and optimality properties will also be proven in this section. In Section IV, the neural network implementation for the optimal control scheme is discussed. In Section V, the numerical results and analyses are presented to demonstrate the effectiveness of the developed optimal control scheme. Finally, in Section VI, the conclusion is drawn and our future work is declared.

II. Problem Formulation

In this paper, we will study the following deterministic discrete-time systems:

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

where $x_k \in \mathbb{R}^n$ is the n-dimensional state vector and $u_k \in \mathbb{R}^m$ is the m-dimensional control vector. Let x_0 be the initial state and $F(x_k, u_k)$ be the system function.

Let $\underline{u}_k = \{u_k, u_{k+1}, \ldots\}$ be an arbitrary sequence of controls from k to ∞. The performance index function for state x_0 under the control sequence $\underline{u}_0 = \{u_0, u_1, \ldots\}$ is defined as

$$J(x_0, \underline{u}_0) = \sum_{k=0}^{\infty} U(x_k, u_k) \tag{2}$$

where $U(x_k, u_k) > 0$, for $\forall\, x_k, u_k \neq 0$, is the utility function.

We will study optimal control problems for (1). The goal of this paper is to find an optimal control scheme, which stabilizes (1) and simultaneously minimizes the performance index function (2). For convenience of analysis, results of this paper are based on the following assumption.

Assumption 2.1: System (1) is controllable; the system state $x_k = 0$ is an equilibrium state of (1) under the control $u_k = 0$, i.e., $F(0, 0) = 0$; the feedback control $u_k = u(x_k)$ satisfies $u_k = u(x_k) = 0$ for $x_k = 0$; the utility function $U(x_k, u_k)$ is a positive definite function for any x_k and u_k.

As (1) is controllable, there exists a stable control sequence $\underline{u}_k = \{u_k, u_{k+1} \ldots\}$ that moves x_k to zero. Let $\underline{\mathfrak{A}}_k$ denote the set, which contains all the stable control sequences. Then, the optimal performance index function can be defined as

$$J^*(x_k) = \inf\left\{J(x_k, \underline{u}_k) : \underline{u}_k \in \underline{\mathfrak{A}}_k\right\}. \tag{3}$$

According to Bellman's principle of optimality, $J^*(x_k)$ satisfies the following discrete-time HJB equation:

$$J^*(x_k) = \inf_{u_k}\left\{U(x_k, u_k) + J^*(F(x_k, u_k))\right\}. \tag{4}$$

Define the law of optimal control as

$$u^*(x_k) = \arg\inf_{u_k}\left\{U(x_k, u_k) + J^*(F(x_k, u_k))\right\}. \tag{5}$$

Hence, the HJB equation (4) can be written as

$$J^*(x_k) = U(x_k, u^*(x_k)) + J^*(F(x_k, u^*(x_k))). \tag{6}$$

We can see that if we want to obtain the optimal control law $u^*(x_k)$, we must obtain the optimal performance index function $J^*(x_k)$. Generally, $J^*(x_k)$ is unknown before all the controls $u_k \in \mathbb{R}^n$ are considered. If we adopt the traditional dynamic programming method to obtain the optimal performance index function at every time step, then we have to face the curse of dimensionality. Furthermore, the optimal control is discussed in infinite horizon. This means the length of the control sequence is infinite, which makes the optimal control nearly impossible to obtain by the HJB equation (6). To overcome this difficulty, a new iterative algorithm based on ADP is developed.

III. Policy Iteration Algorithm

In this section, the discrete-time policy iteration algorithm of ADP is developed to obtain the optimal controller for nonlinear systems. The goal of the developed policy iteration algorithm is to construct an iterative control law $v_i(x_k)$, which moves an arbitrary initial state x_0 to the equilibrium 0, and

simultaneously makes the iterative performance index function to reach the optimum. Stability proofs will be given to show that any of the iterative controls can stabilize the nonlinear system. Convergence and optimality proofs will also be given to show that the performance index functions will converge to the optimum.

A. Derivation of the Policy Iteration Algorithm

For the optimal control problems, the developed control scheme must not only stabilize the control systems, but also make the performance index function finite, i.e., the admissible control law [39].

Definition 3.1: A control law $u(x_k)$ is defined to be admissible with respect to (2) on Ω_1 if $u(x_k)$ is continuous on Ω_1, $u(0) = 0$, $u(x_k)$ stabilizes (1) on Ω_1, and $\forall x_0 \in \Omega_1$, $J(x_0)$ is finite.

In the developed policy iteration algorithm, the performance index function and control law are updated by iterations, with the iteration index i increasing from zero to infinity. Let $v_0(x_k)$ be an arbitrary admissible control law. For $i = 0$, let $V_0(x_k)$ be the iterative performance index function constructed by $v_0(x_k)$ that satisfies the following generalized HJB (GHJB) equation:

$$V_0(x_k) = U(x_k, v_0(x_k)) + V_0(x_{k+1}) \tag{7}$$

where $x_{k+1} = F(x_k, v_0(x_k))$. Then, the iterative control law is computed by

$$\begin{aligned} v_1(x_k) &= \arg\min_{u_k} \{U(x_k, u_k) + V_0(x_{k+1})\} \\ &= \arg\min_{u_k} \{U(x_k, u_k) + V_0(F(x_k, u_k))\}. \end{aligned} \tag{8}$$

For $\forall i = 1, 2, \ldots$, let $V_i(x_k)$ be the iterative performance index function constructed by $v_i(x_k)$, which satisfies the following GHJB equation:

$$V_i(x_k) = U(x_k, v_i(x_k)) + V_i(F(x_k, v_i(x_k))) \tag{9}$$

and the iterative control law is updated by

$$\begin{aligned} v_{i+1}(x_k) &= \arg\min_{u_k} \{U(x_k, u_k) + V_i(x_{k+1})\} \\ &= \arg\min_{u_k} \{U(x_k, u_k) + V_i(F(x_k, u_k))\}. \end{aligned} \tag{10}$$

From the policy iteration algorithm (7)–(10), we can see that the iterative performance index function $V_i(x_k)$ is used to approximate $J^*(x_k)$ and the iterative control law $v_i(x_k)$ is used to approximate $u^*(x_k)$. As (9) is generally not the HJB equation, we have the iterative performance index function $V_i(x_k) \neq J^*(x_k)$ and $v_i(x_k) \neq u^*(x_k)$ for $\forall i = 0, 1, \ldots$. Therefore, it is necessary to determine whether the algorithm is convergent, which means that $V_i(x_k)$ and $v_i(x_k)$ converge to the optimal ones as $i \to \infty$. In the following section, we will show the properties of the policy iteration algorithm.

B. Properties of the Policy Iteration Algorithm

For the policy iteration algorithm of continuous-time nonlinear systems [17], it shows that any of the iterative control laws can stabilize the system. This is a merit of the policy iteration algorithm. For the discrete-time systems, the stability of the system can also be guaranteed under the iterative control law.

Lemma 3.1: For $i = 0, 1, \ldots$, let $V_i(x_k)$ and $v_i(x_k)$ be obtained by the policy iteration algorithm (7)–(10), where $v_0(x_k)$ is an arbitrary admissible control law. If Assumption 2.1 holds, then for $\forall i = 0, 1, \ldots$, the iterative control law $v_i(x_k)$ stabilizes the nonlinear system (1).

For continuous-time policy iteration algorithms [17], according to the derivative of the difference for the iterative performance index functions along with time variable t, it is proven that the iterative performance index functions are nonincreasingly convergent to the optimal solution of the HJB equation. For discrete-time systems, the method for the continuous-time is invalid. Thus, a new convergence proof is established in this paper. We will show that using the policy iteration algorithm of discrete-time nonlinear systems, the iterative performance index functions are also nonincreasingly convergent to the optimum.

Theorem 3.1: For $i = 0, 1, \ldots$, let $V_i(x_k)$ and $v_i(x_k)$ be obtained by (7)–(10). If Assumption 2.1 holds, then for $\forall x_k \in \mathbb{R}^n$, the iterative performance index function $V_i(x_k)$ is a monotonically nonincreasing sequence for $\forall i \geq 0$

$$V_{i+1}(x_k) \leq V_i(x_k). \tag{11}$$

Proof: First, for $i = 0, 1, \ldots$, define a new iterative performance index function $\Gamma_{i+1}(x_k)$ as

$$\Gamma_{i+1}(x_k) = U(x_k, v_{i+1}(x_k)) + V_i(x_{k+1}) \tag{12}$$

where $v_{i+1}(x_k)$ is obtained by (10). According to (9), (10), and (12), for $\forall x_k$, we can obtain

$$\Gamma_{i+1}(x_k) \leq V_i(x_k). \tag{13}$$

Inequality (11) will be proven by mathematical induction. According to Lemma 3.1, for $i = 0, 1, \ldots$, we have that $v_{i+1}(x_k)$ is a stable control law. Then, we have $x_k \to 0$, for $\forall k \to \infty$. Without loss of generality, let $x_N = 0$, where $N \to \infty$. We can obtain

$$V_{i+1}(x_N) = \Gamma_{i+1}(x_N) = V_i(x_N) = 0. \tag{14}$$

First, we let $k = N - 1$. According to (10), we have

$$\begin{aligned} v_{i+1}(x_{N-1}) &= \arg\min_{u_{N-1}} \{U(x_{N-1}, u_{N-1}) + V_i(x_N)\} \\ &= \arg\min_{u_{N-1}} \{U(x_{N-1}, u_{N-1}) + V_{i+1}(x_N)\}. \end{aligned} \tag{15}$$

According to (9), we can obtain

$$\begin{aligned} V_{i+1}(x_{N-1}) &= U(x_{N-1}, v_{i+1}(x_{N-1})) + V_{i+1}(x_N) \\ &= \min_{u_{N-1}} \{U(x_{N-1}, u_{N-1}) + V_i(x_N)\} \\ &\leq U(x_{N-1}, v_i(x_{N-1})) + V_i(x_N) \\ &= V_i(x_{N-1}). \end{aligned} \tag{16}$$

Therefore, the conclusion holds for $k = N - 1$. Assume that the conclusion holds for $k = l + 1$, $l = 0, 1, \ldots$ For $k = l$, we can obtain

$$\begin{aligned} V_{i+1}(x_l) &= U(x_l, v_{i+1}(x_l)) + V_{i+1}(x_{l+1}) \\ &\leq U(x_l, v_{i+1}(x_l)) + V_i(x_{l+1}) \\ &= \Gamma_{i+1}(x_l). \end{aligned} \tag{17}$$

According to (13), for $\forall x_l$, we have

$$\Gamma_{i+1}(x_l) \leq V_i(x_l). \tag{18}$$

According to (17) and (18), we can obtain that for $i = 0, 1, \ldots$, (11) holds for $\forall x_k$. The proof is completed. ■

Corollary 3.1: For $i = 0, 1, \ldots$, let $V_i(x_k)$ and $v_i(x_k)$ be obtained by (7)–(10), where $v_0(x_k)$ is an arbitrary admissible control law. If Assumption 2.1 holds, then for $\forall i = 0, 1, \ldots$, the iterative control law $v_i(x_k)$ is admissible.

From Theorem 3.1, we know that the iterative performance index function $V_i(x_k) \geq 0$ is monotonically nonincreasing and bounded below for iteration index $i = 1, 2, \ldots$ Hence, we can derive the following theorem.

Theorem 3.2: For $i = 0, 1, \ldots$, let $V_i(x_k)$ and $v_i(x_k)$ be obtained by (7)–(10). If Assumption 2.1 holds, then we have that the iterative performance index function $V_i(x_k)$ converges to the optimal performance index function $J^*(x_k)$, as $i \to \infty$

$$\lim_{i\to\infty} V_i(x_k) = J^*(x_k) \tag{19}$$

which satisfies the HJB equation (4).

Proof: The statement can be proven by the following three steps.

1) Show that the limit of the iterative performance index function $V_i(x_k)$ satisfies the HJB equation, as $i \to \infty$.

According to Theorem 3.1, as $V_i(x_k)$ is nonincreasing and lower bounded sequence, we have that the limit the iterative performance index function $V_i(x_k)$ exists as $i \to \infty$. Define the performance index function $V_\infty(x_k)$ as the limit of the iterative function $V_i(x_k)$

$$V_\infty(x_k) = \lim_{i\to\infty} V_i(x_k). \tag{20}$$

According to the definition of $\Gamma_{i+1}(x_k)$ in (12), we have

$$\begin{aligned}\Gamma_{i+1}(x_k) &= U(x_k, v_{i+1}(x_k)) + V_i(x_{k+1}) \\ &= \min_{u_k}\{U(x_k, u_k) + V_i(x_{k+1})\}.\end{aligned} \tag{21}$$

According to (17), we can obtain

$$V_{i+1}(x_k) \leq \Gamma_{i+1}(x_k) \tag{22}$$

for $\forall x_k$. Let $i \to \infty$. We can obtain

$$\begin{aligned}V_\infty(x_k) &= \lim_{i\to\infty} V_i(x_k) \leq V_{i+1}(x_k) \leq \Gamma_{i+1}(x_k) \\ &\leq \min_{u_k}\{U(x_k, u_k) + V_i(x_{k+1})\}.\end{aligned} \tag{23}$$

Thus, we can obtain

$$V_\infty(x_k) \leq \min_{u_k}\{U(x_k, u_k) + V_\infty(x_{k+1})\}. \tag{24}$$

Let $\varepsilon > 0$ be an arbitrary positive number. Since $V_i(x_k)$ is nonincreasing for $i \geq 1$ and $\lim_{i\to\infty} V_i(x_k) = V_\infty(x_k)$, there exists a positive integer p such that

$$V_p(x_k) - \varepsilon \leq V_\infty(x_k) \leq V_p(x_k). \tag{25}$$

Hence, we can obtain

$$\begin{aligned}V_\infty(x_k) &\geq U(x_k, v_p(x_k)) + V_p(F(x_k, v_p(x_k))) - \varepsilon \\ &\geq U(x_k, v_p(x_k)) + V_\infty(F(x_k, v_p(x_k))) - \varepsilon \\ &\geq \min_{u_k}\{U(x_k, u_k) + V_\infty(x_{k+1})\} - \varepsilon.\end{aligned}$$

Since ε is arbitrary, we have

$$V_\infty(x_k) \geq \min_{u_k}\{U(x_k, u_k) + V_\infty(x_{k+1})\}. \tag{26}$$

Combining (24) and (26), we can obtain

$$V_\infty(x_k) = \min_{u_k}\{U(x_k, u_k) + V_\infty(x_{k+1})\}. \tag{27}$$

Next, let $\mu(x_k)$ be an arbitrary admissible control law, and define a new performance index function $P(x_k)$, which satisfies

$$P(x_k) = U(x_k, \mu(x_k)) + P(x_{k+1}). \tag{28}$$

Then, we can declare the second step of the proof.

2) Show that for an arbitrary admissible control law $\mu(x_k)$, the performance index function $V_\infty(x_k) \leq P(x_k)$.

The statement can be proven by mathematical induction. As $\mu(x_k)$ is an admissible control law, we have $x_k \to 0$ as $k \to \infty$. Without loss of generality, let $x_N = 0$, where $N \to \infty$. According to (28), we have

$$\begin{aligned}P(x_k) = \lim_{N\to\infty}\{&U(x_k, \mu(x_k)) + U(x_{k+1}, \mu(x_{k+1})) + \cdots \\ &+ U(x_{N-1}, \mu(x_{N-1})) + P(x_N)\}\end{aligned} \tag{29}$$

where $x_N = 0$. According to (27), the iterative performance index function $V_\infty(x_k)$ can be expressed as

$$\begin{aligned}V_\infty(x_k) &= \lim_{N\to\infty}\{U(x_k, v_\infty(x_k)) + U(x_{k+1}, v_\infty(x_{k+1})) \\ &\quad + \cdots + U(x_{N-1}, v_\infty(x_{N-1})) + V_\infty(x_N)\} \\ &= \lim_{N\to\infty}\left\{\min_{u_k}\left\{U(x_k, u_k) + \min_{u_{k+1}}\left\{U(x_{k+1}, u_{k+1}) + \cdots \right.\right.\right. \\ &\quad \left.\left.\left. + \min_{u_{N-1}}\{U(x_{N-1}, u_{N-1}) + V_\infty(x_N)\}\right\}\right\}\right\}.\end{aligned} \tag{30}$$

According to Corollary 3.1, for $\forall i = 0, 1, \ldots$, since $v_i(x_k)$ is an admissible control law, we have that $v_\infty(x_k) = \lim_{i\to\infty} v_i(x_k)$ is an admissible control law. Then, we can get $x_N = 0$, where $N \to \infty$, which means $V_\infty(x_N) = P(x_N) = 0$. For $N - 1$, according to (27), we can obtain

$$\begin{aligned}P(x_{N-1}) &= U(x_{N-1}, \mu(x_{N-1})) + P(x_N) \\ &\geq \min_{u_{N-1}}\{U(x_{N-1}, u_{N-1}) + P(x_N)\} \\ &= \min_{u_{N-1}}\{U(x_{N-1}, u_{N-1}) + V_\infty(x_N)\} \\ &= V_\infty(x_{N-1}).\end{aligned} \tag{31}$$

Assume that the statement holds for $k = l + 1$, $l = 0, 1, \ldots$ Then, for $k = l$, we have

$$\begin{aligned}P(x_l) &= U(x_l, \mu(x_l)) + P(x_{l+1}) \\ &\geq \min_{u_l}\{U(x_l, u_l) + P(x_{l+1})\} \\ &\geq \min_{u_l}\{U(x_l, u_l) + V_\infty(x_{l+1})\} \\ &= V_\infty(x_l).\end{aligned} \tag{32}$$

Hence, for $\forall x_k$, $k = 0, 1, \ldots$, the inequality

$$V_\infty(x_k) \leq P(x_k) \tag{33}$$

holds. Mathematical induction is completed.

3) Show that the performance index function $V_\infty(x_k)$ is equal to the optimal performance index function $J^*(x_k)$.

According to the definition of $J^*(x_k)$ in (3), for $\forall i = 0, 1, \ldots$, we have

$$V_i(x_k) \geq J^*(x_k). \tag{34}$$

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

$$V_\infty(x_k) \geq J^*(x_k). \tag{35}$$

On the other hand, for an arbitrary admissible control law $\mu(x_k)$, we have (33) holds. Let $\mu(x_k) = u^*(x_k)$, where $u^*(x_k)$ is an optimal control law. Then, we can obtain

$$V_\infty(x_k) \leq J^*(x_k). \tag{36}$$

According to (35) and (36), we can obtain (19). The proof is completed. ■

Corollary 3.2: Let $x_k \in \mathbb{R}^n$ be an arbitrary state vector. If Theorem 3.2 holds, then we have that the iterative control law $v_i(x_k)$ converges to the optimal control law as $i \to \infty$, i.e., $u^*(x_k) = \lim_{i\to\infty} v_i(x_k)$.

Remark 3.1: From Theorems 3.1 and 3.2, we have proven that the iterative performance index functions are monotonically nonincreasing and convergent to the optimal performance index function as $i \to \infty$. The stability of the nonlinear systems can also be guaranteed under the iterative control laws. Hence, we can say that the convergence and stability properties of the policy iteration algorithms for continuous-time nonlinear systems are also valid for the policy iteration algorithms for discrete-time nonlinear systems. However, we emphasize that the analysis techniques between the continuous- and discrete-time policy iteration algorithms are quite different. First, the HJB equations of continuous- and discrete-time systems are different inherently. Second, the analysis method for continuous-time policy iteration algorithm is based on derivative techniques [17], [43]–[47]. The analysis method for continuous-time policy iteration algorithm is invalid for discrete-time situation. In this paper, for the first time, we have established the properties for the discrete-time policy iteration algorithms, which established the theoretical basis of the applications for the optimal control problem of discrete-time nonlinear systems using policy iteration algorithms. Hence, we say that the discussion in this paper is meaningful.

Remark 3.2: In [39], value iteration algorithm of ADP is proposed to obtain the optimal solution of the HJB equation for the following discrete-time affine nonlinear systems:

$$x_{k+1} = f(x_k) + g(x_k)u_k \tag{37}$$

with the performance index function

$$J(x_k) = \sum_{j=k}^{\infty} \left(x_j^T Q x_j + u_j^T R u_j \right) \tag{38}$$

where Q and R are defined as the penalizing matrices for system state and control vectors, respectively. Let $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ be both positive definite matrices. Starting from $V_0(x_k) \equiv 0$, the value iteration algorithm can be expressed as

$$\begin{cases} u_i(x_k) = -\dfrac{1}{2} R^{-1} g(x_k)^T \dfrac{\partial V_i(x_{k+1})}{\partial x_{k+1}} \\ V_{i+1}(x_k) = x_k^T Q x_k + u_i^T(x_k) R u_i(x_k) + V_i(x_{k+1}). \end{cases} \tag{39}$$

It was proven in [39] that $V_i(x_k)$ is a monotonically nondecreasing sequence and bounded, and hence converges to $J^*(x_k)$. However, using the value iteration equation (39), the stability of the system under the iterative control law cannot be guaranteed. We should point out that using the policy iteration algorithm (7)–(10), the stability property can be guaranteed. In Section V, the numerical results and comparisons between the policy and value iteration algorithms will be presented to show these properties.

Hence, we say that if an admissible control of the nonlinear system is known, then policy iteration algorithm is preferred to obtain the optimal control law with good convergence and stability properties. From this point of view, the initial admissible control law is a key to success for the policy iteration algorithm. In the following section, an effective method will be discussed to obtain the initial admissible control law using neural networks.

C. Obtaining the Initial Admissible Control Law

As the policy iteration algorithm requires to start with an admissible control law, the method of obtaining the admissible control law is important to the implementation of the algorithm. Actually, for all the policy iteration algorithms of ADP, including [17] and [43]–[47], the initial admissible control law is necessary to implement their algorithms. Unfortunately, to the best of our knowledge, there is no theory on how to design the initial admissible control law. In this paper, we will give an effective method to obtain the initial admissible control law by the experiments using neural networks.

First, let $\Psi(x_k) \geq 0$ be an arbitrary semipositive definite function. For $i = 0, 1, \ldots$, we define a new performance index function as

$$\Phi_{i+1}(x_k) = U(x_k, \mu(x_k)) + \Phi_i(x_{k+1}) \tag{40}$$

where $\Phi_0(x_k) = \Psi(x_k)$. Then, we can derive the following theorem.

Theorem 3.3: Suppose Assumption 2.1 holds. Let $\Psi(x_k) \geq 0$ be an arbitrary semipositive definite function. Let $\mu(x_k)$ be an arbitrary control law for (1), which satisfies $\mu(0) = 0$. Define the iterative performance index function $\Phi_i(x_k)$ as (40), where $\Phi_0(x_k) = \Psi(x_k)$. Then, $\mu(x_k)$ is an admissible control law if and only if the limit of $\Phi_i(x_k)$ exists as $i \to \infty$.

Proof: We first prove the sufficiency of the statement. Assume that $\mu(x_k)$ is an admissible control law. According to (40), we have

$$\begin{aligned} \Phi_{i+1}(x_k) - \Phi_i(x_k) &= U(x_k, \mu(x_k)) + \Phi_i(x_{k+1}) \\ &\quad -(U(x_k, \mu(x_k)) + \Phi_{i-1}(x_{k+1})) \\ &= \Phi_i(x_{k+1}) - \Phi_{i-1}(x_{k+1}). \end{aligned} \tag{41}$$

Algorithm 1 Obtain the Initial Admissible Control Law

Initialization:

Choose a semi-positive definite function $\Psi(x_k) \geq 0$;
Initialize two neural networks (critic networks for brief) $cnet1$ and $cnet2$ with small random weights;
Let $\Phi_0(x_k) = \Psi(x_k)$;
Give the max iteration of computation $i_{\max}$.

Iteration:

1: Establish a neural network (action network for brief) with small random weights to generate an initial control law $\mu(x_k)$ with $\mu(x_k) = 0$ for $x_k = 0$;
2: Let $i = 0$. Train the critic network $cnet1$ to approximate $\Phi_1(x_k)$, where $\Phi_1(x_k)$ satisfies

$$\Phi_1(x_k) = U(x_k, \mu(x_k)) + \Phi_0(x_{k+1});$$

3: Copy $cnet1$ to $cnet2$, i.e., $cnet2 = cnet1$;
4: Let $i = i + 1$. Use $cnet2$ to get $\Phi_i(x_{k+1})$ and train the critic network $cnet1$ to approximate $\Phi_{i+1}(x_k)$, where $\Phi_{i+1}(x_k)$ satisfies

$$\Phi_{i+1}(x_k) = U(x_k, \mu(x_k)) + \Phi_i(x_{k+1});$$

5: Use $cnet1$ to get $\Phi_{i+1}(x_k)$ and use $cnet2$ to get $\Phi_i(x_k)$. If $|\Phi_{i+1}(x_k) - \Phi_i(x_k)| < \varepsilon$, then goto Step 7. Else goto next step;
6: If $i > i_{\max}$, then goto Step 1. Else goto Step 3;
7: **return** $\mu(x_k)$ and let $v_0(x_k) = \mu(x_k)$.

Then, according to (41), we can obtain

$$\begin{cases} \Phi_{i+1}(x_k) - \Phi_i(x_k) = \Phi_1(x_{k+i}) - \Phi_0(x_{k+i}) \\ \Phi_i(x_k) - \Phi_{i-1}(x_k) = \Phi_1(x_{k+i-1}) - \Phi_0(x_{k+i-1}) \\ \qquad\qquad \vdots \\ \Phi_1(x_k) - \Phi_0(x_k) = \Phi_1(x_k) - \Phi_0(x_k). \end{cases} \tag{42}$$

Then, we can get

$$\Phi_{i+1}(x_k) = \sum_{j=0}^{i} U(x_{k+j}, \mu(x_{k+j})) + \Psi(x_k). \tag{43}$$

Let $i \to \infty$. We can obtain

$$\lim_{i\to\infty} \Phi_i(x_k) = \sum_{j=0}^{\infty} U(x_{k+j}, \mu(x_{k+j})) + \Psi(x_k). \tag{44}$$

If $\mu(x_k)$ is an admissible control law, then we have that $\sum_{j=0}^{\infty} U(x_{k+j}, \mu(x_{k+j}))$ is finite. As for an arbitrary finite x_k, $\Psi(x_k)$ is finite, then for $\forall i = 0, 1, 2, \ldots$, we have that $\Phi_{i+1}(x_k)$ is finite. Hence, $\lim_{i\to\infty} \Phi_i(x_k)$ is finite, which means $\Phi_{i+1}(x_k) = \Phi_i(x_k)$, as $i \to \infty$.

On the other hand, if the limit of $\Phi_i(x_k)$ exists as $i \to \infty$, according to (42)–(44), as $\Psi(x_k)$ is finite, then we can get that $\sum_{j=0}^{\infty} U(x_{k+j}, \mu(x_{k+j}))$ is finite. Since the utility function $U(x_k, u_k)$ is positive definite for $\forall x_k, u_k$, then we can obtain $U(x_{k+j}, \mu(x_{k+j})) \to 0$ as $j \to \infty$. As $\mu(x_k) = 0$ for $x_k = 0$, we can get that $x_k \to 0$ as $k \to \infty$, which means that (1) is stable and $\mu(x_k)$ is an admissible control law. The necessity of the statement is proven and the proof is completed. ■

Algorithm 2 Discrete-Time Policy Iteration Algorithm

Initialization:

Choose randomly an array of initial states x_0;
Choose a computation precision ε;
Give the initial admissible control law $v_0(x_k)$;
Give the max iteration of computation $i_{\max}$.

Iteration:

Let the iteration index $i = 0$;
1: Construct the iterative performance index function $V_0(x_k)$ according to $v_0(x_k)$ by

$$V_0(x_k) = U(x_k, v_0(x_k)) + V_0(x_{k+1});$$

2: Update the iterative control law by

$$v_1(x_k) = \arg\min_{u_k} \{U(x_k, u_k) + V_0(x_{k+1})\};$$

3: Let $i = i + 1$. Construct the iterative performance index function $V_i(x_k)$, which satisfies the following GHJB equation

$$V_i(x_k) = U(x_k, v_i(x_k)) + V_i(F(x_k, v_i(x_k)));$$

4: Update the iterative control law $v_{i+1}(x_k)$ by

$$v_{i+1}(x_k) = \arg\min_{u_k} \{U(x_k, u_k) + V_i(x_{k+1})\};$$

5: If $V_{i-1}(x_k) - V_i(x_k) < \varepsilon$, goto Step 7. Else goto Step 6;
6: If $i < i_{\max}$, then goto Step 3. Else, goto Step 8;
7: **return** $v_i(x_k)$ and $V_i(x_k)$. The optimal control law is achieved;
8: **return** The optimal control law is not achieved within $i_{\max}$ iterations.

According to Theorem 3.3, we can establish an effective iteration algorithm by repeating experiments using neural networks. The detailed implementation of the iteration algorithm is expressed in Algorithm 1.

Remark 3.3: We can see that the previously outlined training procedure can be easily implemented by computer program. Creating an action network, the weights of the critic networks *cnet*1 and *cnet*2 can be updated iteratively. If the weights of the critic network are convergent, then $\mu(x_k)$ must be an admissible control law. As the weights of actin network are chosen randomly, the admissibility of the control law is unknown before the weights of the critic networks are convergent. Hence, the iteration process of Algorithm 1 is implemented offline. For convenience of implementation, in this paper, we choose $\Psi(x_k) \equiv 0$ to obtain the initial admissible control law. The detailed training method of neural networks is shown in the following section.

D. Summary of the Policy Iteration Algorithm of ADP

According to the above preparations, we can summarize the discrete-time policy iteration algorithm in Algorithm 2.

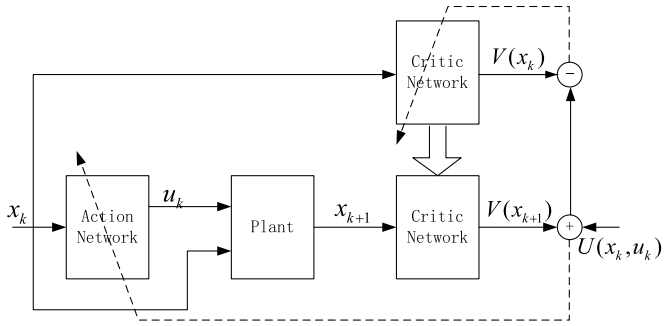

Fig. 1. Structure diagram of the algorithm.

IV. Neural Network Implementation for the Optimal Control Scheme

In this paper, BP neural networks are used to approximate $v_i(x_k)$ and $V_i(x_k)$, respectively. The number of hidden layer neurons is denoted by l, the weight matrix between the input layer and hidden layer is denoted by Y, and the weight matrix between the hidden layer and output layer is denoted by W. Then, the output of three-layer neural network is represented by

$$\hat{F}(X, Y, W) = W^T \sigma(Y^T X + b) \tag{45}$$

where $\sigma(Y^T X) \in R^l, [\sigma(z)]_i = ((e^{z_i} - e^{-z_i})/(e^{z_i} + e^{-z_i}))$, $i = 1, \ldots, l$ are the activation functions and b is the threshold value.

Here, there are two networks, which are critic network and action network, respectively. Both the neural networks are chosen as three-layer feedforward neural network. The whole structure diagram is shown in Fig. 1.

A. Critic Network

The critic network is used to approximate the performance index function $V_i(x_k)$. The output of the critic network is denoted as

$$\hat{V}_i^j(x_k) = W_{ci}^{jT} \sigma(Z_c(k)) \tag{46}$$

where $Z_c(k) = Y_c^T x_k + b_c$. The target function is expressed in (9). Then, we define the error function for the critic network as

$$e_{ci}^j(k) = \hat{V}_i^j(x_k) - V_i(x_k). \tag{47}$$

The objective function to be minimized in the critic network training is

$$E_{ci}^j(k) = \frac{1}{2}(e_{ci}^j(k))^2. \tag{48}$$

Therefore, the gradient-based weight update rule [24] for the critic network is given by

$$\begin{aligned} W_{ci}^{j+1}(k) &= W_{ci}^j(k) + \Delta W_{ci}^j(k) \\ &= W_{ci}^j(k) - \alpha_c \left[\frac{\partial E_{ci}^j(k)}{\partial \hat{V}_{i+1}^j(x_k)} \frac{\partial \hat{V}_{i+1}^j(x_k)}{\partial W_{ci}^j(k)} \right] \\ &= W_{ci}^j(k) - \alpha_c e_{ci}^j(k) \sigma(Z_c(k)) \end{aligned} \tag{49}$$

where $\alpha_c > 0$ is the learning rate of critic network. If the training precision is achieved, then we say that $V_{i+1}(x_k)$ can be approximated by the critic network.

There is an important property we should point out. Usually, the iterative performance index function $V_i(x_k)$ is difficult to obtain directly as the critic network is not trained yet. In this situation, there are two methods to obtain the target function of $V_i(x_k)$. First, define a new iteration index $l = 0, 1, \ldots$ and let the iterative performance index function be expressed as $V_{i,l}(x_k)$. Then, let $V_{i,l}(x_k)$ be updated by

$$V_{i,l+1}(x_k) = U(x_k, v_i(x_k)) + \hat{V}_{i,l}(x_{k+1}) \tag{50}$$

where $\hat{V}_{i,0}(x_{k+1}) = V_{i-1}(x_{k+1})$ is the output of the critic network in the previous iteration. The error function of the critic neural network can be expressed as

$$e_{ci,l}(k) = \hat{V}_{i,l+1}(x_k) - V_{i,l+1}(x_k). \tag{51}$$

Updating the neural networks according to (48) and (49) until $e_{ci,l}(k) \to 0$. Let $l = l + 1$ and repeat (50) and (51), until the following holds:

$$V_{i,l}(x_k) = U(x_k, v_i(x_k)) + V_{i,l}(x_{k+1}). \tag{52}$$

Then, we have $V_i(x_k) = V_{i,l}(x_k)$. This method is simple. However, usually, (52) holds for $l \to \infty$, i.e.,

$$V_{i,\infty}(x_k) = U(x_k, v_i(x_k)) + V_{i,\infty}(x_{k+1}). \tag{53}$$

Thus, the amount of computation is very large and it will take a long time to make the weights of the critic network converge.

The other method can be expressed as follows. According to (9), we have

$$V_i(x_k) = \sum_{j=0}^{N-1} U(x_{k+j}, v_i(x_{k+j})) + V_i(x_{k+N}). \tag{54}$$

As $v_i(x_k)$ is an admissible control law, we have $V_i(x_{k+N}) \to 0$ for $N \to \infty$. Thus, choose a large enough N and we have

$$V_i(x_k) \doteq \sum_{j=0}^{N-1} U(x_{k+j}, v_i(x_{k+j})). \tag{55}$$

Then, the error function (47) can be obtained. We can see that this method is much easier to apply. For convenience of computation, the second method is used in this paper.

B. Action Network

In the action network, the state error x_k is used as input to create the iterative control law as the output of the network. The output can be formulated as

$$\hat{v}_i^j(x_k) = W_{ai}^{jT} \sigma(Z_a(k)) \tag{56}$$

where $Z_a(k) = Y_a^T x_k + b_a$. The target of the output of the action network is given by (10). Thus, we can define the output error of the action network as

$$e_{ai}^j(k) = \hat{v}_i^j(x_k) - v_i(x_k). \tag{57}$$

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

$$E_{ai}^j(k) = \frac{1}{2}(e_{ai}^j(k))^T (e_{ai}^j(k)).$$

The weights updating algorithm is similar to the one for the critic network. By the gradient descent rule, we can obtain

$$\begin{aligned} W_{ai}^{j+1}(k) &= W_{ai}^{j}(k) + \Delta W_{ai}^{j}(k) \\ &= W_{ai}^{j}(k) - \beta_a \left[\frac{\partial E_{ai}^{j}(k)}{\partial e_{ai}^{j}(k)} \frac{\partial e_{ai}^{j}(k)}{\partial \hat{v}_i^{j}(k)} \frac{\partial \hat{v}_i^{j}(k)}{\partial W_{ai}^{j}(k)} \right] \\ &= W_{ai}^{j}(k) - \beta_a \sigma(Z_a(k))(e_{ai}^{j}(k))^T \end{aligned} \tag{58}$$

where $\beta_a > 0$ is the learning rate of action network. If the training precision is achieved, then we say that the iterative control law $v_i(x_k)$ can be approximated by the action network.

In this paper, to enhance the convergence speed of the neural networks, only one layer of neural network is updated during the training procedure. To guarantee the effectiveness of the neural network implementation, the convergence of the neural network weights is proven, which makes the iterative performance index function and iterative control be approximated by the critic and action networks, respectively. The weights convergence property of the neural networks is shown in the following theorem.

Theorem 4.1: Let the target performance index function and the target iterative control law be expressed by

$$V_{i+1}(x_k) = W_{ci}^{*T} \sigma(Z_c(k)) \tag{59}$$

and

$$v_i(x_k) = W_{ai}^{*T} \sigma(Z_a(k)) \tag{60}$$

respectively. Let the critic and action networks be trained by (49) and (58), respectively. If the learning rates α_c and β_a are both small enough, then we have the critic weights $W_{ci}(k)$ and action network weights $W_{ai}(k)$ are asymptotically convergent to the optimal weights $W_{ci}^{*}(k)$ and $W_{ai}^{*}(k)$, respectively.

Proof: Let $\bar{W}_{ci}^{j} = W_{ci}^{j} - W_{ci}^{*}$ and $\bar{W}_{ai}^{j} = W_{ai}^{j} - W_{ai}^{*}$. From (49) and (58), we have

$$\bar{W}_{ci}^{j+1}(k) = \bar{W}_{ci}^{j}(k) - \alpha_c e_{ci}^{j}(k) \sigma(Z_c(k))$$

and

$$\bar{W}_{ai}^{j+1}(k) = \bar{W}_{ai}^{j}(k) - \beta_a e_{ai}^{j}(k) \sigma(Z_a(k)).$$

Consider the following Lyapunov function candidate:

$$L(\bar{W}_{ci}^{j}, \bar{W}_{ai}^{j}) = \mathrm{tr}\left\{ \bar{W}_{ci}^{jT} \bar{W}_{ci}^{j} + \bar{W}_{ai}^{jT} \bar{W}_{ai}^{j} \right\}. \tag{61}$$

Then, the difference of the Lyapunov function candidate (61) is given by

$$\begin{aligned} \Delta L(\bar{W}_{ci}^{j}, \bar{W}_{ai}^{j}) &= \mathrm{tr}\left\{ \bar{W}_{ci}^{(j+1)T} \bar{W}_{ci}^{j+1} + \bar{W}_{ai}^{(j+1)T} \bar{W}_{ai}^{j+1} \right\} \\ &\quad - \mathrm{tr}\left\{ \bar{W}_{ci}^{jT} \bar{W}_{ci}^{j} + \bar{W}_{ai}^{jT} \bar{W}_{ai}^{j} \right\} \\ &= \alpha_c \left\| e_{ci}^{j}(k) \right\|^2 \left(-2 + \alpha_c \| \sigma(Z_c(k)) \|^2 \right) \\ &\quad + \beta_a \left\| e_{ai}^{j}(k) \right\|^2 \left(-2 + \beta_a \| \sigma(Z_a(k)) \|^2 \right). \end{aligned}$$

According to the definition of $\sigma(\cdot)$ in (45), we know that $\|\sigma(Z_c(k))\|^2$ and $\|\sigma(Z_a(k))\|^2$ are both finite for $\forall Z_c(k), Z_a(k)$. Thus, if α_c and β_a are both small enough that satisfy $\alpha_c \le 2/\|\sigma(Z_c(k))\|^2$ and $\beta_a \le 2/\|\sigma(Z_a(k))\|^2$, then we have $\Delta L(\bar{W}_{ci}^{j}, \bar{W}_{ai}^{j}) < 0$. The proof is completed. ■

V. Numerical Analysis

To evaluate the performance of our policy iteration algorithm, four examples have been chosen: 1) a linear system; 2) a discrete-time nonlinear system; 3) a torsional pendulum system; and 4) a complex nonlinear system to show its broad applicability.

Example 1: In the first example, a linear system is considered. The results will be compared with the traditional linear quadratic regulation method to verify the effectiveness of the developed algorithm. We consider the following linear system:

$$x_{k+1} = Ax_k + Bu_k \tag{62}$$

where $x_k = [x_{1k}, x_{2k}]^T$ and $u_k \in \mathbb{R}^1$. Let the system matrices be expressed as

$$A = \begin{bmatrix} 0 & 0.1 \\ 0.3 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}.$$

The initial state is $x_0 = [1, -1]^T$. Let the performance index function be expressed by (2). The utility function is the quadratic form that is expressed as $U(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k$, where $Q = I$, $R = 0.5I$, and I is the identity matrix with suitable dimensions.

Neural networks are used to implement the developed policy iteration algorithm. The critic network and the action network are chosen as three-layer BP neural networks with the structures of 2–8–1 and 2–8–1, respectively. For each iteration step, the critic network and the action network are trained for 80 steps using the learning rate of $\alpha = 0.02$ so that the neural network training error becomes less than 10^{-5}. The initial admissible control law is obtained by Algorithm 1, where the weights and thresholds of action network are obtained as

$$Y_{a,\text{initial}} = \begin{bmatrix} -4.1525 & -1.1980 \\ 0.3693 & -0.8828 \\ 1.8071 & 2.8088 \\ 0.4104 & -0.9845 \\ 0.7319 & -1.7384 \\ 1.2885 & -2.5911 \\ -0.3403 & 0.8154 \\ -0.5647 & 1.3694 \end{bmatrix}$$

$$W_{a,\text{initial}} = [-0.0010, -0.2566, 0.0001, -0.1409, -0.0092, 0.0001, 0.3738, 0.0998]$$

$$b_{a,\text{initial}} = [3.5272, -0.9609, -1.8038, -0.0970, 0.8526, 1.1966, -1.0948, 2.5641]^T$$

respectively. Implement the policy iteration algorithm for six iterations to reach the computation precision $\varepsilon = 10^{-5}$, and the convergence trajectory of the iterative performance index functions is shown in Fig. 2(a). During each iteration, the iterative control law is updated. Applying the iterative control law to the given system (62) for $T_f = 15$ time steps, we can obtain the iterative states and iterative controls, which are shown in Fig. 2(b) and (c), respectively.

We can see that the optimal control law of (62) is obtained after six iterations. On the other hand, it is known that the solution of the optimal control problem for the linear system

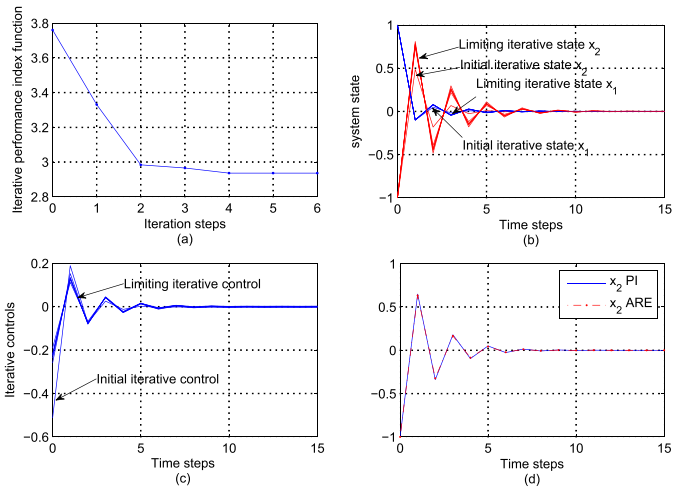

Fig. 2. Numerical results of Example 1. (a) Convergence trajectory of iterative performance index function. (b) Iterative states. (c) Iterative controls. (d) Trajectories of x_2 under the optimal control of policy iteration and ARE.

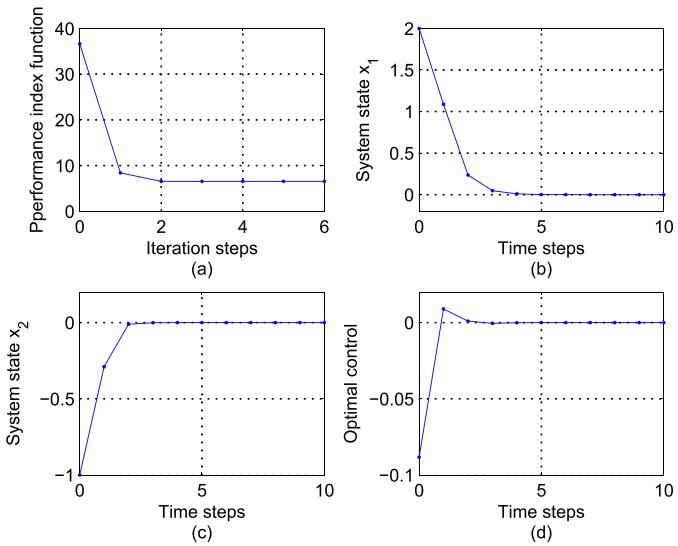

Fig. 3. Numerical results using policy iteration algorithm. (a) Convergence trajectory of iterative performance index function. (b) Optimal trajectory of state x_1. (c) Optimal trajectory of state x_2. (d) Optimal control.

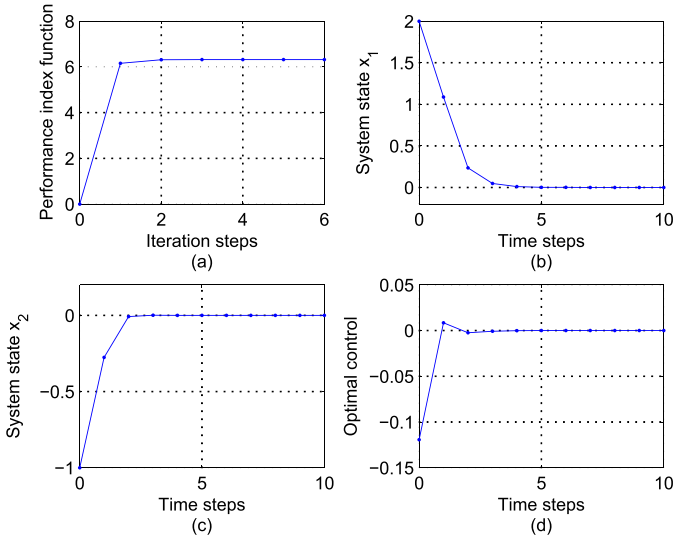

Fig. 4. Numerical results using value iteration algorithm. (a) Convergence trajectory of iterative performance index function. (b) Optimal trajectory of state x_1. (c) Optimal trajectory of state x_2. (d) Optimal control.

is quadratic in the state and given as $J^*(x_k) = x_k^T P x_k$, where P is the solution of the ARE. The solution for the ARE for the given linear system (62) is

$$P = \begin{bmatrix} 1.091 & -0.309 \\ -0.309 & 2.055 \end{bmatrix}.$$

The optimal control can be obtained as $u^*(x_k) = [-0.304, 1.029]x_k$. Applying the optimal control law to the linear system (62), we can obtain the same optimal control results. The optimal trajectories of system state x_2 under the optimal control laws by policy iteration and ARE are shown in Fig. 2(d).

Example 2: The second example is chosen from the example in [32], [39], and [40]. We consider the following nonlinear system:

$$x_{k+1} = f(x_k) + g(x_k)u_k \tag{63}$$

where $x_k = [x_{1k} \quad x_{2k}]^T$ and u_k are the state and control variables, respectively. The system functions are given as

$$f(x_k) = \begin{bmatrix} 0.2x_{1k}\exp(x_{2k}^2) \\ 0.3x_{2k}^3 \end{bmatrix}, \quad g(x_k) = \begin{bmatrix} 0 \\ -0.2 \end{bmatrix}.$$

The initial state is $x_0 = [2, -1]^T$. In this example, two utility functions, which are quadratic and nonquadratic forms, will be considered, respectively. The first utility function is a quadratic form, which is the same as the one in Example 1 where the matrix $Q = R = I$. The configurations of the critic network and the action network are chosen the same as the ones in Example 1. For each iteration step, the critic network and the action network are trained for 100 steps using the learning rate of $\alpha = 0.02$ so that the neural network training error becomes less than 10^{-5}. Implement the policy iteration algorithm for six iterations to reach the computation precision 10^{-5}. The convergence trajectories of the iterative performance index functions are shown in Fig. 3(a). Applying the optimal control law to the given system (63) for $T_f = 10$ time steps, we can obtain the iterative states trajectories and control, which are shown in Fig. 3(b)–(d), respectively.

In [39], it is proven that the optimal control law can be obtained by value iteration algorithm (39). The convergence trajectory of the iterative performance index function is shown in Fig. 4(a). The optimal trajectories of system states and control are shown in Fig. 4(b)–(d), respectively.

Next, we change the utility function into a nonquadratic form in [34] with modifications, where the utility function is expressed as

$$U(x_k, u_k) = \ln(x_k^T Q x_k + 1) + \ln(x_k^T Q x_k + 1)u_k^T R u_k. \tag{64}$$

Let the other parameters remain unchanged. Using the developed policy iteration algorithm, we can also obtain the optimal control law of the system. The performance index function is shown in Fig. 5(a). The optimal trajectories of iterative states and controls are shown in Fig. 5(b)–(d), respectively.

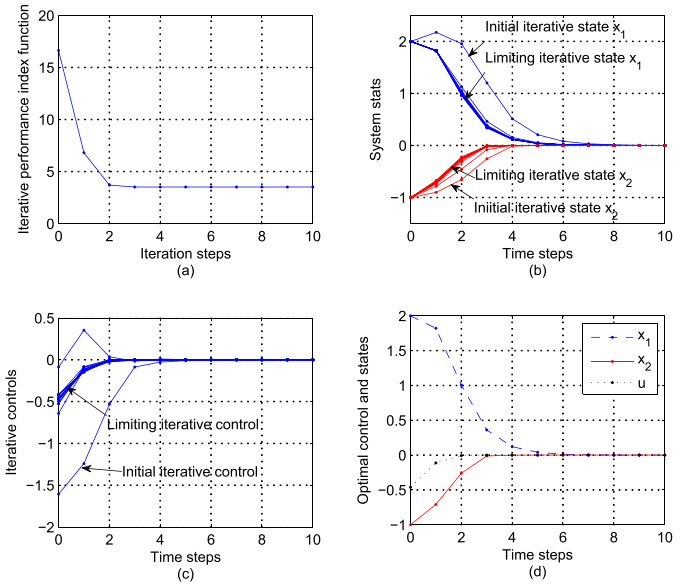

Fig. 5. Numerical results for nonquadratic utility function using policy iteration algorithm. (a) Convergence trajectory of iterative performance index function. (b) Iterative states. (c) Iterative controls. (d) Optimal states and control.

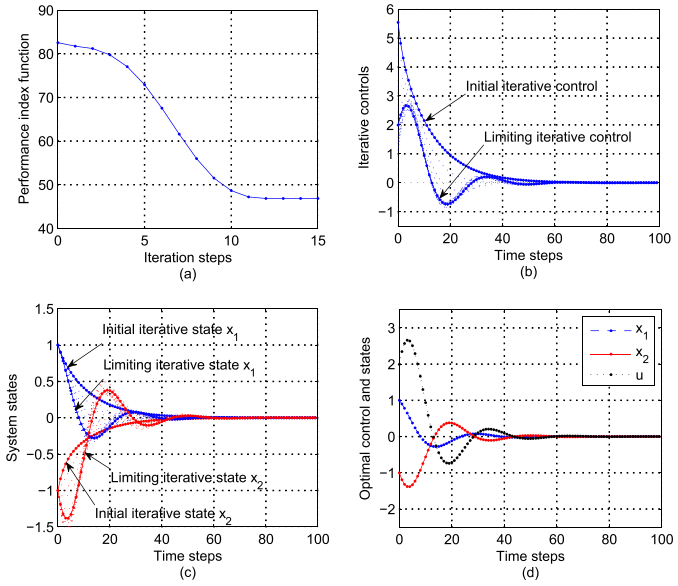

Fig. 6. Numerical results of Example 3 using policy iteration algorithm. (a) Convergence trajectory of iterative performance index function. (b) Iterative controls. (c) Iterative states. (d) Optimal states and control.

Remark 5.1: From the numerical results, we can see that for quadratic and nonquadratic utility functions, the optimal control law of the nonlinear system can both be effectively obtained, respectively. On the other hand, We have shown that using the value iteration algorithm in [39], we can also obtain the optimal control law of the system. However, we should point out that the convergence properties of the iterative performance index functions by the policy and value iteration algorithm are obviously different. This makes the stability of the control system under the iterative control law quite different. In the following example, detailed comparisons will be displayed.

Example 3: We now examine the performance of the developed algorithm in a torsional pendulum system [24]. The dynamics of the pendulum is as follows:

$$\begin{cases} \dfrac{d\theta}{dt} = \omega \\ J\dfrac{d\omega}{dt} = u - Mgl\sin\theta - f_d\dfrac{d\theta}{dt} \end{cases} \tag{65}$$

where $M = 1/3$ kg and $l = 2/3$ m are the mass and length of the pendulum bar, respectively. The system states are the current angle θ and the angular velocity ω. Let $J = 4/3Ml^2$ and $f_d = 0.2$ be the rotary inertia and frictional factor, respectively. Let $g = 9.8$ m/s^2 be the gravity. Discretization of the system function and performance index function using Euler and trapezoidal methods [48] with the sampling interval $\Delta t = 0.1$s leads to

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \end{bmatrix} = \begin{bmatrix} 0.1x_{2k} + x_{1k} \\ -0.49 \times \sin(x_{1k}) - 0.1 \times f_d \times x_{2k} + x_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u_k \tag{66}$$

where $x_{1k} = \theta_k$ and $x_{2k} = \omega_k$. Let the initial state be $x_0 = [1, -1]^T$. Let the utility function be the quadratic form and the structures of the critic and action networks are 2–12–1 and 2–12–1. The initial admissible control law is obtained by Algorithm 1, where the weight matrices are obtained as

$$Y_{a,\text{initial}} = \begin{bmatrix} -0.6574 & 1.1803 \\ 1.5421 & -2.9447 \\ -4.3289 & -3.7448 \\ 5.7354 & 2.8933 \\ 0.4354 & -0.8078 \\ -1.9680 & 3.6870 \\ 1.9285 & 1.4044 \\ -4.9011 & -4.3527 \\ 1.0914 & -0.0344 \\ -1.5746 & 2.8033 \\ 1.4897 & -0.0315 \\ 0.2992 & -0.0784 \end{bmatrix}$$

$$W_{a,\text{initial}} = [-2.1429,\ 1.9276,\ 0.0060,\ 0.0030,\ 4.5618,\ 3.2266,\ 0.0005,\ -0.0012,\ 1.3796\ -0.5338,\ 0.5043,\ -5.1110]$$

$$b_{a,\text{initial}} = [0.8511,\ 4.2189,\ 5.7266,\ -6.0599,\ 0.4998,\ -5.7323,\ -0.6220,\ -0.5142,\ -0.3874,\ 3.3985,\ 0.6668,\ -0.2834]^T.$$

For each iteration step, the critic and action networks are trained for 400 steps using the learning rate of $\alpha = 0.02$ to reach 10^{-5}. Implement the policy iteration algorithm for 16 iterations to reach the computation precision 10^{-5}, and the convergence trajectory of the iterative performance index function is shown in Fig. 6(a). Apply the iterative control laws to the given system for $T_f = 100$ time steps and the trajectories of the iterative control and states are shown in Fig. 6(b) and (c), respectively. The optimal trajectories of system states and control are shown in Fig. 6(d).

From the numerical results, we can see that using the developed policy iteration algorithm, any of the iterative control law

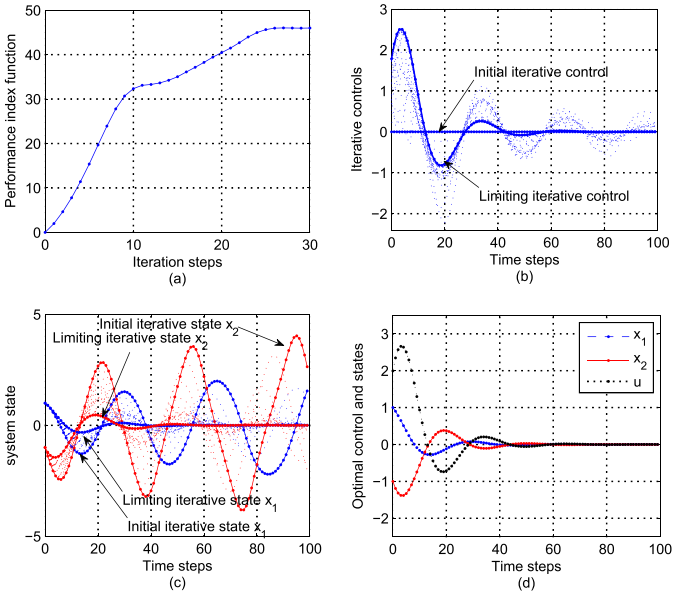

Fig. 7. Numerical results of Example 3 using value iteration algorithm. (a) Convergence trajectory of iterative performance index function. (b) Iterative controls. (c) Iterative states. (d) Optimal states and control.

stabilizes the system. However, for the value iteration algorithm, the situation is quite different. For the value iteration algorithm, the initial performance index function $V_0(x_k) \equiv 0$. Run the value iteration algorithm (39) for 30 iterations to reach the computation precision 10^{-5} and trajectory of the performance index function is shown in Fig. 7(a). Applying the iterative control law to the given system (66) for $T_f = 100$ time steps, we can obtain the iterative states and iterative controls, which are shown in Fig. 7(b) and (c), respectively. The optimal trajectories of control and system states are shown in Fig. 7(d).

For unstable control systems, although the optimal control law can be obtained by value and policy iteration algorithms, for value iteration algorithm, however, we can see that not all the controls can stabilize the control system. Moreover, the properties of the iterative controls obtained by value iteration algorithm cannot be analyzed and this makes the value iteration algorithm only be implemented offline. For the developed policy iteration algorithm, the stability property can be guaranteed. Hence, we can declare the effectiveness of the policy iteration algorithm in this paper.

Example 4: As a real world application of the developed method, the problem of nonlinear satellite attitude control has been selected [49], [50]. Satellite dynamics are represented as

$$\frac{d\omega}{dt} = \Upsilon^{-1}\left(N_{\text{net}} - \omega \times \Upsilon\omega\right) \tag{67}$$

where Υ, ω, and N_{net} are the inertia tensor, angular velocity vector of the body frame with respect to inertial frame, and the vector of the total torque applied on the satellite, respectively. The selected satellite is an inertial pointing satellite. Hence, we are interested in its attitude with respect to the inertial frame. All the vectors are represented in the body frame and the sign $\times$ denotes the cross product of two vectors. Let $N_{\text{net}} = N_{\text{ctrl}} + N_{\text{dis}}$, where N_{ctrl} is the control and N_{dis} is the disturbance. Following [50] and its order of transformation, the kinematic

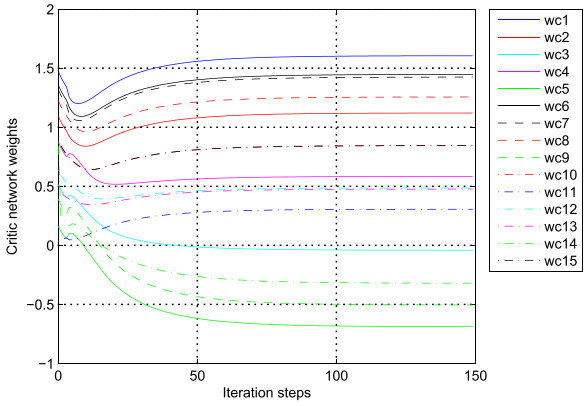

Fig. 8. Convergence trajectories of the critic network.

equation of the satellite is

$$\frac{d}{dt}\begin{bmatrix}\phi\\ \theta\\ \Psi\end{bmatrix} = \begin{bmatrix}1 & \sin(\phi)\tan(\phi) & \cos(\phi)\tan(\theta)\\ 0 & \cos(\phi) & -\sin(\phi)\\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta)\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{68}$$

where ϕ, θ, and Ψ are the three Euler angles describing the attitude of the satellite with respect to x-, y-, and z-axes of the inertial coordinate system, respectively. Subscripts x, y, and z are the corresponding elements of the angular velocity vector ω. The three Euler angles and the three elements of the angular velocity form the elements of the state space for the satellite attitude control problem and form the following state equation:

$$\dot{x} = f(x) + g(x)u \tag{69}$$

where

$$x = [\,\phi\ \theta\ \Psi\ \omega_x\ \omega_y\ \omega_z\,]^T, \quad u = \begin{bmatrix}u_x\ u_y\ u_z\end{bmatrix}$$
$$f(x) = \begin{bmatrix}M_{3\times 1}\\ \Upsilon^{-1}(N_{\text{dis}} - \omega \times \Upsilon\omega)\end{bmatrix}, \quad g(x) = \begin{bmatrix}0_{3\times 3}\\ \Upsilon^{-1}\end{bmatrix} \tag{70}$$

and $M_{3\times 1}$ denotes the right-hand side of (68). The three-by-three null matrix is denoted by $0_{3\times 3}$. The moment of inertia matrix of the satellite is chosen as

$$\Upsilon = \begin{bmatrix}100 & 2 & 0.5\\ 2 & 100 & 1\\ 0.5 & 1 & 110\end{bmatrix}\text{kg}\cdot\text{m}^2. \tag{71}$$

The initial states are 60°, 20°, and 70° for the Euler angles of ϕ, θ, and Ψ, respectively, and -0.001, 0.001, and 0.001 rad/s for the angular rates around x-, y-, and z-axes, respectively. For convenience of analysis, we assume that there is no disturbance in the system. Let the sampling time is 0.25 s using the Euler method. Let the utility function be quadratic form where the state and control weight matrices are selected as $Q = \text{diag}(0.25,\ 0.25,\ 0.25,\ 25,\ 25,\ 25)$ and $R = \text{diag}(25,\ 25,\ 25)$, respectively.

Neural networks are used to implement the developed policy iteration algorithm. The critic and action networks are chosen as three-layer BP neural networks with the structures of 6–15–1 and 6–15–3, respectively. For each iteration step, the critic and action networks are trained for 800 steps using the learning rate of $\alpha = 0.02$ so that the neural network training error becomes less than 10^{-5}. Implement the policy iteration

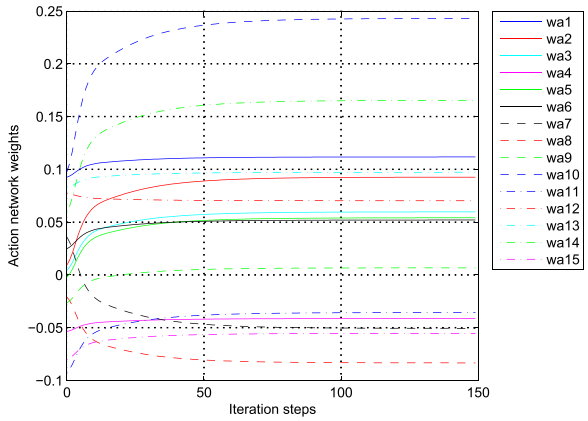

Fig. 9. Convergence trajectories of the first column of weights of the action network.

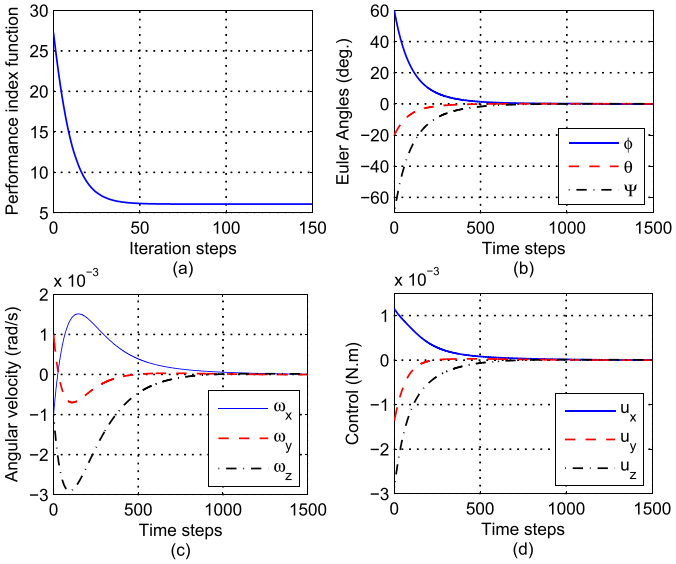

Fig. 10. Numerical results of Example 4 using policy iteration algorithm. (a) Convergence trajectory of iterative performance index function. (b) Trajectories of angular velocities ϕ, θ, and Ψ. (c) Trajectories of angular velocities ω_x, ω_y, and ω_z. (d) Optimal control trajectories.

algorithm for 150 iterations. We have proven that the weights of critic and action networks are convergent in each iteration and thus convergent to the optimal ones. The convergence trajectories of critic network weights are shown Fig. 8. The weight convergence trajectories of the first column of action network are shown in Fig. 9.

The iterative performance index functions are shown in Fig. 10(a). After the weights of the critic and action networks are convergent, we apply the neuro-optimal controller to the given system (62) for $T_f = 1500$ time steps. The optimal state trajectories of ϕ, θ, and Ψ are shown in Fig. 10(b). The trajectories of angular velocities ω_x, ω_y, and ω_z are shown in Fig. 10(c), and the optimal control trajectories are shown in Fig. 10(d). The numerical results illustrate the effectiveness of the developed policy iteration algorithm.

VI. Conclusion

In this paper, an effective policy iteration ADP algorithm is developed to find the infinite horizon optimal control for discrete-time nonlinear systems. It is shown that any of the iterative control law can stabilize the control system. It has been proven that the iterative performance index functions are monotonically nonincreasing and convergent to the optimal solution of the HJB equation. Neural networks are used to approximate the performance index function and compute the optimal control policy, respectively, for facilitating the implementation of the iterative ADP algorithm. Finally, four numerical examples are given to illustrate the performance of the developed method.

On the other hand, for the discrete-time policy iteration algorithm, some further properties can be discussed. In this paper, we assume that the iterative control law $v_i(x_k)$ and the iterative performance index function $V_i(x_k)$ can be well approximated by neural networks. Since neural networks always have approximation errors, generally, the exact $v_i(x_k)$ and $V_i(x_k)$ cannot be achieved. In the case of approximation errors, we should say that the convergence property of the iterative performance index functions and the stability of the system under the iterative control law are not proven. Additional convergence and stability criterions should be established. The convergence and stability properties of the discrete-time policy iteration algorithm with approximation errors will be discussed in our future work.

References

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

[2] P. J. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence," *General Syst. Yearbook*, vol. 22, pp. 25–38, Jan. 1977.

[3] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control*, W. T. Miller, R. S. Sutton, and P. J. Werbos, Ed., Cambridge, MA, USA: MIT Press, 1991, pp. 67–95.

[4] T. Dierks and S. Jagannathan, "Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update," *IEEE Trans. Neurl Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1118–1129, Jul. 2012.

[5] M. Fairbank, E. Alonso, and D. Prokhorov, "Simple and fast calculation of the second-order gradients for globalized dual heuristic dynamic programming in neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1671–1676, Oct. 2012.

[6] Y. Jiang and Z. P. Jiang, "Robust adaptive dynamic programming with an application to power systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1150–1156, Jul. 2013.

[7] 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.

[8] Z. Ni, H. He, and J. Wen, "Adaptive learning in tracking control based on the dual critic network design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 913–928, Jun. 2013.

[9] D. Nodland, H. Zargarzadeh, and S. Jagannathan, "Neural network-based optimal adaptive output feedback control of a helicopter UAV," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1061–1073, Jul. 2013.

[10] D. Wang, D. Liu, D. Zhao, Y. Huang, and D. Zhang, "A neural-network-based iterative GDHP approach for solving a class of nonlinear optimal control problems with control constraints," *Neural Comput. Appl.*, vol. 22, no. 2, pp. 219–227, Feb. 2013.

[11] H. Xu, S. Jagannathan, and F. L. Lewis, "Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses," *Automatica*, vol. 48, no. 6, pp. 1017–1030, Jun. 2012.

[12] H. Xu and S. Jagannathan, "Stochastic optimal controller design for uncertain nonlinear networked control system via neuro dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 471–484, Mar. 2013.

[13] J. Liang, G. K. Venayagamoorthy, and R. G. Harley, "Wide-area measurement based dynamic stochastic optimal power flow control for smart grids with high variability and uncertainty," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 59–69, Mar. 2012.

[14] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.

[15] X. Xu, Z. Hou, C. Lian, and H. He, "Online learning control using adaptive critic designs with sparse kernel machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 762–775, May 2013.

[16] D. Liu, D. Wang, D. Zhao, Q. Wei, and 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.*, vol. 9, no. 3, pp. 628–634, Jul. 2012.

[17] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Part C, Appl, Rev.*, vol. 32, no. 2, pp. 140–153, May 2002.

[18] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, Aug. 2012.

[19] Q. Wei and D. Liu, "An iterative ϵ-optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state," *Neural Netw.*, vol. 32, pp. 236–244, Aug. 2012.

[20] Y. Jiang and Z. P. Jiang, "Approximate dynamic programming for optimal stationary control with control-dependent noise," *IEEE Trans. Neurl Netw.*, vol. 22, no. 12, pp. 2392–2398, Dec. 2011.

[21] 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, Ed., New York, NY, USA: Van Nostrand Reinhold, 1992, ch. 13.

[22] R. Enns and J. Si, "Helicopter trimming and tracking control using direct neural dynamic programming," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 929–939, Aug. 2003.

[23] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA, USA: Athena Scientific, 1996.

[24] 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.

[25] M. Geist and O. Pietquin, "Algorithmic survey of parametric value function approximation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 845–867, Jun. 2013.

[26] K. S. Hwang and C. Y. Lo, "Policy improvement by a model-free Dyna architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 776–788, May 2013.

[27] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Dept. Comput. Sci., Cambridge Univ., Cambridge, U.K., 1989.

[28] T. Huang and D. Liu, "A self-learning scheme for residential energy system control and management," *Neural Comput. Appl.*, vol. 22, no. 2, pp. 259–269, Feb. 2013.

[29] H. Li and D. Liu, "Optimal control for discrete-time affine nonlinear systems using general value iteration," *IET Control Theory Appl.*, vol. 6, no. 18, pp. 2725–2736, Dec. 2012.

[30] D. Liu and Q. Wei, "Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 779–789, Apr. 2013.

[31] H. Modares, F. L. Lewis, and M. B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, Aug. 2013, to be published.

[32] F. Wang, N. Jin, D. Liu, and Q. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ϵ-error bound," *J. Control Theory Appl.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.

[33] 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.

[34] Q. Wei, H. Zhang, and J. Dai, "Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1839–1848, 2009.

[35] H. Zhang, R. Song, Q. Wei, and T. Zhang, "Optimal tracking control for a class of nonlinear discrete-time systems with time delays based on heuristic dynamic programming," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1851–1862, Dec. 2011.

[36] 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.

[37] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[38] R. Beard, "Improving the closed-loop performance of nonlinear systems," Ph.D. dissertation, Rensselaer Polytechnic Inst., Troy, NY, USA, 1995.

[39] 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., Part B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[40] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. Syst., Man, Cybern., Part B, Cybern.*, vol. 38, no. 4, pp. 937–942, Jul. 2008.

[41] D. Liu, H. Li, and D. Wang, "Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm," *Neurocomputing*, vol. 110, pp. 92–100, Jun. 2013.

[42] D. Liu, D. Wang, and X. Yang, "An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs," *Inf. Sci.*, vol. 220, pp. 331–342, Jan. 2013.

[43] 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.

[44] H. Zhang, Q. Wei, and D. Liu, "An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games," *Automatica*, vol. 47, no. 1, pp. 207–214, Jan. 2011.

[45] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598–1611, 2012.

[46] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, Jan. 2013.

[47] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[48] S. K. Gupta, *Numerical Methods for Engineerings*. New Delhi, India: New Age Int. Company, 1995.

[49] A. Heydari and S. N. Balakrishnan, "Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 145–157, Jan. 2013.

[50] J. R. Wertz, *Spacecraft Attitude Determination and Control*. Amsterdam, The Netherlands: Kluwer, 1978, pp. 521–570.

Derong Liu (S'91–M'94–SM'96–F'05) received the Ph.D. degree in electrical engineering from the University of Notre Dame, South Bend, IN, USA, in 1994.

He was a Staff Fellow with the General Motors Research and Development Center, Warren, MI, USA, from 1993 to 1995. He was an Assistant Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, from 1995 to 1999. He joined the University of Illinois at Chicago, Chicago, IL, USA, 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 (six research monographs and eight edited volumes).

Dr. Liu is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He received the Michael J. Birck Fellowship from the University of Notre Dame in 1990, the Harvey N. Davis Distinguished Teaching Award from Stevens Institute of Technology in 1997, the Faculty Early Career Development CAREER Award from the National Science Foundation in 1999, the University Scholar Award from University of Illinois from 2006 to 2009, and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China in 2008. He is a fellow of the INNS.

Qinglai Wei (M'11) received the B.S. degree in automation, the M.S. degree in control theory and control engineering, and the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2002, 2005, and 2008, respectively.

He was a Post-Doctoral Fellow with Institute of Automation, Chinese Academy of Sciences, Beijing, China, from 2009 to 2011. He is currently an Associate Professor with The State Key Laboratory of Management and Control for Complex Systems. His current research interests include neural networks-based control, adaptive dynamic programming, optimal control, nonlinear system, and their industrial applications.