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

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

***Abstract*—This paper is concerned with a novel generalized policy iteration algorithm for solving optimal control problems for discrete-time nonlinear systems. The idea is to use an iterative adaptive dynamic programming algorithm to obtain iterative control laws which make the iterative value functions converge to the optimum. Initialized by an admissible control law, it is shown that the iterative value functions are monotonically nonincreasing and converge to the optimal solution of Hamilton–Jacobi–Bellman equation, under the assumption that a perfect function approximation is employed. The admissibility property is analyzed, which shows that any of the iterative control laws can stabilize the nonlinear system. Neural networks are utilized to implement the generalized policy iteration algorithm, by approximating the iterative value function and computing the iterative control law, respectively, to achieve approximate optimal control. Finally, numerical examples are presented to verify the effectiveness of the present generalized policy iteration algorithm.**

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

I. Introduction

REINFORCEMENT learning, one of the most active research areas in artificial intelligence, is a computational approach to learning whereby an agent tries to optimize the total amount of reward it receives when interacting with its environment [1]–[4]. Associated with reinforcement learning methods and optimal control, adaptive dynamic programming (ADP), proposed by Werbos [5], [6], overcomes the curse of dimensionality problem in dynamic programming (DP) by approximating the performance index function forward-in-time and becomes an important brain-like intelligent method of approximate optimal control for nonlinear systems [7]–[15]. In [16] and [17], ADP approaches were classified into several main schemes which were heuristic DP (HDP), dual HDP (DHP), globalized DHP (GDHP), and their action-dependent versions.

Iterative methods are primary tools in ADP to obtain the approximate solution of the Hamilton–Jacobi–Bellman (HJB) equation and have received more and more attention [18]–[25]. The previous iterative ADP algorithms can be classified into two main schemes which are based on value and policy iterations, respectively [26], [27]. The value iteration algorithm for optimal control of nonlinear systems was first given in [27] and [28]. In [29], the convergence of discrete-time value iteration algorithm was proven. In [30], the value iteration algorithm was applied to solve optimal tracking control problems for nonlinear systems. In [31], the value iteration algorithm was applied by DHP. In [32], the value iteration ADP is implemented by GDHP. Policy iteration algorithms for optimal control of continuous-time systems were given in [33] and [34]. In [35], the policy iteration algorithm was successfully applied to solve continuous-time complex-valued systems. In [36], a discrete-time policy iteration was developed with convergence and stability proofs. Based on the framework of value and policy iteration algorithms, many investigations of iterative ADP algorithms have been developed, such as iterative θ-ADP algorithm [37], [38], ε-optimal control [39], [40], ADP with constraints [41]–[43], zero-sum and nonzero-sum games [44]–[48], finite-approximation-error-based ADP [49]–[52], ADP with unknown and partially-unknown systems [53]–[55], online ADP [56], [57], multiagent optimal control [58], [59], integral reinforcement learning [60], [61], and dual critic network design [62].

In [4], a generalized policy iteration algorithm, which contained policy iteration and value iteration as special cases, was constructed as a new iterative ADP algorithm to solve optimal control problems. Generalized policy iteration algorithms for continuous-time systems were studied in [63] and [64]. The stability and convergence properties of continuous-time generalized policy iteration algorithms were analyzed in [65]. The sketch of the generalized policy iteration algorithm for discrete-time nonlinear systems was described in [4] and [26], respectively. In [4], it was pointed out that most of the discrete-time reinforcement learning methods could be described as generalized policy iteration algorithms. Hence, the investigations of the generalized policy iteration algorithms are important for the development of ADP. However, the generalized

Manuscript received February 9, 2014; revised August 19, 2014 and November 26, 2014; accepted March 14, 2015. Date of publication May 20, 2015; date of current version November 13, 2015. This work was supported in part by the National Natural Science Foundation of China under Grants 61034002, 61233001, 61273140, 61304086, and 61374105, in part by the Beijing Natural Science Foundation under Grant 4132078, and in part by the Early Career Development Award of the State Key Laboratory of Management and Control for Complex Systems. This paper was recommended by Associate Editor A. H. Tan.

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 (e-mail: derong.liu@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/TSMC.2015.2417510

policy iteration algorithms have inherent differences from the value and policy iteration algorithms. This makes the property analysis of value and policy iteration algorithms invalid for generalized policy iteration algorithms. Till now, the discussions on the properties of the generalized policy iteration algorithms for discrete-time control systems were very scarce. To the best of the authors' knowledge, only in [66], the properties of the generalized policy iteration algorithms were analyzed, while the stability property of the system under the iterative control law in [66] cannot be guaranteed. Hence, it is important to establish a new generalized policy iteration algorithm with new analysis methods. This motivates our research.

In this paper, inspired by [4], [26], and [66], a generalized policy iteration algorithm is developed to solve the approximate optimal control problems of discrete-time nonlinear systems. First, the detailed iteration procedure of the generalized policy iteration algorithm for discrete-time nonlinear systems is presented. Second, the properties of the generalized policy iteration algorithm are developed. Initialized by an arbitrary admissible control law, it proves that the general framework of the generalized policy iteration algorithm will converge to the optimal performance index function and the optimal control law, under the strictly-hypothetical assumption that a perfect function approximation is available. It shows that the iterative value function is monotonically nonincreasing and converges to the optimal performance index function. We emphasize that any of the iterative control laws is proven to stabilize the nonlinear systems. Next, some effective methods are developed to overcome the difficulties of obtaining the initial conditions for the generalized policy iteration algorithm, which make the present generalized policy iteration algorithm more suitable in applications. Neural networks are used to make an approximation implementation of the generalized policy iteration algorithm, where the approximate optimal performance index function and control law are obtained. Simulation results will illustrate the effectiveness of the present algorithm.

This paper is organized as follows. In Section II, preliminaries and assumptions of the generalized policy iteration algorithm are presented. In Section III, the monotonicity and convergence properties of the iterative value function of the generalized policy iteration algorithm are developed. The admissibility property of the iterative control laws is also analyzed in this section. In Section IV, the neural network implementation for the generalized policy iteration algorithm is discussed. In Section V, simulation results and comparisons are given to demonstrate the effectiveness of the present algorithm. Finally, in Section VI, the conclusion is drawn.

II. Preliminaries and Assumptions

In this paper, we consider a class of discrete-time nonlinear systems described by

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

where $x_k \in \mathbb{R}^n$ is the state vector and $u_k \in \mathbb{R}^m$ is the control vector. Let x_0 be the given initial state and let $F(x_k, u_k)$ denote the system function, which is known. For any $k = 0, 1, \ldots$, 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 $U(x_k, u_k) > 0$, $\forall\, x_k, u_k \neq 0$, is the utility function.

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

Assumption 1: The system (1) is controllable on a compact set $\Omega_x \subset \mathbb{R}^n$ containing the origin, and the function $F(x_k, u_k)$ is Lipschitz continuous on Ω_x.

Assumption 2: The system state $x_k = 0$ is an equilibrium state of system (1) under the control $u_k = 0$, i.e., $F(0, 0) = 0$.

Assumption 3: The feedback control $u(x_k)$ satisfies $u(x_k) = 0$ for $x_k = 0$.

Assumption 4: The utility function $U(x_k, u_k)$ is a continuous positive definite function of x_k and u_k.

For a given control law μ, the map from initial state to the value of $\sum_{k=0}^{\infty} U(x_k, \mu(x_k))$ is called a performance index function $J^{\mu}(x_0)$. The optimal performance index function is denoted by

$$J^*(x_0) = \inf_{\mu} J^{\mu}(x_0). \tag{3}$$

According to Bellman's principle of optimality, for all $x_k \in \Omega_x$, $J^*(x_k)$ satisfies the discrete-time HJB equation

$$\begin{aligned} J^*(x_k) &= \inf_{u_k}\{U(x_k, u_k) + J^*(x_{k+1})\} \\ &= \inf_{u_k}\{U(x_k, u_k) + J^*(F(x_k, u_k))\}. \end{aligned} \tag{4}$$

Define the optimal control law as $u^*(x_k) = \arg\inf_{u_k}\{U(x_k, u_k) + J^*(F(x_k, u_k))\}$. Then for all $x_k \in \Omega_x$, the HJB equation can be written as

$$J^*(x_k) = U(x_k, u^*(x_k)) + J^*(F(x_k, u^*(x_k))). \tag{5}$$

Remark 1: Generally, the optimal performance index function $J^*(x_k)$ is a nonanalytical nonlinear function. It is nearly impossible to obtain $J^*(x_k)$ for all $x_k \in \Omega_x$ by solving the HJB equation. To overcome this problem, a new generalized policy iteration-based ADP algorithm is developed to obtain the approximate solution of the HJB equation iteratively and the neural network implementation of the present algorithm will be given.

III. Generalized Policy Iteration Algorithm Starting With Admissible Control Law

In this section, a generalized policy iteration algorithm is developed to obtain the optimal control law for discrete-time nonlinear systems. The present generalized policy iteration algorithm starts with an admissible control law, which makes it different from [66]. Both algorithms involve updating the value functions and control laws. The present algorithm guarantees all control laws from the iterative process are admissible (and stable). However, the algorithm developed in [66] starts

with a positive semi-definite function as the initial value function, which has no guarantee to produce stable (or admissible) control law until the iterative value function converges to the optimum.

The goal of the present generalized policy iteration algorithm is to construct an iterative control law $v_i(x_k)$, which moves the system state from an arbitrary initial state x_0 to the equilibrium 0, and simultaneously makes the iterative value function reach the optimum. Under the strictly-hypothetical assumption that the perfect function approximation is available, convergence and admissibility proofs will be given to show that the value function will converge to the optimum and any of the iterative control laws can stabilize the nonlinear system.

A. Derivation of the Generalized Policy Iteration Algorithm

For the optimal control problems, the present control scheme must not only stabilize the control systems, but also make the performance index function finite, i.e., the control law must be admissible [29].

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

Define $\mathcal{A}_u$ as the set of the admissible control laws for system (1) with respect to (2). The present generalized policy iteration algorithm contains two iteration procedures, which are *i*- and *j*-iterations, respectively. We introduce two iteration indices i and j_i and both of the iteration indices increase from 0. Then, the detailed generalized policy iteration algorithm can be described as follows.

For $i = 0$, let $v_0(x_k) \in \mathcal{A}_u$ be an arbitrary admissible control law. For all $x_k \in \Omega_x$, let $V_0(x_k)$ be an iterative value function constructed by $v_0(x_k)$, that satisfies the following generalized HJB (GHJB) equation:

$$V_0(x_k) = U(x_k, v_0(x_k)) + V_0(F(x_k, v_0(x_k))). \tag{6}$$

Let $\{N_1, N_2, \ldots\}$ be an arbitrary sequence, where $N_i \geq 0$, $i = 1, 2, \ldots$, is an arbitrary nonnegative integer. Then, for $i = 1$ and all $x_k \in \Omega_x$, the iterative control law is improved by

$$\begin{aligned} v_1(x_k) &= \arg\min_{u_k}\{U(x_k, u_k) + V_0(x_{k+1})\} \\ &= \arg\min_{u_k}\{U(x_k, u_k) + V_0(F(x_k, u_k))\}. \end{aligned} \tag{7}$$

Let j_1 increase from 0 to N_1. For all $x_k \in \Omega_x$, we update the iterative value function by

$$V_{1,j_1+1}(x_k) = U(x_k, v_1(x_k)) + V_{1,j_1}(F(x_k, v_1(x_k))) \tag{8}$$

where

$$\begin{aligned} V_{1,0}(x_k) &= \min_{u_k}\{U(x_k, u_k) + V_0(F(x_k, u_k))\} \\ &= U(x_k, v_1(x_k)) + V_0(F(x_k, v_1(x_k))). \end{aligned} \tag{9}$$

For all $x_k \in \Omega_x$, define the iterative value function

$$V_1(x_k) = V_{1,N_1}(x_k). \tag{10}$$

For $i = 2, 3, \ldots$ and all $x_k \in \Omega_x$, the generalized policy iteration algorithm can be expressed by the following two iteration procedures.

1) *i*-iteration

$$\begin{aligned} v_i(x_k) &= \arg\min_{u_k}\{U(x_k, u_k) + V_{i-1}(x_{k+1})\} \\ &= \arg\min_{u_k}\{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\}. \end{aligned} \tag{11}$$

2) *j*-iteration

$$V_{i,j_i+1}(x_k) = U(x_k, v_i(x_k)) + V_{i,j_i}(F(x_k, v_i(x_k))) \tag{12}$$

where the iteration index j_i increasing from 0 to N_i and

$$\begin{aligned} V_{i,0}(x_k) &= \min_{u_k}\{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\} \\ &= U(x_k, v_i(x_k)) + V_{i-1}(F(x_k, v_i(x_k))). \end{aligned} \tag{13}$$

For all $x_k \in \Omega_x$, define the iterative value function

$$V_i(x_k) = V_{i,N_i}(x_k). \tag{14}$$

From the above generalized policy iteration algorithm, we can see that for each *i*-iteration, based on the iterative value function $V_i(x_k)$ for some control law, we can always use it to find another policy that is better, or at least no worse. This iteration procedure is known as "policy improvement" procedure. In this iteration procedure, the control law is updated. For each *j*-iteration, it computes the iterative value function of the control law $v_i(x_k)$, which tries to solve the following GHJB equation:

$$V_{i,j_i}(x_k) = U(x, v_i(x_k)) + V_{i,j_i}(F(x_k, v_i(x_k))). \tag{15}$$

This iteration procedure is called "policy evaluation" procedure [4], [26]. In this iteration procedure, the iterative value function $V_{i,j_i}(x_k)$ is updated, while the control law keeps unchanged.

Remark 2: There are two special cases we can identify for the present generalized policy iteration algorithm (6)–(14).

1) For $i = 1, 2, \ldots$, if we let $N_i \equiv 0$, then the algorithm is reduced to a value iteration algorithm [27], [28].
2) For $i = 1, 2, \ldots$, if we let $N_i \to \infty$, then the algorithm becomes a policy iteration algorithm [36].

An obvious difference is that in value iteration algorithms [27], [28] and policy iteration algorithms [36], there is only one iteration index. However, in the present generalized policy iteration algorithm, there are two iteration procedures which are *i*- and *j*-iteration. In [66], a generalized policy iteration algorithm initialized by an arbitrary positive semi-definite function is developed, while the stability property of the system under the iterative control law in [66] cannot be guaranteed. In this paper, the present generalized policy iteration algorithm is initialized by an admissible control law $v_0(x_k)$, which is obviously different from the algorithm in [66]. Hence, new analysis methods will be needed to analyze the present algorithm.

Remark 3: Although the time indices k and $k+1$ are used to indicate the states and actions in two successive time steps, we note that there is no iteration loop for time index k in the generalized policy iteration algorithm (6)–(14), which means that the algorithm does not iterate according to the time sequence. In the present generalized policy iteration algorithm, we say

that the iterative value function $V_{i,j_i}(x_k)$ and the iterative control law $v_i(x_k)$ are updated for all $x_k \in \Omega_x$, according to the two iteration indices $i = 0, 1, \ldots$ and $j_i = 1, 2, \ldots, N_i$. We need to keep the time index this way due to the need of state x at two different time instances as in (7) and it is more clear later of the need in cases like (31) where multiple time instances of state x are involved in an equation.

Remark 4: In the generalized policy iteration algorithm (6)–(14), the iterative value function and iterative control law are improved under the strictly-hypothetical assumption that the perfect function approximation is available. In the next section, the properties of the generalized policy iteration algorithm under the strictly-hypothetical assumption will be developed. On the other hand, we can see that the perfect function approximation in (6)–(14) requires $V_{i,j_i}(x_k)$ and $v_i(x_k)$ to be exactly solved for all $x_k \in \Omega_x$ with an infinite number of points. It is nearly impossible to implement due to finite memory storage and finite time. Hence, in Section IV, approximate structures will be employed to obtain the approximate optimal solutions of the optimal control problem.

B. Properties of the Generalized Policy Iteration Algorithm

Next, we will prove that for any $N_i \geq 0$ and for all $x_k \in \Omega_x$, the iterative value function $V_{i,j_i}(x_k)$ will converge to $J^*(x_k)$ as $i \to \infty$, under the assumption of perfect function approximation. We will also show the admissibility property of the iterative control law $v_i(x_k)$ under the same assumption.

Theorem 1: Let $v_0(x_k) \in \mathcal{A}_u$ be an arbitrary admissible control law which satisfies (6). For $i = 0, 1, \ldots$ and for all $x_k \in \Omega_x$, let the iterative control law $v_i(x_k)$ and the iterative value function $V_{i,j_i+1}(x_k)$ be obtained by (6)–(14). Let $\{N_1, N_2, \ldots\}$ be an sequence, where $N_i \geq 0$, $i = 1, 2, \ldots,$ is an arbitrary nonnegative integer. Then, we have the following properties.

1) For $i = 1, 2, \ldots,$ $j_i = 0, 1, \ldots, N_i$ and for all $x_k \in \Omega_x$

$$V_{i,j_i+1}(x_k) \leq V_{i,j_i}(x_k). \tag{16}$$

2) For $i = 1, 2, \ldots,$ let j_i and $j_{(i+1)}$ be arbitrary constant integers which satisfy $0 \leq j_i \leq N_i$ and $0 \leq j_{(i+1)} \leq N_{i+1}$, respectively. Then, for all $x_k \in \Omega_x$

$$V_{i+1,j_{(i+1)}}(x_k) \leq V_{i,j_i}(x_k). \tag{17}$$

Proof: The theorem can be proven in two steps. We first prove that (16) holds by mathematical induction. Let $i = 1$. According to (6) and (9), for all $x_k \in \Omega_x$

$$\begin{aligned} V_{1,0}(x_k) &= U(x, v_1(x_k)) + V_0(F(x_k, v_1(x_k))) \\ &= \min_{u_k}\{U(x_k, u_k) + V_0(x_{k+1})\} \\ &\leq U(x_k, v_0(x_k)) + V_0(F(x_k, v_0(x_k))) \\ &= V_0(x_k). \end{aligned} \tag{18}$$

For $j_1 = 0$

$$\begin{aligned} V_{1,1}(x_k) &= U(x_k, v_1(x_k)) + V_{1,0}(F(x_k, v_1(x_k))) \\ &\leq U(x_k, v_1(x_k)) + V_0(F(x_k, v_1(x_k))) \\ &= V_{1,0}(x_k). \end{aligned} \tag{19}$$

Assume that (16) holds for $j_1 = l_1 - 1$, $l_1 = 1, 2, \ldots, N_1$. Then for $j_1 = l_1$ and for all $x_k \in \Omega_x$

$$\begin{aligned} V_{1,l_1+1}(x_k) &= U(x_k, v_1(x_k)) + V_{1,l_1}(F(x_k, v_1(x_k))) \\ &\leq U(x_k, v_1(x_k)) + V_{1,l_1-1}(F(x_k, v_1(x_k))) \\ &= V_{1,l_1}(x_k). \end{aligned} \tag{20}$$

Hence, (16) holds for $i = 1$. Next, let $i = 2$. According to (72), the iterative control law $v_2(x_k)$ can be expressed by

$$v_2(x_k) = \arg\min_{u_k}\{U(x_k, u_k) + V_1(F(x_k, u_k))\} \tag{21}$$

where $V_1(x_k) = V_{1,N_1}(x_k)$. According to (13) and (14), for all $x_k \in \Omega_x$, we can get

$$\begin{aligned} V_{2,0}(x_k) &= U(x_k, v_2(x_k)) + V_1(F(x_k, v_2(x_k))) \\ &= \min_{u_k}\{U(x_k, u_k) + V_1(F(x_k, u_k))\} \\ &\leq U(x_k, v_1(x_k)) + V_1(F(x_k, v_1(x_k))) \\ &= V_{1,N_1+1}(x_k) \\ &\leq V_{1,N_1}(x_k) \\ &= V_1(x_k). \end{aligned} \tag{22}$$

For $j_2 = 0$ and for all $x_k \in \Omega_x$

$$\begin{aligned} V_{2,1}(x_k) &= U(x_k, v_2(x_k)) + V_{2,0}(F(x_k, v_2(x_k))) \\ &\leq U(x_k, v_2(x_k)) + V_1(F(x_k, v_2(x_k))) \\ &= V_{2,0}(x_k). \end{aligned} \tag{23}$$

So, (16) holds for $j_2 = 0$. Assume that (16) holds for $j_2 = l_2 - 1$, $l_2 = 1, 2, \ldots, N_2$. Then for $j_2 = l_2$ and for all $x_k \in \Omega_x$

$$\begin{aligned} V_{2,l_2+1}(x_k) &= U(x_k, v_2(x_k)) + V_{2,l_2}(F(x_k, v_2(x_k))) \\ &\leq U(x_k, v_2(x_k)) + V_{2,l_2-1}(F(x_k, v_2(x_k))) \\ &= V_{2,l_2}(x_k). \end{aligned} \tag{24}$$

Then, (16) holds for $i = 2$. Assume that (16) holds for $i = r$, $r = 1, 2, \ldots,$ that is

$$V_{r,j_r+1}(x_k) \leq V_{r,j_r}(x_k). \tag{25}$$

Then for $i = r + 1$ and for all $x_k \in \Omega_x$, the iterative control law can be updated by

$$v_{r+1}(x_k) = \arg\min_{u_k}\{U(x_k, u_k) + V_r(F(x_k, u_k))\} \tag{26}$$

where $V_r(x_k) = V_{r,N_r}(x_k)$. According to (13) and (14), for all $x_k \in \Omega_x$, we can get

$$\begin{aligned} V_{r+1,0}(x_k) &= U(x_k, v_{r+1}(x_k)) + V_r(F(x_k, v_{r+1}(x_k))) \\ &= \min_{u_k}\{U(x_k, u_k) + V_r(x_{k+1})\} \\ &\leq U(x_k, v_r(x_k)) + V_r(F(x_k, v_r(x_k))) \\ &\leq V_{r,N_r}(x_k) \\ &= V_r(x_k). \end{aligned} \tag{27}$$

For $j_{r+1} = 0$ and for all $x_k \in \Omega_x$

$$\begin{aligned} V_{r+1,1}(x_k) &= U(x_k, v_{r+1}(x_k)) + V_{r+1,0}(F(x_k, v_{r+1}(x_k))) \\ &\leq U(x_k, v_{r+1}(x_k)) + V_r(F(x_k, v_{r+1}(x_k))) \\ &= V_{r+1,0}(x_k). \end{aligned} \tag{28}$$

So, (16) holds for $j_{r+1} = 0$. Assume that (16) holds for $j_{r+1} = l_r - 1$, $l_{r+1} = 1, 2, \ldots, N_{r+1}$. Then for $j_{r+1} = l_{r+1}$

$$\begin{aligned} &V_{r+1,l_{r+1}+1}(x_k) \\ &\quad = U(x_k, v_{r+1}(x_k)) + V_{r+1,l_{r+1}}(F(x_k, v_{r+1}(x_k))) \\ &\quad \le U(x_k, v_{r+1}(x_k)) + V_{r+1,l_{r+1}-1}(F(x_k, v_{r+1}(x_k))) \\ &\quad = V_{r+1,l_{r+1}}(x_k). \end{aligned} \tag{29}$$

Hence, (16) holds for $i = 1, 2, \ldots$ and $j_i = 0, 1, \ldots, N_i$. The mathematical induction is completed.

Next, we will prove inequality (17). For $i = 1$, let $0 \le j_1 \le N_1$ and $0 \le j_2 \le N_2$. According to (18)–(24), for all $x_k \in \Omega_x$

$$\begin{cases} V_1(x_k) = V_{1,N_1}(x_k) \le V_{1,j_1}(x_k) \le V_{1,0}(x_k) \le V_0(x_k) \\ V_2(x_k) = V_{2,N_2}(x_k) \le V_{2,j_2}(x_k) \le V_{2,0}(x_k) \le V_1(x_k) \end{cases} \tag{30}$$

which shows that (17) holds for $i = 1$. Using mathematical induction, it is easy to prove that (17) holds for $i = 1, 2, \ldots$. The proof is completed. ∎

Remark 5: In [66], it is proven that the iterative value function $V_{i,j_i}(x_k)$ is convergent as $i \to \infty$, while the monotonicity property of the iterative value function is not guaranteed. Theorem 1 of this paper shows an important monotonicity property of the present generalized policy iteration algorithm. Given an arbitrary initial admissible control law $v_0(x_k) \in \mathcal{A}_u$ which satisfies (6), we have that the iterative value function $V_{i,j_i}(x_k)$ is monotonically nonincreasing for $i = 1, 2, \ldots$ and for $j_i = 0, 1, \ldots, N_i$. From the monotonicity property, the admissibility and convergence properties can be derived.

Lemma 1: Suppose that Assumptions 1–4 hold. For $i = 1, 2, \ldots$ and for $j_i = 0, 1, \ldots, N_i$, the iterative value function $V_{i,j_i}(x_k)$ is a positive definite function for x_k.

Proof: Let $\underline{v}_k^0 = \{v_0(x_k), v_0(x_{k+1}), \ldots\}$. As $v_0(x_k) \in \mathcal{A}_u$ is admissible, according to (2) and (6), for all $x_k \in \Omega_x$, the iterative value function

$$V_0(x_k) = J\left(x_k, \underline{v}_k^0\right) = \sum_{l=0}^{\infty} U(x_{k+l}, v_0(x_{k+l})) \tag{31}$$

is finite. For $x_k = 0$, we have $v_0(x_k) = 0$. According to Assumptions 2 and 3, we can get $x_{k+1} = F(x_0, v_0(x_k)) = 0$. By mathematical induction, for $l = 0, 1, \ldots$, we have $x_{k+l} = 0$. According to (31) and Assumption 4, we can get $V_0(x_k) = 0$. On the other hand, by Assumption 1, as system (1) is controllable and $v_0(x_k)$ is admissible, for all $x_k \in \Omega_x$, $V_0(x_k)$ is finite. According to Assumption 4, $V_0(x_k) \to \infty$, as $x_k \to \infty$. As $U(x_k, u_k) > 0$ for all $x_k \ne 0$, $V_0(x_k) > 0$ for all $x_k \ne 0$. Hence, $V_0(x_k)$ is a positive definite function. According to (6)–(14), using mathematical induction we can easily obtain that $V_{i,j_i}(x_k)$ is positive definite. ∎

Theorem 2: For $i = 1, 2, \ldots$ and $j_i = 0, 1, \ldots, N_i$, let the iterative control law $v_i(x_k)$ and the iterative value function $V_{i,j_i}(x_k)$ be obtained by (6)–(14). If for $i = 1, 2, \ldots$, we let $N_i \to \infty$, then for all $x_k \in \Omega_x$ the iterative value function $V_{i,j_i}(x_k)$ is convergent as $j_i \to \infty$, that is

$$V_{i,\infty}(x_k) = U(x, v_i(x_k)) + V_{i,\infty}(F(x_k, v_i(x_k))) \tag{32}$$

where

$$V_{i,\infty}(x_k) \coloneqq \lim_{j_i \to \infty} V_{i,j_i}(x_k). \tag{33}$$

Proof: According to (16), for $i = 1, 2, \ldots$ and for all $x_k \in \Omega_x$, the iterative value function $V_{i,j_i}(x_k)$ is monotonically nonincreasing as j_i increases from 0 to N_i. On the other hand, according to Lemma 1, $V_{i,j_i}(x_k)$ is a positive definite function for $i = 1, 2, \ldots$ and $j_i = 0, 1, \ldots, N_i$, i.e., $V_{i,j_i}(x_k) > 0$, $\forall x_k \ne 0$. This means that the iterative value function $V_{i,j_i}(x_k)$ is monotonically nonincreasing and lower bounded. Hence, for all $x_k \in \Omega_x$, the limit of $V_{i,j_i}(x_k)$ exists when $j_i \to \infty$. Then, we can obtain (32) directly. ∎

Corollary 1: For $i = 1, 2, \ldots$ and $j_i = 0, 1, \ldots, N_i$, let the iterative control law $v_i(x_k)$ and the iterative value function $V_{i,j_i+1}(x_k)$ be obtained by (6)–(14). Then, for $i = 1, 2, \ldots$ and for all $x_k \in \Omega_x$, the iterative control law $v_i(x_k)$ is admissible.

Proof: Let $\underline{N}_i^{\infty} = \{N_i + 1, N_i + 2, \ldots\}$. For $\bar{j}_i = N_i + 1$, $N_i + 2, \ldots$ and for all $x_k \in \Omega_x$, we construct a value function $\mathcal{V}_{i,\bar{j}_i}(x_k)$ as

$$\mathcal{V}_{i,\bar{j}_i}(x_k) = U(x_k, v_i(x_k)) + \mathcal{V}_{i,\bar{j}_i-1}(F(x_k, v_i(x_k))) \tag{34}$$

where $\mathcal{V}_{i,N_i}(x_k) = V_{i,N_i}(x_k)$. According to (34), we can obtain

$$\mathcal{V}_{i,\bar{j}_i}(x_k) = \sum_{l=0}^{\bar{j}_i - N_i - 1} U(x_{k+l}, v_i(x_{k+l})) + \mathcal{V}_{i,N_i}\left(x_{k+\bar{j}_i-N_i}\right). \tag{35}$$

According to Theorem 2, for all $x_k \in \Omega_x$, the iterative value function $\mathcal{V}_{i,\infty}(x_k)$, which is expressed by

$$\mathcal{V}_{i,\infty}(x_k) = \lim_{\bar{j}_i \to \infty} \sum_{l=0}^{\bar{j}_i - N_i - 1} U(x_{k+l}, v_i(x_{k+l})) + \lim_{\bar{j}_i \to \infty} \mathcal{V}_{i,N_i}\left(x_{k+\bar{j}_i-N_i}\right) \tag{36}$$

is finite. According to Assumption 4, the utility function $U(x_k, v_i(x_k)) > 0$, $\forall x_k \ne 0$. Then, $\lim_{k \to \infty} U(x_k, v_i(x_k)) = 0$, which shows $x_k \to 0$ as $k \to \infty$. On the other hand, according to Lemma 1, $\mathcal{V}_{i,N_i}(x_k) = V_{i,N_i}(x_k)$ is positive definite. Thus, we can get $\lim_{\bar{j}_i \to \infty} \mathcal{V}_{i,N_i}(x_{k+\bar{j}_i-N_i}) = 0$. As $\sum_{l=0}^{N_i} U(x_{k+l}, v_i(x_{k+l}))$ is finite, we obtain

$$\sum_{l=0}^{\infty} U(x_{k+l}, v_i(x_{k+l})) = \sum_{l=0}^{N_i} U(x_{k+l}, v_i(x_{k+l})) + \mathcal{V}_{i,\infty}(x_{k+N_i+1}) \tag{37}$$

also finite. The proof is completed. ∎

Remark 6: In [36], it shows that for $i = 0, 1, \ldots$ and for all $x_k \in \Omega_x$, the iterative control law $v_i(x_k)$ is admissible for the policy iteration algorithm. This property can be well verified for the present generalized policy iteration for $j_i \to \infty$. For the generalized policy iteration, it proves that for an arbitrary nonnegative integer N_i, the iterative control law $v_i(x_k)$ is also admissible. On the other hand, for $i = 0, 1, \ldots$, in the policy iteration algorithm, it requires to solve a GHJB equation to obtain the iterative value function. In the present generalized policy iteration algorithm, solving the GHJB equation is effectively avoided. Hence, we say that the present generalized policy iteration algorithm possesses more potential for applications.

In the following, the convergence property of the generalized policy iteration algorithm will be presented. As the iteration index i increases to ∞, we will show that the optimal performance index function and the optimal control law

can be achieved using the present generalized policy iteration algorithm. Before the main theorem, some lemmas are necessary.

Lemma 2: If a monotonically nonincreasing sequence $\{a_n\}$, $n = 0, 1, \ldots$, contains an arbitrary convergent subsequence, then sequence $\{a_n\}$ is convergent [67].

Lemma 3: For $i = 1, 2, \ldots$, let the iterative value function $V_i(x_k)$ be defined as in (14). Then, for $i = 1, 2, \ldots$, the iterative value function $V_i(x_k)$ is a monotonically nonincreasing and convergent sequence.

Proof: It can be proven according to Theorem 1 and the proof details are omitted here. ■

Theorem 3: For $i = 0, 1, \ldots$ and for all $x_k \in \Omega_x$, let $V_{i,j_i}(x_k)$ and $v_i(x_k)$ be obtained by (6)–(14). If Assumptions 1–4 hold, then for any $N_i \geq 0$, the iterative value function $V_{i,j_i}(x_k)$ converges to the optimal performance index function $J^*(x_k)$, as $i \to \infty$, that is

$$\lim_{i\to\infty} V_{i,j_i}(x_k) = J^*(x_k) \tag{38}$$

which satisfies the HJB equation (4).

Proof: Define a sequence of the iterative value function as

$$\begin{aligned}\left\{V_{i,j_i}(x_k)\right\} \coloneqq \big\{&V_0(x_k), V_{1,0}(x_k), V_{1,1}(x_k), \ldots, V_{1,N_1}(x_k)\\ &V_1(x_k), V_{2,0}(x_k), \ldots, V_{2,N_2}(x_k), \ldots\big\}.\end{aligned} \tag{39}$$

If we let

$$\{V_i(x_k)\} \coloneqq \{V_0(x_k), V_1(x_k), \ldots\} \tag{40}$$

then $\{V_i(x_k)\}$ is a subsequence of $\{V_{i,j_i}(x_k)\}$. According to Lemma 3, the limit of $\{V_i(x_k)\}$ exists. From Lemma 2, we can get that if the sequence $\{V_i(x_k)\}$ is convergent, then $\{V_{i,j_i}(x_k)\}$ is convergent. As a sequence $\{V_{i,j_i}(x_k)\}$ can converge to at most one point [67], the sequence $\{V_{i,j_i}(x_k)\}$ and its subsequence $\{V_i(x_k)\}$ possess the same limit, that is

$$\lim_{i\to\infty} V_{i,j_i}(x_k) = \lim_{i\to\infty} V_i(x_k). \tag{41}$$

Thus, in the following, we will prove:

$$\lim_{i\to\infty} V_i(x_k) = J^*(x_k). \tag{42}$$

The statement (42) can be proven in three steps.

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

According to Lemma 3, for all $x_k \in \Omega_x$, we can define the value function $V_\infty(x_k)$ as the limit of the iterative value function $V_i(x_k)$, that is

$$V_\infty(x_k) = \lim_{i\to\infty} V_i(x_k). \tag{43}$$

According to (14) and (16)

$$\begin{aligned}V_i(x_k) &\leq V_{i,0}(x_k)\\ &= U(x_k, v_i(x_k)) + V_{i-1}(F(x_k, v_i(x_k)))\\ &= \min_{u_k}\{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\}.\end{aligned} \tag{44}$$

Then, we can obtain

$$\begin{aligned}V_\infty(x_k) &= \lim_{i\to\infty} V_i(x_k) \leq V_i(x_k)\\ &\leq \min_{u_k}\{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\}.\end{aligned} \tag{45}$$

Let $i \to \infty$. For all $x_k \in \Omega_x$, we can obtain

$$V_\infty(x_k) \leq \min_{u_k}\{U(x_k, u_k) + V_\infty(F(x_k, u_k))\}. \tag{46}$$

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

$$V_p(x_k) - \varepsilon \leq V_\infty(x_k) \leq V_p(x_k). \tag{47}$$

Hence, we can get

$$\begin{aligned}V_\infty(x_k) &\geq U(x_k, v_p(x_k)) + V_p(F(x_k, v_p(x_k))) - \varepsilon\\ &\geq U(x_k, v_p(x_k)) + V_\infty(F(x_k, v_p(x_k))) - \varepsilon\\ &\geq \min_{u_k}\{U(x_k, u_k) + V_\infty(F(x_k, u_k))\} - \varepsilon.\end{aligned} \tag{48}$$

Since $\varepsilon > 0$ is arbitrary, for all $x_k \in \Omega_x$, we have

$$V_\infty(x_k) \geq \min_{u_k}\{U(x_k, u_k) + V_\infty(F(x_k, u_k))\}. \tag{49}$$

Combining (46) and (49), for all $x_k \in \Omega_x$, we can obtain

$$V_\infty(x_k) = \min_{u_k}\{U(x_k, u_k) + V_\infty(F(x_k, u_k))\}. \tag{50}$$

Next, for all $x_k \in \Omega_x$, let $\mu(x_k)$ be an arbitrary admissible control law, and define a new value function $P(x_k)$, which satisfies

$$P(x_k) = U(x_k, \mu(x_k)) + P(F(x_k, \mu(x_k))). \tag{51}$$

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

Step 2: Show that for an arbitrary admissible control law $\mu(x_k)$, the value function $P(x_k) \geq V_\infty(x_k)$.

The statement can be proven by mathematical induction. As $\mu(x_k)$ is an admissible control law, for all $x_k \in \Omega_x$, $x_k \to 0$ as $k \to \infty$. Without loss of generality, let $x_N = 0$ where $N \to \infty$. According to (51)

$$\begin{aligned}P(x_k) = \lim_{N\to\infty}\{&U(x_k, \mu(x_k)) + U(x_{k+1}, \mu(x_{k+1})) + \cdots\\ &+ U(x_{N-1}, \mu(x_{N-1})) + P(x_N)\}\end{aligned} \tag{52}$$

where $x_N = 0$. If we define

$$v_\infty(x_k) = \arg\min_{u_k}\{U(x_k, u_k) + V_\infty(x_{k+1})\} \tag{53}$$

then according to Corollary 1, $v_\infty(x_k)$ is admissible. According to (50), the iterative value function $V_\infty(x_k)$ can be expressed as

$$\begin{aligned}V_\infty(x_k) = &\ U(x_k, v_\infty(x_k)) + U(x_{k+1}, v_\infty(x_{k+1}))\\ &+ \cdots + U(x_{N-1}, v_\infty(x_{N-1})) + V_\infty(x_N)\\ = &\min_{u_k}\Bigg\{U(x_k, u_k)\\ &+ \min_{u_{k+1}}\Bigg\{U(x_{k+1}, u_{k+1}) + \cdots\\ &+ \min_{u_{N-1}}\{U(x_{N-1}, u_{N-1}) + V_\infty(x_N)\}\Bigg\}\Bigg\}.\end{aligned} \tag{54}$$

As $v_\infty(x_k)$ is an admissible control law, 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 (50), we can obtain

$$\begin{aligned} P(x_{N-1}) &= U(x_{N-1}, \mu(x_{N-1})) + P(x_N) \\ &\geq \min_{u_{N-1}}\{U(x_{N-1}, u_{N-1}) + P(x_N)\} \\ &= \min_{u_{N-1}}\{U(x_{N-1}, u_{N-1}) + V_\infty(x_N)\} \\ &= V_\infty(x_{N-1}). \end{aligned} \tag{55}$$

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

$$\begin{aligned} P(x_l) &= U(x_l, \mu(x_l)) + P(x_{l+1}) \\ &\geq \min_{u_l}\{U(x_l, u_l) + P(x_{l+1})\} \\ &\geq \min_{u_l}\{U(x_l, u_l) + V_\infty(x_{l+1})\} \\ &= V_\infty(x_l). \end{aligned} \tag{56}$$

Hence, for all $x_k \in \Omega_x$, the inequality

$$P(x_k) \geq V_\infty(x_k) \tag{57}$$

holds. Mathematical induction is completed.

Step 3: Show that the value function $V_\infty(x_k)$ equals to the optimal performance index function $J^*(x_k)$.

According to the definition of $J^*(x_k)$ in (3), for $i = 0, 1, \ldots$ and for all $x_k \in \Omega_x$, $V_i(x_k) \geq J^*(x_k)$. Let $i \to \infty$, and then we can obtain $V_\infty(x_k) \geq J^*(x_k)$.

On the other hand, for an arbitrary admissible control law $\mu(x_k)$, (57) holds. For all $x_k \in \Omega_x$, let $\mu(x_k) = u^*(x_k)$, where $u^*(x_k)$ is an optimal control law. Then, we can get $V_\infty(x_k) \leq J^*(x_k)$. Hence, we have (38) holds. The proof is completed. ■

Remark 7: In [66], initialized by a positive semi-definite function, it is proven that the iterative value function converges to the optimal performance index function. However, in [66], the updated iterative control law in i-iteration cannot guaranteed to be admissible. This makes the policy evaluation in j-iteration is not sure to implement for N_i iterations to improve the iterative value function by the iterative control law. In this paper, initialized by an arbitrary admissible control law, it is proven that the iterative value function is monotonically nonincreasing and converges to the optimal performance index function. We emphasize that any of the iterative control laws is admissible, which stabilizes the system. Thus, the policy evaluation in j-iteration is guaranteed to implement for N_i iterations by the obtained iterative control law. This is a merit of the present generalized policy iteration algorithm in this paper. Hence, if an admissible control law is obtained, the present generalized policy iteration algorithm in this paper is preferred. In the next section, the method for obtaining the initial admissible control law will be discussed.

C. Relaxing the Initial Condition of the Generalized Policy Iteration Algorithm

In the previous section, the monotonicity, convergence, and admissibility properties of the generalized policy iteration algorithm have been analyzed. From the generalized policy iteration algorithm (6)–(14), we can see that to implement our algorithm, it requires an admissible control law $v_0(x_k) \in \mathcal{A}_u$ to construct the initial value function $V_0(x_k)$ that satisfies (6). Usually $v_0(x_k) \in \mathcal{A}_u$ and $V_0(x_k)$ are difficult to achieve, which makes the present algorithm difficult to implement. In this section, some effective methods will be presented to relax the initial value function of the algorithm.

Algorithm 1 Policy Evaluation Algorithm for Initial Value Function

Initialization:

Choose randomly an array of system states x_k in Ω_x, i.e., $X_k = (x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(p)})$, where p is a large positive integer;

Choose an arbitrary positive semi-definite function $\Psi(x_k) \geq 0$;

Give the initial admissible control law $v_0(x_k)$.

Iteration:

1: Let the iteration index $j_0 = 0$ and let $V_{0,0}(x_k) = \Psi(x_k)$;

2: For all $x_k \in \Omega_x$, update the control law $v_1^{j_0}(x_k)$ by

$$v_1^{j_0}(x_k) = \arg\min_{u_k}\left\{U(x_k, u_k) + V_{0,j_0}(F(x_k, u_k))\right\}, \tag{58}$$

and improve the iterative value function by

$$\begin{aligned} V_{1,0}^{j_0}(x_k) &= \min_{u_k}\left\{U(x_k, u_k) + V_{0,j_0}(F(x_k, u_k))\right\} \\ &= U\left(x_k, v_1^{j_0}(x_k)\right) + V_{0,j_0}\left(F\left(x_k, v_1^{j_0}(x_k)\right)\right); \end{aligned} \tag{59}$$

3: For all $x_k \in \Omega_x$, if $V_{1,0}^{j_0}(x_k) - V_{0,j_0}(x_k) \leq 0$, goto Step 6. Else goto Step 4;

4: For all $x_k \in \Omega_x$, update the iterative value function by

$$V_{0,j_0+1}(x_k) = U(x_k, v_0(x_k)) + V_{0,j_0}(F(x_k, v_0(x_k))); \tag{60}$$

5: Let $j_0 = j_0 + 1$ and goto Step 2;

6: **return** $V_{0,j_0}(x_k)$ and $v_1^{j_0}(x_k)$. Let $v_1(x_k) = v_1^{j_0}(x_k)$ and $V_{1,0}(x_k) = V_{1,0}^{j_0}(x_k)$.

First, we consider the situation that the admissible control law $v_0(x_k)$ is known. We develop a policy evaluation algorithm to relax the initial value function $V_0(x_k)$. The detailed implementation of the algorithm is expressed in Algorithm 1.

Lemma 4: For all $x_k \in \Omega_x$, let $\Psi(x_k) \geq 0$ be an arbitrary positive semi-definite function. Let $v_0(x_k)$ be an arbitrary admissible control law and let $V_{0,j_0}(x_k)$ be the iterative value function updated by (58)–(60), where $V_{0,0}(x_k) = \Psi(x_k)$. We obtain that $V_{0,j_0}(x_k)$ is convergent as $j_0 \to \infty$.

Proof: According to (60), for all $x_k \in \Omega_x$

$$\begin{aligned} V_{0,j_0+1}(x_k) - V_{0,j_0}(x_k) =\ & U(x_k, v_0(x_k)) + V_{0,j_0}(x_{k+1}) \\ & - \left(U(x_k, v_0(x_k)) + V_{0,j_0-1}(x_{k+1})\right) \\ =\ & V_{0,j_0}(x_{k+1}) - V_{0,j_0-1}(x_{k+1}). \end{aligned} \tag{61}$$

According to (61), we can get

$$\begin{cases} V_{0,j_0+1}(x_k) - V_{0,j_0}(x_k) = V_{0,1}\left(x_{k+j_0}\right) - V_{0,0}(x_{k+j_0}) \\ V_{0,j_0}(x_k) - V_{0,j_0-1}(x_k) = V_{0,1}(x_{k+j_0-1}) - V_{0,0}\left(x_{k+j_0-1}\right) \\ \qquad\qquad \vdots \\ V_{0,1}(x_k) - V_{0,0}(x_k) = V_{0,1}(x_k) - V_{0,0}(x_k). \end{cases} \tag{62}$$

Algorithm 2 Policy Improvement Algorithm for Initial Value Function

Initialization:

Choose randomly an array of system states x_k in Ω_x, i.e., $X_k = (x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(p)})$, where p is a large positive integer;

Iteration:

Let $\varsigma_0 = 0$.

1: Choose arbitrarily a large positive definite function $\bar{\Psi}^{\varsigma_0}(x_k) \geq 0$ and let $V_{0,0}^{\varsigma_0}(x_k) = \bar{\Psi}^{\varsigma_0}(x_k)$;

2: For all $x_k \in \Omega_x$, update the control law $v_1^{\varsigma_0}(x_k)$ by

$$v_1^{\varsigma_0}(x_k) = \arg\min_{u_k}\left\{U(x_k, u_k) + \bar{\Psi}^{\varsigma_0}(F(x_k, u_k))\right\}, \tag{65}$$

and for all $x_k \in \Omega_x$, improve the iterative value function by

$$\begin{aligned} V_{1,0}^{\varsigma_0}(x_k) &= \min_{u_k}\left\{U(x_k, u_k) + V_{0,0}^{\varsigma_0}(F(x_k, u_k))\right\} \\ &= U\big(x_k, v_1^{\varsigma_0}(x_k)\big) + V_{0,0}^{\varsigma_0}\big(F\big(x_k, v_1^{\varsigma_0}(x_k)\big)\big); \end{aligned} \tag{66}$$

3: For all $x_k \in \Omega_x$, if the inequality

$$V_{1,0}^{\varsigma_0}(x_k) - \bar{\Psi}^{\varsigma_0}(x_k) \leq 0 \tag{67}$$

holds, then goto Step 4. Else let $\varsigma_0 = \varsigma_0 + 1$ and goto Step 1;

4: **return** $V_{0,0}^{\varsigma_0}(x_k)$ and $v_1^{\varsigma_0}(x_k)$. Let $v_1(x_k) = v_1^{\varsigma_0}(x_k)$ and $V_{1,0}(x_k) = V_{1,0}^{\varsigma_0}(x_k)$.

Then, we have

$$V_{0,j_0+1}(x_k) = \sum_{l=0}^{j_0} U(x_{k+l}, v_0(x_{k+l})) + \Psi\big(x_{k+j_0+1}\big). \tag{63}$$

Let $j_0 \to \infty$. We can obtain

$$\lim_{j_0\to\infty} V_{0,j_0+1}(x_k) = \sum_{l=0}^{\infty} U(x_{k+l}, v_0(x_{k+l})). \tag{64}$$

As $v_0(x_k)$ is an admissible control law, $\sum_{l=0}^{\infty} U(x_{k+l}, v_0(x_{k+l}))$ is finite. Hence $\lim_{j_0\to\infty} V_{0,j_0}(x_k)$ is finite, which means $V_{0,j_0+1}(x_k) = V_{0,j_0}(x_k)$, as $j_0 \to \infty$. ■

Using the admissible control law $v_0(x_k)$, according to Lemma 4, $V_{0,j_0+1}(x_k) = V_{0,j_0}(x_k)$ holds as $j_0 \to \infty$. It means that there must exist $N_0 > 0$ which satisfies $V_1(x_k) \leq V_{0,N_0}(x_k)$. Hence if we obtain an admissible control law, then we can construct the initial value function by policy evaluation, where the value function $V_0(x_k)$ in (6) can be relaxed. On the other hand, we can see that Algorithm 1 requires an admissible control law $v_0(x_k)$ to implement. Usually, the admissible control law of the nonlinear system is also difficult to obtain. To overcome this difficulty, a policy improvement algorithm can be implemented by experiment. The details of the algorithm can be seen in Algorithm 2.

Theorem 4: For all $x_k \in \Omega_x$, let the iterative control law $v_1^{\varsigma_0}(x_k)$ be expressed as in (65) and let the iterative value function $V_{1,0}^{\varsigma_0}(x_k)$ be expressed as in (66). If the iterative value functions satisfy (67), then the convergence properties (16) and (17) hold for $i = 1, 2, \ldots$ and $j_i = 0, 1, \ldots, N_i$.

Algorithm 3 Generalized Policy Iteration Algorithm

Initialization:

Choose randomly an array of system states x_k in Ω_x, i.e., $X_k = (x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(p)})$, where p is a large positive integer;

Choose a computation precision ε;

Construct a sequence $\{N_i\}$, where $N_i \geq 0$, $i = 1, 2, \ldots,$ is an arbitrary nonnegative integer.

Iteration:

1: Let the iteration index $i = 0$. Obtain $V_{1,0}(x_k)$ and $v_1(x_k)$ by Algorithm Υ, $\Upsilon = 1, 2$;

2: Let j_1 increase from 0 to N_1. For all $x_k \in \Omega_x$, update the iterative value function by

$$V_{1,j_1+1}(x_k) = U(x_k, v_1(x_k)) + V_{1,j_1}(F(x_k, v_1(x_k)));$$

3: Let $i = i + 1$. For all $x_k \in \Omega_x$, do **Policy Improvement**

$$v_i(x_k) = \arg\min_{u_k}\{U(x_k, u_k) + V_{i-1}(F(x_k, u_k))\};$$

4: Let j_i increase from 0 to N_i. For all $x_k \in \Omega_x$, do **Policy Evaluation**

$$V_{i,j_i+1}(x_k) = U(x_k, v_i(x_k)) + V_{i,j_i}(F(x_k, v_i(x_k)));$$

5: Let $V_i(x_k) = V_{i,N_i}(x_k)$;

6: For all $x_k \in \Omega_x$, if $V_{i-1}(x_k) - V_i(x_k) < \varepsilon$, then the approximate optimal performance index function and the approximate optimal control law are obtained. Goto Step 7. Else goto Step 3;

7: **return** $v_i(x_k)$ and $V_{i,j_i}(x_k)$.

Proof: Let $i = 1$ and $j_1 = 0$. As $v_1(x_k) = v_1^{\varsigma_0}(x_k)$ and $V_{1,0}(x_k) = V_{1,0}^{\varsigma_0}(x_k)$, according to (8) and (67), we have

$$\begin{aligned} V_{1,1}(x_k) &= U(x_k, v_1(x_k)) + V_{1,0}(F(x_k, v_1(x_k))) \\ &= U(x_k, v_1^{\varsigma_0}(x_k)) + V_{1,0}^{\varsigma_0}\big(F\big(x_k, v_1^{\varsigma_0}(x_k)\big)\big) \\ &\leq U\big(x_k, v_1^{\varsigma_0}(x_k)\big) + V_{0,0}^{\varsigma_0}\big(F\big(x_k, v_1^{\varsigma_0}(x_k)\big)\big) \\ &= V_{1,0}(x_k). \end{aligned} \tag{68}$$

Using the idea from (18)–(30), the convergence properties (16) and (17) hold for $i = 1, 2, \ldots$ and $j_i = 0, 1, \ldots, N_i$. ■

Remark 8: From Algorithm 2, we can see that the admissible control law $v_0(x_k)$ in Algorithm 1 is avoided. This is a merit of Algorithm 2. However, in Algorithm 2, we should find a positive definite function $\bar{\Psi}^{\varsigma_0}(x_k)$ that satisfies (67). As $\bar{\Psi}^{\varsigma_0}(x_k)$ is randomly chosen, it may take a lot of iterations to determine $\bar{\Psi}^{\varsigma_0}(x_k)$. This is a disadvantage of the algorithm.

D. Generalized Policy Iteration Algorithm

We are now in a position to summarize the generalized policy iteration ADP algorithm (see Algorithm 3).

IV. Neural Network Implementation

Results so far have shown the convergence of iterative value function $V_{i,j_i}(x_k)$ and iterative control law $v_i(x_k)$. Under ideal conditions, they will converge to their corresponding optimal

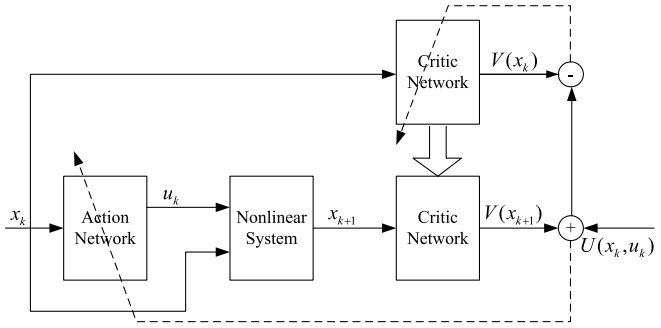


Fig. 1. Structure diagram of the algorithm.

functions. The results given in Sections II and III are under the condition that for $i = 0, 1, \ldots$ and $j_i = 0, 1, \ldots, N_i$, the functions $V_{i,j_i}(x_k)$ and $v_i(x_k)$ can accurately be obtained for all $x_k \in \Omega_x$. In this section, back-propagation (BP) neural networks are introduced to approximate the iterative value function and iterative control law, respectively.

Assume that the number of hidden layer neurons is denoted by $\tilde{\ell}$. The weight matrix between the input layer and hidden layer is denoted by Y. The weight matrix between the hidden layer and output layer is denoted by W. Let b denote the threshold vector of the neural network. Then, the output of three-layer BP network is expressed by

$$\hat{F}(X, Y, W, b) = W^T \sigma\big(Y^T X + b\big) \tag{69}$$

where $\sigma(Y^T X) \in R^{\tilde{\ell}}$, $[\sigma(z)]_i = (e^{z_i} - e^{-z_i})/(e^{z_i} + e^{-z_i})$, $i = 1, \ldots \tilde{\ell}$, are the activation functions. There are two networks, which are critic and action networks, respectively. Both neural networks are chosen as three-layer feedforward network. The whole structure diagram is shown in Fig. 1.

A. Critic Network

The critic network is used to approximate the iterative value function $V_{i,j_i}(x_k)$. For all $x_k \in \Omega_x$, the output of the critic network is denoted as $\hat{V}^l_{i,j_i}(x_k) = W^l_{c(i,j_i)} \sigma(Y^l_{c(i,j_i)} x_k + b^l_{c(i,j_i)})$, $l = 0, 1, \ldots$. The target iterative value function can be written as

$$V_{i,j_i}(x_k) = U\big(x_k, \hat{v}_i(x_k)\big) + \hat{V}_{i,j_i-1}\big(F(x_k, \hat{v}_i(x_k))\big). \tag{70}$$

Then, we define the error function for the critic network as $e^l_{c(i,j_i)} = \hat{V}^l_{i,j_i}(x_k) - V_{i,j_i}(x_k)$. The objective function to be minimized in the critic network is $E^l_{c(i,j_i)} = (1/2)(e^l_{c(i,j_i)})^2$. So, the gradient-based weight update rule [68] for the critic network is given by

$$\begin{aligned} w^{l+1}_{c(i,j_i)} &= w^l_{c(i,j_i)} - \alpha_c \frac{\partial E^l_{c(i,j_i)}}{\partial w^l_{c(i,j_i)}} \\ &= w^l_{c(i,j_i)} - \alpha_c \frac{\partial E^l_{c(i,j_i)}}{\partial \hat{V}^l_{i,j_i}(x_k)} \frac{\partial \hat{V}^l_{i,j_i}(x_k)}{\partial w^l_{c(i,j_i)}} \\ &= w^l_{c(i,j_i)} - \alpha_c e^l_{c(i,j_i)} \frac{\partial \hat{V}^l_{i,j_i}(x_k)}{\partial w^l_{c(i,j_i)}} \end{aligned} \tag{71}$$

where $\alpha_c > 0$ is the learning rate of critic network and $w^l_{c(i,j_i)}$ is the weight matrix of the critic network which can be replaced by $W^l_{c(i,j_i)}$, $Y^l_{c(i,j_i)}$, and $b^l_{c(i,j_i)}$.

Remark 9: In (70), the expression of the target iterative value function $V_{i,j_i}(x_k)$ is given, where the information of the value function $\hat{V}_{i,j_i-1}(F(x_k, \hat{v}_i(x_k)))$ in previous iteration is required. For $i = 0, 1, \ldots,$ and $j_i = 1, 2, \ldots, N_i$, the value functions $\hat{V}_{i,j_i-1}(F(x_k, \hat{v}_i(x_k)))$ are obtained by the critic network approximation in the previous iteration. Thus, we say that the function $\hat{V}_{i,j_i-1}(F(x_k, \hat{v}_i(x_k)))$ is known and the gradient-based weight update rule in (71) can be implemented. On the other hand, besides the gradient-based neural networks method, we say that the Galerkin method [69] and stochastic approximation [70] are also effective approximators to reconstruct the iterative function $V_{i,j_i}(x_k)$ with good convergence performance. The corresponding detailed approximation and analysis methods can be seen in [71] and [72]. As the property analysis of function approximation is not the main research topic of this paper, it is omitted here.

B. Action Network

In the action network, the state $x_k \in \Omega_x$ is used as input to the network. The output can be formulated as $\hat{v}^l_i(x_k) = W^l_{ai} \sigma(Y^l_{ai} x_k + b^l_{ai})$, $l = 0, 1, \ldots$. For $i = 1, 2, \ldots,$ the target of the output of the action network is given by

$$v_i(x_k) = \arg\min_{u_k} \Big\{ U(x_k, u_k) + \hat{V}_{i-1}(F(x_k, u_k)) \Big\}. \tag{72}$$

So, we can define the output error of the action network as $e^l_{ai} = \hat{v}^l_i(x_k) - v_i(x_k)$. The weights of the action network are updated to minimize the following performance error measure $E^l_{ai} = (1/2)(e^l_{ai})^T e^l_{ai}$. The gradient-based weight update rule [68] for the action network is given by

$$\begin{aligned} w^{l+1}_{ai} &= w^l_{ai} - \beta_a \frac{\partial E^l_{ai}}{\partial w^l_{ai}} \\ &= w^l_{ai} - \beta_a \frac{\partial E^l_{ai}}{\partial e^l_{ai}} \frac{\partial e^l_{ai}}{\partial \hat{v}^l_i(x_k)} \frac{\partial \hat{v}^l_i(x_k)}{\partial w^l_{ai}} \end{aligned} \tag{73}$$

where $\beta_a > 0$ is the learning rate of action network and w^l_{ai} is the weight matrix of the critic network which can be replaced by W^l_{ai}, Y^l_{ai}, and b^l_{ai}.

Remark 10: From Theorems 1–3, we can see that the convergence and admissibility properties of the present generalized policy iteration algorithm are independent of the approximation structures, such as neural networks. Hence, we say that the present generalized policy iteration algorithm and the corresponding proofs possess theoretical significance. On the other hand, implementing our algorithm by neural networks, approximation errors of neural networks inherently exist. Hence, we declare that an approximate optimal solution of the HJB equation (4) is actually obtained instead of the exact optimal one. To make the iterative value functions and iterative control laws closer to their optimal ones, it requires collecting enough training data and enhancing the training precisions of the neural networks.

V. Simulation Study

In this section, two simulation examples are used to show the performance of the present generalized policy iteration algorithm for solving the approximate optimal control problems.

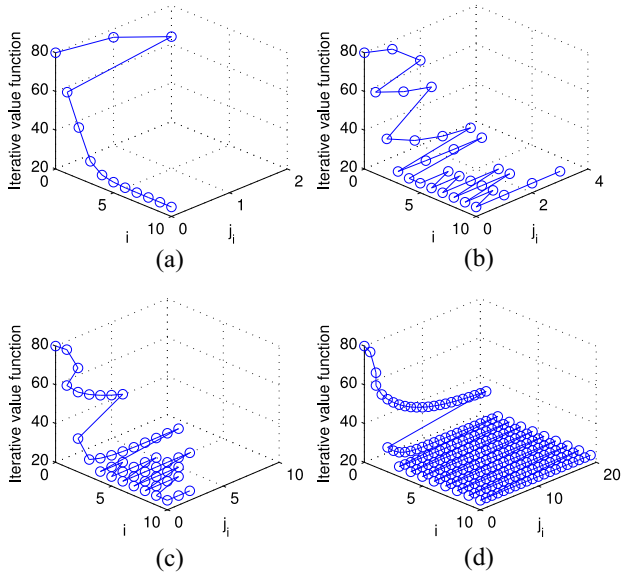


Fig. 2. Iterative value functions $V_{i,j_i}(x_k)$ for $i = 0, 1, \ldots, 10$ and $x_k = x_0$. $V_{i,j_i}(x_k)$ for (a) $\{N_i^1\}$, (b) $\{N_i^2\}$, (c) $\{N_i^3\}$, and (d) $\{N_i^4\}$.

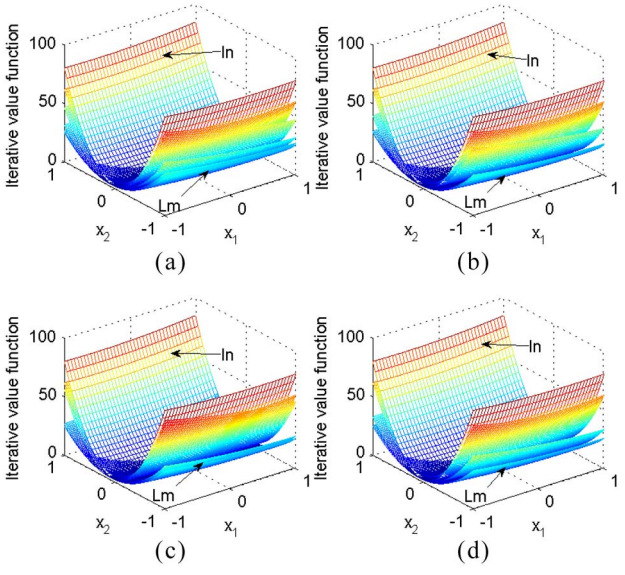


Fig. 3. Iterative value functions $V_i(x_k)$, for $i = 0, 1, \ldots, 10$. $V_i(x_k)$ for (a) $\{N_i^1\}$, (b) $\{N_i^2\}$, (c) $\{N_i^3\}$, and (d) $\{N_i^4\}$.

Example 1: First, let us consider the following spring-mass-damper system [73]:

$$M\frac{d^2y}{dt^2} + b\frac{dy}{dt} + \kappa y = u$$

where y is the position and u is the control input. Let $M = 0.1\,\text{kg}$ denote the mass of object. Let $\kappa = 2\,\text{kgf/m}$ be the stiffness coefficient of spring and let $b = 0.1$ be the wall friction. Let $x_1 = y$ and $x_2 = (dy/dt)$. Discretizing the system function with the sampling interval $\Delta t = 0.1$ s leads to

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ -\dfrac{\kappa}{M}\Delta T & 1 - \dfrac{b}{M}\Delta T \end{bmatrix} \begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{\Delta T}{M} \end{bmatrix} u_k. \tag{74}$$

Let the initial state be $x_0 = [1, -1]^T$. Let the performance index function be expressed by (2). The utility function is expressed as $U(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k$, where $Q = I_1$, $R = I_2$, and I_1 and I_2 denote the identity matrix with suitable dimensions.

Let the state space be $\Omega_x = \{x_k \mid -1 \le x_{1k} \le 1, -1 \le x_{2k} \le 1\}$. We randomly choose the $p = 5000$ states in Ω_x to implement the generalize policy iteration algorithm to obtain the optimal control law. Neural networks are used to implement the present generalized value 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. Define the two neural networks as group "NN1." For system (74), we can obtain an admissible control law $u(x_k) = Kx_k$, where $K = [0.13, -0.17]^T$. Let $\Psi(x_k) = x_k^T P_0 x_k$, where $P_0 = \begin{bmatrix} 80 & 1 \\ 1 & 2 \end{bmatrix}$. As the initial admissible control law K is known, policy evaluation in Algorithm 1 is implemented. It can be seen that it takes three iterations to obtain $v_1(x_k)$ and $V_{1,0}(x_k)$ and the simulation results for the initial iteration is displayed in Fig. 2 (see the trajectories of the iterative value functions for $i = 0$). Let iteration index $i = 10$. To illustrate the effectiveness of the algorithm, we choose four different iteration sequences $\{N_i^\gamma\}$, $\gamma = 1, 2, 3, 4$. For $\gamma = 1$ and $i = 0, 1, \ldots, 10$, we let $N_i^1 = 0$. For $\gamma = 2$, iteration sequence is chosen as $\{N_i^2\} = \{2, 3, 3, 0, 1, 1, 2, 2, 1, 3\}$. For $\gamma = 3$, iteration sequence is chosen as $\{N_i^3\} = \{5, 0, 8, 2, 4, 6, 4, 3, 0, 2\}$. For $\gamma = 4$ and $i = 0, 1, \ldots, 10$, let $N_i^4 = 20$. Train the critic and the action networks under the learning rate 0.01 and set the neural network training errors as 10^{-6}. Under the iteration indices i and j_i, the trajectories of iterative value functions $V_{i,j_i}(x_k)$ for $x_k = x_0$ are shown in Fig. 2. The curves of the iterative value functions $V_i(x_k)$ are shown in Fig. 3, where we let "In" denote initial iteration and "Lm" denote limiting iteration.

For $\{N_i^1\} = 0$, the generalized policy iteration algorithm is reduced to value iteration algorithm [27], [28]. From Figs. 2(a) and 3(a), we can see that the iterative value function converges to the approximate optimum which justifies the effectiveness of our algorithm. For $\{N_i^4\} = 20$, we can see that for $i = 1, \ldots, 10$, the iterative value function $V_{i,j_i}(x_k)$ is convergent for j_i. The generalized policy iteration algorithm is transformed into the policy iteration algorithm [36], where the convergence property can be justified. For arbitrary sequence $\{N_i\}$, such as $\{N_i^2\}$ and $\{N_i^3\}$, From Figs. 2(b) and (c) and 3(b) and (c), the iterative value function can also converge to the approximate optimum. Hence, we can say that value and policy iteration algorithms are special cases of the present generalized policy iteration algorithms and the convergence properties of our algorithm can be justified. The stability property of system (74) under the iterative control law $v_i(x_k)$ is shown in Figs. 4 and 5, respectively.

From the above simulation results, we can see that for $i = 0, 1, \ldots,$ the iterative control law $v_i(x_k)$ is admissible. For linear system (74), we know that the optimal performance index function $J^*(x_k) = x_k^T P^* x_k$. According to the discrete algebraic Riccati equation, we know that $P^* = \begin{bmatrix} 26.61 & 1.81 \\ 1.81 & 1.90 \end{bmatrix}$

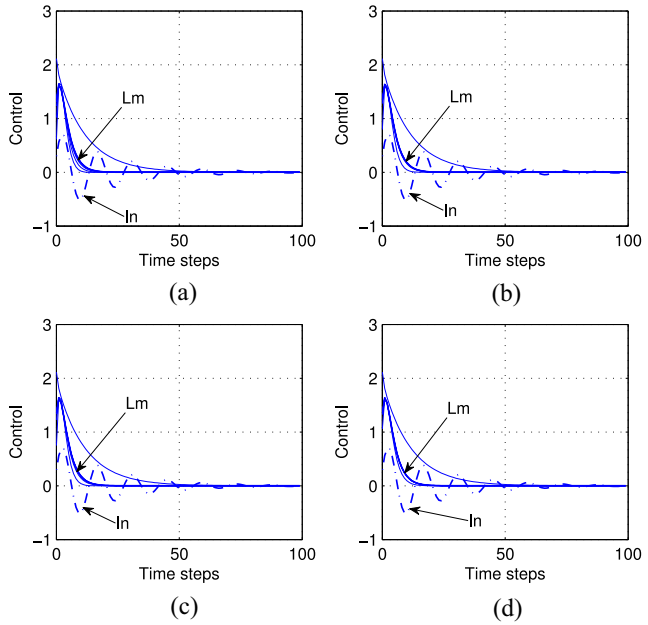


Fig. 4. Trajectories of iterative control law $v_i(x_k)$, $i = 0, 1, \ldots, 10$. $v_i(x_k)$ for (a) $\{N_i^1\}$, (b) $\{N_i^2\}$, (c) $\{N_i^3\}$, and (d) $\{N_i^4\}$.

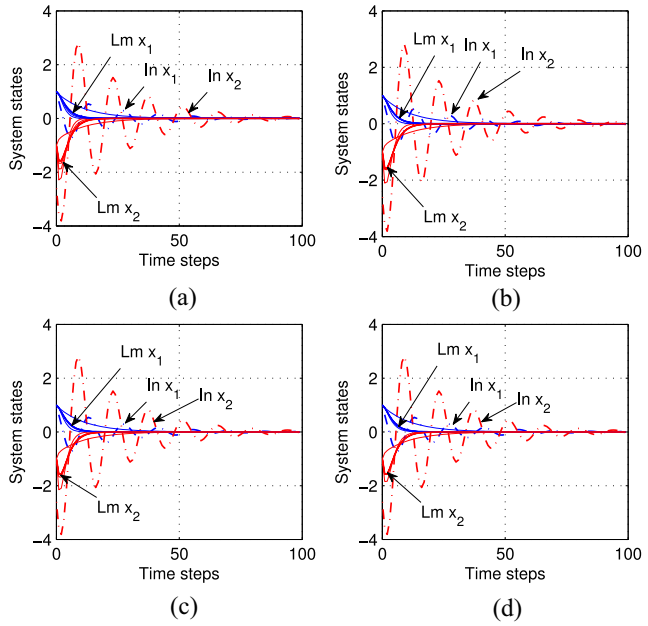


Fig. 5. Trajectories of system state. State trajectories for (a) $\{N_i^1\}$, (b) $\{N_i^2\}$, (c) $\{N_i^3\}$, and (d) $\{N_i^4\}$.

and the effectiveness of the present algorithm can be justified for linear systems.

On the other hand, we know that the structure of the neural networks is important for its approximation performance. To show the influence of the neural network structure, we change the structures of the critic and action networks to 2–4–1 and 2–4–1, respectively, and other parameters of the neural networks are kept unchanged. Define the two neural networks as group "NN2." Choose $\{N_i^2\}$ for the j-iteration. Implement our algorithm for $i = 10$ iterations. The iterative value functions by NN1 and NN2 are shown in Fig. 6(a). We can see that if the number of hidden layer is reduced, the approximate

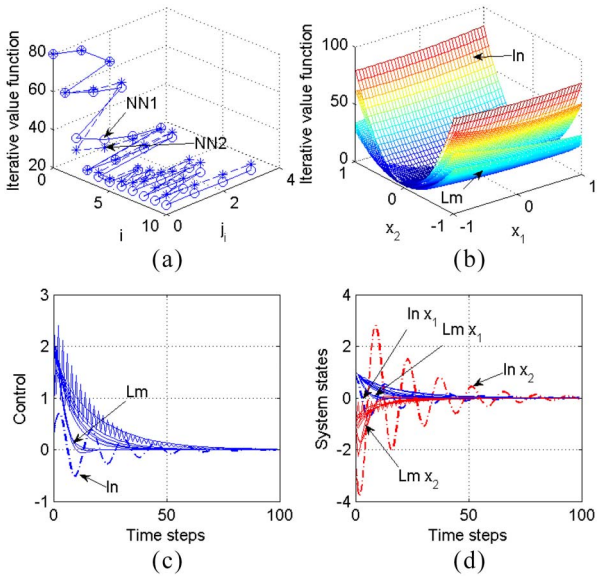


Fig. 6. Simulation results for $i = 0, 1, \ldots, 10$ and $\{N_i^2\}$. (a) Value function at $x = x_0$ for NN1 and NN2. (b) $V_i(x_k)$ by NN2. (c) Iterative control law by NN2. (d) System states by NN2.

performance of the neural networks may decrease. The plot of $V_i(x_k)$ is shown in Fig. 6(b). The corresponding trajectories of iterative states and control are shown in Fig. 6(c) and (d), respectively. We can see that if the structure of the neural networks is not set appropriately, the performance of the control system will be decreased.

Example 2: We now examine the performance of our algorithm in a torsional pendulum system [36], [68] with modifications. The dynamics of the pendulum is given as follows:

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

where $M = 1/3$ kg and $l = 2/3$ m are the mass and length of the pendulum bar, respectively. Let $J = 4/3Ml^2$ and $f_d = 0.2$ be the rotary inertia and frictional factor, respectively. $x_1 = \theta$ and $x_2 = \omega$. Let $g = 9.8$ m/s^2 be the gravity and the sampling time interval $\triangle T = 0.1$ s. Then, the discretized system can be expressed by

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \end{bmatrix} = \begin{bmatrix} 0.1x_{2k} + x_{1k} \\ -0.49\sin(x_{1k}) - 0.1f_d x_{2k} + x_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u_k. \tag{75}$$

Let the initial state be $x_0 = [1, -1]^T$ and let the utility function be the same as the one in Example 1.

Neural networks are also used to implement the generalized policy iteration algorithm, where the structures of the critic network and the action network are the same as the ones in Example 1. We choose $p = 10000$ states in Θ to implement the generalized value iteration algorithm. For nonlinear system (75), the initial admissible control law is difficult to obtain. Thus we implement policy improvement algorithm in Algorithm 2, and we can obtain the initial value function $\bar{\Psi}^{[0]}(x_k) = x_k^T \bar{P}_0 x_k$, where $\bar{P}_0 = \begin{bmatrix} 145.31 & 8.43 \\ 8.43 & 28.42 \end{bmatrix}$.

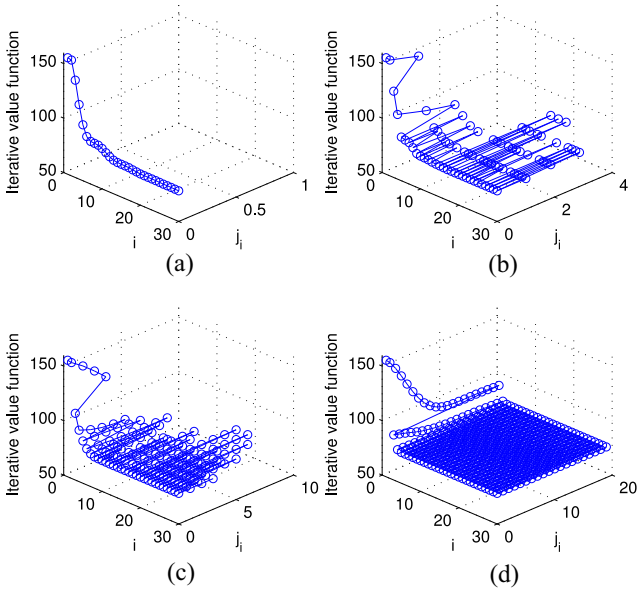


Fig. 7. Iterative value functions $V_{i,j_i}(x_k)$ for $i = 0, 1, \ldots, 30$ and $x_k = x_0$. $V_{i,j_i}(x_k)$ for (a) $\{N_i^1\}$, (b) $\{N_i^2\}$, (c) $\{N_i^3\}$, and (d) $\{N_i^4\}$.

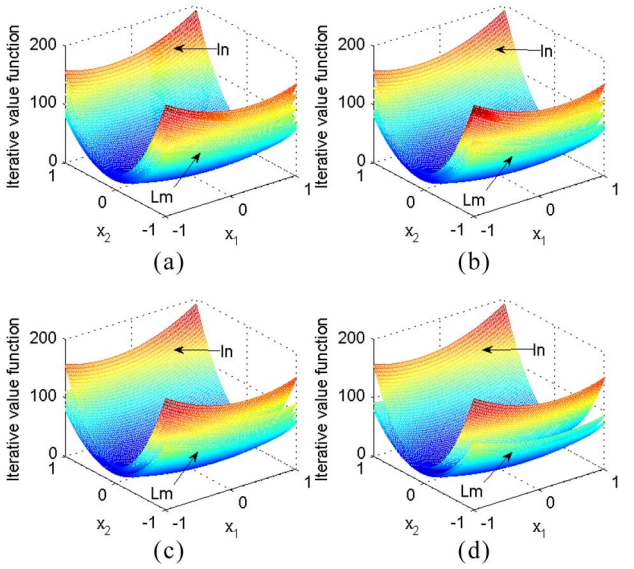


Fig. 8. Iterative value functions $V_i(x_k)$, $i = 0, 1, \ldots, 30$. $V_i(x_k)$ for (a) $\{N_i^1\}$, (b) $\{N_i^2\}$, (c) $\{N_i^3\}$, and (d) $\{N_i^4\}$.

Let iteration index $i = 30$. To illustrate the effectiveness of the algorithm, we choose four different iteration sequences $\{N_i^\gamma\}$, $\gamma = 1, 2, 3, 4$. For $\gamma = 1$ and $\forall i = 0, 1, \ldots, 30$, we let $N_i^1 = 0$. For $\gamma = 2$, let N_i^2, $i = 1, 2, \ldots, 30$, be arbitrary nonnegative integer that satisfies $0 \leq N_i^2 \leq 4$. For $\gamma = 3$, let N_i^3, $i = 1, 2, \ldots, 30$, be arbitrary nonnegative integer that satisfies $0 \leq N_i^3 \leq 10$. For $\gamma = 4$ and $\forall i = 0, 1, \ldots, 30$, let $N_i^4 = 20$. Train the critic and the action networks under the learning rate 0.01 and set the neural network training errors as 10^{-6}. Under the iteration indices i and j_i, the trajectories of iterative value functions $V_{i,j_i}(x_k)$ for $x = x_0$ are shown in Fig. 7. The curves of the iterative value functions $V_i(x_k)$ are shown in Fig. 8.

From Figs. 7 and 8, we can see that given an arbitrary nonnegative integer sequence $\{N_i\}$, $i = 0, 1, \ldots,$ the iterative value function $V_{i,j_i}(x_k)$ is monotonically nonincreasing and converges to the approximate optimum using the present generalized policy iteration algorithm. The convergence property of the present generalized policy iteration algorithm for nonlinear systems can be justified. The convergence properties of value and policy iteration algorithms can also be justified by our algorithm. The stability property of system (74) under the iterative control law $v_i(x_k)$ is shown in Figs. 9 and 10, respectively.

We can see that for $i = 0, 1, \ldots,$ the iterative control law $v_i(x_k)$ is admissible, and hence the effectiveness of the present algorithm can be justified for nonlinear systems.

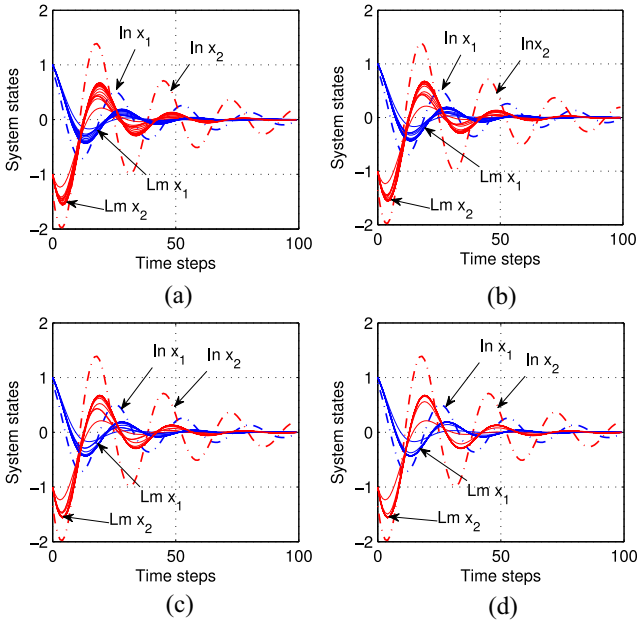


Fig. 9. Trajectories of system state. State trajectories for (a) $\{N_i^1\}$, (b) $\{N_i^2\}$, (c) $\{N_i^3\}$, and (d) $\{N_i^4\}$.

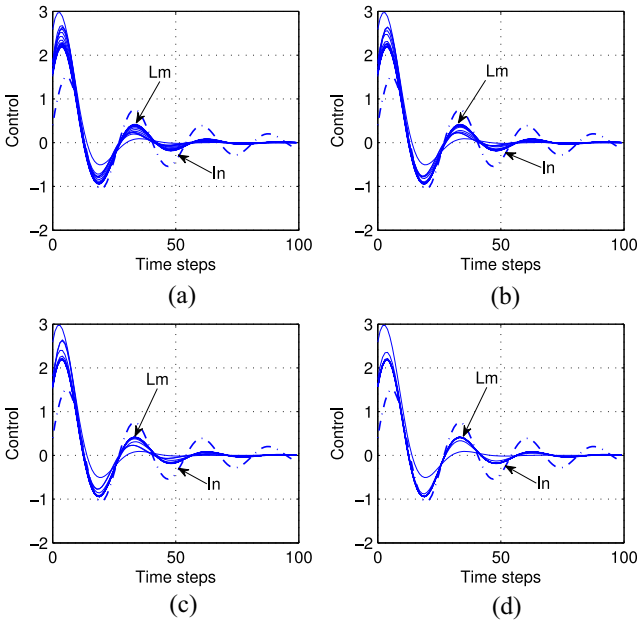


Fig. 10. Trajectories of iterative control law $v_i(x_k)$. $v_i(x_k)$ for (a) $\{N_i^1\}$, (b) $\{N_i^2\}$, (c) $\{N_i^3\}$, and (d) $\{N_i^4\}$.

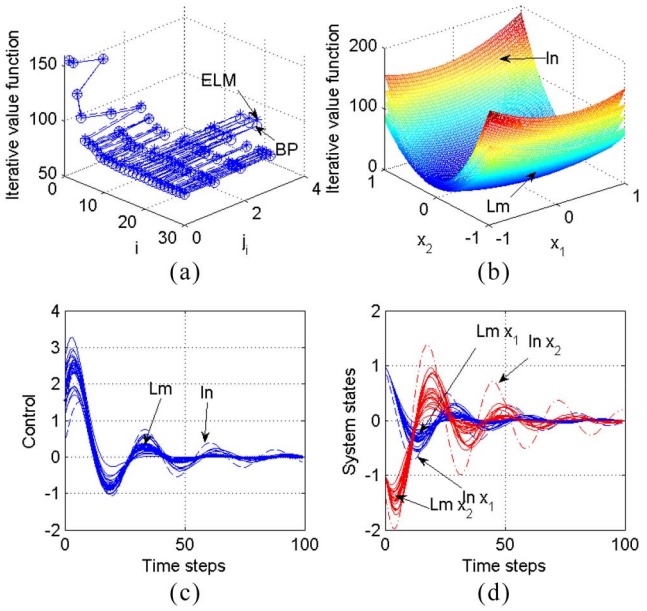


Fig. 11. Simulation results for $i = 0, 1, \ldots, 30$ and $\{N_i^2\}$. (a) Value function at $x_k = x_0$. (b) $V_i(x_k)$ by ELM. (c) Iterative control law by ELM. (d) System states by ELM.

Remark 11: One property should be pointed out. From Figs. 7(a) and 8(a), as $N_i \equiv 0$, for $i = 1, 2, \ldots, 30$, we can see that it takes 15 i-iterations to make the iterative value function converge to the optimal performance index function. From Figs. 7(d) and 8(d), as N_i is large, i.e., $N_i = 20$ for $i = 1, 2, \ldots, 30$, we can see that it only takes four i-iterations to make the iterative value function converge to the optimum which is much less than the situation for $N_i \equiv 0$. In each i-iteration, however, it has to take 20 j-iterations to make $V_{i,j_i}(x_k)$ convergent. Thus, if we want to obtain the optimal performance index function by the least times of policy improvement, then we can enlarge the number of j-iteration. On the other hand, if we want to obtain the optimal performance index function by the least times of policy evaluation, then reducing the number of j-iteration can be an effective method. For value and policy iteration algorithms, the numbers of j-iteration are fixed at 0 and ∞, respectively, which means the convergence of value and policy iteration algorithms is nonregulatable. For the present generalized policy iteration algorithm, we can regulate the convergence property of the iterative value function by defining a suitable sequence $\{N_i\}$. This is another merit of the generalized policy iteration algorithm.

In the above simulations, BP neural networks are used to implement our algorithm. We have also tried extreme learning machine (ELM) [74], [75] to show the effectiveness of the present algorithm. For $j = 1, 2, \ldots, p$, let $\tilde{N}$ be the number of hidden nodes. The standard ELM is expressed by

$$f_L\left(x_k^{(j)}\right) = \sum_{i=1}^{\tilde{N}} \beta_i h\left(x_k^{(j)}\right) = \sum_{i=1}^{\tilde{N}} \beta_i h\left(\tilde{w}_i x_k^{(j)} + b_i\right) \qquad (76)$$

where $\tilde{w}_i$ is the weight vector connecting the ith hidden node and the input nodes. Let β_i be the vector connecting the ith hidden node and the output nodes and let b_i be the threshold of the ith hidden node. Let $\tilde{w}_i$ and β_i be random matrices with suitable dimensions. Choose $\tilde{N} = 500$ and $h(\cdot) = \sigma(\cdot)$. We use ELM [74], [75] to approximate the iterative value function and the iterative control law to implement the present generalized policy iteration algorithm. Choose $\{N_i^2\}$ for the j-iteration. Implement our algorithm for $i = 30$ iterations. The iterative value functions $V_i(x_0)$ by ELM and BP network are shown in Fig. 11(a). We can see that using ELM training, the value function can also converge to its optimum. The plot of $V_i(x_k)$ is shown in Fig. 11(b). By ELM, it takes 1267.49 s to complete implementing the algorithm, while the running time is 5254.84 s by BP network (standard BP algorithm). Using ELM, the initial weights of neural networks are solved directly by Moore–Penrose generalized inverse method (see [17, eq. (21)]), which may lead to faster convergence. The corresponding trajectories of iterative states and control are shown in Fig. 11(c) and (d), respectively.

VI. Conclusion

A generalized policy iteration algorithm is developed for solving infinite horizon approximate optimal control problems of discrete-time nonlinear systems. The present iterative ADP algorithm is initialized by an arbitrary admissible control law. Under the assumption of perfect function approximation, it is proven for the first time that the iterative value function of the generalized policy iteration algorithm is monotonically nonincreasing and converges to the optimal performance index function. Admissibility of the iterative control law is also established. Effective methods are given to relax the initial value function of the present algorithm. Neural networks are employed to implement the generalized policy iteration algorithm to obtain the approximate optimal solution of the HJB equation. Finally, two simulation examples are utilized to illustrate the performance of the present algorithm.

As is known, approximation errors inherently exist during the neural network implementation. We say that the converged iterative value function and iterative control law are approximations to the optimal ones. The property analysis of approximation errors based on iterative θ-ADP algorithm has been investigated in [19]. Hence, the property analysis of the present algorithm with approximation errors will be our future research topic.

References

[1] K. S. Hwang, Y. J. Chen, and C. J. Wu, "Fusion of multiple behaviors using layered reinforcement learning," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 4, pp. 999–1004, Jul. 2012.

[2] T. Landelius, "Reinforcement learning and distributed local model synthesis," Ph.D. dissertation, Dept. Electr. Eng., Linkoping Univ., Linkoping, Sweden, 1997.

[3] H. Modares, F. L. Lewis, and M. B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, Jan. 2014.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

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

[6] 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, Eds. Cambridge, MA, USA: MIT Press, 1991, pp. 67–95.

[7] T. Dierks, B. Brenner, and S. Jagannathan, "Neural network-based optimal control of mobile robot formations with reduced information exchange," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1407–1415, Jul. 2013.

[8] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[9] A. Konar, I. G. Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, "A deterministic improved Q-learning for path planning of a mobile robot," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 5, pp. 1141–1153, Sep. 2013.

[10] 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.

[11] 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.

[12] 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.*, vol. 24, no. 10, pp. 1513–1525, Oct. 2013.

[13] D. Molina, G. K. Venayagamoorthy, J. Liang, and R. G. Harley, "Intelligent local area signals based damping of power system oscillations using virtual generators and approximate dynamic programming," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 498–508, Jan. 2013.

[14] 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.

[15] H. Zhang and F. L. Lewis, "Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics," *Automatica*, vol. 48, no. 7, pp. 1432–1439, Jul. 2012.

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

[17] 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, NY, USA: Van Nostrand Reinhold, 1992, ch. 13.

[18] S. Bhasin *et al.*, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, Jan. 2013.

[19] 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.

[20] Q. Wei and D. Liu, "Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1020–1036, Oct. 2014.

[21] 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, Mar. 2009.

[22] Q. Wei, D. Liu, G. Shi, and Y. Liu, "Multibattery optimal coordination control for home energy management systems via distributed iterative adaptive dynamic programming," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4203–4214, Jul. 2015.

[23] Q. Wei, D. Liu, and G. Shi, "A novel dual iterative Q-learning method for optimal battery management in smart residential environments," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2509–2518, Apr. 2015.

[24] 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.

[25] Q. Wei, D. Liu, and Y. Xu, "Policy iteration optimal tracking control for chaotic systems by adaptive dynamic programming approach," *Chin. Phys. B*, vol. 24, no. 3, Mar. 2015, Art. ID 0305021.

[26] 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.

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

[28] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA, USA: Athena Scientific, 2007.

[29] 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.

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

[31] H. Zhang, Y. Luo, and D. Liu, "The RBF neural network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraint," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sep. 2009.

[32] 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, Mar. 2012.

[33] 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.

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

[35] R. Song, W. Xiao, H. Zhang, and C. Sun, "Adaptive dynamic programming for a class of complex-valued nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1733–1739, Sep. 2014.

[36] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.

[37] Q. Wei and D. Liu, "A novel iterative θ-adaptive dynamic programming for discrete-time nonlinear systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1176–1190, Oct. 2014.

[38] Q. Wei and D. Liu, "Numerical adaptive learning control scheme for discrete-time nonlinear systems," *IET Control Theor. Appl.*, vol. 7, no. 11, pp. 1472–1486, Jul. 2013.

[39] 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," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.

[40] 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.

[41] 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.

[42] X. Yang, D. Liu, and Y. Huang, "Neural-network-based online optimal control for uncertain nonlinear continuous-time systems with control constraints," *IET Control Theor. Appl.*, vol. 7, no. 17, pp. 2037–2047, Nov. 2013.

[43] 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.

[44] K. G. Vamvoudakis and F. L. Lewis, "Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton–Jacobi equations," *Automatica*, vol. 47, no. 8, pp. 1556–1569, Aug. 2011.

[45] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 206–216, Feb. 2013.

[46] 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.

[47] 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.

[48] D. Liu, H. Li, and D. Wang, "Online synchronous approximate optimal learning algorithm for multiplayer nonzero-sum games with unknown dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 8, pp. 1015–1027, Aug. 2014.

[49] Q. Wei and D. Liu, "Data-driven neuro-optimal temperature control of water gas shift reaction using stable iterative adaptive dynamic programming," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6399–6408, Nov. 2014.

[50] Q. Wei and D. Liu, "Stable iterative adaptive dynamic programming algorithm with approximation errors for discrete-time nonlinear systems," *Neural Comput. Appl.*, vol. 24, no. 6, pp. 1355–1367, May 2014.

[51] Q. Wei and D. Liu, "Neural-network-based adaptive optimal tracking control scheme for discrete-time nonlinear systems with approximation errors," *Neurocomputing*, vol. 149, no. 3 pp. 106–115, Feb. 2015.
[52] Q. Wei, F. Wang, D. Liu, and X. Yang, "Finite-approximation-error based discrete-time iterative adaptive dynamic programming," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2820–2833, Dec. 2014.
[53] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 3051–3056, Nov. 2014.
[54] B. Kiumarsi and F. L. Lewis, "Actor-critic based optimal tracking for partially unknown nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 140–151, Jan. 2015.
[55] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M. B. Naghibi-Sistani, "Reinforcement *Q*-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, Apr. 2014.
[56] D. Liu, D. Wang, F. Y. Wang, H. Li, and X. Yang, "Neural-network-based online HJB solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2834–2847, Dec. 2014.
[57] D. Liu, D. Wang, and H. Li, "Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 418–428, Feb. 2014.
[58] P. Pennesi and I. C. Paschalidis, "A distributed actor-critic algorithm and applications to mobile sensor network coordination problems," *IEEE Trans. Autom. Control*, vol. 55, no. 2, pp. 492–497, Feb. 2010.
[59] 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.
[60] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems," *Automatica*, vol. 48, no. 11, pp. 2850–2859, Nov. 2012.
[61] H. Li, D. Liu, and D. Wang, "Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 706–714, Jul. 2014.
[62] 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.
[63] D. Vrabie and F. L. Lewis, "Generalized policy iteration for continuous-time systems," in *Proc. Int. Joint Conf. Neural Netw.*, Atlanta, GA, USA, Jun. 2009, pp. 3224–3231.
[64] D. Vrabie, K. Vamvoudakis, and F. L. Lewis, "Adaptive optimal controllers based on generalized policy iteration in a continuous-time framework," in *Proc. 17th Mediterr. Conf. Control Autom.*, Thessaloniki, Greece, Jun. 2009, pp. 1402–1409.
[65] J. Y. Lee, T. Y. Chun, J. B. Park, and Y. H. Choi, "On generalized policy iteration for continuous-time linear systems," in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, Orlando, FL, USA, Dec. 2011, pp. 1722–1728.
[66] Q. Wei, D. Liu, and X. Yang, "Infinite horizon self-learning optimal control of nonaffine discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 866–879, Apr. 2015.
[67] T. M. Apostol, *Mathematical Analysis*, 2nd ed. Boston, MA, USA: Addison-Wesley, 1974.
[68] 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.
[69] P. J. Werbos, "Consistency of HDP applied to a simple reinforcement learning problem," *Neural Netw.*, vol. 3, no. 2, pp. 179–189, Apr. 1990.
[70] H. R. Maei, C. Szepesvari, S. Bhatnagar, and R. S. Sutton, "Toward off-policy learning control with function approximation," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 719–726.
[71] J. N. Tsitsiklis and B. V. Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Trans. Autom. Control*, vol. 42, no. 5, pp. 674–690, May 1997.
[72] H. Maei *et al.*, "Convergent temporal-difference learning with arbitrary smooth function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2009, pp. 1204–1212.
[73] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 12th ed. New York, NY, USA: Prentice Hall, 2011.
[74] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Aug. 2006.
[75] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

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

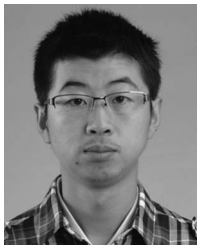
He was a Staff Fellow with General Motors Research and Development Center, Warren, MI, USA, from 1993 to 1995. He was an Assistant Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, from 1995 to 1999. He joined the University of Illinois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

Prof. Liu was a recipient of the Michael J. Birck Fellowship from the University of Notre Dame in 1990, the Harvey N. Davis Distinguished Teaching Award from the Stevens Institute of Technology in 1997, the Faculty Early Career Development Award from the National Science Foundation in 1999, the University Scholar Award from the 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 currently an Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He is a Fellow of the International Neural Network Society.

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, all from Northeastern University, Shenyang, China, in 2002, 2005, and 2008, respectively.

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

Dr. Wei has been an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS since 2014. He is currently an Associate Editor of *Acta Automatica Sinica*.



Pengfei Yan received the B.S. degree in information and computing science from Wuhan University, Wuhan, China, in 2011. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His current research interests include adaptive dynamic programming, data-driven control, adaptive control, and neural-network-based control.