Continuous-Time Distributed Policy Iteration for Multicontroller Nonlinear Systems

Qinglai Wei, *Member, IEEE*, Hongyang Li, Xiong Yang, *Member, IEEE*, and Haibo He, *Fellow, IEEE*

***Abstract*—In this article, a novel distributed policy iteration algorithm is established for infinite horizon optimal control problems of continuous-time nonlinear systems. In each iteration of the developed distributed policy iteration algorithm, only one controller's control law is updated and the other controllers' control laws remain unchanged. The main contribution of the present algorithm is to improve the iterative control law one by one, instead of updating all the control laws in each iteration of the traditional policy iteration algorithms, which effectively releases the computational burden in each iteration. The properties of distributed policy iteration algorithm for continuous-time nonlinear systems are analyzed. The admissibility of the present methods has also been analyzed. Monotonicity, convergence, and optimality have been discussed, which show that the iterative value function is nonincreasingly convergent to the solution of the Hamilton–Jacobi–Bellman equation. Finally, numerical simulations are conducted to illustrate the effectiveness of the proposed method.**

***Index Terms*—Adaptive dynamic programming (ADP), approximate dynamic programming, distributed policy iteration, nonlinear systems, optimal control.**

I. Introduction

OPTIMAL control has attracted many researchers from the control field due to its superiority and practicability [1]–[5]. In the complex industrial process control, lots of real systems are controlled by multiple controllers with each using an individual strategy. The distributed coordination control of multicontroller systems, which avoids high-dimensional controller design of the systems, has attracted compelling attention [6]–[8], where the desired goal of the distributed control is to make all the system states in a cooperative fashion through a series of distributed control laws. Examples of distributed control for multicontroller systems arise from transportation networks, power systems, energy Internet, and multiagent systems [9]–[13]. In fact, many distributed control methods focus on the stability of the nonlinear systems with the distributed control [14]–[18], while the optimality for the multicontroller systems is scarcely analyzed. The difficulty for obtaining the optimal control for the multicontroller systems lies in finding the solutions of the Hamilton–Jacobi–Bellman (HJB) equations. Up to now, there are still no general analytical solutions of HJB equations for nonlinear systems. In multicontroller systems, directly solving the HJB equations is not a good option to obtain the optimal control laws due to the high dimensions of the control. In this situation, many methods have been proposed for achieving the approximate optimal goal.

The adaptive dynamic programming (ADP), which is very effective in achieving the optimal control of nonlinear system [19]–[27], is proposed by Werbos [28], [29]. The ADP has been applied in multicontroller systems for the optimal control laws. In [30], the optimal control laws of decentralized uncertain nonlinear systems with mismatched interconnections were acquired by ADP. In [31], the ADP was employed to solve the optimal multi-ESM scheduling to track ground moving targets. In [32]–[34], neural-optimal control laws of multiplayer nonzero-sum games were obtained via ADP for nonlinear systems in continuous-time and discrete-time cases, respectively. In [35] and [36], ADP was used to obtain the optimal laws of energy management in smart residential microgrids. In [37], the optimal control for multiple-model systems was obtained via discrete-time off-policy ADP. However, it can be seen that traditional ADP methods obtain the optimal multicontroller systems via a centralized control technique, which implies the heavy computation burden if the number of the controller is large. Thus, it is necessary to investigate distributed ADP methods in multicontroller systems for optimal control laws.

Iterative methods that are advantageous in analyzing the performance have been combined with ADP to solve HJB equations indirectly [32], [38]–[42]. Policy iteration, which is one of the iterative ADP algorithms, that has been widely investigated [43]–[48]. To deal with the optimal problems for affine nonlinear systems with continuous-time cases, a policy iteration algorithm was developed under the quadratic utility function in [49]. Then, to deal with the cases where the control inputs are constrained in continuous-time systems, a developed policy iteration algorithm was presented by Abu-Khalaf and Lewis [50]. In [51], with transforming H_∞

Manuscript received January 5, 2020; accepted March 3, 2020. Date of publication April 1, 2020; date of current version April 15, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61722312, and in part by the National Science Foundation under Grant ECCS 1917275. This article was recommended by Associate Editor H. M. Schwartz. *(Corresponding author: Qinglai Wei.)*

Qinglai Wei and Hongyang Li are with The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Qingdao Academy of Intelligent Industries, Qingdao 266109, China (e-mail: qinglai.wei@ia.ac.cn; lihongyang2019@ia.ac.cn).

Xiong Yang is with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: xiong.yang@tju.edu.cn).

Haibo He is with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: haibohe@uri.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2020.2979614.

Digital Object Identifier 10.1109/TCYB.2020.2979614

optimal control to a zero-sum problem, the optimal control laws for the systems with disturbance were achieved by the application of the proposed policy iteration algorithm. In [52], a data-based policy iteration algorithm was established to solve the optimal control problem for continuous-time nonlinear systems with weak coupling. Some distributed policy iteration methods were concerned to achieve the optimum for multicontroller systems, especially for multiagent systems. In [53], a cooperative policy iteration algorithm for graphical games was developed for the synchronization of multiagent systems. In [54], an event-triggered policy iteration algorithm was proposed for distributed linear dynamics. In [55], the distributed optimal output control law for heterogeneous multiagent systems was obtained. It should be pointed out that most previous distributed policy iteration algorithms were focused on the linear multicontroller systems, which were not available for nonlinear systems. Up to now, the investigation about the distributed policy iteration algorithm for multicontroller systems is scarce, and the proposed research is motivated by the situation.

In this article, to solve the optimal control problems for continuous-time nonlinear systems with infinite horizon, a novel distributed policy iteration algorithm is proposed. The main advantage of the present method is to improve the iterative control law one by one, instead of updating all the control laws at each iteration, which effectively releases the computation burden. The contents of this article can be concluded as follows. First, the procedure of the proposed iteration algorithm is introduced. In the distributed policy iteration algorithm, it is shown that only one controller's control law is updated at each iteration, while other control laws are unchanged. Second, some novel property analysis methods are developed for the distributed policy iteration algorithm. Although only one controller is updated in each iteration, all of the iterative control laws in any iteration are admissible for the system. Finally, analysis about the convergence is given, which can prove that the iterative value functions can converge to the optimum with monotonically nonincreasing feature.

The remainder of this article is given as follows. In Section II, the problem formulation is described. In Section III, the continuous-time distributed policy iteration algorithm is introduced and some proofs about the admissibility, convergence, and optimality properties are also shown in this section. Then, in Section IV, simulation results are utilized to demonstrate the effectiveness of the developed algorithm. The conclusions are finally drawn in Section V.

II. Problem Formulations

Consider the following continuous-time multicontroller nonlinear system:

$$\dot{x} = F(x, u_1, \ldots, u_N) \tag{1}$$

where the system state is denoted by $x = x(t) \in \mathbb{R}^n$, and $u_i = u_i(t) \in \mathbb{R}^{m_i}$, $i = 1, 2, \ldots, N$, represents the control inputs. $F(\cdot)$ is regarded as the system function. N stands for the number of the controllers, which is usually a positive integer. Let x_0 be the initial condition of the nonlinear system. Some assumptions are given in the following for further analysis.

Assumption 1: The system (1) is controllable, and the system states all belong to a compact set where the origin is contained; the system function $F(x, u_1, \ldots, u_n)$ is Lipschitz continuous for x and u_i, $i = 1, 2, \ldots, N$; the equilibrium point of the system (1) is $x = 0$, when the control inputs satisfy $u = 0$, that is, $F(0, 0, \ldots, 0) = 0$; and the multicontrol law $u_i(x)$, $i = 1, 2, \ldots, N$, is continuous on Ω and $u_i = u_i(x) = 0$ always holds for $x = 0$.

To analyze the optimal control problem of system (1), the performance index function is given with the following definition:

$$J(x) = \int_t^\infty U(x(s), u_1(s), \ldots, u_N(s))\mathrm{d}s \tag{2}$$

where $U(x, u_1, \ldots, u_N)$ represents the utility function and positive definite for x and u_i, $i = 1, 2, \ldots, N$.

The admissible control laws of multicontrollers can be defined as $\mu_i \in \Psi(\Omega)$, $i = 1, 2, \ldots, N$, and $\Psi(\Omega)$ can be considered as the set of admissible controls on Ω. Under the admissible controls, the value function is given as

$$V(x) = \int_t^\infty U(x(s), \mu_1(x(s)), \mu_2(x(s)), \ldots, \mu_N(x(s)))\mathrm{d}s. \tag{3}$$

If the value function is continuously differentiable respect to t, it can be transformed into the following form which is called the nonlinear Lyapunov equation:

$$U(x, \mu_1, \ldots, \mu_N) + \left(\frac{\partial V(x)}{\partial x}\right)^{\mathsf{T}} F(x, \mu_1, \ldots, \mu_N) = 0. \tag{4}$$

Based on the definition in (2), the optimal performance index function is defined as

$$J^*(x) = \min_{\mu_1, \ldots, \mu_N \in \Psi(\Omega)} \left\{ \int_t^\infty U(x(s), \mu_1(s), \ldots, \mu_N(s))\mathrm{d}s \right\} \tag{5}$$

which can satisfy the HJB equation with $J^*(0) = 0$, and the following equation can be derived:

$$\begin{aligned} &\min_{\mu_1, \ldots, \mu_N} \left\{ U(x, \mu_1, \ldots, \mu_N) + \left(\frac{\partial J^*(x)}{\partial x}\right)^{\mathsf{T}} F(x, \mu_1, \ldots, \mu_N) \right\} \\ &\quad = U(x, \mu_1^*(x), \ldots, \mu_N^*(x)) + \left(\frac{\partial J^*(x)}{\partial x}\right)^{\mathsf{T}} \\ &\quad\quad \times F\big(x, \mu_1^*(x), \ldots, \mu_N^*(x)\big) \\ &\quad = 0 \end{aligned} \tag{6}$$

where $\mu_1^*(x), \ldots, \mu_N^*(x)$ are the optimal control laws. Generally, it is almost impossible to obtain $J^*(x)$ by directly solving the HJB equations, especially for multicontroller nonlinear systems. Hence, developing a novel distributed policy iteration algorithm to overcome this difficulty is very necessary.

III. Continuous-Time Distributed Policy Iteration: Derivations and Properties

In this section, the derivations of the continuous-time distributed policy iteration algorithm for multicontroller nonlinear systems will be discussed. Furthermore, some new methods to analyze the convergence and monotonicity will be introduced and the admissibility of the multicontrol laws will also be proven.

A. Derivations of the Continuous-Time Distributed Policy Iteration Algorithm

Let $v_1^0(x), \ldots, v_N^0(x)$ $\forall x \in \mathbb{R}^n$ be arbitrary admissible control laws. Let $V^0(x)$ denote the initial iterative value function, such that

$$U\Big(x, v_1^0(x), \ldots, v_N^0(x)\Big) + \left(\frac{\partial V^0(x)}{\partial x}\right)^{\mathsf T} \times F\Big(x, v_1^0(x), \ldots, v_N^0(x)\Big) = 0. \tag{7}$$

Let $k = 1$ and $\tau_1 \in \mathcal{N}, \mathcal{N} = \{1, 2, \ldots, N\}$. Let $u_{(i)} = \{u_j{:}j \in \mathcal{N}, j \neq i\}$. Then, we have $U(x, u_{\tau_1}, u_{(\tau_1)}) = U(x, u_1, \ldots, u_N)$. For $k = 1$, the control law $v_{\tau_1}^1(x)$ can be calculated by

$$v_{\tau_1}^1(x) = \arg\min_{u_{\tau_1}} \left\{ U\Big(x, u_{\tau_1}, v_{(\tau_1)}^0(x)\Big) + \left(\frac{\partial V^0(x)}{\partial x}\right)^{\mathsf T} F\Big(x, u_{\tau_1}, v_{(\tau_1)}^0(x)\Big) \right\}. \tag{8}$$

Let $v_j^1(x) = v_j^0(x)$, for all $j \in \mathcal{N}$ and $j \neq \tau_1$. According to $v_1^1(x), v_2^1(x), \ldots, v_N^1(x)$, the corresponding value function $V^1(x)$ is calculated by

$$U\Big(x, v_1^1(x), \ldots, v_N^1(x)\Big) + \left(\frac{\partial V^1(x)}{\partial x}\right)^{\mathsf T} \times F\Big(x, v_1^1(x), \ldots, v_N^1(x)\Big) = 0. \tag{9}$$

For $k = 1, 2, \ldots$ let $\tau_k \in \mathcal{N}$ and $U(x, u_{\tau_k}, u_{(\tau_k)}) = U(x, u_1, \ldots, u_N)$, the iterative control law $v_{\tau_k}^k(x)$ can be derived by

$$v_{\tau_k}^k(x) = \arg\min_{u_{\tau_k}} \left\{ U\Big(x, u_{\tau_k}, v_{(\tau_k)}^{k-1}(x)\Big) + \left(\frac{\partial V^{k-1}(x)}{\partial x}\right)^{\mathsf T} F\Big(x, u_{\tau_k}, v_{(\tau_k)}^{k-1}(x)\Big) \right\}. \tag{10}$$

Let $v_j^k(x) = v_j^{k-1}(x)$, for all $j \in \mathcal{N}$ and $j \neq \tau_k$. According to $v_1^k(x), v_2^k(x), \ldots, v_N^k(x)$, the iterative value function $V^k(x)$ is updated by

$$U\Big(x, v_1^k(x), \ldots, v_N^k(x)\Big) + \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf T} \times F\Big(x, v_1^k(x), \ldots, v_N^k(x)\Big) = 0. \tag{11}$$

Then, we can obtain the distributed policy iteration algorithm as Algorithm 1.

In this article, the function $J^*(x)$ denotes the optimal performance index function under the optimal control laws $u_1^*(x), u_2^*(x), \ldots, u_N^*(x)$. For $k = 0, 1, \ldots$, the function $V^k(x)$ is used in the iteration process, which denotes the iterative value function under the iterative control laws $v_1^k(x), v_2^k(x), \ldots, v_N^k(x)$. In the following, the properties of $V^k(x)$ will be analyzed and the relationship between $V^k(x)$ and $J^*(x)$ will be proven.

Algorithm 1 Distributed Policy Iteration Algorithm for Multicontroller Nonlinear Systems

```
Initialization:
    Choose randomly an admissible control law $v_i^0(x), i = 1, \ldots, N$;
    Choose a computation precision $\varepsilon$.
Iteration:
 1: Let the iteration index $k = 0$. Construct an iterative value function $V^0(x)$ to satisfy (7);
 2: Let $k = k + 1$, choose $\tau_k \in \mathcal{N}$ randomly. Do Policy Improvement
        $v_{\tau_k}^k(x) = \arg\min_{u_{\tau_k}} \Big\{ U\big(x, u_{\tau_k}, v_{(\tau_k)}^{k-1}(x)\big) + \Big(\frac{\partial V^{k-1}(x)}{\partial x}\Big)^{\mathsf T} F\big(x, u_{\tau_k}, v_{(\tau_k)}^{k-1}(x)\big) \Big\}$;
 3: Do Policy Evaluation
        $U(x, v_1^k(x), \ldots, v_N^k(x)) + \Big(\frac{\partial V^k(x)}{\partial x}\Big)^{\mathsf T} \times F(x, v_1^k(x), \ldots, v_N^k(x)) = 0$;
 4: If $V^{k-1}(x) - V^k(x) > \varepsilon$, goto Step 2.
 5: return $v_1^k(x), \ldots, v_N^k(x)$, $V^k(x)$.
```

B. Property Analysis

In this section, the corresponding properties, such as convergence and admissibility of the distributed policy iteration algorithm, are analyzed. For traditional policy iteration algorithms [43], [45], [49], [50], [58], [59], all the control laws of the system must be updated in each iteration simultaneously to guarantee the convergence of $V^k(x)$ and the admissibility of the control laws. However, for distributed policy iteration algorithm (7)–(11), only one control input is updated such that the traditional analysis methods are unavailable for the distributed policy iteration algorithm. Thus, some novel analysis methods will be established in this section. First, the admissibility of the distributed iterative control laws will be analyzed and some lemmas are given in the following.

Lemma 1: If $v_1^k(x), \ldots, v_N^k(x)$, $k = 0, 1, \ldots$, are admissible control laws for system (1), there exists a value function $V^k(x)$ to satisfy

$$U\Big(x, v_1^k(x), \ldots, v_N^k(x)\Big) + \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf T} \times F\Big(x, v_1^k(x), \ldots, v_N^k(x)\Big) = 0. \tag{12}$$

Theorem 1: For $k = 0, 1, \ldots$, the iterative value function $V^k(x)$ and the distributed iterative control laws $v_1^k(x), \ldots, v_N^k(x)$ can be obtained by (7)–(11). If control laws

$v_1^k(x), \dots, v_N^k$ are admissible for nonlinear system (1), then $v_1^{k+1}(x), \dots, v_N^{k+1}$ are admissible control laws.

Proof: For $k = 0, 1, \dots$, as $v_1^k(x), \dots, v_N^k$ are admissible control laws, (12) is always satisfied based on Lemma 1. Letting $\tau_{k+1} \in \mathcal{N}$, $v_{\tau_{k+1}}^{k+1}(x)$ can be obtained by (10), which is expressed as

$$v_{\tau_{k+1}}^{k+1}(x) = \arg\min_{u_{\tau_{k+1}}} \Bigg\{ U\Big(x, u_{\tau_{k+1}}, v_{(\tau_{k+1})}^{k}(x)\Big) + \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F\Big(x, u_{\tau_{k+1}}, v_{(\tau_{k+1})}^{k}(x)\Big)\Bigg\}. \tag{13}$$

According to (12), it can be derived that

$$\left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F\Big(x, v_1^{k+1}(x), \dots, v_N^{k+1}(x)\Big) + U\Big(x, v_1^{k+1}(x), \dots, v_N^{k+1}(x)\Big) \le 0 \tag{14}$$

where $v_{\tau_{k+1}}^{k+1}(x)$ is obtained by (13) and $v_j^{k+1}(x) = v_j^k(x)$, for all $j \in \mathcal{N}$ and $j \ne \tau_{k+1}$.

According to Assumption 1, it can be derived that $V^k(x) = \int_t^\infty U(x(s), v_1^k(x(s)), \dots, v_N^k(x(s)))\mathrm{d}s$ is always positive due to the characteristics of the utility function. Then, $V^k(x)$ is said to be a positive-definite function. Choose $V^k(x)$ as the Lyapunov function candidate. Based on (14), we have the following inequality by calculating the derivative of $V^k(x)$ along $(v_1^{k+1}(x), \dots, v_N^{k+1}(x))$:

$$\begin{aligned}\dot{V}^k(x) &= \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F\Big(x, v_1^{k+1}(x), \dots, v_N^{k+1}(x)\Big)\\ &\le -U\Big(x, v_1^{k+1}(x), \dots, v_N^{k+1}(x)\Big).\end{aligned} \tag{15}$$

Thus, $v_1^{k+1}(x), \dots, v_N^{k+1}(x)$ are stable control laws for system (1). Then, it can be derived that $\lim_{t\to\infty} U(x, v_1^{k+1}(x), \dots, v_N^{k+1}(x)) = 0$.

Let $\Upsilon^{k+1}(x)$, $k = 0, 1, \dots$, is a value function, such that

$$\Upsilon^{k+1}(x) = \int_t^\infty U\Big(x(s), v_1^{k+1}(x(s)), \dots, v_N^{k+1}(x(s))\Big)\mathrm{d}s. \tag{16}$$

Next, we will prove that $\Upsilon^{k+1}(x)$ $\forall x \in \mathbb{R}^n$, is finite under the control laws $v_1^{k+1}(x), \dots, v_N^{k+1}(x)$. Taking the derivative of $\Upsilon^{k+1}(x)$ along time t, we have

$$\dot{\Upsilon}^{k+1}(x) = -U\Big(x, v_1^{k+1}(x), \dots, v_N^{k+1}(x)\Big). \tag{17}$$

Considering (15) and (17), we can obtain

$$\dot{V}^k(x) \le \dot{\Upsilon}^{k+1}(x) \quad \forall x \in \mathbb{R}^n. \tag{18}$$

As $v_1^k(x), \dots, v_N^k(x)$ are admissible control laws, define $V^k(x(\infty)) = \lim_{t\to\infty} V^k(x(t)) = 0$. From (16), we know that $\Upsilon^{k+1}(x(\infty)) = \lim_{t\to\infty} \Upsilon^{k+1}(x(t)) = 0$. According to (12) and (16), we can derive

$$\begin{aligned}&\int_t^\infty \frac{\mathrm{d}V^k(x(s))}{\mathrm{d}s}\mathrm{d}s\\ &= V^k(x(\infty)) - V^k(x(t))\\ &\le \int_t^\infty \frac{d\Upsilon^{k+1}(x(s))}{\mathrm{d}s}\mathrm{d}s\\ &= \Upsilon^{k+1}(x(\infty)) - \Upsilon^{k+1}(x(t))\\ &= -\int_t^\infty U\Big(x(s), v_1^{k+1}(x(s)), \dots, v_N^{k+1}(x(s))\Big)\mathrm{d}s.\end{aligned} \tag{19}$$

Then, we can obtain

$$\int_t^\infty U\Big(x(s), v_1^{k+1}(x(s)), \dots, v_N^{k+1}(x(s))\Big)\mathrm{d}s \le V^k(x(t)) \tag{20}$$

which shows that the distributed control laws $v_1^{k+1}(x), \dots, v_N^{k+1}(x)$ are admissible for the system (1). The proof is complete. ■

Furthermore, the properties for the iterative value function $V^k(x)$ will be discussed in the following theorem.

Theorem 2: For $k = 0, 1, \dots$, let the iterative value function $V^k(x)$ and the distributed iterative control laws $v_1^k(x), \dots, v_N^k(x)$ be obtained by (7)–(11). If $v_1^0(x), \dots, v_N^0(x)$ are admissible control laws, the iterative value function $V^k(x)$, $k = 0, 1, \dots$, is monotonically nonincreasing as k increases, that is

$$V^{k+1}(x) \le V^k(x) \quad \forall x \in \mathbb{R}^n. \tag{21}$$

Proof: Consider $k = 0$. For the admissible control laws $v_1^0(x), \dots, v_N^0(x)$, according to Theorem 1, the iterative control laws $v_1^1(x), \dots, v_N^1(x)$ are admissible control laws. According to (16), it can be easily derived that $V^1(x) = \Upsilon^1(x)$. Considering the derivative of $V^0(x)$ along $v_1^1(x), \dots, v_N^1(x)$, according to (14), we can obtain

$$\begin{aligned}\dot{V}^0(x) &= \left(\frac{\partial V^0(x)}{\partial x}\right)^{\mathsf{T}} F\Big(x, v_1^1(x), \dots, v_N^1(x)\Big)\\ &\le -U\Big(x, v_1^1(x), \dots, v_N^1(x)\Big)\end{aligned} \tag{22}$$

such that

$$\dot{V}^0(x) \le \dot{V}^1(x) \quad \forall x \in \mathbb{R}^n. \tag{23}$$

According to Theorem 1, as $v_1^0(x), v_2^0(x), \dots, v_N^0(x)$ are admissible control laws, then $v_1^1(x), v_2^1(x), \dots, v_N^1(x)$ are admissible. These indicate that $V^0(x(\infty)) = 0$ and $V^1(x(\infty)) = 0$, which imply that $V^0(x(t)) \ge V^1(x(t))$ $\forall x \in \mathbb{R}^n$. By the implementation of mathematical induction, (21) can be guaranteed to hold for any $k = 0, 1, \dots$ The proof is complete. ■

According to Theorem 2, increasing iterative index k, the iterative value function is monotonically nonincreasing. Next, the optimality of the iterative value function will be discussed.

Theorem 3: For $k = 0, 1, \dots$, the iterative value function $V^k(x)$ and the distributed iterative control laws $v_1^k(x), \dots, v_N^k(x)$ are derived by (7)–(11). Then, $V^k(x)$ converges to a suboptimal performance index function as $k \to \infty$.

Proof: According to Lemma 1, we can derive

$$V^k(x) = \int_t^\infty U\Big(x(s), v_1^k(x(s)), \dots, v_N^k(x(s))\Big)\mathrm{d}s. \tag{24}$$

As the utility function $U(x, u_1, \ldots, u_N)$ is a positive-definite function for x and u_i, $i = 1, 2, \ldots, N$, according to Assumption 1, we know that $V^k(x) = 0$ for $x = 0$ and $V^k(x) > 0$ for all $x \neq 0$. Hence, for $k = 0, 1, \ldots$, $V^k(x)$ is a positive-definite function for x.

For $k \to \infty$, there must exist a controller which is improved for infinite times. Without loss of generality, controller τ^o, $\tau^o \in \mathcal{N}$, is assumed to improve for infinite times. Let $\mathcal{K}$ denote a set of iteration indices, which is defined as

$$\mathcal{K} = \left\{k | k = 0, 1, \ldots, \tau_k = \tau^o, \tau^o \in \mathcal{N}\right\}. \tag{25}$$

Let $\kappa_j \in \mathcal{K}$, $j = 0, 1, \ldots$ Without loss of generality, let $\kappa_0 < \kappa_1 < \cdots$ According to Theorem 2, $V^k(x)$ has been proved to be nonincreasing as $k \to \infty$ and have the lower limit, which is defined as $V^\infty(x)$, that is

$$V^\infty(x) = \lim_{k \to \infty} V^k(x). \tag{26}$$

By considering (9) and (11), for $k = 0, 1, \ldots$ and $\Delta T \geq 0$, it can be derived that

$$\begin{aligned} V^k(x) = &\int_t^{t+\Delta T} U\Big(x(s), v_{\tau_k}^k(x(s)), v_{(\tau_k)}^k(x(s))\Big)\mathrm{d}s \\ &+ V^k(x(t + \Delta T)). \end{aligned} \tag{27}$$

According to (27), for $\kappa_j \in \mathcal{K}$, $j = 0, 1, \ldots$, consider a new iterative value function Γ^{κ_j+1} as

$$\begin{aligned} \Gamma^{\kappa_j+1}(x) = &\int_t^{t+\Delta T} U(x(s), v_{\tau^o}^{\kappa_j+1}(x(s)), v_{(\tau^o)}^{\kappa_j}(x(s)))\mathrm{d}s \\ &+ V^{\kappa_j}(x(t + \Delta T)) \\ = &\int_t^{t+\Delta T} U\Big(x(s), v_{\tau^o}^{\kappa_j+1}(x(s)), v_{(\tau^o)}^{\kappa_j+1}(x(s))\Big)\mathrm{d}s \\ &+ V^{\kappa_j}(x(t + \Delta T)) \end{aligned} \tag{28}$$

where $v_{\tau^o}^{\kappa_j+1}(x)$ is defined by (10) for $k = \kappa_j$ and $v_{(\tau^o)}^{\kappa_j+1}(x) = v_{(\tau^o)}^{\kappa_j}(x)$. According to Theorem 2, the following conclusion can be drawn:

$$V^{\kappa_j+1}(x) \leq \Gamma^{\kappa_j+1}(x). \tag{29}$$

If $k \to \infty$, it is obvious that $j \to \infty$ and $\kappa_j \to \infty$. Then, for $k \to \infty$, we have

$$\begin{aligned} V^\infty(x) \leq &\int_t^{t+\Delta T} U\Big(x(s), v_{\tau^o}^\infty(x(s)), v_{(\tau^o)}^\infty(x(s))\Big)\mathrm{d}s \\ &+ V^\infty(x(t + \Delta T)). \end{aligned} \tag{30}$$

Define ε as a positive constant, that is, $\varepsilon > 0$. Due to

$$\lim_{j \to \infty} V^{\kappa_j}(x) = \lim_{k \to \infty} V^k(x) = V^\infty(x) \tag{31}$$

there must be a positive integer $\kappa_\rho < \infty$, such that

$$V^{\kappa_\rho}(x) - \varepsilon \leq V^\infty(x) \leq V^{\kappa_\rho}(x). \tag{32}$$

Based on (27) and (32), we have

$$\begin{aligned} V^\infty(x) \geq &\int_t^{t+\Delta T} U(x(s), v_{\tau^o}^{\kappa_\rho}(x(s)), v_{(\tau^o)}^{\kappa_\rho}(x(s)))\mathrm{d}s \\ &+ V^{\kappa_\rho}(x(t + \Delta T)) - \varepsilon \\ \geq &\int_t^{t+\Delta T} U(x(s), v_{\tau^o}^{\kappa_\rho}(x(s)), v_{(\tau^o)}^{\kappa_\rho}(x(s)))\mathrm{d}s \\ &+ V^\infty(x(t + \Delta T)) - \varepsilon \\ = &\int_t^{t+\Delta T} U\Big(x(s), v_{\tau^o}^{\kappa_\rho}(x(s)), v_{(\tau^o)}^\infty(x(s))\Big)\mathrm{d}s \\ &+ V^\infty(x(t + \Delta T)) - \varepsilon. \end{aligned} \tag{33}$$

As ε is arbitrary, we have

$$\begin{aligned} V^\infty(x) \geq &\int_t^{t+\Delta T} U\Big(x(s), v_{\tau^o}^{\kappa_\rho}(x(s)), v_{(\tau^o)}^\infty(x(s))\Big)\mathrm{d}s \\ &+ V^\infty(x(t + \Delta T)). \end{aligned} \tag{34}$$

According to (10), define $v_{\tau^o}^\infty$ as

$$\begin{aligned} v_{\tau^o}^\infty(x) = \arg\min_{u_{\tau^o}} \Bigg\{ &U\Big(x, u_{\tau^o}, v_{(\tau^o)}^\infty(x)\Big) \\ &+ \left(\frac{\partial V^\infty(x)}{\partial x}\right)^{\mathsf{T}} F(x, u_{\tau^o}, v_{(\tau^o)}^\infty(x)) \Bigg\}. \end{aligned} \tag{35}$$

According to (34) and (35), we have

$$\begin{aligned} V^\infty(x) \geq &\int_t^{t+\Delta T} U\Big(x(s), v_{\tau^o}^\infty(x(s)), v_{(\tau^o)}^\infty(x(s))\Big)\mathrm{d}s \\ &+ V^\infty(x(t + \Delta T)). \end{aligned} \tag{36}$$

Combining (30) and (36), we can obtain

$$\begin{aligned} V^\infty(x) = &\int_t^{t+\Delta T} U\Big(x(s), v_{\tau^o}^\infty(x(s)), v_{(\tau^o)}^\infty(x(s))\Big)\mathrm{d}s \\ &+ V^\infty(x(t + \Delta T)). \end{aligned} \tag{37}$$

As $v_1^k(x), v_2^k(x), \ldots, v_N^k(x)$, $k = 0, 1, \ldots$, are admissible control laws, which indicates $V^\infty(x(k + \Delta T)) = 0$ as $\Delta T \to \infty$. According to (35), the following equation of optimality can be derived with $\Delta T \to \infty$:

$$\begin{aligned} 0 = &U\Big(x, v_{\tau^o}^\infty(x), v_{(\tau^o)}^\infty(x)\Big) + \left(\frac{\partial V^\infty(x)}{\partial x}\right)^{\mathsf{T}} \\ &\times F\Big(x, v_{\tau^o}^\infty(x), v_{(\tau^o)}^\infty(x)\Big) \\ = &\min_{u_{\tau^o}} \Bigg\{ U(x, u_{\tau^o}, v_{(\tau^o)}^\infty(x)) + \left(\frac{\partial V^\infty(x)}{\partial x}\right)^{\mathsf{T}} \\ &\times F\Big(x, u_{\tau^o}, v_{(\tau^o)}^\infty(x)\Big) \Bigg\}. \end{aligned} \tag{38}$$

Thus, as $k \to \infty$, $V^k(x)$ converges to a suboptimal performance index function. The proof is complete. ■

Remark 1: It shows in Theorem 3 that as $k \to \infty$, $V^\infty(x)$, which is the limit of the iterative value function and defined as $V^\infty(x) = \lim_{k \to \infty} V^k(x)$, converges to the solution of the HJB equation in (38), not in (6). Thus, the suboptimal performance index is actually achieved as $k \to \infty$.

In order to obtain the global convergence analysis, a new criterion is necessary. Before analyzing the global convergence property, some denotations should be defined, such as $\mathcal{T}_i = \{k \mid \tau_k = i, i \in \mathcal{N}\}$ and π_i, which describes how many elements are in $\mathcal{T}_i$.

Theorem 4 (Global Convergence Property): For $k = 0, 1, \ldots$, the iterative value function $V^k(x)$ and the distributed iterative control laws $v_1^k(x), \ldots, v_N^k(x)$ are obtained by (7)–(11). If $\pi_i \to \infty\ \forall\, i \in \mathcal{N}$, as $k \to \infty$, $V^k(x)$ is convergent to the optimal performance index function, that is

$$\lim_{k\to\infty} V^k(x) = J^*(x). \tag{39}$$

Proof: The proof is given by the following two steps.

1) Show that the iterative value function $V^k(x)$ will satisfy

$$\lim_{k\to\infty} V^k(x) \geq J^*(x). \tag{40}$$

Letting

$$\begin{aligned}&\Big(\nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x)\Big)\\ &\quad = \arg \min_{u_1,u_2,\ldots,u_N} \Bigg\{ U(x, u_1, \ldots, u_N)\\ &\qquad\qquad + \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F(x, u_1, \ldots, u_N)\Bigg\}\end{aligned} \tag{41}$$

according to (7)–(11), for any $\tau_k \in \mathcal{N}$, $k = 0, 1, \ldots$, we can derive

$$\begin{aligned}&\min_{u_1,u_2,\ldots,u_N} \Bigg\{ U(x, u_1, \ldots, u_N) + \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F(x, u_1, \ldots, u_N)\Bigg\}\\ &\quad = U\Big(x, \nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x)\Big)\\ &\qquad + \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F\Big(x, \nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x)\Big)\\ &\quad \leq \min_{u_{\tau_{k+1}}} \Bigg\{ U\Big(x, u_{\tau_{k+1}}, u^k_{(\tau_{k+1})}(x)\Big)\\ &\qquad\qquad + \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F\Big(x, u_{\tau_{k+1}}, u^k_{(\tau_{k+1})}(x)\Big)\Bigg\}\\ &\quad = U\Big(x, \nu^{k+1}_{\tau_{k+1}}(x), \nu^k_{(\tau_{k+1})}(x)\Big)\\ &\qquad + \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F(x, \nu^{k+1}_{\tau_{k+1}}(x), \nu^k_{(\tau_{k+1})}(x))\\ &\quad \leq 0\\ &\quad = U\big(x, u_1^*(x), \ldots, u_N^*(x)\big)\\ &\qquad + \left(\frac{\partial J^*(x)}{\partial x}\right)^{\mathsf{T}} F(x, u_1^*(x), \ldots, u_N^*(x))\\ &\quad = \min_{u_1,u_2,\ldots,u_N} \Bigg\{ U(x, u_1, \ldots, u_N)\\ &\qquad\qquad + \left(\frac{\partial J^*(x)}{\partial x}\right)^{\mathsf{T}} F(x, u_1, \ldots, u_N)\Bigg\}.\end{aligned} \tag{42}$$

Taking derivatives of $V^k(x)$ and $J^*(x)$ along $(\nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x))$, according to (42), we obtain

$$\begin{aligned}\dot{V}^k(x) - \dot{J}^*(x) &= \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F\Big(\nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x)\Big)\\ &\quad + U\Big(\nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x)\Big)\\ &\quad - \left(\frac{\partial J^*(x)}{\partial x}\right)^{\mathsf{T}} F(\nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x))\\ &\quad - U\Big(\nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x)\Big)\\ &= \min_{u_1,u_2,\ldots,u_N} \Bigg\{ U(x, u_1, \ldots, u_N)\\ &\qquad + \left(\frac{\partial V^k(x)}{\partial x}\right)^{\mathsf{T}} F(x, u_1, \ldots, u_N)\Bigg\}\\ &\quad - \min_{u_1,u_2,\ldots,u_N} \Bigg\{ U(x, u_1, \ldots, u_N)\\ &\qquad + \left(\frac{\partial J^*(x)}{\partial x}\right)^{\mathsf{T}} F(x, u_1, \ldots, u_N)\Bigg\}\\ &\quad + \min_{u_1,u_2,\ldots,u_N} \Bigg\{ U(x, u_1, \ldots, u_N)\\ &\qquad + \left(\frac{\partial J^*(x)}{\partial x}\right)^{\mathsf{T}} F(x, u_1, \ldots, u_N)\Bigg\}\\ &\quad - \Bigg(\left(\frac{\partial J^*(x)}{\partial x}\right)^{\mathsf{T}} F(\nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x))\\ &\qquad + U\Big(\nu_1^{k+1}(x), \ldots, \nu_N^{k+1}(x)\Big)\Bigg)\\ &\leq 0.\end{aligned} \tag{43}$$

Based on (43), we know that $\int_t^\infty \dot{V}^k(x(s))\mathrm{d}s \leq \int_t^\infty \dot{J}^*(x(s))\mathrm{d}s$, which is derived as $V^k(x(\infty)) - V^k(x(t)) \leq J^*(x(\infty)) - J^*(x(t))$. According to Theorem 1, the iterative control laws $\nu_1^k(x), \ldots, \nu_N^k(x)$ are admissible for system (1), which indicates that $V^k(x(\infty)) = 0$. The optimal control laws $u_1^*(x), u_2^*(x), \ldots, u_N^*(x)$ are admissible, which indicates that $J^*(x(\infty)) = 0$. Thus, we can derive $V^k(x) \geq J^*(x)$ $\forall k = 0, 1, \ldots$ As $k \to \infty$, (40) is always satisfied.

2) Show that the iterative value function $V^k(x)$ can satisfy

$$\lim_{k\to\infty} V^k(x) \leq J^*(x). \tag{44}$$

For $\pi_i \to \infty$, it shows that $k \to \infty$. For $k \to \infty$ and $\pi_i \to \infty$, according to Theorem 2, define

$$\mathcal{V}^\infty(x) := \lim_{k\to\infty} V^k(x). \tag{45}$$

As $\pi_i \to \infty\ \forall\, i \in \mathcal{N}$, we can derive

$$\begin{aligned}\nu_i^\infty(x) = \arg\min_{u_i} \Bigg\{ & U\Big(x, u_i, \nu^\infty_{(i)}(x)\Big)\\ & + \frac{\partial \mathcal{V}^\infty(x)}{\partial x} F(x, u_i, \nu^\infty_{(i)}(x))\Bigg\}.\end{aligned} \tag{46}$$

According to (11) and (46), we can derive

$$\begin{aligned}&U\big(x, \nu_1^\infty(x), \ldots, \nu_N^\infty(x)\big)\\ &\quad + \frac{\partial \mathcal{V}^\infty(x)}{\partial x} F\big(x, \nu_1^\infty(x), \ldots, \nu_N^\infty(x)\big)\\ &\quad = \min_{u_1} \Bigg\{ U\big(x, u_1, \nu_2^\infty(x), \ldots, \nu_N^\infty(x)\big)\\ &\qquad\qquad + \frac{\partial \mathcal{V}^\infty(x)}{\partial x} F\big(x, u_1, \nu_2^\infty(x), \ldots, \nu_N^\infty(x)\big)\Bigg\}\\ &\quad = \min_{u_2} \Bigg\{ \min_{u_1} \Bigg\{ U\big(x, u_1, u_2, \nu_3^\infty(x), \ldots, \nu_N^\infty(x)\big)\\ &\qquad\qquad + \frac{\partial \mathcal{V}^\infty(x)}{\partial x} F\big(x, u_1, u_2, \nu_3^\infty(x), \ldots, \nu_N^\infty(x)\big)\Bigg\}\Bigg\}\end{aligned}$$

$$
\begin{aligned}
&= \min_{u_N}\left\{\cdots\left\{\min_{u_2}\left\{\min_{u_1}\left\{U(x, u_1, \ldots, u_N)\right.\right.\right.\right.\\
&\qquad\qquad \left.\left.\left.\left. + \frac{\partial \mathcal{V}^{\infty}(x)}{\partial x}F(x, u_1, \ldots, u_N)\right\}\right\}\right\}\cdots\right\}\\
&= \min_{u_1, u_2, \ldots, u_N}\left\{U(x, u_1, \ldots, u_N)\right.\\
&\qquad\qquad \left. + \frac{\partial \mathcal{V}^{\infty}(x)}{\partial x}F(x, u_1, \ldots, u_N)\right\}\\
&= 0.
\end{aligned}
\tag{47}
$$

According to (47), we have

$$
\begin{aligned}
\mathcal{V}^{\infty}(x) = &\int_t^{t+\Delta T} U\big(x(s), v_1^{\infty}(x(s)), \ldots, v_N^{\infty}(x(s))\big)\mathrm{d}s\\
&+ \mathcal{V}^{\infty}(x(t+\Delta T)).
\end{aligned}
\tag{48}
$$

Let $\mu_1(x), \ldots, \mu_N(x)$ be admissible control laws for system (1). For $k = 0, 1, \ldots$, we define

$$
\begin{aligned}
\Phi^{k+1}(x) = &\int_t^{t+\Delta T} U(x(s), \mu_1(x), \ldots, \mu_N(x))\mathrm{d}s\\
&+ \Phi^k(x(t+\Delta T))
\end{aligned}
\tag{49}
$$

where $\Phi^0(x) = \mathcal{V}^{\infty}(x)$. Next, we will prove that

$$
\Phi^{k+1}(x) \geq \mathcal{V}^{\infty}(x) \quad \forall k = 0, 1, \ldots
\tag{50}
$$

For $k = 0$, it is easy to obtain

$$
\begin{aligned}
\Phi^1(x) = &\int_t^{t+\Delta T} U(x(s), \mu_1(x(s)), \ldots, \mu_N(x(s)))\mathrm{d}s\\
&+ \mathcal{V}^{\infty}(x(t+\Delta T))\\
\geq &\min_{u_1, u_2, \ldots, u_N}\left\{\int_t^{t+\Delta T} U(x(s), u_1, \ldots, u_N)\mathrm{d}s\right.\\
&\qquad\qquad \left. + \mathcal{V}^{\infty}(x(t+\Delta T))\right\}\\
= &\mathcal{V}^{\infty}(x).
\end{aligned}
\tag{51}
$$

If (50) is satisfied for $k = l - 1$, $l = 1, 2, \ldots$ When $k = l$, we can obtain

$$
\begin{aligned}
\Phi^{l+1}(x) \geq &\int_t^{t+\Delta T} U(x(s), \mu_1(x(s)), \ldots, \mu_N(x(s)))\mathrm{d}s\\
&+ \mathcal{V}^{\infty}(x(t+\Delta T))\\
\geq &\mathcal{V}^{\infty}(x).
\end{aligned}
\tag{52}
$$

By using mathematical induction, the above result (50) can be proven.

According to (49), we have

$$
\begin{aligned}
\Phi^{k+1}(x) = &\int_t^{t+(k+1)\Delta T} U(x(s), \mu_1(x(s)), \ldots, \mu_N(x(s)))\mathrm{d}s\\
&+ \Phi^0(x(t+(k+1)\Delta T)).
\end{aligned}
\tag{53}
$$

As $\mu_1(x), \ldots, \mu_N(x)$ be admissible control laws, based on the result in (50), we can obtain

$$
\begin{aligned}
\lim_{k\to\infty} \Phi^{k+1}(x) = &\int_t^{\infty} U(x(s), \mu_1(x(s)), \ldots, \mu_N(x(s)))\mathrm{d}s\\
\geq &\mathcal{V}^{\infty}(x).
\end{aligned}
\tag{54}
$$

As $\mu_1(x), \ldots, \mu_N(x)$ are arbitrary admissible control laws, the following inequality can be derived:

$$
\begin{aligned}
\mathcal{V}^{\infty}(x) \leq &\min_{\mu_1, \ldots, \mu_N \in \Psi(\Omega)}\left\{\int_t^{\infty} U(x(s), \mu_1(s), \ldots, \mu_N(s))\mathrm{d}s\right\}\\
= &J^*(x_k)
\end{aligned}
\tag{55}
$$

which proves (44). By combining the results in (40) and (44), (39) can be derived. The proof is complete. ■

Remark 2: Theorem 4 indicates that the optimal control law can be obtained by the distributed policy iteration algorithm (7)–(11). The distributed policy iteration algorithm (7)–(11) possesses inherent differences from the traditional policy iteration algorithms [43], [45], [49], [50], [58], [59]. In each iteration of traditional policy iteration algorithms, all the control laws in the multicontrol nonlinear systems have to be updated simultaneously. If the dimension of the control is large, the computation burden for the traditional policy iteration increases. According to the present distributed policy iteration algorithm (7)–(11), there is only one control law to update in each iteration, which effectively reduces the computation burden of the policy iteration algorithms. This is an important advantage of the distributed policy iteration. On the other hand, for traditional policy iteration algorithms [43], [45], [49], [50], [58], [59], it has been proven that the iterative value function is convergent to the optimal performance index function as the iteration index increases to infinity. However, it is pointed out that if there exist controllers that are improved for finite times in the distributed policy iteration algorithm, then the iterative value function is convergent to a suboptimal performance index function instead of the global optimal one. It is required that all the distributed controllers are improved for infinite times to guarantee the global optimal performance index function. In traditional policy iteration algorithms, the iterative value function is sure to converge the global optimal performance index function, where the suboptimality will not happen. This is the disadvantage of the distributed policy iteration algorithm.

IV. Simulation Examples

In this section, three different simulation examples are conducted with the proposed distributed policy iteration algorithm to illustrate the corresponding performance. In the simulation examples, we use BP neural networks to realize the policy evaluation and improvement.

Example 1: In this example, the simulation of two inverted pendulums connected by a spring [56] is investigated, whose structure is shown in Fig. 1. The dynamics of the two inverted pendulum system can be described as the following equations:

$$
\begin{aligned}
\dot{x}_{1,1} &= x_{1,2}\\
\dot{x}_{1,2} &= \left(\frac{m_1 g r}{J_1} - \frac{k r^2}{4J_1}\right)\sin\big(x_{1,1}\big) + \frac{kr}{2J_1}(l-b)\\
&\quad + \frac{u_1}{J_1} + \frac{kr^2}{4J_1}\sin\big(x_{2,1}\big)\\
\dot{x}_{2,1} &= x_{2,2}
\end{aligned}
$$

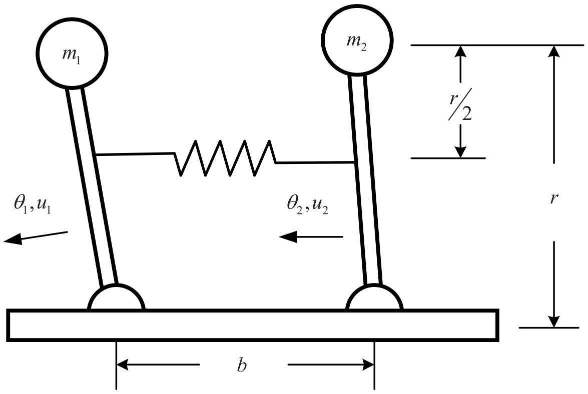


Fig. 1. Structure of two inverted pendulums.

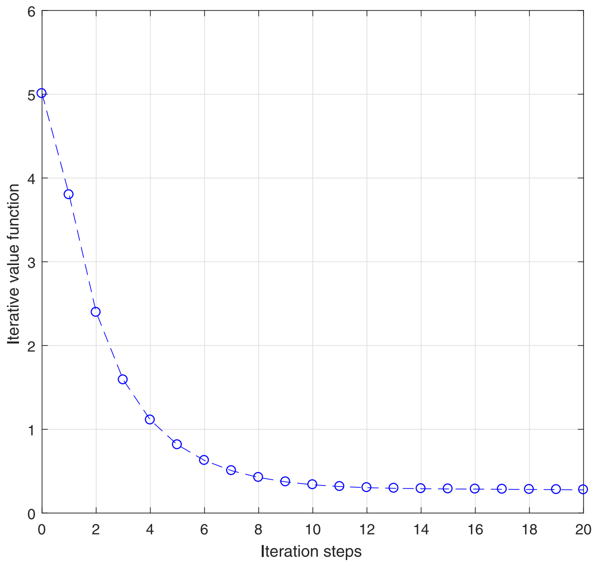


Fig. 2. Iterative value function with 20 iterations in Example 1.

$$\dot{x}_{2,2} = \left(\frac{m_2 g r}{J_2} - \frac{k r^2}{4J_2}\right)\sin(x_{2,1}) + \frac{kr}{2J_2}(l-b) + \frac{u_2}{J_2} + \frac{kr^2}{4J_2}\sin(x_{1,1}) \tag{56}$$

where $x_{1,1}$ and $x_{2,1}$ represent the angular displacements of the pendulums from vertical. The initial conditions of the systems are $x_0 = [0.1, -0.5, -0.1, 0.5]^T$. m_1 and m_2 denote the masses of the end of two pendulums, and they are considered as $m_1 = 2$ kg and $m_2 = 2.5$ kg in this example. The moments of inertia are denoted by J_1 and J_2, which are adopted as $J_1 = 0.5$ kg·m^2 and $J_2 = 0.625$ kg·m^2 here. The spring constant and natural length of the spring are represented by $k = 100$ N/m and $l = 0.5$ m, respectively. The pendulum height and the distance between the pendulum are defined as $r = 0.5$ m and $b = 0.4$ m. $g = 9.81$ m/s^2 stands for gravitational acceleration.

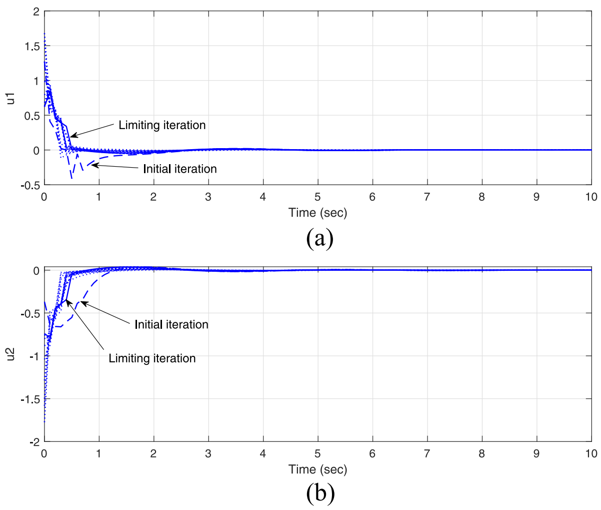
Define the performance index function as

$$J_1(x) = \int_t^{\infty} \left(x^{\mathsf{T}} Q x + u_1 R_1 u_1 + u_2 R_2 u_2\right) \mathrm{d}s \tag{57}$$

where Q, R_1, and R_2 represent identity matrices with suitable dimensions.

To implement the proposed distributed policy iteration algorithm, one critic network and two action networks are adopted with BP algorithms, which all have a three-layers structure of 4-10-1. Let the learning rate be $\alpha = 0.02$ and the training error be 10^{-5}. In this example, the initial control laws are chosen as $v_1^0(x) = -K_1 x$ and $v_2^0(x) = -K_2 x$, where $K_1 = [8.07, \ 2.13, \ 10.04, \ 1.93]$ and $K_2 = [8.63, \ 1.59, \ 10.69, \ 2.73]$, respectively, which are admissible control laws for the two inverted pendulum systems. $\eta = 0, 1, \ldots$ represents a non-negative integer series and let $\mathcal{N} = \{1, 2\}$. Let $\tau_k = 1$ for $k = 2\eta$ and let $\tau_k = 2$ for $k = 2\eta + 1$. Fig. 2 shows the trajectory of the iterative value function $V^k(x)$ at $x = x_0$ with 20 iterations by implementing the developed continuous-time distributed policy iteration algorithm. As shown in Fig. 2, the iterative value function is monotonically nonincreasing as the iteration index increases and finally, it converges to the optimum, which verifies the validity of theory analysis.

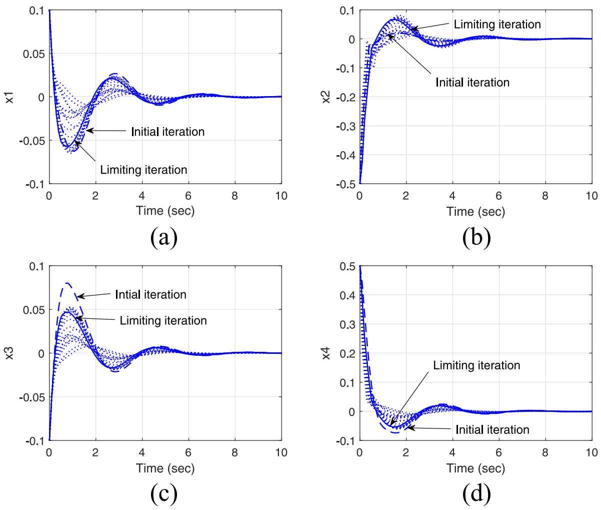
The trajectories of the iterative multicontrol laws are illustrated in Fig. 3. Although for each iteration, only one of the iterative control laws is updated for the system and the other control laws remain unchanged, the system can still be



(a)

(b)

Fig. 3. Trajectories of the iterative control laws in Example 1. (a) u_1. (b) u_2.



(a)

(b)

(c)

(d)

Fig. 4. State trajectories of the two inverted pendulum systems. (a) x_1. (b) x_2. (c) x_3. (d) x_4.

maintained stable by the distributed iterative control laws. In Fig. 4, the states of the two inverted pendulum system are illustrated. Thus, the correctness of the theoretical analysis can be verified.

Example 2: In the second example, the torsional pendulum system [57] with modifications, where two additional control inputs are added, is introduced for examining the performance of the developed algorithm. The dynamic model of the modified pendulum torsional pendulum system can be described as follows:

$$\begin{cases} \frac{d\theta}{dt} = \omega + \theta u_1 \\ \mathcal{J}\frac{d\omega}{dt} = u_2 - \text{Mgl}\sin\theta - f_d\frac{d\theta}{dt} + \omega u_3 \end{cases} \tag{58}$$

where the mass of the pendulum bar is denoted by $M = 1/3$ kg and the length is represented by $l = 2/3$ m. Let $\mathcal{J} = 4/3\ \text{Ml}^2$ kg·m^2 be the rotary inertia and $f_d = 0.2$ denotes the frictional factor. The gravity acceleration is represented by $g = 9.8$ m/s^2. By replacing θ and ω by x_1 and x_2, the model of the torsional pendulum system can be rewritten as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{\text{Mgl}}{\mathcal{J}}\sin x_1 - \frac{f_d}{\mathcal{J}}x_2 \end{bmatrix} + \begin{bmatrix} x_1 \\ -\frac{f_d x_1}{\mathcal{J}} \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ \frac{1}{\mathcal{J}} \end{bmatrix} u_2 + \begin{bmatrix} 0 \\ \frac{x_2}{\mathcal{J}} \end{bmatrix} u_3. \tag{59}$$

The corresponding performance index function is defined as

$$J_2(x) = \int_t^{\infty} \left(x^{\mathsf{T}}\mathcal{Q}x + u_1\mathcal{R}_1u_1 + u_2\mathcal{R}_2u_2 + u_3\mathcal{R}_3u_3\right)\text{d}s \tag{60}$$

where $\mathcal{Q}$, $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$ are positive-definite matrices with suitable dimensions, which are considered as identity matrices in this example.

To apply the developed methods, four neural networks, consisting of one critic network and three action networks, are adopted in the systems. The four neural networks all adopt the BP algorithm to train the weights with three-layers structure of 2-8-1. Let the learning rate be $\alpha = 0.02$ and let the training error be 10^{-5}. The initial conditions of the multicontrol laws are chosen as $v_1^0(x) = -\mathcal{K}_1x$, $v_2^0(x) = -\mathcal{K}_2x$, and $v_3^0(x) = -\mathcal{K}_3x$, where $\mathcal{K}_1 = [-0.0007, -0.1249]$, $\mathcal{K}_2 = [0.0037, 0.6247]$, and $\mathcal{K}_3 = [0, 0]$, respectively. Let $\mathcal{N} = \{1, 2, 3\}$. Let $\tau_k = 1$ for $k = 2\eta$, $\tau_k = 2$ for $k = 2\eta + 1$, and $\tau_k = 3$ for $k = 2\eta + 2$. To implement the developed continuous-time distributed policy iteration, the algorithm has been iterated for 30 steps. In Fig. 5, the trajectory of the iterative value function $V^k(x)$ at $x = x_0$ is given to show its convergence, which implies that the iterative value function is monotonously nonincreasing and will converge to the optimum as the iteration index increases. The correctness of the theory analysis can be verified.

The trajectories of the iterative control laws are illustrated in Fig. 6. Although only one of multicontrol laws is updated for each iteration and other control laws remain unchanged, the stability of systems can still be achieved by the distributed iterative control laws presented in this article. To show the convergence of the system states, the state trajectories are illustrated in Fig. 7. Thus, for nonlinear system with multicontrollers (59), it is feasible to update the multicontrol laws one by one in the distributed policy iteration for obtaining the global optimal control law of the system, and the advantages of the distributed policy iteration will be remarkable for the systems with high dimensions in control.

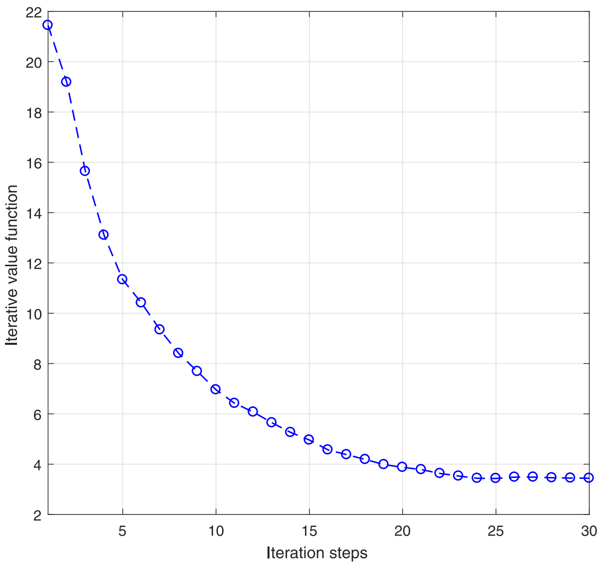
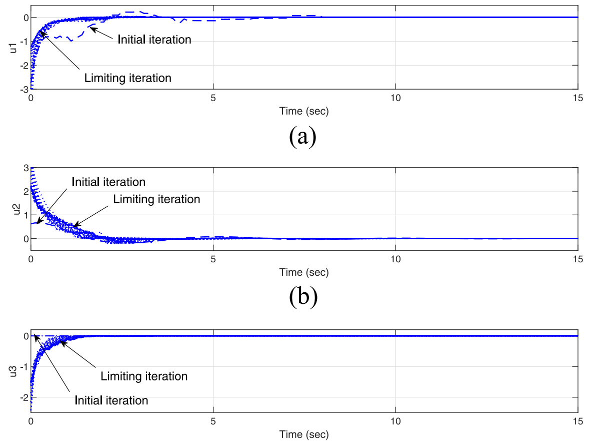


Fig. 5. Iterative value function with 30 iterations in Example 2.



(a)

(b)

(c)

Fig. 6. Trajectories of multicontrollers in Example 2. (a) u_1. (b) u_2. (c) u_3.

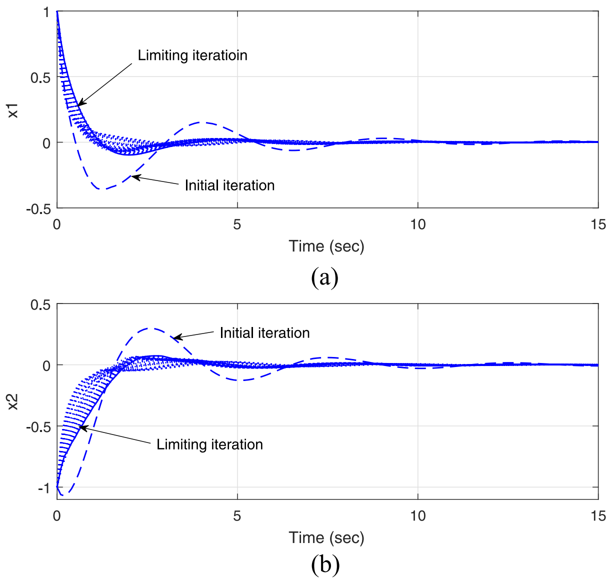
Example 3: In the third example, a nonaffine nonlinear system is introduced for examining the performance of the developed algorithm, and a comparison experiment is conducted with the traditional policy iteration algorithm. The nonaffine nonlinear system is chosen in [60], where the dynamic model of the system is described as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 + x_1u_1 \\ \dot{x}_2 &= x_1^2 + 0.15u_2^3 + 0.1\left(4 + x_2^2\right)u_2 + \sin(0.1u_2). \end{aligned} \tag{61}$$

The corresponding performance index function is defined as

$$J_3(x) = \int_t^{\infty} \left(x^{\mathsf{T}}\mathcal{Q}x + u_1\mathcal{R}_1u_1 + u_2\mathcal{R}_2u_2\right)\text{d}s \tag{62}$$

where $\mathcal{Q}$, $\mathcal{R}_1$, and $\mathcal{R}_2$ are positive-definite matrices with suitable dimensions, which are considered as identity matrices in this example.



(a)

(b)

Fig. 7. State trajectories of the torsional pendulum system in Example 2. (a) x_1. (b) x_2.

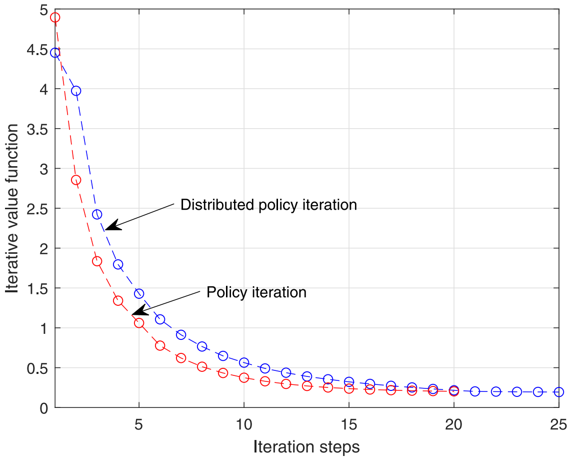
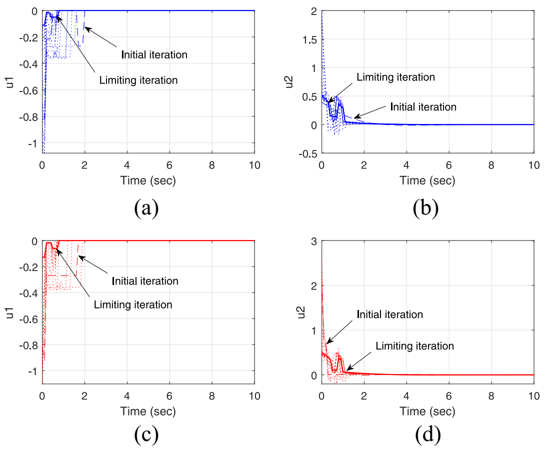


Fig. 8. Iterative value function.

To apply the developed method, one critic network and two action networks are adopted in the system, which all have a three-layers structure of 2-10-1. Let the learning rate be $\alpha = 0.02$ and let the training error be 10^{-5}. The initial conditions of the multicontrol laws are chosen as $v_1^0(x) = -\mathcal{K}_1 x$ and $v_2^0(x) = -\mathcal{K}_2 x$, where $\mathcal{K}_1 = [0, 0]$ and $\mathcal{K}_2 = [0.4668, 0.9642]$, respectively. Let $\bar{\mathcal{N}} = \{1, 2\}$. Let $\tau_k = 1$ for $k = 2\eta$ and $\tau_k = 2$ for $k = 2\eta + 1$. To implement the developed continuous-time distributed policy iteration, the algorithm has been iterated for 25 steps. In Fig. 8, the trajectory of the iterative value function $V^k(x)$ at $x = x_0$ is given to show its convergence, and the trajectory of the traditional policy iteration is given as a contrast. As shown in Fig. 8, the traditional policy iteration has a faster convergence speed, because all the control laws in the multicontrol nonlinear system have to be updated simultaneously in this algorithm, and it increases the computation burden. The correctness of the theoretical analysis can be verified.

The trajectories of the iterative control laws are illustrated in Fig. 9. To show the convergence of the system states, the state trajectories are illustrated in Fig. 10. For the nonaffine nonlinear system with multicontrollers (61), the correctness of the proposed distributed policy iteration algorithm can be demonstrated.



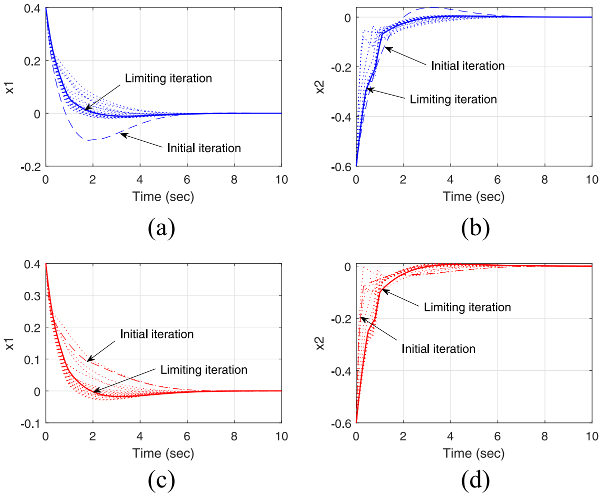
(a)

(b)

(c)

(d)

Fig. 9. Trajectories of multicontrollers. (a) u_1 by the distributed policy iteration. (b) u_2 by the distributed policy iteration. (c) u_1 by the traditional policy iteration. (d) u_2 by the traditional policy iteration.



(a)

(b)

(c)

(d)

Fig. 10. State trajectories. (a) x_1 by the distributed policy iteration. (b) x_2 by the distributed policy iteration. (c) x_1 by the traditional policy iteration. (d) x_2 by the traditional policy iteration.

V. Conclusion

In this article, a novel continuous-time distributed policy iteration algorithm is proposed to be applied in multicontroller nonlinear systems for achieving the infinite horizon optimal control. In each iteration of the proposed algorithms, only one of the multicontrol laws is updated instead of all the control laws, which implies that the control laws are improved one by one. First, the detailed iterative methods of the distributed policy iteration are introduced. Second, this article also discussed the admissibility of the proposed multicontrol laws. In addition, the iterative value function can converge to optimum, which is the solution of HJB equations. Finally, some numerical simulations are conducted to verify the effectiveness of the presented methods.

References

[1] Y. Yuan, Z. Wang, P. Zhang, and H. Dong, "Nonfragile near-optimal control of stochastic time-varying multiagent systems with control and state-dependent noises," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2605–2617, Jul. 2019.

[2] X. Zhong and H. He, "GrHDP solution for optimal consensus control of multiagent discrete-time systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, doi: 10.1109/TSMC.2018.2814018.

[3] Y. Li, K. Sun, and S. Tong, "Observer-based adaptive fuzzy fault-tolerant optimal control for SISO nonlinear systems," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 649–661, Feb. 2019.

[4] X. Yang and B. Zhao, "Optimal neuro-control strategy for nonlinear systems with asymmetric input constraints," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 575–583, Mar. 2020.

[5] K. D. Do, "Stability in probability and inverse optimal control of evolution systems driven by Levy processes," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 405–419, Mar. 2020.

[6] L. Cui, Y. Li, R. Yang, and X. Zhang, "Asymptotical cooperative tracking control for unknown high-order multi-agent systems via distributed adaptive critic design," *IEEE Access*, vol. 6, pp. 24650–24659, 2018.

[7] X. Li and D. Zhu, "An adaptive SOM neural network method for distributed formation control of a group of AUVs," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 8260–8270, Oct. 2018.

[8] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Robust distributed predictive control of waterborne AGVs: A cooperative and cost-effective approach," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2449–2461, Aug. 2018.

[9] R. Han, L. Meng, J. M. Guerrero, and J. C. Vasquez, "Distributed nonlinear control with event-triggered communication to achieve current-sharing and voltage regulation in DC microgrids," *IEEE Trans. Power Electron.*, vol. 33, no. 7, pp. 6416–6433, Jul. 2018.

[10] Y. H. Choi and S. J. Yoo, "Minimal-approximation-based distributed consensus tracking of a class of uncertain nonlinear multiagent systems with unknown control directions," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 1994–2007, Aug. 2017.

[11] Y. Xu, E. Alfonsetti, P. C. Weeraddana, and C. Fischione, "A semidistributed approach for the feasible min–max fair agent-assignment problem with privacy guarantees," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 333–344, Mar. 2018.

[12] M. Mastio, M. Zargayouna, G. Scemama, and O. Rana, "Distributed agent-based traffic simulations," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 1, pp. 145–156, Jan. 2018.

[13] H. Zhang, Y. Li, D. W. Gao, and J. Zhou, "Distributed optimal energy management for energy Internet," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3081–3097, Dec. 2017.

[14] Y. Shang, B. Chen, and C. Lin, "Consensus tracking control for distributed nonlinear multiagent systems via adaptive neural backstepping approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, doi: 10.1109/TSMC.2018.2816928.

[15] F. Chen and W. Ren, "A connection between dynamic region-following formation control and distributed average tracking," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1760–1772, Jun. 2018.

[16] L. Jin and S. Li, "Distributed task allocation of multiple robots: A control perspective," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 5, pp. 693–701, May 2018.

[17] Y. Li, C. Hua, S. Wu, and X. Guan, "Output feedback distributed containment control for high-order nonlinear multiagent systems," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 2032–2043, Aug. 2017.

[18] F. Wang, B. Chen, C. Lin, and X. Li, "Distributed adaptive neural control for stochastic nonlinear multiagent systems," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1795–1803, Jul. 2017.

[19] Q. Wei, D. Liu, Q. Lin, and R. Song, "Adaptive dynamic programming for discrete-time zero-sum games," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 957–969, Apr. 2018.

[20] X. Xu, H. Chen, C. Lian, and D. Li, "Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6202–6213, Dec. 2018.

[21] Q. Wei, Z. Liao, Z. Yang, B. Li, and D. Liu, "Continuous-time time-varying policy iteration," *IEEE Trans. Cybern.*, early access, doi: 10.1109/TCYB.2019.2926631.

[22] Q. Wei, R. Song, Z. Liao, B. Li, and F. L. Lewis, "Discrete-time impulsive adaptive dynamic programming," *IEEE Trans. Cybern.*, early access, doi: 10.1109/TCYB.2019.2906694.

[23] Z. Wang, L. Liu, Y. Wu, and H. Zhang, "Optimal fault-tolerant control for discrete-time nonlinear strict-feedback systems based on adaptive critic design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2179–2191, Jun. 2018.

[24] X. Yang, H. He, and X. Zhong, "Adaptive dynamic programming for robust regulation and its application to power systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5722–5732, Jul. 2018.

[25] Q. Wei, L. Wang, Y. Liu, and M. Polycarpou, "Optimal elevator group control via deep asynchronous actor–critic learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, doi: 10.1109/TNNLS.2020.2965208.

[26] Q. Wei, F. L. Lewis, D. Liu, R. Song, and H. Lin, "Discrete-time local value iteration adaptive dynamic programming: Convergence analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 6, pp. 875–891, Jun. 2018.

[27] X. Yang and H. He, "Decentralized event-triggered control for a class of nonlinear-interconnected systems using reinforcement learning," *IEEE Trans. Cybern.*, early access, doi: 10.1109/TCYB.2019.2946122.

[28] P. J. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence," *Gen. Syst. Yearbook*, vol. 22, no. 2, pp. 25–38, 1977.

[29] 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.

[30] X. Yang and H. He, "Adaptive dynamic programming for decentralized stabilization of uncertain nonlinear large-scale systems with mismatched interconnections," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, doi: 10.1109/TSMC.2018.2837899.

[31] K. Wan, X. Gao, B. Li, and F. Li, "Using approximate dynamic programming for multi-ESM scheduling to track ground moving targets," *J. Syst. Eng. Electron.*, vol. 29, no. 1, pp. 74–85, Jan. 2018.

[32] H. Jiang, H. Zhang, Y. Luo, and J. Han, "Neural-network-based robust control schemes for nonlinear multiplayer systems with uncertainties via adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 3, pp. 579–588, Mar. 2019.

[33] R. Song, F. L. Lewis, and Q. Wei, "Off-policy integral reinforcement learning method to solve nonlinear continuous-time multiplayer nonzero-sum games," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 704–713, Mar. 2017.

[34] H. Zhang, H. Jiang, C. Luo, and G. Xiao, "Discrete-time nonzero-sum games for multiplayer using policy-iteration-based adaptive dynamic programming algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3331–3340, Oct. 2017.

[35] Q. Wei, Z. Liao, R. Song, P. Zhang, Z. Wang, and J. Xiao, "Self-learning optimal control for ice storage air conditioning systems via data-based adaptive dynamic programming," *IEEE Trans. Ind. Electron.*, early access.

[36] Q. Wei, D. Liu, Y. Liu, and R. Song, "Optimal constrained self-learning battery sequential management in microgrids via adaptive dynamic programming," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 2, pp. 168–176, Apr. 2017.

[37] J. Skach, B. Kiumarsi, F. L. Lewis, and O. Straka, "Actor–critic off-policy learning for optimal control of multiple-model discrete-time systems," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 29–40, Jan. 2018.

[38] B. Luo, D. Liu, and H.-N. Wu, "Adaptive constrained optimal control design for data-based nonlinear discrete-time systems with critic-only structure," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2099–2111, Jun. 2018.

[39] Y. Huang, "Optimal guaranteed cost control of uncertain non-linear systems using adaptive dynamic programming with concurrent learning," *IET Control Theory Appl.*, vol. 12, no. 8, pp. 1025–1035, Aug. 2018.

[40] H. Zhang, X. Cui, Y. Luo, and H. Jiang, "Finite-horizon H_∞ tracking control for unknown nonlinear systems with saturating actuators," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1200–1212, Apr. 2018.

[41] Q. Wei, F. L. Lewis, G. Shi, and R. Song, "Error-tolerant iterative adaptive dynamic programming for optimal renewable home energy scheduling and battery management," *IEEE Trans. Ind. Electron.*, vol. 64, no. 12, pp. 9527–9537, Dec. 2017.

[42] C. Mu, D. Wang, and H. He, "Data-driven finite-horizon approximate optimal control for discrete-time nonlinear systems using iterative HDP approach," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2948–2961, Oct. 2018.

[43] 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.

[44] X. Yang, H. He, and X. Zhong, "Approximate dynamic programming for nonlinear-constrained optimizations," *IEEE Trans. Cybern.*, early access.

[45] D. P. Bertsekas, "Value and policy iterations in optimal control and adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 500–509, Mar. 2017.

[46] Q. Wei, B. Li, and R. Song, "Discrete-time stable generalized self-learning optimal control with approximation errors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1226–1238, Apr. 2018.

[47] Q. Wei, D. Liu, F. L. Lewis, Y. Liu, and J. Zhang, "Mixed iterative adaptive dynamic programming for optimal battery energy control in smart residential microgrids," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4110–4120, May 2017.

[48] H. Jiang and H. Zhang, "Iterative ADP learning algorithms for discrete-time multi-player games," *Artif. Intell. Rev.*, vol. 50, no. 1, pp. 75–91, Jun. 2018.

[49] 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.

[50] 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.

[51] Y. Zhu, D. Zhao, X. Yang, and Q. Zhang, "Policy iteration for H_∞ optimal control of polynomial nonlinear systems via sum of squares programming," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 500–509, Feb. 2018.

[52] C. Li, D. Liu, and D. Wang, "Data-based optimal control for weakly coupled nonlinear systems using policy iteration," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 511–521, Apr. 2018.

[53] 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, Aug. 2012.

[54] D. Ye, M.-M. Chen, and H.-J. Yang, "Distributed adaptive event-triggered fault-tolerant consensus of multiagent systems with general linear dynamics," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 757–767, Mar. 2019.

[55] H. Zhang, H. Liang, Z. Wang, and T. Feng, "Optimal output regulation for heterogeneous multiagent systems via adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 18–29, Jan. 2017.

[56] J. T. Spooner and K. M. Passino, "Decentralized adaptive control of nonlinear systems using radial basis neural networks," *IEEE Trans. Autom. Control*, vol. 44, no. 11, pp. 2050–2057, Nov. 1999.

[57] 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.

[58] H. Modares and F. L. Lewis,s "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, Jul. 2014.

[59] 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.

[60] X. Zhang, "Research on approximate optimal control of nonaffine nonlinear systems based on neural networks," Ph.D. dissertation, School Inf. Sci. Eng., Northeastern Univ., Shenyang, China, 2012.

Qinglai Wei (Member, IEEE) received the B.S. degree in automation and the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2002 and 2009, respectively.

From 2009 to 2011, he was a Postdoctoral 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 a Professor with the Institute of Automation and the Associate Director of the State Key Laboratory of Management and Control for Complex Systems. His research interests include adaptive dynamic programming, neural-networks-based control, optimal control, and nonlinear systems and their industrial applications.

Dr. Wei was a recipient of the IEEE/CAA JOURNAL OF AUTOMATICA SINICA Best Paper Award, the IEEE System, Man, and Cybernetics Society Andrew P. Sage Best Transactions Paper Award, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS Outstanding Paper Award, the Outstanding Paper Award of *Acta Automatica Sinica*, the IEEE 6th Data Driven Control and Learning Systems Conference in 2017 Best Paper Award, the Zhang Siying Outstanding Paper Award of Chinese Control and Decision Conference, the Shuang-Chuang Talents in Jiangsu Province, China, the Young Researcher Award of Asia–Pacific Neural Network Society, the Young Scientist Award, and the Yang Jiachi Tech Award of Chinese Association of Automation (CAA). He has been the Secretary of the IEEE Computational Intelligence Society Beijing Chapter since 2015. He was a guest editor for several international journals. He is a Board of Governors Member of the International Neural Network Society and a Council Member of CAA.

Hongyang Li received the bachelor's degree in automation from North China Electric Power University, Baoding, China, in 2016, and the master's degree in control science and engineering from Tsinghua University, Beijing, China, in 2019. He is currently pursuing the Ph.D. degree in technology of computer applications with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, and the University of Chinese Academy of Sciences, Beijing.

His current research interests include reinforcement learning, adaptive dynamic programming, optimal control, and neural networks.

Xiong Yang (Member, IEEE) received the B.S. degree in mathematics and applied mathematics from Central China Normal University, Wuhan, China, in 2008, the M.S. degree in pure mathematics from Shandong University, Jinan, China, in 2011, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

From 2016 to 2018, he was a Postdoctoral Fellow with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA. He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. His research interests include intelligent control, reinforcement learning, deep neural networks, event-triggered control, and their applications.

Dr. Yang was a recipient of the Excellent Award of Presidential Scholarship of the Chinese Academy of Sciences in 2014.

Haibo He (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University, Athens, OH, USA, in 2006.

He is currently the Robert Haas Endowed Chair Professor with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA. His research areas include computational intelligence, neural networks, reinforcement learning, and their applications.

Dr. He was a recipient of the IEEE International Conference on Communications Best Paper Award in 2014, the IEEE CIS Outstanding Early Career Award in 2014, and the U.S. National Science Foundation CAREER Award in 2011. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He was the General Chair of the IEEE Symposium Series on Computational Intelligence in 2014.