GrHDP Solution for Optimal Consensus Control of Multiagent Discrete-Time Systems

Xiangnan Zhong, *Member, IEEE*, and Haibo He, *Fellow, IEEE*

***Abstract*—This paper develops a new online learning consensus control scheme for multiagent discrete-time systems by goal representation heuristic dynamic programming (GrHDP) techniques. The agents in the whole system are interacted with each other through a communication graph structure. Therefore, each agent can only receive the information from itself and its neighbors. Our goal is to design the GrHDP method to achieve consensus control which makes all the agents track the desired dynamics and simultaneously makes the performance indices reach Nash equilibrium. The new local internal reinforcement signals and local performance indices are provided for each agent and the corresponding distributed control laws are designed. Then, GrHDP algorithm is developed to solve the multiagent consensus control problem with the proof of convergence. It is shown that the designed local internal reinforcement signals are bounded signals and the local performance indices can monotonically converge to their optimal values. Moreover, the desired distributed control laws can also achieve optimal. Two simulation studies, including one with four agents and another with ten agents, are applied to validate the theoretical analysis and also demonstrate the effectiveness of the proposed method.**

***Index Terms*—Adaptive dynamic programming (ADP), consensus control, goal representation, multiagent systems online learning, neural networks.**

I. Introduction

CONSENSUS control problems of multiagent systems has attracted increasing significant attention in recent years [1]–[5], especially in sensor networks [6], [7], unmanned aerial vehicles [8], flocking [9], among others. Multiagent systems [10]–[12] are a group of autonomous systems, interacting with each other through communication or sensing networks. Such systems can perform certain challenge tasks which cannot be accomplished by a single agent. In [13], a distributed secure consensus tracking control problem was investigated for multiagent systems. The authors established a hybrid stochastic secure control framework to design a distributed secure control law. In [14], a networked multiagent predictive control scheme was provided for multiagent systems to achieve output consensus and also compensate for the communication delays and data loss actively. A fully distributed integrated solution was presented in [15] for multiarea topology identification and state estimation problems of power systems. So far, many of the studies of multiagent systems focus on solving the optimal consensus control problem based on accurate system functions and/or models. However, in many real-world applications, the likelihood to access the complete knowledge of system functions is either infeasible or very difficult to obtain. To solve the problem, a learning-based method, adaptive dynamic programming (ADP), was integrated into the multiagent systems control designs to approximate the solution of coupled Hamilton–Jacobi–Bellman (HJB) equation [16]–[20]. In the literature, the exact information of the system models was not required and only the input/output data were used to estimate the optimal solution.

In recent years, ADP techniques have witnessed extensive studies from both theoretical research and real-world applications [21]–[26]. Because ADP method is totally data-driven, which means it can solve the optimal control problems without the information of system functions, this method has been widely recognized as one of the "core methodologies" to achieve optimal control for intelligent systems in a general case [27], [28]. Usually, ADP can be categorized into three typical schemes: 1) heuristic dynamic programming (HDP); 2) dual HDP (DHP); and 3) globalized DHP (GDHP). Specifically, the HDP method develops an action network to approximate the control law and a critic network to estimate the corresponding performance index or total cost-to-go in Bellman equation. In [29], the neural-network-based implementation process of HDP was provided with explicit backpropagation rules for both action and critic networks. The authors further analyzed the stability of this method. It was shown that the estimation errors of the neural network weights were uniformly ultimately bounded by Lyapunov stability construct. Many other researches and publications of HDP design from both theoretical and application studies were also provided and demonstrated in [22] and [30]–[35]. Later, Werbos went beyond the critic network approximating just the performance index and further developed two new schemes: DHP and GDHP. The core idea of DHP is to design the critic network estimating the derivatives of the performance index, which have the high quality comparing with the performance index itself. Moreover, GDHP method combines the advantage of both HDP and DHP methods and approximates the performance index and its derivatives at the

Manuscript received May 10, 2017; revised February 12, 2018; accepted March 4, 2018. Date of publication April 3, 2018; date of current version June 16, 2020. This work was supported by the National Science Foundation under Grant CMMI 1526835 and Grant ECCS 1731672. This paper was recommended by Associate Editor Y.-J. Liu. *(Corresponding author: Haibo He.)*

X. Zhong was with the Department of Electrical Engineering, University of North Texas, Denton, TX 76207 USA (e-mail: xiangnan.zhong@unt.edu).

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

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

Digital Object Identifier 10.1109/TSMC.2018.2814018

same time. The differences of the learning processes for the HDP, DHP, and GDHP methods were provided in [21]. The GDHP method was developed in [36] and [37] for a class of unknown discrete-time nonlinear systems. The authors also compared the performance of the HDP, DHP, and GDHP controller to show that the GDHP method can achieve better control performance. Various versions of ADP have been developed based on these three typical schemes, such as the action-dependent version and model-dependent version.

Generally, in the traditional ADP method, it is assumed that the agent knows what the immediate reinforcement signal is or how the immediate reinforcement signal is computed as the function of the system states and actions. Recently, by considering the general-propose reinforcement signals with the capability of adaptive learning overtime, a series of goal representation HDP (GrHDP) design was developed to facilitate the learning process [38]–[40]. The authors integrated an additional neural network, goal/reference network, into the traditional ADP design to generate an internal reinforcement signal. Reference [41] further proved this designed internal reinforcement signal could give the agent more information by considering more distant lookahead. So far, the proposed GrHDP architecture has been applied to various realistic and complex control problems, for instance, tracking problems [42], maze navigation [43], [44], power systems [45], among others. Furthermore, this idea of goal representation has been later integrated into the DHP and GDHP design. In [46], it was shown that the GrDHP method can improve the control performance on certain nonlinear examples, including power system examples. The goal representation GDHP (Gr-GDHP) method was proposed in [47] and the control performance had been compared with the GrHDP, GrDHP, and GDHP methods. Moreover, many researchers also followed this trend and applied the three-network HDP framework from different aspects [48], [49].

In this paper, motivated by the literature research, we develop a data-driven GrHDP method to solve the optimal consensus control problem for a class of unknown discrete-time multiagent systems. In the proposed method, the agent can only receive the information from itself and its neighbors. The goal is to make all the follower agents track the desired dynamics (leader). We include the neighbors' control signals into the external reinforcement signals for each agent to closely connect the agent with its neighbors. Moreover, we design the internal reinforcement signals based on the external reinforcement signals to facilitate the learning process. It is shown that the designed internal reinforcement signals have more information and therefore they are more effective. The major contributions of this paper can be summarized as follows. First, we extend the single-agent GrHDP method to the multiagent consensus control problems. New local internal reinforcement signals and local performance indices are designed in consideration of the information from the agent itself and its neighbors. This is, to our best knowledge, the first time of designing GrHDP method for multiagent consensus control problems. Second, Nash equilibrium solution of the proposed GrHDP method is analyzed. It is proved that the designed new local performance indices can reach Nash equilibrium. Third, we develop the iterative GrHDP algorithm for multiagent systems under communication network structure. The convergence proof of the proposed algorithm is also provided. It is shown that the designed local internal reinforcement signals are bounded. The local performance indices and the designed distributed control laws can converge to the optimal values, respectively. Forth, we compare our results with the traditional HDP method. From the comparison, we observe that our proposed GrHDP method can achieve better performance in the consensus control process. Neural network techniques are applied to implement the proposed method. The goal, critic, and action networks are designed for each agent to estimate the internal reinforcement signals, performance indices, and distributed control laws, respectively. The simulation results show the effectiveness of the proposed method.

The rest of this paper is organized as follows. In Section II, we define the error dynamics in multiagent consensus control problems. The relationship between the synchronization error and the overall tracking error is also provided. The local internal reinforcement signals and performance indices are defined and discussed in Section III for the preparation of the analysis conducted next. The Nash equilibrium of the designed performance indices is also proved in this section. The proposed GrHDP algorithm is presented in Section IV with explicit convergence proof. Then, Section V develops the neural-network-based implementation process of the proposed GrHDP design. The simulation results are presented in Section VI to demonstrate the effectiveness of this method. Finally, Section VII concludes this paper.

II. Synchronization of Multiagent Discrete-Time Systems

A. Preliminary

Let $\mathcal{F} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ be a directed graph which is composed of a nonempty finite set of N vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, a set of edge $\mathcal{E} = \{p_{ij} = (v_i, v_j)\} \subseteq \mathcal{V} \times \mathcal{V}$, and a weighted adjacency matrix $\mathcal{A} = [p_{ij}]$ with non-negative adjacency elements $p_{ij} \geq 0$. If and only if $p_{ij} = (v_i, v_j) \in \mathcal{E}$, then $p_{ij} > 0$, which means node i can receive information from node j; otherwise, $p_{ij} = 0$. The set of neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in \mathcal{E}\}$. The in-degree matrix $\mathcal{D}$ is defined as a diagonal matrix $\mathcal{D} = \text{diag}\{d_i\}$ with $d_i = \sum_{j \in N_i} p_{ij}$ the weighted in-degree of node i. Then, the graph Lapalacian matrix $\mathcal{L} = \mathcal{D} - \mathcal{A}$. A directed path from node v_i to node v_r is described as a sequence of edges $v_i, v_{i+1}, \ldots, v_r$, such that $(v_j, v_{j+1}) \in \mathcal{E}, j \in \{i, i+1, \ldots, r\}$. If there is a node v_0, called the leader, such that the directed paths from the leader to any other nodes are in the graph, we call the graph as a spanning tree.

B. Synchronization and Node Error Dynamics

Consider the multiagent discrete-time systems with N agents distributed on communication graph $\mathcal{F}$

$$x_i(k+1) = Ax_i(k) + B_i u_i(k), \quad i = 1, 2, \ldots, N \tag{1}$$

where $x_i(k) \in R^n$ is the state of agent i and $u_i(k) \in R^{m_i}$ is its control input. The system matrices $A \in R^{n\times n}$ and $B_i \in R^{n\times m_i}$ are considered unknown in this paper.

The leader system, which has command generator dynamics, is defined as

$$x_0(k+1) = Ax_0(k) \tag{2}$$

where $x_0(k) \in R^n$ is the consensus objective state. Usually, the leader is only directly connected to a small percentage of the systems in the multiagent graph.

Our goal is to design the distributed control laws $u_i(k)$ for each agent i using the information only from the agent itself and its neighbor agents, such that all agent states synchronize to the leader state, which is $\lim_{k\to\infty} ||x_i(k) - x_0(k)|| = 0,\ \forall i$.

In order to investigate the consensus control problem on directed graphs, we define the local neighborhood tracking error as

$$\delta_i(k) = \sum_{j\in N_i} p_{ij}\big(x_i(k) - x_j(k)\big) + q_i(x_i(k) - x_0(k)) \tag{3}$$

where $q_i \geq 0$ is the pining gain of agent i. We have $q_i > 0$ if agent i is coupled to the leader x_0, otherwise, $q_i = 0$.

The overall tracking error vector for the entire multiagent systems [17], [20] is given by

$$\begin{aligned}\delta(k) &= (\mathcal{L}\otimes I_n)x(k) - (\mathcal{L}\otimes I_n)\bar{x}_0(k) + (\mathcal{B}\otimes I_n)\\ &\quad\times (x(k) - \bar{x}_0(k))\\ &= ((\mathcal{L}+\mathcal{B})\otimes I_n)(x(k) - \bar{x}_0(k))\end{aligned} \tag{4}$$

where $\mathcal{L} = [l_{ij}] \in R^{N\times N}$ is the Laplacian matrix, $\mathcal{B} = [q_{ij}] \in R^{N\times N}$ is a diagonal matrix with the diagonal elements $q_{ii} = q_i$, $\otimes$ denotes the Kronecker product operator, $\bar{x}_0(k) = \underline{I}\otimes x_0$ with $I = \underline{1}\otimes I_n$, I_n is an $n\times n$ identity matrix, and $\underline{1}$ is an N-dimensional vector of ones.

Equation (4) can be further rewritten as

$$\delta(k) = ((\mathcal{L}+\mathcal{B})\otimes I_n)\eta(k) \tag{5}$$

where

$$\eta(k) = x(k) - \bar{x}_0(k) \tag{6}$$

is the global disagreement vector or the synchronization error vector. Note that if the graph contains a spanning tree and $q_i \neq 0$ for a leader node, then $(\mathcal{L}+\mathcal{B})$ is nonsingular.

Now, consider (4) and (6), we can summarize the relationship between the synchronization error $\eta(k)$ and the overall tracking error $\delta(k)$ in Lemma 1.

Lemma 1 [16]: Let $(\mathcal{L}+\mathcal{B})$ be nonsingular. Then the synchronization error is bounded by

$$\eta(k) \leq \frac{\|\delta(k)\|}{\lambda_{\min}(\mathcal{L}+\mathcal{B})} \tag{7}$$

where $\lambda_{\min}(\mathcal{L}+\mathcal{B})$ is the minimum singular value of $(\mathcal{L}+\mathcal{B})$.

From Lemma 1, we know when $\|\delta(k)\| \to 0$, $||\eta(k)|| \to 0$. This means the synchronization error can be made arbitrarily small by making the neighborhood tracking errors small.

The dynamics of the local neighborhood tracking error for agent i are defined as

$$\begin{aligned}\delta_i(k+1) &= \sum_{j\in N_i} p_{ij}\big(x_i(k+1) - x_j(k+1)\big)\\ &\quad + q_i(x_i(k+1) - x_0(k+1)).\end{aligned} \tag{8}$$

It can be further rewritten as

$$\delta_i(k+1) = A\delta_i(k) + (d_i + q_i)B_iu_i(k) - \sum_{j\in N_i} p_{ij}B_ju_j(k) \tag{9}$$

where $d_i = \sum_{j\in N_i} p_{ij}$. These tacking error dynamics are interacting dynamical systems driven by the control action of agent i itself and all of its neighbors. Our goal is to minimize the local neighborhood tracking error $\delta_i(k)$, which according to Lemma 1 will guarantee approximate synchronization.

III. Nash Equilibrium Solution on GrHDP Technique

In this section, GrHDP method is designed to solve the consensus control problem. Based on the error dynamics (9), we define the local internal reinforcement signals, local performance indices, and distributed control laws for each agent. Then, Nash equilibrium is discussed and the designed control laws are proved to provide Nash equilibrium solution for the multiagent dynamic systems.

A. Bellman Equation of GrHDP Method Under Communication Graphs

In the GrHDP method, we design the internal reinforcement signals to help the systems achieve their goals. Here, we define the local internal reinforcement signals [38], [39], [41] as

$$\begin{aligned}s_i(\delta_i(k)) &= \sum_{m=k}^{\infty} \alpha^{m-k} r_i(\delta_i(m), u_i(m), u_{-i}(m))\\ &= r_i(\delta_i(k), u_i(k), u_{-i}(k)) + \alpha s_i(\delta_i(k+1))\end{aligned} \tag{10}$$

where $u_{-i}(k) = \{u_j(k) | j \in N_i\}$ is the control actions from the neighbors of agent i, $0 < \alpha < 1$ is the discount factor, and $r_i(\delta_i(k), u_i(k), u_{-i}(k)) = \delta_i^T(k)Q_{ii}\delta_i(k) + u_i^T(k)R_{ii}u_i(k) + \sum_{j\in N_i} u_j^T R_{ij}u_j(k)$ is the external reinforcement signal with $Q_{ii} > 0$, $R_{ii} > 0$, and $R_{ij} > 0$, which are all positive symmetric weighting matrices.

It can be observed that the designed local internal reinforcement signals $s_i(\delta_i(k))$ contain the information of future external reinforcement signals $r_i(k+1), r_i(k+2), \ldots$ Comparing with the traditional HDP method which only provides one single external reinforcement signal to the agent, these designed internal reinforcement signals can give us more information by considering more distant lookahead. This means the internal reinforcement signals look forward in time to the future information for each state visited and therefore these signals are more effective.

Then, the local performance indices are given by

$$J_i(\delta_i(k)) = s_i(\delta_i(k)) + \gamma J_i(\delta_i(k+1)) \tag{11}$$

where $0 < \gamma < 1$ is the discount factor.

From (10) and (11), we notice that both local internal reinforcement signals and local performance indices use the information only from agent i itself and its neighbors. Our goal is to design the optimal distributed control laws to minimize the local performance indices (11) to make all the agents achieve consensus with the target state x_0.

Definition 1 [22]: The control laws $u_i(k)$ for $\forall i$ are said to be admissible if they do not only stabilize the systems (9), but also guarantee that performance indices (11) are finite.

Based on Bellman optimality principle, the optimal local performance indices $J_i^*(\delta_i(k))$ satisfy the coupled discrete-time HJB equation

$$J_i^*(\delta_i(k)) = \min_{u_i(k)}\left\{s_i(\delta_i(k)) + \gamma J_i^*(\delta_i(k+1))\right\} \tag{12}$$

where

$$s_i(\delta_i(k)) = r_i(\delta_i(k), u_i(k), u_{-i}(k)) + \alpha s_i(\delta_i(k+1)) \tag{13}$$

are the local internal reinforcement signals.

Therefore, the local optimal distributed control laws can be described as

$$u_i^*(k) = \arg\min_{u_i(k)}\left\{s_i(\delta_i(k)) + \gamma J_i^*(\delta_i(k+1))\right\}. \tag{14}$$

Note that, from (12), the designed distributed control law decides what is the best strategy to combine the local internal reinforcement signals, which contain the information of future external reinforcement signals. Assume that an agent i is standing on a given state, first calculating the local internal reinforcement signals $s_i(k)$ for all the possible local distributed control actions to provide the adaptive and effective information, then determining which is the optimal control action according to the discounted cumulative local internal reinforcement signals, which is the local performance index $J_i(k)$.

B. Nash Equilibrium Analysis

Definition 2: A sequence of N control laws $\{u_1^*, u_2^*, \ldots, u_N^*\}$ is refer to as a global Nash equilibrium solution for an N multiagent system, if for all $i \in N$

$$J_i^* \triangleq J_i\left(u_i^*, u_{\bar{i}}^*\right) \leqslant J_i\left(u_i, u_{\bar{i}}^*\right) \tag{15}$$

where $u_{\bar{i}}$ denote the actions of all the other agents in the graph excluding i, namely $u_{\bar{i}} = \{u_j | j \in N, j \neq i\}$. The N-tuple $\{J_1^*, J_2^*, \ldots, J_N^*\}$ is called the Nash equilibrium of the N-player game.

According to Definition 2, the coupled discrete-time HJB equation can be expressed as

$$J_i^*(\delta_i(k)) = s_i\big(\delta_i(k), u_i^*(k), u_{-i}^*(k)\big) + \gamma J_i^*(\delta_i(k+1)) \tag{16}$$

where

$$s_i\big(\delta_i(k), u_i^*(k), u_{-i}^*(k)\big) = r_i\big(\delta_i(k), u_i^*(k), u_{-i}^*(k)\big) + \alpha s_i\big(\delta_i(k+1), u_i^*(k+1), u_{-i}^*(k+1)\big). \tag{17}$$

Now, we will prove that the designed control laws which are given in terms of the solutions of (16) provide Nash equilibrium solution for the multiagent systems.

Theorem 1: Let the graph contains a spanning tree with at least one nonzero pining gain. For $\forall i$, if $J^*(\delta_i(k))$ is a solution of the coupled discrete-time HJB equation (16) with the local internal reinforcement signals (17), and the optimal distributed control laws $u_i^*(k)$ in (14), then all agents are in Nash equilibrium.

Proof: We can further rewrite (11) and (16) as

$$\begin{aligned} J_i(\delta_i(k)) &= s_i(\delta_i(k), u_i(k), u_{-i}(k)) + \gamma J_i(\delta_i(k+1)) \\ &= \sum_{l=k}^{\infty} \gamma^{l-k} s_i(\delta_i(l), u_i(l), u_{-i}(l)) \end{aligned} \tag{18}$$

and

$$\begin{aligned} J_i^*(\delta_i(k)) &= s_i\big(\delta_i(k), u_i^*(k), u_{-i}^*(k)\big) + \gamma J_i^*(\delta_i(k+1)) \\ &= \sum_{l=k}^{\infty} \gamma^{l-k} s_i\big(\delta_i(l), u_i^*(l), u_{-i}^*(l)\big). \end{aligned} \tag{19}$$

Subtract (18) from (19), it follows:

$$\begin{aligned} J_i^*(\delta_i(k)) - J_i(\delta_i(k)) = &\sum_{l=k}^{\infty} \gamma^{l-k} s_i\big(\delta_i(l), u_i^*(l), u_{-i}^*(l)\big) \\ &- \sum_{l=k}^{\infty} \gamma^{l-k} s_i(\delta_i(l), u_i(l), u_{-i}(l)). \end{aligned} \tag{20}$$

Since the optimal local performance index for each agent is the minimal value, such that $J_i^*(\delta_i(k)) - J_i(\delta_i(k)) \leq 0$. Therefore, we have

$$\sum_{l=k}^{\infty} \gamma^{l-k} s_i\big(\delta_i(l), u_i^*(l), u_{-i}^*(l)\big) - \sum_{l=k}^{\infty} \gamma^{l-k} s_i(\delta_i(l), u_i(l), u_{-i}(l)) \leq 0. \tag{21}$$

This means

$$J_i\left(u_i^*, u_{\bar{i}}^*\right) \leqslant J_i\left(u_i, u_{\bar{i}}^*\right). \tag{22}$$

According to Definition 2, all the agents are in Nash equilibrium, which completes the proof. ■

IV. GrHDP-Based Optimal Consensus Control

In this section, GrHDP algorithm for multiagent systems is first provided to estimate $s_i(\delta_i(k))$, $J_i(\delta_i(k))$, and $u_i(k)$, respectively. Then, the convergence proof of the proposed algorithm is also presented. It is an extension from the single-agent HDP algorithm to the multiagent dynamic systems.

A. GrHDP Algorithm for Multiagent Systems

Step 1: Start with arbitrary initial admissible control laws $u_i^0(k)$.

Step 2: Once the iterative control laws $u_i^l(k)$, $\forall i$, are determined, solve for $s_i^{l+1}(\delta(k))$ by using the following equation:

$$s_i^{l+1}(\delta_i(k)) = r_i\left(\delta_i(k), u_i^l(k), u_{-i}^l(k)\right) + \alpha s_i^l(\delta_i(k+1))). \tag{23}$$

Step 3: Then, the iterative performance indices are solved by

$$J_i^{l+1}(\delta_i(k)) = s_i^{l+1}(\delta_i(k)) + \gamma J_i^l(\delta_i(k+1)). \tag{24}$$

Step 4: Update the control laws as

$$u_i^{l+1}(k) = \arg\min_{u_i(k)} \Big\{ s_i(\delta_i(k)) + \gamma J_i^l(\delta_i(k+1)) \Big\}. \quad (25)$$

Step 5: On convergence of $\|J_i^{l+1}(\delta_i(k)) - J_i^l(\delta_i(k))\|$ end. Else, let $l = l+1$ and go back to step 2.

Note that, the proposed GrHDP algorithm is an incremental optimization process which is implemented forward in time and online. The following section provides the convergence of this GrHDP algorithm.

B. Convergence Analysis of GrHDP Algorithm

Theorem 2: Assume there exist admissible control laws u_i, $\forall i$. Let the local internal reinforcement signals $s_i^l(\delta_i(k))$, performance indices $J_i^l(\delta_i(k))$, and distributed control laws $u_i^l(k)$ for all the agents be updated by (23)–(25), respectively. Then:

1) the sequence $J_i^l(\delta_i(k))$ for each agent is monotonic convergence;
2) there exist finite upper bounds $\overline{M}$ and $\overline{U}$ for sequences $s_i^l(\delta_i(k))$ and $J_i^l(\delta_i(k))$, i.e., $0 \le s_i^l(\delta_i(k)) \le \overline{M}$ and $0 \le J_i^l(\delta_i(k)) \le \overline{U}$.

Proof: For $\forall i$ and $\forall l$, consider the sequence which is given by

$$\Psi_i^{l+1}(\delta_i(k)) = \tau_i^{l+1}(\delta_i(k)) + \gamma \Psi_i^l(\delta_i(k+1)) \quad (26)$$

where

$$\tau_i^{l+1}(\delta_i(k)) = r_i\Big(\delta_i(k), \mu_i^l(k), \mu_{-i}^l(k)\Big) + \alpha \tau_i^l(\delta_i(k+1))) \quad (27)$$

in which $\mu_i^l(k)$ and $\mu_{-i}^l(k)$ are the given arbitrary stabilizing and admissible control laws for agent i and its neighbors, $r_i(\delta_i(k), \mu_i^l(k), \mu_{-1}^l(k)) = \delta_i^T(k) Q_{ii} \delta_i(k) + \mu_i^T(k) R_{ii} \mu_i(k) + \sum_{j \in N_i} \mu_j^T R_{ij} \mu_j(k)$.

Notice that, $u_i^l(k)$ is any stabilizing and admissible control sequence and minimizes the right-hand side of (24). Hence, by setting $\tau_i^0 = s_i^0 = 0$, $\Psi_i^0 = J_i^0 = 0$, we have $0 \le J_i^l(\delta_i(k)) \le \Psi_i^l(\delta_i(k))$. In the following part, we will show that $J_i^{l+1}(\delta_i(k)) \ge \Psi_i^l(\delta_i(k))$ by mathematical induction.

Starting with $l = 0$ and setting $\tau_i^0 = s_i^0 = 0$, $\Psi_i^0 = J_i^0 = 0$, it yields

$$\begin{aligned} J_i^1(\delta_i(k)) - \Psi_i^0(\delta_i(k)) &= s_i^1(\delta_i(k)) \\ &= r_i\Big(\delta_i(k), u_i^0(k), u_{-i}^0(k)\Big) \\ &\ge 0 \end{aligned} \quad (28)$$

which means $J_i^1(\delta_i(k)) \ge \Psi_i^0(\delta_i(k))$.

Now, assume that there exists $J_i^l(\delta_i(k)) \ge \Psi_i^{l-1}(\delta_i(k))$ for the $(l-1)$th iteration step. Then, by setting the stabilizing and admissible control law $\mu_i^{l-1} = u_i^l(k)$ and the summation of external reinforcement signal $\tau_i^l(k+1) = s_i^l(\delta_i(k))$. We have

$$\begin{aligned} \Psi_i^l(\delta_i(k)) = r_i\Big(\delta_i(k), u_i^l(k), u_{-i}^l(k)\Big) \\ + \alpha s_i^l(\delta_i(k+1)) + \gamma \Psi_i^{l-1}(\delta_i(k+1)). \end{aligned} \quad (29)$$

Consider (23) and (24), it follows:

$$\begin{aligned} J_i^{l+1}(\delta_i(k)) = r_i\Big(\delta_i(k), u_i^l(k), u_{-i}^l(k)\Big) \\ + \alpha s_i^l(\delta_i(k+1)) + \gamma J_i^l(\delta_i(k+1)). \end{aligned} \quad (30)$$

Hence, by subtracting (29) from (30), we obtain

$$\begin{aligned} J_i^{l+1}(\delta_i(k)) - \Psi_i^l(\delta_i(k)) \\ = \gamma\Big(J_i^l(\delta_i(k+1)) - \Psi_i^{l-1}(\delta_i(k+1))\Big) \ge 0. \end{aligned} \quad (31)$$

This indicates that $J_i^{l+1}(\delta_i(k)) \ge \Psi_i^l(\delta_i(k))$, $\forall i$. Combining with the conclusion that $0 \le J_i^l(\delta_i(k)) \le \Psi_i^l(\delta_i(k))$, we obtain $0 \le J_i^l(\delta_i(k)) \le \Psi_i^l(\delta_i(k)) \le J_i^{l+1}(\delta_i(k))$, namely, $0 \le J_i^l(\delta_i(k)) \le J_i^{l+1}(\delta_i(k))$. Hence, the sequence $J_i^l(\delta_i(k))$ is a monotonically nondecreasing sequence. This completes the proof of part (1).

Notice that the sequence $J_i^l(\delta_i(k))$ is positive and monotonically nondecreasing. Hence, we can conclude that

$$0 \le J_i^l(\delta_i(k)) \le J_i^\infty(\delta_i(k)). \quad (32)$$

Set $\sigma_i(k)$ and $\sigma_{-i}(k)$ be any stabilizing and admissible control laws for agent i and its neighbors. A new sequence ϕ is defined as

$$\phi_i^{l+1}(\delta_i(k)) = r_i(\delta_i(k), \sigma_i(k), \sigma_{-i}(k)) + \alpha \phi_i^l(\delta_i(k+1)). \quad (33)$$

We can further rewrite (33) as

$$\begin{aligned} \phi_i^{l+1}(\delta_i(k)) &= r_i(\delta_i(k), \sigma_i(k), \sigma_{-i}(k)) + \alpha \phi_i^l(\delta_i(k+1)) \\ &= r_i(\delta_i(k), \sigma_i(k), \sigma_{-i}(k)) \\ &\quad + \alpha r_i(\delta_i(k+1), \sigma_i(k+1), \sigma_{-i}(k+1)) \\ &\quad + \alpha^2 \phi_i^{l-1}(\delta_i(k+2)) \\ &\quad \vdots \\ &= r_i(\delta_i(k), \sigma_i(k), \sigma_{-i}(k)) \\ &\quad + \alpha r_i(\delta_i(k+1), \sigma_i(k+1), \sigma_{-i}(k+1)) \\ &\quad + \ \ldots + \alpha^l r_i(\delta_i(k+i), \sigma_i(k+i), \sigma_{-i}(k+i)) \\ &\quad + \alpha^{l+i} \phi_i^0(\delta_i(k+i+1)) \end{aligned} \quad (34)$$

with $\phi_i^0(\delta_i(k+i+1)) = 0$.

Therefore,

$$\begin{aligned} \phi_i^{l+1}(\delta_i(k)) &= \sum_{m=0}^{l} \alpha^m r_i(\delta_i(k+m), \sigma_i(k+m), \sigma_{-i}(k+m)) \\ &= \sum_{m=k}^{i+k} \alpha^{m-k} r_i(\delta_i(m), \sigma_i(m), \sigma_{-i}(m)) \\ &\le \sum_{m=k}^{\infty} \alpha^{m-k} r_i(\delta_i(m), \sigma_i(m), \sigma_{-i}(m)). \end{aligned} \quad (35)$$

Then $\forall l$, we have

$$\phi_i^{l+1}(\delta_i(k)) \le \sum_{m=k}^{\infty} \alpha^{m-k} r_i(\delta_i(m), \sigma_i(m), \sigma_{-i}(m)). \quad (36)$$

By setting $\sigma_i^l(k) = u_i^l(k)$, $\sigma_{-i}^l(k) = u_{-i}^l(k)$, $\phi_i^l(\delta_i(k+1)) = s_i^l(\delta_i(k+1))$, we have

$$s^{l+1}(\delta_i(k)) \le \sum_{m=k}^{\infty} \alpha^{m-k} r_i(\delta_i(m), \sigma_i(m), \sigma_{-i}(m)). \quad (37)$$

Define $\overline{M} = \sum_{m=k}^{\infty} \alpha^{m-k} r_i(\delta_i(m), \sigma_i(m), \sigma_{-i}(m))$, and hence $s_i^l(\delta_i(k)) \le \overline{M}$. Because sequence $s_i^l(\delta_i(k))$ is positive definite, then $0 \le s_i^l(\delta_i(k)) \le \overline{M}$, which completes the conclusion that $\overline{M}$ is the upper bound of sequence $s_i^l(\delta_i(k))$.

Now, we will show there also exists an upper bound for sequence $J_i^l(\delta_i(k))$. Rewrite (24) as

$$\begin{aligned} J_i^{l+1}(\delta_i(k)) &= s_i^{l+1}(\delta_i(k)) + \gamma J_i^l(\delta_i(k+1)) \\ &= s_i^{l+1}(\delta_i(k)) + \gamma s_i^l(\delta_i(k+1)) + \gamma^2 J_i^{l-1}(\delta_i(k+2)) \\ &\quad \vdots \\ &= s_i^{l+1}(\delta_i(k)) + \gamma s_i^l(\delta_i(k+1)) \\ &\quad + \cdots + \gamma^l s_i^i(\delta_i(k+i)) + \gamma^{i+1} J_i^0(\delta_i(k+i+1)) \\ &= \sum_{m=k}^{k+i} \gamma^{m-k} s_i^{l+k-m}(k). \end{aligned} \quad (38)$$

Because $s_i^l(\delta_i(k)) \le \overline{M}$, it yields that

$$J_i^{l+1}(\delta_i(k)) \le \sum_{m=k}^{\infty} \gamma^{m-k} \overline{M}. \quad (39)$$

Define $\overline{U} = \sum_{m=k}^{\infty} \gamma^{m-k} \overline{M}$, such that $0 \le J_i^l(\delta_i(k)) \le \overline{U}$. Note that both $\overline{M}$ and $\overline{U}$ are determined by the admissible and stabilizing control laws $\sigma_i(k)$ and $\sigma_{-i}(k)$. This means when $l \to \infty$, it follows $\delta_i(k) \to 0$, $\sigma_i(k) \to 0$, $\sigma_{-i}(k) \to 0$. Therefore $\lim_{k\to\infty} r_i(\delta_i(k), \sigma_i(k), \sigma_{-i}(k)) = 0$, indicating that $\overline{M}$ and $\overline{U}$ are finite values. Therefore, the proof of part (2) is completed. ■

Theorem 2 proves that both internal reinforcement signal sequence $s_i^l(\delta_i(k))$ and performance index sequence $J_i^l(\delta_i(k))$ exist upper bounds. Furthermore, the local performance index sequence $J_i^l(\delta_i(k))$ is also monotonically nondecreasing. This means $s_i^l(\delta_i(k))$ and $J_i^l(\delta_i(k))$ cannot go infinity. Next theorem proves that sequences $J_i^l(\delta_i(k))$ and $u_i^l(k)$ will converge to their optimal values, respectively, when $l \to \infty$.

Theorem 3: For $\forall i$ and $\forall l$, let sequences $s_i^l(\delta_i(k))$ and $J_i^l(\delta_i(k))$ be computed as (23) and (24). The arbitrary admissible control laws are given as (25). Then, as $l \to \infty$, $J_i^l(\delta_i(k))$ and $u_i^l(k)$ will converge to their optimal values, namely, $J_i^l(\delta_i(k)) \to J_i^*(\delta_i(k))$ and $u_i^l(k) \to u_i^*(k)$.

Proof: Define another new performance index sequence $\Lambda_i^l(\delta_i(k))$ as

$$\Lambda_i^{l+1}(\delta_i(k)) = \phi_i^{l+1}(\delta_i(k)) + \gamma \Lambda_i^l(\delta_i(k+1)) \quad (40)$$

where $\phi_i^{l+1}(\delta_i(k))$ is defined in (33). According to (26) and (27), we know $J_i^l(\delta_i(k)) \le \Lambda_i^l(\delta_i(k))$ by setting $\tau_i^l(\delta_i(k)) = \phi_i^l(\delta_i(k))$ and $\Psi_i^l(\delta_i(k)) = \Lambda_i^l(\delta_i(k))$. We further obtain

$$\begin{aligned} &J_i^{l+1}(\delta_i(k)) \\ &\le \phi_i^{l+1}(\delta_i(k)) + \gamma \phi_i^l(\delta_i(k+1)) + \gamma^2 \phi_i^{l-1}(\delta_i(k+2)) + \ldots \\ &= \sum_{t=0}^{l} \gamma^{t+k} \left(\sum_{m=k}^{l-k} \alpha^{m-k} r_i(\delta_i(m+k), \sigma_i(m+k), \sigma_{-i}(m+k)) \right). \end{aligned} \quad (41)$$

Let $l \to \infty$, it yields

$$\begin{aligned} &J_i^{\infty}(\delta_i(k)) \\ &\le \sum_{t=0}^{\infty} \gamma^{t+k} \left(\sum_{m=k}^{\infty} \alpha^{m-k} r_i(\delta_i(m+k), \sigma_i(m+k), \sigma_{-i}(m+k)) \right) \\ &= \mathcal{G}(\delta_i(k), \sigma_i(k), \sigma_{-i}(k)). \end{aligned} \quad (42)$$

Let $\sigma_i(k) = u_i^*(k)$, $\sigma_{-i}(k) = u_{-i}^*(k)$, then

$$J_i^{\infty}(\delta_i(k)) \le \mathcal{G}\big(\delta_i(k), u_i^*(k), u_{-i}^*(k)\big) = J_i^*(\delta_i(k)). \quad (43)$$

On the other hand, since $J_i^*(\delta_i(k))$ is the optimal performance index and the sequences $J_i^l(\delta_i(k))$ is monotonically nondecreasing, we can also attain

$$J_i^*(\delta_i(k)) \le J_i^{\infty}(\delta_i(k)). \quad (44)$$

Combining (43) and (44), it follows:

$$J_i^*(\delta_i(k)) = \lim_{l\to\infty} J_i^l(\delta_i(k)) = J_i^{\infty}(\delta_i(k)). \quad (45)$$

Now let us consider the convergence of the control law. Based on (25), we have

$$u_i^{\infty}(k) = \arg\min_{u_i(k)} \big\{ s_i(\delta_i(k)) + \gamma J_i^{\infty}(\delta_i(k+1)) \big\} \quad (46)$$

$$u_i^*(k) = \arg\min_{u_i(k)} \big\{ s_i(\delta_i(k)) + \gamma J_i^*(\delta_i(k+1)) \big\}. \quad (47)$$

Therefore, we can obtain that $\lim_{i\to\infty} u_i^l(k) = u_i^*(k)$ if (45) holds. This completes the conclusion. ■

Theorem 3 shows that the sequence $J_i^l(\delta_i(k))$ can monotonically converge to the optimal solution, which means this algorithm can be used to solve the discrete-time HJB equation (12). Furthermore, the designed control law sequence $u_i^l(k)$ can also converge to the optimal value. This means the error dynamics $\delta_i(k)$ in (9) can be driven to the optimal state, which is zero in this paper. According to Lemma 1, we also have $\eta(k) \to 0$ as $\delta(k) \to 0$, which means all the agents will synchronize to the leader dynamics (2). In the next section, the neural-network-based GrHDP implementation is explicitly developed.

V. Neural-Network-Based Implementation for Multiagent Systems

This section provides the neural-network-based implementation process of the GrHDP algorithm. Comparing with the traditional adaptive critic design [21], [29], [50], an additional neural network, goal network, is integrated to facilitate the learning process. Hence, the proposed architecture contains three neural networks for each agent, namely action network, critic network, and goal network. The action network is

designed to approximate the optimal control laws. Its structure is kept as the same as in [21], [29], and [50]. The goal network is developed to generate the internal reinforcement signals $s_i(\delta_i(k))$, which have the information of future external reinforcement signals. To closely connect the goal network with the critic network, we also set $s_i(\delta_i(k))$ to be included within the input of the critic network to help estimate the corresponding performance indices. All the neural networks designed in this paper are three-layer neural networks. Furthermore, to avoid using the model network, one step is set backward in the implementation. The following sections provide the explicit learning rules of these three neural networks.

A. Goal Network Design

In the traditional HDP design, an instant reward signal is assigned from the environment which, in this paper, is called the external reinforcement signal. In this paper, a goal network is integrated into the traditional HDP design to generate an internal reinforcement signal. According to the online algorithm in [38] and [41] for single-agent system, we define the local internal reinforcement signals for multiagent systems as

$$s_i(\delta_i(k)) = \mathcal{Y}\Big(\omega_{g2i}^T(k) \cdot \mathcal{Y}\Big(\omega_{g1i}^T(k) \cdot Z_{gi}(k)\Big)\Big) \tag{48}$$

where $Z_{gi}(k)$ is the goal network input of agent i and it is a vector of the information from $\delta_i(k)$, $u_i(k)$, and $u_{-i}(k)$, and $\omega_{g1i}(k)$ and $\omega_{g2i}(k)$ denote the input-to-hidden and hidden-to-output layer weights, respectively, of the goal network. Moreover, $\mathcal{Y}$ is a sigmoid function with the definition as

$$\mathcal{Y}(x) = \frac{1 - e^{-x}}{1 + e^{-x}}. \tag{49}$$

Note that the purpose of the sigmoid function is to constrain the output into $[-1, 1]$. In the goal network, we apply the sigmoid function on both hidden and output layer nodes.

The error function of goal network for agent i is denoted as

$$e_{gi}(k) = \alpha s_i(\delta_i(k)) - \big[s_i(\delta_i(k-1)) - r_i(\delta_i(k-1), u_i(k-1), u_{-i}(k-1))\big]. \tag{50}$$

To update the neural network weights is to minimize the following objective function:

$$E_{gi}(k) = \frac{1}{2} e_{gi}^T(k) e_{gi}(k). \tag{51}$$

Gradient descent method is adopted to minimize (51). Then, we obtain the goal network weights updating rules for agent i as

$$\omega_{gi}^{l+1}(k) = \omega_{gi}^{l}(k) - \beta_{gi}\left(\frac{\partial E_{gi}(k)}{\partial \omega_{gi}(k)}\right) \tag{52}$$

where $0 < \beta_{gi} < 1$ is the goal network learning rate. Here, we apply $\omega_{gi}(k)$ to represent both $\omega_{g1i}(k)$ and $\omega_{g2i}(k)$. Based on the chain-backpropagation rules, we derive that

$$\frac{\partial E_{gi}(k)}{\partial \omega_{gi}(k)} = \frac{\partial E_{gi}(k)}{\partial s_i(\delta_i(k))} \frac{\partial s_i(\delta_i(k))}{\partial \omega_{gi}(k)} = \alpha \cdot e_{gi}(k) \cdot \frac{\partial s_i(\delta_i(k))}{\partial \omega_{gi}(k)}. \tag{53}$$

Notice that, in the implementation process, (52) needs to be calculated in a component-by-component fashion.

We can further drive the term $\partial s_i(\delta_i(k))/\partial \omega_{gi}(k)$ for the weights between the hidden and output layers as

$$\frac{\partial s_i(\delta_i(k))}{\partial \omega_{g2i}(k)} = \frac{1}{2}\Big(1 - s_i^2(\delta_i(k))\Big)\mathcal{Y}_{gi} \tag{54}$$

and for the weights between the input and hidden layers as

$$\frac{\partial s_i(\delta_i(k))}{\partial \omega_{g1i}(k)} = \frac{1}{2}\Big(1 - s_i^2(\delta_i(k))\Big)\omega_{g2i}^T(k) \cdot \frac{1}{2}\Big(1 - \mathcal{Y}_{gi}^2\Big) Z_{gi}(k) \tag{55}$$

where $\mathcal{Y}_{gi} = \mathcal{Y}(\omega_{g1i}^T(k) \cdot Z_{gi}(k))$.

B. Critic Network Design

The local performance index $J_i(\delta_i(k))$ for each agent is estimated by the critic network. Set the input-to-hidden layer weights as $\omega_{c1i}(k)$ and the hidden-to-output layer weights as $\omega_{c2i}(k)$, then

$$J_i(\delta_i(k)) = \omega_{c2i}^T(k)\mathcal{Y}\big(\omega_{c1i}^T(k) \cdot Z_{ci}(k)\big) \tag{56}$$

where $Z_{ci}(k)$ is the input of the critic network and it is a vector of $\delta_i(k)$, $u_i(k)$, $u_{-i}(k)$, and $s_i(\delta_i(k))$. Note that we include the local internal reinforcement signal $s_i(\delta_i(k))$ into the input of the critic network to closely connect the goal network and the critic network.

The objective function of the critic network can be described as

$$e_{ci}(k) = \gamma J_i(\delta_i(k)) - [J_i(\delta_i(k-1)) - s_i(\delta_i(k-1))] \tag{57}$$

$$E_{ci}(k) = \frac{1}{2} e_{ci}^T(k) e_{ci}(k). \tag{58}$$

We notice that it is the internal reinforcement signal $s_i(\delta_i(k))$ applied to the critic network rather than the (external) reinforcement signal $r_i(\delta_i(k), u_i(k), u_{-i}(k))$ in literature. Based on the graduate decent rules, we update the critic network weights as

$$\omega_{ci}^{l+1}(k) = \omega_{ci}^{l}(k) - \beta_{ci}\left(\frac{\partial E_{ci}(k)}{\partial \omega_{ci}(k)}\right) = \omega_{ci}^{l}(k) - \beta_{ci}\left(\frac{\partial E_{ci}(k)}{\partial J_i(\delta_i(k))} \frac{\partial J_i(\delta_i(k))}{\partial \omega_{ci}(k)}\right) \tag{59}$$

where $0 < \beta_{ci} < 1$ is the critic network learning rate. Here, $\omega_{ci}(k)$ is applied to express both $\omega_{c1i}(k)$ and $\omega_{c2i}(k)$.

The term $\partial E_{ci}(k)/\partial J_i(\delta_i(k))$ in (59) can be directly obtained as $\gamma e_{ci}(k)$. Then, term $\partial J_i(\delta_i(k))/\partial \omega_{ci}(k)$ is derived as follows: for hidden-to-output layer

$$\frac{\partial J_i(\delta_i(k))}{\partial \omega_{c2i}(k)} = \mathcal{Y}_{ci} \tag{60}$$

and input-to-hidden layer

$$\frac{\partial J_i(\delta_i(k))}{\partial \omega_{c1i}(k)} = \frac{1}{2}\omega_{c2i}^T(k)\Big(1 - \mathcal{Y}_{ci}^2\Big) Z_{ci}(k) \tag{61}$$

where $\mathcal{Y}_{ci} = \mathcal{Y}(\omega_{c1i}^T(k) \cdot Z_{ci}(k))$.

C. Action Network Design

The distributed control laws are approximated by the action network as

$$u_i(k) = \mathcal{Y}\left(\omega_{a2i}^T(k) \cdot \mathcal{Y}\left(\omega_{a1i}^T(k) \cdot \delta_i(k)\right)\right) \tag{62}$$

where $\delta_i(k)$ is the local tracking error and is also the action network input for each agent, ω_{a1i} are the action network weights from the input to the hidden layer, and $\omega_{a2i}(k)$ are the action network weights from the hidden to the output layer. The sigmoid function is applied to both hidden and output layer.

Realize that the goal of the control laws are to minimize the performance index. Therefore, we set the error function as the difference between $J_i(\delta_i(k))$ and the desired ultimate cost-to-go objective

$$e_{ai}(k) = J_i(\delta_i(k)) - U_c \tag{63}$$

where U_c is the ultimate utility function. Here, since at the optimal equilibrium, both tracking errors $\delta_i(k)$ and control signals $u_i(k)$ will be drive to zero, we assume $U_c = 0$.

The objective function of the action network can be therefore denoted as

$$E_{ai}(k) = \frac{1}{2} e_{ai}^T(k) e_{ai}(k). \tag{64}$$

The weights updating rule is derived based on the gradient descent method as

$$\begin{aligned}\omega_{ai}^{l+1}(k) &= \omega_{ai}^{l}(k) - \beta_{ai}\left(\frac{\partial E_{ai}(k)}{\partial \omega_{ai}(k)}\right) \\ &= \omega_{ai}^{l}(k) - \beta_{ai}\left(\frac{\partial E_{ai}(k)}{\partial J_i(\delta_i(k))}\frac{\partial J_i(\delta_i(k))}{\partial u_i(k)}\frac{\partial u_i(k)}{\partial \omega_{ai}(k)}\right) \\ &= \omega_{ai}^{l}(k) - \beta_{ai} \cdot e_{ai}(k) \cdot \frac{1}{2}\left(1 - \mathcal{Y}_{ci}^2\right)\omega_{c1i,u}(k)\frac{\partial u_i(k)}{\partial \omega_{ai}(k)}\end{aligned} \tag{65}$$

where $0 < \beta_{ai} < 1$ is the action network learning rate and $\omega_{c1i,u}(k)$ is the input-to-hidden layer weights of the critic network corresponding to input $u_i(k)$. $\omega_{ai}(k)$ represents both $\omega_{a1i}(k)$ and $\omega_{a2i}(k)$. We have $\partial u_i(k)/\partial \omega_{a2i}(k) = 1/2(1 - u_i^2(k))\mathcal{Y}_{ai}$ for hidden-to-output layer, and $\partial u_i(k)/\partial \omega_{a1i}(k) = 1/2(1-u_i^2(k))\omega_{a2i}^T \cdot 1/2(1-\mathcal{Y}_{ai}^2)\delta_i(k)$, where $\mathcal{Y}_{ai} = \mathcal{Y}(\omega_{a1i}^T(k) \cdot \delta_i(k))$.

In this GrHDP design, the training process is in the order of the goal network, the critic network, and the action network. Specifically, for each agent i, after the weights ω_{g1i} and ω_{g2i} of the goal network are learned, we fix them thereafter and start to train the weights ω_{c1i} and ω_{c2i} of the critic network. Then, we fix ω_{c1i} and ω_{c2i} and start to train the weights ω_{a1i} and ω_{a2i} of the action network. In this learning process, the information of the explicit system functions are not required. There is also no any identification scheme needed. Only the state and control input data of the current and past time steps are used. This is important since the exact information of system functions are difficult to obtain in many practical situations. Furthermore, this learning process is conducted online with adaptive capability, so that the optimal control laws can still be determined when the system parameters are changed. In the next section, simulations have shown the effectiveness of this method.

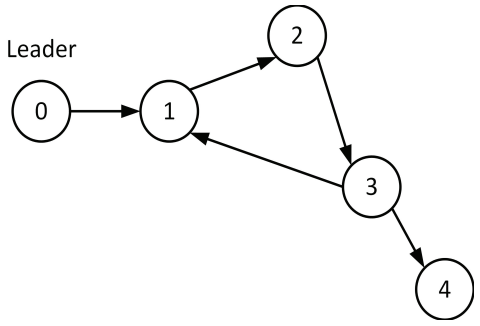
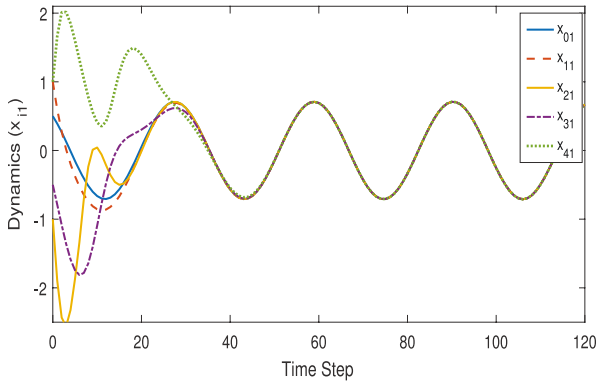


Fig. 1. Communication network structure of four-agent dynamic system.



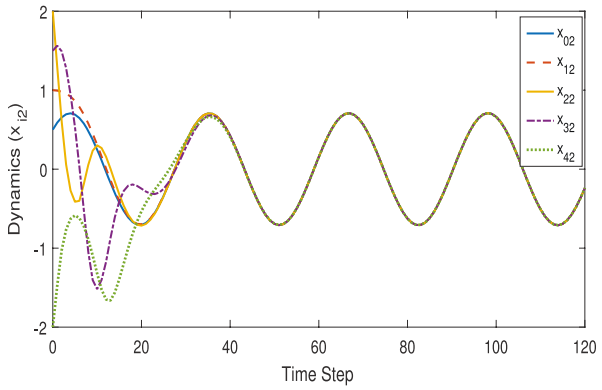


Fig. 2. Dynamics of the leader and follower agents.

VI. Simulation Study

A. Four-Agent System

First, we consider a four-agent system with the communication network structure shown in Fig. 1. The plant and input matrices for each agent are provided as

$$A = \begin{bmatrix} 0.9801 & -0.1987 \\ 0.1987 & 0.9801 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; B_2 = \begin{bmatrix} 1 \\ 0.9 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} 0 \\ 0.8 \end{bmatrix}, \; B_4 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}.$$

The pining gains are chosen as $q_1 = 1$ and $q_2 = q_3 = q_4 = 0$, and the edge weights are given by $p_{21} = 1$, $p_{32} = 1$, $p_{13} = 1$, and $p_{43} = 1$. Furthermore, define the matrices in the performance indices as $Q_{11} = Q_{22} = Q_{33} = Q_{44} = I_{2\times 2}$,

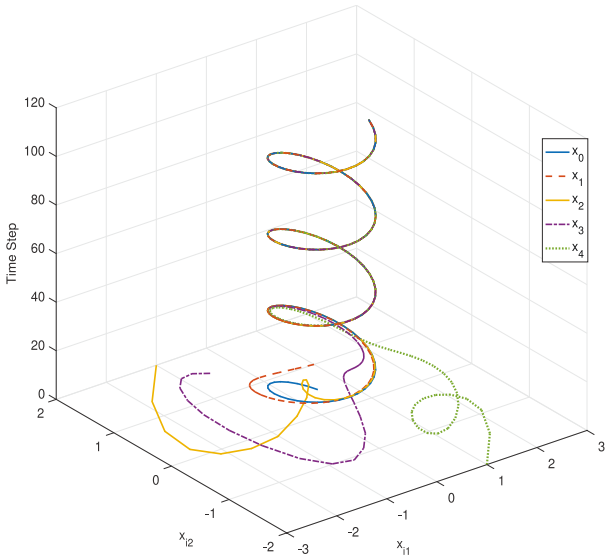


Fig. 3. 3-D phase plane plot of the states in four-agent system.

$R_{11} = R_{22} = R_{33} = R_{44} = 1$, and $R_{21} = R_{32} = R_{13} = R_{43} = 1$.

The developed GrHDP method is applied to solve this multiagent problem. Three-layer neural networks are designed for each agent as the goal, the critic, and the action network. The learning rates are chosen as $\beta_{gi} = \beta_{ci} = \beta_{ai} = 0.005$, $i = 1, 2, 3, 4$. The discount factors for the local internal reinforcement signals and local performance indices are chosen as $\alpha = \gamma = 0.95$. Let the initial states of each agent in the system be

$$x_1(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; x_2(0) = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$
$$x_3(0) = \begin{bmatrix} -0.5 \\ 1.5 \end{bmatrix} x_4(0) = \begin{bmatrix} 1 \\ -2 \end{bmatrix}.$$

The dynamics of the leader and all the follower agents are provided in Fig. 2. It is shown that all the agents start from different initial states and can synchronize to the leader dynamics after a few time steps. Fig. 3 shows the phase plane plot of these four agents. All the trajectories converge to the desired dynamics (leader). In this learning process, the designed control laws for these four agents are presented in Fig. 4. The iterative trajectories of performance index for each agent at $k = 1$ are provided in Fig. 5. Furthermore, in order to show the effectiveness of our developed method, we compare our results with the traditional HDP method. Define $x_{ei} = x_i - x_0$, $i = 1, 2, 3, 4$, which is the tracking error between each follower agent and the leader. Fig. 6 shows the comparison of the tracking errors for both GrHDP and HDP methods. We can observe our developed method can quickly push the tracking errors vanish in the learning process. This means our GrHDP method can achieve better performance in the consensus control process.

B. Ten-Agent System

In this section, a ten-agent system is considered to show the effectiveness of our proposed method. The designed

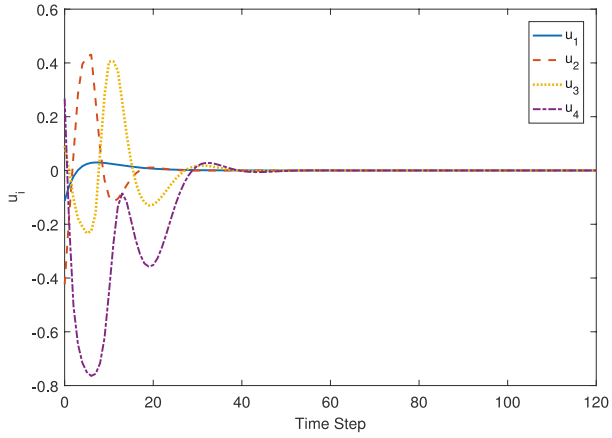


Fig. 4. Evolution of control laws for each follower agent.

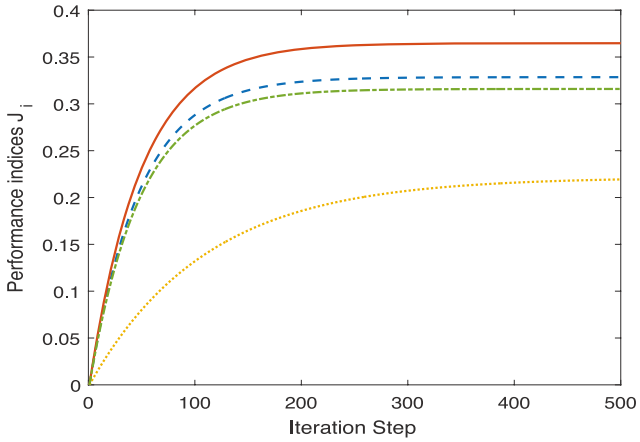


Fig. 5. Trajectories of performance indices when $k = 1$ for each agent.

communication network digraph is present in Fig. 7. Agent 0 is the leader with the system function as

$$x_0(k+1) = Ax_0(k) \tag{66}$$

where $A = \begin{bmatrix} 0.995 & -0.09983 \\ 0.09983 & 0.995 \end{bmatrix}$. Agent 1 can receive the information from the leader, while other agents 2–10 can only receive the information from itself and its neighbors. For instance, agent 2 can receive the information from itself and its neighbors agents 1 and 4. The system functions for agents 1–10 can be described as

$$x_i(k+1) = Ax_i(k) + B_i u_i(k), \quad i = 1, 2, \ldots, 10 \tag{67}$$

where

$$B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 0.9 \end{bmatrix}, B_3 = \begin{bmatrix} 0 \\ 0.8 \end{bmatrix}, B_4 = \begin{bmatrix} 0.25 \\ 0.27 \end{bmatrix}$$
$$B_5 = \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix}, B_6 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, B_7 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, B_8 = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$$
$$B_9 = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}, B_{10} = \begin{bmatrix} 0.199 \\ 1 \end{bmatrix}.$$

According to the communication network provided in Fig. 7, we define the pining gain as $q_1 = 1$ and $q_i = 0$, $i = 2, 3, \ldots, 10$, and the edge weights $p_{21} = 1$, $p_{24} = 1$, $p_{32} = 1$, $p_{43} = 1$, $p_{54} = 1$, $p_{5,10} = 1$, $p_{65} = 1$, $p_{76} = 1$, $p_{87} = 1$,

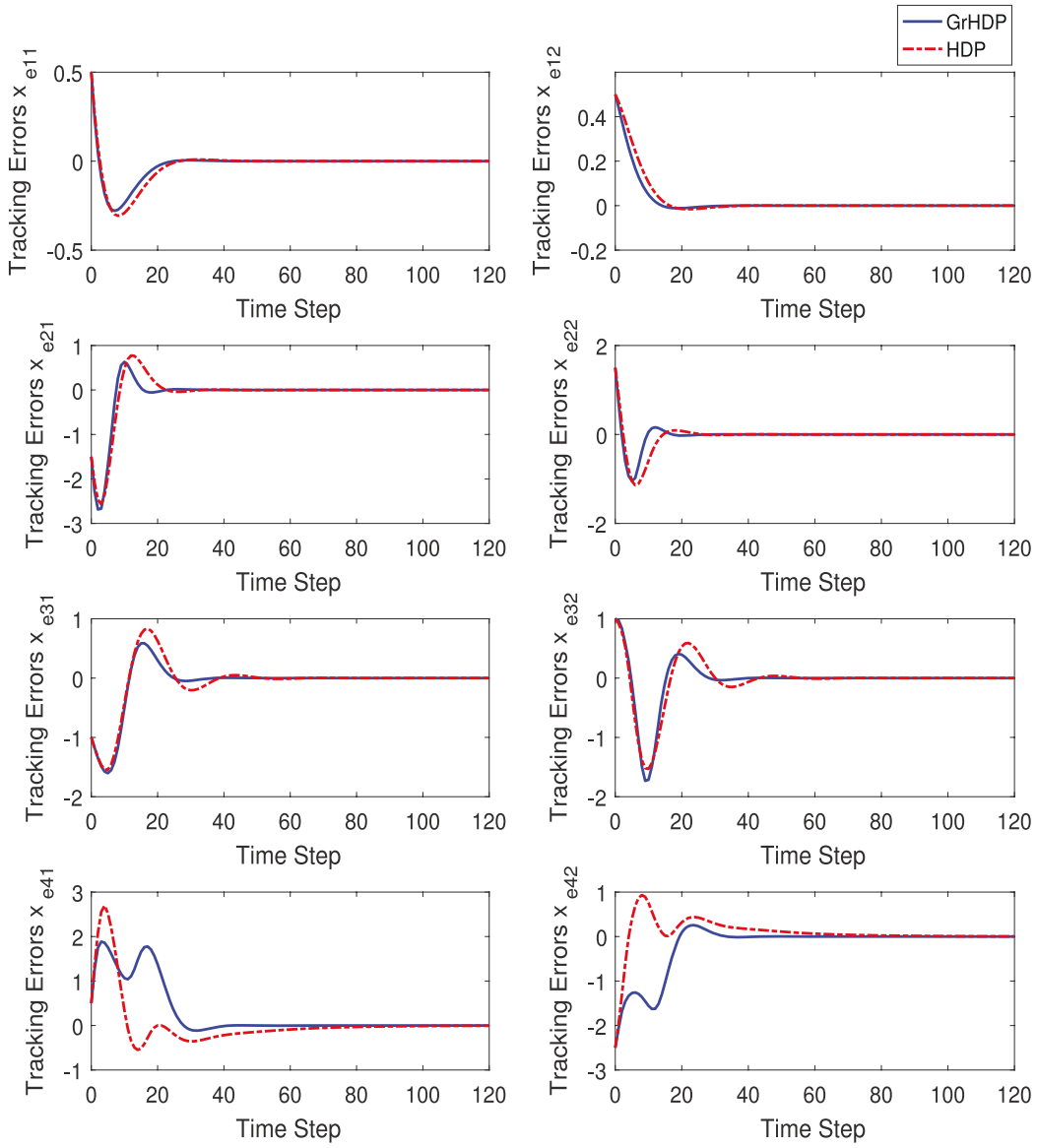


Fig. 6. Tracking error comparisons between the GrHDP method and HDP method.

$p_{98} = 1$, and $p_{10,9} = 1$. The weighting matrices in the performance indices are chosen as $Q_{ii} = I_{2\times 2}$, $R_{ii} = 1$, for $i = 1, 2, \ldots, 10$, $R_{21} = R_{24} = R_{32} = R_{43} = R_{54} = R_{5,10} = R_{65} = R_{76} = R_{87} = R_{98} = R_{10,9} = 1$.

The GrHDP method is applied to control system (67). The goal, critic, and action networks are designed for each agent. Choose the learning rates as $\beta_{gi} = \beta_{ci} = \beta_{ai} = 0.005$, $i = 1, \ldots, 10$. Set the discount factors as $\alpha = \gamma = 0.95$ for local internal reinforcement signals and local performance indices, respectively. Let the initial states of each agent be

$$x_1(0) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, x_2(0) = \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}, x_3(0) = \begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix}$$
$$x_4(0) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, x_5(0) = \begin{bmatrix} 0.8 \\ 0.5 \end{bmatrix}, x_6(0) = \begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix}$$
$$x_7(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, x_8(0) = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}, x_9(0) = \begin{bmatrix} 0 \\ 0.8 \end{bmatrix}$$
$$x_{10}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, x_0(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

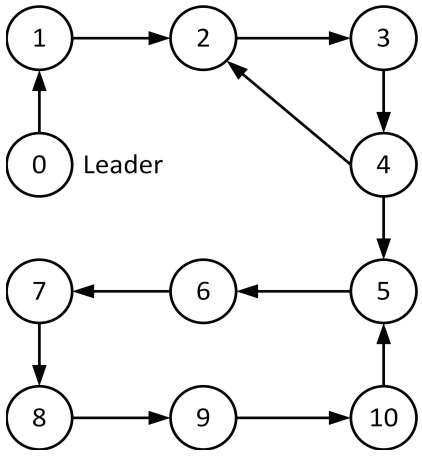


Fig. 7. Communication network structure of ten-agent dynamic system.

Figs. 8 and 9 show the dynamics of the system states. The corresponding three-dimensional (3-D) phase plane plot of all the agents are provided in Fig. 10. We can observe that the

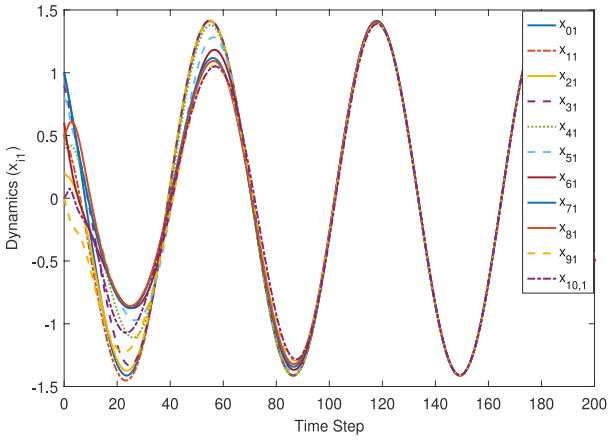


Fig. 8. Dynamics x_{i1} of the leader and follower agents.

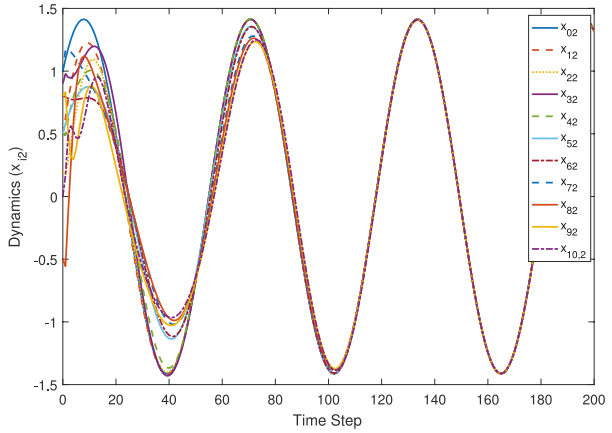


Fig. 9. Dynamics x_{i2} of the leader and follower agents.

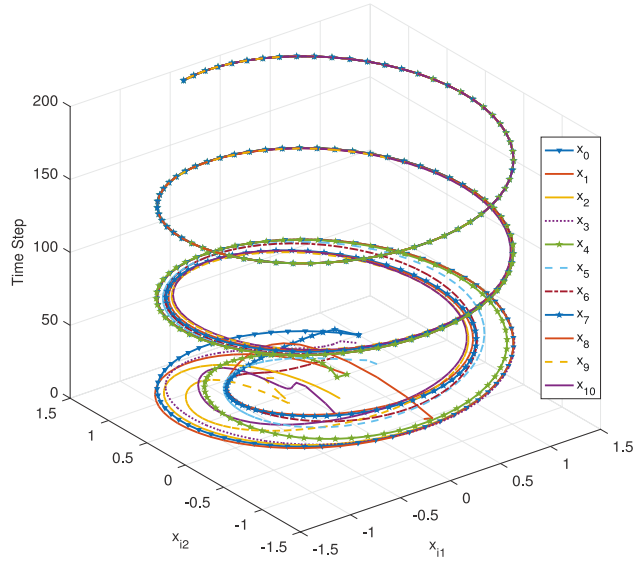


Fig. 10. 3-D phase plane plot of the states in ten-agent system.

proposed method can make the follower agent states track the desired state trajectories. Furthermore, Figs. 11 and 12 show that the tracking errors x_{ei} between the follower agents and the leader system are vanish after about 120 time steps. All the

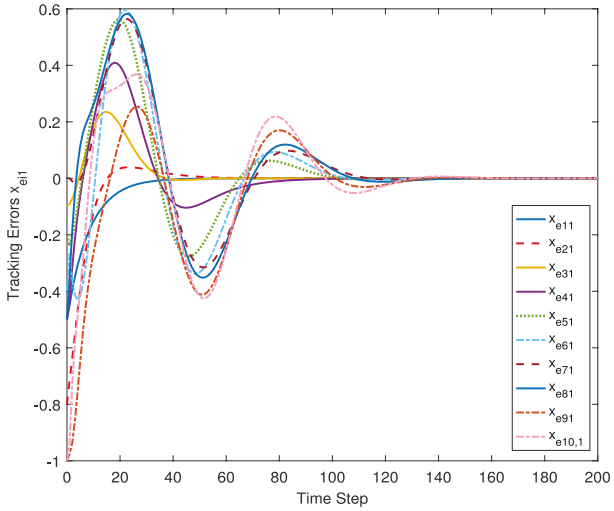


Fig. 11. Tracking errors x_{ei1} of ten agents.

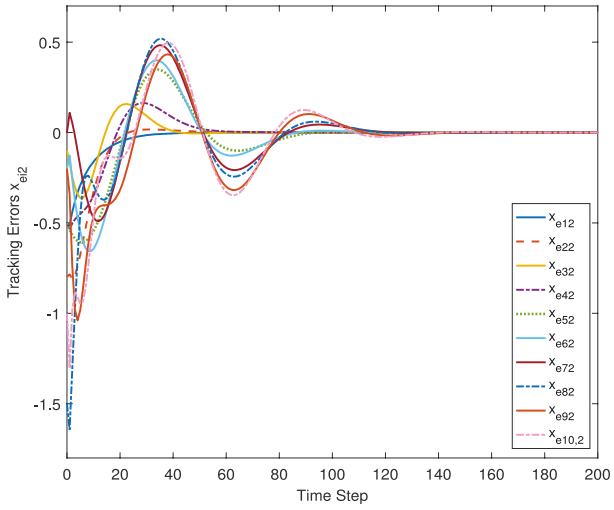


Fig. 12. Tracking errors x_{ei2} of ten agents.

simulation results establish that the designed GrHDP method is effective in consensus control problem.

VII. Conclusion

In this paper, we investigated a class of multiagent discrete-time dynamic systems and designed a new online consensus control method by GrHDP techniques. The proposed method only required the current and past data rather than the explicit information of system models. The theoretical analysis of the proposed method was developed to demonstrate the convergence of the local performance indices and the boundedness of local internal reinforcement signals. Two simulation examples were provided to show the effectiveness of the proposed method.

Although, in this paper, we improved the performance of learning-based consensus control problem, there still exist a number of ongoing challenges for multiagent systems in a distributed environment. For instance, in this paper, we assume the data is public and available for other agents at any time. Autonomous systems, however, usually encapsulate personal

information describing their principle, and therefore communication and learning among the various autonomous agents involve dealing with privacy and security issues [51], [52]. We are interested to research the data privacy in multiagent systems. In addition, we are extending the learning-based multiagent consensus control design in disturbance environment to research the robustness of this method.

REFERENCES

[1] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 427–438, Feb. 2013.

[2] J. Lin, A. S. Morse, and B. D. O. Anderson, "The multi-agent rendezvous problem—The asynchronous case," in *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, vol. 2, 2004, pp. 1926–1931.

[3] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[4] C. L. P. Chen, G.-X. Wen, Y.-J. Liu, and F.-Y. Wang, "Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1217–1226, Jun. 2014.

[5] G. Wen, C. L. P. Chen, Y.-J. Liu, and Z. Liu, "Neural network-based adaptive leader-following consensus control for a class of nonlinear multiagent state-delay systems," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 2151–2160, Aug. 2017.

[6] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, Los Angeles, CA, USA, 2005, p. 9.

[7] V. Lesser, C. L. Ortiz, Jr., and M. Tambe, Eds., *Distributed Sensor Networks: A Multiagent Perspective*. Boston, MA, USA: Kluwer Academic, 2003.

[8] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. Amer. Control Conf.*, Portland, OR, USA, 2005, pp. 1859–1864.

[9] J. Zhu, J. Lu, and X. Yu, "Flocking of multi-agent non-holonomic systems with proximity graphs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 1, pp. 199–210, Jan. 2013.

[10] W. Meng, Q. Yang, J. Sarangapani, and Y. Sun, "Distributed control of nonlinear multiagent systems with asymptotic consensus," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 5, pp. 749–757, May 2017.

[11] H. Wang, X. Liao, T. Huang, and C. Li, "Cooperative distributed optimization in multiagent networks with delays," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 2, pp. 363–369, Feb. 2015.

[12] C. L. P. Chen, C.-E. Ren, and T. Du, "Fuzzy observed-based adaptive consensus tracking control for second-order multiagent systems with heterogeneous nonlinear dynamics," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 4, pp. 906–915, Aug. 2016.

[13] Z. Feng, G. Wen, and G. Hu, "Distributed secure coordinated control for multiagent systems under strategic attacks," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1273–1284, May 2017.

[14] G.-P. Liu, "Consensus and stability analysis of networked multiagent predictive control systems," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1114–1119, Apr. 2017.

[15] W. Zhang, W. Liu, C. Zang, and L. Liu, "Multiagent system-based integrated solution for topology identification and state estimation," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 714–724, Apr. 2017.

[16] M. I. Abouheaf, F. L. Lewis, K. G. Vamvoudakis, S. Haesaert, and R. Babuska, "Multi-agent discrete-time graphical games and reinforcement learning solutions," *Automatica*, vol. 50, no. 12, pp. 3038–3053, 2014.

[17] H. Zhang, H. Jiang, Y. Luo, and G. Xiao, "Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4091–4100, May 2017.

[18] Q. Wei, D. Liu, and F. L. Lewis, "Optimal distributed synchronization control for continuous-time heterogeneous multi-agent differential graphical games," *Inf. Sci.*, vol. 317, pp. 96–113, Oct. 2015.

[19] H. Zhang, J. Zhang, G.-H. Yang, and Y. Luo, "Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 1, pp. 152–163, Feb. 2015.

[20] M. I. Abouheaf and F. L. Lewis, "Approximate dynamic programming solutions of multi-agent graphical games using actor-critic network structures," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Dallas, TX, USA, 2013, pp. 1–8.

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

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

[23] D. Liu, Y. Zhang, and H. G. 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.

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

[25] H. G. Zhang, Q. L. Wei, and Y. H. 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, Aug. 2008.

[26] C.-F. Juang and C.-M. Lu, "Ant colony optimization incorporated with fuzzy q-learning for reinforcement fuzzy control," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 3, pp. 597–608, May 2009.

[27] P. J. Werbos, "Intelligence in the brain: A theory of how it works and how to build it," *Neural Netw.*, vol. 22, no. 3, pp. 200–212, 2009.

[28] P. J. Werbos, "Foreword—ADP: The key direction for future research in intelligent control and understanding brain intelligence," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 898–900, Aug. 2008.

[29] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

[30] 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.

[31] 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.

[32] 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.

[33] X. Zhong, Z. Ni, Y. Tang, and H. He, "Data-driven partially observable dynamic processes using adaptive dynamic programming," in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn. (ADPRL)*, Orlando, FL, USA, 2015, pp. 1–8.

[34] X. Zhong, H. He, and D. V. Prokhorov, "Robust controller design of continuous-time nonlinear system using neural network," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Dallas, TX, USA, Aug. 2013, pp. 1–8.

[35] Y.-J. Liu, Y. Gao, S. Tong, and C. L. P. Chen, "A unified approach to adaptive neural control for nonlinear discrete-time systems with nonlinear dead-zone input," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 1, pp. 139–150, Jan. 2016.

[36] D. Liu, D. Wang, D. Zhao, Q. Wei, and N. Jin, "Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 628–634, Jul. 2012.

[37] D. Liu and D. Wang, "Optimal control of unknown nonlinear discrete-time systems using the iterative globalized dual heuristic programming algorithm," in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Hoboken, NJ, USA: Wiley, 2013, pp. 52–74.

[38] H. He, Z. Ni, and J. Fu, "A three-network architecture for on-line learning and optimization based on adaptive dynamic programming," *Neurocomputing*, vol. 78, no. 1, pp. 3–13, 2012.

[39] H. He, *Self-Adaptive Systems for Machine Intelligence*. Hoboken, NJ, USA: Wiley, 2011.

[40] Z. Ni, H. He, X. Zhong, and D. V. Prokhorov, "Model-free dual heuristic dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1834–1839, Aug. 2015.

[41] X. Zhong, Z. Ni, and H. He, "A theoretical foundation of goal representation heuristic dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2513–2525, Dec. 2016.

[42] 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.

[43] Z. Ni, H. He, J. Wen, and X. Xu, "Goal representation heuristic dynamic programming on maze navigation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 2038–2050, Dec. 2013.
[44] Z. Ni and H. He, "Heuristic dynamic programming with internal goal representation," *Soft Comput.*, vol. 17, no. 11, pp. 2101–2108, 2013.
[45] Y. Tang, C. Mu, and H. He, "SMES-based damping controller design using fuzzy-GrHDP considering transmission delay," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 7, pp. 1–6, Oct. 2016.
[46] Z. Ni, H. He, D. Zhao, X. Xu, and D. V. Prokhorov, "GrDHP: A general utility function representation for dual heuristic dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 614–627, Mar. 2015.
[47] X. Zhong, Z. Ni, and H. He, "Gr-GDHP: A new architecture for globalized dual heuristic dynamic programming," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3318–3330, Oct. 2017.
[48] X. Luo, J. Si, and Y. Zhou, "An integrated design for intensified direct heuristic dynamic programming," in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn. (ADPRL)*, Singapore, 2013, pp. 183–190.
[49] J. Chen and Z. Li, "A novel adaptive tropism reward ADHDP method with robust property," in *Advances in Brain Inspired Cognitive Systems*. Heidelberg, Germany: Springer, 2013, pp. 288–295.
[50] F. Liu, J. Sun, J. Si, W. Guo, and S. Mei, "A boundedness result for the direct heuristicdynamic programming," *Neural Netw.*, vol. 32, pp. 229–235, Aug. 2012.
[51] J. M. Such, A. Espinosa, and A. García-Fornes, "A survey of privacy in multi-agent systems," *Knowl. Eng. Rev.*, vol. 29, no. 3, pp. 314–344, 2014.
[52] K. Mivule, D. Josyula, and C. Turner, "An overview of data privacy in multi-agent learning systems," in *Proc. 5th Int. Conf. Adv. Cogn. Technol. Appl.*, 2013, pp. 14–20.

Xiangnan Zhong (M'17) received the B.S. and M.S. degrees in automation, and control theory and control engineering from Northeastern University, Shenyang, China, in 2010 and 2012, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA.

She is currently an Assistant Professor with the Department of Electrical Engineering, University of North Texas, Denton, TX, USA. Her current research interests include computational intelligence, reinforcement learning, cyber-physical systems, networked control systems, neural network, and optimal control.

Ms. Zhong was a recipient of the Chinese Government Award for Outstanding Students Abroad by Chinese Government in 2017, and the URI Enhancement of Graduate Research Award in 2016. She has been actively involved in numerous conference and workshop organization committees in the society.

Haibo He (SM'11–F'18) 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. He has published one sole-author research book (Wiley), edited one book (Wiley-IEEE) and six conference proceedings (Springer), and authored and coauthored over 280 peer-reviewed journal and conference papers. His current research interests include computational intelligence, machine learning, data mining, and various 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, the National Science Foundation CAREER Award in 2011, and the Providence Business News "Rising Star Innovator Award" in 2011. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.