This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

On Distributed Nash Equilibrium Computation: Hybrid Games and a Novel Consensus-Tracking Perspective

Maojiao Ye, *Member, IEEE*, Le Yin, Guanghui Wen, *Senior Member, IEEE*, and Yuanshi Zheng

***Abstract*—With the incentive to solve Nash equilibrium computation problems for networked games, this article tries to find answers for the following two problems: 1) how to accommodate hybrid games, which contain both continuous-time players and discrete-time players? and 2) are there any other potential perspectives for solving continuous-time networked games except for the consensus-based gradient-like algorithm established in our previous works? With these two problems in mind, the study of this article leads to the following results: 1) a hybrid gradient search algorithm and a consensus-based hybrid gradient-like algorithm are proposed for hybrid games with their convergence results analytically investigated. In the proposed hybrid strategies, continuous-time players adopt continuous-time algorithms for action updating, while discrete-time players update their actions at each sampling time instant and 2) based on the idea of consensus tracking, the Nash equilibrium learning problem for continuous-time games is reformulated and two new computation strategies are subsequently established. Finally, the proposed strategies are numerically validated.**

***Index Terms*—Consensus tracking, continuous games, hybrid games, Nash equilibrium learning.**

I. Introduction

As AN effective analysis tool to accommodate the cooperation and confliction among multiple interacting decision-making entities, game theory is active in fields varying from the smart grid, the wireless communication networks, and economic markets, to cloud computing (see [1]–[3]). Stimulated by the extensive application fields for games, Nash equilibrium learning has received a lot of attention in the last few years. Quite a few Nash equilibrium computation strategies have been reported among which the methods in [4]–[8] are in discrete time while the algorithms in [2], [3], and [10]–[18] are in continuous time.

Together, with a parallel update method, a random update method as well as a gradient update method were introduced in [4] to address flow control games. Games with limited information flow were addressed by gossip-based Nash equilibrium computation strategies in [5]. Lemke's method was adopted in [6] to deal with games with shared constraints. Through measuring the players' cost functions, an extremum seeker was designed in a discrete-time scenario to solve games subject to linear dynamic constraints [7]. By utilizing the ideas from projection and primal–dual techniques, Zhu and Frazzoli [8] solved the generalized convex games. Operator theory was employed in [9] to accommodate aggregative games with shared convex constraints. Continuous-time repeated matrix games were accommodated by dynamic fictitious play and gradient play in [12]. A sinusoidal probing-based extremum seeker was designed in [3] for nonmodeled-based Nash equilibrium computation. Energy consumption among a network of price-anticipating electricity users was modeled as an aggregative game, which was addressed by a dynamic average consensus-based strategy in [2]. General noncooperative games were further investigated under undirected communication graphs [13], switching communication topologies [14], and weight-balanced digraphs [15], respectively. Continuous-time and discrete-time algorithms were, respectively, designed in [16] and [17] for two-network zero-sum games. Furthermore, an N-cluster noncooperative game was developed and solved through designing singularly perturbed dynamics in [18]–[20].

The above works presented some clearsighted ideas to achieve Nash equilibrium computation through designing either discrete-time algorithms or continuous-time algorithms. Nevertheless, a lot of real engineering systems are of hybrid characteristics and intrinsically multimodal containing both continuous-time and discrete-time subsystems (see [22]–[28]). Take the heating and cooling system as an example. In the heating and cooling system, subsystems, including the air

Manuscript received October 11, 2019; revised April 1, 2020; accepted June 14, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61803202, Grant 61773303, and Grant 61963032; in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180455; in part by the Nature Science Basic Research Plan in Shaanxi Province of China under Grant 2018JQ6058; and in part by the Fundamental Research Funds for the Central Universities under Grant 30920032203. This article was presented in part at the International Conference on Control, Automation, Robotics and Vision, 2018 [30]. This article was recommended by Associate Editor X.-M. Zhang. *(Corresponding author: Maojiao Ye.)*

Maojiao Ye is with the School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: ye0003ao@e.ntu.edu.sg).

Le Yin is with the School of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: yinle0002@swu.edu.cn).

Guanghui Wen is with the School of Mathematics, Southeast University, Nanjing 210096, China, and also with the School of Engineering, RMIT University, Melbourne, VIC 3001, Australia (e-mail: wenguanghui@gmail.com).

Yuanshi Zheng is with the Center for Complex Systems, School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China (e-mail: zhengyuanshi2005@163.com).

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

Digital Object Identifier 10.1109/TCYB.2020.3003372

conditioner, the furnace, as well as the home heat loss dynamics are continuous whereas the thermostat is a discrete-time device [27]. Systems composed of both digital and analog components [28] are also typical examples that fit the considered scenario. As the existing Nash equilibrium computation strategies are established for games with either discrete-time or continuous-time dynamics, hybrid games still remain to be addressed. Therefore, Nash equilibrium computation for a class of hybrid games is considered in this article. In particular, both continuous-time players and discrete-time players exist in the considered game. Literally, the continuous-time players are of continuous-time dynamics while the discrete-time players are of discrete-time dynamics. Note that in the considered games, the continuous-time players and discrete-time players are interacting with each other. However, their actions evolve in different time scenarios. Hence, it is challenging to find a unified framework to simultaneously analyze their behaviors, especially when the players are of complex dynamics.

Besides the consideration of games in hybrid systems, this article also presents some novel insights into the design of Nash equilibrium learning strategies for continuous-time games. From the existing works, we have learned that distributed Nash equilibrium computation can be achieved by driving the players' estimation variables to a consensus state, which is the Nash equilibrium point. Similarly, consensus-tracking problems with time-varying reference signals are concerned with a network of agents that are governed to follow some reference trajectories (see [29]). These two observations motivate us to explore the potential linkages between distributed Nash equilibrium learning problems and consensus-tracking problems. To find out the answer, two new Nash equilibrium computation strategies are developed from a consensus-tracking perspective. The proposed consensus-tracking-based Nash equilibrium computation strategies relieve the gap between distributed Nash equilibrium learning problems and consensus-tracking problems.

In summary, with part of the paper presented in [30], the article has the following contributions.

1) A hybrid gradient search algorithm, in which discrete-time players amend their actions by discrete-time gradient search while continuous-time ones take samples on the gradient values and update their actions by utilizing the sampled gradient values, is first proposed for the *hybrid games*. Then, games under distributed networks (see [13]) were further accommodated by introducing consensus protocols to the hybrid gradient search algorithm. The resulting learning method avoids the usage of centralized information (i.e., the other players' actions) and hence, it is suitable for applications in distributed systems.
2) The article proposes two novel distributed Nash equilibrium computation strategies for *continuous-time games* by adapting the ideas from consensus tracking. In the consensus-tracking-based algorithms, the players' actions, which are generated via the gradient play, are treated as time-varying tracking reference signals. Based on the reference signals, two consensus-tracking protocols are adapted to achieve the distributed Nash equilibrium learning for continuous-time games.
3) By constructing the Lyapunov candidate functions, the stability of the Nash equilibrium is analytically explored for all the developed methods.

We arrange the remaining sections as follows. In Section II, the notations and preliminaries are given. Together with a hybrid gradient search algorithm, a distributed hybrid gradient-like algorithm is presented in Section III to learn the Nash equilibrium for the considered hybrid games. Moreover, two novel consensus-tracking-based Nash equilibrium learning strategies are presented in Section IV. Numerical verifications of the theoretical results are given in Section V. Finally, Section VI provides the conclusions for this article.

II. Notations and Preliminaries

Notations: The set of real numbers is represented by $\mathbb{R}$. A column one vector of dimension N is denoted as 1_N and an identity matrix of dimension $M \times M$ is denoted as $I_{M\times M}$. In addition, $\mathrm{diag}\{k_{ij}\}(\mathrm{diag}\{k_i\})$, where $k_{ij} \in \mathbb{R}(k_i \in \mathbb{R})$ and $i, j \in \{1, 2, \ldots, N\}$, is a diagonal matrix and its diagonal elements are $k_{11}, k_{12}, \ldots, k_{1N}, k_{21}, \ldots, k_{NN}(k_1, k_2, \ldots, k_N)$, successively. Likewise, $[k_{ij}]_{vec}([k_i]_{vec})$ defines a column vector with its elements being $k_{ij}(k_i)$, successively. Moreover, $\lambda_{\min}(P)$ denotes the minimum eigenvalue of P, where P is a symmetric real matrix. Furthermore, $\max_{i\in\{1,2,\ldots,N\}}\{k_i\}$ equals the maximum value of k_i for $i \in \{1, 2, \ldots, N\}$. In the remainder, we utilize $\otimes$ to denote the Kronecker product.

Algebraic Graph Theory: Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, in which $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of nodes and $\mathcal{E}$ is the set of edges, be a graph. In particular, the elements in $\mathcal{E}$ are denoted by (i, j), which stands for an edge from i to j. Let the weight on each edge $(i, j) \in \mathcal{E}$ be $a_{ji} > 0$. Note that $a_{ii} = 0$, implying that the graph has no self-loop. The graph is undirected if $a_{ij} = a_{ji}\ \ \forall i, j \in \mathcal{N}$. Furthermore, we say that the undirected graph is connected given that for any pair of distinct vertices, there is a path. The matrix with its element on the ith row and jth column being $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$, else, $a_{ij} = 0$ is called the adjacency matrix of graph $\mathcal{G}$, and is written as $\mathcal{A}$. In addition, $\mathcal{L} = \mathcal{D} - \mathcal{A}$, in which $\mathcal{D}$ is a diagonal matrix whose ith diagonal element is $\sum_{j=1}^{M} a_{ij}$, defines the Laplacian matrix of $\mathcal{G}$ [13].

Definition 1 (A Normal Form Game): A normal form game Γ is outlined as $\Gamma \triangleq \{\mathcal{N}, X, f\}$, where $\mathcal{N}$ defines the set of N players, $X = X_1 \times \cdots \times X_N$, $X_i \subseteq \mathbb{R}$ ($X_i = \mathbb{R}$ in this article) represents the set of actions for player i, and $f = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_N(\mathbf{x}))$, where $f_i(\mathbf{x})$ stands for player i's cost function, $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ and $x_i \in X_i$ stands for the action of player i [13].

Definition 2 (Nash Equilibrium): An action profile $\mathbf{x}^* = (x_i^*, \mathbf{x}_{-i}^*) \in X$ is Nash equilibrium given that

$$f_i\left(x_i^*, \mathbf{x}_{-i}^*\right) \le f_i\left(x_i, \mathbf{x}_{-i}^*\right) \quad \forall i \in \mathcal{N} \tag{1}$$

for all $x_i \in X_i$ [13].

Note that in (1), $\mathbf{x}_{-i} = [x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N]^T$. In the rest, we may write $(x_i, \mathbf{x}_{-i})$ as $\mathbf{x}$ for notational convenience.

III. Distributed Nash Equilibrium Learning for Hybrid Games

In the following, a Nash equilibrium learning problem for hybrid games will be first stated. Following the problem formulation, two hybrid Nash equilibrium learning algorithms will be given.

A. Problem Statement

This section considers an N-player hybrid game. Among the engaged players, $m(m < N)$ players are of continuous-time dynamics whereas the remaining $N-m$ players are of discrete-time dynamics. Let $\mathcal{C}$ and $\mathcal{D}$ ($\mathcal{C} \cup \mathcal{D} = \mathcal{N}$ and $\mathcal{C} \cap \mathcal{D} = \emptyset$), respectively, be the sets of continuous-time players and discrete-time players. Then, the dynamics of the players can be described by

$$\begin{aligned} \dot{x}_i(t) &= u_i(t), i \in \mathcal{C} \\ x_i(t_{k+1}) &= x_i(t_k) + u_i(t_k), \quad i \in \mathcal{D} \end{aligned} \tag{2}$$

in which $x_i \in \mathbb{R}$ and u_i denote player i's action and control input, respectively. In addition, $t_k = hk$, for $k = 0, 1, 2, \ldots,$ denote the sampling time instants and h is the sampling period. This section aims to design control inputs such that the players' actions can be driven to the Nash equilibrium of the hybrid game.

For notational convenience, let

$$\mathcal{P}(\mathbf{x}) = \left[\frac{\partial f_1(\mathbf{x})}{\partial x_1}, \frac{\partial f_2(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial f_N(\mathbf{x})}{\partial x_N}\right]^T.$$

The following conditions will be utilized to establish the convergence results.

Assumption 1: For each $i \in \mathcal{N}$, $f_i(\mathbf{x})$ is a $\mathcal{C}^2$ function and $([\partial f_i(\mathbf{x})]/\partial x_i)$ is globally Lipschitz with constant $\bar{l}_i$ for $\mathbf{x} \in \mathbb{R}^N$.

Assumption 2: For $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$

$$(\mathbf{x} - \mathbf{z})^T (\mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{z})) \geq L \|\mathbf{x} - \mathbf{z}\|^2 \tag{3}$$

where L is a positive constant.

Assumption 3: The players are provided with an undirected and connected communication graph $\mathcal{G}$.

Remark 1: Assumption 2 is a commonly utilized condition in related works. By this assumption, the game admits a unique Nash equilibrium [13], [34]. For more detailed elaborations on this condition, we refer interested readers to [8], [13]–[18], and [34].

In the upcoming sections, a hybrid gradient search algorithm and a distributed learning strategy will be investigated, successively.

B. Hybrid Gradient Search Algorithm

To compute the Nash equilibrium, we design the learning strategy as

$$\begin{aligned} \dot{x}_i(t) &= -\bar{\alpha}_i \frac{\partial f_i}{\partial x_i}(\mathbf{x}(t_k)), \quad i \in \mathcal{C}, \quad t \in (t_k, t_{k+1}] \\ x_i(t_{k+1}) &= x_i(t_k) - \alpha_i \frac{\partial f_i}{\partial x_i}(\mathbf{x}(t_k)), \quad i \in \mathcal{D} \end{aligned} \tag{4}$$

where $\bar{\alpha}_i$ for all $i \in \mathcal{C}$ are fixed positive control gains and α_i for $i \in \mathcal{D}$ are the step sizes to be determined. Moreover, $(\partial f_i/\partial x_i)(\mathbf{q}) = ([\partial f_i(\mathbf{x})]/\partial x_i)|_{\mathbf{x}=\mathbf{q}}$ throughout the rest of this article. Then, the following theorem can be obtained.

Theorem 1: There exist $h^* > 0$ and $\alpha^* > 0$ such that for each $h \in (0, h^*)$, $\alpha_i \in (0, \alpha^*)$, $i \in \mathcal{D}$, $\mathbf{x}(t_k) \to \mathbf{x}^*$ as $k \to \infty$ under (4) given that Assumptions 1 and 2 hold.

Proof: The conclusion can be obtained by constructing the Lyapunov candidate function as

$$V(\mathbf{x}(t_k)) = \left(\mathbf{x}(t_k) - \mathbf{x}^*\right)^T \alpha^{-1} \left(\mathbf{x}(t_k) - \mathbf{x}^*\right) \tag{5}$$

where $\alpha = \text{diag}\{\alpha_i\}, i \in \mathcal{N}$. ■

This section accommodates the Nash equilibrium computation problem for the considered hybrid games via a gradient search algorithm. In the gradient search algorithm, continuous-time players continuously amend their actions by utilizing sampled gradient information. In contrast, discrete-time players renew their actions at sampling time instants. Nevertheless, in (4), the gradient values might possibly depend on all the players' actions. Hence, (4) might be unsuitable for games in distributed networks (see also [13]). Therefore, similar to [13], we further adapt the gradient search algorithm for games under distributed networks in the upcoming section.

C. Leader-Following Consensus-Based Gradient Search Algorithm

The exploitation of the distributed Nash equilibrium learning strategy will be investigated for hybrid games in this section. To address hybrid games, the gradient search algorithm in [13] is adapted as

$$\begin{aligned} \dot{x}_i(t) &= -\delta \bar{\alpha}_i \frac{\partial f_i}{\partial x_i}(x_i(t_k), \mathbf{y}_{-i}(t_k)) \\ \dot{y}_{ij}(t) &= -\left(\sum_{l=1}^{N} a_{il}\left(y_{ij}(t_k) - y_{lj}(t_k)\right) + a_{ij}\left(y_{ij}(t_k) - x_j(t_k)\right)\right) \end{aligned} \tag{6}$$

for $i \in \mathcal{C}$, $j \in \mathcal{N}$, and $t \in (t_k, t_{k+1}]$, where δ denotes a small positive control gain to be determined, $\bar{\alpha}_i$ is a fixed positive parameter, and $\mathbf{y}_{-i} = [y_{i1}, y_{i2}, \ldots, y_{i,i-1}, y_{i,i+1}, \ldots, y_{iN}]^T$. In addition, the discrete-time players learn the Nash equilibrium by utilizing

$$\begin{aligned} x_i(t_{k+1}) &= x_i(t_k) - \delta \alpha_i \frac{\partial f_i}{\partial x_i}(x_i(t_k), \mathbf{y}_{-i}(t_k)) \\ y_{ij}(t_{k+1}) &= y_{ij}(t_k) - h\left(\sum_{l=1}^{N} a_{il}\left(y_{ij}(t_k) - y_{lj}(t_k)\right)\right. \\ &\quad \left. + a_{ij}\left(y_{ij}(t_k) - x_j(t_k)\right)\right), \quad i \in \mathcal{D}, \quad j \in \mathcal{N} \end{aligned} \tag{7}$$

where α_i is a fixed positive constant.

Let $\mathcal{H} = I_{N^2 \times N^2} - h(\mathcal{L} \otimes I_{N \times N} + A_0)$, where $A_0 = \text{diag}\{a_{ij}\}, i, j \in \mathcal{N}$ and n be a positive integer. Then, by following the proof of [21, Lemma 3.1], the following supportive lemma can be obtained.

Lemma 1: Suppose that

$$h < \min_{i,j \in \mathcal{N}} \frac{1}{\sum_{l=1}^{N} a_{il} + a_{ij}}. \tag{8}$$

Then, $\|\mathcal{H}^n\|_\infty < 1$ and all eigenvalues of $\mathcal{H}$ are within the unit circle if Assumption 3 holds.

By utilizing Lemma 1, the upcoming result can be established.

Theorem 2: Given that

$$h < \min_{i,j \in \mathcal{N}} \frac{1}{\sum_{l=1}^{N} a_{il} + a_{ij}} \tag{9}$$

there exists a positive constant $\delta^*(h)$ so that for any $\delta \in (0, \delta^*)$, $\mathbf{x}(t_k) \to \mathbf{x}^*$ and $\mathbf{y}(t_k) \to \mathbf{1}_N \otimes \mathbf{x}(t_k) \to \mathbf{1}_N \otimes \mathbf{x}^*$ as $k \to \infty$ under (6) and (7) if Assumptions 1–3 hold.

Proof: See Appendix A for the proof. ■

Remark 2: The learning strategy in (6) and (7) is adapted from the learning strategy in [13]. Nevertheless, unlike the strategy in [13] in which the players require continuous-time communication, only discrete-time communication is utilized in (6) and (7). Hence, the proposed strategy is suitable for systems where continuous-time communication is not available.

In this section, the gradient play and the Nash equilibrium computation strategy in [13] are adapted to accommodate the considered hybrid games. In the subsequent section, some new ideas will be investigated to provide more insights on how to solve the Nash equilibrium learning problems for continuous-time games.

IV. Consensus-Tracking Perspectives for Distributed Nash Equilibrium Learning

The core idea of many existing distributed Nash equilibrium seeking strategies can be summarized as follows. First, a consensus protocol is included to estimate the required information by utilizing local communication among the players. Then, based on the estimated values, gradient-like algorithms are implemented to learn the Nash equilibrium in a distributed way (see [2], [13], [14], [18], [19], and the references therein). This idea is relatively simple and might be restrictive to some extent. Hence, we ask the following question: are there any other perspectives to distributively solve noncooperative games? To answer this question, we take a continuous-time game (i.e., all players are continuous-time players) as an example and further propose consensus-tracking perspectives for distributed Nash equilibrium computation problems in this section. To be more clear, the following problem is formulated.

A. Problem Statement

Assume that in game Γ, the dynamics of player i can be described by

$$\dot{x}_i = u_i \quad \forall i \in \mathcal{N} \tag{10}$$

in which $x_i \in \mathbb{R}$ and u_i denote the player i's action and control input, respectively. This session aims to propose novel consensus-tracking perspectives for the design of the control inputs such that $\mathbf{x}(t) \to \mathbf{x}^*$ as $t \to \infty$.

The main principle of achieving distributed Nash equilibrium computation lies in developing a method that drives the players' estimates on the Nash equilibrium to the actual Nash equilibrium point. Moreover, the Nash equilibrium is stable by utilizing the gradient play under certain conditions (see [13]). Hence, it can be formulated as a tracking problem in which each player generates an estimation vector $\mathbf{y}_i$ that tracks a reference signal $\mathbf{s}(t)$ produced by

$$\dot{\mathbf{s}}(t) = -\mathcal{P}(\mathbf{s}(t)). \tag{11}$$

In the following, we present two new consensus-tracking-based Nash equilibrium learning strategies based on this idea.

B. Consensus-Tracking-Based Nash Equilibrium Learning Algorithms

Consensus-Tracking-based Nash Equilibrium Learning Algorithm 1: Motivated by the consensus-tracking protocol in [29], the new Nash equilibrium learning strategy is designed as

$$\begin{aligned}
\dot{x}_i(t) &= -\delta\alpha_i \frac{\partial f_i}{\partial x_i}(x_i, \mathbf{y}_{-i}) \\
\dot{y}_{ij}(t) &= \frac{1}{\sum_{k=1}^{N} a_{ik} + a_{ij}} \sum_{k=1}^{N} a_{ik}\big(\dot{y}_{kj} - \gamma\big(y_{ij} - y_{kj}\big)\big) \\
&\quad + \frac{a_{ij}}{\sum_{k=1}^{N} a_{ik} + a_{ij}} \big(\dot{x}_j - \gamma\big(y_{ij} - x_j\big)\big)
\end{aligned} \tag{12}$$

for $i, j \in \mathcal{N}$, where γ is a fixed positive constant, α_i for $i \in \mathcal{N}$ are fixed positive constants, and δ is a small positive parameter.

The upcoming theorem illustrates the convergence result for the method in (12).

Theorem 3: There exists a positive constant δ^* such that for each $\delta \in (0, \delta^*)$, the equilibrium $(\mathbf{x}^*, \mathbf{1}_N \otimes \mathbf{x}^*)$ is globally exponentially stable under (12) if Assumptions 1–3 hold.

Proof: See Appendix B for the proof. ■

Alternatively, one can treat γ in (12) as a positive constant to be further determined and define δ as a fixed positive constant to derive the following corollary.

Corollary 1: There exists a positive constant γ^* so that for $\gamma > \gamma^*$, the equilibrium $(\mathbf{x}^*, \mathbf{1}_N \otimes \mathbf{x}^*)$ is globally exponentially stable under (12) if Assumptions 1–3 hold.

Consensus-Tracking-Based Nash Equilibrium Learning Algorithm 2: The Nash equilibrium seeking strategy can also be designed as

$$\begin{aligned}
\dot{x}_i &= -\delta\alpha_i \frac{\partial f_i}{\partial x_i}(x_i, \mathbf{y}_{-i}) \\
\dot{y}_{ij} &= \frac{1}{\sum_{k=1}^{N} a_{ik}} \sum_{k=1}^{N} a_{ik}\big(\dot{y}_{kj} - \gamma\big(y_{ij} - y_{kj}\big)\big) \quad \forall j \neq i \\
y_{ii} &= x_i
\end{aligned} \tag{13}$$

where γ is a positive constant, α_i for $i \in \mathcal{N}$ are fixed positive constants, and δ is a small positive parameter.

The upcoming theorem illustrates the convergence result for the method in (13).

Theorem 4: There exists a positive constant δ^* such that for $\delta \in (0, \delta^*)$, the equilibrium $(\mathbf{x}^*, \mathbf{1}_N \otimes \mathbf{x}^*)$ is globally exponentially stable under (13) if Assumptions 1–3 hold.

Proof: See Appendix C for the proof. ■

Similar to Corollary 1, one can treat γ in (13) as a positive constant to be further determined and define δ as a fixed positive constant to derive the following corollary.

Corollary 2: There exists a positive constant γ^* so that for $\gamma > \gamma^*$, the equilibrium $(\mathbf{x}^*, \mathbf{1}_N \otimes \mathbf{x}^*)$ is globally exponentially stable under (12) if Assumptions 1–3 hold.

Remark 3: This section provides two novel methods that achieve distributed Nash equilibrium learning in continuous-time games. In the proposed methods, the trajectories generated by the gradient play are treated as time-varying reference signals and two consensus-tracking protocols are adopted to achieve the objective. It is worth noting that the proposed algorithms are nontrivial compared with related works.

1) As the information available for each player is restricted, the estimated gradients, which are functions of all the players' estimates, are utilized in the proposed methods. Hence, the time-varying reference signals are trajectories depending on the players' estimates. This is a core difference between the proposed methods and the consensus-tracking protocols studied in [29] in which the reference trajectory does not depend on the agents' states. Moreover, in (12) and (13), the consensus-tracking part cannot be independently analyzed from the gradient search part as they are interacting with each other.
2) Compared with the distributed Nash equilibrium learning algorithm in [13] and [14], the newly developed algorithms in (12) and (13) have advantages in the sense that they have great potential to be adapted to deal with quadratic time-varying objective functions (see [33]). Moreover, they lay solid foundations for the accommodations of games under time-varying communication topologies, especially for jointly connected topologies, which are still open questions to be addressed. Last but not least, the novel consensus-tracking-based algorithms bridge the gaps between distributed Nash equilibrium seeking problems and consensus-tracking problems. They open the door toward the distributed Nash equilibrium seeking for more complex games by taking the advantages of the existing consensus-tracking algorithms.

Remark 4: Theorems 3 and 4 indicate that δ should be selected to be sufficiently small to ensure the stabilities of the proposed learning strategies in (12) and (13). Alternatively, the stability results can be obtained by selecting γ to be sufficiently large as illustrated in Corollaries 1 and 2. Note that the quantifications of δ^* can be found in the proofs of the theorems. In addition, the quantifications of γ^* in Corollaries 1 and 2 can be obtained in a similar way. From the proofs, it can be seen that the determinations of δ^* or γ^* depend on the communication topology, the Lipschitz constants of players' objective functions, the number of the players, and the strong monotonicity constant L.

Remark 5: In this article, we consider the Nash equilibrium seeking under Assumption 2, which ensures the uniqueness of the Nash equilibrium. Note that if there are multiple Nash equilibria, it can still be obtained that the Nash equilibrium that satisfies [13, Assumptions 3 and 4] is exponentially stable.

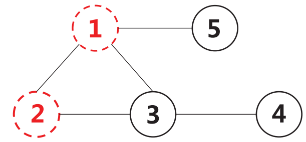

Fig. 1. Illustration of the communication topology for the players.

Moreover, it is worth mentioning that the presented results can be directly adapted to the case in which $x_i \in \mathbb{R}^m$, where $m > 1$ is an integer, though we suppose that $x_i \in \mathbb{R}$ for presentation simplicity.

Remark 6: Though the method in (6) and (7) and the consensus-tracking-based algorithms in (12) and (13) are all developed based on the gradient play, the ideas behind the designs are different. The seeking strategy in (6) and (7) is designed based on the idea that a leader-following consensus protocol can be included to estimate $\mathbf{x}$ such that the gradient play can be implemented in a distributed fashion (see [13] and [14]). Differently, the consensus-tracking-based algorithms in (12) and (13) are designed based on the idea that each player i can generate a local estimate $\mathbf{y}_i(t)$ to track a time-varying trajectory that would converge to the Nash equilibrium.

V. Numerical Simulations

In this section, a 5-player game is considered. The cost functions associated with the players are

$$\begin{aligned}
f_1(\mathbf{x}) &= \frac{1}{12}x_1^4 + 5x_1^2 - 5x_1x_2 + 10\\
f_2(\mathbf{x}) &= 5x_2^2 - 5x_1x_2 + x_2x_3 - \frac{4}{5}x_2 + 5\\
f_3(\mathbf{x}) &= \frac{5}{4}x_3^2 - x_3(x_1 + x_2 + x_4 + x_5) - \frac{7}{3}x_3 + 2\\
f_4(\mathbf{x}) &= x_4^2 + x_4x_5 + x_4 + 1\\
f_5(\mathbf{x}) &= x_5^2 + x_4x_5 + 5
\end{aligned} \tag{14}$$

respectively.

Setting the pseudogradient vector $\mathcal{P}(\mathbf{x})$ to 0 gives the unique Nash equilibrium, which is $\mathbf{x}^* = [0, 0, (4/5), -(2/3), (1/3)]^T$. In the following simulations for the hybrid games, we assume that the dynamics of players 1 and 2 are in discrete time while the dynamics of players 3–5 are in continuous time. Moreover, in Section V-C, all players are considered to be continuous-time players for the verification of consensus-tracking-based Nash equilibrium seeking strategies. In the subsequent simulations, the communication graph is depicted in Fig. 1. The hybrid gradient search algorithm, the hybrid leader-following consensus-based gradient-like algorithm, and the consensus-tracking-based methods will be numerically verified successively.

A. Hybrid Gradient Search Algorithm

This section conducts numerical simulations for the hybrid gradient search algorithm in (4). In the simulation, the initial value of $\mathbf{x}$ is selected as $\mathbf{x}(0) = [3, -4, -5, -2, 2]^T$. Moreover, the sampling period is selected as $h = 0.1$.

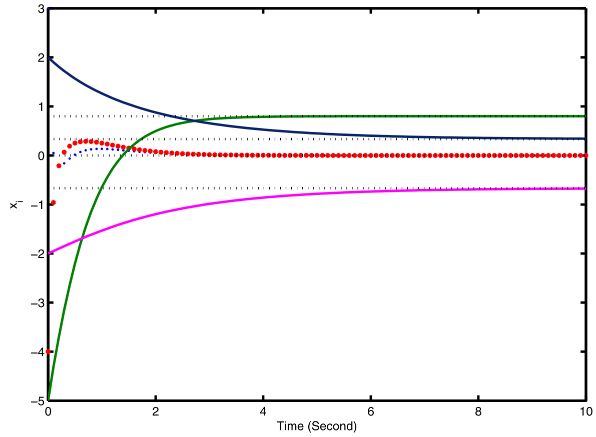

Fig. 2. Trajectories of the players' actions $x_i(t), i \in \{1, 2, \ldots, 5\}$, produced by the hybrid gradient search algorithm in (4).

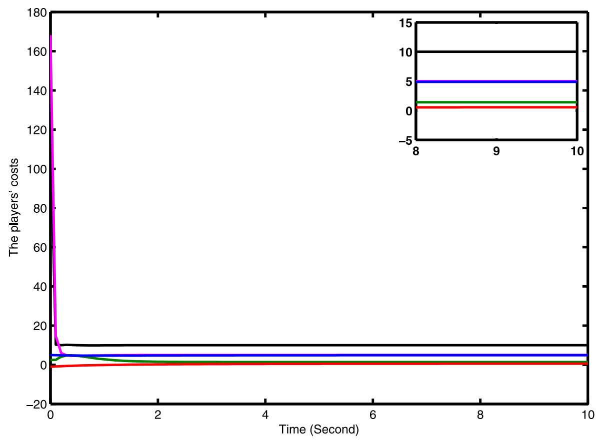

Fig. 3. Trajectories of the players' cost functions produced by the hybrid gradient search algorithm in (4).

Generated by the method in (4), the simulation results are plotted in Figs. 2 and 3 in which Fig. 2 illustrates the players' actions and Fig. 3 shows the trajectories of the players' costs.

Fig. 2 demonstrates that actuated by the gradient search algorithm in (4), the players' actions would converge to the Nash equilibrium of the hybrid game thus providing numerical verification for the result in Theorem 1.

B. Leader-Following Consensus-Based Hybrid Gradient Search Algorithm

This section simulates the hybrid leader-following consensus-based gradient search algorithm. In the simulation, we set $\mathbf{x}(0)$ to 0. The trajectories of the players' actions and their cost functions produced by (6) and (7) are illustrated in Figs. 4 and 5, respectively.

Fig. 4 illustrates that driven by the algorithm in (6) and (7), $\mathbf{x}(t) \to \mathbf{x}^*$ as $t \to \infty$ thus verifying the convergence result in Theorem 2.

C. Consensus-Tracking-Based Algorithms

This section simulates the consensus-tracking-based strategies in (12) and (13). Initializing $\mathbf{x}$ at $[3, -4, -5, -2, 2]^T$, the players' actions generated by (12) are given in Fig. 6 and the output values of their cost functions are plotted in Fig. 7. Fig. 6

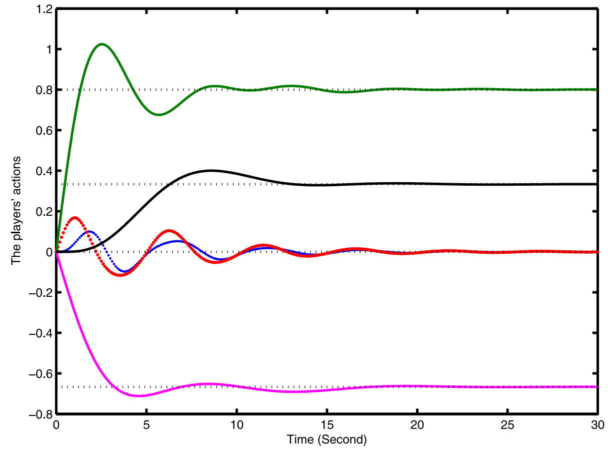

Fig. 4. Trajectories of the players' actions $x_i(t), i \in \{1, 2, \ldots, 5\}$, produced by the leader-following consensus-based hybrid gradient search algorithm in (6) and (7).

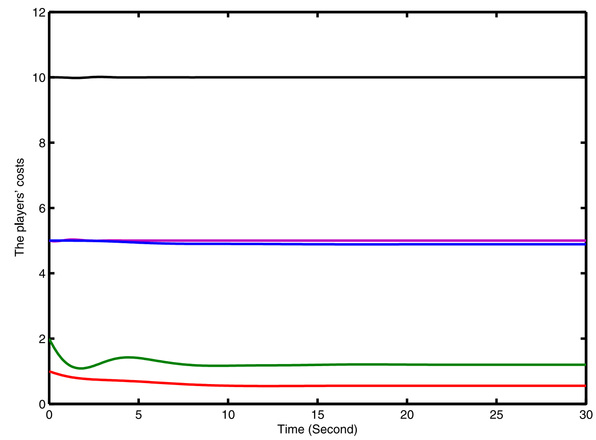

Fig. 5. Trajectories of the players' cost functions produced by the leader-following consensus-based hybrid gradient search algorithm in (6) and (7).

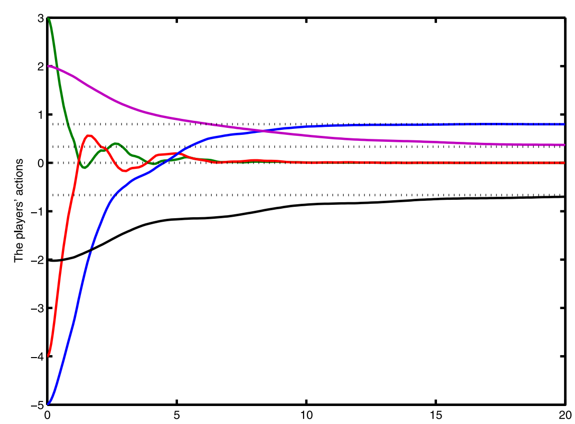

Fig. 6. Trajectories of the players' actions $x_i(t), i \in \{1, 2, \ldots, 5\}$, generated by the consensus-tracking-based method in (12).

demonstrates that by adopting (12), $\mathbf{x}(t) \to \mathbf{x}^*$ as $t \to \infty$. Hence, Theorem 3 is numerically verified.

Likewise, with the same initial conditions, the trajectories of the players' actions and their costs produced by (13) are given in Figs. 8 and 9, from which we see that the players' actions are driven to the Nash equilibrium asymptotically. Therefore, Theorem 4 is numerically validated.

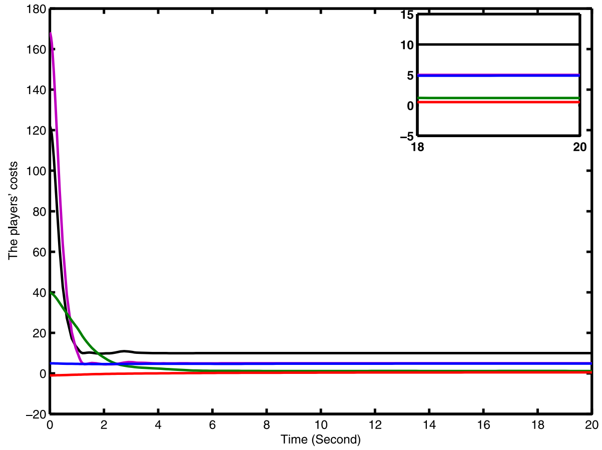

Fig. 7. Trajectories of the players' cost functions generated by the consensus-tracking-based method in (12).

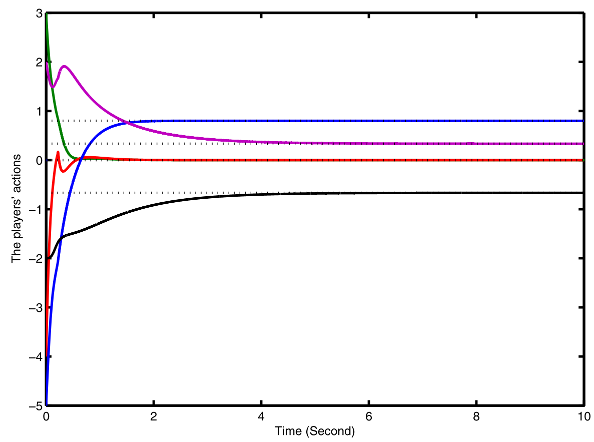

Fig. 8. Trajectories of the players' actions $x_i(t)$, $i \in \{1, 2, \ldots, 5\}$, generated, by the consensus-tracking-based method in (13).

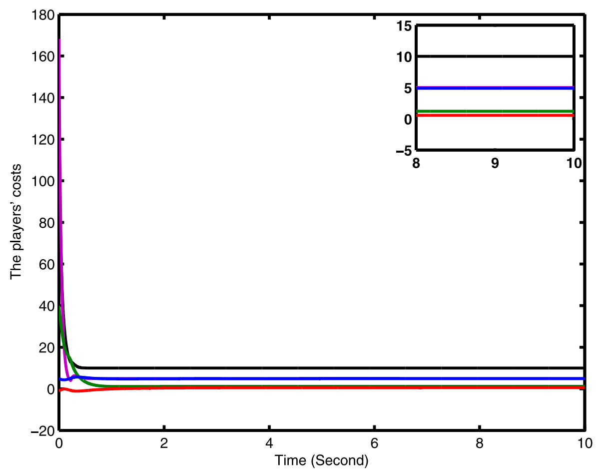

Fig. 9. Trajectories of the players' cost functions generated by the consensus-tracking-based method in (13).

VI. Conclusion

Two Nash equilibrium computation problems were investigated in this article. In the first problem, the considered game was hybrid in the sense that some engaged players were of discrete-time dynamics while the rest were of continuous-time dynamics. Correspondingly, we proposed a hybrid gradient search algorithm and a consensus-based hybrid gradient search algorithm to accommodate the hybrid games. Moreover, this article proposed a novel design perspective for distributed Nash equilibrium computation under distributed communication networks by utilizing the ideas from consensus tracking. In particular, the distributed Nash equilibrium seeking problem was formulated as a consensus-tracking problem in which the players' estimates on the other players' actions were driven to time-varying reference states, which are the trajectories of the gradient play. This provided a linkage between consensus-tracking problems and distributed Nash equilibrium seeking problems for noncooperative games. Under the given conditions, the convergence results of the hybrid seeking strategies and the consensus-tracking-based algorithms were theoretically analyzed. It would be interesting future works to consider games with more general dynamics (e.g., [35]).

Appendix A
Proof of Theorem 2

Taking integrations on both sides of (6) over $(t_k, t]$, where $t \in (t_k, t_{k+1}]$, we obtain that

$$\begin{aligned}\int_{t_k}^{t} \dot{x}_i(\tau)d\tau &= -\int_{t_k}^{t} \delta\bar{\alpha}_i \frac{\partial f_i}{\partial x_i}(x_i(t_k), \mathbf{y}_{-i}(t_k))d\tau \\ \int_{t_k}^{t} \dot{y}_{ij}(\tau)d\tau &= -\int_{t_k}^{t}\Bigg(\sum_{l=1}^{N} a_{il}\big(y_{ij}(t_k) - y_{lj}(t_k)\big) \\ &\quad + a_{ij}\big(y_{ij}(t_k) - x_j(t_k)\big)\Bigg)d\tau \end{aligned} \tag{15}$$

for $i \in \mathcal{C}$, $j \in \mathcal{N}$, and $t \in (t_k, t_{k+1}]$.

Hence, for the continuous-time players, it can be derived that for $t \in (t, t_{k+1}]$

$$\begin{aligned} y_{ij}(t) &= y_{ij}(t_k) - \Bigg(\sum_{l=1}^{N} a_{il}\big(y_{ij}(t_k) - y_{lj}(t_k)\big) \\ &\quad + a_{ij}\big(y_{ij}(t_k) - x_j(t_k)\big)\Bigg)(t - t_k) \\ x_i(t) &= x_i(t_k) - \delta\bar{\alpha}_i \frac{\partial f_i}{\partial x_i}(x_i(t_k), \mathbf{y}_{-i}(t_k)))(t - t_k). \end{aligned} \tag{16}$$

Therefore, for $t = t_{k+1}$

$$\begin{aligned} y_{ij}(t_{k+1}) &= y_{ij}(t_k) - h\Bigg(\sum_{l=1}^{N} a_{il}\big(y_{ij}(t_k) - y_{lj}(t_k)\big) \\ &\quad + a_{ij}\big(y_{ij}(t_k) - x_j(t_k)\big)\Bigg) \\ x_i(t_{k+1}) &= x_i(t_k) - h\bar{\alpha}_i\delta \frac{\partial f_i}{\partial x_i}(x_i(t_k), \mathbf{y}_{-i}(t_k))), \quad i \in \mathcal{C}. \end{aligned} \tag{17}$$

Let $h\bar{\alpha}_i = \alpha_i$ $\forall i \in \mathcal{C}$. Then, it can be derived that

$$\begin{aligned} y_{ij}(t_{k+1}) &= y_{ij}(t_k) - h\Bigg(\sum_{l=1}^{N} a_{il}\big(y_{ij}(t_k) - y_{lj}(t_k)\big) \\ &\quad + a_{ij}\big(y_{ij}(t_k) - x_j(t_k)\big)\Bigg) \\ x_i(t_{k+1}) &= x_i(t_k) - \alpha_i\delta \frac{\partial f_i}{\partial x_i}(x_i(t_k), \mathbf{y}_{-i}(t_k))), \quad i \in \mathcal{N}. \end{aligned} \tag{18}$$

Define $\mathbf{y}(t_k) = [y_{11}(t_k), y_{12}(t_k), \dots, y_{1N}(t_k), y_{2N}(t_k), \dots, y_{NN}(t_k)]^T$ and $\bar{\mathbf{y}}(t_k) = \mathbf{y}(t_k) - \mathbf{1}_N \otimes \mathbf{x}(t_k)$. Then

$$\begin{aligned}\bar{\mathbf{y}}(t_{k+1}) &= \mathbf{y}(t_{k+1}) - \mathbf{1}_N \otimes \mathbf{x}(t_{k+1}) \\ &= \mathbf{y}(t_k) - h(\mathcal{L} \otimes I_{N\times N} + A_0)\bar{\mathbf{y}}(t_k) - \mathbf{1}_N \otimes \mathbf{x}(t_{k+1}) \\ &= \mathbf{y}(t_k) - (\mathbf{1}_N \otimes (\mathbf{x}(t_k) - \delta\alpha\mathcal{P}(\eta(t_k)))) \\ &\quad - h(\mathcal{L} \otimes I_{N\times N} + A_0)\bar{\mathbf{y}}(t_k) \\ &= \big(I_{N^2\times N^2} - h(\mathcal{L} \times I_{N\times N} + A_0)\big)\bar{\mathbf{y}}(t_k) \\ &\quad + \delta\mathbf{1}_N \otimes (\alpha\mathcal{P}(\eta(t_k))) \\ &= \mathcal{H}\bar{\mathbf{y}}(t_k) + \delta\mathbf{1}_N \otimes (\alpha\mathcal{P}(\eta(t_k)))\end{aligned} \tag{19}$$

where

$$\mathcal{P}(\eta(t_k)) = \left[\frac{\partial f_1}{\partial x_1}(x_1(t_k), \mathbf{y}_{-1}(t_k)), \frac{\partial f_2}{\partial x_2}(x_2(t_k), \mathbf{y}_{-2}(t_k)), \dots \frac{\partial f_N}{\partial x_N}(x_N(t_k), \mathbf{y}_{-N}(t_k))\right]^T.$$

By Lemma 1, one can obtain that there are symmetric positive-definite matrices $\tilde{P}$ and $\tilde{Q}$ such that

$$\mathcal{H}\tilde{P}\mathcal{H} - \tilde{P} = -\tilde{Q}. \tag{20}$$

Motivated by [13], define

$$\begin{aligned}V(\xi(t_k)) &= \frac{c}{2}\big(\mathbf{x}(t_k) - \mathbf{x}^*\big)^T \alpha^{-1}\big(\mathbf{x}(t_k) - \mathbf{x}^*\big) \\ &\quad + (1-c)\bar{\mathbf{y}}(t_k)^T\tilde{P}\bar{\mathbf{y}}(t_k)\end{aligned} \tag{21}$$

where $c \in (0,1)$ is a constant and $\xi(t_k) = [(\mathbf{x}(t_k) - \mathbf{x}^*)^T, \bar{\mathbf{y}}(t_k)^T]^T$ as the Lyapunov candidate function.

Then

$$\begin{aligned}\Delta V(\xi(t_k)) &= V(\xi(t_{k+1})) - V(\xi(t_k)) \\ &= \frac{c}{2}\big(\mathbf{x}(t_{k+1}) - \mathbf{x}^*\big)^T\alpha^{-1}\big(\mathbf{x}(t_{k+1}) - \mathbf{x}^*\big) \\ &\quad - \frac{c}{2}\big(\mathbf{x}(t_k) - \mathbf{x}^*\big)^T\alpha^{-1}\big(\mathbf{x}(t_k) - \mathbf{x}^*\big) \\ &\quad + (1-c)\bar{\mathbf{y}}(t_{k+1})^T\tilde{P}\bar{\mathbf{y}}(t_{k+1}) \\ &\quad - (1-c)\bar{\mathbf{y}}(t_k)^T\tilde{P}\bar{\mathbf{y}}(t_k).\end{aligned} \tag{22}$$

By Assumption 2, we have $\mathcal{P}(\mathbf{x}^*) = \mathbf{0}_N$ and hence

$$\big(\mathbf{x}(t_k) - \mathbf{x}^*\big)^T\mathcal{P}(\mathbf{x}(t_k)) \geq L\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|^2. \tag{23}$$

Therefore

$$\begin{aligned}&\frac{1}{2}\big(\mathbf{x}(t_{k+1}) - \mathbf{x}^*\big)^T\alpha^{-1}\big(\mathbf{x}(t_{k+1}) - \mathbf{x}^*\big) \\ &\quad - \frac{1}{2}\big(\mathbf{x}(t_k) - \mathbf{x}^*\big)^T\alpha^{-1}\big(\mathbf{x}(t_k) - \mathbf{x}^*\big) \\ &= \frac{1}{2}\big(\mathbf{x}(t_k) - \mathbf{x}^* - \delta\alpha\mathcal{P}(\eta(t_k))\big)^T\alpha^{-1} \\ &\quad \times \big(\mathbf{x}(t_k) - \mathbf{x}^* - \delta\alpha\mathcal{P}(\eta(t_k))\big) \\ &\quad - \frac{1}{2}\big(\mathbf{x}(t_k) - \mathbf{x}^*\big)^T\alpha^{-1}\big(\mathbf{x}(t_k) - \mathbf{x}^*\big) \\ &\leq -\delta L\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|^2 + \delta l_1\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|\|\bar{\mathbf{y}}(t_k)\| \\ &\quad + \frac{\delta^2}{2}\big(\mathcal{P}(\eta(t_k)) - \mathcal{P}(\mathbf{x}(t_k)) + \mathcal{P}(\mathbf{x}(t_k)) - \mathcal{P}\big(\mathbf{x}^*\big)\big)^T \\ &\quad \times \alpha\big(\mathcal{P}(\eta(t_k)) - \mathcal{P}(\mathbf{x}(t_k)) + \mathcal{P}(\mathbf{x}(t_k)) - \mathcal{P}\big(\mathbf{x}^*\big)\big) \\ &\leq -\delta L\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|^2 + \delta l_1\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|\|\bar{\mathbf{y}}(t_k)\| \\ &\quad + \frac{1}{2}\delta^2\max_{i\in\mathcal{N}}\{\alpha_i\}\big(l_1\|\bar{\mathbf{y}}(t_k)\| + l_2\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|\big)^2\end{aligned} \tag{24}$$

where $l_1 = \max_{i\in\mathcal{N}}\{\bar{l}_i\}$ and $l_2 = \sqrt{N}\max_{i\in\mathcal{N}}\{\bar{l}_i\}$ by noticing that $\|\mathcal{P}(\eta(t_k)) - \mathcal{P}(\mathbf{x}(t_k))\| \leq \max_{i\in\mathcal{N}}\{\bar{l}_i\}\|\mathbf{y}(t_k) - \mathbf{1}_N \otimes \mathbf{x}(t_k)\|$ and $\|\mathcal{P}(\mathbf{x}(t_k)) - \mathcal{P}(\mathbf{x}^*)\| \leq \sqrt{N}\max_{i\in\mathcal{N}}\{\bar{l}_i\}\|\mathbf{x}(t_k) - \mathbf{x}^*\|$ based on Assumption 1.

Moreover

$$\begin{aligned}&\bar{\mathbf{y}}(t_{k+1})^T\tilde{P}\bar{\mathbf{y}}(t_{k+1}) - \bar{\mathbf{y}}(t_k)^T\tilde{P}\bar{\mathbf{y}}(t_k) \\ &= (\mathcal{H}\bar{\mathbf{y}}(t_k) + \delta\mathbf{1}_N \otimes (\alpha\mathcal{P}(\eta(t_k))))^T\tilde{P}(\mathcal{H}\bar{\mathbf{y}}(t_k) \\ &\quad + \delta\mathbf{1}_N \otimes (\alpha\mathcal{P}(\eta(t_k)))) - \bar{\mathbf{y}}(t_k)^T\tilde{P}\bar{\mathbf{y}}(t_k) \\ &= \bar{\mathbf{y}}(t_k)^T\mathcal{H}\tilde{P}\mathcal{H}\bar{\mathbf{y}}(t_k) - \bar{\mathbf{y}}(t_k)^T\tilde{P}\bar{\mathbf{y}}(t_k) \\ &\quad + 2\delta\bar{\mathbf{y}}(t_k)^T\mathcal{H}\tilde{P}(\mathbf{1}_N \otimes (\alpha\mathcal{P}(\eta(t_k)))) \\ &\quad + (\delta\mathbf{1}_N \otimes (\alpha\mathcal{P}(\eta(t_k))))^T\tilde{P}\delta(\mathbf{1}_N \otimes (\alpha\mathcal{P}(\eta(t_k)))) \\ &= -\bar{\mathbf{y}}(t_k)^T\tilde{Q}\bar{\mathbf{y}}(t_k) + 2\delta\bar{\mathbf{y}}(t_k)^T\mathcal{H}\tilde{P}(\mathbf{1}_N \otimes \alpha\mathcal{P}(\eta(t_k))) \\ &\quad + \delta^2(\mathbf{1}_N \otimes (\alpha\mathcal{P}(\eta(t_k))))^T\tilde{P}(\mathbf{1}_N \otimes (\alpha\mathcal{P}(\eta(t_k)))) \\ &\leq -\lambda_{\min}\big(\tilde{Q}\big)\|\bar{\mathbf{y}}(t_k)\|^2 \\ &\quad + 2\delta\max_{i\in\mathcal{N}}\{\alpha_i\}\|\bar{\mathbf{y}}(t_k)\|\big(l_3\|\bar{\mathbf{y}}(t_k)\| + l_4\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|\big) \\ &\quad + \delta^2\big\|\tilde{P}\big\|\max_{i\in\mathcal{N}}\big\{\alpha_i^2\big\}\big(l_5\|\bar{\mathbf{y}}(t_k)\| + l_6\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|\big)^2\end{aligned} \tag{25}$$

where $l_3 = \|\mathcal{H}\tilde{P}\|\sqrt{N}\max_{i\in\mathcal{N}}\{\bar{l}_i\}$, $l_4 = \|\mathcal{H}\tilde{P}\|N\max_{i\in\mathcal{N}}\{\bar{l}_i\}$, $l_5 = \sqrt{N}\max_{i\in\mathcal{N}}\{\bar{l}_i\}$, and $l_6 = N\max_{i\in\mathcal{N}}\{\bar{l}_i\}$.

Hence

$$\begin{aligned}\Delta V(\xi(t_k)) \leq &-\bigg(\lambda_{\min}\big(\tilde{Q}\big)(1-c) - 2\delta\max_{i\in\mathcal{N}}\{\alpha_i\}(1-c)l_3 \\ &\quad - \delta^2\|\tilde{P}\|\max_{i\in\mathcal{N}}\big\{\alpha_i^2\big\}l_5^2(1-c) \\ &\quad - \frac{1}{2}\delta^2\max_{i\in\mathcal{N}}\{\alpha_i\}cl_1^2\bigg)\|\bar{\mathbf{y}}(t_k)\|^2 \\ &- \bigg(\delta Lc - \delta^2\|\tilde{P}\|\max_{i\in\mathcal{N}}\big\{\alpha_i^2\big\} \\ &\quad \times (1-c)l_6^2 - \frac{\delta^2}{2}\max_{i\in\mathcal{N}}\{\alpha_i\}l_2^2c\bigg)\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|^2 \\ &+ \bigg(2\delta(1-c)l_4\max_{i\in\mathcal{N}}\{\alpha_i\} + 2\delta^2\max_{i\in\mathcal{N}}\big\{\alpha_i^2\big\}\|\tilde{P}\|l_5l_6(1-c) \\ &\quad + \delta l_1c + \delta^2c\max_{i\in\mathcal{N}}\{\alpha_i\}l_1l_2\bigg)\|\bar{\mathbf{y}}(t_k)\|\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|.\end{aligned} \tag{26}$$

In addition, for any positive constant d_1, it can be derived that

$$\|\bar{\mathbf{y}}(t_k)\|\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\| \leq \frac{1}{2}\left(\frac{\|\bar{\mathbf{y}}(t_k)\|^2}{d_1} + d_1\big\|\mathbf{x}(t_k) - \mathbf{x}^*\big\|^2\right). \tag{27}$$

Define

$$\begin{aligned}\beta_1 = &\lambda_{\min}\big(\tilde{Q}\big)(1-c) - 2\delta\max_{i\in\mathcal{N}}\{\alpha_i\}(1-c)l_3 \\ &- \delta^2\|\tilde{P}\|\max_{i\in\mathcal{N}}\big\{\alpha_i^2\big\}l_5^2(1-c) - \frac{1}{2}\delta^2\max_{i\in\mathcal{N}}\{\alpha_i\}cl_1^2 \\ &- \frac{1}{2d_1}\bigg(2\delta(1-c)l_4\max_{i\in\mathcal{N}}\{\alpha_i\} + \delta l_1c + 2\delta^2\|\tilde{P}\| \\ &\quad \times \max_{i\in\mathcal{N}}\big\{\alpha_i^2\big\}l_5l_6(1-c) + \delta^2c\max_{i\in\mathcal{N}}\{\alpha_i\}l_1l_2\bigg)\end{aligned} \tag{28}$$

and

$$\beta_2 = \delta Lc - \delta^2 \|\tilde{P}\| \max_{i\in\mathcal{N}}\left\{\alpha_i^2\right\}(1-c)l_6^2 - \frac{\delta^2}{2}\max_{i\in\mathcal{N}}\{\alpha_i\}l_2^2 c - \frac{d_1}{2}\Bigg(2\delta(1-c)l_4\max_{i\in\mathcal{N}}\{\alpha_i\} + \delta l_1 c + 2\delta^2\|\tilde{P}\| \times \max_{i\in\mathcal{N}}\left\{\alpha_i^2\right\}l_5 l_6(1-c) + \delta^2 c\max_{i\in\mathcal{N}}\{\alpha_i\}l_1 l_2\Bigg). \tag{29}$$

Let $d_1 < (2Lc/[2(1-c)l_4 \max_{i\in\mathcal{N}}\{\alpha_i\} + l_1 c])$. Then, it can be derived that there exists $\delta^* > 0$ such that for each $\delta \in (0, \delta^*)$, $\beta_1 > 0$, and $\beta_2 > 0$. If this is the case

$$\Delta V(\xi(t_k)) \leq -\bar{l}\|\xi(t_k)\|^2 \tag{30}$$

where $\bar{l} = \min\{\beta_1, \beta_2\}$, thus arriving at the conclusion.

Appendix B
Proof of Theorem 3

Define a new vector

$$\varrho = (\mathcal{L}\otimes I_{N\times N} + A_0)\mathbf{y} - A_0\mathbf{1}_N\otimes\mathbf{x} \tag{31}$$

where $A_0 = \mathrm{diag}\{a_{ij}\}$. Note that as the communication graph is undirected and connected, $\mathcal{L}\otimes I_{N\times N} + A_0$ is symmetric positive definite and hence invertible [13].

Then, by (12), it can be derived that

$$\dot{\varrho} = -\gamma\varrho. \tag{32}$$

Recalling that

$$\dot{x}_i = -\delta\alpha_i\frac{\partial f_i}{\partial x_i}(x_i, \mathbf{y}_{-i}) \tag{33}$$

for $i \in \mathcal{N}$, define

$$V = \frac{1}{2}\varrho^T\varrho + \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^*\right)^T\alpha^{-1}\left(\mathbf{x}-\mathbf{x}^*\right) \tag{34}$$

where $\alpha = \mathrm{diag}\{\alpha_i\}$ as the Lyapunov candidate function.

According to Assumption 2, we obtain that $-(\mathbf{x}-\mathbf{x}^*)^T[(\partial f_i/\partial x_i)(\mathbf{x})]_{vec} \leq -L\|\mathbf{x}-\mathbf{x}^*\|^2$. Therefore

$$\begin{aligned}\dot{V} &= \varrho^T(-\gamma\varrho) - \delta\left(\mathbf{x}-\mathbf{x}^*\right)^T\left[\frac{\partial f_i}{\partial x_i}(x_i, \mathbf{y}_{-i})\right]_{vec}\\ &= -\gamma\|\varrho\|^2 - \delta\left(\mathbf{x}-\mathbf{x}^*\right)^T\left[\frac{\partial f_i}{\partial x_i}(\mathbf{x})\right]_{vec}\\ &\quad - \delta\left(\mathbf{x}-\mathbf{x}^*\right)^T\left[\frac{\partial f_i}{\partial x_i}(x_i, \mathbf{y}_{-i}) - \frac{\partial f_i}{\partial x_i}(\mathbf{x})\right]_{vec}\\ &\leq -\gamma\|\varrho\|^2 - \delta L\left\|\mathbf{x}-\mathbf{x}^*\right\|^2 + \delta l_1\left\|\mathbf{x}-\mathbf{x}^*\right\|\|\varrho\|\end{aligned} \tag{35}$$

where $l_1 = \max_{i\in\mathcal{N}}\{\bar{l}_i\}\|(\mathcal{L}\otimes I_{N\times N} + A_0)^{-1}\|$. Note that the last term in the last inequality is obtained by the Lipschitz condition of the gradient vectors and

$$\begin{aligned}\mathbf{y} - \mathbf{1}_N\otimes\mathbf{x} &= (\mathcal{L}\otimes I_{N\times N} + A_0)^{-1}(\rho + A_0\mathbf{1}_N\otimes\mathbf{x}) - \mathbf{1}_N\otimes\mathbf{x}\\ &= (\mathcal{L}\otimes I_{N\times N} + A_0)^{-1}\varrho.\end{aligned} \tag{36}$$

Define $\delta^* = [(4\gamma L)/l_1^2]$. Then, for each $\delta \in (0, \delta^*)$

$$\dot{V} \leq -\lambda_{\min}(\Theta)\left\|\left[\varrho^T, \left(\mathbf{x}-\mathbf{x}^*\right)^T\right]^T\right\|^2 \tag{37}$$

where $\Theta = \begin{bmatrix}\gamma & -\frac{\delta l_1}{2}\\ -\frac{\delta l_1}{2} & \delta L\end{bmatrix}$ and $\lambda_{\min}(\Theta) > 0$. Recalling the definition of the Lyapunov candidate function, we conclude that $\|[\varrho^T(t), (\mathbf{x}(t)-\mathbf{x}^*)^T]^T\| \to 0$ as $t\to\infty$, exponentially. By further recalling the definition of ϱ, the conclusion can be derived.

Appendix C
Proof of Theorem 4

By the second equation in (13), we obtain that

$$\sum_{k=1}^N a_{ik}\left(\dot{y}_{ij} - \dot{y}_{kj}\right) = -\gamma\sum_{k=1}^N a_{ik}\left(y_{ij} - y_{kj}\right) \quad \forall i \neq j. \tag{38}$$

Moreover, $\sum_{k=1}^N a_{ik}(y_{ij} - y_{kj})$, where $i \neq j$, can be written as $\sum_{k=1,k\neq j}^N a_{ik}(y_{ij} - y_{kj}) + a_{ij}(y_{ij} - y_{jj})$, for all $i \neq j$. Note that $y_{jj} = x_j$ by the third equation in (13).

Let $\mathcal{G}_{-i}$ be the subgraph of $\mathcal{G}$ by removing node i and the corresponding edges from $\mathcal{G}$. Denote the Laplacian matrix of $\mathcal{G}_{-i}$ as $\mathcal{L}_{-i}$. Let $\mathcal{L}_s$ be a matrix whose diagonal blocks successively are $\mathcal{L}_{-i}$, for $i \in \mathcal{N}$ (the other blocks are 0). Moreover, let $\bar{\mathbf{y}}_{-i} = [y_{1i}, y_{2i}, \ldots, y_{i-1,i}, y_{i+1,i}, \ldots, y_{Ni}]^T$ and define $\mathbf{y}_s = [\bar{\mathbf{y}}_{-1}^T, \bar{\mathbf{y}}_{-2}^T, \ldots, \bar{\mathbf{y}}_{-N}^T]^T$. Correspondingly, define $\mathcal{A}_s$ as a diagonal matrix with its diagonal blocks successively being $\mathcal{A}_i$ $i \in \mathcal{N}$, in which $\mathcal{A}_i$ is a diagonal matrix whose diagonal entries are $a_{1i}, a_{2i}, \ldots, a_{i-1,i}, a_{i+1,i}, \ldots, a_{Ni}$, successively. Then, the concatenated vector form of $\sum_{k=1}^N a_{ik}(y_{ij} - y_{kj})$, where $i \neq j, i \in \mathcal{N}$ can be written as $(\mathcal{L}_s + \mathcal{A}_s)(\mathbf{y}_s - \mathbf{x}_s)$, where $\mathbf{x}_s = [\mathbf{1}_{N-1}^T\otimes x_1, \mathbf{1}_{N-1}^T\otimes x_2, \ldots, \mathbf{1}_{N-1}^T\otimes x_N]^T$.

To establish the stability of the equilibrium under the closed-loop system, define

$$V = \frac{1}{2}\phi^T\phi + \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^*\right)^T\alpha^{-1}\left(\mathbf{x}-\mathbf{x}^*\right) \tag{39}$$

where $\phi = (\mathcal{L}_s + \mathcal{A}_s)(\mathbf{y}_s - \mathbf{x}_s)$. Then

$$\begin{aligned}\dot{V} &= -\gamma\|\phi\|^2 - \delta\left(\mathbf{x}-\mathbf{x}^*\right)^T\left[\frac{\partial f_i(\mathbf{x})}{\partial x_i}\right]_{vec}\\ &\quad + \delta\left(\mathbf{x}-\mathbf{x}^*\right)^T\left[\frac{\partial f_i(\mathbf{x})}{\partial x_i} - \frac{\partial f_i}{\partial x_i}(x_i, \mathbf{y}_{-i})\right]_{vec}.\end{aligned} \tag{40}$$

By the Lipschitz condition in Assumption 1 and the strong monotonicity condition in Assumption 2, we obtain that there exists a positive constant $l_1 = \max_{i\in\mathcal{N}}\{\bar{l}_i\}$ such that

$$\dot{V} \leq -\gamma\|\phi\|^2 - \delta L\left\|\mathbf{x}-\mathbf{x}^*\right\|^2 + \delta l_1\left\|\mathbf{x}-\mathbf{x}^*\right\|\|\mathbf{y} - \mathbf{1}_N\otimes\mathbf{x}\|. \tag{41}$$

Noticing that from the third equation of (13) as well as the definitions of $\mathbf{y}_s$ and $\mathbf{x}_s$, we obtain that $\|\mathbf{y} - \mathbf{1}_N\otimes\mathbf{x}\| = \|\mathbf{y}_s - \mathbf{x}_s\|$. Therefore

$$\dot{V} \leq -\gamma\|\phi\|^2 - \delta L\left\|\mathbf{x}-\mathbf{x}^*\right\|^2 + l_1\delta\left\|\mathbf{x}-\mathbf{x}^*\right\|\|\mathbf{y}_s - \mathbf{x}_s\|. \tag{42}$$

Recalling the definition of ϕ, we obtain that

$$\mathbf{y}_s - \mathbf{x}_s = (\mathcal{L}_s + \mathcal{A}_s)^{-1}\phi \tag{43}$$

where we have utilized the conclusion that $\mathcal{L}_s + \mathcal{A}_s$ is invertible by Assumption 3 and [32, Lemma 4].

Hence

$$\|\mathbf{y}_s - \mathbf{x}_s\| \leq \left\|(\mathcal{L}_s + \mathcal{A}_s)^{-1}\right\| \|\phi\| \tag{44}$$

and

$$\begin{aligned}\dot{V} \leq &-\gamma\|\phi\|^2 - L\delta\left\|\mathbf{x} - \mathbf{x}^*\right\|^2 \\ &+ l_1\delta\left\|(\mathcal{L}_s + \mathcal{A}_s)^{-1}\right\|\left\|\mathbf{x} - \mathbf{x}^*\right\|\|\phi\|.\end{aligned} \tag{45}$$

Let $\delta^* = ([4\gamma L]/[l_1^2\|(\mathcal{L}_s + \mathcal{A}_s)^{-1}\|^2])$, then, for each $\delta \in (0, \delta^*)$

$$\dot{V} \leq -\lambda_{\min}(B)\|\left[\phi^T, \mathbf{x} - \mathbf{x}^*\right]^T\|^2 \tag{46}$$

where

$$B = \begin{bmatrix} \gamma & -\frac{\delta l_1\|(\mathcal{L}_s+\mathcal{A}_s)^{-1}\|}{2} \\ -\frac{\delta l_1\left\|(\mathcal{L}_s+\mathcal{A}_s)^{-1}\right\|}{2} & \delta L \end{bmatrix}$$

and $\lambda_{\min}(B) > 0$. Hence, $\|[\phi(t)^T, \mathbf{x}(t) - \mathbf{x}^*]^T\| \to 0$ as $t \to \infty$, exponentially thus arriving at the conclusion.

References

[1] D. Niyato, A. Vasilakos, and Z. Kun, "Resource and revenue sharing with coalition formation of cloud providers: Game theoretic approach," in *Proc. IEEE/ACM Int. Symp. Clust. Cloud Grid Comput.*, 2011, pp. 215–224.

[2] M. Ye and G. Hu, "Game design and analysis for price based demand response: An aggregate game approach," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 720–730, Mar. 2017.

[3] P. Frihauf, M. Krstić, and T. Başar, "Nash equilibrium seeking in non-coopearive games," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1192–1207, May 2012.

[4] T. Alpcan and T. Başar, "Distributed algorithms for Nash equilibria of flow control games," in *Advances in Dynamic Games*. Boston, MA, USA: Birkhäuser, 2005, pp. 473–498.

[5] J. Koshal, A. Nedić, and U. Shanbhag, "Distributed algorithms for aggregative games on graphs," *Oper. Res.*, vol. 64, pp. 680–704, Mar. 2016.

[6] D. Schiro, J. Pang, and U. Shanbhag, "On the solution of affine generalized Nash equilibrium problems with shared constraints by Lemke's method," *Math. Program.*, vol. 142, pp. 1–46, Mar. 2012.

[7] M. S. Stanković, K. H. Johansson, and D. M. Stipanović, "Distributed seeking of Nash equilibria with applications to mobile sensor networks," *IEEE Trans. Autom. Control*, vol. 57, no. 4, pp. 904–919, Apr. 2012.

[8] M. Zhu and E. Frazzoli, "Distributed robust adaptive equilibrium computation for generalized convex games," *Automatica*, vol. 63, pp. 82–91, Jun. 2016.

[9] S. Grammatico, "Dynamic control of agents playing aggregative games with coupling constraints," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4537–4548, Sep. 2017.

[10] M. Ye, "Distributed robust seeking of Nash equilibrium for networked games: An extended-state observer based approach," *IEEE Trans. Cybern.*, early access, May 21, 2020, doi: 10.1109/TCYB.2020.2989755.

[11] K. Lu, G. Jing, and L. Wang, "Distributed algorithms for searching generalized Nash equilibrium of noncooperative games," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2362–2371, Jun. 2019.

[12] J. S. Shamma and G. Arslan, "Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria," *IEEE Trans. Autom. Control*, vol. 50, no. 3, pp. 312–327, Mar. 2005.

[13] M. Ye and G. Hu, "Distributed Nash equilibrium seeking by a consensus based approach," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4811–4818, Sep. 2017.

[14] M. Ye and G. Hu, "Distributed Nash equilibrium seeking in multiagent games under switching communication topologies," *IEEE Trans. Cybern.*, vol. 48, no. 11, pp. 3208–3217, Nov. 2018.

[15] Z. Deng and X. Nian, "Distributed generalized Nash equilibrium seeking algorithm deign for aggregative games over weight-balanced digraphs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 695–706, Mar. 2019.

[16] B. Gharesifard and J. Cortés, "Distributed convergence to Nash equilibria in two-network zero-sum games," *Automatica*, vol. 49, pp. 1683–1692, Apr. 2013.

[17] Y. Lou, Y. Hong, L. Xie, G. Shi, and K. H. Johansson, "Nash equilibrium computation in subnetwork zero-sum games with switching communications," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2920–2935, Oct. 2016.

[18] M. Ye, G. Hu, and F. Lewis, "Nash equilibrium seeking for N-coalition non-cooperative games," *Automatica*, vol. 95, pp. 266–272, May 2018.

[19] M. Ye, G. Hu, F. L. Lewis, and L. Xie, "A unified strategy for solution seeking in graphical N-coalition noncooperative games," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4645–4652, Nov. 2019.

[20] M. Ye, G. Hu, and S. Xu, "An extremum seeking-based approach for Nash equilibrium seeking in N-cluster noncooperative Games," *Automatica*, vol. 114, Apr. 2020, Art. no. 108815.

[21] Y. Cao, W. Ren, and Y. Li, "Distributed discrete-time coordinated tracking with a time-varying reference state and limited communication," *Automatica*, vol. 45, no. 5, pp. 1299–1305, 2009.

[22] R. Goebel, R. Sanfelice, and A. Teel, *Hybrid Dynamical Systems: Modeling, Stability and Robustness*. Princeton, NJ, USA: Princeton Univ. Press, 2012.

[23] J. Ma, M. Ye, Y. Zheng, and Y. Zhu, "Consensus analysis of hybrid multi-agent systems: A game-theoretic approach," *Int. J. Robust Nonlinear Control*, vol. 29, pp. 1840–1853, Jan. 2019.

[24] Y. Zheng, J. Ma, and L. Wang, "Consensus of hybrid multi-agent systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1359–1365, Jan. 2017.

[25] Y. Zheng and L. Wang, "Consensus of switched multiagent systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 3, pp. 314–318, Mar. 2016.

[26] V. Marchenko, "Hybrid discrete-continuous systems with control: II. State space method," *Differential Equ.*, vol. 51, no. 1, pp. 54–68, 2015.

[27] P. Antsaklis, J. Stiver, and M. Lemmon, "Hybrid system modeling and autonomous control systems," *Hybrid Syst.*, vol. 736, pp. 366–392, Jun. 1993.

[28] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Syst. Mag.*, vol. 29, no. 2, pp. 28–93, Apr. 2009.

[29] W. Ren, "Multi-vehicle consensus with a time-varying reference state," *Syst. Control Lett.*, vol. 56, nos. 7–8, pp. 474–483, 2007.

[30] M. Ye, "Nash equilibrium seeking for games in hybrid systems," in *Proc. Int. Conf. Control Autom. Robot. Vis.*, 2018, pp. 140–145.

[31] H. Khailil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[32] W. Ni and D. Cheng, "Leader-following consensus of multi-agent systems under fixed and switching topologies," *Syst. Control Lett.*, vol. 59, nos. 3–4, pp. 209–217, 2010.

[33] C. Sun, M. Ye, and G. Hu, "Distributed time-varying quadratic optimization for multiple agents under undirected graphs," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3687–3694, Jul. 2017.

[34] F. Facchinei and J. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. New York, NY, USA: Springer-Verlag, 2003.

[35] J. Qin, M. Li, Y. Shi, Q. Ma, and W. X. Zheng, "Optimal synchronization control of multiagent systems with input saturation via off-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 85–96, Jan. 2019.

Maojiao Ye (Member, IEEE) received the B.Eng. degree in automation from the University of Electronic Science and Technology of China, Chengdu, China, in 2012, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2016.

She was a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University from 2016 to 2017. She is currently a Professor with the School of Automaton, Nanjing University of Science and Technology, Nanjing, China. Her research interests include distributed control, optimization, and games in networked systems.

Dr. Ye was a recipient of the Guan Zhao-Zhi Award in the 36th Chinese Control Conference 2017 and the Best Paper Award in the 15th IEEE International Conference on Control and Automation 2019.

Le Yin received the B.Eng. and M.Eng. degrees in communication engineering from Sichuan University, Chengdu, China, in 2009 and 2012, respectively, and the Ph.D. degree in control engineering from Nanyang Technological University, Singapore, in 2018.

He is currently a Lecturer with the School of Computer and Information Science, Southwest University, Chongqing, China. His research interests include state estimation, outlier detection, and robust filtering and optimization.

Guanghui Wen (Senior Member, IEEE) received the Ph.D. degree in mechanical systems and control from Peking University, Beijing, China, in 2012.

He is currently a Professor with the Department of Systems Science, School of Mathematics, Southeast University, Nanjing, China. His current research interests include autonomous intelligent systems, complex networked systems, distributed control and optimization, resilient control, and distributed reinforcement learning.

Prof. Wen was a recipient of the Best Student Paper Award in the 6th Chinese Conference on Complex Networks in 2010. He has been named a Highly Cited Researcher by Clarivate Analytics since 2018. He was awarded a National Natural Science Fund for Excellent Young Scholars in 2017. Moreover, he was a recipient of the Australian Research Council Discovery Early Career Researcher Award in 2018 and the Asia–Pacific Neural Network Society Young Researcher Award in 2019. He is a reviewer of *American Mathematical Review* and is an active reviewer for many journals. He currently serves as an Editorial Board Member and an Associate Editor for the IEEE JOURNAL OF EMERGING AND SELECTED TOPICS IN INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: SYSTEMS, and the *Asian Journal of Control*.

Yuanshi Zheng was born in Jiangshan, China. He received the bachelor's and master's degrees from Ningxia University, Yinchuan, China, in 2006 and 2009, respectively, and the Doctoral degree from Xidian University, Xi'an, China, in 2012.

He is currently an Associate Professor with Xidian University. His research interests include coordination of multiagent systems, game theory, and collective intelligence.