Recent Advances in Consensus of Multi-Agent Systems: A Brief Survey

Jiahu Qin, *Member, IEEE*, Qichao Ma, *Student Member, IEEE*, Yang Shi, *Fellow, IEEE*, and Long Wang, *Member, IEEE*

***Abstract*—In this paper, we mainly review the topics in consensus and coordination of multi-agent systems, which have received a tremendous surge of interest and progressed rapidly in the past few years. Focusing on different kinds of constraints on the controller and the self-dynamics of each individual agent, as well as the coordination schemes, we categorize the recent results into the following directions: consensus with constraints, event-based consensus, consensus over signed networks, and consensus of heterogeneous agents. We also review some applications of the very well developed consensus algorithms to the topics such as economic dispatch problem in smart grid and *k*-means clustering algorithms.**

***Index Terms*—Actuator saturation/fault, consensus in multi-agent systems (MASs), event-based control, heterogeneous systems, signed networks.**

I. Introduction

DISTRIBUTED coordination in multi-agent systems (MASs) has received much attention from multidisciplinary researchers in a wide range, including biology, physics, and engineering [4], [51]. This is partly due to its wide applications to attitude alignment of satellites, cooperative control of unmanned aerial vehicles, aggregation behavior analysis of animals, etc. [51], [52]. An important problem in distributed coordination is to develop information flow algorithms or protocols, which specify the information exchange between an agent and its neighbors, such that the group as a whole can reach an agreement regarding a certain quantity of interest. This problem is usually termed *consensus* or *synchronization* problem [16], [52].

Manuscript received May 29, 2016; revised August 25, 2016; accepted October 22, 2016. Date of publication December 7, 2016; date of current version May 10, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61473269 and Grant 61473116, in part by the Fundamental Research Funds for the Central Universities under Grant WK2100100023, and in part by the Youth Innovation Promotion Association of the Chinese Academy of Sciences.
(Corresponding author: Yang Shi.)

J. Qin and Q. Ma are with the Department of Automation, University of Science and Technology of China, Hefei 230027, China (e-mail: jhqin@ustc.edu.cn; mqc0214@mail.ustc.edu.cn).

Y. Shi is with the Department of Mechanical Engineering, University of Victoria, Victoria, BC V8W 2Y2, Canada (e-mail: yshi@uvic.ca).

L. Wang is with the Intelligent Control Laboratory, Center for Systems and Control, College of Engineering, Peking University, Beijing 100871, China (e-mail: longwang@pku.edu.cn).

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

Digital Object Identifier 10.1109/TIE.2016.2636810

The past decade has witnessed the intensive researches on consensus of MASs from various perspectives, such as consensus over switching network topology [23], [52], [59], [60], [84], consensus with delays [10], [59], [61], [78], [91], [100], [104], optimal consensus [32], [50], [79], sampled-data consensus [61], adaptive consensus [17], [119], quantized consensus [28], second-order consensus [62], [63], [118], consensus of generic linear agents [65], [90], [114], and consensus with multiple leaders [25], [57], [64].

Interested readers are referred to survey papers [4], [51] for an excellent review of the progresses made before 2013 in multi-agent coordination problem. However, these survey papers are far from exhaustive review of the literatures especially the recent ones on consensus of MASs. It is observed that there are a large number of works on MASs focusing on practical issues, especially the issues arising from the industrial applications of consensus of MASs. These works involve practical constraints into the design of the controllers, aiming to satisfy the requirement on agent dynamics, communication, etc., in industry or other fields. Examples for the constraints on agents include actuator saturation [86], existence of various faults [14], [27], and differences in system dynamics [99]; for the constraints on interagent communication/interaction include the limited communication ability due to the lack of energy resources [15], and coexistence of cooperative and competitive relationship between agents [1], [80]. To tackle these constraints, various control schemes developed for single systems are employed to design controllers to realize consensus of MASs. Event-based control scheme, which can save communication cost, is a typical example [15]. Although the results on consensus of MASs are developed in the field of systems and control, some other fields also seek potential applications for the well-developed consensus algorithms. There have been several interesting results in the application of consensus algorithms, for instance, the application to economic dispatch problem (EDP) [116] in power systems and clustering problem [20] in data mining.

The above observation motivates our overview of recent progress made in consensus of MASs. This survey is primarily written from a control perspective, aiming to provide the interested researchers a comprehensive overview of the very recent progresses in the consensus of MASs that are not covered or elaborated in the previous review papers but receive a tremendous surge of interest attention in the past few years. More specifically, this review contains mainly the following topics:

1) consensus subject to various constraints such as actuator saturation and faults, which may happen in the interaction topology or an individual agent;
2) event-based consensus, where the update of the controller is event triggered rather than time scheduled;
3) consensus over signed networks, where agents may be either cooperative or competitive. Both bipartite consensus and group/cluster consensus problems are reviewed;
4) consensus of heterogeneous agents, where agents may take different system dynamics for which usually output consensus rather than state consensus is required; and
5) application of the well-developed consensus algorithms to EDP in power systems and k-means clustering algorithm in data mining.

Though the aforementioned topics will be reviewed independently, they actually have overlaps to some extent. For example, when one looks at the controller design, it is reasonable to taking into account both the actuator saturation and resource-limited digital microprocessor in each agent. Thus, investigation of the problem of event-based consensus of MASs subject to input saturation should be of practical interests. However, due to space limitation, some overlaps of practical interests are not elaborated in this paper.

II. Consensus With Constraints

A. Consensus in MASs With Actuator Saturation

In real control systems, all control actuation devices are subject to amplitude saturation. Force, voltage, flow rate, and every conceivable physical input in every conceivable application of control technology are ultimately limited [85]. Saturation constraints have also been taken into consideration for consensus of MASs recently.

Here, we consider MASs with a group of N agents in $\mathbb{R}^n$, and each agent has the following system dynamics:

$$\dot{x}_i = Ax_i + B\delta_\Delta(u_i), \; i = 1, \ldots, N \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $u_i \in \mathbb{R}^m$ is the control input acting on agent i, and for a positive scalar Δ, $\delta_\Delta : \mathbb{R} \to \mathbb{R}$ is a scalar valued saturation function defined as $\delta_\Delta(u) = \mathrm{sgn}(u)\min\{|u|, \Delta\}$. We also abuse the notation by using δ_Δ to denote a vector valued saturation as $\delta_\Delta(u_i) = [\delta_\Delta(u_{i1}), \ldots, \delta_\Delta(u_{im})]^{\mathrm{T}}$. The matrix pair (A, B) is assumed to be asymptotically null controllable with bounded controls (ANCBC), i.e., the system is stabilizable in the usual linear systems theory sense and all its open-loop poles are located in the closed left-half plane. Note that single integrator, double integrator, and neutrally stable linear agents are all the special case of the ANCBC linear agents.

1) Semiglobal Consensus Control Problem: For the MASs (1), the semiglobal consensus control problems mean that for any *a priori* given bounded set $\mathscr{X} \subset \mathbb{R}^n$, to conduct a control law $u_i(t)$ for each agent i, such that $\lim\limits_{t\to\infty} \|x_i(t) - x_j(t)\| = 0$ for any $i, j = 1, \ldots, N$ as long as $x_i(0) \in \mathscr{X}$, $i = 1, \ldots, N$.

The low gain feedback technique widely used in the control problem of individual linear systems subject to actuator saturation [30], [124] is the main effective method in achieving semiglobal consensus of agents represented by ANCBC linear systems [86]. By using the low gain feedback technique, the semiglobal consensus problems for ANCBC linear MASs with saturation constraints are considered under different frameworks including fixed [66], [92] and switching communication topologies [86], [87]. The main idea behind semiglobal stabilization by low gain feedback is to tune the feedback gain low enough according to the given bounded set, such that the system operates in the linear region of its input as long as all the initial states are inside the given set. The low gain feedback design is usually carried out in the steps as shown in Algorithm 1.

Algorithm 1: Low gain feedback controller design.

Step 1: Solve the following parametric algebraic Riccati equation

$$P(\varepsilon)A + A^T P(\varepsilon) - \gamma P(\varepsilon)BB^T P(\varepsilon) + \varepsilon I = 0$$

or the parametric Lyapunov equation

$$P(\varepsilon)A + A^T P(\varepsilon) - \gamma P(\varepsilon)BB^T P(\varepsilon) + \varepsilon P(\varepsilon) = 0,$$

where γ is a positive constant, which may influence the convergence rate of the multi-agent system, and ε is a positive constant determined by the given bounded set.

Step 2: Design the control law for agent i as $u_i(t) = K\sum_{j=1}^N a_{ij}(x_j(t) - x_i(t))$, where $K = B^T P(\varepsilon)$.

2) Global Consensus Control Problem: For MASs (1), the global consensus control problems mean that for any initial states, to conduct a control law $u_i(t)$ for each agent i, such that $\lim\limits_{t\to\infty} \|x_i(t) - x_j(t)\| = 0$ for any $i, j = 1, \ldots, N$.

The well-established control laws used in the MASs without actuator saturation are widely applied to deal with the global consensus control problems [31], [45], [115], among which, [31] is the first work in showing that the consensus laws designed with neglecting the saturation constraints can still work well in the presence of saturation for the global leaderless consensus problems in single-integrator MASs over directed communication topology. Besides, the global consensus problems are also investigated for double-integrator MASs and neutrally stable linear MASs, respectively, over a strongly connected and detailed balanced communication topology in [45] by using the control laws established without considering the actuator saturation.

All the above-reviewed works design the linear feedback controller to deal with the global consensus control problems. In fact, some nonlinear algorithms can be also applied with the aid of some saturation functions. In [73], the consensus algorithm for double-integrator dynamics under an undirected interaction graph is proposed by utilizing the saturation function $\tanh(\cdot)$. Two consensus algorithms are proposed in [33] for agents with double-integrator dynamics to deal with global consensus problems by applying a strictly increasing continuously differentiable function. The global leader-following consensus is investigated by using a multihop relay protocol with the help of saturation function $\delta_{\frac{\Delta}{2}}(\cdot)$ in [125] for ANCBC linear MASs with a strongly connected and detailed balanced communication topology.

B. Consensus in MASs With Fault

Malfunctions of systems, brought about by wear or damage, in actuators, sensors, or other system components are inevitably for a single system [126]. For MASs, the probability of agents facing faults is growing due to the increasing scale and complexity of the systems [14]. Moreover, the MASs can also suffer from topology faults [27]. As communication topology of MASs plays an important role in the consensus problem, topology faults may influence the overall performance or even stability of the MASs. Similarly, a fault occurring in a single agent can also affect the entire system performance. Therefore, when faults occur in the MASs, it is necessary to find out the faults and take measures to ensure system performance, which correspond, respectively, to the fault detection and isolation (FDI) problem and the fault-tolerant control (FTC) problem. During the past decades, numerous results on FDI and FTC have been achieved [58], [126]. However, most of the results in the literature are for single agent system and could not be used directly for MASs. In recent years, FDI and FTC for MASs have been attracting more attention.

Based on the architecture, FDI in MASs can be divided into centralized FDI and distributed FDI. In the centralized architecture, there is a central agent, which installs the fault diagnosis algorithms and gathers all the necessary system information to perform fault diagnosis. Different from the centralized case, in the distributed architecture, all agents are equipped with fault diagnosis algorithms, which only need local information and receive messages from its neighboring agents. Generally, distributed FDI has more advantages than the centralized FDI for that the required computational load and communication bandwidth are much less for distributed FDI [48]. Due to the advantages of distributed FDI methods, most of the FDI algorithms proposed for MASs in the literature are distributed FDI to deal with different types of faults, such as actuator faults [48], [49], sensor faults [14], [127], plant faults [2], and topology faults [89]. A common method used in these works is to use the residual signal to detect and isolate the faulty agents or faulty edges. For instance, in [89] the residual signal r_j^i for agent i to detect agent j is defined as $r_j^i = y_i(t) - C_i\hat{x}_j^i$, where $\hat{x}_j^i$ denotes the state estimate decoupled from a faulty agent j and calculated by agent i. Then, by designing the threshold Θ_j^i for r_j^i, one can determine whether agent j is faulty. Similar to FDI, a more challenging work is fault estimation (also called fault detection and diagnosis in the literature), which not only detects and isolates a fault but also gives an estimation of the magnitude and shape of the fault [47].

The objective of FTC is to design control systems capable of tolerating potential faults such that the reliability and availability of systems is improved while maintaining a desirable performance. Similar to the case in FDI, distributed FTC methods for MASs are more promising than the centralized methods due to the advantages of communication savings, robustness, scalability, etc. Till now, most of the faults considered in the literature for MASs are actuator faults [9], [81], [82], [93], [94], [121], which are generally modeled as

$$u_{oi}(t) = \rho_i(t)u_{ih}(t) + r_i(t), \quad t \geq t_{if} \tag{2}$$

where $u_{ih}(t)$ represents the healthy control effort of the ith agent, $u_{oi}(t)$ is the actual output of the actuator, $\rho_i(t) \in (0, 1]$ is the "healthy indicator," which implies that the actuator of the ith agent faces partial loss of effectiveness (also termed multiplicative fault), $r_i(t)$ is the bias fault (also termed additive fault), and t_{if} represents the time fault occurring.

According to the design method, FTC for MASs can be broadly classified into two categories: passive fault-tolerant control (PFTC) and active fault-tolerant control (AFTC). In PFTC, the parameters of controller are fixed and the controller is designed to be robust against a class of presumed faults [93], [94]. In the PFTC, it should be mentioned that there is no requirement on online faults information. Different from PFTC approach, AFTC involves automatically detecting and identifying faulty components and then reacting to faults actively by reconfiguring control actions. The AFTC method with adaptive controller where the feedback gains of the controller are updated online by estimating the value of actuator faults is investigated in [9] and [121]. In [82], an actuator fault accommodation strategy was proposed by adjusting weights of communication topology. It is worth noting that FTC for MASs with other types of faults has also been investigated in the literature, such as sensor faults, plant faults, and topology faults, which can refer to [13], [26],[56], and the references therein.

III. Event-Based Consensus Control

In time-scheduled control, data transmission and controller update are implemented periodically, independently from the state of the system. Considering that the constant sampling and transmission period have to guarantee the stability in the worst case scenario, time-scheduled control policy is conservative in the number of control updates, which constrains its applicability to a broader class of applications of practical interests. For instance, autonomous agents are often equipped with resource-limited digital microprocessors to communicate with other agents through the network with limited bandwidth. There may occur a period when the microprocessor and network can tolerate may not be able to guarantee the stability. Furthermore, even if the period is acceptable, it is clearly a waste of the network and computation resources especially when there is very little fluctuation between two successive sampled data. To overcome the drawbacks caused by the conservativeness of time-scheduled control, event-based control is proposed for digital platforms [88]. In event-based control, the time instants for data transmission and control update are determined by certain events that are triggered depending on predefined rules. For example, when the magnitude of some measurement error reaches a prescribed threshold, an event can be triggered.

Starting from the pioneering work in [15], which introduces an event-triggered control mechanism [88] to MASs, much great progress has been made from various perspectives, examples include single-integrator consensus [18], [83], [120], double-integrator consensus [6], [40], [83], consensus of generic linear MASs [46], [105], [122], consensus of nonlinear MASs [36],

H_∞ consensus [41], and consensus with communication delay [46], [83], [103], [122], just to name a few.

A. General Event-Triggered Control Problem

Consider a group of N agents in $\mathbb{R}^n$, where each agent is modeled by the following general nonlinear autonomous system:

$$\dot{x}_i = f(x_i, u_i),\ i = 1, \ldots, N \tag{3}$$

where $x_i \in \mathbb{R}^n$ is the state of agent i, and $u_i \in \mathbb{R}^m$ is the control input acting on agent i.

Under this setup, the event-triggered control problems can be stated as follows: design a control law $u_i(t)$ for each agent i using only its own and its neighbors' states, where the states are transmitted according to a certain triggering condition based on current measurements, such that for any initial values, the states of the agents satisfy

$$\lim_{t\to\infty} \|x_i(t) - x_j(t)\| \le \boldsymbol{\delta}, i, j = 1, \ldots, N$$

where $\boldsymbol{\delta} \ge 0$ and meanwhile, there is no *Zeno behavior*, i.e., there is a lower bound, which is positive, on the minimum interevent time. Note that the case that $\boldsymbol{\delta} > 0$ is known as *bounded consensus* or *practical consensus*, whereas $\boldsymbol{\delta} = 0$ corresponds to *consensus*.

Although different kinds of event-triggered control algorithms are proposed, from the aspect of the detection of the events, the current works mainly consider the following three frameworks.

1) An event is detected through continuously monitoring its own and its neighbors' states [21], [36], [40].
2) An event is detected through continuously monitoring its own state and discontinuously monitoring its neighbors' states [46], [83], [120].
3) An event is detected through discontinuously monitoring its own and its neighbors' states [36].

The event-triggered control designed in framework 3) is also termed as self-triggered control. In the following, we will first summarize some event-triggered control mechanisms in the frameworks of 1) and 2), and then briefly introduce the mechanisms of self-triggered control 3). At last, a special event-based control scheme termed *edge event-driven* control, which differs from the above three control schemes in that it is based on edges rather than agents, is introduced.

B. Event-Triggered Control Mechanisms

In event-triggered control mechanisms, an event is triggered when the magnitude of the measurement error reaches the prescribed threshold. Such an event can lead to the transmission of data and the update of controller. Furthermore, the type of the controller and when to transmit the current data are usually relevant to the type of the measurement error. Generally speaking, the measurement error under consideration mainly has three different forms, which are, respectively, trivial form [15], [46], [83], [120], combinational form [18], [36], [122], and model-based form [21]; while the threshold can be simply divided into state independent [83], [120], [122], state dependent [15], [18], [21], [36], [46], [120], and hybrid [46], [122] categories.

1) Measurement Error:

Assume that the triggering time sequences of agent i are $t_0^i = 0, t_1^i, \ldots, t_k^i, \ldots$, then the trivial measurement error [15], [46], [83], [120] can be described as

$$e_i(t) = x_i(t_k^i) - x_i(t), t \in [t_k^i, t_{k+1}^i) \tag{4}$$

and the corresponding controller is

$$u_i(t) = K \sum_{j=1}^{N} a_{ij} \left[x_j\left(t^j_{k'_j(t)}\right) - x_i(t_k^i) \right], t \in [t_k^i, t_{k+1}^i) \tag{5}$$

where $k'_j(t) = \arg\max_{l\in\mathbb{N}:t\ge t_l^j}\{t - t_l^j\}$ denote the latest sampled time of agent j. When utilizing the above measurement error (4) and controller (5), the triggering of an event usually leads the agent to transmit its current states to its neighbors and the controller will be updated at its own and its neighbors' triggering time. Therefore, the controller update frequency is usually very high. In order to reduce the update frequency of controller, Fan *et al.* [18] propose the following combinational measurement error and its corresponding controller.

Denoting by the combinational measurement [18], [36], [122] $q_i(t) = \sum_{j=1}^{N} a_{ij}(x_j(t) - x_i(t))$, one has the combinational measurement error

$$e_i(t) = q_i(t_k^i) - q_i(t), t \in [t_k^i, t_{k+1}^i) \tag{6}$$

and the controller

$$u_i(t) = K q_i(t_k^i), t \in [t_k^i, t_{k+1}^i). \tag{7}$$

In this event-triggered control scheme, once the event of an agent is triggered, the agent needs to obtain all the states of its neighbors and further, the controller (7) updates according to the obtained states. Therefore, the update frequency of controller can be largely reduced. However, such a control scheme suffers from the drawback that it may not be applicable to directed communication topology. This is because the transmission of data is unidirectional in directed graph, making the neighbors of an agent do not know when to transmit their states [46] or when an event is triggered at one agent, it has to request its neighbors for additional information, thus increasing the communication cost [8].

The aforementioned two measurement errors and the corresponding controllers are mostly used in the single-integrator MASs [15], [18], [83], [120] for which it is mainly the communications among agents that determine the group behavior. However, for generic linear MASs, considering the influence of the self-dynamics of each individual agent, a kind of model-based measurement error [21] is proposed. The model-based measurement error can be modeled as follows:

$$e_i(t) = e^{A(t-t_k^i)} x_i(t_k^i) - x_i(t), \quad t \in [t_k^i, t_{k+1}^i) \tag{8}$$

and

$$u_i(t) = K \sum_{j=1}^{N} a_{ij} \left[e^{A(t - t^j_{k'(t)})} x_j(t^j_{k'(t)}) - e^{A(t-t_k^i)} x_i(t_k^i) \right],$$
$$t \in [t_k^i, t_{k+1}^i).$$

Since for generic linear MASs, the trivial measurement and combinational measurement errors may grow very quickly [21], making the interevent time being very short. The matrix exponential function e^{At} in (8) is used to estimate the current states of the agent, thus the model-based measurement error usually grows slower. Therefore, the interevent time can be increased and the controller updates frequency can also be reduced.

2) Threshold: The state-independent thresholds [83], [120], [122] are some decreasing functions with nonnegative lower bounds, which usually take the form of $c_i + d_i e^{-\alpha_i t}$ with $c_i + d_i > 0$, $c_i \geq 0$, $d_i \geq 0$, and $\alpha_i > 0$. The advantages of the state-independent threshold are twofold. One is that by using the state-independent threshold, it is easy to conclude that the Zeno behavior can be excluded, namely, the interevent time is larger than a positive number, since the evolution rate of the measurement error is limited and the state-independent threshold is lower bounded by a positive number over any time intervals. The other is that no neighbors' states are involved in determining each agent's threshold, therefore, the communication with its neighbors can be avoided. However, it also means that the threshold has no information about the system, which would make the system performance changed, for example, the convergence rate may be governed by the state-independent threshold [18]. Furthermore, when $c_i \neq 0$, bounded convergence instead of asymptotic convergence is achieved. When $c_i = 0$, the triggering of the events is sensitive to the external disturbance. Especially, when the time approaches infinity, the threshold approaches zeros, even a very small disturbance would lead to the triggering of an event.

To date, the state-dependent thresholds do not have a unified form. Nevertheless, the existing works consider mainly two such kinds of state-dependent thresholds as continuous threshold and piecewise constant threshold. For continuous threshold, $\beta_i \| \sum_{j=1}^{N} a_{ij} \left(x_j(t) - x_i(t)\right) \|$ [15], [18] and $\beta_i \sqrt{\| \sum_{j=1}^{N} a_{ij} \left(x_j(t) - x_i(t)\right)^2 \|}$ [36], [120] are chosen, where β_i is a positive number, which relates to the information of the system dynamics as well as the network topology. Obviously, continuous monitoring of the neighbors' states is a must. The piecewise constant threshold can be used to avoid the continuous monitoring since the continuous states are replaced by the latest sampled states. When employing the trivial measurement and model-based measurement errors, since each agent samples its state at its triggering time, one can choose $\beta_i \| \sum_{j=1}^{N} a_{ij} (x_j(t^j_{k'_j(t)}) - x_i(t^i_k)) \|$ [46] or $\beta_i \sqrt{\| \sum_{j=1}^{N} a_{ij} (x_j(t^j_{k'_j(t)}) - x_i(t^i_k))^2 \|}$ [120] as the threshold. While for combinational measurement error, the threshold can be chosen as $\beta_i \| \sum_{j=1}^{N} a_{ij} \left(x_j(t^i_k) - x_i(t^i_k)\right) \|$ and $\beta_i \sqrt{\| \sum_{j=1}^{N} a_{ij} (x_j(t^i_k) - x_i(t^i_k))^2 \|}$.

The state-dependent threshold has at least two drawbacks worthy of notation. First, it is sensitive to external disturbances, and second, with state-dependent threshold, it is usually very hard to exclude the Zeno behavior. To address the issue of Zeno behavior, some techniques have been proposed. For example, one can choose $t^i_{k+1} = t^i_k + \max\{\tau^i_k, \tau_i\}$ as the next triggering time, where τ^i_k is the time interval determined by the abovementioned methods and τ_i is a constant relevant to the systems dynamics and the network topology [19], [105].

Remark 1: The exclusion of Zeno behavior is of utmost importance when proposing an event-triggered control algorithm, however, is not rigorously addressed in some of the existing works, in particular those using state-dependent threshold. For continuous threshold, only one agent's Zeno behavior is excluded in [15]. For piecewise constant threshold [18], [40], the technique used to exclude the Zeno behavior is similar to the state-independent threshold. However, as is also observed in [46], having the fact that $t^i_{k+1} - t^i_k > 0$, $\forall i = 1, \ldots, N$, $k = 1, 2, \ldots$ cannot guarantee the exclusion of Zeno behavior. Consider, for example, the special case that $t^i_{k+1} - t^i_k \geq \frac{1}{k}$, one can easily obtain that the interevent time approaches zero as k approaches infinity.

Hybrid thresholds [46], [122] are the sum of the state-dependent and state-independent thresholds. This kind of threshold can combine the advantages of the state-dependent and the state-independent thresholds, such as the utilization of the system information and the exclusion of the Zeno behavior, though the parameters design and the convergence analysis are technically more challenging.

Note that the above event-triggered control mechanisms are proposed under continuous-time framework, where the states can be detected and the controller can be updated at any time instants only if there are no Zeno behaviors. There are also some other algorithms proposed under discrete-time framework, for example, the periodic-triggered algorithms [42]. In the periodic-triggered algorithms, the time interval between the operations including data transmission, event detection, and controller updates must be the integral multiple of a positive number. Here, we take the work in [42] as an example to illustrate the periodic-triggered algorithms. The triggering time in [42] can be formulated as

$$t^i_{k+1} = t^i_k + h \inf\left\{l : \left\|e_i(t^i_k + lh)\right\| > \beta_i \left\|z_i(t^i_k + lh)\right\|\right\} \tag{9}$$

where $e_i(t^i_k + lh) = x_i(t^i_k + lh) - x_i(t^i_k)$ and $z_i(t^i_k + lh) = \sum_{j=1}^{N} a_{ij} \left[x_j(t^i_k + lh) - x_i(t^i_k + lh)\right]$. Although periodic-triggered algorithms are proposed in the beginning to avoid the continuous monitoring and detection problems, the Zeno behaviors can also be avoided naturally. Of course, there are also some problems to be dealt with when applying the periodic-triggered algorithms. One is how to sample and transmit the states of the agents in a synchronous manner. The other is the design of the parameter h. If the h designed to ensure the convergence is less than the period of the time-triggered schemes, the data transmission frequency is even higher than that in time-triggered schemes.

C. Self-Triggered Control Mechanisms

In self-triggered mechanisms, the next event time is generally precomputed at a control update time based on predictions using previously received data and knowledge on the plant dynamics. Since the precomputation operations are implemented in a discrete-time manner, continuous detection is thus avoided.

The self-triggered control mechanisms [18], [19], [36] are usually designed based on the event-triggered control mechanisms aforementioned in the above section. In the event-triggered control mechanisms, the time instant, say t_0, at which the measurement error reaches the threshold, say $\boldsymbol{\delta}$, is chosen as the triggering time. It is reasonable that any time instant before time instant t_0 can be chosen as the triggering time. With this in mind, if there exists a triggering function taking simple form but is not less than the magnitude of the measurement error, and a variable which is piecewise constant and not larger than the aforementioned threshold $\boldsymbol{\delta}$, then one can choose the time instant at which this triggering function crosses the piecewise constant variable as the triggering time. Note that since the triggering function is in a simple form, which is relatively easy to be manipulated, it is thus easier to compute the triggering time instant. This is the main idea adopted in most of the existing self-triggered control mechanisms. Of course, the conservation of the time instant will result in the higher frequency of data transmission and controller updates, although the self-triggered control mechanisms can avoid the continuous monitoring and detection.

D. Edge Event-Driven Control Mechanism

In the event-triggered mechanism introduced in the preceding sections, the triggering of an event makes one agent to transmit its states or update its controller based on the state information of the agent and its neighbors. Recently, a different event-based control scheme termed edge event-driven control is developed [83], where the triggering of the event makes two agent connected via an undirected edge to update their controllers based only on the states of their own [102]. Assume that there are two agents i and j connected via an undirected edge (i, j). Given a time sequence associated to these two agents $t_0^{ij}, t_1^{ij}, \ldots$ satisfying $t_k^{ij} < t_{k+1}^{ij}$ and $t_k^{ij} = t_k^{ji}$. At these time instants, agent i and j sample the relative state information and update controllers simultaneously if their states satisfy the prescribed rules. Such an event only relevant to the edge (i, j) is called the *edge event* [102] of the edge (i, j). The controller in [102] takes the following form:

$$u_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(t) \left(x_j \left(t^{ij}_{k^{ij}(t)} \right) - x_i \left(t^{ij}_{k^{ij}(t)} \right) \right)$$

where $k^{ij}(t) = \max\{k : t_k^{ij} \leq t\}$. The measurement error and the threshold are given in terms of the states of agent i and j [102]. When utilizing the above edge event-driven control scheme, the continuous monitoring and detection may be required, therefore edge event-driven control scheme is usually combined with the periodic control scheme such that the monitoring and detection are operated periodically.

Following the framework established in [102], consensus problem has been further investigated for undirected networks of integrator agents [5], [7], [102]. Note that only the information of the two agents associated to the edge at the predefined time instants is required during the edge event. This can reduce the communication cost. Moreover, it can be inferred from the edge event-driven control mechanism that the Zeno behavior can be excluded by choosing appropriate time sequences, e.g., restrict the event-detection instants to the periodic times such that $t_{k+1}^{ij} = t_k^{ij} + h$ with $h > 0$ [7]. It is worth mentioning that the current works concerning edge event-driven control mechanisms are all built in the framework of undirected network topologies, thereby confining their capability of dealing with directed network topologies in practical applications.

IV. Consensus Over Signed Networks

In this section, we review the results for agents communicating over signed networks, where agents may be either cooperative or competitive.

A. Bipartite Consensus

In social networks, mutual relationships between two individuals may be either friendly or hostile [77], and usually such relationship leads to two antagonistic groups. This raises the concept of bipartite consensus where the agents are split into two groups such that all the agents converge to a common decision in each group, while the decisions of the two groups are opposite. It is explicitly shown in [3] how a signed network can be constructed according to the like/cooperate, dislike/compete relationship in social network and how opinions evolve over the signed network. Along this line of research, there have been a number of works on bipartite consensus recently, starting from the first work from control perspective in [1].

To achieve bipartite consensus, a control algorithm is given from the gradient system of a Laplacian potential as follows [1]:

$$\dot{x}_i = -\sum_{j=1}^{N} |a_{ij}| (x_i - \operatorname{sgn}(a_{ij}) x_j).$$

It is shown in [1] that a necessary condition for bipartite consensus is that the signed network is *structurally balanced*, which means that the node set $\mathscr{V}$ can be divided into two subset $\mathscr{V}_1$ and $\mathscr{V}_2$ such that $\mathscr{V}_1 \cup \mathscr{V}_2 = \mathscr{V}$, $\mathscr{V}_1 \cap \mathscr{V}_2 = \emptyset$, and $a_{ij} > 0, i, j \in \mathscr{V}_p, p = 1, 2$; $a_{ij} \leq 0$ otherwise. Note that the sign associated with the weight a_{ij} indicates the relationship between two agents i and j, that is "$+$" implies friendship, whereas "$-$" implies hostility. Under this structurally balanced condition, there exists a *gauge transformation* [1], which can transform the bipartite consensus problem equivalently into the standard consensus problem for which the well-established techniques/methods for cooperative networks of agents are applicable. Then, the bipartite consensus result has been well established for fixed network topology [43], [66], [123] and time-varying network topology [44], [54]. Very recently, Qin *et al.*[66] extend bipartite consensus to the network of linear systems with actuator saturation by proposing a novel low gain feedback design method.

In networks with cooperative and competitive interactions, there is another framework with more than two interacting clusters of agents, where agents within the same clusters are cooperative, while they may be either cooperative or competitive if they belong to different clusters. This falls into the topic of group/cluster consensus, as shown in the following section.

B. Group/Cluster Consensus

Generally speaking, *group consensus* refers to the scenario that a network of agents are divided into subgroups such that agents in the same subgroup reach consensus [107], [108]. If further for any initial states of the agents, there is no consensus between any two different clusters, such a scenario is termed *cluster consensus* [70], [95].

For illustration, assume, without loss of generality, that there are q ($q \geq 2$) clusters of agents, say $\{\mathscr{V}_1, \ldots, \mathscr{V}_q\}$, in a MAS containing N agents. Let $\bar{i}$ denote the subscript of the cluster to which node i belongs. Now, consider the case that the agents are governed by the following system dynamics:

$$\dot{x}_i(t) = f(x_i(t), t) + \sum_{j=1}^{N} a_{ij} H\left(x_j(t) - x_i(t)\right), \; i = 1, \ldots, N \tag{10}$$

where the coupling function $H : \mathbb{R}^n \to \mathbb{R}^n$ is continuous. A prerequisite requirement for investigating group consensus problem is that the corresponding group consensus manifold $\mathscr{S}(n) = \left\{[x_1^{\mathrm{T}}, \ldots, x_N^{\mathrm{T}}]^{\mathrm{T}} : x_i = x_j, \forall \bar{i} = \bar{j}\right\}$ should be invariant through (10). To this end, the common intercluster condition is imposed on network topology [22], [35], which guarantees the invariance of $\mathscr{S}(n)$ and can be described as follows.

Assumption 1 (common intercluster condition):

$$\sum_{j \in \mathscr{V}_\ell} a_{ij} = s_{k\ell} \quad \forall i \in \mathscr{V}_k, \; k, \ell = 1, \ldots, q, k \neq \ell$$

where $s_{k\ell}$ is a constant relating only to the indices of clusters $\mathscr{V}_k$ and $\mathscr{V}_\ell$. A simplified yet interesting assumption termed *in-degree balanced condition*, which requires that $s_{k\ell} \equiv 0$, is also widely used in existing literature [95], [106], [108].

1) Integrator Agent Case:

To achieve and maintain group/cluster consensus, the widely used consensus protocol $\dot{x}_i = -\sum_{j=1}^{N} a_{ij}(x_j - x_i)$ can be well applied [107]. Inspired by this, group consensus problem for single-integrator MASs with or without time delay, communication noise and switching network topology are extensively investigated in [106]–[108]. The authors therein build in-depth relations between the network behavior and the spectral property of the graph Laplacian. Bearing in mind that the agents within the same clusters are cooperative, while they may be either cooperative or competitive if they belong to different clusters, one may then expect to enhance the couplings within each cluster, which are positively weighted, to dominate over the competition caused by the intercluster couplings, which can be negatively weighted, such that group consensus can be achieved. Motivated by this observation, the model in [107] is revisited in [67] and [68] under directed network topology via a novel Lyapunov method based analysis, resulting in intuitive structural conditions, viz., ones involving the coupling topology as well as the coupling strength, that guarantee the group consensus. Very recently, this framework is extended to double-integrator case under two different frameworks, viz., the framework that all agents share the same position and velocity interaction topology, and the framework that the position and velocity topologies are modeled by totally independent graphs. Both, the case without leaders and the leader-following case are systematically investigated. Different systems models are analyzed accordingly using various different techniques [69]. In order to finally maintain cluster consensus, Han *et al.* [22] introduce nonidentical inputs to the agents in different clusters of single-integrator MASs.

2) Generic Linear Agent Case:

The network of identical generic linear systems can be described by

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \;\; i = 1, \ldots, N \tag{11}$$

where (A, B) is stabilizable. Qin and Yu [70] exploit pinning control technique to achieve cluster consensus under the assumption that the network topology is collectively acyclic. This result is then extended to a general network topology [109]. Later in [110], the authors propose a unified Lyapunov approach to analyze the collective behavior with respect to network topology and coupling configuration for leaderless case, which includes the leader-following one studied in [70] and [109] as a special case. The more general result that deals with the heterogeneous case where agents from different clusters are nonidentical but with the same dimension is reported in [72]. The aforementioned works follow the research line in [67]–[69]. A consistent and intuitive result obtained in [70], [109], and [110] for generic linear systems and in [67]–[69] for integrator agents indicates that as long as the underlying graph of each cluster contains a directed spanning tree, and the couplings among the agents within each cluster are strong enough, group consensus can be guaranteed.

Remark 2: When performing the convergence analysis, the influences of self-dynamics and the competitive links, which are negatively weighted, make the nonnegative matrix theory, which was widely employed in dealing with the convergence of cooperative agents, especially those with integrator dynamics, being inapplicable. In most works, Lyapunov method based analysis is a preference [65], [70], [95], [111], [119]. In particular, Qin *et al.*[65] provide a unified analysis to deal with the consensus of linear and nonlinear agents, though the nonlinear function f, which describes the evolution of a single agent, should satisfy the global Lipschitz-like conditions. Furthermore, the proposed technique is applicable to the cluster/group consensus once one is clear about the relation between the location of the eigenvalues of the Laplacian matrix and the strength and structure of the coupling topology. This is what has been systematically investigated in [68], [70], [109], and [110], where the agents under consideration take various different system dynamics, including generic linear system dynamics and nonlinear system dynamics [34], [95]. However, these Lyapunov-based methods are invalid for more general case, e.g., the case that f is only locally Lipschitz, for which one can consider using nonlinear contraction theory [37].

Intuitively, a system $\dot{x} = f(x, t)$ is contracting if the initial conditions are somehow "forgotten," i.e., if the final behavior of the system is independent of the initial conditions [37]. This concept is later extended to *partial contraction* such that contraction theory can be applied to more general nonlinear systems. The key in contraction analysis is to prove the uniform negative definiteness of $F = (\dot{\Theta} + \Theta\frac{\partial f}{\partial x})\Theta^{-1}$, where $\Theta(x, t)$ is chosen such that $\Theta^{\mathrm{T}}(x, t)\Theta(x, t)$ is uniformly positive definite.

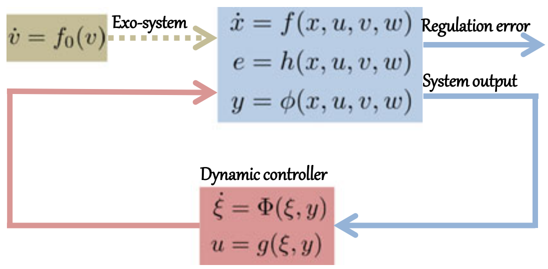

Fig. 1. General output regulation framework. The dashed line from the exosystem indicates that the exosystem may be a virtual system produced by the dynamic controller.

Recently, application of contraction theory to investigate the synchronization of network of nonlinear systems has been discussed in [55], [74], and [96]. Generally speaking, if one proves that the synchronization manifold remains invariant for a network of general nonlinear systems, and furthermore the system dynamics of the network of the nonlinear systems is globally/locally contracting with respect to the manifold, then synchronization can be guaranteed globally/locally. Following this principle, concurrent synchronization and group cooperation are further discussed [55], [96].

It is worth pointing out that contraction theory, which is developed for general nonlinear system, provides us with a new analysis tool from a viewpoint different from Lyapunov stability theory. However, in the application of contraction theory, it is usually a rather challenging task to find the metric matrix $\Theta(x,t)$ and prove that F is uniformly negative.

V. Consensus of Heterogeneous Agents

Different from the aforementioned works concerning homogeneous agents [72], [97] where the states of the agents have exactly the same dimension, in this section, we will review the more general framework where the states of agents may have different dimension. In such a framework, state agreement is usually impossible and output synchronization is the focus. One of the first works dealing with consensus in heterogeneous linear MASs is [98]. This work is fueled by the consideration that the evolution of agents in the network may be described by different system dynamics, due to possible friction or damping coefficients in dynamical systems, or changing masses [98]. To make the consensus problem feasible, from the viewpoint of internal model principle, a necessary condition is provided [98], which is proved to be a sufficient condition [99] as well together with a dynamic controller.

A general framework for output regulation of individual agent is shown in Fig. 1. This framework is revisited, as pointed out in the first paragraph, in [98] and then extended for MASs [24], [75], [99], [112]. Consider a network of N heterogeneous linear systems taking the following form:

$$\dot{x}_i = A_i x_i + B_i u_i, \quad y_i = C_i x_i, \quad i = 1, \ldots, N \tag{12}$$

while the network of N heterogeneous nonlinear systems can be described by

$$\dot{x}_i = f_i(x_i) + g_i(x_i)u_i, \quad y_i = h_i(x_i), \quad i = 1, \ldots, N \tag{13}$$

where $x_i \in \mathbb{R}^{n_i}, u_i \in \mathbb{R}^{m_i}, y_i \in \mathbb{R}^p\mathbb{R}^q$ represent the state, input, and output of the heterogeneous system.

To synchronize the output of the heterogeneous systems, a *dynamic controller* rather than *static controller* is preferred. The general local dynamic controllers for heterogeneous linear systems take the following form (12) [99]:

$$\begin{aligned} \dot{\xi}_i &= D_i\xi_i + E_i y_i + F_i\nu_i, \quad \zeta_i = P_i\xi_i + Qy_i \\ \nu_i &= \sum_{j=1}^{N} a_{ij}(t)(\zeta_j - \zeta_i), \quad u_i = G_i\xi_i + M_i y_i + O_i\nu_i \end{aligned}$$

where $\xi_i \in \mathbb{R}^{\bar{n}_i}$ denotes the controller state, $\zeta_i \in \mathbb{R}^p$ represents controller output, and $\nu_i \in \mathbb{R}^p$ is the input to the dynamic controller that represents the exchange of information between different agents. Correspondingly, the local output-feedback controller for nonlinear heterogeneous systems (13) is [24]

$$\begin{aligned} \dot{\xi}_i &= \Phi_i(\xi_i, y_i, \tilde{\nu}_i), \ u_i = \Gamma_i(\xi_i, y_i, \tilde{\nu}_i) \\ \vartheta_i &= \Theta_i(\xi_i, y_i), \ \tilde{\nu}_i = \sum_{j=1}^{N} a_{ij}\left(\vartheta_j - \vartheta_i\right) \end{aligned}$$

where $\tilde{\nu}_i$ is also the input to the dynamic controller that represents the exchange of information between different agents, $\vartheta_i \in \mathbb{R}^p$ is a measurement taken at agent i. Intuitively, in leader-following framework, the dynamic controller regulates the output of the follower systems to that of the exosystem [12], [75]. While in leaderless framework, the dynamic controller produces the reference trajectory (function as *virtual exosystem*) for each system to track [24], [29], [38], [99]. All the above works are based on embedding an internal model into each system using the dynamic controller, and then regulate each system to the (virtual) exosystem with additional requirement that regulation equation holds. One of the advantages of the dynamic controller presented above is that the output synchronization problem can generally be solved within two steps. First, the coupled local reference generators, which produce the reference trajectories, synchronize with each other. Second, the output of each agent tracks the local reference generator. This advantage simplifies the analysis even when the information flow among agents is directed.

Note that the dynamic controller may require the information of controller state of each agent and its neighbors. For example, ν_i contains the term $\sum_{j=1}^{N} a_{ij}(t)(\xi_j - \xi_i)$ for linear systems in [75] and [99] and $\tilde{\nu}_i$ contains the term $\sum_{j=1}^{N} a_{ij}(\vartheta_j - \vartheta_i)$ for nonlinear systems in [24]. However, the transmission of controller state may be impractical due to communication constraint. Therefore, one may be interested in designing static controller relying on output or state information [76], [117] that can be obtained directly through specific sensors. Another effective approach from the perspective of passivity is provided in [11] to synchronize the output of the nonlinear systems via only relative information. Later in [39], the relation between the passivity of linear systems and the internal model principle for linear systems is built. It is shown therein that the nontrivial synchronization of the passive linear systems is realized if and only if an internal model exists. Though the static controllers using only output information [11], [39], [117] take simpler form than

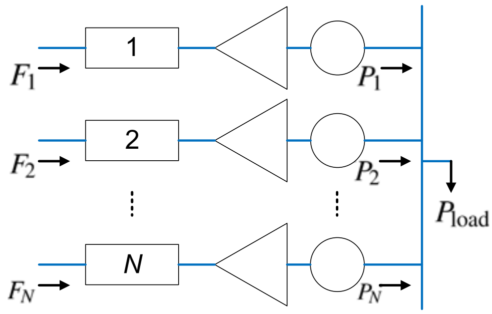

Fig. 2. Network of power generators. P_i is the power of the generator i, F_i represents the cost of each generator, which relates to P_i, $i = 1, \ldots, N$.

the dynamic controller [24], [99], the static controllers work for heterogeneous MASs with restricted network topology.

VI. Applications of Consensus Algorithm

In this section, we discuss briefly some applications of consensus algorithms, which have received growing interest from researchers working on other topics such as EDP in smart grid and k-means clustering algorithms in data mining.

A. Incremental Cost Consensus in Economic Dispatch

A common scenario in power systems is that a group of generators feed electricity power to a common load or bus (see Fig. 2). Since the generators usually have different cost functions, each of which characterizes the relationship between the input cost and the output power of each generator, it is natural to ask how much power each generator should produce to minimize the total cost while meeting the power balance and security constraints. This problem is usually referred to as *economic dispatch problem* (or EDP for short) and formulated as a convex or nonconvex optimization problem [101]. The simplified form of the convex EDP is

$$\min_{P_i} \sum_{i=1}^{N} F_i(P_i), \quad \text{s.t.} \sum_{i=1}^{N} P_i = P_D \quad \text{(energy balance constraint)}$$

where P_D, P_i, and $F_i(P_i)$ are the predicted power demand, generator i's power output, and cost function, respectively. The optimization theory gives that the optimal solution must satisfy $\frac{dF_1}{dP_1} = \frac{dF_2}{dP_2} = \cdots = \frac{dF_N}{dP_N} = \lambda$, which is the equal incremental cost condition, and the power balance constraint. It is worth noting that the MASs under a strongly connected and balanced communication topology possess two useful properties: one is the consensus of the state variables, and the other is the invariant summation of the state variables. By utilizing the first and the second property to meet the equal incremental cost condition and the power constraint, respectively, distributed consensus-based algorithms can be designed to solve the EDP.

A number of papers have proposed different algorithms utilizing the above-mentioned properties to solve the EDP. The paper [128] was the first one utilizing these properties. Zhang and Chow [128] proposed a discrete-time consensus-based algorithm under a connected undirected graph with the existence of a leader node. In this algorithm, only the first property was used. Then in [116], both of the properties were utilized in a leaderless paradigm under a strongly connected and balanced graph. More distributed algorithms can be found in [113] and [129], all of which show that utilizing the above-mentioned properties of MASs is an appealing approach to solve the EDP in a distributed way.

B. Clustering Algorithms

Clustering is the task of partitioning data into clusters with high intracluster and low intercluster similarities, which is used in many application fields including machine learning, pattern recognition, and data mining. A great deal of algorithms have been proposed to deal with the clustering problems, among which the k-means algorithm is the most popular one [53]. Generally speaking, a k-means algorithm proceeds by alternating between an assignment step to assign each data to the cluster characterized by the nearest centroid and an update step to calculate the new averages to be the centroids of the data in the new clusters after an initialization setup. In large-scale distributed systems, for example, sensor networks consisting of thousands of small sensor nodes, it is important to propose a distributed k-means algorithm since it is impractical to centralize the whole distributed data to one fusion node to perform the k-means algorithm. To design the distributed k-means algorithm, a main step is to choose the average of a cluster of data as the centroid of the cluster in a distributed way. Since the consensus algorithms in MASs are effective in computing the average in a distributed way, they have been utilized to help design the distributed k-means algorithms [20], [53], [71]. In [20], [53], and [71], gossip-based consensus algorithm and finite-time average consensus algorithms are, respectively, used to propose the distributed k-means algorithms.

VII. Conclusion

We have presented a brief survey of the results for consensus of MASs, mainly from a control perspective, which aims to provide the interested readers with an overview of some interesting topics, which have received much attention in the past few years. However, it was worth pointing out that this survey is far from an exhaustive literature review and there may still be some important results missing in the review due to space limitation.

References

[1] C. Altafini, "Consensus problems on networks with antagonistic interactions," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 935–946, Apr. 2013.

[2] F. Arrichiello , A. Marino , and F. Pierri, "Observer-based decentralized fault detection and isolation strategy for networked multirobot systems," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1465–1476, Jul. 2015.

[3] D. Cartwright and F. Harary, "Structural balance: A generalization of Heider's theory," *Psychol. Rev.*, vol. 63, no. 5, pp. 277–293, Sep. 1956.

[4] 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.

[5] M. Cao, F. Xiao, and L. Wang, "Second-order leader-following consensus based on time and event hybrid-driven control," *Syst. Control Lett.*, vol. 74, pp. 90–97, Dec. 2014.

[6] M. Cao, F. Xiao, and L. Wang, "Event-based second-order consensus control for multi-agent systems via synchronous periodic event detection," *IEEE Trans. Autom. Control*, vol. 60, no. 9, pp. 2452–2457, Sep. 2015.

[7] M. Cao, F. Xiao, and L. Wang, "Second-order consensus in time-delayed networks based on periodic edge-event driven control," *Syst. Control Lett.*, vol. 96, pp. 37–44, Oct. 2016.
[8] Y. Cheng and V. Ugrinovskii, "Event-triggered leader-following tracking control for multivariable multi-agent systems," *Automatica*, vol. 70, pp. 204–210, Aug. 2016.
[9] S. Chen, D. W. C. Ho, L. Li, and M. Liu, "Fault-tolerant consensus of multi-agent system with distributed adaptive protocol," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2142–2155, Oct. 2015.
[10] Y. Chen and Y. Shi, "Consensus for linear multi-agent systems with time-varying delays: A frequency domain perspective," *IEEE Trans. Cybern.*, 2016, to be published, doi: 10.1109/TCYB.2016.2590480.
[11] N. Chopra, "Output synchronization on strongly connected graphs," *IEEE Trans. Autom. Control*, vol. 57, no. 11, pp. 2896–2901, Nov. 2012.
[12] Z. Ding, "Consensus output regulation of a class of heterogeneous nonlinear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 10, pp. 2648–2653, Oct. 2013.
[13] M. Davoodi, N. Meskin, and K. Khorasani, "Simultaneous fault detection and consensus control design for a network of multi-agent systems," *Automatica*, vol. 66, pp. 185–194, Apr. 2016.
[14] M. Davoodi, K. Khorasani, H. A. Talebi, and H. R. Momeni, "Distributed fault detection and isolation filter design for a network of heterogeneous multi-agent systems," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1061–1069, May 2014.
[15] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, May 2012.
[16] M. H. DeGroot, "Reaching a consensus," *J. Amer. Statist. Assoc.*, vol. 69, no. 345, pp. 118–121, Apr. 1974.
[17] P. Delellis, M. di Bernardo, and F. Garofalo, "Novel decentralized adaptive strategies for the synchronization of complex networks," *Automatica*, vol. 45, no. 5, pp. 1312–1318, May 2009.
[18] Y. Fan, G. Feng, Y. Wang, and C. Song, "Distributed event-triggered control of multi-agent systems with combinational measurements," *Automatica*, vol. 49, no. 2, pp. 671–675, Feb. 2013.
[19] Y. Fan, L. Liu, G. Feng, and Y. Wang, "Self-triggered consensus for multi-agent systems with zeno-free triggers," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2779–2784, Oct. 2015.
[20] G. Di Fatta, F. Blasa, S. Cafiero, and G. Fortino, "Fault tolerant decentralised k-means clustering for asynchronous large-scale networks," *J. Parallel Distrib. Comput.*, vol. 73, no. 3, pp. 317–329, Mar. 2013.
[21] E. Garcia, Y. Cao, and D. W. Casbeer, "Decentralized event-triggered consensus with general linear dynamics," *Automatica*, vol. 50, no. 10, pp. 2633–2640, Oct. 2014.
[22] Y. Han, W. Lu, and T. Chen, "Achieving cluster consensus in continuous-time networks of multi-agents with inter-cluster non-identical inputs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 793–798, Mar. 2015.
[23] Y. Hong, J. Hu, and L. Gao, "Tracking control for multi-agent consensus with an active leader and variable topology," *Automatica*, vol. 42, no. 7, pp. 1177–1182, Jul. 2006.
[24] A. Isidori, L. Marconi, and G. Casadei, "Robust output synchronization of a network of heterogeneous nonlinear agents via nonlinear regulation theory," *IEEE Trans. Autom. Control*, vol. 59, no. 10, pp. 2680–2691, Oct. 2014.
[25] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa, "Containment control in mobile networks," *IEEE Trans. Autom. Control*, vol. 53, no. 8, pp. 1972–1975, Sep. 2008.
[26] X. Jin and J. H. Park, "Adaptive synchronization for a class of faulty and sampling coupled networks with its circuit implement," *J. Franklin Inst.*, vol. 351, no. 8, pp. 4317–4333, Aug. 2014.
[27] N. Kashyap, C. W. Yang, S. Sierla, and P. G. Flikkema, "Automated fault location and isolation in distribution grids with distributed control and unreliable communication," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2612–2619, Apr. 2015.
[28] A. Kashyap, T. Basar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, Jul. 2007.
[29] H. Kim, H. Shim, and J. H. Seo, "Output consensus of heterogeneous uncertain linear multi-agent systems," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 200–206, Jan. 2011.
[30] Z. Lin, *Low Gain Feedback*. London, U.K.: Springer-Verlag, 1998.
[31] Y. Li, J. Xiang, and W. Wei, "Consensus problems for linear time-invariant multi-agent systems with saturation constraints," *IET Control Theory Appl.*, vol. 5, no. 6, pp. 823–829, Apr. 2011.
[32] H. Li and Y. Shi, *Robust Receding Horizon Control for Networked and Distributed Nonlinear Systems*. New York, NY, USA: Springer, 2017.
[33] J. Lyu, J. Qin, D. Gao, and Q. Liu, "Consensus for constrained multi-agent systems with input saturation," *Int. J. Robust Nonlinear Control*, vol. 26, no. 14, pp. 1099–1239, Dec. 2016.
[34] X. Liu and T. Chen, "Cluster synchronization in directed networks via intermittent pinning control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 22, no. 7, pp. 1009–1020, May 2011.
[35] W. Lu, B. Liu, and T. Chen, "Cluster synchronization in networks of coupled nonidentical dynamical systems," *Chaos*, vol. 20, no. 1, Mar. 2010, Art. no. 013120.
[36] W. Lu, Y. Han, and T. Chen, "Synchronization in networks of linearly coupled dynamical systems via event-triggered diffusions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3060–3069, Dec. 2015.
[37] W. Lohmiller and J .-J. E. Slotine, "On contraction analysis for non-linear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, Jun. 1998.
[38] Z. Liu and W. S. Wong, "Output cluster synchronization of heterogeneous linear multi-agent systems," in *Proc. 54th Annu. IEEE Conf. Decis. Control*, Osaka, Japan, Dec. 2015, pp. 2853–2858.
[39] S.-J. Lee, K .-K. Oh, and H.-S. Ahn, "Non-trivial output synchronization of heterogeneous passive systems," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3322–3327, Dec. 2015.
[40] H. Li, X. Liao, T. Huang, and W. Zhu, "Event-triggering sampling based leader-following consensus in second-order multi-agent systems," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 1998–2003, Jul. 2015.
[41] Q. Liu, Z. Wang, X. He, and D. H. Zhou, "Event-based H_∞ consensus control of multi-agent systems with relative output feedback: The finite-horizon case," *IEEE Trans. Autom. Control*, vol. 60, no. 9, pp. 2553–2558, Sep. 2015.
[42] X. Meng and T. Chen, "Event based agreement protocols for multi-agent networks," *Automatica*, vol. 49, no. 7, pp. 2125–2132, Jul. 2013.
[43] D. Meng, Y. Jia, and J. Du, "Nonlinear finite-time bipartite consensus protocol for multi-agent systems associated with signed graphs," *Int. J. Control*, vol. 88, no. 10, pp. 2074–2085, Jun. 2015.
[44] Z. Meng, G. Shi, K. H. Johansson, M. Cao, and Y. Hong, "Behaviors of networks with antagonistic interactions and switching topologies," *Automatica*, vol. 73, pp. 110–116, 2016.
[45] Z. Meng, Z. Zhao, and Z. Lin, "On global leader-following consensus of identical linear dynamic systems subject to actuator saturation," *Syst. Control Lett.*, vol. 62, no. 2, pp. 132–142, Feb. 2013.
[46] N. Mu, X. Liao, and T. Huang, "Event-based consensus control for a linear directed multi-agent system with time delay," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 3, pp. 281–285, Mar. 2015.
[47] P. P. Menon and C. Edwards, "Robust fault estimation using relative information in linear multi-agent networks," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 477–482, Feb. 2014.
[48] N. Meskin and K. Khorasani, "Actuator fault detection and isolation for a network of unmanned vehicles," *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 835–840, Apr. 2009.
[49] N. Meskin, K. Khorasani, and C. A. Rabbath, "A hybrid fault detection and isolation strategy for a network of unmanned vehicles in presence of large environmental disturbances," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 6, pp. 1422–1429, Nov. 2010.
[50] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.
[51] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
[52] 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.
[53] G. Oliva, R. Setola, and C. N. Hadjicostis, "Distributed k-means algorithm," arXiv:abs/1312.4176.
[54] A. Proskurnikov and M. Cao, "Opinion dynamics using Altafini's model with a time-varying directed graph," in *Proc. IEEE Int. Symp. Intell. Control*, Antibes, France, Oct. 2014, pp. 849–854.
[55] Q.-C. Pham and J .-J. Slotine, "Stable concurrent synchronization in dynamic system networks," *Neural Netw.*, vol. 20, no. 1, pp. 62–77, Jan. 2007.
[56] S. Patterson, B. Bamieh, and A. E. Abbadi, "Convergence rates of distributed average consensus with stochastic link failures," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 880–892, Apr. 2010.

[57] Z. Peng, D. Wang, Y. Shi, H. Wang, and W. Wang, "Containment control of networked autonomous underwater vehicles with model uncertainty and ocean disturbance guided by multiple leaders," *Inf. Sci.*, vol. 316, pp. 163–179, Sep. 2015.
[58] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.
[59] J. Qin, H. Gao, and W. X. Zheng, "Second-order consensus for multi-agent systems with switching topology and communication delay," *Syst. Control Lett.*, vol. 60, no. 6, pp. 390–397, Jun. 2011.
[60] J. Qin, C. Yu, and H. Gao, "Coordination for linear multi-agent systems with dynamic interaction topology in the leader-following framework," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2412–2422, May 2014.
[61] J. Qin and H. Gao, "A sufficient condition for convergence of sampled-data consensus for double-integrator dynamics with nonuniform and time-varying communication delays," *IEEE Trans. Autom. Control*, vol. 57, no. 9, pp. 2417–2422, Sep. 2012.
[62] J. Qin, W. X. Zheng, and H. Gao, "Consensus of multiple second-order vehicles with a time-varying reference signal under directed topology," *Automatica*, vol. 47, no. 9, pp. 1983–1991, Sep. 2011.
[63] J. Qin, W. X. Zheng, and H. Gao, "Coordination of multiple agents with double-integrator dynamics under generalized interaction topologies," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 44–57, Feb. 2012.
[64] J. Qin, W. X. Zheng, H. Gao, Q. Ma, and W. Fu, "Containment control for second-order multi-agent systems communicating over heterogeneous networks," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2574830.
[65] J. Qin, H. Gao, and W. X. Zheng, "Exponentially synchronization of complex networks of linear systems and nonlinear oscillators: A unified analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 510–521, Mar. 2015.
[66] J. Qin, W. Fu, W. X. Zheng, and H. Gao, "On the bipartite consensus for generic linear multi-agent systems with input saturation," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2612482.
[67] J. Qin, Q. Ma, W. X. Zheng, and H. Gao, "H_∞ group consensus for cluster of agents with model uncertainty and external disturbance," in *Proc. 54th Annu. IEEE Conf. Decis. Control*, Osaka, Japan, Dec. 2015, pp. 2841–2846.
[68] J. Qin and C. Yu, "Group consensus of multiple integrator agents under general topology," in *Proc. 52nd Annu. IEEE Conf. Decis. Control*, Florence, Italy, Dec. 2013, pp. 2752–2757.
[69] J. Qin, C. Yu, and B. D. O. Anderson, "On leaderless and leader-following consensus for interacting clusters of double-integrator multi-agent systems," *Automatica*, vol. 74, pp. 214–221, Dec. 2016.
[70] J. Qin and C. Yu, "Cluster consensus control of generic linear multi-agent systems under directed topology with acyclic partition," *Automatica*, vol. 49, no. 9, pp. 2898–2905, Sep. 2013.
[71] J. Qin, W. Fu, H. Gao, and W. X. Zheng, "Distributed k-means algorithm and fuzzy c-means algorithm for sensor networks based on multi-agent consensus theory," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2526683.
[72] J. Qin, Q. Ma, H. Gao, Y. Shi, and Y. Kang, "On group synchronization for interacting clusters of heterogeneous systems," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2600753.
[73] W. Ren, "On consensus algorithms for double-integrator dynamics," *IEEE Trans. Autom. Control*, vol. 53, no. 6, pp. 1503–1509, Jul. 2008.
[74] G. Russo and M. di Bernardo, "Contraction theory and master stability function: Linking two approaches to study synchronization of complex networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 2, pp. 177–181, Feb. 2009.
[75] Y. Su and J. Huang, "Cooperative output regulation of linear multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, no. 4, pp. 1062–1066, Apr. 2012.
[76] G. S. Seyboth, D. V. Dimarogonas, K. H. Johansson, P. Frasca, and F. Allgöwer, "On robust synchronization of heterogeneous linear multi-agent systems with static couplings," *Automatica*, vol. 53, pp. 392–399, Mar. 2015.
[77] G. Shi, A. Proutiere, M. Johansson, J. S. Baras, and K. H. Johansson, "Emergent behaviors over signed random dynamical networks: State-flipping model," *IEEE Trans. Control Syst. Technol.*, vol. 2, no. 2, pp. 142–153, Jun. 2015.
[78] G. Shi and Y. Hong, "Global target aggregation and state agreement of nonlinear multi-agent systems with switching topologies," *Automatica*, vol. 45, no. 5, pp. 1165–1175, May 2009.
[79] G. Shi and K. H. Johansson, "Randomized optimal consensus of multi-agent systems," *Automatica*, vol. 48, no. 12, pp. 3018–3030, Dec. 2012.
[80] G. Shi, M. Johansson, and K. H. Johansson, "How agreement and disagreement evolve over random dynamic networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1061–1071, Jun. 2013.
[81] E. Semsar-Kazerooni and K. Khorasani, "Analysis of actuator faults in a cooperative team consensus of unmanned systems," in *Proc. Amer. Control Conf.*, St. Louis, MO, USA, Jun. 2009, pp. 2618–2623.
[82] I. Saboori and K. Khorasani, "Actuator fault accommodation strategy for a team of multi-agent systems subject to switching topology," *Automatica*, vol. 62, pp. 200–207, Dec. 2015.
[83] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, Jan. 2013.
[84] L. Scardovi and R. Sepulchre, "Synchronization in networks of identical linear systems," *Automatica*, vol. 45, no. 11, pp. 2557–2562, Jan. 2009.
[85] D. S. Bernstein and A. N. Michel, "A chronological bibliography on saturating actuators," *Int. J. Robust Nonlinear Control*, vol. 5, no. 5, pp. 375–380, Mar. 1995.
[86] H. Su, M. Z. Q. Chen, J. Lam, and Z. Lin, "Semi-global leader-following consensus of linear multi-agent systems with input saturation via low gain feedback," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 7, pp. 1881–1889, Jul. 2013.
[87] H. Su, M. Z. Q. Chen, X. Wang, and J. Lam, "Semiglobal observer-based leader-following consensus with input saturation," *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 2842–2850, Jun. 2014.
[88] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.
[89] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Distributed fault detection and isolation resilient to network model uncertainties," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2024–2037, Nov. 2014.
[90] S. Tuna, "Synchronizing linear systems via partial-state coupling," *Automatica*, vol. 44, no. 8, pp. 2179–2184, Aug. 2008.
[91] J. Wu and Y. Shi, "Consensus in multi-agent systems with random delays governed by a Markov chain," *Syst. Control Lett.*, vol. 60, no. 10, pp. 863-870, Oct. 2011.
[92] Q. Wang, C. Yu, and H. Gao, "Synchronization of identical linear dynamic systems subject to input saturation," *Syst. Control Lett.*, vol. 64, pp. 107–113, Feb. 2014.
[93] Y. Wang, Y. Song, and F. L. Lewis, "Robust adaptive fault-tolerant control of multi-agent systems with uncertain nonidentical dynamics and undetectable actuation failures," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3978–3988, Jun. 2015.
[94] Y. Wang, Y. Song, M. Krstic, and C. Wen, "Fault-tolerant finite time consensus for multiple uncertain nonlinear mechanical systems under single-way directed communication interactions and actuation failures," *Automatica*, vol. 63, pp. 374–383, Jan. 2016.
[95] W. Wu, W. Zhou, and T. Chen, "Cluster synchronization of linearly coupled complex networks under pinning control," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 4, pp. 829–839, Apr. 2009.
[96] W. Wang and J .-J. E. Slotine, "Contraction analysis of time-delayed communications and group cooperation," *IEEE Trans. Autom. Control*, vol. 51, no. 4, pp. 712–717, Apr. 2006.
[97] J. Wang, J. Feng, C. Xu, and Y. Zhao, "Cluster synchronization of nonlinearly-coupled complex networks with nonidentical nodes and asymmetrical coupling matrix," *Nonlinear Dyn.*, vol. 67, no. 2, pp. 1635–1646, Jan. 2012.
[98] P. Wieland and F. Allgöwer, "An internal model principle for consensus in heterogeneous linear multi-agent systems," in *Proc. 1st IFAC Workshop Estimation Control Netw. Syst.*, Venice, Italy, Sep. 2009, pp. 7–12.
[99] P. Wieland, R. Sepulchre, and F. Allgöwer, "An internal model principle is necessary and sufficient for linear output synchronization," *Automatica*, vol. 47, no. 5, pp. 1068–1074, May 2011.
[100] J. Wu and Y. Shi, "Consensus in multi-agent systems with random delays governed by a Markov chain," *Syst. Control Lett.*, vol. 60, no. 10, pp. 863–870, Oct. 2011.
[101] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. Hoboken, NJ, USA: Wiley, 2012.
[102] F. Xiao, X. Meng, and T. Chen, "Average sampled-data consensus driven by edge events," in *Proc. 31st Chin. Control Conf.*, Hefei, China, Jul. 2012, pp. 6239–6244.
[103] F. Xiao, T. Chen, and H. Gao, "Synchronous hybrid event- and time-driven consensus in multi-agent networks with time delays," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1165–1174, May 2016.

[104] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Trans. Autom. Control*, vol. 53, no. 8, pp. 1804–1816, Sep. 2008.

[105] W. Xu, D. W. C. Ho, L. Li, and J. Cao, "Event-triggered schemes on leader-following consensus of general linear multi-agent systems under different topologies," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 212–223, Jan. 2017.

[106] W. Xia and M. Cao, "Clustering in diffusively coupled networks," *Automatica*, vol. 47, no. 11, pp. 2395–2405, Nov. 2011.

[107] J. Yu and L. Wang, "Group consensus in multi-agent systems with switching topologies and communication delays," *Syst. Control Lett.*, vol. 59, no. 6, pp. 340–348, Nov. 2010.

[108] J. Yu and L. Wang, "Group consensus of multi-agent systems with directed information exchange," *Int. J. Syst. Sci.*, vol. 43, no. 2, pp. 334–348, Feb. 2012.

[109] C. Yu, J. Qin, and H. Gao, "Cluster synchronization in directed networks of partial-state coupled linear systems under pinning control," *Automatica*, vol. 50, no. 9, pp. 2341–2349, Sep. 2014.

[110] C. Yu and J. Qin, "Synchronization for interacting clusters of generic linear agents and nonlinear oscillators: A unified analysis," in *Proc. 19th World Congr. Int. Fed. Autom. Control*, Cape Town, South Africa, Aug. 2014, pp. 1965–1970.

[111] T. Yang, Z. Meng, G. Shi, Y. Hong, and K. H. Johansson, "Network synchronization with nonlinear dynamics and switching interactions," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 3103–3108, Oct. 2016.

[112] T. Yang, A. Saberi, A. A. Stoorvogel, and H. F. Grip, "Output synchronization for heterogeneous networks of introspective right-invertible agents," *Int. J. Robust Nonlinear Control*, vol. 24, no. 13, pp. 1821–1844, Sep. 2014.

[113] T. Yang, D. Wu, Y. Sun, and J. Lian, "Minimum-time consensus-based approach for power system applications," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1318–1328, Feb. 2016.

[114] T. Yang, S. Roy, Y. Wan, and A. Saberi, "Constructing consensus controllers for networks with identical general linear agents," *Int. J. Robust Nonlinear Control*, vol. 21, no. 11, pp. 1237–1256, Aug. 2011.

[115] T. Yang, Z. Meng, D. V. Dimarogonas, and K. H. Johansson, "Global consensus for discrete-time multi-agent systems with input saturation constraints," *Automatica*, vol. 50, no. 2, pp. 499–506, Feb. 2014.

[116] S. Yang, S. Tan, and J.-X. Xu, "Consensus based approach for economic dispatch problem in a smart grid," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4416–4426, Nov. 2013.

[117] Y. Zheng and L. Wang, "Finite-time consensus of heterogeneous multi-agent systems with and without velocity measurements," *Syst. Control Lett.*, vol. 61, no. 8, pp. 871–878, Aug. 2012.

[118] W. Yu, G. Chen, and M. Cao, "Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems," *Automatica*, vol. 46, no. 6, pp. 1089–1095, Jun. 2010.

[119] W. Yu, P. Delellis, G. Chen, M. di Bernardo, and J. Kurths, "Distributed adaptive control of synchronization in complex networks," *IEEE Trans. Autom. Control*, vol. 57, no. 8, pp. 2153–2158, Aug. 2012.

[120] X. Yi, W. Lu, and T. Chen, "Distributed event-triggered consensus for multi-agent systems with directed topologies," arXiv:abs/1407.3075v2.

[121] Z. Zuo, J. Zhang, and Y. Wang, "Adaptive fault-tolerant tracking control for linear and Lipschitz nonlinear multi-agent systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3923–3931, Jun. 2015.

[122] W. Zhu and Z. P. Jiang, "Event-based leader-following consensus of multi-agent systems with input time delay," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1362–1367, May 2015.

[123] H. Zhang and J. Chen, "Bipartite consensus of general linear multi-agent systems," in *Proc. Amer. Control Conf.*, Portland, OR, USA, Jun. 2014, pp. 808–812.

[124] B. Zhou, G. Duan, and Z. Lin, "A parametric Lyapunov equation approach to the design of low gain feedback," *IEEE Trans. Autom. Control*, vol. 53, no. 6, pp. 1548–1554, Jul. 2008.

[125] Z. Zhao and Z. Lin, "Global leader-following consensus of a group of general linear systems using bounded controls," *Automatica*, vol. 68, pp. 294–304, Jun. 2016.

[126] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annu. Rev. Control*, vol. 32, no. 2, pp. 229–252, Dec. 2008.

[127] Q. Zhang and X. Zhang, "Distributed sensor fault diagnosis in a class of interconnected nonlinear uncertain systems," *Annu. Rev. Control*, vol. 37, no. 1, pp. 170–179, Apr. 2013.

[128] Z. Zhang and M .-Y. Chow, "Convergence analysis of the incremental cost consensus algorithm under different communication network topologies in a smart grid," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 1761–1768, Nov. 2012.

[129] W. Zhang, W. Liu, X. Wang, L. Liu, and F. Ferrese, "Online optimal generation control based on constrained distributed gradient algorithm," *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 35–45, Jan. 2015.

Jiahu Qin (M'12) received the first Ph.D. degree in control science and engineering from Harbin Institute of Technology, Harbin, China, and the second Ph.D. degree in systems and control from the Australian National University, Canberra, Australia, in 2012 and 2014, respectively.

Since July 2013, he has been with the University of Science and Technology of China, Hefei, China, where he is currently a Professor. His research interests include consensus and coordination in multi-agent systems, as well as complex network theory and its application.

Qichao Ma (S'16) received the B.S. degree in electronic information science and technology from the China University of Mining and Technology, Xuzhou, China, in 2013. He is currently working toward the Ph.D. degree in control science and engineering at the University of Science and Technology of China, Hefei, China.

His research interests include consensus in multi-agent systems and synchronization of complex dynamical networks.

Yang Shi (S'97–M'05–SM'09–F'17) received the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2005.

He is currently a Professor in the Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada. His research interests include networked and distributed systems, model predictive control, system identification, mechatronics, autonomous vehicles, cyber-physical systems, and energy system applications.

Prof. Shi serves as an Associate Editor for the IEEE/ASME TRANSACTIONS MECHATRONICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, IEEE TRANSACTIONS ON CYBERNETICS, *IET Control Theory and Applications*, and *ASME Journal of Dynamic Systems, Measurement, and Control*. He is currently a Fellow of the American Society of Mechanical Engineers and the Canadian Society for Mechanical Engineers, and a Registered Professional Engineer in the Province of British Columbia, Canada.

Long Wang (M'99) received the Ph.D. degree in dynamics and control from Peking University, Beijing, China, in 1992.

He has held research positions at the University of Toronto, Toronto, ON, Canada, and the German Aerospace Center, Munich, Germany. He is currently a Cheung Kong Chair Professor of Dynamics and Control, and the Director of the Center for Systems and Control, Peking University. He is also a Guest Professor at Wuhan University, Wuhan, China, and Beihang University, Beijing, China. He serves as the Chairman of the Chinese Intelligent Networked Things Committee, and a Member of the IFAC Technical Committee on Networked Systems. His research interests include networked systems, collective intelligence, and biomimetic robotics.

Prof. Wang is a member of the Editorial Boards of *Science in China*, *Journal of Intelligent Systems*, *Journal of Control Theory and Applications*, *PLoS ONE*, *Journal of Intelligent and Robotic Systems*, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, etc.