Convergence Rate of Accelerated Average Consensus With Local Node Memory: Optimization and Analytic Solutions

Jing-Wen Yi ©, Member, IEEE, Li Chai ©, Member, IEEE, and Jingxin Zhang ©, Member, IEEE

Abstract—Previous research works have shown that adding local memory can accelerate the consensus. It is natural to question what is the fastest rate achievable by the $M$-tap memory acceleration, and what are the corresponding control parameters. This article introduces a set of effective and previously unused techniques to analyze the convergence rate of accelerated consensus with $M$-tap memory of local nodes and to design the control protocols. These effective techniques, including the Kharitonov stability theorem, the Routh stability criterion, and the robust stability margin, have led to the following new results: first, the direct link between the convergence rate and the control parameters; second, explicit formulas of the optimal convergence rate and the corresponding optimal control parameters for $M \leq 2$ on a given graph; third, analytic formulas of the optimal worst-case convergence rate and the corresponding optimal control parameters for the memory $M \geq 1$ on a set of uncertain graphs. We show that the acceleration with the memory $M = 1$ provides the optimal convergence rate in the sense of the worst-case performance. Several numerical examples are given to demonstrate the validity and performance of the theoretical results.

Index Terms—Accelerated algorithm, average consensus, convergence rate, multiagent systems.

I. INTRODUCTION

Distributed consensus algorithms have recently received renewed interests due to their wide applications in various fields, such as multirobot systems, wireless sensor networks, and smart grids [1], [2], [3]. For distributed average consensus of multiagent system (MAS), each agent only gets information from its local neighbors, and the whole network of agents coordinates to reach the average of their initial states.

Accelerated consensus has been an active research area in MAS driven by a variety of applications. Revealing the relationship between the convergence rate of consensus and the property of the network is a key to the design of accelerated consensus algorithms. This has motivated a great deal of research on the analysis of accelerated consensus and its convergence rate.

The work of [4], [5], [6], [7], and [8] showed that the consensus can be accelerated by optimizing the weight matrix of MAS to improve the network connectivity. A variety of consensus protocols were proposed in [9], [10], [11], and [12] to achieve faster convergence rates for MAS on time-varying graphs. For MAS with fixed and known graph topology, it was shown in [13], [14], [15], [16], [17], and [18] that the finite-time consensus can be achieved by choosing the time-varying control gain to be the reciprocal of nonzero Laplacian eigenvalues. For MAS on uncertain graphs, Yi et al. [19] established the direct link between the consensus convergence rate and the graph filter representing a periodically time varying consensus protocol, and provided the explicit formulas for the convergence rate and the parameters of the control protocols. Memory information has been employed in various consensus algorithms to accelerate convergence. It was used in a gossip algorithm in [20] to improve the convergence rate of expectation, and the fastest convergence rate of expectation was derived in [21] under a symmetry assumption. For faster convergence, Johansson and Johansson [22] introduced the shift-register algorithm and simulated some possible optimization schemes to find the optimal weight matrix; Moradian and Kia [23] proposed a dynamic average consensus algorithm using only the delayed states and studied the relationship among the delay bound, the graph degree, the convergence rate and the tracking error bound. The work of [24] designed a hybrid model with memory-based connectivity and proved the stability properties ensuring asymptotic convergence to local agreements/cluster.

As shown in [25], the average consensus is a special type of optimization problems with a convex and smooth cost function whose gradient can be computed decentrally by the agents at nodes. Thus, any accelerated first-order gradient algorithm for solving such optimizations can be viewed as an accelerated consensus algorithm, and the results on the accelerated first-order
gradient algorithms for solving the general convex optimizations may be used to analyze and develop average consensus algorithms. Specifically, Nesterov’s optimal worst-case convergence rate for a general class of accelerated first-order gradient algorithms [26] may be used to quantify the optimal worst-case convergence rate of accelerated consensus. Along this line, Ghadimi et al. [25] formulated a distributed resource allocation problem as an optimization problem, with average consensus as a special case. By introducing an accelerated gradient algorithm with one-tap memory, Ghadimi et al. [25] derived the optimal parameters and step size, and studied the sensitivity of the optimal convergence rate to parameter uncertainty. It was shown in [27] that the convergence rate of the decentralized Nesterov’s accelerated gradient algorithm is related to the condition number of the cost function defined in [26] and the diameter of the network. By utilizing the Chebyshev acceleration technique, Scaman et al. [27] provided a multistep dual accelerate method, and proved that the optimal convergence rate depends on the condition number of the local functions and the eigengap of the system Laplacian matrix. More results on the optimization based analysis and algorithms can be found in the recent surveys [28], [29].

Approaching the accelerated consensus problem from different angles, the abovementioned works have provided invaluable insights into and effective solutions for the problem under their respective assumptions and formulations. Despite their diversity, these works share one commonality that their algorithms implement explicitly or implicitly some nonlinear and/or time-varying consensus protocols which yield a nonlinear and/or time-varying closed loop system when applied to an MAS.

In this article, we consider a special type of accelerated average consensus problem: the consensus protocols and the resulting closed system are LTI, each node updates its state at each iteration using its own current and memory states and local neighbors’ current states [30], [31], [32], [33]. The decentralized computations at nodes implicitly minimizes a simple convex cost function of the system states \( \min_x f(x) = \min_{x_1=x_2=\ldots=x_N} \sum_{i=1}^{N} (x_i - x_i(0))^2 \) to attain the average consensus at the fastest possible rate.

This type of accelerated consensus is attractive because it can accelerate the convergence rate without increasing the communication cost and its computations are simpler as compared with those reviewed above. Also, since its closed-loop system is LTI, some powerful analysis and design tools from control theory can be applied to obtain analytical and precise solutions. We, therefore, focus on this special type of accelerated consensus in this article to address some open problems of the following MAS:

\[
x_i(k+1) = x_i(k) + \alpha \sum_{j \in N_i} a_{ij} (x_j(k) - x_i(k)) + \sum_{m=0}^{M} \theta_m x_i(k-m) \quad (1)
\]

where \( x_i(k) \in \mathbb{R} \) is the state of agent \( i \) at time \( k \), \( N_i \) is the set of neighbors of agent \( i \), \( \alpha, \theta_0, \ldots, \theta_M \) is the constant control parameters, and \( a_{ij} \geq 0 \) is the weight coupling agents \( i \) and \( j \).

Let \( x(k) = [x_1(k), \ldots, x_N(k)]^T \) and \( L \) be the Laplacian matrix of (1). The overall system becomes

\[
x(k+1) = x(k) - \alpha L x(k) + \sum_{m=0}^{M} \theta_m x(k-m). \quad (2)
\]

The accelerated consensus algorithms of [30], [31], [32], [33], [34], and [35] can be regard as special cases of (2). When \( M = 0 \) and \( \theta_0 = 0 \), system (2) can reach average consensus if and only if the weight matrix \( W = I - \alpha L \) has a simple eigenvalue 1 and all other eigenvalues inside the unit circle; its convergence rate is determined by \( \rho_s(W) \), the second largest eigenvalue modulus of \( W \) [34], and attains the optimal convergence rate \( \frac{2\lambda_N - \lambda_2}{\lambda_N + \lambda_2} \) at \( \alpha = \frac{1}{\lambda_N + \lambda_2} \), where \( \lambda_N \) and \( \lambda_2 \) are, respectively, the largest and the second smallest eigenvalues of \( L \) [35]. To accelerate the convergence rate, Muthukrishnan et al. [30] augmented the state-update equation by setting \( \alpha = 1 + g \), \( \theta_0 = g \lambda_1 = -g \) for system (2) with \( M = 1 \), where \( g \in (0,1) \) was a real-valued parameter dependent on \( W = I - L \). Choosing \( g \in (0, \rho_s(W)^2) \), Liu and Morse [31] proved that the augmented system of [30] converges faster than the original system (2) with \( M = 0 \) and \( \theta_0 = 0 \), and attains the fastest convergence rate \( \frac{\rho_s(W)}{1 + \sqrt{1 - \rho_s(W)^2}} \).

Setting \( \alpha = 1 + g + \beta_3 \), \( \theta_0 = g \beta_2 + g \beta_3 - g \), \( \theta_1 = g \beta_1 \), with \( \beta_1, \beta_2, \beta_3 \) the prespecified parameters, Oreshkin et al. [32] studied a more general one-tap memory acceleration scheme and provided a feasible design of the parameter \( g \). The state-update equations with multiple-tap memory was considered in [33], but the solution was only for the simplest case of \( M = 0 \) and, hence, coincided fully with that of [35].

To improve the convergence speed, Olshevsky [36] proposed a protocol for each agent to use the memories of the node and neighbors, and Pasolini et al. [37] utilized the past information of the node and its neighbors to design an FIR loop filter-based consensus algorithm. It was claimed that the analytical solution of [37] is unavailable except in the special case \( M = 1 \). The work of [38] introduced an accelerate consensus protocol for continuous-time systems with agents using neighbors’ memory information to anticipate their future state, and proved that the convergence rate corresponding to the dominant root \(-1\) can be guaranteed.

Despite the excellent works discussed above, there are still some unsolved problems in the accelerated consensus of MAS (1) by the LTI protocols with memory information.

1) For the accelerated consensus with one-tap memory, it is unclear which one is the best among existing algorithms.
2) There are no results about the convergence rate and the corresponding parameters for the accelerated consensus with general \( M \)-tap memory (\( M \geq 2 \)). Intuitively, a faster rate than that of \( M = 1 \) is expected. However, the problem becomes very difficult even for \( M = 2 \).
3) What is the limitation of the optimal convergence rate when the memory tap \( M \) approaches to infinity?
4) What is the optimal worst-case convergence rate if the MAS is on a set of unknown graphs with general \( M \)-tap memory?
These unsolved problems are fundamental to the analysis, design, and practical applications of accelerated consensus and have motivated us to introduce, a set of effective and previously unused techniques for their solutions. These techniques include the Kharitonov stability theorem, the Routh stability criterion, and the robust stability margin. Using these techniques we derive the following new results.

1) For the MAS on a fixed graph with $N$ agents and $M$-tap memory, we present a necessary and sufficient condition for the average consensus, and prove that the convergence rate equals the maximum modulus root of $N$ polynomials with order no greater than $M + 1$.

2) For the MAS on a fixed graph with $M$-tap memory ($M \leq 2$), we derive the analytic formulas of the optimal convergence rate and the corresponding parameters. These formulas show a counter-intuition result that the optimal rate for $M = 2$ equals that of $M = 1$, which means adding one more memory does not accelerate the convergence rate. The proof technique is novel which includes finding the analytic solution of an optimization problem with 13 nonlinear constraints and proving its uniqueness.

3) For the MAS on a star graph with 3-tap memory, we show through an example that the convergence rate can be faster than that of $M \leq 2$. We make a conjecture through numerical experiments that the optimal convergence rate of $M = 2$ is equal to that of $M = 2K - 1$, $K = 1, 2, \ldots$. However, we cannot prove it.

4) For the MAS with general $M$-tap memory on a set of graphs with $[\lambda_2, \lambda_N] \subseteq [\underline{\lambda}, \bar{\lambda}]$, we show that the optimal worst-case convergence rate is achieved by only one-tap memory ($M = 1$), and present the analytic formulas for the optimal rate and the corresponding parameters. The technique of using gain margin optimization and $H_\infty$ norm computation are novel in the convergence analysis of MAS.

The rest of this article is organized as follows. Section II introduces some key technical tools and an optimization formulation of the accelerated average consensus problem. Section III presents a necessary and sufficient condition for the average consensus on a given graph with $M$-tap memory. It then derives new analytical solutions for the optimal convergence rate and the corresponding control parameter and compares them with the existing results. Section IV derives the optimal convergence rate for an uncertain graph set and the corresponding control parameters, and relates these to the results of Section III. Section V uses numerical experiment results to show the validity of the proposed algorithms and their advantages over the existing accelerated schemes. Finally, Section VI concludes this article.

### II. PRELIMINARIES AND PROBLEM FORMULATION

In this section, we briefly review some basic results from the stability theory to be used in later sections. Then, we present the consensus algorithm accelerated by node memory and formulate the problem of convergence rate optimization.

#### A. Stability Criterion and Gain Margin Optimization

Throughout this article, the variables $s$ and $z$ in transfer functions or polynomials indicate the continuous-time system and the discrete-time system, respectively. Denote $D(0, r) = \{z : |z| \leq r\}$ as the closed circular disc with center at the origin and radius $r$ in the complex plane.

A continuous-time linear time-invariant (LTI) system with transfer function $G(s)$ is said to be stable if all its poles are on the left half of the (complex) $s$-plane with negative real parts. A discrete-time LTI system with transfer function $G(z)$ is said to be stable if all its poles are within $D(0, 1)$ of the (complex) $z$-plane. Since the poles of a transfer function are the roots of its denominator polynomial, we can define the stability of polynomials.

**Definition 1:** A continuous-time polynomial $H(s)$ is said to be stable if all its roots have negative real parts. Accordingly, a discrete-time polynomial $H(z)$ is said to be stable if all its roots are within $D(0, 1)$.

Routh’s stability criterion is a well known technical tool to determine the stability without solving the roots of a polynomial. As we only need to deal with a polynomial of third order in this article, we list the result of the third-order polynomials in Table I as follows for clarity and simplicity. More general results can be found in [39].

**Lemma 1 (Routh’s stability criterion):** $H(s) = a_0 s^3 + a_1 s^2 + a_2 s + a_3$ is stable if and only if all elements in the first column of Table I (Routh Table) are positive.

Consider the continuous-time interval polynomial $H(s) = \sum_{k=0}^{n} a_k s^k$, $a_k \in [\underline{a}_k, \bar{a}_k]$. There are $2^n + 1$ corner polynomials whose coefficients are either $\underline{a}_k$ or $\bar{a}_k$, $k = 0, \ldots, n$. Intuitively, one should at least check all $2^n + 1$ corner polynomials to determine the stability of the interval polynomial $H(s)$. According to the Kharitonov stability theorem [40], [41], a necessary and sufficient condition for $H(s)$ being stable is that only four special corner polynomials are stable. However, this theorem does not hold for discrete-time systems. In fact, even the stability of all $2^n + 1$ corner polynomials is not sufficient to guarantee the stability of the whole set of discrete-time interval polynomials. Generally, for the discrete-time interval polynomials with degree greater than three, their stability condition becomes very complex.
complex and there is not a counterpart of Kharitonov stability theorem. Nevertheless, for the degree less than or equal to three, the following Lemma holds \cite{40,42}.

**Lemma 2:** Consider the interval polynomial \( H(z) = \sum_{k=0}^{n} a_k z^k \), \( a_k \in [\tilde{a}_k, \tilde{a}_k] \), \( n \leq 3 \). A necessary and sufficient condition for its stability is that all \( 2^{n+1} \) corner polynomials are stable.

The \( H_\infty \) norm of a stable system \( G(s) \) is defined as \( \|G(s)\|_\infty := \text{sup}_{\omega} |G(j\omega)| \). Let \( P(s) \) be the nominal plant and \( P_k(s) = kP(s) \), the gain margin optimization problem is to find the largest \( \tilde{k} \) such that there exists a controller \( C(s) \) achieving internal stability for every \( P_k(s) \) with \( 1 \leq k \leq \tilde{k} \). Denote the largest \( \tilde{k} \) as \( k_{\text{sup}} = \sup \tilde{k} \) and call it the optimal gain margin. \( k_{\text{sup}} \) can be computed directly from

\[
\gamma_{\text{inf}} := \inf_{C(s)} \|T(s)\|_\infty
\]

where \( T(s) = \frac{P(s)C(s)}{1+P(s)C(s)} \). Next we summarize the method that directly computes \( \gamma_{\text{inf}} \) and \( k_{\text{sup}} \) for given \( P(s) \) \cite{43,Ch. 11}.

**Lemma 3:** Let \( P(s) = \frac{U(s)}{V(s)} \) be a coprime factorization of \( P(s) \) satisfying \( U(s)Y_k(s) + V(s)Y_1(s) = 1 \), where \( U(s), V(s), Y_k(s), \) and \( Y_1(s) \) are stable transfer functions. Let \( c_i, i = 1, \ldots, n \) be zeros of \( U(s)V(s) \) in the right half plane, and \( b_i = U(c_i)Y_k(c_i) \), \( i = 1, \ldots, n \). Define

\[
B_1 = \left( \frac{1}{c_i + c_j} \right)_{i,j}, B_2 = \left( \frac{b_i b_j}{c_i + c_j} \right)_{i,j}
\]

where \( c_j \) and \( b_j \) are the complex conjugate of \( c_j \) and \( b_j \), \( i, j = 1, \ldots, n \). Then, the following statements hold. (i) \( \gamma_{\text{inf}} = \sqrt{\lambda(B_1^{-1}B_2)} \) where \( \lambda(B_1^{-1}B_2) \) is the largest eigenvalue of \( B_1^{-1}B_2 \). (ii) If \( P(s) \) is stable or minimum phase, then \( k_{\text{sup}} = \infty \). Otherwise, \( k_{\text{sup}} = \left( \frac{\gamma_{\text{inf}} + \gamma_{\text{inf}}^2}{\gamma_{\text{inf}}^2} \right)^2 \).

**B. Accelerated Average Consensus**

Consider a set of \( N \) agents communicating information through a network described by an undirected graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}) \), where \( \mathcal{V} = \{v_1, v_2, \ldots, v_N\} \) is the set of vertices, \( \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \) is the set of edges, and \( \mathcal{A} = \{a_{ij}\} \) is the adjacency matrix, with \( a_{ii} = 0 \) and \( a_{ij} = a_{ji} > 0 \) if and only if \( (v_i, v_j) \in \mathcal{E} \).

An MAS on \( \mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}) \) with \( M \)-tap memory is in the form of (1). Defining the last two terms in (1) as a control signal, we can rewrite (1) as

\[
x_i(k+1) = x_i(k) + u_i(k)
\]

\[
u_i(k) = \alpha \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)) + \sum_{m=0}^{M} \theta_m x_i(k-m)
\]

where \( x_i(k) \) and \( u_i(k) \) are, respectively, the state and the control signal of agent \( i, i = 1, \ldots, N \), and \( \alpha, \theta_0, \theta_1, \ldots, \theta_M \) are the parameters to be designed. As seen from (4), each agent updates its state by using the current states of itself and its neighbors and its own past states stored in memory. The initial values of each agent are set as

\[
x_i(-M) = \cdots = x_i(-1) = x_i(0), i = 1, \ldots, N.
\]

**Definition 2:** The average consensus of the MAS (3) and (4) is said to be reached asymptotically if for any initial states \( x_i(0) \), \( \lim_{k \to \infty} x_i(k) = \bar{x} \) when the MAS achieves average consensus. It then follows from (3) that \( \lim_{k \to \infty} x_i(k+1) = \lim_{k \to \infty} x_i(k+1)(k) + u_i(k)) = \bar{x} \), which implies \( \lim_{k \to \infty} u_i(k) = 0 \). Furthermore, from (4) and \( \lim_{k \to \infty} x_i(k+1) = \lim_{k \to \infty} x_i(k) = \bar{x} \), there must be \( \lim_{k \to \infty} u_i(k) = \sum_{m=0}^{M} \theta_m \).

The degree of each vertex \( v_i \) is represented by \( d_i = \sum_{j=1}^{N} a_{ij} \). The Laplacian matrix of \( \mathcal{G} \) is defined as \( \mathcal{L} = \mathcal{D} - \mathcal{A} \), where \( \mathcal{D} := \text{diag}(d_1, \ldots, d_N) \) is the degree matrix. It is obvious that the Laplacian matrix \( \mathcal{L} \) is positive semidefinite, and all the eigenvalues are real and can be written in ascending order as \( 0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N \leq 2d \), where \( d = \max \{d_i\} \) is the maximum degree of the graph.

Denote \( x(k) = [x_1(k), x_2(k), \ldots, x_N(k)]^T \in \mathbb{R}^N \). By direct algebraic manipulation, the whole system can be written as

\[
x(k+1) = ((1 + \theta_0)I - \alpha\mathcal{L})x(k) + \sum_{m=1}^{M} \theta_m x(k-m)
\]

with the initial values (6) are \( x(-M) = \cdots = x(-1) = x(0) \).

Denote \( X(k) = [x(k)^T, x(k-1)^T, \ldots, x(k-M)^T]^T \in \mathbb{R}^N(M+1) \). Then, (6) becomes

\[
X(k+1) = \Phi_M X(k)
\]

where

\[
\Phi_M = \begin{bmatrix}
(1 + \theta_0)I - \alpha\mathcal{L} & \theta_1 I & \cdots & \theta_{M-1} I & \theta_M I \\
I & 0 & \cdots & 0 & 0 \\
0 & I & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & I & 0
\end{bmatrix}
\in \mathbb{R}^{N(M+1) \times N(M+1)}
\]

and the initial value in (7) is \( X(0) = [x(0)^T, x(0)^T, \ldots, x(0)^T]^T \).

The compact form (7) is essentially the same as that of \cite{33}, and the models in \cite{31} and \cite{32} can all be viewed as special cases of (7).

**Lemma 5:** If \( \sum_{m=0}^{M} \theta_m = 0 \) and \( \mathcal{G} \) is connected, define

\[
\theta_m = \frac{\theta_m}{1 - \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \theta_{ij}}, m = 0, 1, \ldots, M - 1.
\]

Then, \( \bar{x} \) is a simple eigenvalue of \( \Phi_M \) with the corresponding left eigenvector \( \varphi_1 \) and right eigenvector \( \bar{1}^T_{N(M+1)} \), where

\[
\varphi_1 = \begin{bmatrix}
\frac{1}{N} \sum_{j=0}^{M-1} \frac{\varphi_0}{N} \bar{1}^T_{N(M+1)} - \frac{\theta_0 + \theta_1}{N} \bar{1}^T_{N(M+1)} - \cdots - \frac{\theta_{M-1} + \theta_M}{N} \bar{1}^T_{N(M+1)}
\end{bmatrix}
\]
\[ \bar{T}_N(M+1) \text{ and } \bar{T}_N \text{ are all 1 vectors of dimensions } N(M + 1) \text{ and } N, \text{ respectively.} \]

**Corollary 1:** For system (7) with initial vector \( X(0) = [x(0)^T, \ldots, x(0)^T]^T \in \mathbb{R}^{N(M+1)}, \) \( \lim_{k \to \infty} X(k) = \bar{T}_N(M+1) \) holds for arbitrary \( x(0) \in \mathbb{R}^N \) if and only if all eigenvalues of \( \Phi_M \) are within the unit circle except its simple eigenvalue 1.

The proofs of Lemma 5 and Corollary 1 are given in Appendices A and B.

Similar to the definition in [7], [32], [33], and [35], the convergence rate of the consensus of system (7) is defined as

\[ \gamma_M := \rho_s(\Phi_M) \quad (10) \]

where \( \rho_s(\Phi_M) \) is the second largest eigenvalue modulus of the matrix \( \Phi_M. \)

**Remark 1:** Let \( e(k) = x(k) - \bar{x} \) be the consensus error. Obviously \( e(k) \) is bounded by \( \gamma_M, \) that is, \( \|e(k)\| = O(\gamma_M^k) \) for \( k \) large enough. In terms of the control theory, \( \gamma_M \) determines the settling time of a system. The smaller the convergence rate \( \gamma_M \) is, the faster the error \( e(k) \) converges.

Define \( \Theta_M := [\theta_0, \theta_1, \ldots, \theta_M]. \) The goal of acceleration algorithm design is to find the parameters \( \alpha \) and \( \Theta_M \) such that the convergence rate \( \gamma_M \) is as small as possible. Denote \( \gamma_M, \alpha^* \) and \( \Theta_M^* \) the optimal convergence rate and the corresponding optimal parameters, respectively. The algorithm design problem can be cast into the following optimization problem:

\[ \gamma_M^* = \min_{\{\alpha, \Theta_M\}} \gamma_M = \min_{\{\alpha, \Theta_M\}} \rho_s(\Phi_M). \quad (11) \]

Most of current results are presented based on the property of \( \Phi_M. \) However, the dimension of \( \Phi_M \) is proportional to the number of network nodes, which may be very large and difficult to analyze. Moreover, it is difficult to derive the analytic formulas of the parameters from \( \Phi_M. \)

### III. Optimization of the Convergence Rate and Some Analytical Solutions

In this section, we address the consensus acceleration for the MAS on a given graph with \( M \)-tap memory, as described by (3) and (4). We first compute the characteristic polynomial of \( \Phi_M \) and convert the optimization problem (11) to an equivalent one, which is to minimize the maximum modulus root of \( N \) polynomials. Then, we derive, for \( M \leq 2, \) the analytical formulas of the optimal convergence rate \( \gamma_M^* \) and the corresponding optimal parameters, followed by the comparison with existing results.

For a connected graph \( G, \) recall that \( 0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_N \) are the eigenvalues of its Laplacian matrix \( L. \) Define the polynomials as follows:

\[ h_1(z) = z^M - \theta_0 z^{M-1} - (\theta_0 + \theta_1) z^{M-2} - \cdots - \sum_{m=0}^{M-1} \theta_m \quad (12) \]

\[ h(z, \lambda) = z^M + 1 - (1 + \theta_0 - \alpha \lambda) z^M - \sum_{m=1}^M \theta_m z^{M-m} \quad (13) \]

where \( \alpha \) and \( \theta_m, m = 0, 1, \ldots, M, \) are parameters in (7). As \( \sum_{m=0}^M \theta_m = 0, \) it can be derived that \( h(z, \lambda) = (z - 1) h_1(z) + \alpha \lambda. \) For a polynomial \( h(z), \) denote \( \bar{r}(h(z)) \) its maximum modulus root, i.e.,

\[ \bar{r}(h(z)) = \max \{|z_i| : h(z_i) = 0\}. \quad (14) \]

The following key lemma shows the relation between the polynomials (12) and (13) and the characteristic polynomial of \( \Phi_M. \) The proof is given in Appendix C.

**Lemma 6:** Let \( \Phi_M \) be defined by (8) with a connected graph \( G \) and \( \sum_{m=0}^M \theta_m = 0. \) The characteristic polynomial of \( \Phi_M \) is given by

\[ \det(zI - \Phi_M) = (z-1) h_1(z) \prod_{i=2}^N h(z, \lambda_i) \quad (15) \]

with \( h_1(z), h(z, \lambda) \) defined in (12) and (13).

The following results are immediate by combining Corollary 1 and (10) with Lemma 6.

**Theorem 1:** Consider system (3) and (4) on a connected graph \( G \) with \( \sum_{m=0}^M \theta_m = 0. \) Let \( h_1(z), h(z, \lambda) \) be defined by (12) and (13). Then, we have the following:

(i) the average consensus is reached asymptotically if and only if \( h_1(z) \) and \( h(z, \lambda_i), i = 2, \ldots, N, \) are stable, i.e., \( \bar{r}(h_1(z)) < 1 \) and \( \bar{r}(h(z, \lambda_i)) < 1, i = 2, \ldots, N; \)

(ii) the convergence rate \( \gamma_M \) in (10) can be computed by

\[ \gamma_M = \max_{i=2, \ldots, N} \{\bar{r}(h_1(z)), \bar{r}(h(z, \lambda_i))\} \]

where \( \bar{r}(\cdot) \) is defined by (14).

Theorem 1 establishes the direct link between the convergence rate and the roots of \( N \) polynomials \( h_1(z) \) and \( h(z, \lambda_i), i = 2, \ldots, N. \) Thus, the optimization problem of the accelerated consensus can be rewritten as

\[ \gamma_M^* = \min_{\{\alpha, \Theta_M\}} \gamma_M = \min_{\{\alpha, \Theta_M\}} \max_{i=2, \ldots, N} \{\bar{r}(h_1(z)), \bar{r}(h(z, \lambda_i))\}. \quad (16) \]

Hence, the convergence of system (7) with order \( N(M+1) \) is determined by \( N \) polynomials of orders no greater than \( M + 1. \) When the length of memory is much less than the number of agents, i.e., \( M \ll N, \) the analysis and computation can be greatly simplified.

#### A. Explicit Formula of the Optimal Convergence Rate for MAS With Memory of \( M \leq 2 \)

In this section, we derive the analytic formulas for the fastest convergence rate and the corresponding optimal parameters when \( M \leq 2. \)

Notice that the polynomials \( h(z, \lambda_i), i = 2, \ldots, N, \) are a subset of interval polynomials, with the coefficient of the term \( z^M \) being within \([\alpha \lambda_2 - 1 - \theta_0, \alpha \lambda_N - 1 - \theta_0]. \) For \( M \leq 2, \) it follows from Lemma 2 that the stability of the \( N - 1 \) polynomials \( h(z, \lambda_i), i = 2, \ldots, N, \) is determined by the corner polynomials, which in this case are \( h(z, \lambda_2) \) and \( h(z, \lambda_N). \) Hence, we have the following result.

**Theorem 2:** Consider system (3) and (4) on a connected graph \( G \) with \( M \leq 2 \) and \( \sum_{m=0}^M \theta_m = 0. \) Then, we have the following:
i) the average consensus is reached asymptotically if and only if \( h_1(z), h(z, \lambda_2) \) and \( h(z, \lambda_N) \) are stable;

ii) the optimal convergence rate \( \gamma_M^* \) defined in (16) can be simplified to

\[
\gamma_M^* = \min_{\{\alpha, \Theta_M\}} \gamma_M
= \min_{\{\alpha, \Theta_M\}} \max_{j \in N_i} \{ \bar{r}(h_1(z)), \bar{r}(h(z, \lambda_2)), \bar{r}(h(z, \lambda_N)) \}
\]  

(17)

where \( \bar{r}(\cdot) \) is defined by (14).

In the following, we will use Theorem 2 to derive the analytic formulas for the optimal convergence rate \( \gamma_M^* \) and the corresponding optimal parameters \( \alpha^* \) and \( \Theta_M^* \). The main idea is as follows. As seen from (17), MAS (3) and (4) achieves consensus with convergence rate \( \gamma_M \) if and only if all roots of three polynomials \( h_1(z), h(z, \lambda_2), \) and \( h(z, \lambda_N) \) are within the circular disc \( D(0, \gamma_M) \). Note that the roots of a polynomial \( h(z) \) are within \( D(0, r) \) if and only if the roots of \( h(rz) \) are within the unit disc \( D(0, 1) \), i.e. the discrete-time polynomial \( h(rz) \) is stable. Using the bilinear transformation \( z = \frac{s + 1}{s - 1} \), the stability of the discrete-time polynomial \( h(rz) \) is transformed to the stability of a corresponding continuous-time polynomial \( f(s) \). Then, the Routh criterion can be utilized to formulate the optimization problem. Finally, we obtain the explicit formulas via subtle algebraic manipulations.

The controller with two-tap memory \( M = 2 \) is in the form

\[
u_i(k) = \alpha \sum_{j \in N_i} a_{ij} x_j(k) - x_i(k) + \theta_0 x_i(k) + \theta_1 x_i(k - 1) + \theta_2 x_i(k - 2).
\]

(18)

When applied to MAS (3), it yields the closed loop system

\[
x(k + 1) = \left((1 + \theta_0)I - \alpha L\right) x(k) + \theta_1 x(k - 1) + \theta_2 x(k - 2).
\]

(19)

**Theorem 3:** Consider system (19) on a connected graph \( G \) with \( \theta_0 + \theta_1 + \theta_2 = 0 \). Let \( \lambda_2 \) and \( \lambda_N \) be the smallest and the largest nonzero eigenvalues of \( L \), respectively. Then, the optimal convergence rate \( \gamma_2^* \) defined by (17) is given by

\[
\gamma_2^* = \frac{\sqrt{\lambda_N/\lambda_2} - 1}{\sqrt{\lambda_N/\lambda_2} + 1}
\]

(20)

and the corresponding optimal parameters are

\[
\alpha^* = \frac{4}{\left(\sqrt{\lambda_N} + \sqrt{\lambda_2}\right)^2}
\]

(21)

\[
\theta_0^* = \left(\frac{\sqrt{\lambda_N/\lambda_2} - 1}{\sqrt{\lambda_N/\lambda_2} + 1}\right)^2
\]

(22)

\[
\theta_1^* = -\theta_0^*
\]

(23)

\[
\theta_2^* = 0
\]

(24)

**Remark 2:** \( \theta_2^* = 0 \) in (24) implies that the optimal controller for \( M = 2 \) has actually used only one-tap memory, and increasing memory from \( M = 1 \) to \( M = 2 \) has not yielded a faster convergence rate. The complete proof is given in Appendix D.

For \( M = 1 \), the control algorithm is given by

\[
u_i(k) = \alpha \sum_{j \in N_i} a_{ij} x_j(k) - x_i(k) + \theta_0 x_i(k) + \theta_1 x_i(k - 1).
\]

(25)

The resulting closed-loop system is

\[
x(k + 1) = \left((1 + \theta_0)I - \alpha L\right) x(k) + \theta_1 x(k - 1).
\]

(26)

The following result follows directly from Theorem 3.

**Corollary 2:** Consider system (26) on a connected graph \( G \). Let \( \lambda_2 \) and \( \lambda_N \) be the smallest and the largest nonzero eigenvalues of \( L \), respectively. Then, the optimal convergence rate \( \gamma_1^* \) defined by (17) is given by

\[
\gamma_1^* = \frac{\sqrt{\lambda_N/\lambda_2} - 1}{\sqrt{\lambda_N/\lambda_2} + 1}
\]

(27)

and the corresponding optimal parameters are

\[
\alpha^* = \frac{4}{\left(\sqrt{\lambda_N} + \sqrt{\lambda_2}\right)^2}
\]

(28)

\[
\theta_0^* = \left(\frac{\sqrt{\lambda_N/\lambda_2} - 1}{\sqrt{\lambda_N/\lambda_2} + 1}\right)^2
\]

(29)

\[
\theta_1^* = -\theta_0^*
\]

(30)

Remark 3: Corollary 2 is a direct result of Theorem 3. However, without proving Theorem 3, it is impossible to know the results for \( M = 1 \) is the same as that of \( M = 2 \) and \( \theta_2^* = 0 \).

**B. Comparison With Existing Results**

In this section, we compare the performance of our proposed one-tap memory scheme with five accelerated consensus algorithms in the existing literature, which are closely related to our work.

1. The best constant gain scheme (BC-L) proposed in [35]

\[
u_i(k) = \alpha \sum_{j \in N_i} a_{ij} x_j(k) - x_i(k).
\]

(31)

2. The graph filtering scheme with T-periodic sequence (GF-L) proposed in [19]

\[
u_i(k) = \varepsilon(k) \sum_{j \in N_i} a_{ij} x_j(k) - x_i(k).
\]

(32)

3. The one-tap memory scheme (Mem-W) proposed in [31]

\[
u_i(k) = (1 + g) \sum_{j \in N_i} a_{ij} x_j(k) - x_i(k)
+ g x_i(k) - g x_i(k - 1).
\]

(33)

4. The general one-tap memory scheme (GMem-W) proposed in [32]
TABLE II

| Algorithm                  | Optimal control parameters | Convergence rate                  |
|----------------------------|-----------------------------|-----------------------------------|
| BC-L [35]                  | $\gamma^* = \frac{2}{\lambda_N + \lambda_2}$ | $\gamma^*_G = \frac{\lambda_N - \lambda_2}{\lambda_N + \lambda_2}$ |
| GF-L [19]                  | $\gamma^* = \frac{2}{\lambda_N + \lambda_2}$ | $\gamma^*_G = \frac{\lambda_N - \lambda_2}{\lambda_N + \lambda_2}$ |
| Mem-W [31]                 | $\gamma^* = \frac{2}{\lambda_N + \lambda_2}$ | $\gamma^*_G = \frac{\lambda_N - \lambda_2}{\lambda_N + \lambda_2}$ |
| GMem-W [32]                | $\gamma^* = \frac{2}{\lambda_N + \lambda_2}$ | $\gamma^*_G = \frac{\lambda_N - \lambda_2}{\lambda_N + \lambda_2}$ |
| FIRMem-L [37]              | $\gamma^* = \frac{2}{\lambda_N + \lambda_2}$ | $\gamma^*_G = \frac{\lambda_N - \lambda_2}{\lambda_N + \lambda_2}$ |
| OptMem-proposed            | $\gamma^* = \frac{2}{\lambda_N + \lambda_2}$ | $\gamma^*_G = \frac{\lambda_N - \lambda_2}{\lambda_N + \lambda_2}$ |

5) The FIR memory-enhanced scheme (FIRMem-L) proposed in [37]

$$u_i(k) = (1 - g + g \beta_3) \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)) + g(\beta_2 + \beta_3 - 1)x_i(k) + g \beta_1 x_i(k - 1).$$

(34)

For the memoryless schemes (31) and (32), it can be verified that

$$\gamma^*_\text{BC} = \frac{\lambda_N - \lambda_2}{\lambda_N + \lambda_2} > \frac{\sqrt{\lambda_N - \lambda_2}}{\sqrt{\lambda_N + \lambda_2}}$$

and the limitation of $\gamma^*_G$ is

$$\frac{\sqrt{\lambda_N - \lambda_2}}{\sqrt{\lambda_N + \lambda_2}}$$

as $T \to \infty$. Thus, the optimal convergence rate of our proposed one-tap memory scheme is faster than that of the memoryless schemes (31) and (32) proposed in [35] and [19].

For the one-tap memory schemes (33) and (34), we have the following result.

**Corollary 3:** Consider MAS (3) under the control of algorithms (25), (33), and (34), respectively. Then, the optimal convergence rates for these algorithms satisfy

$$\gamma_1^* \leq \gamma^*_\text{Mem} = \gamma^*_\text{GMem}$$

(36)

and the first equality from left holds only when $\lambda_2 + \lambda_N = 2$.

**Proof:** From Table II, it has

$$\gamma^*_\text{Mem} = \frac{(\rho_s(W))^2}{(1 + \sqrt{1 - (\rho_s(W))^2})(1 - \sqrt{1 - (\rho_s(W))^2})}$$

$$= \frac{(\rho_s(W))^2}{1 - (\rho_s(W))^2} = 1.$$
are $\lambda_N = 5, \lambda_2 = 1$. Using controller (4) with $M = 3$, and setting $\alpha = 0.3894, \theta_0 = 0.1681, \theta_1 = -0.1706, \theta_2 = 0$, and $\theta_3 = 0.0024$, we have

$$\Phi_3 = \begin{bmatrix} (1.1681)I - 0.3894L & -0.1706I & 0 & 0.0024I \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \end{bmatrix}.$$  

Then, we can compute that $\gamma_3 = \lambda_3(\Phi_3 - \frac{1}{\bar{\lambda}_N} \Phi_1) = 0.2920$. Clearly $\gamma_3 < \gamma_1^* = \frac{\sqrt{\lambda_3/\lambda_2}}{\sqrt{\lambda_3/\lambda_2+1}} = 0.3820$. Hence, increasing memory from $M = 1$ (or $M = 2$) to $M = 3$ yields an improved convergence rate for the system on the star graph. Table III gives the optimal convergence rates of controller (4) with $M = 0$ to 6 for the same system with $N = 5, 10, 15$, and 20, respectively. As seen from the table

$$\gamma_0 > \gamma_1 > \gamma_2 > \gamma_3 > \gamma_4 > \gamma_5 = \gamma_6.$$  

(37)

Clearly, $M = 1$ is no longer the optimal memory tap for $M \geq 3$, and $\gamma_M^*$ decreases in an interesting pattern as $M$ increases from $M = 3$. To confirm this, we have done hundreds of numerical experiments on different graphs, such as cycle graphs and random graphs. For the same incremental changes of numerical experiments on different graphs, such as cycle graphs and random graphs. For the same incremental changes of $M$, $\gamma_M^*$ always show the same pattern of (37). Based on the observation of extensive numerical examples, we make the following conjecture, but are unable to prove it by now.

Conjecture 1: For the MAS (3) and (4) on a connected graph $G$, increasing memory from $M = 2K - 1$ to $M = 2K$ cannot improve the optimal convergence rate for $K = 1, 2, \ldots$, that is

$$\gamma_{2K}^* = \gamma_{2K-1}^*.$$  

(38)

IV. OPTIMAL WORST-CASE CONVERGENCE RATE ON A SET OF GRAPHS

In this section, we show that the one-tap memory protocol provides the optimal worst-case convergence rate for an uncertain graph set. Our proof depends on the gain margin optimization of robust stability stated in Lemma 3.

Let $\{G\}_{\bar{\lambda}}$ be the set of all connected graphs with $[\lambda_2, \lambda_N] \subseteq [\bar{\lambda}, \bar{\lambda}]$. Consider system (3) and (4) on $\{G\}_{\bar{\lambda}}$ with $\sum_{m=0}^M \theta_m = 0$. The worst-case convergence rate is defined as

$$\gamma_M = \sup_{\theta \in \{G\}_{\bar{\lambda}}} (\Phi_M) = \sup_{\theta \in \{G\}_{\bar{\lambda}}} \max_{\{i=2, \ldots, N\}} \{\bar{r}(h_1(z)), \bar{r}(h(z, \lambda_i))\}.$$  

(39)

The goal is to find the parameters $\alpha$ and $\Theta_M$ such that the worst-case convergence rate is as small as possible, which can be described by the following optimization problem:

$$\min_{\{\alpha, \Theta_M\}} \gamma_M = \min_{\{\alpha, \Theta_M\}} \sup_{\theta \in \{G\}_{\bar{\lambda}}} \max_{\{i=2, \ldots, N\}} \{\bar{r}(h_1(z)), \bar{r}(h(z, \lambda_i))\}.$$  

(40)

Theorem 4: Consider the MAS (3) and (4) on $\{G\}_{\bar{\lambda}}$ with $\sum_{m=0}^M \theta_m = 0$. For any $M \geq 1$, the solutions of (39) are as follows:

$$\gamma_M^* = \frac{\sqrt{\lambda_3/\lambda_2} - 1}{\sqrt{\lambda_3/\lambda_2} + 1}.$$  

(41)

$$\alpha^* = \frac{4}{(\sqrt{\lambda_3} + \sqrt{\lambda_2})^2}.$$  

(42)

$$\theta_m^* = -\theta_0^*.$$  

(43)

Before proving the theorem, we need some notations and a lemma that is proved in Appendix E.

Let $P(z, \lambda) = \lambda \frac{r(z)}{\lambda - r(z)}$ be a plant and $C(z) = \frac{\alpha z^M}{h(z, \lambda)}$ be the negative feedback controller of the plant, as shown in Fig. 1. The transfer function of the closed-loop system is given by

$$T(z, \lambda) = \frac{P(z, \lambda)C(z)}{1 + P(z, \lambda)C(z)} = \frac{\alpha \lambda z^M}{1 + \alpha \lambda z^M}.$$  

(44)

Lemma 7: The optimal gain margin $k_{\text{sup}}$ of $P(rz, \lambda) = \frac{\lambda}{\lambda - rz} - 1$ is given by $k_{\text{sup}} = \left(\frac{1 + \alpha}{1 - \alpha}\right)^2$.

Proof: (Proof of Theorem 4): The optimization problem (39) is equivalent to

$$\min_{\{\alpha, \Theta_M\}} \max_{\lambda \in \{\bar{\lambda}, \bar{\lambda}\}} \left\{\bar{r}(h_1(z)), \bar{r}(h(z, \lambda_i))\right\}.$$  

(45)

Note that all roots of the polynomial $h(z)$ are within the circle $D(0, r)$ if and only if $h(rz)$ is stable. Then, we have $\bar{r}(h(z)) = \frac{\alpha z^M}{h(z, \lambda)}.$  

(46)
inf \{ r : h(rz) \text{ is stable} \}. Hence, (45) can be written as follows:

$$\min_{\{\alpha, \Theta M\}} r$$

s.t. (i) $h_1(rz)$ is stable

(ii) $h(rz, \lambda)$ is stable for any $\lambda \in [\bar{\lambda}, \bar{\lambda}]$.

It follows from the representation of $P(z)$ and $T(z)$ that the abovementioned optimization problem is equivalent to

$$\min_{\{\alpha, \Theta M\}} r$$

s.t. (i) $C(rz)$ is stable

(ii) $T(rz, \lambda)$ is stable for any $\lambda \in [\bar{\lambda}, \bar{\lambda}]$. (46)

$T(rz, \lambda)$ is stable for any $\lambda \in [\bar{\lambda}, \bar{\lambda}]$ means that $T(rz, \lambda)$ is stable and the gain margin is at least $\bar{\lambda}/\bar{\lambda}$. If we relax the stability constraint on $C(rz)$, the problem becomes finding $C(rz)$ that achieves stability for every $P(rz, \lambda)$ in

$$\mathcal{P} = \{ kP(rz, \lambda) : 1 \leq k \leq \bar{\lambda}/\bar{\lambda} \}.$$ 

This means that the optimal gain margin of $P(rz, \lambda)$ should be at least $\bar{\lambda}/\bar{\lambda}$, i.e., $k_{\text{sup}} \geq \bar{\lambda}/\bar{\lambda}$. It follows from Lemma 7 that $k_{\text{sup}} = \left(\frac{1+\bar{r}}{\bar{r}}\right)^2$. Hence, we get $\left(\frac{1+\bar{r}}{\bar{r}}\right)^2 \geq \bar{\lambda}/\bar{\lambda}$, which gives $r \geq \frac{\sqrt{\bar{\lambda}/\bar{\lambda} - 1}}{\sqrt{\bar{\lambda}/\bar{\lambda} + 1}}$. On the other hand, we know from Corollary 2 that the lower bound $r^* = \frac{\sqrt{\bar{\lambda}/\bar{\lambda} - 1}}{\sqrt{\bar{\lambda}/\bar{\lambda} + 1}}$ can be achieved by (28)–(30) when $M = 1$. Then, it is easy to check that the corresponding $C(r^*z)$ is stable.

Theorem 3 shows that for the MAS on any fixed graph, increasing $M$ from 1 to 2 does not further accelerate the convergence of the system. However, there might be an MAS with memory of $M > 2$ on a particular graph (for example, a star graph) that has a faster convergence rate than $\gamma_M^* = \frac{\sqrt{\bar{\lambda}/\bar{\lambda} - 1}}{\sqrt{\bar{\lambda}/\bar{\lambda} + 1}}$. On the other hand, for the MAS on a set of graphs with $[\lambda_2, \lambda_N] \subseteq [\bar{\lambda}, \bar{\lambda}]$, the optimization problem turns out to be (39), which is different to (17) for Theorem 2. Theorem 4 shows that the one-tap memory algorithm is the optimal one in terms of worst-case convergence performance.

The optimal worst-case convergence rate $\gamma_M^*$ in (40) actually does not depend on $M$. The key to obtain such result is to build the relation between the convergence rate and the optimal gain margin. Recall that in the definition of the optimal gain margin, the controller achieving $k_{\text{sup}}$ is not necessarily stable. With Theorem 4, we know that the controller $C(r^*z) = \frac{\alpha}{1+r^*z}$ stabilizes all the plants in $\mathcal{P} = \{ kP(r^*z, \lambda) : 1 \leq k \leq \bar{\lambda}/\bar{\lambda} \}$ and is stable itself.

Remark 4: The optimal worst-case convergence rate $\gamma_M^*$ is closely related to Nesterov’s lower complexity bound $\frac{\sqrt{Q_f - 1}}{\sqrt{Q_f + 1}}$, where $Q_f$ is the condition number of the cost function defined in [26]. We would like to point out that the set of cost functions in Nesterov’s case is much larger than the underlying objective function of our average consensus problem. As discussed in Section I, our cost function is a very special case of the general strongly convex functions in [26]. Intuitively, the rate $\gamma_M^*$ should be smaller (better) than $\frac{\sqrt{Q_f - 1}}{\sqrt{Q_f + 1}}$, because our worse performance is defined on a much smaller function set. In this sense, our worst-case convergence rate in Theorem 4 is stronger than Nesterov’s lower complexity bound. This is demonstrated in numerical examples section by comparing our algorithm with the algorithms of [27], which are based on the general optimization results of [26].

Remark 5: For the MAS on the fixed network with a large number of agents, the eigenvalues $\lambda_i$, $i = 2, \ldots, N$, usually distributed densely in $[\lambda_2, \lambda_N]$. If this is the case, one may expect from Theorem 4 that the one-tap memory controller (25) with parameters given by Theorem 3 presents the optimal convergence rate. Roughly speaking, the one-tap memory controller with convergence rate $\gamma_1^*$ is a very good choice for any MAS systems. Although we consider the MAS on a set of graphs, the control protocol is still time-invariant because the topology is fixed. The analysis here may not be directly applicable to dynamic networks.

### V. NUMERICAL EXAMPLES

This section presents simulation and numerical experiments to show the effectiveness of the theoretical results. Experiments have been designed to compare our proposed algorithm with the five algorithms theoretically compared in Section III-B, i.e., BC-L, GF-L, Mem-W, GMem-W, and FIRMem-L algorithms. Consider an MAS with eight agents on six unweighted sample graphs:

- a) the cycle graph $G_1$;
- b) the path graph $G_2$;
- c) the star graph $G_3$;
- d) the complete bipartite graph $G_4$ with 3 + 5 vertices;
- e) the graph $G_5$ generated by a small-world network model shown in Fig. 2(a);
- f) the graph $G_6$ generated by a BA scale-free network model shown in Fig. 2(b).

For unweighted graphs, the doubly stochastic weight matrices can be chosen as $W = I - \frac{1}{\lambda}L$.

Let the consensus error be defined as $\frac{\| x(t) - x_{12} \|_2}{\| x(t) - x_{12} \|_2}$, and the tolerable error be $\epsilon = 10^{-4}$. We now investigate the consensus performances of the abovementioned algorithms on six sample graphs.

#### A. Cycle Graph $G_1$

The eigenratio of the graph Laplacian $L_{G_1}$ is $\frac{\lambda_2}{\lambda_N} = 0.1464$, and the second largest eigenvalue of the weight matrix $W_{G_1}$ is
\( \rho_s(W_{G_1}) = 1 \). It can be derived that \( \gamma_{\text{Mem}}^* = 1 \) and \( \gamma_{\text{GMem}}^* = 1 \), which means the algorithms proposed in [31] and [32] cannot reach consensus. The consensus error trajectories of MAS by different algorithms on \( G_1 \) are shown in Fig. 3(a). It can be seen that the MAS under the BC-L scheme in [35], the GF-L scheme in [19], the FIRMem-L scheme in [37], and our proposed method can achieve consensus. It appears that our proposed method has the fastest convergence rate.

**B. Path Graph \( G_2 \)**

The eigenratio of the graph Laplacian \( \mathcal{L}_{G_2} \) is \( \frac{\lambda_2}{\lambda_N} = 0.0396 \), and the second largest eigenvalue of the weight matrix \( W_{G_2} \) is \( \rho_s(W_{G_2}) = 0.9239 \). The consensus error trajectories of MAS by different algorithms on \( G_2 \) are shown in Fig. 3(b). It can be seen that the MAS under the six discussed algorithms can achieve consensus, and the consensus performance by the Mem-W scheme in [31], GMem-W scheme in [32], and our proposed method are the same with the fastest convergence rate.

**C. Star Graph \( G_3 \)**

The eigenratio of the graph Laplacian \( \mathcal{L}_{G_3} \) is \( \frac{\lambda_2}{\lambda_N} = 0.1250 \), and the second largest eigenvalue of the weight matrix \( W_{G_3} \) is \( \rho_s(W_{G_3}) = 0.8571 \). The consensus error trajectories of MAS...
by different algorithms on $G_5$ are shown in Fig. 3(c). It can be seen that the memory-enhanced algorithms have the faster convergence rates than the BC-L scheme in [35], and our proposed method has the fastest convergence rate. Moreover, the FIRMem-L scheme in [37] is the worst among the memory-enhanced algorithms. This shows that the faster convergence rates can be achieved by using one-tap memory of each agent itself, and our proposed method has the fastest convergence rate.

**D. Complete Bipartite Graph $G_4$**

The eigenratio of the graph Laplacian $L_{G_4}$ is $\lambda_{N} = 0.3750$, and the second largest eigenvalue of the weight matrix $W_{G_4}$ is $\rho_s(W_{G_4}) = 0.6000$. The consensus error trajectories of MAS by different algorithms on $G_4$ are shown in Fig. 3(d). Although the convergence rate of the FIRMem-L scheme in [37] is faster than that of the Mem-W scheme in [31] and the GMem-W scheme in [32], it is still slower than the optimal scheme proposed in this article.

**E. Graph $G_5$ Generated by a Small-World Network Model**

The eigenratio of the graph Laplacian $L_{G_5}$ is $\lambda_{N} = 0.2201$, and the second largest eigenvalue of the weight matrix $W_{G_5}$ is $\rho_s(W_{G_5}) = 0.7066$. The consensus error trajectories of MAS by different algorithms on $G_5$ are shown in Fig. 3(e). It can be seen that the convergence rate of the GF-L scheme in [19] is faster than that of the FIRMem-L scheme in [37], and our proposed method has the fastest convergence rate.

**F. Graph $G_6$ Generated by a BA Scale-Free Network Model**

The eigenratio of the graph Laplacian $L_{G_6}$ is $\lambda_{N} = 0.1583$, and the second largest eigenvalue of the weight matrix $W_{G_6}$ is $\rho_s(W_{G_6}) = 0.8$. The consensus error trajectories of MAS by different algorithms on $G_6$ are shown in Fig. 3(f). It can be seen that the optimal one-tap memory scheme proposed in this article always has the fastest convergence rate.

Table IV gives the exact values of the convergence rates on the six sampled graphs by the six discussed consensus algorithms. Compared to existing algorithms, the optimal algorithm proposed in this article always has the fastest convergence rate, which corroborates the analysis in Section III-B.

It can be seen that the convergence rates of the Mem-W scheme in [31] and the GMem-W scheme in [32] are both determined by the construction of the weight matrix $W$. If $W$ is improperly chosen, the MAS will diverge, such as the cycle graph $G_1$. If $W$ is chosen appropriately, the convergence rates of the Mem-W scheme in [31] and the GMem-W scheme in [32] are the same as that of our proposed optimal scheme, such as the path graph $G_2$. We further discuss three different choices of the weight matrix $W$. First is the Metropolis–Hastings weight matrix [4] defined as $W^{mh} = [w^{mh}_{ij}]$, where

$$w^{mh}_{ij} = \begin{cases} 1/(i,j) \in \mathcal{E} \\ 1 - \sum_{j \in \mathcal{N}_i} w_{ij} \\ 0 \text{ else.} \end{cases}$$

Second is the Lazy Metropolis–Hastings weight matrix [36] defined as $W^{lmh} = 2^{-1}I + 2^{-1}W^{mh}$. Third is the equal neighbor weight matrix [34] defined as $W^{en} = [w^{en}_{ij}]$, where

$$w^{en}_{ij} = \begin{cases} \frac{1}{d_{i-1}} \quad (i,j) \in \mathcal{E} \\ 0 \text{ else.} \end{cases}$$

For the weighted graphs, the Laplacian matrix can be chosen as $L = I - W$. Table V shows the exact values of the convergence rates of the Mem-W scheme in [31], the GMem-W scheme in [32] and our proposed method on the cycle graph $G_1$ and the small-world graph $G_5$ with different $W$. It can be seen that the convergence rates of the Mem-W scheme in [31] and the GMem-W scheme in [32] can be further improved by choosing

| Table IV | Convergence Rates by the Discussed Algorithms on Six Sample Graphs |
|----------|---------------------------------|
| $G_1$ Cycle graph $G_2$ | $G_3$ Star graph $G_4$ | $G_5$ Bipartite graph $G_6$ | $G_5$ SW graph $G_5$ | $G_6$ BA graph $G_6$ |
| $\frac{\lambda_{N}}{\lambda_N}$ | $\rho_s(W)$ | $\gamma_{BC}$ | $\gamma_{GP}$ | $\gamma_{Mem}$ | $\gamma_{GMem}$ | $\gamma_{FIRMem}$ | $\gamma_{OptMem}$ |
| 0.1464 | 0.0396 | 0.1250 | 0.3750 | 0.2201 | 0.1583 |
| 0.7445 | 0.5610 | diverse | diverse | 0.5930 | 0.4465 |

| Table V | Convergence Rates by Mem-W [31], GMem-W [32], and OptMem-Proposed With Different W |
|----------|----------------------------------|
| $\gamma_{Mem}$ | $\gamma_{GMem}$ | $\gamma_{OptMem}$ |
| $w^{mh}_{ij}$ | $w^{lmh}_{ij}$ | $w^{en}_{ij}$ |
| $G_1$ Cycle graph $G_1$ | $G_2$ Cycle graph $G_1$ | $G_3$ Cycle graph $G_1$ | $G_4$ SW graph $G_5$ | $G_5$ SW graph $G_5$ | $G_6$ SW graph $G_6$ |
| diverse | diverse | diverse | 0.5612 | 0.5050 | 0.4259 |
| 0.5050 | 0.4259 | 0.3863 | 0.3863 | 0.3056 |
the weight matrix appropriately, but they will never be faster than the optimal one-tap memory algorithm proposed in this article.

Next, we compare our proposed optimal one-tap memory scheme with the single-step dual accelerated method and the multistep dual accelerated method proposed in [27], which are based on the results of the Nesterov’s accelerated first-order gradient algorithms for solving the general convex optimization problems. Fig. 4(a)-(f) shows the convergence rates by the three algorithms on six unweighted sample graphs. As seen from the figure, the optimal one-tap memory algorithm proposed in this article always has the fastest convergence rate compared with the two optimization algorithms of [27].

VI. Conclusion

We have introduced the Kharitonov stability theorem, the Routh stability criterion, and the robust stability margin for the analysis and design of the multiagent average consensus with local memory. Using these techniques, we have presented a necessary and sufficient condition for the average consensus of MASs with M-tap memory, and revealed the direct link between the convergence rate and the control protocols. For M ≤ 2, we have shown that the convergence rate equals the maximum modulus root of only three polynomials for any N agents. And the analytical formulas of the optimal convergence rate and the corresponding optimal parameters have been derived. We have also compared the existing results with the optimal ones presented in this article. For MASs with M ≥ 1 on a set of uncertain graphs, we have shown that the acceleration with M = 1 presents the optimal convergence rate in the sense of worst-case performance by using gain margin optimization of robust stability. Numerical experiments have demonstrated the validity, effectiveness, and advantages of these results and methods.

These new results have provided deeper insight into the MAS average consensus with local memory and effective tools for the design and quantitative performance evaluation of the consensus protocols.

The analysis techniques of this article can also be applied to directed graphs as long as the eigenvalues of the Laplacian matrix are real. For such graphs, the characteristic polynomial of Φ_M in (15), Theorem 1 and the resulting analysis remain unchanged. Since the average consensus can be viewed as a special case of the distributed optimization with objective function \( \sum_{i=1}^{N} (x_i - x_i(0))^2 \), we are currently working on extending the presented results to acceleration algorithms of general distributed optimization problems.

APPENDIX

A. Proof of Lemma 5

As \( \sum_{m=0}^{M} \theta_m = 0 \), it is easy to verify that \( \Phi_M \tilde{I}_{N(M+1)} = \tilde{I}_{N(M+1)} \), where \( \tilde{I}_{N(M+1)} \) is an all 1 vector of dimension \( N(M+1) \). Thus, \( \tilde{I}_{N(M+1)} \) is the right eigenvector corresponding to 1. If \( G \) is connected, then 0 is the simple eigenvalue of the graph Laplacian matrix \( L \), and 1 is a simple eigenvalue of \( \Phi_M \). Let \( \varphi_1 = [\varphi_{11}, \varphi_{12}, \ldots, \varphi_{1(M+1)}] \in \mathbb{R}^{1 \times N(M+1)} \), be the left eigenvector corresponding to 1, where \( \varphi_{1i} \in \mathbb{R} \), \( i = 1, \ldots, M + 1 \), it has \( \varphi_2 \Phi_M = \varphi_1 \). Then, we have

\[
\begin{cases}
(1 + \theta_0)\varphi_{11} - \alpha \mathcal{L} \varphi_{11} + \varphi_{12} = \varphi_{11} \\
\theta_1 \varphi_{11} + \varphi_{13} = \varphi_{12} \\
\vdots \\
\theta_{M-1} \varphi_{11} + \varphi_{1(M+1)} = \varphi_{1M} \\
\theta_M \varphi_{11} = \varphi_{1(M+1)}. 
\end{cases}
\]

(47)

One solution of the abovementioned linear equations can be obtained as \( \varphi_{11} = \frac{1}{\rho} \tilde{I}_{N}, \varphi_{12} = -\theta_0 \frac{1}{\rho^2} \tilde{I}_{N}, \varphi_{13} = (-\theta_0 + \theta_1) \frac{1}{\rho^2} \tilde{I}_{N}, \ldots, \varphi_{1(M+1)} = -\sum_{m=0}^{M-1} \theta_m \frac{1}{\rho^2} \tilde{I}_{N} \), where \( \tilde{I}_{N} \) is the all 1 vector of dimension \( N \). As \( \varphi_1 \tilde{I}_{N(M+1)} = c \sum_{i=1}^{M+1} \varphi_{1i} \tilde{I}_{N} = 1 \), it has \( c = \frac{1}{1-c \sum_{i=0}^{M} \theta_{i}} \). Thus, the explicit formula of \( \varphi_1 \) can be derived as shown in (9).

B. Proof of Corollary 1

1) Sufficiency: Assume that \( \Phi_M \) has a simple eigenvalue equal to 1, and the corresponding right eigenvector is \( u_1 = \tilde{I}_{N(M+1)} \). Then, \( \Phi_M \) can be written in a Jordan canonical form as

\[
\Phi_M = V \begin{bmatrix} 1 & \alpha \\ \alpha J & O \end{bmatrix} V^{-1}
\]

(48)

where \( V = [v_1, \ldots, v_{N(M+1)}] \), and the Jordan block matrix \( J \in \mathbb{R}^{(N(M+1)-1) \times (N(M+1)-1)} \) corresponding to the eigenvalues of \( \Phi_M \) within the unit circle. Therefore, one can obtain

\[
\lim_{k \to \infty} \Phi_M^k = V \begin{bmatrix} 1 & \alpha \\ \alpha J & O_{N(M+1)-1} \end{bmatrix} V^{-1}.
\]

(49)

Then, the consensus state of (7) are given by

\[
\lim_{k \to \infty} X(k) = v_1 \varphi_1 X(0) = \frac{1}{N} \sum_{i=1}^{N} x_i(0) \tilde{I}_{N} = \bar{x} \tilde{I}_{NM}
\]

(50)

that is, \( \lim_{k \to \infty} x_i(k) = \bar{x} \). Then, the MAS (6) reaches average consensus asymptotically.

2) Necessity: If the MAS achieves average consensus, i.e., \( \lim_{k \to \infty} x_i(k) = \lim_{k \to \infty} x_j(k) = \bar{x} \), it has \( 0 = \lim_{k \to \infty} u_i(k) = \sum_{m=0}^{M} \theta_m \lim_{k \to \infty} x_i(k - m) = \bar{x} \sum_{m=0}^{M} \theta_m \). Hence, \( \sum_{m=0}^{M} \theta_m = 0 \). Set \( X(0) = \tilde{I}_{N(M+1)} \), it has \( \Phi_M X(0) = \tilde{I}_{N(M+1)} \). Thus, the matrix \( \Phi_M \) has at least one eigenvalue equal to 1. If the MAS can reach average consensus, that is \( |X(k) - \bar{x} \tilde{I}_{N+1}| \to 0 \) as \( k \to \infty \). Then it implies that \( \Phi_M^k \) has rank one as \( k \to \infty \), which in turn implies that \( J^k \) equal to a zero matrix as \( k \to \infty \). It follows that all eigenvalues of \( \Phi_M \) are within the unit circle except that 1 is a simple eigenvalue.

C. Proof of Lemma 6

To prove Lemma 6, we first introduce Schur’s Formula [44].

For a matrix \( M = \begin{bmatrix} R & S \\ P & Q \end{bmatrix} \), with \( R \in \mathbb{R}^{n \times n}, S \in \mathbb{R}^{n \times m}, P \in \mathbb{R}^{m \times n}, Q \in \mathbb{R}^{m \times m} \), and nonsingular,

\[
\det \{ M \} = \det \{ Q \} \det \{ R - SQ^{-1}P \}.
\]

(51)

Then, we prove the lemma.

The eigenvalues of \( \Phi_M \) are the roots of its characteristic polynomials \( \det(zI - \Phi_M) = 0 \). To compute the determinant...
that \(M\) achieves consensus with con-
\[\gamma^2 \alpha + \theta \gamma - \theta \gamma = \theta - \theta \gamma = \theta \gamma - \theta \gamma \]
with \(z = \theta\) if and only if all roots of the three polynomials
\[\theta \gamma (\theta - \theta \gamma + \theta \gamma - \theta \gamma) = \theta \gamma (\theta + \theta \gamma + \theta \gamma - \theta \gamma) = \theta \gamma (\theta - \theta \gamma - \theta \gamma) = \theta \gamma - \theta \gamma = \theta \gamma - \theta \gamma \]
is given by
\[\theta \gamma - \theta \gamma = \theta \gamma - \theta \gamma \]
for the optimization problem
\[\det(\Phi_M) = \prod_{i=1}^{N} (\gamma + \theta - \lambda) \gamma \gamma = \theta - \theta \gamma = \theta \gamma - \theta \gamma \]
and
\[\gamma = \frac{1}{\gamma - \theta \gamma} \gamma - \theta \gamma \]
are within the left-half plane. Hence, the roots of the polynomials \(f_i(s), i = 1, 2, N\) are within the left-half plane (including the imaginary axis), where
\[f_1(s) = (\gamma_2 - \theta_0 \gamma_2 + \theta_0) s^2 + 2(\gamma_2 - \theta_0) s + \gamma_2 \theta_0 \gamma_2 + \theta_2 \]
\[f_2(s) = (\gamma_3 - (1 + \theta_0 - \alpha \lambda_2) \gamma_3 + (\theta_0 + \theta_2) \gamma_2 + (\theta_0 - \theta_2) \gamma_2) s^3 + (3 \gamma_3^2 - (1 + \theta_0 - \alpha \lambda_2) \gamma_3^2 + (\theta_0 + \theta_2) \gamma_2 + 3 \theta_2) s^2 + (3 \gamma_3^3 + (1 + \theta_0 - \alpha \lambda_2) \gamma_3^3 - (\theta_0 + \theta_2) \gamma_2 + 3 \theta_2) s + \gamma_2^2 + (1 + \theta_0 - \alpha \lambda_2) \gamma_2 + (\theta_0 \gamma_2 + \theta_2) \gamma_2 \gamma_2 + \theta_2 \]
\[f_N(s) = (\gamma_N^2 - (1 + \theta_0 - \alpha \lambda_N) \gamma_N^2 + (\theta_0 + \theta_2) \gamma_2 + (\theta_0 - \theta_2) \gamma_2) s^3 + (3 \gamma_N^3 - (1 + \theta_0 - \alpha \lambda_N) \gamma_N^3 + (\theta_0 + \theta_2) \gamma_2 + 3 \theta_2) s^2 + (3 \gamma_N^3 + (1 + \theta_0 - \alpha \lambda_N) \gamma_N^3 - (\theta_0 + \theta_2) \gamma_2 + 3 \theta_2) s + \gamma_2^2 + (1 + \theta_0 - \alpha \lambda_N) \gamma_2 + (\theta_0 \gamma_2 + \theta_2) \gamma_2 + \theta_2 \]
and the zeros of the second-order polynomial \(f_1(s)\) are within the closed left-half plane if and only if all coefficients are nonnegative. Construct the Routh table corresponding to \(f_2(s)\) and \(f_N(s)\) shown in Tables VI and VII, respectively. Based on the Routh stability criterion, the zeros of \(f_2(s)\) and \(f_N(s)\) are within the closed left-half plane if and only if the elements of the first column in the Routh tables are nonnegative. Hence (17)
is transformed to the following optimization problem

\[
\begin{align*}
\min_{\{\alpha, \theta_1, \theta_2\}} & \quad \gamma_2 \\
\text{s.t.} & \quad \gamma_2^3 - (1 + \theta_1 - \alpha \lambda_2)\gamma_2^2 + (\theta_1 + \theta_2)\gamma_2 - \theta_2 & \geq 0 \\
& \quad 3\gamma_2^3 - (1 + \theta_1 - \alpha \lambda_2)\gamma_2^2 - (\theta_1 + \theta_2)\gamma_2 + 3\theta_2 & \geq 0 \\
& \quad 3\gamma_2^3 + (1 + \theta_1 - \alpha \lambda_N)\gamma_2^2 - (\theta_1 + \theta_2)\gamma_2 - 3\theta_2 & \geq 0 \\
& \quad \gamma_2^3 + (1 + \theta_1 - \alpha \lambda_N)\gamma_2^2 + (\theta_1 + \theta_2)\gamma_2 + \theta_2 & \geq 0 \\
& \quad 2\gamma_2^2 - 2\theta_2 & \geq 0 \\
& \quad \gamma_2^2 + \theta_0\gamma_2 + \theta_2 & \geq 0
\end{align*}
\]

(57)

Recall that \(\alpha \geq 0\) and \(\gamma_2 \in (0, 1)\). By (63),

\[
\frac{4\gamma_2}{1-\gamma_2} - (\lambda_N - \lambda_2) > 0.
\]

As \(\gamma_2 \in (0, 1)\) decreases, \(\frac{4\gamma_2}{1-\gamma_2} - (\lambda_N - \lambda_2)\) increases monotonically, and \(\frac{4\gamma_2}{\lambda_N - \lambda_2}\) decreases monotonically. Hence, the smallest \(\gamma_2\) for (64) to hold must satisfy

\[
\frac{4\gamma_2}{\lambda_N - \lambda_2} = \frac{4(1 - \gamma_2)}{(1 - \gamma_2)^2} - (\lambda_N - \lambda_2).
\]

This is equivalent to

\[
\frac{4\lambda_2 \gamma}{(1 - \gamma)^2} - (\lambda_N - \lambda_2) \gamma_2 = (\lambda_N - \lambda_2)(1 - \gamma_2).
\]

By further simplification, we get

\[
4\lambda_2 \gamma_2 = (\lambda_N - \lambda_2)(1 - \gamma_2)^2
\]

and \(\gamma_2^2 - 2 \frac{\lambda_N + \lambda_2}{\lambda_N - \lambda_2} \gamma_2 + 1 = 0\). The unique solution of \(\gamma_2 \in (0, 1)\) from the last equation is \(\gamma_2 = \sqrt{\frac{\lambda_N + \lambda_2}{\lambda_N - \lambda_2}}\), which is the unique \(\alpha\) given in (21). Substituting \(\gamma_2^*\) into (64) gives \(\alpha^* = \frac{\lambda_N + \lambda_2}{\lambda_N - \lambda_2}\), which is the unique solution satisfying (58a)–(58d).

Finally, by substituting \(\gamma_2^*, \alpha^*, \theta_1^*, \theta_2^*\) to the remaining inequalities (58e)–(58m), we can check that all of them are greater than 0.

**E. Proof of Lemma 7**

Let \(z = \frac{r + 1}{r - 1}\). The corresponding continuous system of \(P(r, z, \lambda)^{-1}\) is

\[
P(s) = \frac{\lambda}{1 - r - sz + \frac{1}{r}}.
\]

Next we use Lemma 3 to compute the optimal gain margin. Do coprime factorization of \(P(s) = \frac{U(s)}{V(s)}\) as follows:

\[
U(s) = \frac{\lambda}{1 - r s + \frac{1}{r}} , \quad V(s) = -s + \frac{1}{r}.
\]

\[
Y_u(s) = \frac{1 - r}{2r} \quad \text{and} \quad Y_v(s) = \frac{1 - r}{r}.
\]

It is easy to check that \(U(s)Y_u(s) + V(s)Y_v(s) = 1\) and \(U(s)V(s)\) has two zeros in the right half plane, \(c_1 = 1\) and \(c_2 = \frac{1}{r^2} + 1\). Then, we obtain

\[
b_1 = U(c_1)Y_u(c_1) = 0
\]

\[
b_2 = U(c_2)Y_u(c_2) = \frac{1 + r}{r^2} - \frac{1}{r^2} = 1.
\]

It follows from direct computation that

\[
B_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} , \quad B_2 = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}.
\]

By Lemma 3, we have

\[
\gamma_{\inf} = \sqrt{\frac{B_1^{-1}B_2}{r}} = \frac{1}{r}.
\]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and associate editor for their constructive comments and insightful suggestions, which have improved the quality of this article.

REFERENCES

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

[2] S. Knorn, Z. Chen, and R. H. Middleton, “Overview: Collective control of multiagent systems,” IEEE Trans. Control Netw. Syst., vol. 3, no. 4, pp. 334–347, Dec. 2016.

[3] J. Qin, Q. Ma, Y. Shi, and L. Wang, “Recent advances in consensus of multi-agent systems: A brief survey,” IEEE Trans. Ind. Electron., vol. 64, no. 6, pp. 4972–4983, Jun. 2017.

[4] L. Xiao, S. Boyd, and S. J. Kim, “Distributed average consensus with least-mean-square-deviation,” J. Parallel Distrib. Comput., vol. 67, no. 1, pp. 73–87, Jan. 2007.

[5] S. Knorn, Z. Chen, and R. H. Middleton, “Overview: Collective control of multi-agent systems,” IEEE Trans. Autom. Control, vol. 63, no. 1, pp. 54–57, Jan. 2018.

[6] A. Nedic and J. Liu, “On convergence rate of weighted-averaging dynamics,” IEEE Trans. Autom. Control, vol. 63, no. 1, pp. 77–103, Mar. 2018.

[7] D. Jakovetić, J. Xavier, and J. M. F. Moura, “Weight optimization for consensus algorithms with correlated switching topology,” IEEE Trans. Signal Process., vol. 58, no. 7, pp. 3788–3801, Jul. 2010.

[8] M. Cao, A. S. Morse, and B. D. O. Anderson, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” IEEE Trans. Autom. Control, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[9] A. Nedic and J. Liu, “On convergence rate of weighted-averaging dynamics for consensus problems,” IEEE Trans. Autom. Control, vol. 52, no. 2, pp. 271–284, Feb. 2007.

[10] A. Nedic and J. Liu, “Convergence rate of weighted-averaging dynamics for consensus problems,” IEEE Trans. Autom. Control, vol. 52, no. 6, pp. 2565–2576, Nov. 2013.

[11] J. Qin, Q. Ma, Y. Shi, and L. Wang, “Recent advances in consensus of multi-agent systems: A brief survey,” IEEE Trans. Ind. Electron., vol. 64, no. 6, pp. 4972–4983, Jun. 2017.

[12] A. Nedic and J. Liu, “On convergence rate of weighted-averaging dynamics,” IEEE Trans. Autom. Control, vol. 63, no. 1, pp. 77–103, Mar. 2018.

[13] B. N. Oreshkin, M. J. Coates, and M. G. Rabbat, “Optimization and analysis of distributed averaging with short node memory,” IEEE Trans. Signal Process., vol. 58, no. 5, pp. 2850–2865, May 2010.

[14] M. Cao, D. A. Spielman, and E. M. Yeh, “Accelerated gossip algorithms for distributed computation,” in Proc. 44th Annu. Allerton Conf. Control, Comm., Comput., 2006, pp. 952–959.

[15] J. Liu, B. Anderson, M. Cao, and S. Morse, “Analysis of accelerated gossip algorithms,” Automatica, vol. 49, pp. 878–883, Apr. 2013.

Jing-Wen Yi (Member, IEEE) received the B.E. degree in automation from Yantai University, Yantai, China, in 2010, and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2016. She was a Visiting Student with Nanyang Technological University, Singapore, from September 2014 to May 2015. In April 2016, she joined the Wuhan University of Science and Technology, Wuhan, China, where she is currently an Associate Professor with the School of Information Science and Engineering. Her research interests include multiagent systems, distributed optimization, social networks, and graph signal processing.
Li Chai (Member, IEEE) received the B.S. degree in applied mathematics and the M.S. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2002. From August 2002 to December 2007, he was with Hangzhou Dianzi University, Hangzhou, China. He was a Professor with the Wuhan University of Science and Technology, Wuhan, China, from 2008 to 2022. In August 2022, he joined Zhejiang University, where he is currently a Professor with the College of Control Science and Engineering. He has been a Postdoctoral Researcher or Visiting Scholar with Monash University, Melbourne, VIC, Australia, Newcastle University, Callaghan, NSW, Australia, and Harvard University, Cambridge, MA, USA. He has authored or coauthored more than 100 fully refereed papers in prestigious journals and leading conferences. His research interests include distributed optimization, filter banks, graph signal processing, and networked control systems.

Dr. Chai is the recipient of the Distinguished Young Scholar of the National Science Foundation of China. He is currently the Associate Editor for the IEEE TRANSACTIONS ON CIRCUIT AND SYSTEMS II: EXPRESS BRIEFS, Journal of Control and Decision, and Journal of Image and Graphs.

Jingxin Zhang (Member, IEEE) received the M.E. and Ph.D. degrees in electrical engineering from Northeastern University, Shenyang, China, in 1983 and 1988, respectively. Since 1989, he has held research and academic positions with Northeastern University, Shenyang, China, University of Florence, Florence, Italy, University of Melbourne, Melbourne, VIC, Australia, the University of South Australia, Adelaide, SA, Australia, Deakin University, Geelong, VIC, Australia, and Monash University, Melbourne, VIC, Australia. He is currently an Associate Professor of Electrical Engineering with the Swinburne University of Technology, Melbourne, VIC, Australia, and an Adjunct Associate Professor of Electrical and Computer Systems Engineering with Monash University, Melbourne, VIC, Australia. His research interests include signals and systems and their applications to biomedical and industrial systems.

Dr. Zhang was the recipient of the 1989 Fok Ying Tong Educational Foundation (Hong Kong) for the Outstanding Young Faculty Members in China, and 1992 China National Education Committee Award for the Advancement of Science and Technology.