Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Exploiting the Agent's Memory in Asymptotic and Finite-time Consensus over Multi-agent Networks

(Article begins on next page)

28 November 2024

Exploiting the Agent's Memory in Asymptotic and Finite-time Consensus over Multi-agent Networks

Gianni Pasolini, *Member, IEEE*, Davide Dardari, *Senior, IEEE*, and Michel Kieffer, *Senior, IEEE*

Abstract—This paper proposes two average consensus algorithms exploiting the memory of agents. The performance of the proposed as well as of several state-of-the-art consensus algorithms is evaluated considering different communication ranges, and evaluating the impact of transmission errors. To compare asymptotic and finite-time average consensus schemes, the ε-convergence time is adopted for a fair comparison. A discussion about memory requirements, transmission overhead, *a priori* information on network topology, and robustness to errors is provided.

I. INTRODUCTION

This paper addresses the problem of distributed consensus in a wireless multi-agent network (MAN). Agents have to reach an agreement, without the help of a central coordinator, regarding a certain quantity of interest that depends on their individual measurement or state [1]. Specifically, this paper focuses on distributed average consensus [2], [3], where each agent locally estimates the average of quantities measured by all agents in the MAN, by exchanging information with its one-hop neighbours only.

To address this problem, several algorithms have been proposed, either belonging to the *asymptotic* consensus class [4]–[7] or to the *finite-time* consensus class [8]–[13]. In both cases, the average consensus is achieved through a proper exchange of consensus estimates among neighbours.

We propose and analyze an original asymptotic consensus scheme with memory based on finite impulse response (FIR) filtering. This scheme achieves, in many configurations, lower convergence times or better robustness to transmission errors compared to state-of-the-art asymptotic consensus schemes with memory. In addition, we introduce a finite-time average consensus algorithm with memory and assess its performance.

As for the comparison of algorithms, it is noticeable that in practical applications, consensus up to a given tolerance ε is usually accepted. Therefore, the ε-convergence time [14], defined as the time needed to reach the required accuracy level ε, appears as an appropriate performance figure to compare the convergence speed of both finite-time and asymptotic schemes.

In this paper, the performances of the proposed, as well as of some state-of-the-art consensus algorithms, are thus evaluated in the numerical results in terms of ε-convergence time with

G. Pasolini and D. Dardari are with the Department of Electrical, Electronic and Information Engineering (DEI) and CNIT, University of Bologna, Italy.
M. Kieffer is with L2S, CNRS–CentraleSupelec–Univ Paris-Sud, Univ Paris-Saclay, France.

and without transmission losses. A discussion about memory, needed *a priori* information and robustness to errors is also provided.

II. RELATED WORK

Consensus schemes are the core of several distributed signal processing approaches. Applications can be found, for example, in distributed maximum likelihood [15] and least square estimation [16], distributed Kalman [17], [18] and particle filtering [19], distributed optimization [20] as well as distributed algorithms for principal component analysis [21]. Focusing on wireless networks, consensus schemes represent efficient fully distributed approaches to let wireless agents agree on a specific quantity without paying the price of the large overhead caused by routing algorithms [22], [23]. In Internet of Things (IoT) scenarios [24]–[26], they have been proposed for decision making of services at edge nodes [27] as well as in vehicular networks [28].

In the literature, many memory-less average consensus algorithms have been proposed and proved to converge asymptotically (*e.g.*, [4]–[7]). When the graph topology is known, a semi-definite programming approach has been proposed by [6] to determine the optimal graph link weights to get the best convergence rate towards consensus.

The memory of agents is exploited in [29]–[33] to show that the consensus may be further accelerated compared to the schemes proposed in [6]. A polynomial filtering of successive consensus estimates stored in the memory of agents is introduced in [34]. The computation of the coefficients of the optimal polynomial filter is formulated as a semi-definite programming problem. This idea has been extended in [35], where Chebychev polynomials are considered.

The schemes proposed in [30]–[32], where only the previous consensus estimates are kept in memory by each agent, are further analyzed in [36]: when only lower and upper bounds are known for the eigenvalues of the linear mapping used in the iterative consensus estimation, keeping the last consensus estimate is enough to get the fastest convergence.

More sophisticated filtering schemes have been introduced in [37], [38], where the problem of dynamic consensus is also considered. Past messages as well as past estimates are stored over a given time horizon in agents' memory and processed periodically to speed-up convergence, generalizing previous results from [30]–[32], [34], [35].

In parallel, several variants of average consensus algorithms capable of converging in finite time have been proposed, see *e.g.*, [8], [9], [11], [13], [39], [40]. In the approach proposed by [8], [39], linear update iterations, as in asymptotic approaches, are performed first. During this phase, each agent stores the succession of its consensus estimates. Once N_d iterations have been performed, where N_d is the number of distinct eigenvalues of the matrix associated to the linear consensus update equation, the final consensus is obtained at each agent by a linear combination of its N_d past estimates. This estimation process has been further improved in [39], which shows that the final consensus value can be evaluated in the minimum number of steps, without knowing N_d, by checking a rank condition of a Hankel matrix of the observations of each agent.

A matrix factorization approach has been proposed in [41], resulting in a communication scheduling between neighbour agents that reaches the consensus in a finite number of steps. In [42], polynomial filtering is also used to accelerate the convergence toward average consensus on undirected networks. The gain of a second-order filtering is investigated. This paper also establishes the benefit of combining polynomial filtering with optimization of link weights. This approach is also considered in [13], where the network topology is assumed to be known. A selection technique is proposed for the weights associated with the edges of the graph so as to be able to obtain a minimal polynomial, to get the fastest finite time consensus.

In [9], [40] a set of N_d consensus matrices are designed to successively cancel the N_d distinct eigenvalues of the Laplacian matrix of the network graph. The consensus is again reached in N_d iterations. Nevertheless, this approach requires an accurate estimate of the eigenvalues of the graph Laplacian, and may be numerically unstable. This technique has been extended in [40], which characterizes the finite-time behavior of successive eigenspace nulling algorithms, for which the algorithm of [9] represents a special case. In parallel, [11] introduced an approach inspired by graph filtering, leading to the convergence in N_d iterations. In that case, the transient behavior may lead to numerical instability. These approaches are well suited to relatively small networks, since, except for well-structured networks, N_d is usually of the same order of magnitude as the number of agents of the network.

III. OUR CONTRIBUTION AND PAPER ORGANIZATION

With respect to the literature, our paper provides the following original contributions.

First, after the average consensus problem along with the performance metrics is formalized in Sec. IV, we introduce in Sec. V a novel asymptotic consensus algorithm which exploits the agents' memory to increase the convergence speed. We model such an algorithm as a discrete-time linear system with a feedback loop including a simple FIR filter. We provide a closed-form expression of the optimum filter coefficients in the case of memory depth 1. Even though the general modeling of asymptotic consensus as a feedback loop is not new [38], the adoption of a FIR loop filter and the derivation of its coefficients in the case of memory depth 1 is novel. In the numerical results, we show that, in the presence of

transmission failures, our algorithm outperforms the state-of-the-art asymptotic scheme based on infinite impulse response (IIR) loop-filtering proposed in [38] as well as the scheme proposed in [33]. We will name these reference schemes asymptotic infinite impulse response (A-IIR) and asymptotic-Kruzik (A-Kruzik), respectively.

Second, in Sec. VI we introduce an original hybrid finite-time consensus algorithm (named finite-time memory sum with propagation (F-MSwP)) that does not require the knowledge of the eigenvalues of the network's Laplacian matrix. In the numerical results, we show that this algorithm achieves an impressive performance level, very close to the theoretic lower-bound given by the network diameter D, which is typically much less than N_d. It is worth noting that, differently from F-MSwP, most of the finite-time consensus schemes, such as those proposed in [9], [40], require a very accurate computation of the Laplacian eigenvalues, to an extent that they might fail to converge when their computation is not extremely accurate. This prevents their use in practical schemes that usually work with low numerical precision.

Third, to the best of the authors' knowledge, our paper is the first one that compares the performance of state-of-the-art asymptotic and finite-time consensus algorithms. In fact, asymptotic consensus schemes, which converge to the consensus only asymptotically, are usually compared in terms of convergence rate (that is, in terms of convergence speed in the asymptotic regime). Contrarily, finite-time algorithms are usually compared in terms of number of iterations needed to achieve the exact convergence. Clearly, the two figures of merit are not comparable, making it impossible to have a clear picture on which consensus algorithm (asymptotic or finite-time) is better suited to a given application, which in general requires the convergence up to a tolerable error ϵ. The ϵ-convergence time [14], adopted in our paper as a common performance metric, allowed for the first time the comparison of algorithms belonging to both categories, which is an original contribution shown in the numerical results in Sec. VII.

Finally, the conclusions are drawn in Sec. VIII.

IV. PROBLEM FORMULATION

A. Problem formulation and notations

Consider a MAN consisting of N communicating agents. The communication links between agents are described by an undirected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of agent indexes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges describing the communication links. For any agent $n \in \mathcal{V}$, let $\mathcal{N}_n$ denote its set of one-hop neighbours. One assumes that $n \notin \mathcal{N}_n$. For further convenience, we define the symmetric *weighted adjacency matrix* $\mathbf{W} = [w_{nj}]$, which is usually chosen such that $\mathbf{W} \in \mathcal{W}$, where

$$\mathcal{W} = \{\mathbf{W} \in \mathbf{R}^{N \times N} | w_{nj} = 0 \text{ if } \{n, j\} \notin \mathcal{E} \text{ or } n = j\}. \quad (1)$$

One also defines the diagonal *degrees matrix* $\mathbf{D} = \mathrm{diag}\,(d(1), d(2), \ldots, d(N))$, where $d(n) = \sum_j w_{nj}$ denotes the degrees of agent n, and the *Laplacian matrix* $\mathbf{L}_w = \mathbf{D} - \mathbf{W}$. The spectrum of $\mathbf{L}_w$, given by the set of its eigenvalues, provides several insights on the graph properties. For instance,

denoting with $\{\lambda_1, \lambda_2, \ldots \lambda_N\}$ the ordered set of eigenvalues of $\mathbf{L}_w$, with $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$, the graph is connected if $\lambda_1 = 0$ and $\lambda_2 > 0$. In our analysis we only consider connected graphs.

When the entries (weights) of $\mathbf{W}$ belong to $\{0, 1\}$ (*i.e.*, the communication link between two agents is simply either absent or present), the classic *adjacency matrix* $\mathbf{W} = \mathbf{A} = [a_{nj}]$ is obtained, and $d(n) = |\mathcal{N}_n|$ denotes the number of neighbours of agent n, so that $\mathbf{L}_w = \mathbf{L}$ and $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

Assume that time is slotted and that at time slot (round) $k=0$ each agent $n \in \mathcal{V}$ performs some scalar measurement $x_n(0)$. In this paper we investigate the distributed average consensus problem, meaning that all agents aim at computing

$$\overline{x} = \frac{1}{N}\left[x_1(0) + x_2(0) + \ldots + x_N(0)\right], \qquad (2)$$

that is, the average of all measurements $\{x_n(0)\}$. In each of the following time slots ($k \geq 1$), a round of the consensus algorithm takes place, under the constraint that each agent broadcasts no more than one scalar during each round ("one scalar" rule).

Although the generalization to the case of consensus on vectors is straightforward, in the following we stick on scalar consensus to lighten notations.

B. Performance metrics

The performance metric usually adopted to characterize asymptotic consensus algorithms is the *Network Convergence Rate*

$$R_c = \lim_{k \to \infty} \log \frac{|\sum_n x_n(k) - N\overline{x}|}{|\sum_n x_n(k+1) - N\overline{x}|}, \qquad (3)$$

which provides an indication on how fast the algorithm converges to the consensus in the asymptotic regime, and hence it is typically used as a design guideline, as it will be clear in the next section. Nonetheless, depending on the application, one might be more interested in comparing different schemes in terms of the time needed to reach a certain tolerable error ε by introducing the n-th *agent ε-convergence time*

$$T_\varepsilon^{(n)} = \inf\{k^* : |x_n(k) - \overline{x}| < \varepsilon \ \forall k \geq k^*\} \qquad (4)$$

and the *network ε-convergence time*, or simply *ε-convergence time* [14]

$$T_\varepsilon = \max\left\{T_\varepsilon^{(n)}\right\}. \qquad (5)$$

V. PROPOSED ASYMPTOTIC CONSENSUS METHOD

The proposed asymptotic consensus approach extends the classical memoryless update equation

$$x_n(k+1) = x_n(k) + \sum_{j \in \mathcal{N}_n} w_{nj}\left(x_j(k) - x_n(k)\right), \qquad (6)$$

by including a linear combination of past innovation components as follows

$$x_n(k+1) = x_n(k) + \sum_{\ell=0}^{M} d_\ell \sum_{j \in \mathcal{N}_n} w_{nj}(x_j(k-\ell) - x_n(k-\ell)), \qquad (7)$$

thus accounting for the estimates made by agent n as well as the messages it received from its neighbours from round $k-M$

to round k. Here, M represents the memory depth of each agent. In vector form, we have

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \sum_{\ell=0}^{M} d_\ell \mathbf{W} \mathbf{x}(k-\ell). \qquad (8)$$

We consider (7) starting from $k=0$, assuming that $x_n(0)$ is equal to the measurement taken by agent n at round $k=0$ and $x_n(k) \triangleq 0$ for $k < 0$. The local estimate $x_n(k)$ is also denoted as *state* of agent n at round k.

Eq. 7 entails that each agent keeps in memory the last M scalars received by each of its neighbours so that it can update the state. The degrees of freedom of (7) are thus the non-zero coefficients of $\mathbf{W}$ and the vector $\mathbf{d} = [d_0, \ldots, d_M]^T$ of weights given to the estimates and received messages in the M past rounds. Notably, (7) can be used considering already known, well-investigated, $\mathbf{W}$ matrices, usually adopted by basic memoryless schemes ruled by (6), which are enhanced thanks to the memory contributions weighted by the vector $\mathbf{d}$. In particular, in the following we will consider the matrix $\mathbf{W}$ of two well-know basic consensus schemes, hereafter recalled.

1) asymptotic-constant edge weights (A-CEW) consensus. In A-CEW consensus, all non-zero components of $\mathbf{W}$ are equal, *i.e.*, $w_{nj} = d_0 \, a_{nj}$ and the maximum convergence rate is achieved when $d_0 = \frac{2}{\lambda_2 + \lambda_N}$ [6]. In the following, we will always assume such optimal implementation of A-CEW. When the A-CEW weights will be used in (7), the corresponding memory-enhanced scheme will be named asymptotic finite-impulse response CEW (A-FIR-CEW).

2) asymptotic Metropolis (A-Metro) Consensus. Choosing the entries of $\mathbf{W}$ according to $w_{nj} = [1 + \max\{d(n), d(j)\}]^{-1}$ for $j \in \mathcal{N}_n$ and zero otherwise, leads to the Metropolis consensus [7]. Note that Metropolis requires that each agent knows the number of its neighbors, that is, its local topology. When the Metropolis weights are used in (7), the corresponding memory-enhanced scheme will be referred to as asymptotic finite-impulse response Metropolis (A-FIR-Metro).

A. Eigenvalue Decomposition

Adopting the representation introduced in [37], [38], the linear average consensus with memory (8) can be represented as a feedback network system, described in Fig. 1(a), with

$$\mathbf{A}(z) = A(z)\, \mathbf{I}_N \qquad \mathbf{B}(z) = B(z)\, \mathbf{I}_N, \qquad (9)$$

where z^{-1} denotes the unitary delay and $\mathbf{I}_N$ the identity matrix of size N. $A(z)$ and $B(z)$ represent, respectively, the transfer function of the forward and feedback filters applied at each agent. In particular, the system described by (8) is equivalent to that depicted in Fig. 1(a) provided that $A(z)$ corresponds to an integrator, *i.e.*,

$$A(z) = \frac{1}{1 - z^{-1}}, \qquad (10)$$

and $B(z)$ to a FIR filter with $M+1$ coefficients

$$B(z) = \sum_{l=0}^{M} d_l \, z^{-l}. \qquad (11)$$

(a) Full system.

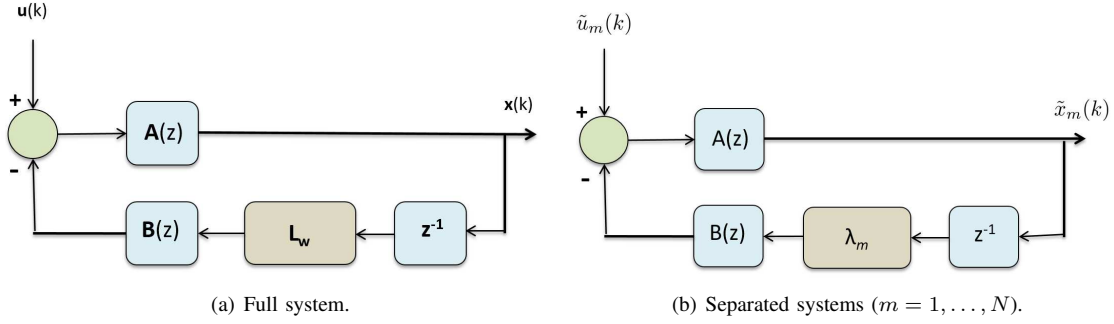(b) Separated systems ($m = 1, \ldots, N$).

Fig. 1. Linear average consensus scheme with memory represented as a feedback network system.

The system is excited by the impulsive signal $\mathbf{u}(k) = \delta(k)\mathbf{x}(0)$, with $\delta(k) = 1$ for $k = 0$ and zero otherwise.

As described in [37], the Laplacian matrix (supposed symmetric) can be diagonalized as $\mathbf{L}_w = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$, where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$ contains the eigenvalues of $\mathbf{L}_w$, and $\mathbf{V} = [\mathbf{v}_1 \ldots \mathbf{v}_N]$ contains the corresponding eigenvectors. For a fully connected graph, $\lambda_1 = 0$ and $\mathbf{v}_1 = \frac{\mathbf{1_N}}{\sqrt{N}}$. Then, the estimate (which is also the state and output) at round k can be decomposed as sum of N components

$$\mathbf{x}(k) = \mathbf{V}\mathbf{V}^T\mathbf{x}(k) = \mathbf{v}_1\,\tilde{x}_1(k) + \sum_{m=2}^{N} \mathbf{v}_m\,\tilde{x}_m(k), \quad (12)$$

where $\tilde{x}_m(k) = \mathbf{v}_m^T\mathbf{x}(k)$ is the projection of $\mathbf{x}(k)$ onto the eigenvector $\mathbf{v}_m$. Therefore, the scheme in Fig. 1(a) can be represented as a set of N separated systems, each one elaborating a different component $\tilde{x}_m(k)$, as shown in Fig. 1(b). The transfer function of the m-th system is

$$H_m(z) = \frac{A(z)}{1 + \lambda_m z^{-1}B(z)A(z)}, \quad (13)$$

whereas its input is the impulsive signal

$$\tilde{u}_m(k) = \mathbf{v}_m^T\mathbf{u}(k) = \mathbf{v}_m^T\delta(k)\mathbf{x}(0). \quad (14)$$

Since the inputs of the separated systems $H_m(z)$, for $m = 1, 2, \ldots, N$, are impulsive signal, the analysis of their outputs reduces to the analysis of their impulse responses. Specifically, considering the system $m = 1$ in Fig. 1(b), being $\mathbf{v}_1 = \frac{\mathbf{1_N}}{\sqrt{N}}$ one immediately obtains from (14) that the input $\tilde{u}_1(k)$ is

$$\tilde{u}_1(k) = \frac{1}{\sqrt{N}}\sum_{n=1}^{N} x_n(0)\delta(k) = \sqrt{N}\,\overline{x}\,\delta(k). \quad (15)$$

Since $\lambda_1 = 0$, the feedback path is absent, hence the output of system $m = 1$ is simply the integrator response to the discrete time input (15), that is $\tilde{x}_1(k) = \sqrt{N}\,\overline{x}\,s(k)$, where $s(k) = 1$ for $k \geq 0$ and zero otherwise is the unit step sequence. The first component in (12) is therefore

$$\mathbf{v}_1\tilde{x}_1(k) = [\overline{x}\,\overline{x}\ldots\overline{x}]^T, \quad (16)$$

for $k \geq 0$. One observes that the component $\mathbf{v}_1\tilde{x}_1(k)$ of (12) converges to the target average consensus as soon as $k = 0$. Therefore, a sufficient condition to make the system converge to the average consensus, is designing the feedback filter $B(z)$ such that the impulse response of all systems with $m > 1$ tends to zero.

B. Analysis and filter design for the proposed asymptotic consensus algorithm

1) General case: Given (10) and (11), the transfer function of the system in Fig. 1(b) is

$$H_m(z) = \frac{1}{1 - z^{-1} + \lambda_m \sum_{l=0}^{M} d_l z^{-l-1}}. \quad (17)$$

The desired behavior is that the m-th component of (12), with $m > 1$, vanishes to zero as fast as possible for increasing k, *i.e.*, the impulse response of $H_m(z)$ converge to zero as fast as possible. The behavior depends on the zeros z_{mi}, $m = 2, 3, \ldots N$, $i = 1, 2, \ldots M+1$, of the denominator

$$F(z, \lambda, \mathbf{d}) = 1 - z^{-1} + \lambda_m \sum_{l=0}^{M} d_l\, z^{-l-1} \quad (18)$$

of $H_m(z)$, which must lie in the unit circle and close to zero (for stability and fast convergence). The zeros of (18) depend on the eigenvalues λ_m of $\mathbf{L}_w$ and can be adjusted by properly designing the coefficients d_l (filter taps). Specifically, one wants to maximize the network convergence rate determined by the zero z_{mi} of (18) having the maximum absolute value

$$R_c = -\log \max_{\substack{m\in\{2,3,\ldots,N\} \\ i\in\{1,2,\ldots,M+1\}}} |z_{mi}|. \quad (19)$$

This corresponds to solve the optimization problem

$$\mathbf{d}^* = \arg\min_{\mathbf{d}} \max_{\substack{\lambda\in\{\lambda_2,\lambda_3,\ldots,\lambda_N\} \\ z\in\mathbb{C}}} |z| \quad (20)$$

subject to $F(z, \lambda, \mathbf{d}) = 0$.

For instance, when $M = 0$, $F(z, \lambda, \mathbf{d}) = 0$ has a single solution $z_{m1} = 1 - d_0\lambda_m$ for each m.

The analytical derivation of the solution of (20) in the general case appears unrealistic. In the numerical results we resort to constrained optimization except in the case $M = 1$, for which an analytical solution is provided in what follows.

2) Case with memory $M = 1$: For $M = 1$, the constraint $F(z, \lambda, \mathbf{d}) = 0$ in (20) boils down to a second-order equation with roots, for each eigenvalue λ_m, given by

$$z_{m1} = z_{m1}(d_0, d_1) = \frac{1}{2}\left[1 - d_0\lambda_m - \sqrt{(1 - d_0\lambda_m)^2 - 4d_1\lambda_m}\right] \quad (21)$$

$$z_{m2} = z_{m2}(d_0, d_1) = \frac{1}{2}\left[1 - d_0\lambda_m + \sqrt{(1 - d_0\lambda_m)^2 - 4d_1\lambda_m}\right] \quad (22)$$

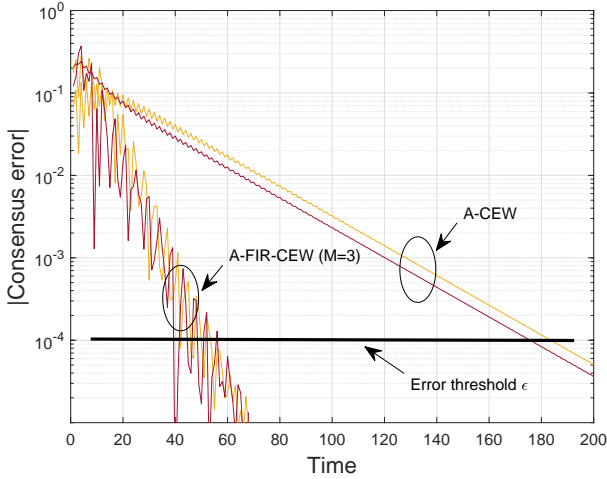with $(d_0, d_1) \in \mathbb{R}^2 \setminus \{\frac{1}{\lambda_m}, 0\}$.

Fig. 2. Absolute errors experienced by two agents as function of the time for A-CEW and A-FIR-CEW with $M = 3$.

Theorem 1. *The solutions of* (20) *for $M = 1$ are*

$$d_0^* = \frac{\lambda_2 + 3\lambda_N}{\lambda_N (\lambda_N + 3\lambda_2)} \ and \ d_1^* = \frac{(\lambda_2 - \lambda_N)^2}{\lambda_N (\lambda_N + 3\lambda_2)^2}.$$

Proof. See the Appendix. $\square$

The corresponding convergence rate is

$$R_c^{\text{A-FIR}} = -\log \max_{m \in \{2,3,...,N\}} |z_{m2}(d_0^*, d_1^*)| = \log \frac{3\lambda_2 + \lambda_N}{\lambda_N - \lambda_2}. \tag{23}$$

The improvement with respect to the convergence rate

$$R_c^{\text{A-CEW}} = -\log \frac{\lambda_N - \lambda_2}{\lambda_N + \lambda_2} \tag{24}$$

of the A-CEW memory-less consensus introduced in [6] is

$$R_c^{\text{A-FIR}} - R_c^{\text{A-CEW}} = \log \frac{3\lambda_2 + \lambda_N}{\lambda_2 + \lambda_N}, \tag{25}$$

which is always positive, meaning that the length-2 FIR filtering is beneficial.

Remark. The convergence rate (19) rules the rapidity with which agents converge to the consensus in the asymptotic regime, that is, after the conclusion of the initial transient that depends on the agents' own measurements. For a given network topology, different consensus algorithms simply correspond to different values of the zeros z_{mi} in (19), which in turn correspond to different convergence rates (that is, different convergence speeds). This is evident in Fig. 2, which shows the time evolution of the absolute consensus errors of two agents adopting either the classic A-CEW consensus [6] or its enhanced version A-FIR-CEW (proposed in this paper) with memory $M = 3$.

Clearly, increasing the converge rate of asymptotic consensus algorithms automatically reduces the ϵ-convergence time, as can be seen in Fig. 2, where an example of the error threshold ϵ is also depicted: provided that the initial transient is over, a higher convergence rate makes consensus errors reach sooner the ϵ threshold, which results in a lower ϵ-convergence time, the latter depending on the measurements.

This means that, even when the ϵ-convergence is adopted as performance metric, asymptotic consensus algorithms can still be designed adopting the maximization of the convergence rate as design criterion. Moreover, one observes that as the accuracy requirement becomes more stringent (as ϵ decreases), the benefit provided by the enhanced strategies proposed in this paper gets more and more significant.

VI. PROPOSED FINITE-TIME CONSENSUS WITH MEMORY

In this section, we introduce the finite-time memory sum (F-MS) algorithm, a finite-time consensus approach exploiting the memory of agents, and its enhanced version denoted F-MSwP, which significantly improves its performance.

F-MS: This algorithm, which is an extension of that proposed by the authors in [43], works as follows:

- At round $k = 0$, each agent n makes its measurement. The measured value $x_n(0)$, which represents the information owned by agent n at round $k = 0$, is broadcast in the same round.
- At each round $k = 1, 2, ..., T-1$, with T denoting the number of rounds needed for the convergence of all agents (i.e., all agents have the necessary information to compute the consensus), each agent n computes a new scalar $x_n(k)$ by linearly combining the scalars $x_j(k-1), j \in \mathcal{N}_n$, received from its neighbours in round $k-1$. The updated scalar $x_n(k)$ represents the aggregated information that agent n broadcasts in the same round, which is given by

$$x_n(k) = \sum_{j \in \mathcal{N}_n} r_{nj}(k) \, x_j(k-1), \tag{26}$$

where the $r_{nj}(k)$ are independent, identically distributed (i.i.d.) random coefficients belonging to $\{-1, 1\}$.

One observes, by the way, that (26) is not a state update equation, as $x_n(k)$ is not a state variable that is eventually meant to converge to the consensus, as for asymptotic consensus schemes. Instead, (26) is simply aimed at aggregating in a single scalar the information received from the neighbours in the previous round.

Considering the $N \times N$ adjacency matrix $\mathbf{A} = [a_{nj}]$, for $k \geqslant 1$ we can define the *randomized adjacency matrix* $\mathbf{A}_{\text{R}}(k) = [a_{\text{R}_{nj}}(k)]$ as

$$a_{\text{R}_{nj}}(k) = \begin{cases} 0 \ \text{if} \ a_{nj} = 0 \\ r_{nj}(k) \ \text{if} \ a_{nj} = 1. \end{cases} \tag{27}$$

Observing that $\mathbf{x}(1) = \mathbf{A}_{\text{R}}(1) \mathbf{x}(0)$, $\mathbf{x}(2) = \mathbf{A}_{\text{R}}(2) \mathbf{x}(1) = \mathbf{A}_{\text{R}}(2)\mathbf{A}_{\text{R}}(1) \mathbf{x}(0)$, and so on recursively, the expression of the data transmitted (at the network level) at round $k \geqslant 1$ is

$$\mathbf{x}(k) = \mathbf{A}_{\text{R}}(k) \cdots \mathbf{A}_{\text{R}}(2)\mathbf{A}_{\text{R}}(1) \mathbf{x}(0). \tag{28}$$

At each round $k \geqslant 1$, each agent n also collects in the vector $\mathbf{y}_n(k)$ the data (*i.e.*, the information) received at round $k - 1$ from its neighbours:

$$\mathbf{y}_n(k) = \mathbf{I}_{\mathcal{N}_n} [\mathbf{x}(k-1)] = \mathbf{I}_{\mathcal{N}_n} \mathbf{A}_{\text{R}}(k-1) \ldots \mathbf{A}_{\text{R}}(1) \mathbf{x}(0), \tag{29}$$

where $\mathbf{I}_{\mathcal{N}_n}$ represents the matrix composed of the rows of the identity matrix whose indexes are included in the set $\mathcal{N}_n$.

Therefore, after k rounds, the total amount of data collected by agent n, including its own measurement $y_n(0) = x_n(0)$, can be stacked in the column vector

$$
\mathbf{c}_n(k) = \begin{bmatrix} y_n(0) \\ \mathbf{y}_n(1) \\ \vdots \\ \mathbf{y}_n(k) \end{bmatrix} = \begin{pmatrix} \mathbf{e}_n^T \\ \mathbf{I}_{\mathcal{N}_n} \\ \mathbf{I}_{\mathcal{N}_n}\mathbf{A}_{\mathrm{R}}(1) \\ \vdots \\ \mathbf{I}_{\mathcal{N}_n}\mathbf{A}_{\mathrm{R}}(k-1)\cdots\mathbf{A}_{\mathrm{R}}(1) \end{pmatrix} \mathbf{x}(0) ,
$$

$$
= \mathbf{F}_n(k)^T\mathbf{x}(0) , \tag{30}
$$

where $\mathbf{e}_n$ is a column vector of all zeros except a one in the n-th position, to account for the fact that each agent knows its own measurement, whose stored value is labelled as $y_n(0)$.

Each agent n aims then at evaluating (2) from the knowledge of (30).[1] Using a linear estimator, one has to determine whether there exists for some $k > 0$ a vector $\mathbf{w}_n(k)$ such that

$$
\mathbf{c}_n(k)^T\mathbf{w}_n(k) = \overline{x} . \tag{31}
$$

The weight vector $\mathbf{w}_n(k)$ can be found by solving

$$
\mathbf{F}_n(k)\mathbf{w}_n(k) = \frac{1}{N}\mathbf{e} , \tag{32}
$$

where $\mathbf{e}$ is a column vector of all ones. Equation (32) admits at least one solution if and only if

$$
\mathrm{rank}\left(\mathbf{F}_n(k)\right) = \mathrm{rank}\left(\mathbf{F}_n(k)\big|\mathbf{e}\right) \tag{33}
$$

which is given by

$$
\mathbf{w}_n(k) = \frac{1}{N}\mathbf{F}_n(k)^T\left(\mathbf{F}_n(k)\mathbf{F}_n(k)^T\right)^{-1}\mathbf{e} . \tag{34}
$$

If we denote with $T^{(n)}$ the minimum number of rounds needed to find at least one solution for agent n, the convergence time for the network, expressed in terms of rounds, is given by $T = \max_n T^{(n)}$. Assuming that all agents know the matrices $\mathbf{A}_{\mathrm{R}}(k)$, each agent can compute in advance (with respect to the algorithm execution) its weights vector $\mathbf{w}_n(k) \,\forall k$ and the number of rounds needed for the convergence within the whole network.

Note that (33) is a condition that may be checked in advance to determine whether the convergence will be achieved for all agents, for at least one agent, or for no agent. Specifically, if (33) is satisfied for all agents, the algorithm can be executed as it surely converges in a finite-time. Instead, if (33) is not satisfied for all agents, F-MS still deserves consideration because it is the basic component of the two-step hybrid algorithm introduced in the following subsection, denoted F-MSwP. If (33) is not satisfied for any agent, a different algorithm must be chosen. Fortunately, according to our simulations with a random topology, reported in the numerical results, this event never happened so that we can conjecture that it is very rare.

F-MSwP: Since a node can check (33) in advance, it is able to determine whether and at which round it is able to evaluate the consensus. The performance of F-MS can then

be dramatically improved if, as soon as an agent reaches the consensus, the network stops broadcasting (26) and simply starts propagating the consensus value.

As it will be shown in the numerical results, this procedure significantly improves the chances of convergence and ensures that after no more than D rounds from the first-agent consensus event, all agents achieve the consensus. As a matter of fact, with F-MSwP it is sufficient that (31) leads only one agent to the convergence to make all agents converge.

VII. NUMERICAL RESULTS

A. Simulation setup and performance figures

The original algorithms introduced in this paper and benchmark state-of the-art algorithms are implemented with Matlab and compared in a reference scenario. We consider $N_{\mathrm{R}} = 1000$ network realizations, each with $N = 100$ agents randomly deployed in a square region of unit area.

At each run, the initial state $x_n(0)$ of the generic agent n is randomly chosen with a zero-mean unit-variance Gaussian distribution, representing its local measurement. All agents have the same coverage range R, and a direct communication link exists between two agents if their Euclidean distance is below R. Only the realizations resulting in a connected network are considered. Transmission losses are also simulated: a communication between two neighbouring agents might fail with probability P_{err}. The communication channel is assumed reciprocal, so that a loss in one direction in a given round implies a loss in the opposite direction in the same round. Transmission losses are considered uncorrelated in different rounds and as well among different links. For each network realization, each algorithm is iterated until the time step $\widetilde{T}_\varepsilon$ in which the following condition is fulfilled by all agents

$$
E^{(n)}\left(\widetilde{T}_\varepsilon\right) < \varepsilon \quad \text{with } n = 1, 2, \ldots, N , \tag{35}
$$

where $E^{(n)}(k) = |x^{(n)}(k) - \overline{x}|$ is the consensus estimation error of agent n at iteration k, and ε represents the maximum tolerable error. Clearly, for the algorithm under investigation and for the given network realization, $\widetilde{T}_\varepsilon$ (in the following referred to as ε-convergence time) represents a lower bound of the ε-convergence time as defined in (5).[2] All numerical results are obtained setting $\varepsilon = 10^{-4}$. Considering all the N_{R} realizations, for each algorithm we derive the (empirical) cumulative distribution function (CDF) of the ε-convergence time $\widetilde{T}_\varepsilon$.

We remark that, in general, finite-time schemes do not gradually approach (iteration after iteration) the average consensus, as occurs in the case of asymptotic schemes, but, when successful, they compute the exact consensus (up to numerical accuracy) as soon as they collect enough information. This means that the ϵ-convergence time coincides with the exact convergence time.

B. Considered schemes

For the performance comparison, we consider the following asymptotic and finite-time algorithms, with and without memory, namely

---

[1]This entails that each agent knows $\mathbf{A}_{\mathrm{R}}(k)$ for $k = 1, 2, \ldots, T-1$. This assumption is not critical provided that agents share the same seed for the generation of random numbers.

[2]Due to the finite simulation time, (4) cannot be verified for all $k > \widetilde{T}_\varepsilon$.

- the memoryless asymptotic average consensus algorithms A-CEW [6], A-Metro [7];
- the recently introduced asymptotic algorithms with memory A-IIR [38] and A-Kruzik [33], as well as our memory-enhanced versions of A-CEW and A-Metro, that is, A-FIR-CEW and A-FIR-Metro, with $M = 1$, $M = 2$ and $M = 3$ (Section V);
- the finite-time average consensus algorithms based on eigenvalues cancellation proposed independently in [9] and in [40], which guarantee that the average consensus is always reached in exactly $T = N_d - 1$ rounds. Since the cancellation order impacts the time evolution of the residual errors experienced by agents at intermediate rounds before the exact convergence is reached, to obtain a benchmark (not implementable in real networks), we consider a greedy cancellation order through an exhaustive search by choosing at each round the eigenvalue to cancel which yields the maximum reduction of the residual mean absolute error (over all agents). In the following, the implementation adopting this policy will be named finite-time optimal eigenvalues cancellation (F-OEC). We also consider a more realistic case in which all agents adopt a random order for the eigenvalue cancellation (the same for each agent) denoted as finite-time random eigenvalues cancellation (F-REC). Such schemes are compared with our F-MS and F-MSwP algorithms (Sec. VI).

For comparison purposes, we also include the results obtained using the classic network coding (NC) algorithm [44]–[46], which is here oriented to the average consensus calculation and denoted finite-time network coding consensus (F-NCC) [43]. In particular, instead of retrieving each single measurement, as in the case of classic NC, our F-NCC is aimed at deriving the average of all measurements (*i.e.*, the average consensus) in a finite amount of time.

However, one observes that F-NCC requires the transmission of an overhead (the NC header containing the coding coefficients), which is in contrast with the basic "one scalar" rule that holds for consensus algorithms. In fact, *pure* consensus algorithms are required to transmit only one scalar per round per each agent. As will be discussed in the following, the overhead is the price to be paid to make this algorithm resilient to transmission errors. As done for the F-MS algorithm, also in this case we consider an enhanced version of F-NCC which starts propagating the consensus value as soon as one node converges. The F-NCC adopting this policy will be denoted finite-time network coding consensus with propagation (F-NCCwP) in the following.

Regarding performance indicators, one observes that the convergence speed may not be the only figure to be considered as, for practical implementations, also *a priori* information requirements on network topology, memory usage, and transmission overhead might play a fundamental role in the choice of the particular algorithm to be adopted.

As for the *a priori* availability of network related information, at the beginning of the paper we assumed that each agent knows the adjacency matrix **A**, which means that all agents are aware of the network topology. This assumption is well suited for static networks, in which the network discovery phase,

aimed at retrieving **A**, is rarely executed by agents. In this case, the corresponding signaling overhead could be negligible with respect to the amount of side information that should be transmitted by the different consensus algorithms when **A** is not available. Owing to space limitations this aspect is not discussed here, but some results on how network topology can be inferred can be found in [47].

Nevertheless, the full knowledge of **A** is not strictly needed for all the investigated algorithms, as some of them require less information. In Table I, a comparison between the algorithms discussed in this paper is reported in terms of required *a priori* knowledge about the network topology, memory usage at each agent, and transmission overhead at each round.

Other important practical issues are related to the robustness to transmission errors, that will be investigated in the next section, and the sensitivity to numerical precision. In particular, an outcome of our investigations which is worth mentioning, although not discussed here due to lack of space, indicates that finite-time algorithms may suffer from numerical precision issues. Those based on eigenvalue cancellation, for instance, are extremely critical, as they show diverging behaviors for low degrees of network connectivity, as also mentioned in [9].

C. Simulation results

Fig. 3(a) shows the CDFs of the ε-convergence time for the asymptotic algorithms A-CEW and A-Metro, as well as for their memory-enhanced versions A-FIR-CEW and A-FIR-Metro with $M = 1$, $M = 2$ and $M = 3$. We consider a transmission coverage range $R = 0.25$, and no transmission losses ($P_{err} = 0$). Further, Fig. 3(a) shows the CDF of the network diameter, which represents a lower bound for the CDF of the convergence time. One observes that exploiting the memory has a beneficial impact on the ε-convergence time, which reduces for increasing values of M. Notably, Fig. 3(a) also shows that increasing the memory from $M = 2$ to $M = 3$ provides only a marginal improvement, which clearly highlights a saturation effect on the gain provided by additional contributions from the memory.

Focusing the attention on the best performing A-FIR-CEW and A-FIR-Metro algorithms (i.e., those adopting $M = 3$), Fig. 3(b) provides a comparison with A-IIR and A-Kruzik. In this regards, one observes that A-FIR-Metro with $M = 3$ achieves almost the same performance provided by A-IIR [38] (which exploits memory too) and slightly outperforms A-Kruzik [33], which operates with the same amount of memory.[3]

Finite-time algorithms are investigated under the same conditions ($R = 0.25$, $P_{err} = 0$) in Fig.4(a). Firstly, one notices the significant impact of the order of eigenvalue cancellation in F-OEC and F-REC. As expected, F-REC performs worse than the optimal F-OEC, even though the latter cannot be implemented in practice. The same figure also shows the CDFs obtained with F-NCC and F-NCCwP as well as with our F-MS and F-MSwP algorithms. As for the settings of

---

[3]The parameter d in [33] defines how many past rounds must be considered to derive the future state. With respect to the parameter M of our algorithm, it is $M = d - 1$.

Algorithm	Asympt/Finite Time	A-priori Knowledge	Stored Scalars	Convergence	Transmission overhead
A-CEW [6]	Asymptotic	λ_2 and λ_N	$Neigh + 1$	yes	no
A-FIR-CEW with memory $M = 1$	Asymptotic	λ_2 and λ_N	$2Neigh + 1$	yes	no
A-FIR-CEW with memory $M > 1$	Asymptotic	Network topology	$Neigh + 1 + MNeigh$	yes	no
A-Metro [6]	Asymptotic	Local topology	$Neigh + 1$	yes	no
A-FIR-Metro with memory $M = 1$	Asymptotic	λ_2 and λ_N	$2Neigh + 1$	yes	no
A-FIR-Metro with memory $M > 1$	Asymptotic	Network topology	$Neigh + 1 + MNeigh$	yes	no
A-IIR [38]	Asymptotic	λ_2 and λ_N	$Neigh + 4$	yes	no
A-Kruzik [33] applied to A-Metro	Asymptotic	Network Topology	$Neigh + d$	yes	no
F-OEC/F-REC [9]	Finite Time	Network Topology	$Neigh + 1$	yes	no
F-MS - F-MSwP	Finite Time	Network Topology	$Neigh \times Rounds$	Not granted	no
F-NCC - F-NCCwP	Finite Time	None	$Neigh \times Rounds$	Not granted	yes

TABLE I

COMPARISON OF CONSENSUS ALGORITHMS. $Neigh$ DENOTES THE NUMBER OF NEIGHBOURS OF THE GENERIC AGENT.
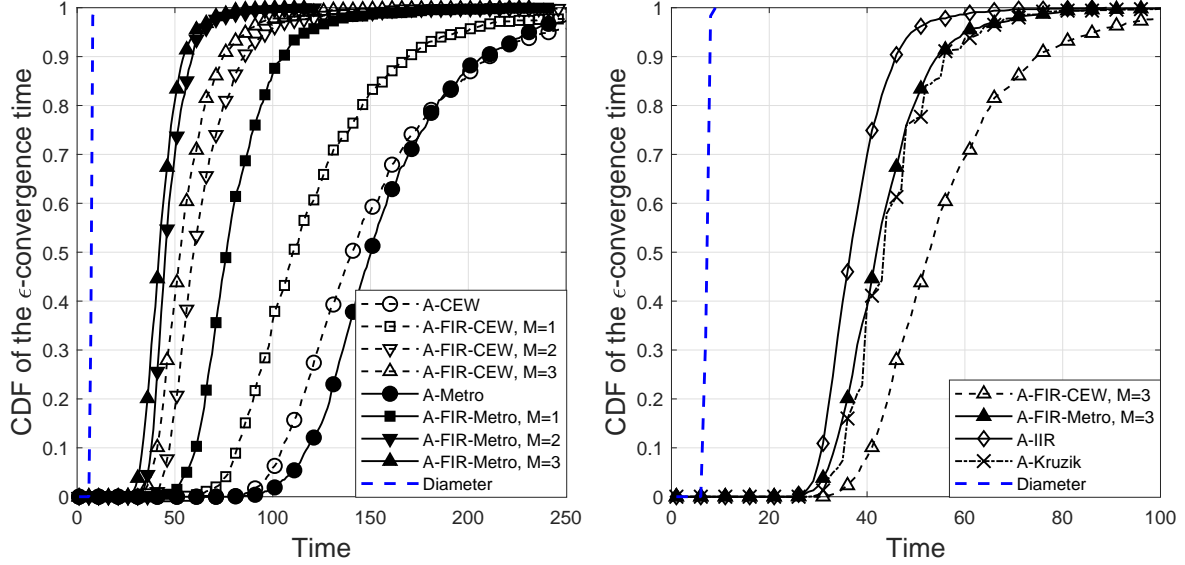
(a) Asymptotic CEW and Metropolis consensus schemes with their enhanced versions, $R = 0.25$, $P_{err} = 0$.

(b) Comparison of A-IIR and A-Kruzik with A-FIR-CEW and A-FIR-Metro when $M = 3$, $R = 0.25$, $P_{err} = 0$.

Fig. 3. CDF of the ε-convergence time, $\varepsilon = 10^{-4}$, $N = 100$, $N_R = 1000$ network realizations.
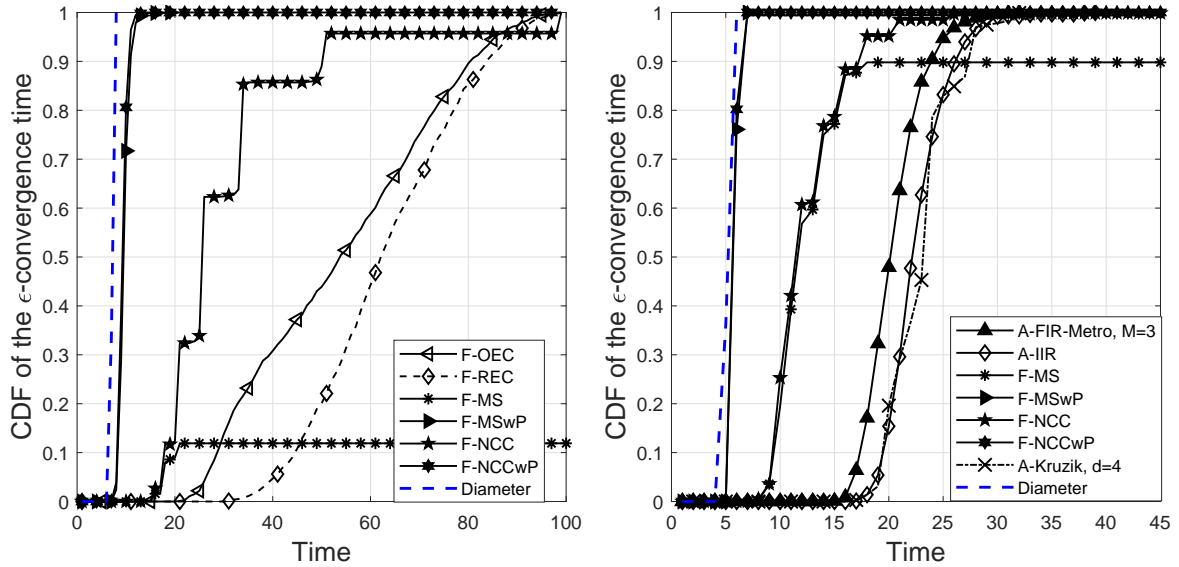
(a) Finite-time consensus schemes, $R = 0.25$, $P_{err} = 0$.

(b) Mixed consensus schemes, $R = 0.35$, $P_{err} = 0$.

Fig. 4. CDF of the ε-convergence time, $\varepsilon = 10^{-4}$, $N = 100$, $N_R = 1000$ network realizations.

F-NCC and F-NCCwP, they both perform linear combinations with random coefficients belonging to GF(2^8), the Galois field containing 2^8 elements. One observes, first of all, the good performance exhibited by F-NCC, which leads all agents to
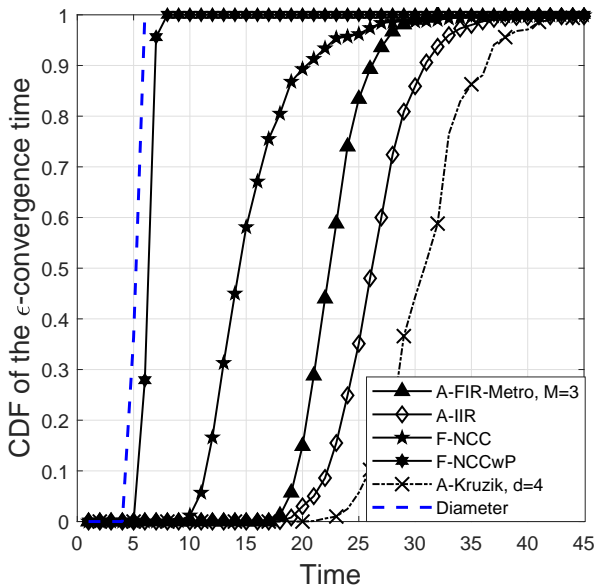
Fig. 5. Mixed consensus schemes, $R = 0.35$, $P_{err} = 0.2$.

the consensus in 100% of cases, whereas F-MS achieves the network convergence only in less than 15% of cases. Such a gap is completely recovered when the consensus propagation strategy is enabled. In this regard, the convergence speed-up provided by the consensus propagation strategy is impressive, as the CDF of both F-NCCwP and F-MSwP is very close to the lower bound represented by the CDF of the network diameter D. This is an important result because most finite-time algorithms achieve consensus after $N_d - 1$ rounds [9] or more [11]. Considering that in most network realizations the number N_d of distinct eigenvalues (usually close to N) is much larger than D, alternative schemes, such as the proposed F-MSwP, might provide a significant performance gain.

Comparing Figs. 3(a) and 3(b) with Fig.4(a), it is worth noticing that many of the investigated asymptotic algorithms reach the ε-convergence earlier than the F-REC and F-OEC finite-time schemes, which highlights that the ε-convergence time is certainly an appropriate metric to capture the convergence properties of heterogeneous consensus schemes.

Figure 4(b), when compared with Figs. 3(a), 3(b) and 4(a), allows to understand the role played by the coverage range R. It was derived by setting $R = 0.35$, $P_{err} = 0$, and considering only the best performing algorithms, namely, A-IIR, A-FIR-Metro with $M = 3$ and A-Kruzik, as representatives of the asymptotic consensus algorithms, and both F-NCC and F-MS, along with their enhanced versions F-NCCwP and F-MSwP, for the finite-time category.

Firstly, one observes that the increased coverage range R leads all algorithms to a faster convergence, which is expected owing to the increased connectivity of agents. Regarding the asymptotic algorithms, our A-FIR-Metro with $M = 3$ outperforms both A-IIR and A-Kruzik. Further investigations, not discussed here for space limitations, show that the superiority of A-FIR-Metro, when compared to A-IIR, increases for larger values of R.

As for the finite-time algorithms, F-MS exhibits a significant improvement thanks to the higher connectivity, driving all

agents to the consensus in 90% of network realizations, while F-NCC confirms its good performance, reaching the network convergence in 100% of cases. Remarkably, both algorithms, when enhanced with the consensus propagation capability, provide extraordinarily good performances, showing CDFs almost coincident with the lower bound.

The impact of transmission losses is investigated in Fig. 5, which was derived assuming $R = 0.35$ and $P_{err} = 0.2$. It can be noticed that the presence of link failures prevents the convergence to the consensus of F-OEC and F-REC, whose CDFs are therefore absent. In fact, the eigenvalue cancellation technique implicitly assumes that the network topology does not change over time, which is no more true in the presence of even temporary link failures (*i.e.*, broken links) that introduce modifications to the network graph. Similarly, transmission losses prevent also F-MS and F-MSwP from reaching the consensus, as the matrix $\mathbf{F}_n(k)$ used in (34) to derive the weights is no more consistent with the actual dissemination of data through the network, which is affected by losses.

As far as the remaining algorithms are concerned, we only considered the best performing ones, that is, A-IIR, our A-FIR-Metro with $M = 3$ and A-Kruzik, as representatives of the asymptotic consensus algorithms, and F-NCC along with F-NCCwP for the finite-time category.

One notices that link failures have a detrimental impact on the convergence speed of all algorithms, although to different extents. In fact, all curves shown in Fig. 5 are right-hand shifted with respect to the corresponding ones in Fig. 4(b). Regarding the asymptotic schemes, our A-FIR-Metro still outperforms A-IIR and A-Kruzik, also showing better resilience to transmission losses. In fact, comparing the results in Fig. 5 with those in Fig. 4(b), it turns out that A-IIR suffers a deterioration by 18.2% in terms of median ε-convergence time, which is almost twice the 10% worsening experienced by A-FIR-Metro. Clearly, even larger differences are observed with respect to A-Kruzik.

With reference to F-NCC and F-NCCwP, they reach the consensus in 100% of cases, thus proving to be not only the most rapidly converging schemes but also very robust against link failures. In particular, the consensus propagation technique makes F-NCCwP almost insensitive to transmission losses, which have a very little impact on the convergence speed. It has to be remarked, however, that this performance is achieved thanks to the transmission of the Network Coding header, which implicitly bears the information about possible transmission errors.

Clearly, this makes the comparison with the other algorithms unfair. In fact, the transmission of an header containing how the different measurements have been propagated and combined through the network, would make also F-MS and F-MSwP robust to transmission errors. In any case, if the "one-scalar" rule has to be observed, the asymptotic algorithms represent the only feasible choice and the proposed A-FIR-Metro algorithm, with $M = 3$, proved to be a good candidate.

VIII. CONCLUSION

In this paper a comparison of state-of-the-art and newly introduced asymptotic and finite-time consensus algorithms

has been performed adopting common performance metrics. It has been pointed out how the ε-convergence time and its CDF appear the most appropriate performance indexes for a fair comparison of heterogeneous algorithms and to analyze their performance under delay or residual error constraints.

A discussion on practical aspects about memory, *a priori* knowledge on network topology, transmission overhead and robustness to errors has been provided. The convergence properties of state-of-the-art as well as of the proposed algorithms have been investigated under different conditions in terms of network connectivity and transmission errors, by showing the benefits gained by letting agents exploit the memory. In particular, our finite-time consensus scheme has demonstrated to achieve a performance close to the theoretical limit given by the network diameter. However, in the presence of transmission errors finite-time schemes might fail in reaching the consensus and asymptotic algorithms result to be more appropriate as that proposed in this paper.

APPENDIX A. OPTIMAL d IN THE CASE $M = 1$

When $M = 1$, given (21) and (22), the constrained optimization problem (20) can be reformulated as

$$\mathbf{d}^* = (d_0^*, d_1^*) = \arg \min_{\mathbf{d}=(d_0,d_1)} \max_{m \in \{2,3,\ldots,N\}} \left\{ \left| z_m(d_0,d_1) \right| \right\}, \quad (36)$$

where

$$\left| z_m(d_0,d_1) \right| \triangleq \max \left\{ \left| z_{m1}(d_0,d_1) \right|, \left| z_{m2}(d_0,d_1) \right| \right\}. \quad (37)$$

As an example case, Fig. 6(a) shows $\left| z_m(d_0,d_1) \right|$ as a function of d_0 and d_1 when $\lambda_m = 3$. One observes the presence of left and right *wings*, which correspond to regions of the (d_0,d_1) plane where $z_m(d_0,d_1)$ is real, and of a *tail*, where $z_m(d_0,d_1)$ is complex. The boundary between the two regions is the parabola

$$d_1(d_0) = \frac{(1 - d_0\lambda_m)^2}{4\lambda_m} \quad \text{with } d_0 \neq \frac{1}{\lambda_m}. \quad (38)$$

In the tail, where $z_m(d_0,d_1)$ is complex, $\left| z_m(d_0,d_1) \right|$ has the following simple expression

$$\left| z_m(d_0,d_1) \right| = \sqrt{d_1\lambda_m}. \quad (39)$$

It also appears that the graph of $\left| z_m(d_0,d_1) \right|$ is symmetric with respect to the plane $d_0 = \frac{1}{\lambda_m}$.

To find the solution of (36) one has to firstly study the monotonicity of $\left| z_m(d_0,d_1) \right|$ in the (d_0,d_1) plane. For space limitation, the gradient of $\left| z_m(d_0,d_1) \right|$, whose evaluation is easy, is only reported in Fig.6(a) by arrows, indicating the directions along which $\left| z_m(d_0,d_1) \right|$ is monotonically increasing.

To compare the graphs of $\left| z_m(d_0,d_1) \right|$ for different values of m, one considers firstly

$$\left| \overleftarrow{z_m}(\overleftarrow{d_0},d_1) \right| = \left| z_m(d_0 = \overleftarrow{d_0} + \frac{1}{\lambda_m}, d_1) \right|$$
$$= \frac{1}{2} \left| \overleftarrow{d_0}\lambda_m \right| \left| 1 + \sqrt{1 - \frac{4d_1\lambda_m}{(\overleftarrow{d_0}\lambda_m)^2}} \right|, \quad (40)$$

$\forall (\overleftarrow{d_0},d_1) \in \mathbb{R}^2 \setminus \{0,0\}$.

Eq. (40) is a $1/\lambda_m$ translated version of $\left| z_m(d_0,d_1) \right|$. From (40) one deduces that for different values of m, the corresponding $\left| \overleftarrow{z_m}(\overleftarrow{d_0},d_1) \right|$ have the same symmetry plane $\overleftarrow{d_0} = 0$ and that $\forall (\overleftarrow{d_0},d_1) \in \mathbb{R}^2 \setminus \{0,0\}$

$$\text{if } \lambda_i < \lambda_j \Rightarrow \left| \overleftarrow{z_i}(\overleftarrow{d_0},d_1) \right| < \left| \overleftarrow{z_j}(\overleftarrow{d_0},d_1) \right|. \quad (41)$$

Consequently, (*i*) the graph of the different $\left| \overleftarrow{z_m}(\overleftarrow{d_0},d_1) \right|$ do not intersect and (*ii*) $\forall (\overleftarrow{d_0},d_1) \in \mathbb{R}^2 \setminus \{0,0\}$, $\left| \overleftarrow{z_N}(\overleftarrow{d_0},d_1) \right|$ is larger than $\left| \overleftarrow{z_m}(\overleftarrow{d_0},d_1) \right|$, $m = 2,\ldots,N-1$.

Considering now $\left| z_m(d_0,d_1) \right|$ for different values of m, one observes in Fig. 6(b) that intersections appear and the property $\left| z_N(d_0,d_1) \right| > \left| z_m(d_0,d_1) \right|$, $m = 2,\ldots,N-1$, does not hold anymore in the whole (d_0,d_1) plane. In this regard, one observes that the graph of $\left| z_m(d_0,d_1) \right|$ is a shifted version of the graph of $\left| \overleftarrow{z_m}(\overleftarrow{d_0},d_1) \right|$ along the d_0 axis by $1/\lambda_m$, and that this shift is larger for small values of m. Therefore, the left wings of the graphs of $\left| z_m(d_0,d_1) \right|$ for different values of m cannot intersect. Additionally, since the tail of the graph of $\left| z_m(d_0,d_1) \right|$ is constant along the d_0 axis for a given value of d_1 (see (39)), two tails for different values of m cannot intersect too. Consequently, considering the graphs of $\left| z_m(d_0,d_1) \right|$ and $\left| z_{m'}(d_0,d_1) \right|$, with $m \neq m'$, intersections may only exist between a left and a right wing, between a right wing and a tail, or between two right wings, as evident in Fig.6(b).

The solution $\mathbf{d}^* = (d_0^*, d_1^*)$ of (36) belongs to the set of (d_0,d_1) where the different $\left| z_m(d_0,d_1) \right|$ are equal to $\left| z_N(d_0,d_1) \right|$, *i.e.*, where the graph of $\left| z_N(d_0,d_1) \right|$ intersects the graphs of $\left| z_m(d_0,d_1) \right|$. The right wing of $\left| z_m(d_0,d_1) \right|$ is monotonically decreasing when d_0 and d_1 increase. Moreover, the left wing of $\left| z_N(d_0,d_1) \right|$ is monotonically increasing when d_0 increases and monotonically decreasing when d_1 increase. Consequently, the solution for any given $m = 2,\ldots,N-1$ of the following problem
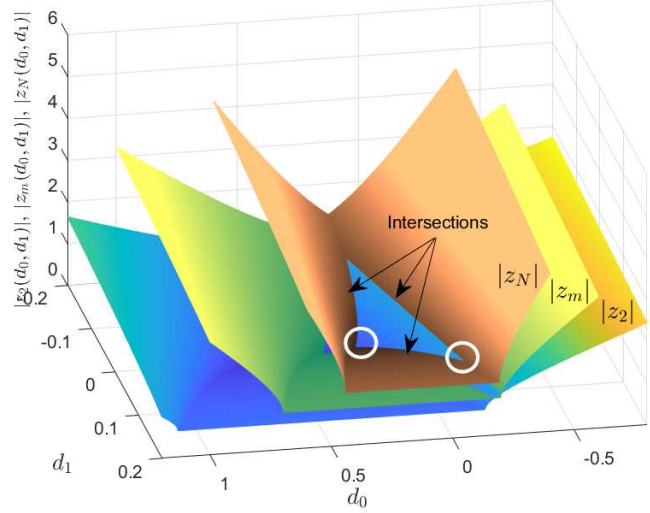
$$\left(\widehat{d}_0^{(m)}, \widehat{d}_1^{(m)} \right) = \arg \min_{\mathbf{d}=(d_0,d_1)} \max \left\{ \left| z_m(d_0,d_1) \right|, \left| z_N(d_0,d_1) \right| \right\},$$

is either found (*i*) at the point intersection $\left(d_0^{(1)}(\lambda_m), d_1^{(1)}(\lambda_m) \right)$ of three surfaces, namely, the right wing of $\left| z_m(d_0,d_1) \right|$, the right wing of $\left| z_N(d_0,d_1) \right|$, and the tail of $\left| z_N(d_0,d_1) \right|$ (see the circle on the right side of Fig. 6(b) in the case $m = 2$) or (*ii*) at the point intersection $\left(d_0^{(2)}(\lambda_m), d_1^{(2)}(\lambda_m) \right)$ of the right wing of $\left| z_m(d_0,d_1) \right|$ with the left wing of $\left| z_N(d_0,d_1) \right|$ and the tail of $\left| z_N(d_0,d_1) \right|$ (see the circle on the left side of Fig. 6(b) in the case $m = 2$).

In fact, since the solution of (36) corresponds to the minimum of $\max \left\{ \left| z_m(d_0,d_1) \right| \right\}_{m=2}^{N}$, one focuses on the twol point intersections that belong to the tail of $\left| z_N(d_0,d_1) \right|$, where the lowest values of $\max \left\{ \left| z_m(d_0,d_1) \right| \right\}_{m=2}^{N}$ are obtained. One observes that for each m both candidates belong to the curve defined by the intersection between the right wing
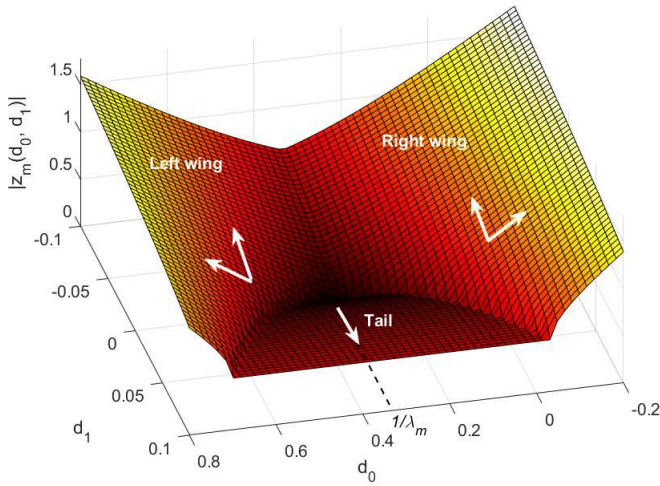
(a) $|z_m(d_0, d_1)|$ with $\lambda_m = 3$. The arrows indicate the directions along which $|z_m(d_0, d_1)|$ is monotonically increasing.

(b) $|z_2(d_0, d_1)|$, $|z_m(d_0, d_1)|$, $|z_N(d_0, d_1)|$ given $\lambda_2 < \lambda_m < \lambda_N$.

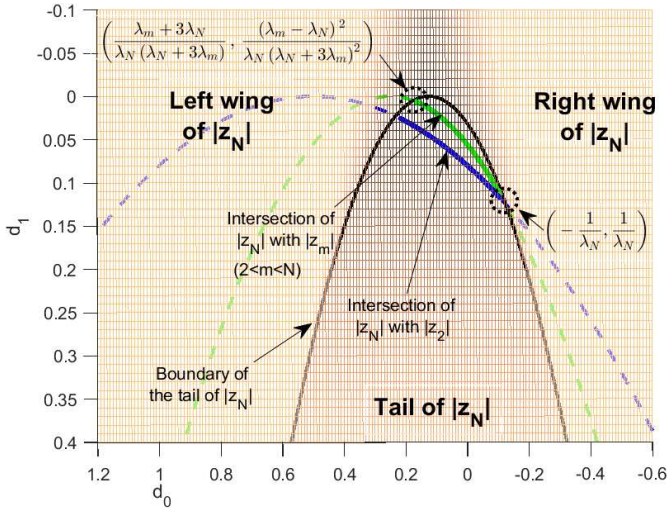Fig. 6. Graphical representations of $|z_m(d_0, d_1)|$.

Fig. 7. Projection of $|z_N(d_0, d_1)|$ on the (d_0, d_1), plane and intersections of its tail with $|z_2(d_0, d_1)|$ (blue solid line) and $|z_m(d_0, d_1)|$ (green solid line) when $\lambda_2 < \lambda_m < \lambda_N$.

of $|z_m(d_0, d_1)|$, $m = 2, \ldots, N-1$, and the tail of $|z_N(d_0, d_1)|$, which is obtained by solving $|z_m(d_0, d_1)| = \sqrt{d_1 \lambda_N}$ to get

$$d_1 = \frac{\lambda_N(1 - d_0 \lambda_m)^2}{(\lambda_m + \lambda_N)^2} \tag{42}$$

$$\text{subject to } d_1 \geq \frac{(1 - d_0 \lambda_N)^2}{4\lambda_N}, \tag{43}$$

where (43) limits the validity of (42) to the tail[4] of $|z_N(d_0, d_1)|$. The set $\mathcal{P}^{(m)}$ of points (d_0, d_1) satisfying (42) and (43) is a part of a parabola.

The projection of $|z_N(d_0, d_1)|$ on the (d_0, d_1) plane is represented in Figure 7, where darker regions corresponds to lower values of $|z_N(d_0, d_1)|$. Given (42) and (43), Fig. 7 also

[4]The right-hand side of (43) is the boundary of the tail of $|z_N(d_0, d_1)|$ (see (38)).

shows $\mathcal{P}^{(2)}$ (blue solid curve) and $\mathcal{P}^{(m)}$ (green solid curve) with $\lambda_2 < \lambda_m < \lambda_N$.

The parabola defined by (38) and $\mathcal{P}^{(m)}$ intersect at

$$\left(d_0^{(1)}(\lambda_m), d_1^{(1)}(\lambda_m)\right) = \left(-\frac{1}{\lambda_N}, \frac{1}{\lambda_N}\right) \tag{44}$$

independently of m and at

$$\left(d_0^{(2)}(\lambda_m), d_1^{(2)}(\lambda_m)\right) = \left(\frac{\lambda_m + 3\lambda_N}{\lambda_N(\lambda_N + 3\lambda_m)}, \frac{(\lambda_m - \lambda_N)^2}{\lambda_N(\lambda_N + 3\lambda_m)^2}\right). \tag{45}$$

The above introduced point intersections are highlighted by circles in Fig. 7 for the generic m. One may easily verify that

$$d_0^{(2)}(\lambda_2) > d_0^{(2)}(\lambda_m) \, \forall m = 3, \ldots, N-1.$$

Moreover, for all $(d_0, d_1) \in \mathcal{P}^{(m)}$, $m = 2, \ldots, N-1$, combining (39) and (42), one gets

$$|z_m(d_0, d_1)| = \frac{\lambda_N(1 - d_0 \lambda_m)}{\lambda_2 + \lambda_N}. \tag{46}$$

Consequently, for all $d_0 \in \left[d_0^{(1)}(\lambda_m), d_0^{(2)}(\lambda_m)\right]$, using (46), one may easily show that

$$\left|z_2\left(d_0, \frac{\lambda_N(1 - d_0 \lambda_2)^2}{(\lambda_2 + \lambda_N)^2}\right)\right| \geq \left|z_m\left(d_0, \frac{\lambda_N(1 - d_0 \lambda_m)^2}{(\lambda_m + \lambda_N)^2}\right)\right|, \tag{47}$$

where equality holds only for $d_0 = d_0^{(1)}$.

Given (47) and considering the monotonic increase of $|z_2(d_0, d_1)|$ for decreasing values of d_0 and d_1 (that is, in the directions of the other intersections), one concludes that the minimum of all maxima required by (36) belongs to the intersection between $|z_2(d_0, d_1)|$ and the tail of $|z_N(d_0, d_1)|$. This proves that (d_0^*, d_1^*) only depends on λ_2 and λ_N. Moreover, since the derivative of (46) with respect to d_0 is negative, (46) is monotonically decreasing as d_0 increases from $d_0^{(1)}$ to $d_0^{(2)}(\lambda_2)$. It follows that the minimum in (36) is achieved at

$$\left(d_0^{(2)}(\lambda_2), d_1^{(2)}(\lambda_2)\right) = \left(\frac{\lambda_2 + 3\lambda_N}{\lambda_N\left(\lambda_N + 3\lambda_2\right)}, \frac{\left(\lambda_2 - \lambda_N\right)^2}{\lambda_N\left(\lambda_N + 3\lambda_2\right)^2}\right).$$
(48)

REFERENCES

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[2] N. A. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers, San Francisco, 1996.

[3] Lin Xiao, Stephen Boyd, and Seung-Jean Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33 – 46, 2007.

[4] R. O. Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proc. American Control Conference*, June 2003, vol. 2, pp. 951–956.

[5] 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, Sept 2004.

[6] Lin Xiao and Stephen Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[7] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Intern. Symp. on Information Processing in Sensor Networks*, April 2005, pp. 63–70.

[8] S. Sundaram and C. N. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," in *2007 American Control Conference*, July 2007, pp. 711–716.

[9] A. Y. Kibangou, "Graph laplacian based matrix design for finite-time distributed average consensus," in *2012 American Control Conference*, June 2012, pp. 1901–1906.

[10] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan 2009.

[11] A. Sandryhaila, S. Kar, and J. M. F. Moura, "Finite-time distributed consensus through graph filters," in *2014 IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 1080–1084.

[12] J. M. Hendrickx, G. Shi, and K. H. Johansson, "Finite-time consensus using stochastic matrices with positive diagonals," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 1070–1073, April 2015.

[13] Z. Wang and C. Jin Ong, "Speeding up finite-time consensus via minimal polynomial of a weighted graph - a numerical approach," *Automatica*, vol. 93, pp. 415 – 421, 2018.

[14] J. Gutierrez-Gutierrez, M. Zarraga-Rodriguez, and X. Insausti, "Analysis of known linear distributed average consensus algorithms on cycles and paths," *Sensors*, vol. 18, no. 4, 2018.

[15] D. Blatt and A. Hero, "Distributed maximum likelihood estimation for sensor networks," in *2004 IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing*, May 2004, vol. 3, pp. iii–929.

[16] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[17] A. Ribeiro, I. D. Schizas, S. I. Roumeliotis, and G. Giannakis, "Kalman filtering in wireless sensor networks," *IEEE Control Systems Magazine*, vol. 30, no. 2, pp. 66–86, April 2010.

[18] S. Das and J. M. F. Moura, "Distributed Kalman filtering with dynamic observations consensus," *IEEE Trans. on Signal Processing*, vol. 63, no. 17, pp. 4458–4473, Sept 2015.

[19] O. Hlinka, O. Sluciak, F. Hlawatsch, P. M. Djuric, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4334–4349, Aug 2012.

[20] G. Notarstefano and F. Bullo, "Distributed abstract optimization via constraints consensus: Theory and applications," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2247–2261, Oct 2011.

[21] S. X. Wu, H. Wai, L. Li, and A. Scaglione, "A review of distributed algorithms for principal component analysis," *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1321–1340, Aug 2018.

[22] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, *Wireless Sensor and Actuator Networks: technologies, analysis and design*, Elsevier Ltd, London, 2008.

[23] S. Kar and J. M. F. Moura, "Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 99–109, May 2013.

[24] Gianni Pasolini, Anna Guerra, Francesco Guidi, Nicoló Decarli, and Davide Dardari, "Crowd-based cognitive perception of the physical world: Towards the Internet of Senses," *Sensors*, vol. 20, no. 9, 2020.

[25] Gianni Pasolini, Chiara Buratti, Luca Feltrin, Flavio Zabini, Cristina De Castro, Roberto Verdone, and Oreste Andrisano, "Smart city pilot projects using LoRa and IEEE802.15.4 technologies," *Sensors*, vol. 18, no. 4, 2018.

[26] O. Andrisano *et al.*, "The need of multidisciplinary approaches and engineering tools for the development and implementation of the smart city paradigm," *Proceedings of the IEEE*, vol. 106, no. 4, pp. 738–760, 2018.

[27] S. Li, G. Oikonomou, T. Tryfonas, T. M. Chen, and L. D. Xu, "A distributed consensus algorithm for decision making in service-oriented Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1461–1468, 2014.

[28] E. Cinque, H. Wymeersch, C. Lindberg, and M. Pratesi, "Toward a standard-compliant implementation for consensus algorithms in vehicular networks," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.

[29] S. Muthukrishnan, B. Ghosh, and M. H. Schultz, "First- and second-order diffusive methods for rapid, coarse, distributed load balancing," *Theory of Computing Systems*, vol. 31, no. 4, 1998.

[30] B. Johansson and M. Johansson, "Faster linear iterations for distributed averaging," in *Proc. 17th IFAC World Congress*, 2008.

[31] B. N. Oreshkin, M. J. Coates, and M. G. Rabbat, "Optimization and analysis of distributed averaging with short node memory," *IEEE Trans. on Signal Processing*, vol. 58, no. 5, pp. 2850–2865, May 2010.

[32] Ji Liu and A. Stephen Morse, "Accelerated linear iterations for distributed averaging," *Annual Reviews in Control*, vol. 35, no. 2, pp. 160 – 165, 2011.

[33] S. Kruzick and J. M. F. Moura, "Optimal filter design for signal processing on random graphs: Accelerated consensus," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1258–1272, March 2018.

[34] E. Kokiopoulou and P. Frossard, "Polynomial filtering for fast convergence in distributed consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 342–354, Jan 2009.

[35] E. Montijano, J. I. Montijano, and C. Sagues, "Chebyshev polynomials in distributed consensus applications," *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 693–706, Feb 2013.

[36] A. Sarlette, "Adding a single state memory optimally accelerates symmetric linear maps," *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3533–3538, Nov 2016.

[37] B. Van Scoy, R. A. Freeman, and K. M. Lynch, "Design of robust dynamic average consensus estimators," in *Proc. 54th IEEE Conf. on Decision and Control (CDC)*, Dec 2015, pp. 6269–6275.

[38] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 40–72, June 2019.

[39] Y. Yuan, G.-B. Stan, L. Shi, M. Barahona, and J. Goncalves, "Decentralised minimum-time consensus," *Automatica*, vol. 49, no. 5, pp. 1227 – 1235, 2013.

[40] S. Safavi and U. A. Khan, "Revisiting finite-time distributed algorithms via successive nulling of eigenvalues," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 54–57, Jan 2015.

[41] C. Ko and X. Gao, "On matrix factorization and finite-time average-consensus," in *Proc. 48th IEEE Conf. on Decision and Control (CDC)*, Dec 2009, pp. 5798–5803.

[42] S. Apers and A. Sarlette, "Accelerating consensus by spectral clustering and polynomial filters," *IEEE Trans. on Control of Network Systems*, vol. 4, no. 3, pp. 544–554, Sep. 2017.

[43] A. Calisti, D. Dardari, G. Pasolini, and M. Kieffer, "Exploiting node memory for finite-time average consensus over WSNs," in *2018 IEEE 29th Annual Intern. Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2018, pp. 1–7.

[44] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of the 41st Allerton Conf.*, Monticello, IL, 2003.

[45] L. Keller, E. Atsan, K. J. Argyraki, and C. Fragouli, "Sensecode: Network coding for reliable sensor networks," *TOSN*, vol. 9, no. 2, pp. 25, 2013.

[46] Y.D. Bromberg, Q. Dufour, and D. Frey, "Multisource Rumor Spreading with Network Coding," in *INFOCOM 2019 - IEEE Intern. Conf. on Computer Communications*, Paris, France, Apr. 2019, pp. 1–10, IEEE.

[47] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.