Accelerated Multi-Stage Discrete Time Dynamic Average Consensus

Eduardo Sebastián Eduardo Montijano Carlos Sagüés Mauro Franceschelli Andrea Gasparri

***Abstract*— This paper presents a novel solution for the discrete time dynamic average consensus problem. Given a set of time-varying input signals over the nodes of an undirected graph, the proposed algorithm tracks, at each node, the average of the input signals. Recently, a discrete time multi-stage consensus algorithm has been proposed, where the average is computed through a sequence of consensus stages such that no global knowledge, k-th order differences of the inputs nor conservation of the network state average is needed to achieve convergence. The counterpart is a slow convergence and a significant steady-state error, specially at the initial consensus stages. To overcome these issues, we present a second order diffusive protocol based on the multi-stage consensus filter that, by only storing an additional value at each node, accelerates the convergence speed and improves the steady-state error in orders of magnitude. The protocol is extended to an asynchronous and randomized version that follows a gossiping scheme. We analytically study the convergence properties of the algorithms, validating the proposal through simulated experiments.**

I. Introduction

The problem of *consensus* in control theory [1] consists of finding a protocol such that a set of nodes in a network agree on the value of a certain quantity of interest. The consensus problem has a long history due to its relevance in applications such as smart grids [2] or tracking by sensor networks [3].

In particular, the discrete time dynamic average consensus [4] is interesting because the tracked signals usually evolve with time, and protocols are implemented in computing units that work in discrete steps. Several solutions have been proposed to improve the capabilities of discrete time dynamic average consensus algorithms. For instance, some works achieve an arbitrarily small steady-state error by exploiting the k-th differences of the input signal [5], [6]. This is problematic when the inputs are noisy since noise breaks the boundedness of the input signal. An alternative is to rely only on the input signal, achieving an arbitrarily small steady-state error by concatenating a cascade of consensus filters [7]–[9], as we do in this work. The counterpart is a slow convergence in both the continuous and discrete time consensus.

This work was supported by ONR Global grant N62909-19-1-2027, the Spanish projects PID2021-125514NB-I00, PID2021-124137OB-I00, TED2021-130224B-I00 funded by MCIN/AEI/10.13039/501100011033, by ERDF A way of making Europe and by the European Union NextGenerationEU/PRTR, by the Gobierno de Aragón under Project DGA T45-20R, and by Spanish grant FPU19-05700.

E. Sebastián, E. Montijano and C. Sagüés are with the Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain (email:esebastian@unizar.es, emonti@unizar.es, csagues@unizar.es)

M. Franceschelli is with the Department of Electrical and Electronic Engineering, University of Cagliari, Italy (email: mauro.franceschelli@unica.it).

A. Gasparri is with the Department of Engineering, Roma Tre University, Italy (email:gasparri@dia.uniroma3.it).

The issue of slow convergence has been addressed from two main perspectives. Ghosh et al. [10] present second order diffusion methods. By storing $x_i(k-1)$, the convergence speed can be significantly improved. This is formally studied by Liu and Morse [11], providing the optimal parameters and the corresponding fastest convergence rate. Since then, other authors have proposed variants of the same protocol [12], all of them achieving a similar performance [13]. These works address the static consensus problem, where the input signals do not vary with time. In contrast, our work deals with the more challenging problem of dynamic consensus. Polynomial filters [14] are the second relevant research line to accelerate consensus. Polynomial filters consider a sequence of p consensus iterations as the evaluation of a polynomial of order $p-1$. Montijano et al. [15] analyze Chebyshev polynomials and prove that they reduce to a second order difference equation for an arbitrary p, while the convergence speed is significantly increased. The robust and dynamic version of the protocol is also studied [6]. However, some mild knowledge on topology is required, and the k-th order differences of the input signal are exploited. This work opts for a second order method instead of a polynomial method to speed up convergence. Closely related to our paper, Van Scoy et al. [16] present a fast and robust discrete time dynamic average consensus estimator. Compared to them, our proposal does not need bounded inputs nor a known bound on the signal values, while achieving the same robustness against initialization errors and accelerated convergence.

From a different perspective, many applications deal with time-varying networks, where nodes connect and disconnect depending on events exogenous to the consensus protocol. Examples can be found in, e.g., robotics [17] or smart grids control [18]. To account for this, *gossip* algorithms study consensus when the protocols are computed at random instants of time with random neighbors. There also exist deterministic gossip algorithms [19] where the intermittent communication is decided by a deterministic rule [20], in an event-triggered fashion. This motivates our second proposed algorithm, which is an accelerated version of a discrete time dynamic average consensus protocol that runs asynchronously and with randomized communication links.

This paper builds from the works by Franceschelli and Gasparri [8], [9]. They present a multi-stage discrete time dynamic average consensus filter that is fully distributed, does not use the k-th order differences of the inputs signals, and is robust again non-averaged nodes' initialization. Besides, the steady-state error can be arbitrarily reduced by increasing the number of stages. The main limitation

. Preprint Received March 16, 2023 05:53:03 Pacific Time

is the slow convergence, specially in the last stages of the filter, so there is a trade-off between accuracy and speed. To mitigate this issue, we propose a second order method to accelerate the convergence, which permits to either increase the number of stages (and reduce the steady-state error) with the same convergence speed of the original filter, or speed up the convergence for the same number of stages while maintaining the steady-state error. Our main contributions are two algorithms: the accelerated multi-stage discrete time dynamic average consensus, and an asynchronous and randomized version of the former with their associated formal convergence guarantees.

II. Preliminaries

The system[1] under study is a network composed by $N > 1$ nodes. The network is described by an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, \dots, i, \dots, N\}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Nodes i and j can exchange information if and only if $(i, j) \in \mathcal{E}$, which implies that $(j, i) \in \mathcal{E}$. The neighborhood of node i is $\mathcal{N}_i = \{j | (j, i) \in \mathcal{E}\}$. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ associated to $\mathcal{G}$ is such that $\mathbf{A}_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and 0 otherwise. $\mathbf{A}_{ii} = 0$ always because we do not allow self-loops. The degree matrix $\mathbf{D}$ associated to $\mathcal{G}$ is such that $\mathbf{D}_{ij} = \mathsf{car}(\mathcal{N}_i)$ $\forall i = j$ and 0 otherwise. The Laplacian matrix associated to $\mathcal{G}$ is $\mathbf{L} = \mathbf{D} - \mathbf{A}$. For undirected graphs, it holds that $\mathbf{L}$ is symmetric and has real eigenvalues. Besides, denote the sorted eigenvalues of $\mathbf{L}$ as $\lambda_1(\mathbf{L}) < \lambda_2(\mathbf{L}) \leq \ldots \leq \lambda_N(\mathbf{L}) \leq 2D_{\max}$, where $D_{\max} = \max(\{\mathbf{D}_{ii}\}_{i=1}^N)$. For a connected graph, $\lambda_1(\mathbf{L}) = 0$ with associated eigenvector $\mathbf{v}_1(\mathbf{L}) = \mathbf{1}$. The second smallest eigenvalue, $\lambda_2(\mathbf{L})$, is called *algebraic connectivity*. We denote $\mathbf{v}_{r,i}(\mathbf{L}), \mathbf{v}_{l,i}(\mathbf{L})$ the right and left eigenvectors of $\mathbf{L}$ associated to the i-th eigenvalue.

A. Second Order Diffusion Methods

The linear (first-order) discrete time *static* average consensus algorithm ([21]) is

$$x_i(k+1) = \mathbf{W}_{ii} x_i(k) + \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} x_j(k), \tag{1}$$

with $x_i(0)$ the initial condition and $\mathbf{W} \in \mathbb{R}^{N \times N}$ a weighted matrix. In matrix form, (1) leads to

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k). \tag{2}$$

If $\mathbf{W}$ is doubly stochastic, i.e., $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$ and $\mathbf{W}\mathbf{1} = \mathbf{1}$, then it is known that the protocol in Eq. (2) converges to the average of the initial states $\bar{x}_i(0) = (1/N)\sum_{i=1}^N x_i(0)$.

Without loss of generality, we consider $\mathbf{W} = \mathbf{I} - \epsilon\mathbf{L}$. The convergence speed can be too slow, specially in networks with a small algebraic connectivity ($\lambda_2(\mathbf{W}) = 1 - \epsilon\lambda_2(\mathbf{L})$). To accelerate convergence, Ghosh et al. ([10]) proposed a second-order modification:

$$\mathbf{x}(k+1) = \gamma\mathbf{W}\mathbf{x}(k) + (1-\gamma)\mathbf{x}(k-1). \tag{3}$$

The properties of the protocol in Eq. (3) are well-studied (e.g., [10] or [22]). Protocol (3) improves the convergence speed of (2) if $\gamma \in (1, 2)$, with the fastest convergence being $\gamma = \frac{2}{1+\sqrt{1-\lambda_2^2(\mathbf{W})}}$.

B. Multi-stage Discrete Time Dynamic Averaged Consensus

The concepts in Subsection II-A apply to the static consensus problem, but this work deals with a dynamic consensus problem. Node i has an input $r_i(k) \in \mathbb{R}$ and an estimate $x_i(k) \in \mathbb{R}$ associated to a quantity of interest $r(k) \in \mathbb{R}$. The joint estimate and input of the network are $\mathbf{x}(k) = [x_1(k), \dots, x_N(k)]^T$ and $\mathbf{r}(k) = [r_1(k), \dots, r_N(k)]^T$ respectively. Nodes, by means of $\mathbf{x}(k)$, cooperate to track the average $\bar{r}(k) = \frac{1}{N}\sum_{i=1}^N r_i(k)$ by exchanging information with their neighbors $j \in \mathcal{N}_i$.

The multi-stage discrete time dynamic average consensus algorithms presented by Franceschelli and Gasparri ([8], [9]) solve the problem for two cases: the synchronous and time-invariant topology, and the asynchronous and randomized topology. The former is solved by the following protocol:

$$\begin{aligned} \mathbf{x}^1(k+1) =& (\mathbf{I} - \epsilon\mathbf{L})\mathbf{x}^1(k) + \alpha(\mathbf{r}(k) - \mathbf{x}^1(k)), \\ \mathbf{x}^s(k+1) =& (\mathbf{I} - \epsilon\mathbf{L})\mathbf{x}^s(k) + \alpha(\mathbf{x}^{s-1}(k) - \mathbf{x}^s(k)). \end{aligned} \tag{4}$$

Here, $s = \{1, \dots, m\}$ denotes the stages of the filter, $\epsilon < \frac{1}{2D_{\max}}$ to ensure stability, and $\alpha \in (0, 1)$ is a parameter that trades-off steady-state accuracy and convergence speed. From (4) it is seen that the filter is a chain of linear discrete time dynamic average consensus protocols. The higher the stage, the lower the steady-state error and speed convergence.

The asynchronous and randomized algorithm follows the same multi-stage architecture. In this case, each node selects randomly and at each instant a single node $j \in \mathcal{N}_i$ to communicate. For a given edge (i, j), the algorithm is:

$$\begin{aligned} \mathbf{x}^1(k+1) =& \mathbf{P}_{ij}\mathbf{x}^1(k) + \frac{\alpha}{\mathbf{D}_{ii}}\mathbf{p}_i\mathbf{p}_i^T\mathbf{r}(k), \\ \mathbf{x}^s(k+1) =& \mathbf{P}_{ij}\mathbf{x}^s(k) + \frac{\alpha}{\mathbf{D}_{ii}}\mathbf{p}_i\mathbf{p}_i^T\mathbf{x}^{s-1}(k), \end{aligned} \tag{5}$$

with $\mathbf{P}_{ij} = \mathbf{I} + \frac{\mathbf{p}_i\mathbf{p}_j^T}{2} - \frac{(1+2/\mathbf{D}_{ii})\mathbf{p}_i\mathbf{p}_i^T}{2}$ and $\mathbf{p}_i \in \mathbb{R}^N$ a vector of zeros except for the $i-$th element, which is equal to 1.

III. Proposed consensus algorithm

In this section, we present two algorithms. The first solves the problem of discrete time dynamic average consensus by means of a second order recurrence and the multi-stage architecture inspired by [8], [9]. The second solves the same problem but under asynchronous and randomized restrictions, where the nodes randomly communicate in a gossip-like style [19].

[1]**Notation**: we use lower-case for scalars, bold lower-case for vectors, bold capital for matrices, and calligraphic capital for sets. We use $\geq$ for greater than or equal, $\succeq$ for positive semidefiniteness, $\sigma(\ )$ for the eigenvalues of a matrix, ^{T} for the transpose, $\mathsf{car}(\ )$ for the cardinality of a set, $\mathsf{E}[\ ]$ for the expectancy operator, $|\ |$ the absolute value, and $||\ ||$ for the 2-norm of a vector/matrix. $\mathbf{P}_{ij}$ (resp. $\mathbf{p}_i$) denotes the ij-th (resp. i-th) element of matrix $\mathbf{P}$ (resp. vector $\mathbf{p}$). $\mathbf{I}$ is the identity matrix of appropriate dimensions, $\mathbf{0}$ is the zero matrix of appropriate dimension, and $\mathbf{1}$ is a column vector of ones.

A. Accelerated Multi-Stage Filter

The first proposed consensus protocol is based on two steps. Given the current input $\mathbf{r}_i(k)$ and estimates from neighbors $\{x_j^s(k)\}_{s=0}^m \ \forall j \in \mathcal{N}_i \cup \{i\}$, node i updates $\{x_i^s(k)\}_{s=0}^m$ following the multi-stage filter in Eq. (4), obtaining $\{\tilde{x}_i^s(k)\}_{s=0}^m$, where $\tilde{x}_i^s(k) \in \mathbb{R}$ for $s = 1, \dots, m$ is a temporal estimate used in the second step of the protocol. The updated estimate is then corrected using the second order method in Eq. (3), leading to the next estimate $\{x_i^s(k+1)\}_{s=0}^m$. Algorithm 1 details the protocol. We remark that the division in two steps is to ease the intuition and the understanding of the protocol. Nevertheless, the two steps can be fused in a single one, as it is done in some of the following proofs.

Algorithm 1 Accelerated Multi-Stage Dynamic Consensus Protocol at node i

1: **State of agent**: $x_i^s(-1)$ and $x_i^s(0)$, for $s = 1, \dots, m$
2: **Parameters**: $\gamma \in (1, 2)$, $\epsilon \in (0, \frac{1}{2D_{\max}})$, $\alpha \in (0, \frac{1}{\gamma})$
3: **while** True **do**
4: Measure $r_i(k)$
5: Gather $x_j^s(k)$ for $s = 1, \dots, m$ and $j \in \mathcal{N}_i$
6: Update $x_i^s(k)$ for $s = 1, \dots, m$ as follows:

$$\tilde{x}_i^1(k) = x_i^1(k) - \sum_{j \in \mathcal{N}_i} \epsilon(x_i^1(k) - x_j^1(k)) + \alpha(r_i(k) - x_i^1(k))$$
$$\tilde{x}_i^s(k) = x_i^s(k) - \sum_{j \in \mathcal{N}_i} \epsilon(x_i^s(k) - x_j^s(k)) + \alpha(x_i^{s-1}(k) - x_i^s(k))$$
$$x_i^s(k+1) = \gamma(\tilde{x}_i^s(k)) + (1-\gamma)x_i^s(k-1) \text{ for } s = 1, \dots, m$$

7: **end while**

Note that Algorithm 1 is equal to (4) for $\gamma = 1$. First, for space convenience, in the following we use the sub-index $_k$ to abbreviate (k). We now prove the convergence properties of the proposed protocol. The next result shows that the second order method does not alter the steady-state properties of the original multi-stage filter in [8], [9].

Proposition 1. *Assume that $\mathbf{r}_k = \mathbf{r}$ is constant, $\mathcal{G}$ connected, $\alpha \in (0, 1)$, and $\epsilon \in (0, \frac{1}{2D_{\max}})$. Then, the steady-state equilibrium $\mathbf{x}_k^{m,*}$ at time k of the protocol in* (4) *and Algorithm 1 is*

$$\mathbf{x}_k^{m,*} = \bar{r}\mathbf{1} + \sum_{i=2}^{N} \left(\frac{1}{1 + \epsilon\lambda_2(\mathbf{L})/\alpha} \right)^m \mathbf{v}_{r,i}(\mathbf{L})\mathbf{v}_{l,i}^T(\mathbf{L})\mathbf{r} \quad (6)$$

Proof. From Algorithm 1 and using the matrix form, the operations at stage $s = 1$ and steady-state can be written in a single equation

$$\mathbf{x}_k^{1,*} = \gamma(\mathbf{I} - \epsilon\mathbf{L})\mathbf{x}_k^{1,*} + \gamma\alpha\mathbf{r} - \gamma\alpha\mathbf{x}_k^{1,*} + (1-\gamma)\mathbf{x}_k^{1,*}. \quad (7)$$

This leads to

$$((1 + \gamma\alpha - 1 + \gamma - \gamma)\mathbf{I} + \gamma\epsilon\mathbf{L})\,\mathbf{x}_k^{1,*} = \gamma\alpha\mathbf{r}. \quad (8)$$

Dividing both sides of Eq. (8) by γ, and concatenating the m stages,

$$\mathbf{x}_k^{m,*} = (\alpha\mathbf{I} + \epsilon\mathbf{L})^{-1}\alpha^m\mathbf{r}, \quad (9)$$

which is the steady-state equilibrium in the proof of Theorem 3.1 in [9]. The rest of the proof follows from there. □

Corollary 1. *Assume that $\mathbf{r}_k = \mathbf{r}$ is constant and $\mathcal{G}$ connected. Then, the 2-norm of the error at equilibrium of a network under Algorithm 1 is*

$$||\bar{r}\mathbf{1} - \mathbf{x}_k^{m,*}|| \leq (N-1)\left(\frac{1}{1 + \epsilon\lambda_2(\mathbf{L})/\alpha} \right)^m ||\bar{r}\mathbf{1} - \mathbf{r}||. \quad (10)$$

The statement is a direct consequence of Proposition 1 and Theorem 3.2 in [8].

Proposition 2. *Let $\mathbf{y}_k^s = \mathbf{x}_k^s - \mathbf{x}_k^{s,*}$ be the error at the s-th stage of the filter in Algorithm 1. Then, the error dynamics can be expressed as*

$$\begin{pmatrix} \mathbf{y}_{k+1}^s \\ \mathbf{y}_k^s \end{pmatrix} = \hat{\mathbf{Q}} \begin{pmatrix} \mathbf{y}_k^s \\ \mathbf{y}_{k-1}^s \end{pmatrix} + \alpha\hat{\mathbf{R}} \begin{pmatrix} \Delta\mathbf{u}_k^s \\ \Delta\mathbf{u}_{k-1}^s \end{pmatrix}, \quad (11)$$

where $\hat{\mathbf{Q}} = \begin{pmatrix} \gamma\mathbf{Q} & (1-\gamma)\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{pmatrix}$, $\hat{\mathbf{R}} = \begin{pmatrix} \mathbf{R} & \mathbf{R}(1-\gamma) \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$, $\mathbf{Q} = (1-\alpha)\mathbf{I} - \epsilon\mathbf{L}$, $\mathbf{R} = (\alpha\mathbf{I} + \epsilon\mathbf{L})^{-1}$, $\Delta\mathbf{u}_k^s = \mathbf{u}_k^s - \mathbf{u}_{k+1}^s$, $\mathbf{u}_k^1 = \mathbf{r}_k$ *and* $\mathbf{u}_k^s = \mathbf{x}_k^{s-1}$ *for* $s = 2, \dots, m$.

Proof. Using the steps in Algorithm 1 in a single operation and matrix form,

$$\begin{aligned} \mathbf{y}_{k+1}^s &= \gamma\mathbf{Q}\mathbf{y}_k^s + \gamma\mathbf{x}_k^{s,*} - \mathbf{x}_{k+1}^{s,*} + (1-\gamma)\mathbf{x}_{k-1}^s = \\ &\gamma\mathbf{Q}\mathbf{y}_k^s + \gamma\mathbf{x}_k^{s,*} - \mathbf{x}_{k+1}^{s,*} + (1-\gamma)\mathbf{y}_{k-1}^s + (1-\gamma)\mathbf{x}_{k-1}^{s,*} \end{aligned} \quad (12)$$

Eq. (12) can be rewritten using Eq. (9):

$$\mathbf{y}_{k+1}^s = \gamma\mathbf{Q}\mathbf{y}_k^s + (1-\gamma)\mathbf{y}_{k-1}^s + \alpha\mathbf{R}\Delta\mathbf{u}_k^s + (1-\gamma)\alpha\mathbf{R}\Delta\mathbf{u}_{k-1}^s. \quad (13)$$

Eq. (11) follows from Eq.(13), concluding the proof. □

Theorem 1. *Consider a network that executes Algorithm 1 with $\mathbf{r}_k = \mathbf{r}$ constant, with $\alpha \in (0, \frac{1}{\gamma})$, $\gamma \in (1, 2)$ and $\epsilon \in (0, \frac{1}{2D_{\max}})$. Then, the convergence rate for the s-th stage is $\beta^s = 1 - \alpha\gamma$.*

Proof. Let us consider the following Lyapunov function:

$$V_k^s = \begin{pmatrix} \mathbf{y}_k^{s,T} & \mathbf{y}_{k-1}^{s,T} \end{pmatrix} \begin{pmatrix} \mathbf{y}_k^{s,T} & \mathbf{y}_{k-1}^{s,T} \end{pmatrix}^T = \hat{\mathbf{y}}_k^s \hat{\mathbf{y}}_k^{s,T}, \quad (14)$$

with $\hat{\mathbf{y}}_k^s = \begin{pmatrix} \mathbf{y}_k^{s,T} & \mathbf{y}_{k-1}^{s,T} \end{pmatrix}$. Then, the Lyapunov difference is

$$\Delta V_k^s = V_{k+1}^s - V_k^s = \hat{\mathbf{y}}_{k+1}^s \hat{\mathbf{y}}_{k+1}^{s,T} - \hat{\mathbf{y}}_k^s \hat{\mathbf{y}}_k^{s,T}. \quad (15)$$

Exploiting Eq. (11) and the fact that $\mathbf{r}_k$ is constant, Eq. (15) leads to

$$\Delta V_k^s = \hat{\mathbf{y}}_k^s (\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T - \mathbf{I}) \hat{\mathbf{y}}_k^{s,T}. \quad (16)$$

Now, since $\gamma \in (1, 2)$ and the spectral radius of $\mathbf{Q}$ is less than 1 by construction, it can be shown that Eq. (16) is upper-bounded by the following inequality:

$$\Delta V_k^s \leq -\hat{\mathbf{y}}_k^s \begin{pmatrix} \gamma^2\mathbf{I} - \gamma^2\mathbf{Q}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \hat{\mathbf{y}}_k^{s,T} = -\hat{\mathbf{y}}_k^s \mathbf{F} \hat{\mathbf{y}}_k^{s,T}. \quad (17)$$

The eigenvalues of matrix $\mathbf{F}$ are the eigenvalues of $\gamma^2\mathbf{I} - \gamma^2\mathbf{Q}^2$ and $\mathbf{I}$. If $\alpha < \frac{1}{\gamma}$, then $\gamma^2\mathbf{I} - \gamma^2\mathbf{Q}^2 \succeq \gamma\mathbf{I} - \gamma\mathbf{Q}$ and $\sigma(\gamma\mathbf{I} - \gamma\mathbf{Q}) > 0$. Therefore, $\mathbf{F}$ is positive definite, and

$$V_{k+1}^s - V_k^s \leq -\bar{\lambda}\hat{\mathbf{y}}_k^s \hat{\mathbf{y}}_k^{s,T} = -\bar{\lambda} V_k^s. \quad (18)$$

$\bar{\lambda} = \min_{\lambda_i \in \sigma(\gamma\mathbf{I}-\gamma\mathbf{Q})\cup\sigma(\mathbf{I})}\{\lambda_i\} = \alpha\gamma$. Thus, $V^s_{k+1} \leq (1-\alpha\gamma)V^s_k$. □

Proposition 3. *Algorithm 1 always converges faster than the protocol in Eq.* (4) *if* $\gamma \in (1,2)$.

Proof. The convergence rate of the protocol in Eq. (4) ($\gamma = 1$) is $\beta^s = 1-\alpha$, which coincides with the convergence rate obtained by Franceschelli and Gasparri ([8]). It turns out that $1-\alpha > 1-\alpha\gamma$ for $\gamma \in (1,2)$. Thus, the bound in Eq. (18) is smaller, and the decrease in the Lyapunov function defined in Eq. (14) is greater, which implies a faster convergence of Algorithm 1 than the protocol in Eq. (4). □

Corollary 2. *The convergence rate is maximized for* $\gamma = \frac{2}{1+\sqrt{1-\lambda_2^2(\mathbf{W})}}$.

The corollary follows directly from the literature on accelerated consensus methods (Theorem 11.5 in [23]).

All the previous results address the case where the input is constant, i.e., the consensus is static. The following result considers a dynamic input.

Proposition 4. *Consider a time-varying input* $\mathbf{r}_k$ *and a network under Algorithm 1, with* $\alpha \in (0, \frac{1}{\gamma})$. *Besides, define* $\Delta\mathbf{u}^s_\infty = \sup_{k=0,\ldots,\infty} \Delta\mathbf{u}^s_k$. *Then, the next ISS property holds:*

$$||\mathbf{y}^s_k|| \leq (1-\gamma\alpha)^k ||\mathbf{y}^s_0|| + \frac{1}{\alpha\gamma}\Delta\mathbf{u}^s_\infty. \tag{19}$$

Proof. The result follows from Input-to-State Stability results for linear systems. From Eq. (13) we have that

$$||\mathbf{y}^s_k|| \leq ||\mathbf{K}^k|| \cdot ||\mathbf{y}^s_0|| + \alpha\left(\sum_{h=0}^{k} ||\mathbf{K}^h||||\mathbf{R}||\right)\Delta\mathbf{u}^s_\infty, \tag{20}$$

with $\mathbf{K} = (\gamma\mathbf{Q} + (1-\gamma)\mathbf{I})$. First, note that $||\mathbf{R}|| \leq \frac{1}{\alpha}$. Second, $||\mathbf{K}|| \leq (1-\gamma) + \gamma(1-\alpha) = 1-\gamma\alpha$. Then,

$$||\mathbf{y}^s_k|| \leq (1-\gamma\alpha)^k ||\mathbf{y}^s_0|| + \alpha\left(\sum_{h=0}^{k} \frac{(1-\gamma\alpha)^h}{\alpha}\right)\Delta\mathbf{u}^s_\infty. \tag{21}$$

Eq. (21) directly yields to Eq. (19). □ □

Therefore, under dynamic inputs the system still converges. Moreover, the convergence rate only depends on the variations of the signal.

Corollary 3. *The convergence rate of Algorithm 1 is faster than the convergence rate of the original filter in* (4) *under time-varying input signals* $\mathbf{r}_k$, $\alpha \in (0, \frac{1}{\gamma})$ *and* $\gamma \in (1,2)$.

Proof. In Proposition 3.5 from ([8]) it is shown that

$$||\mathbf{y}^s_k|| \leq (1-\alpha)^k ||\mathbf{y}^s_0|| + \frac{1}{\alpha}\Delta\mathbf{u}^s_\infty. \tag{22}$$

Comparing Eqs. (21) and (22), the first term experiences a faster decay in (21) since $1-\gamma\alpha < 1-\alpha$. The second term is lower in (21) since $\frac{1}{\alpha\gamma} < \frac{1}{\alpha}$. □

In summary, the proposed algorithm proves to enhance the convergence speed towards the dynamic average consensus without modifying the steady-state error at any of the stages of the filter. This is achieved by using an additional memory slot for the estimate at the previous instant of time. Therefore, the practitioner can either: (i) increase m to improve the steady-state error and maintain the settling time; (ii) decrease α to improve steady-state error, and maintain m (so maintaining the message size) and the settling time; (iii) or improve the convergence speed for the same m and α.

B. Accelerated Asynchronous Randomized Protocol

The second contribution is the asynchronous and randomized version of Algorithm 1. Now, instead of using all the estimates from the neighborhood, node i selects one of its neighbors $j \in \mathcal{N}_i$ according to an independent and identical distribution (i.i.d.) with uniform probability. The rest of the protocol follows the same reasoning, detailed in Algorithm 2.

Algorithm 2 Accelerated Asynchronous Randomized Multi-Stage Dynamic Consensus Protocol at node i

1: **State of agent**: $x^s_i(-1)$ and $x^s_i(0)$, for $s = 1,\ldots,m$
2: **Parameters**: $\gamma \in (1,2)$, $\alpha \in (0, \frac{1}{\gamma})$, $D_i = \mathbf{L}_{ii}$
3: **while** True **do**
4: Measure $r_i(k)$
5: Select at random one neighbor $j \in \mathcal{N}_i$ and gather $x^s_j(k)$
6: Update $x^s_i(k)$ for $s = 1,\ldots,m$ as follows:

$$\tilde{x}^1_i(k) = \frac{x^1_i(k) + x^1_j(k)}{2} + \frac{\alpha}{D_i}(r_i(k) - x^1_i(k))$$
$$\tilde{x}^s_i(k) = \frac{x^s_i(k) + x^s_j(k)}{2} + \frac{\alpha}{D_i}(x^{s-1}_i(k) - x^s_i(k))$$
$$x^s_i(k+1) = \gamma(\tilde{x}^s_i(k)) + (1-\gamma)x^s_i(k-1) \text{ for } s = 1,\ldots,m$$

7: **end while**

Note that Algorithm 2 is equal to (5) for $\gamma = 1$. The main result in this section consists in showing the equivalence in expectation from Algorithm 2 and Algorithm 1. Then, the results provided in subsection III-A can be extrapolated to the asynchronous and randomized setting.

Theorem 2. *Consider a network under Algorithm 2 with* $\mathcal{G}$ *connected,* $\mathbf{r}(k) = \mathbf{r}$ *constant,* $\alpha \in (0, \frac{1}{\gamma})$, *and* $\gamma \in (1,2)$. *If the sequence of selected edges is i.i.d. with uniform probability, then Algorithm 2 preserves the steady-state error properties of the original filter in* (5)*, i.e,:*

- $\mathbf{x}^m(k)$ *converges in distribution to a random variable* $\mathbf{x}^m_\infty$*, and this distribution is unique.*
- $\lim_{k\to\infty} \mathsf{E}[\mathbf{x}^m(k)] = \mathsf{E}[\mathbf{x}^m_\infty] = \mathbf{x}^{m,*}$.

Proof. The first statement of the proof is a direct extrapolation of the proof of Theorem 6 in [9], which is based on the results obtained by Ravazzi et al. ([24]). Algorithm 2 fulfills all the requirements that hold in the proof of Theorem 6 in [9]: discrete time, Schur stability, affine dynamics and the sequence of edges is i.i.d. with uniform probability.

Regarding the second statement, Theorem 4.1 presented in [8] proves that

$$\mathsf{E}[\bar{\mathbf{x}}^1(k+1)] = (\mathbf{I} - \epsilon'\mathbf{L})\mathsf{E}[\mathbf{x}^1(k)] + \alpha'(\mathbf{r} - \mathsf{E}[\mathbf{x}^1(k)]), \quad (23)$$

with $\epsilon' = \frac{1}{2\mathsf{car}(\mathcal{E})}$ and $\alpha' = \frac{\alpha}{\mathsf{car}(\mathcal{E})}$. Accordingly, the protocol in Algorithm 2 is

$$\begin{aligned}\mathsf{E}[\mathbf{x}^1(k+1)] =& \gamma(\mathbf{I} - \epsilon'\mathbf{L})\mathsf{E}[\mathbf{x}^1(k)] + \gamma\alpha'(\mathbf{r} - \mathsf{E}[\mathbf{x}^1(k)]) + \\ & (1-\gamma)\mathsf{E}[\mathbf{x}^1(k-1)]).\end{aligned} \quad (24)$$

The rest of the proof follows from the fact that, if we replace $\mathsf{E}[\mathbf{x}^1(k)] = \mathsf{E}[\mathbf{x}^1(k+1)] = \mathsf{E}[\mathbf{x}^1(k-1)] = \mathsf{E}[\mathbf{x}^{m,*}(k)]$, then the dynamics in Eq. (24) are the same that appear in Eq. (7), so the same procedure can be followed to prove the second statement. □

Therefore, Algorithms 1 and 2 are equivalent in expectation, so the (expected) steady-state error is the same for both algorithms and it is achieved with the same convergence speed. Interestingly, another advantage of our accelerated proposal compared to the original filter is that, now, α can be tuned to have lower values while preserving the convergence speed. This is important because lower values of α yield to lower noise associated to the random selection of nodes, so for the same speed convergence, the noise level can be decreased.

IV. Illustrative Examples

In this section we evaluate Algorithms 1 and 2 with numerical simulations, assessing the theoretical improvements of the accelerated version of the filters with respect to the original versions developed in [8], [9].

We consider the network of $N = 8$ nodes in Fig. 1. The parameters are $\alpha = 0.04$, $\epsilon = 0.01$, and $m = 5$. The signals $r_i(k)$ evolve according to a uniform random process, such that, every 2000 steps, $r_i(k) \sim \mathcal{U}(0,1)\ \forall i \in \mathcal{V}$. Besides, $\mathbf{x}^s(-1) = \mathbf{x}^s(0) = [0.99, 0.27, 0.02, 0.48, 0.18, 0.24, 0.65, 0.50]^T\ \forall s$. The value of γ can be computed by means of distributed algorithms that estimate the algebraic connectivity of the graph (see, e.g., [25], [26]).

The evolution of the estimates for the synchronous and non-randomized algorithms is shown in Fig. 2. The succession of stages reduces significantly the steady-state error for both the original and accelerated algorithms. However, for the same protocol, the convergence time increases with the number of stages, specially in the original multi-stage filter (Fig. 2 (left)). With the additional memory slot, the convergence time has been substantially improved by the accelerated multi-stage filter in Algorithm 1 (Fig. 2 (right)). This is more evident in the final stages (e.g., $s = m = 5$). To better compare the steady-state error and convergence speed, Fig. 3 draws the absolute error between the average estimate across the network and the average of the input signals $e^s(k) = \frac{1}{N}\sum_{i=1}^{N}|x_i^s(k) - r_i(k)|$ and $\mathbf{e}(k) = [e^1(k), \ldots, e^m(k)]^T$. For the same convergence speed, the accelerated filter can be designed with more stages and improve the steady-state error in various orders of magnitude.

In the randomized algorithms, we set $\alpha = 0.0005$ to highlight the improvements achieved by the second order method. Besides, since the topology changes arbitrarily with time, we cannot assume that the second largest eigenvalue can be computed distributively, so we fix $\gamma = 1.7$. Finally, the signals $r_i(k)$ change every 20000 steps.

Fig. 4 represents the result of the experiments. With a low

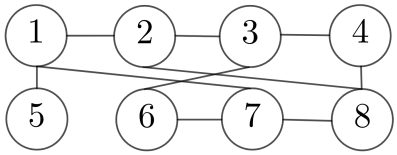

Fig. 1: Graph considered in the simulations.

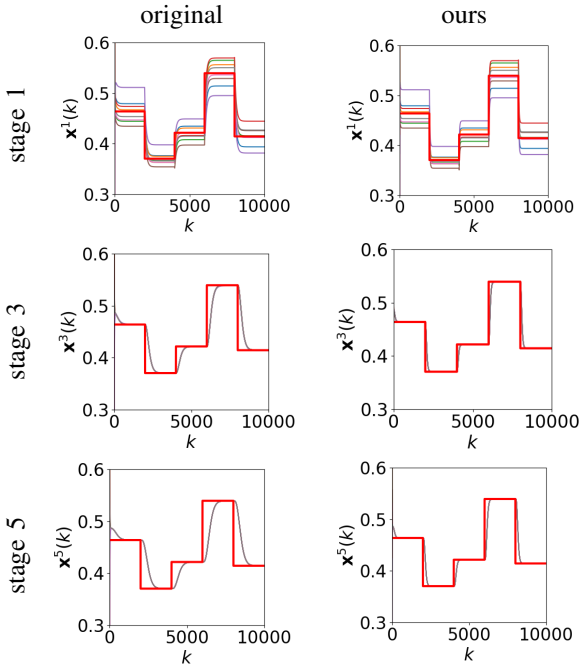

Fig. 2: Simulation results for (left) the original multi-stage and (right) the accelerated multi-stage algorithm (ours).

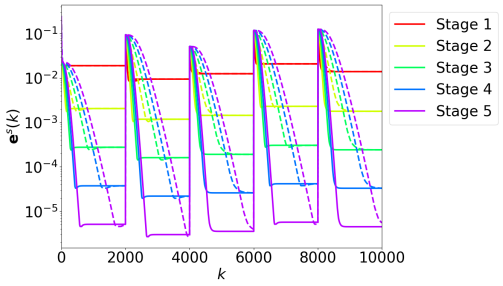

Fig. 3: Evolution of the absolute error between the average estimate across the network and the average of the input signals. The accelerated multi-stage filter is in solid lines, whereas the original multi-stage filter is in dashed lines.

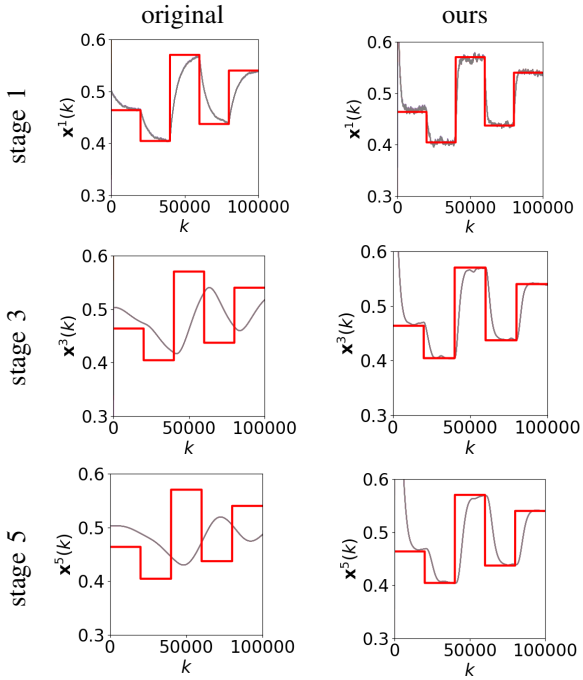

Fig. 4: Simulation results for (left) the randomized original multi-stage and (right) the accelerated randomized multi-stage algorithm (ours).

value of α, the noise due to the randomized links is filtered. The accelerated filter can compensate the degradation in convergence speed, while the original filter is too slow to converge before the signals change.

V. Conclusions

This paper has presented two novel accelerated discrete time dynamic average consensus protocols based on a multi-stage sequence of filters and a second order recurrence. The combination allows to alleviate the trade-off between convergence speed and steady-state error. The multi-stage scheme can reduce arbitrarily the steady-state error by increasing the number of stages, but this implies a reduction of the convergence speed that can impact the performance depending on the application. Thanks to the second order recurrence, the convergence is sped up, counteracting the drawback of the multi-stage, specially at final stages of the protocol. This conclusion, drawn for the synchronous and deterministic consensus protocol, is shared by its asynchronous and randomized version. In the latter, parameter α manifests a trade-off between convergence speed, noise and average steady-state error that the acceleration due to the second order method compensates.

References

[1] F. Garin and L. Schenato, "A survey on distributed estimation and control applications using linear consensus algorithms," in *Networked control systems*. Springer, 2010, pp. 75–107.

[2] M. Franceschelli, A. Pilloni, and A. Gasparri, "Multi-agent coordination of thermostatically controlled loads by smart power sockets for electric demand side management," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 731–743, 2020.

[3] E. Sebastián, E. Montijano, and C. Sagüés, "All-in-one: Certifiable optimal distributed Kalman filter under unknown correlations," in *IEEE Conference on Decision and Control*, 2021, pp. 6578–6583.

[4] 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, 2019.

[5] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[6] E. Montijano, J. I. Montijano, C. Sagüés, and S. Martínez, "Robust discrete time dynamic average consensus," *Automatica*, vol. 50, no. 12, pp. 3131–3138, 2014.

[7] F. Chen, C. Chen, G. Guo, C. Hua, and G. Chen, "Delay and packet-drop tolerant multistage distributed average tracking in mean square," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, p. 9535 – 9545, 2022.

[8] M. Franceschelli and A. Gasparri, "Multi-stage discrete time dynamic average consensus," in *IEEE Conference on Decision and Control*, 2016, pp. 897–903.

[9] ——, "Multi-stage discrete time and randomized dynamic average consensus," *Automatica*, vol. 99, pp. 69–81, 2019.

[10] B. Ghosh, S. Muthukrishnan, and M. H. Schultz, "First and second order diffusive methods for rapid, coarse, distributed load balancing," in *ACM symposium on Parallel algorithms and architectures*, 1996, pp. 72–81.

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

[12] G. Karagiorgos and N. M. Missirlis, "Accelerated diffusion algorithms for dynamic load balancing," *Information Processing Letters*, vol. 84, no. 2, pp. 61–67, 2002.

[13] R. Diekmann, A. Frommer, and B. Monien, "Efficient schemes for nearest neighbor load balancing," *Parallel computing*, vol. 25, no. 7, pp. 789–812, 1999.

[14] E. Kokiopoulou and P. Frossard, "Polynomial filtering for fast convergence in distributed consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 342–354, 2008.

[15] E. Montijano, J. I. Montijano, and C. Sagues, "Chebyshev polynomials in distributed consensus applications," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 693–706, 2012.

[16] B. Van Scoy, R. A. Freeman, and K. M. Lynch, "A fast robust nonlinear dynamic average consensus estimator in discrete time," *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 191–196, 2015.

[17] M. Franceschelli and A. Gasparri, "Gossip-based centroid and common reference frame estimation in multiagent systems," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 524–531, 2013.

[18] J. Lai, X. Lu, F. Wang, P. Dehghanian, and R. Tang, "Broadcast gossip algorithms for distributed peer-to-peer control in ac microgrids," *IEEE Transactions on Industry Applications*, vol. 55, no. 3, pp. 2241–2251, 2019.

[19] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[20] J. Liu, S. Mou, A. S. Morse, B. D. Anderson, and C. Yu, "Deterministic gossiping," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1505–1524, 2011.

[21] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009, vol. 27.

[22] J. Liu, B. D. Anderson, M. Cao, and A. S. Morse, "Analysis of accelerated gossip algorithms," *Automatica*, vol. 49, no. 4, pp. 873–883, 2013.

[23] F. Bullo, *Lectures on Network Systems*. Kindle Direct Publishing, 2020.

[24] C. Ravazzi, P. Frasca, R. Tempo, and H. Ishii, "Ergodic randomized algorithms and dynamics over networks," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 1, pp. 78–87, 2014.

[25] M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu, "Decentralized estimation of Laplacian eigenvalues in multi-agent systems," *Automatica*, vol. 49, no. 4, pp. 1031–1036, 2013.

[26] E. Montijano, J. I. Montijano, and C. Sagues, "Fast distributed algebraic connectivity estimation in large scale networks," *Journal of the Franklin Institute*, vol. 354, no. 13, pp. 5421–5442, 2017.