Multi-Stage Discrete Time and Randomized Dynamic Average Consensus

Mauro Franceschelli [a] Andrea Gasparri [b]

[a] *Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy.*

[b] *Department of Engineering, Roma Tre University, Rome, Italy.*

Abstract

In this paper we propose a novel local interaction protocol which solves the discrete time dynamic average consensus problem, i.e., the consensus problem on the average value of a set of time-varying input signals in an undirected graph. The proposed interaction protocol is based on a multi-stage cascade of dynamic consensus filters which tracks the average value of the inputs with small and bounded error. We characterize its convergence properties for time-varying discrete-time inputs with bounded variations. The main novelty of the proposed algorithm is that, with respect to other dynamic average consensus protocols, we obtain the next unique set of advantages: i) The protocol, inspired by proportional dynamic consensus, does not exploit integral control actions or input derivatives, thus exhibits robustness to re-initialization errors, changes in the network size and noise in the input signals; ii) The proposed design allows to trade-off the quantity of information locally exchanged by the agents, i.e., the number of stages, with steady-state error, tracking error and convergence time; iii) The protocol can be implemented with randomized and asynchronous local state updates and keep in expectation the performance of the discrete-time version. Numerical examples are given to corroborate the theoretical findings, including the case where a new agent joins and leaves the network during the algorithm execution to show robustness to re-initialization errors during runtime.

1 Introduction

The consensus problem, popularized by Olfati-Saber et al. (2007) and many others, consists in the design of local interaction rules for networks of dynamical systems, i.e., agents, with the objective to drive their state variables towards a common value by exploiting only locally available information. The literature on the consensus problem has grown significantly in the past decade and different scenarios and assumptions on the network topology, the agents' dynamical models and the nature of the interactions have been explored. Regarding the network topology, from the initial assumption of static undirected graphs the problem formulation has been successively extended to the case of dynamic (switching) directed graphs, see the early works by Franceschelli et al. (2009), Cai and Ishii (2012), Domínguez-García and Hadjicostis (2011), and Montijano et al. (2015). Regarding the agents' dynamical model, from the initial agents' modeling as first order integrators, the problem formulation has been generalized to the case of second order integrators, higher order dynamics and nonlinear dynamics Yu et al. (2010); Bauso et al. (2006) and many others.

The consensus problem has proven fundamental for the decentralization of a multitude of algorithms for networked multi-agent systems. Typical applications concern the design of distributed data-fusion and estimation algorithms, as in the works by Cattivelli and Sayed (2010) and Franceschelli and Gasparri (2014) or the development of distributed clock-synchronization protocols, see for instance the works by Carli et al. (2011) and Garone et al. (2015).

In this paper we focus our attention on the problem of dynamic average consensus, which is a consensus problem where the state variables of the agents are driven towards the average of a set of external time-varying signals given as input to the agents. Early pioneers of dynamic average consensus are, to the best of our knowledge, Spanos et al. (2005). Their main idea was to consider the derivative of these inputs so that the average value of the state variables in the network would track the average value of the set of input signals. In their approach, noise added to the input derivatives would disrupt the algorithm convergence properties. Several works followed, which explored dynamic average consensus in different settings.

⋆ This work was supported by the Italian Ministry of Research and Education (MIUR) with the grant "CoNetDomeSys", code RBSI14OF6H, under call SIR 2014.

⋆⋆https://doi.org/10.1016/j.automatica.2018.10.009

Email addresses: mauro.franceschelli@diee.unica.it (Mauro Franceschelli), gasparri@dia.uniroma3.it (Andrea Gasparri).

Similar issues can be found also in nonlinear versions of the dynamic average consensus protocols such as in Nosrati et al. (2012). Input derivatives are exploited also in event-triggered approaches such as in Kia et al. (2015a) which extends the dynamic average consensus problem to networks with intermittent communications. Notably, Zhu and Martínez (2010), proposed a discrete time version of dynamic average consensus exploiting finite differences of the inputs instead of their derivatives. The discrete time implementation is achieved at the expense of a finite error in the consensus value which has been characterized as function of the network parameters.

Another work particularly significant in the context of dynamic average consensus is the one by Montijano et al. (2014). The authors propose a discrete time dynamic average consensus algorithm which is able to achieve both small steady state error and be robust with respect to re-initialization errors by the addition of a "damping" factor to mitigate past errors. In this paper, with respect to Montijano et al. (2014), we do not exploit the finite differences of order k of the inputs to achieve dynamic average consensus. The proposed protocol exploits solely knowledge of the time-varying magnitude of the inputs and aims to the objective of achieving robustness to past errors due to re-initialization, noise or else by not keeping any memory of past states and exploiting only their instantaneous value. This is in contrast with the work by Montijano et al. (2014) which dampens past errors by exploiting a so-called forgetting-factor in the agents' dynamics.

Freeman et al. (2006), one of the earliest works in the field, proposed proportional (P) and proportional-integral (PI) dynamic consensus algorithms in continuous time, which do not exploit input derivatives. Their proposed P dynamic consensus algorithm is able to track with finite error the average of the input signals while the PI is able to achieve zero error for constant inputs. The authors propose an error bound for their P dynamic consensus algorithm which is proportional to the norm of a weighted sum of the inputs and their derivatives, thus being able to achieve tunable errors with resepct to the magnitude and rate of change of the inputs. The approach is able to achieve a trade-off between small steady state error at the price of a large convergence rate and vice-versa. In this paper we present the multi-stage dynamic consensus algorithm which is composed by a cascade of m stages of discrete-time proportional dynamic consensus filters and show that its steady-state error error for constant inputs decreases geometrically with the number of stages while the tracking error with respect to changes in the time-varying average of the inputs increases only linearly with m, thus enabling an appropriate tuning depending on the characteristics of the inputs. Furthermore, our discrete-time approach is designed to enable a randomized and asynchronous version of the proposed algorithm which is provided. In Freeman et al. (2006) it is also proposed a PI dynamic consensus algorithm to improve the performance regarding the steady state error for constant inputs by assuming that the sum of the co-state variables representing the integral part of algorithm is time-invariant. This method allows to achieve zero steady state error, its main disadvantage is a vulnerability to re-initialization errors which may cause a persistent bias in the estimation in the presence of networks with time-varying size, noise or faulty agents. In this paper we propose a method which avoids the vulnerabilities of dynamic consensus algorithms based on integral control actions and that can be implemented with randomized interactions.

In Bai et al. (2010), the authors propose an alternative method for PI dynamic average consensus which is robust to re-initialization errors for given classes of time-varying inputs such as ramps or sinusoids of known frequency for which an estimator based on the internal model principle can be designed. In this paper we provide and characterize results for general time-varying inputs.

In Scoy et al. (2015) the proportional dynamic average consensus algorithm has been improved significantly and elegantly with respect to the work by Freeman et al. (2006) and Bai et al. (2010). By exploiting a nonlinear scheme for the local state update which involves a mapping from the reals to the torus the authors achieve robustness to initialization errors and exponential convergence by assuming that the inputs are bounded and this bound known, hence the main disadvantage of the algorithm consists in not being able to address ramp inputs or inputs with unknown upper bound due to the need to exploit this bound for the correct tuning of the algorithm. In this paper we propose a different approach based on a cascade of modified P dynamic average consensus algorithms which aims to improve the error performance while retaining robustness to re-initialization errors, we also design our algorithms to enable a randomized and asynchronous implementation of the local state update rule.

Another recent work dealing with the dynamic average consensus problem is by Kia et al. (2015b). The authors consider networks modeled by weight-balanced strongly connected graphs and propose a PI algorithm which tracks the average of the inputs with average errors and is able to achieve zero error for special classes of inputs. The authors propose a continuous time algorithm and discuss how to choose a proper step-size for a discrete time implementation. Initialization and re-initialization errors are present only if the main assumption made by authors is violated, that is if the graph structure is not balanced at some instant of time thus introducing a bias in the estimation even if the graph structure is corrected after the fault, this scenario may occur if faulty agents or faulty communication links are considered. The main contribution of their work consists in dealing with time-varying topologies, control with limited authority (saturation of control variables), tunable rate of convergence and a characterization of the privacy preservation properties of their algorithms.

A particularly interesting version of consensus algorithms is represented by gossip based algorithms. One of the early proposers of these algorithms for the problem of distributed averaging were Boyd et al. (2006). Gossip algorithms consist in local interaction rules executed at random instants of time with random neighbors in the communication network. This scheme of interaction is particularly convenient when implementing these algorithms in large scale networks with possibly fast changing topologies because there is no need to synchronize the state updates and the inherent parallelism of the local interactions can be fully exploited. These algorithms have been exploited in several applications which make use of the average of a set of measurements or data, see the work by Dimakis et al. (2010) and references therein for an introduction to the topic.

Other approaches, such as the work by Habibi et al. (2015), address the dynamic average consensus problem in a randomized and asynchronous framework and are based on the idea of implementing a pipeline in which new instances of the gossip algorithm in Boyd et al. (2006), with new and updated initial conditions, are initialized at each discrete instant of time while the older instances output the current estimated average of the set of network variables.

A different framework of gossiping has been proposed by Ravazzi et al. (2015), the authors propose a method to randomize affine dynamics (linear dynamics with inputs) and characterize the convergence properties based on the ergodicity property of the considered stochastic process. This allowed the authors to show that despite the state variables do not converge to a limit, they converge in distribution to a random variable with known expected value. This expected value can be computed by the agents by taking the time average of their own state trajectory.

The **main contribution** of this paper is the design of a local interaction protocol among agents of a network to solve the dynamic average consensus problem, i.e., achieve consensus on the time-varying average of a set of discrete time signals $u_i(k)$ with average $\bar{u}(k) = \frac{\sum_{i=1}^{n} u_i(k)}{n}$ which has a unique set of features and advantages with respect to the state of the art.

Our basic idea is to consider a cascade of m modified proportional dynamic average consensus filters to achieve a small steady-state error by the design of the size of the cascade and other tuning parameters while retaining the robustness of approaches based on proportional dynamic consensus with respect to re-initialization errors due to faults or changes in the network. We characterize an explicit upper bound to the maximum tracking error for time-varying inputs and propose a randomized and asynchronous implementation of the algorithm. One of the main results is that by increasing linearly the number of state variables exchanged locally among agents we obtain a geometric reduction in steady state error .

The method proposed in this paper does not make use of input derivatives that may be disruptive in the case the input signals are affected by noise. This feature, together with the absence of integral actions in the form of auxiliary variables, avoids the need of re-initialization in the case new agents join or leave the network or faulty agents influence the network disrupting the average value of the network state. In addition, our proposed design allows to trade-off convergence rate with steady-state error by choosing a proper number of stages in the cascade. Finally, we propose an asynchronous and randomized (gossip-like) version of the protocol and characterize in expectation its convergence properties.

The rest of the paper is organized as follows. In Section 2, some fundamental results on graph theoretic modeling of multi-agent systems are reviewed. In Section 3, we propose the multi-stage discrete time dynamic average consensus protocol and characterize some of its convergence properties. In Section 4 a version of the proposed protocol which can be implemented according to a randomized and asynchronous pair-wise communication scheme is detailed. In Section 5 numerical simulations are given to corroborate the theoretical findings, while in Section 6 conclusive remarks are drawn and future work is discussed.

2 Preliminaries

In this section we review some preliminary results and notation of graph theoretic modeling of multi-agent systems.

Consider a system of n agents whose network topology can be described by an undirected graph $\mathcal{G} = (V, E)$, where $V = \{1, \ldots, n\}$ is the set of agents, and $E \subseteq V \times V$ is the set of edges: an edge $e_{i,j}$ exists between agents i and j if agent i interacts with agent j. Note that, since the graph is undirected, the existence of an edge $e_{i,j} \in E$ implies that $e_{j,i} \in E$ as well. Let A be the $n \times n$ adjacency matrix of the graph $\mathcal{G}$ whose elements are $a_{i,j} = 1$ if the edge $e_{i,j}$ exists, i.e., $e_{i,j} \in E$, $a_{i,j} = 0$ otherwise. Let $\mathcal{N}_i$ define the neighborhood of agent i, that is the set of agents j for which $e_{i,j} \in E$. Let $D_i = |\mathcal{N}_i|$, denote the degree of the agent i, that is the number of incident edges to agent i, and let us denote with $D = \operatorname{diag}(D_1, \ldots, D_n)$, a diagonal matrix for which the i-th element on the main diagonal is the degree of the i-th agent. Let $L = D - A$ be the Laplacian matrix which encodes the graph topology. Note that, since graph $\mathcal{G}$ is undirected, the Laplacian matrix is symmetric by construction and thus all its eigenvalues are real. Furthermore, for a connected graph, it has one null eigenvalue with corresponding unique eigenvector equal to the vector of ones, i.e., $\mathbf{1} \in \mathbb{R}^n$. In addition according to the Geršchgorin disc Theorem, a Laplacian matrix has all its eigenvalues located within $[0, 2\,D_{\max}]$ where $D_{\max}$ is the maximum degree among the agents in the graph. In the sequel, we will denote the eigenvalues of the Laplacian matrix as $\lambda_{L,i} \in \sigma(L)$, sorted by their magnitude that is $0 = \lambda_{L,1} < \lambda_{L,2} \leq \ldots \leq \lambda_{L,n} \leq 2\,D_{\max}$. We assume that an upper bound to the maximum degree $D_{\max}$ is available to the agents in the network.

Algorithm 1 Multi-Stage Dynamic Consensus Protocol

State of Agent i: $x_i^s(k)$, for $s = 1, \ldots, m$, $x_i^s(0) = x_{i0}^s$;
Input signal: $u_i(k)$, for $i = 1, \ldots, n$;
Tuning parameters: $\alpha, \varepsilon \in \mathbb{R}^+$, $\varepsilon < \frac{1}{2D_{max}}$, $\alpha < 1 - \varepsilon D_{max}$;
Protocol execution: All agents repeat indefinitely the next two operations, here reported for agent i:
- Gather $x_j^s(k)$ for $s = 1, \ldots, m$ and $j \in \mathcal{N}_i$.
- Update state variables $\{x_i^s(k)\}_{s=1,\ldots,m}$ as in eq. (1).

Finally, let us denote with $\bar{u}$ the average value of a vector $\mathbf{u} \in \mathbb{R}^n$, that is $\bar{u} = \frac{1}{n}\sum_{i\in V} u_i$.

3 Discrete Time Protocol

In this section we present a novel discrete time multi-stage consensus protocol and characterize some of its convergence properties. The proposed protocol is based on a multi-stage cascade of damped consensus filters and involves the local update of m variables based on the states of each agent's neighbors. More specifically, each agent i has a state $x_i = [x_i^1, \ldots, x_i^m]^T$ which evolves according to the following update rule

$$\begin{aligned}
x_i^1(k+1) &= x_i^1(k) - \sum_{j\in\mathcal{N}_i} \varepsilon\left(x_i^1(k) - x_j^1(k)\right) \\
&\qquad + \alpha\left(u_i(k) - x_i^1(k)\right), \\
x_i^2(k+1) &= x_i^2(k) - \sum_{j\in\mathcal{N}_i} \varepsilon\left(x_i^2(k) - x_j^2(k)\right) \\
&\qquad + \alpha\left(x_i^1(k) - x_i^2(k)\right), \\
&\vdots \\
x_i^m(k+1) &= x_i^m(k) - \sum_{j\in\mathcal{N}_i} \varepsilon\left(x_i^m(k) - x_j^m(k)\right) \\
&\qquad + \alpha\left(x_i^{m-1}(k) - x_i^m(k)\right),
\end{aligned} \tag{1}$$

where α, $\varepsilon \in \mathbb{R}$ are tuning parameters.

As detailed in Algorithm 1, to execute the protocol, the generic agent i must send a message M_i composed by a set of m local variables $M_i = \left\{x_i^1, x_i^2, \ldots, x_i^m\right\}$. Therefore, each agent needs only to have sufficient memory storage capability to handle the reception of $|\mathcal{N}_i|$ messages $\{M_j\}_{j\in\mathcal{N}_i}$ from each of its neighbors, i.e., a maximum of D_{max} messages.

The evolution of the agents' states in a network running protocol (1) can be compactly represented in vector format as follows

$$\begin{aligned}
x^1(k+1) &= \left(I - \varepsilon L\right)x^1(k) + \alpha\left(u(k) - x^1(k)\right), \\
x^2(k+1) &= \left(I - \varepsilon L\right)x^2(k) + \alpha\left(x^1(k) - x^2(k)\right), \\
&\vdots \\
x^m(k+1) &= \left(I - \varepsilon L\right)x^m(k) + \alpha\left(x^{m-1}(k) - x^m(k)\right).
\end{aligned} \tag{2}$$

3.1 *Multi-Stage Steady State and Estimation Accuracy*

We are now ready to characterize the steady state equilibrium for the proposed Multi-Stage Dynamic Consensus Protocol (1). In this regard, let us denote with $x^\star(k)$ the steady-state that the system would reach if the input signal were held constant at $\mathbf{u}(k)$ for some sufficiently long time, that is

$$x^\star(k) = \left[x^{m,\star}(k)^T, \ldots, x^{1,\star}(k)^T\right]^T, \quad x^\star(k) \in \mathbb{R}^{nm\times 1},$$

were $x^{s,\star}(k) \in \mathbb{R}^n$, with $s = 1, \ldots, m$ represents the equivalent steady-state of each single stage s of the cascaded consensus filter. Next, in Theorem 1 we characterize the value of $x^{s,\star}(k)$ as function of $\mathbf{u}(k)$.

Theorem 1 *Consider a multi-agent system running Algorithm 1. If the graph $\mathcal{G}$ is connected, $\alpha \in (0, 1 - \varepsilon D_{max})$, $\varepsilon \in \left(0, \frac{1}{2D_{max}}\right)$, then it holds*

$$x^{s,\star}(k) = \left(\alpha I + \varepsilon L\right)^{-s}\alpha^s \mathbf{u}(k). \tag{3}$$

If $\mathbf{u}(k)$ is constant, then (3) *is the steady-state value of the s-th stage.*

Proof:
See the proof in Appendix 7.1. □

Note that, for the sake of brevity in the sequel we will assume that the conditions dictated by Theorem 1 are always satisfied, that is the graph $\mathcal{G}$ is connected and $\alpha \in (0, 1 - \varepsilon D_{max})$ and $\varepsilon \in \left(0, \frac{1}{2D_{max}}\right)$.

Now we characterize the relationship between $x^{s,\star}(k)$ and the time varying average of the input signals, $\bar{u}(k)$. This relationship gains the meaning of *steady-state estimation accuracy* for the s-th stage of the protocol if the inputs at some point become steady, i.e., constant, thus allowing the system to reach a steady state.

Theorem 2 *Consider a multi-agent system running Algorithm 1 on a connected graph $\mathcal{G}$ with $\alpha \in (0, 1 - \varepsilon D_{max})$, $\varepsilon \in \left(0, \frac{1}{2D_{max}}\right)$, $\beta = \alpha + \varepsilon\lambda_{2,L}$. It holds*

$$\left\|\bar{u}(k)\,\mathbf{1} - x^{m,\star}(k)\right\|_2 \leq \left(\frac{\alpha}{\beta}\right)^m \left\|\hat{\mathbf{u}}(k)\right\|_2, \tag{4}$$

with $\hat{\mathbf{u}}(k) = (\mathbf{u}(k) - \bar{u}(k)\,\mathbf{1})$. If $\mathbf{u}(k)$ is constant, then (4) *is the steady-state estimation error of the m-th stage.*

Proof:

See the proof in Appendix 7.2 □

Notably, what (4) tells us is that having a cascade of m systems with a proper choice of the parameters α and m allows to reduce the distance from the steady-state equilibrium of the m-th stage in the cascade and the average of the input signals.

3.2 Protocol Convergence Properties

We are now interested in characterizing the convergence properties for the proposed algorithm for a generic input $u(k)$. The dynamics of the cascade of consensus filters of protocol (1) can be represented in a compact form as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ x^m(k) &= Cx(k), \end{aligned}$$

where $x(k) = \left[x^m(k)^T, \ldots, x^1(k)^T\right]^T \in \mathbb{R}^{nm \times 1}$, matrix $A \in \mathbb{R}^{nm \times nm}$ is defined as

$$A = \begin{bmatrix} I-Q & \alpha I & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & I-Q & \alpha I & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & I-Q & \alpha I \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & I-Q \end{bmatrix}, \tag{5}$$

matrix $Q \in \mathbb{R}^{n \times n}$ is defined as

$$Q = \alpha I + \varepsilon L, \tag{6}$$

matrix $B \in \mathbb{R}^{nm \times n}$ is defined as

$$B = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \alpha I \end{bmatrix}, \tag{7}$$

and matrix $C \in \mathbb{R}^{n \times nm}$ defined as

$$C = [\; I \;\; \mathbf{0} \;\; \ldots \;\; \mathbf{0} \;].$$

Let us now define the *estimation error* $e(k)$ of the proposed multi-stage cascade of consensus filters at time k as

$$e(k) = x(k) - x^\star(k). \tag{8}$$

with $e(k) = \left[e^m(k)^T, \ldots, e^1(k)^T\right]^T \in \mathbb{R}^{nm \times 1}$. The next proposition characterizes the error dynamics of the proposed cascade of consensus filters.

Proposition 1 *The dynamics of the error defined in* (8) *can be expressed as*

$$e(k+1) = Ae(k) + \delta^\star(k), \tag{9}$$

with $\delta^\star(k) = x^\star(k) - x^\star(k+1)$ *and the matrix* A *defined as in eq.* (5).

Proof:

See the proof in Appendix 7.3 □

We now characterize the convergence rate for the error dynamics given in (9). In particular, motivated by the results of Theorem 2, with no lack of generality we focus only on the error $e^m(k)$, that is the convergence rate related to the m-th stage of the proposed multi-stage cascade of consensus filters. To this end, let us recall that $e^m(k) = C\, e(k)$, and let us define the *average error value* $\bar{e}^m(k)$ and the *disagreement error vector* $\hat{e}^m(k)$ for the m-th stage at time k as

$$\bar{e}^m(k) = \frac{\mathbf{1}^T e^m(k)}{n}, \tag{10}$$

and

$$\hat{e}^m(k) = e^m(k) - \bar{e}^m(k)\mathbf{1} = P\, e^m(k), \tag{11}$$

where $P = (I - \frac{\mathbf{1}\mathbf{1}^T}{n})$ is an orthogonal projection matrix. In addition, let us write the estimation error $e^m(k)$ of the m-th stage as the combination of the natural response $e^m_n(k)$ and the forced response $e^m_f(k)$ of the dynamical system given in (9), that is

$$e^m(k) = e^m_n(k) + e^m_f(k).$$

We emphasise that in this context we refer to the natural response of the error dynamics in (9) and not that of protocol in (2). In particular, the natural response of the error dynamics corresponds to the state trajectory of the solution of eq. (9) due to the initial condition $e(0)$ with $\delta^\star(k) = 0$ for all $k \geq 0$, therefore the case in which the input vector $u(k)$ in (2) is constant at all times but possibly different from zero. Similarly, we consider the forced response of the error dynamics in (9) instead of forced response of the protocol in (2). Now, let us define $\delta_u(k)$ as

$$\delta_u(k) = \mathbf{u}(k) - \mathbf{u}(k+1) = \bar{\delta}_u(k)\mathbf{1} + \hat{\delta}_u(k), \tag{12}$$

where $\bar{\delta}_u(k)$ is the average value of the vector $\delta_u(k)$, $\hat{\delta}_u(k)$ is such that $\delta_u(k) = \bar{\delta}_u(k)\mathbf{1} + \hat{\delta}_u(k)$ with $\hat{\delta}_u(k)^T\mathbf{1} = \mathbf{0}$.

The following theorem provides an upper bound on the norm of the natural response for the error dynamics given in (9) which characterizes the convergence rate of the algorithm.

Theorem 3 *Consider a multi-agent system running Algorithm 1. Then, for any initial condition* $e(0)$*, the natural response of the error dynamics in* (9) *for the* m*-th stage satisfies*

$$\|e^m_n(k)\|_2 \leq (1-\alpha)^k \, \frac{k^{m-1}}{(m-1)!} \, \frac{1 - \left(\frac{\alpha}{1-\alpha}\right)^m}{1 - \frac{\alpha}{1-\alpha}} \, e_{max}(0),$$

where $e_{max}(0) = \max_{s \in \{1,\ldots,m\}} \{\|e^s(0)\|_2\}$.

Proof:

See the proof in Appendix 7.4. □

Now, a few observations are in order:

- The convergence rate is geometric and determined by the value of α. In addition, it is possible to reduce the convergence rate to decrease the steady-state error.
- The cascade of m stages tuned with $\alpha = \alpha'$ is inherently faster than a single stage implementation tuned with $\alpha = (\alpha')^m$.

The following theorem provides a bound on the maximum of the average error value and of the norm of disagreement error vector of the forced response for the error dynamics given in (9).

Theorem 4 *Consider a multi-agent system running Algorithm 1. For all $k \geq 0$ the forced response of the error dynamics of the m-th stage satisfies*

$$|\bar{e}_f^m(k)| \leq \frac{m}{\alpha} \bar{\delta}_{u,max}, \tag{13}$$

and

$$\|\hat{e}_f^m(k)\| \leq \frac{m}{\beta} \left(\frac{\alpha}{\beta}\right)^m \hat{\delta}_{u,max}, \tag{14}$$

where $\bar{e}_n^m(k)$ and $\hat{e}_n^m$ are defined respectively in (10) *and* (11)*, $\beta = \alpha + \varepsilon\lambda_{L,2}$, $\bar{\delta}_{u,max} = \max_{k\in\{0,\infty\}} |\bar{\delta}_u(k)|$ and $\hat{\delta}_{u,max} = \max_{k\in\{0,\infty\}} \|\hat{\delta}_u(k)\|_2$.*

Proof:
See the proof in Appendix 7.5. □

The next corollary summarizes the results of Theorem 2, Theorem 3,and Theorem 4 to compute the maximum tracking error of Algorithm 1 for time-varying inputs.

Corollary 1 *For all $k \geq 0$ the maximum tracking error for the m-th stage of a multi-agent system running Algorithm 1 is upper bounded by*

$$\begin{aligned} \max_{i\in V} |x_i^m(k) - \bar{u}(k)| \leq & \frac{m}{\alpha}\bar{\delta}_{u,max} \\ & + \frac{m}{\beta}\left(\frac{\alpha}{\beta}\right)^m \hat{\delta}_{u,max} \\ & + \left(\frac{\alpha}{\beta}\right)^m \|\hat{\mathbf{u}}(k)\|_2 \\ & + \|e_n^m(k)\|_2 , \end{aligned} \tag{15}$$

where $e_n^m(k)$ is the vanishing natural response of the error dynamics as in Theorem 3, $\beta = \alpha + \varepsilon\lambda_{L,2}$ and $\bar{\delta}_{u,max}$, $\hat{\delta}_{u,max}$, $\hat{\mathbf{u}}(k)$ are as defined in Theorem 2 and Theorem 4. □

Now, a few observations are in order:

- The bound on the maximum tracking error is the sum of four components: i) $\frac{m}{\alpha}\bar{\delta}_{u,max}$, which grows linearly with m and is proportional to the worst case change in the average value of the inputs; ii) $\frac{m}{\beta}\left(\frac{\alpha}{\beta}\right)^m \hat{\delta}_{u,max}$, which decreases geometrically with m and is proportional to the worst case variation of the disagreement vector; iii) $\left(\frac{\alpha}{\beta}\right)^m \|\hat{\mathbf{u}}(k)\|_2$, which decreases geometrically with m and is proportional to the disagreement vector at time k; and iv) $\|e_n^m(k)\|_2$, which decreases geometrically with k and is the error due to the natural response of the the filter, which vanishes to zero as $k \to \infty$ as characterized in Theorem 3.
- The proposed algorithm has its best performance in the case in which the average of value of a large set of inputs changes slowly with respect to the changes of the single inputs, i.e., $\bar{\delta}_{u,max} << \hat{\delta}_{u,max}$ and the optimal choice of m is $m > 1$.
- For a fixed α, the parameter m can be increased to reduce the steady-state error. However, a large m introduces what can be described as a lagging effect or time-delay in the error dynamics.

To better understand the role played the parameters α and m in terms of convergence speed and steady-state error, it should be noticed that the dynamical matrix A is sub-stochastic with spectral radius strictly less then 1. Therefore the norm of the error decreases monotonically at each iteration for constant inputs. In particular, the largest eigenvalue of the matrix A has algebraic multiplicity equal to m. Thus, in the worst case the natural response vanishes to zero as $O\left((1-\alpha)^k \frac{k^{m-1}}{(m-1)!}\right)$, as shown in Theorem 3. This upper bound has a maximum at $k \approx \frac{m-1}{log(\frac{1}{1-\alpha})}$ which grows linearly with m. Therefore, as m is increased we can guarantee that the geometric rate of convergence determined by $(1-\alpha)^k$ starts to dominate the other term only for $k > \frac{m-1}{log(\frac{1}{1-\alpha})}$. The optimal choice of m to minimize the tracking error for a given specified convergence rate and steady-state error depends on the characteristics of the inputs, i.e., on $\bar{\delta}_{u,max}$ and $\hat{\delta}_{u,max}$. Figure 1 depicts how this small time-delay increases linearly with m in the case of a step-change for a constant input value.

4 Asynchronous and Randomized Protocol

In this section we introduce an asynchronous and randomized version (gossip-like, see Boyd et al. (2006)), of the multi-stage protocol given in (1) and detailed in Algorithm 1. Regarding the asynchronous setting, each agent i has a state $x_i = [x_i^1, \ldots, x_i^m]^T$ which evolves, upon the (random) selection of a neighboring agent j, according to the following protocol

$$\begin{aligned} x_i^1(k+1) &= \frac{x_i^1(k) + x_j^1(k)}{2} + \frac{\hat{\alpha}}{D_i}\left(u_i(k) - x_i^1(k)\right), \\ x_i^2(k+1) &= \frac{x_i^2(k) + x_j^2(k)}{2} + \frac{\hat{\alpha}}{D_i}\left(x_i^1(k) - x_i^2(k)\right), \\ &\vdots \\ x_i^m(k+1) &= \frac{x_i^m(k) + x_j^m(k)}{2} + \frac{\hat{\alpha}}{D_i}\left(x_i^{m-1}(k) - x_i^m(k)\right), \end{aligned} \tag{16}$$

where $\hat{\alpha} \in \mathbb{R}$ is a tuning parameter.

In particular, we show that the agents' states in a network running the protocol given in (16) as detailed in

Algorithm 2 evolve in expectation with dynamics formally equivalent to the multi-stage discrete time version of the protocol.

Algorithm 2 Multi-Stage Asynchronous and Randomized Dynamic Consensus Protocol

State of Agent i: $x_i^s(k)$, for $s = 1, \dots, m$, $x_i^s(0) = x_{i0}^s$;
Input signal: $u_i(k)$, for $i = 1, \dots, n$;
Tuning parameters: $\hat{\alpha} \in \mathbb{R}^+$, $\hat{\alpha} < 0.5$;
Protocol execution: All agents repeat indefinitely the next operations, here reported for agent i:
- Select uniformly at random one neighbor $j \in \mathcal{N}_i$
- Gather $\{x_j^s(k)\}$ for $s = 1, \dots, m$.
- Update state variables $\{x_i^s(k)\}_{s=1,\dots,m}$ as in eq. (16).

We point out that Algorithm 2 exploits a **directed information flow**, i.e., the generic agent i gathers the state information from its neighbors and updates its own state without sending the information about its own state to the neighbors at each iteration. Therefore, at each instant of time, only a directed information flow occurs thus simplifying the implementation of the proposed protocol with UDP internet communication protocols among devices. On the other hand, the underlying graph which encodes the available communication channels is undirected, i.e., information may flow in either direction at different instants of time, this is captured by the symmetry of the corresponding Laplacian matrix.

The evolution of the agents' states in a network running protocol (16) given in Algorithm 2 can be compactly represented in vector format, for a given selection of an edge (i, j), as follows

$$\begin{aligned} x^1(k+1) &= W_{ij} x^1(k) + \frac{\hat{\alpha}}{D_i} e_i e_i^T u(k), \\ x^2(k+1) &= W_{ij} x^2(k) + \frac{\hat{\alpha}}{D_i} e_i e_i^T x^1(k), \\ &\vdots \\ x^m(k+1) &= W_{ij} x^m(k) + \frac{\hat{\alpha}}{D_i} e_i e_i^T x^{m-1}(k), \end{aligned} \tag{17}$$

where the matrix W_{ij} is defined as

$$W_{ij} = I + \frac{e_i e_j^T}{2} - \frac{(1 + 2\hat{\alpha}/D_i) e_i e_i^T}{2}, \tag{18}$$

with e_i a vector whose generic element j is zero if $j \neq i$ and one if if $j = i$.

The following theorem characterizes the convergence properties of the proposed m-th order asynchronous and randomized dynamic consensus protocol described in (16) (and (17)). Briefly, this result follows from the application of (Ravazzi et al., 2015, Theorem 1), which in turn requires to establish an equivalence (in expectation) with the m-th order discrete-time dynamic consensus protocol described in Algorithm 1.

Theorem 5 *Consider a multi-agent system that executes Algorithm 2 with time varying inputs $u(k)$. Let $\hat{\alpha} \in (0, 0.5)$. If $\mathcal{G}$ is connected and the sequence of selected edges is i.i.d. with uniform probability distribution, then the expected value of the state variables of protocol* (17) *evolves as follows*

$$\begin{aligned} \mathbb{E}\left[x(k+1)\right] &= \mathbb{E}\left[A(k)\right] \mathbb{E}\left[x(k)\right] + \mathbb{E}\left[B(k)\right] u(k) \\ &= A \mathbb{E}\left[x(k)\right] + B u(k) m \end{aligned} \tag{19}$$

where matrix A and B defined as in eq. (5) *and* (7) *with parameters $\varepsilon = \dfrac{1}{2|E|}$ and $\alpha = \dfrac{\hat{\alpha}}{|E|}$.*

Proof:
See the proof in Appendix 7.6 □

Next, we provide a further result whose proof is based on the work by Ravazzi et al. (2015).

Theorem 6 *Consider a multi-agent system that executes Algorithm 2 with constant inputs* $\mathbf{u}$. *Let $\bar{u} = \frac{1}{n} \sum_{i \in V} u_i$ denote the average value of vector* $\mathbf{u}$. *Let $\hat{\alpha} \in (0, 0.5)$. If $\mathcal{G}$ is connected and the sequence of selected edges is i.i.d. with uniform probability distribution, then*
- *$x^m(k)$ converges in distribution to a random variable x_∞^m and this distribution is unique;*
- *it holds*

$$\lim_{k \to \infty} E\left[x^m(k)\right] = E\left[x_\infty^m\right] = x^{m,\star}.$$

Proof:
See the proof in Appendix 7.7. □

5 Numerical Results

In this section we propose some numerical examples to corroborate the theoretical results and show the performance of the proposed methods.

We considered a network of $N = 10$ agents interacting over a network topology described by an undirected graph $\mathcal{G} = \{V, E\}$ with $V = \{1, \dots, N\}$ and $|E| = 38$, obtained as an Erdős-Rényi random graph with parameter $p = 0.46$

Numerical simulations of Algorithm 1 and Algorithm 2 have been performed with $m = 10$ stages and with a choice of parameters ε and α such that Algorithm 1 simulates the expected value of the state of Algorithm 2, according to the results of Theorem 5. In particular, by choosing $\hat{\alpha} = 0.1$ for Algorithm 2, it holds $\varepsilon = \dfrac{1}{2|E|} = 0.0263$ and $\alpha = \dfrac{\hat{\alpha}}{|E|} = 0.0132$. Thus, we choose the free parameters ε and α of Algorithm 1 as computed above. This results in a slow down of Algorithm 1 with respect to its potential to allow a fair

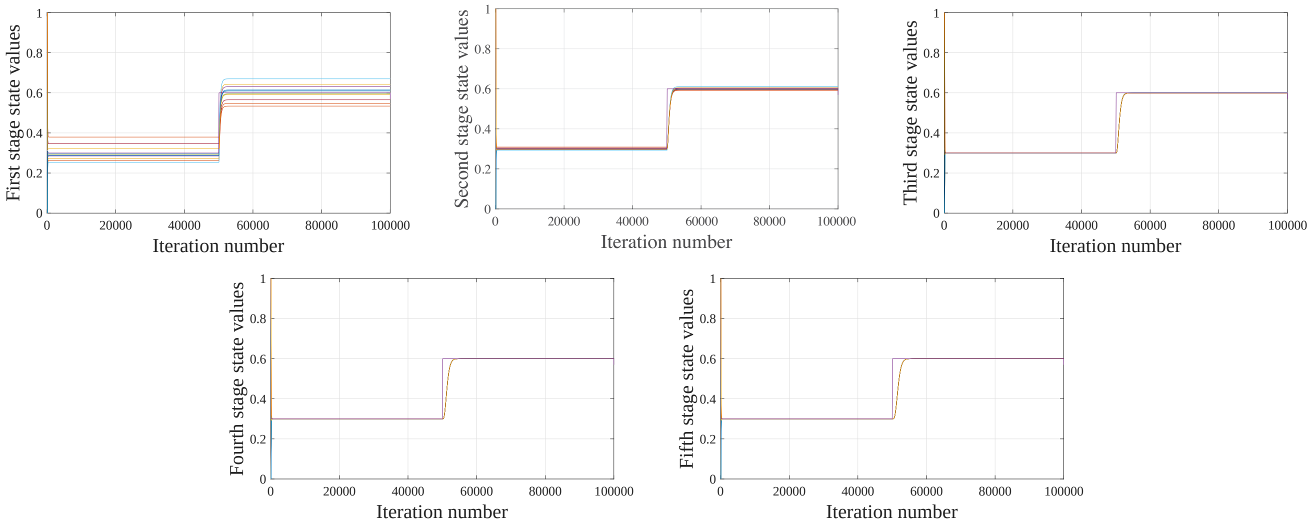

Fig. 1. State evolution of Algorithm 1 (Discrete-time protocol) with respect to a step change to constant input values.

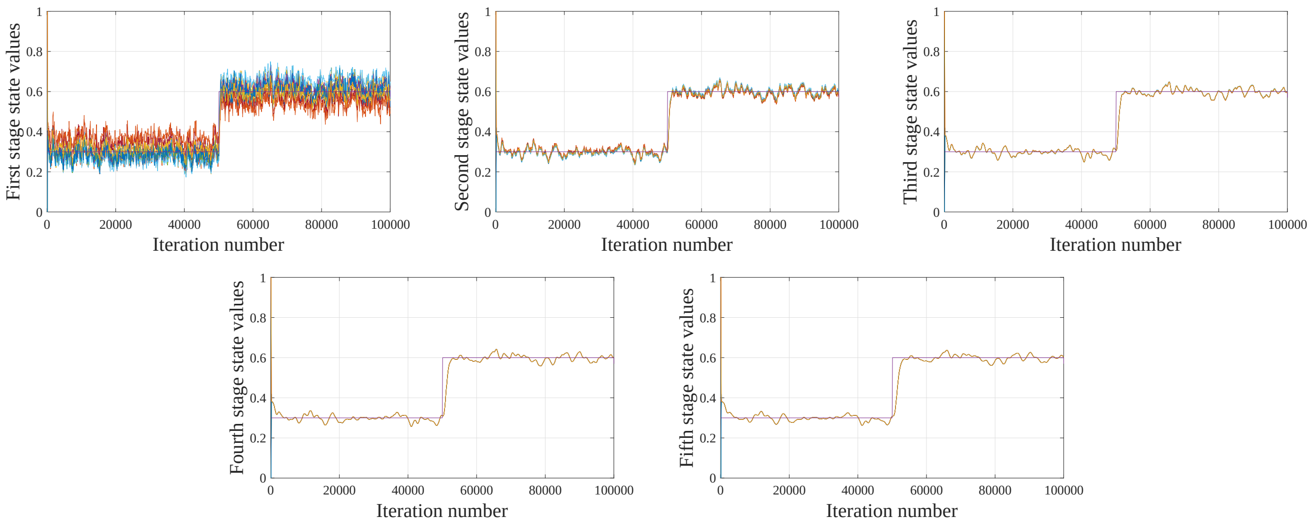

Fig. 2. State evolution of Algorithm 2 (Asynchronous and randomized protocol) with respect to a step change to constant input values.

comparison with Algorithm 2 which performs the update of a single random node at each iteration instead of an update that involves all nodes and their neighbors. We point out that in a real scenario the requirement of synchronicity of the updates of Algorithm 1 is a bottleneck in large scale network, thus making a randomized and asynchronous implementation to be preferable. The algebraic connectivity of the randomly generated graph resulted to be $\lambda_{L,2} = 1.5494$, thus according to Theorem 2 we expect a steady-state error for constant inputs equal to $\left(\frac{\alpha}{\beta}\right)^{10} \|\hat{\mathbf{u}}\|_2 = 7.69 \cdot 10^{-7} \|\hat{\mathbf{u}}\|_2$, where $\|\hat{\mathbf{u}}\|_2$ is the disagreement vector of the inputs.

In Figure 1 and 2 we show the evolution of, respectively, the first five stages of Algorithm 1 and Algorithm 2 executed in parallel on the same network, inputs and initial conditions. The simulations show how the algorithms behave with respect to a step change to constant input values. Only the average of the inputs is depicted in each subfigure. The change in average value occurs at iteration $k = 45000$. It can be seen that even a small number of stages of the proposed algorithm is able to greatly reduce the steady-state error.

In Figure 3 we consider a set of inputs composed by sinusoids with the same frequency but with random phase affected by noise. A comparison of the state evolution of the tenth stage of Algorithm 1 and Algorithm 2 can be seen, respectively, in the middle and right subfigures. It can be seen that while the disagreement among the agents' state is negligible, the average value of the error is finite and bounded.

In Figure 4 we consider a set of inputs composed by linear

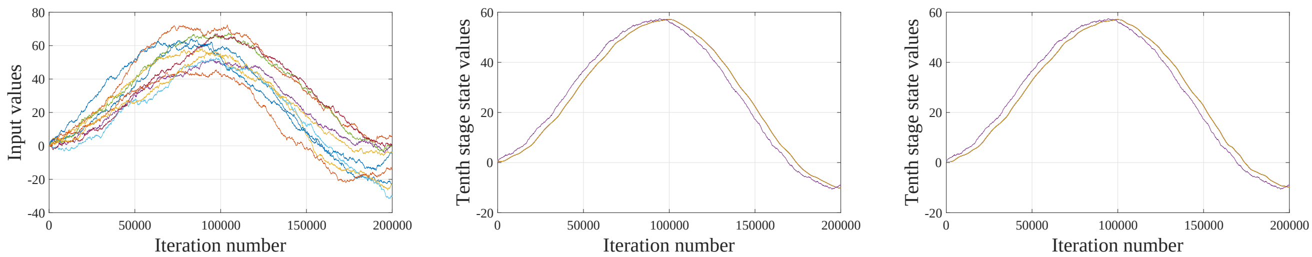

Fig. 3. Network inputs (left), tracking by Algorithm 1 (middle), tracking by Algorithm 2 (right).

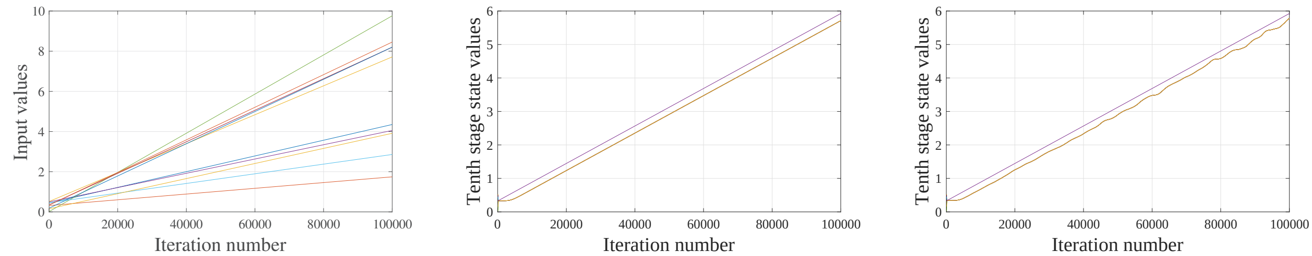

Fig. 4. Network inputs (left), tracking by Algorithm 1 (middle), tracking by Algorithm 2 (right).

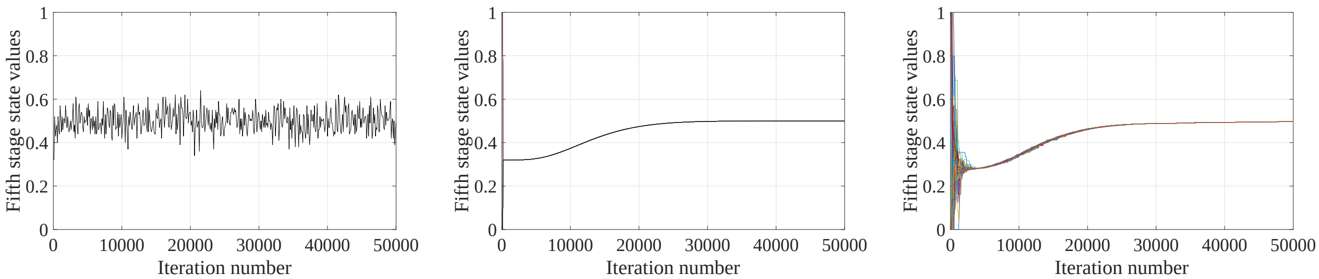

Fig. 5. Boolean random inputs. Instantaneous average of 100 random boolean inputs (left), tracking by Algorithm 1 (middle), tracking by Algorithm 2 (right).

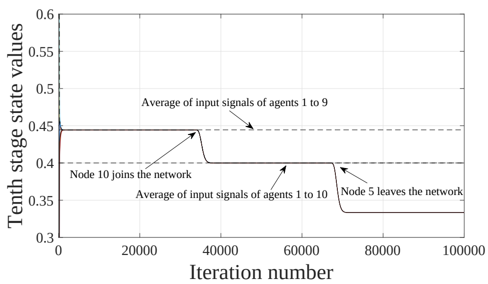

Fig. 6. Time-varying number of nodes with constant inputs. At iteration $k = 3.3 \cdot 10^4$ node 10 joins the network, at $k = 6.7 \cdot 10^4$ node 5 leaves the network. Algorithm 1 shows robustness to changes in the network composition.

ramps with random angular coefficient chosen uniformly at random in the interval $[0, 1]$. A comparison of the state evolution of the tenth stage of Algorithm 1 and Algorithm 2 can be seen, respectively, in the middle and right subfigures.

In Figure 5 we show a numerical example that strongly motivates the use of proposed algorithm design. We consider a network of $N = 100$ agents and a set of boolean inputs which takes $\{0, 1\}$ values randomly at each instant of time. It can be seen that the output of the multi-stage consensus filter first experiences a transient due to its initial condition and then, once it reaches the expected value of the average of the boolean inputs, it appears to be constant. It seems constant because changes in the instantaneous average of the inputs are filtered out by the "slow" dynamics (with respect to that of the inputs) of

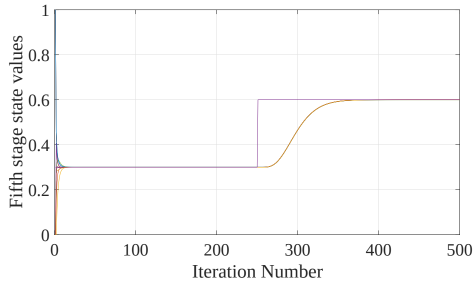

Fig. 7. State evolution of Algorithm 1 (Discrete-time protocol) tuned for high convergence rate with respect to a step change to constant input values.

the consensus filter and because the random process has a time-independent (constant) expected value. It is key to note here that only the expected value of the average of the inputs is constant, the actual instantaneous average of the inputs changes at each instant of time being it the realization of a stochastic process. In this example, our aim is to show that in this particular case the algorithm is able to converge to the expected average value filtering out the random changes of the inputs' average with respect to their expected value. Furthermore, this scenario intends to model what may occur during the transient behavior of distributed optimization algorithms such as the one in Franceschelli et al. (2016, 2018) where the input signals represent the planned trajectories of the energy consumption of a large population of thermostatically controlled loads, for instance electric thermal systems such as domestic water heaters. It can be seen that the proposed algorithms behave well despite the erratic behavior of the inputs. In particular, while the approach to discrete time dynamic average consensus proposed in Montijano et al. (2014) and related works is able to achieve great performance when the derivatives of the inputs are available and these derivatives predict future samples with small errors, i.e., there exist a model of the inputs, in this example (or in Franceschelli et al. (2016, 2018)) the input derivatives are meaningless and our approach which considers input signals as black boxes and does not exploit input derivatives seems to show superior performance.

In Figure 6 we consider the case when a new node joins the network during the execution of Algorithm 1. The example considers 9 agents with constant inputs up to iteration $k = 3.3 \cdot 10^4$ when at that point agent 10 joins the network, thus changing the average value of the input signals despite no input of the original 9 agents changed. At iteration $k = 6.7 \cdot 10^4$ node 5 leaves the network, altering again the average value of the input signals. It can be seen that the proposed approach shows robustness with respect to a time-varying number of nodes. In particular, since the proposed algorithm tracks the instantaneous average of the input signals with no memory of their past values, a change in the number of nodes does not have a significant effect on the convergence properties, which are preserved.

Finally, in Figure 7 we repeat a numerical simulation of Algorithm 1 with the same scenario but different tuning parameters to maximize convergence speed. We choose $\alpha = 0.1$ and $\varepsilon = 0.1357$. It can be seen that the number of iterations is greatly reduced with respect to the simulation in Figure 1 which was provided to show a fair comparison with the asynchronous and randomized version of the algorithm.

6 Conclusions

In this paper we proposed a novel dynamic consensus protocol which solves the dynamic average consensus problem on a set of time-varying input signals in an undirected graph. The proposed protocol is based on a multi-stage cascade of dynamic consensus filters which tracks the average value of the inputs with small and tunable error. Simulation results have been provided to corroborate the theoretical findings. The approach is robust with respect to re-initialization errors during runtime or measurement noise in the inputs because, differently from other dynamic consensus approaches, it does not exploit integral control actions or input derivatives. Furthermore, by tuning the number of stages of the cascade, i.e., by increasing the information locally exchanged at each state update, it is possible to tune the steady-state error and the maximum tracking error thus improving the performance with respect to proportional dynamic consensus. In addition, we provided an asynchronous and randomized version of the protocol which has the same advantages of the discrete time version and which exploits only directed interactions at each instant of time.

Future work will be focused on characterizing the performance of the protocol for networks with a time-varying number of agents and its use for the estimation of the average aggregate power consumption of thermostatically controlled loads in a scenario of distributed optimization of their energy consumption for electric demand side management applications.

7 Appendix

Fact 1 *Let matrices A and Q be defined as in eqs.* (5) *and* (6)*, respectively. Let $S = I - Q$ and $\binom{k}{i} = 0$ whenever*

$i > k$. The following holds

$$A^k = \left[\begin{array}{c:c:c:c:c} \binom{k}{0}S^k & \binom{k}{1}S^{k-1}\alpha & \binom{k}{2}S^{k-2}\alpha^2 & \dots & \binom{k}{m-1}S^{k-m+1}\alpha^{m-1} \\ \hdashline \mathbf{0} & \binom{k}{0}S^k & \binom{k}{1}S^{k-1}\alpha & \dots & \binom{k}{m-1}S^{k-m+2}\alpha^{m-2} \\ \hdashline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hdashline \mathbf{0} & \dots & \mathbf{0} & \binom{k}{0}S^k & \binom{k}{1}S^{k-1}\alpha \\ \hdashline \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \binom{k}{0}S^k \end{array}\right].$$

■

Fact 2 *Let us consider a matrix $T_{a,b} \in \mathbb{R}^{m\times m}$ defined as*

$$T_{a,b} = \left[\begin{array}{c:c:c:c:c} a & b & 0 & \dots & 0 \\ \hdashline 0 & a & b & \dots & 0 \\ \hdashline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hdashline 0 & 0 & \dots & a & b \\ \hdashline 0 & 0 & \dots & 0 & a \end{array}\right],$$

with $a \in (0,1)$ and $b \in (0,1)$.
The following holds true for the matrix $T^k_{a,b}$

$$T^k_{a,b} = \left[\begin{array}{c:c:c:c:c} \binom{k}{0}a^k & \binom{k}{1}a^{k-1}b & \binom{k}{2}a^{k-2}b^2 & \dots & \binom{k}{m-1}a^{k-m+1}b^{m-1} \\ \hdashline 0 & \binom{k}{0}a^k & \binom{k}{1}a^{k-1}b & \dots & \binom{k}{m-1}a^{k-m+2}b^{m-2} \\ \hdashline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hdashline 0 & \dots & 0 & \binom{k}{0}a^k & \binom{k}{1}a^{k-1}b \\ \hdashline 0 & \dots & 0 & 0 & \binom{k}{0}a^k \end{array}\right],$$

The following holds true for the matrix $(I - T_{a,b})^{-1}$

$$(I - T_{a,b})^{-1} = \left[\begin{array}{c:c:c:c:c} 1/a & b/a^2 & b^2/a^3 & \dots & b^{m-1}/a^m \\ \hdashline 0 & 1/a & b/a^2 & \dots & b^{m-2}/a^{m-1} \\ \hdashline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hdashline 0 & \dots & 0 & 1/a & b/a^2 \\ \hdashline 0 & \dots & 0 & 0 & 1/a \end{array}\right].$$

Lemma 1 *Let $\alpha \in (0,1)$ and $\beta \in (\alpha,1)$. It holds*

$$\sum_{\ell=0}^{\infty}\sum_{s=0}^{m-1}\binom{\ell}{s}(1-\beta)^{\ell-s}\alpha^s\left(\frac{\alpha}{\beta}\right)^{m-s} = \frac{m}{\beta}\left(\frac{\alpha}{\beta}\right)^m. \quad (20)$$

Proof:
Let us denote by $y(k) \in \mathbb{R}$ the output of the next linear system

$$\begin{aligned} z(k+1) &= T_{1-\beta,\alpha}\, y(k) + G_{\alpha,\beta}\, u, \\ y(k+1) &= C\, z(k+1). \end{aligned} \quad (21)$$

where $z(k) \in \mathbb{R}^m$, $u \in \mathbb{R}$ is a constant input, $C = [1, 0, \dots, 0] \in \mathbb{R}^m$, $T_{1-\beta,\alpha}$ is a matrix defined according to Fact 2 and vector $G_{\alpha,\beta}$ is defined as

$$G_{\alpha,\beta} = \left[\begin{array}{c:c:c:c:c} (\alpha/\beta)^m & \dots & (\alpha/\beta)^s & \dots & (\alpha/\beta) \end{array}\right].$$

Now by exploiting Fact 2, we can compute the forced response of the system in eq. (21) as

$$y(k) = C\sum_{\ell=0}^{k-1} T^{k-\ell-1}_{1-\beta,\alpha} G_{\alpha,\beta}\, u, \quad (22)$$

which can be rewritten as

$$y(k) = u\sum_{l=0}^{k-1}\sum_{s=0}^{m-1}\binom{k-l-1}{s}(1-\beta)^{k-l-1-s}\alpha^s\left(\frac{\alpha}{\beta}\right)^{m-s}. \quad (23)$$

At this point by letting $\ell = k - l - 1$, we obtain

$$y(k) = u\sum_{\ell=0}^{k-1}\sum_{s=0}^{m-1}\binom{\ell}{s}(1-\beta)^{\ell-s}\alpha^s\left(\frac{\alpha}{\beta}\right)^{m-s}. \quad (24)$$

Since for $\alpha \in (0,1)$ and $\beta \in (\alpha,1)$ the system in eq. (21) is asymptotically stable, the steady state for a constant input exists. We are now interested in computing the steady state value obtained for $k \to \infty$, i.e.,

$$y(\infty) = u\sum_{\ell=0}^{\infty}\sum_{s=0}^{m-1}\binom{\ell}{s}(1-\beta)^{\ell-s}\alpha^s\left(\frac{\alpha}{\beta}\right)^{m-s}, \quad (25)$$

Since the system is asymptotically stable, such steady state value can be computed as

$$y(\infty) = C\,(I - T_{1-\beta,\alpha})^{-1}\, G_{\alpha,\beta}\, u.$$

Now, by recalling the closed-form for the matrix $(I - T_{1-\beta,\alpha})^{-1}$ from Fact 2, it holds

$$C\,(I - T_{1-\beta,\alpha})^{-1}\, G_{\alpha,\beta} =$$

$$= C\left[\begin{array}{c:c:c:c:c} 1/\beta & \alpha/\beta^2 & \alpha^2/\beta^3 & \dots & \alpha^{m-1}/\beta^m \\ \hdashline \mathbf{0} & 1/\beta & \alpha/\beta^2 & \dots & \alpha^{m-2}/\beta^{m-1} \\ \hdashline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hdashline \mathbf{0} & \dots & \mathbf{0} & 1/\beta & \alpha/\beta^2 \\ \hdashline \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & 1/\beta \end{array}\right]\left[\begin{array}{c} (\alpha/\beta)^m \\ \dots \\ (\alpha/\beta)^s \\ \dots \\ (\alpha/\beta) \end{array}\right], \quad (26)$$

which by means of some manipulations it can be rewritten as

$$C\,(I - T_{1-\beta,\alpha})^{-1}\, G_{\alpha,\beta} = \sum_{s=0}^{m-1}\frac{\alpha^s}{\beta^{s+1}}\frac{\alpha^{m-s}}{\beta^{m-s}}. \quad (27)$$

Thus, the output of system eq. (21) at steady-state is

$$y_\beta(\infty) = uC\,(I - T_{1-\beta,\alpha})^{-1}\,G_{\alpha,\beta} = u\frac{m}{\beta}\left(\frac{\alpha}{\beta}\right)^m. \tag{28}$$

By equating the steady-state value computed in eq. (25) with that computed in eq. (28) and dividing both for the constant input value $u \neq 0$ we find that

$$\sum_{\ell=0}^{\infty}\sum_{s=0}^{m-1}\binom{\ell}{s}(1-\beta)^{\ell-s}\alpha^s\left(\frac{\alpha}{\beta}\right)^{m-s} = \frac{m}{\beta}\left(\frac{\alpha}{\beta}\right)^m, \tag{29}$$

which proves the equality in eq. (20) in the statement of this lemma.

7.1 **Proof of Theorem 1**

First, let us notice that $\alpha > 0$ and L is symmetric and positive semi-definite. From 2 we notice that the first system (stage) can be written in a compact form as:

$$\begin{aligned} x^1(k+1) &= \big(I - \varepsilon L\big)\,x^1(k) + \alpha\,\big(u(k) - x^1(k)\big), \\ &= \big((1-\alpha)I - \varepsilon L\big)\,x^1(k) + \alpha u(k). \end{aligned} \tag{30}$$

At this point, we notice that by construction the matrix $(1-\alpha)I - \varepsilon L$ is symmetric, thus all the eigenvalues are real and have the following form

$$\lambda_i = 1 - \alpha - \varepsilon\lambda_{L,i}, \quad \lambda_{L,i} \in \sigma(L). \tag{31}$$

At this point, by assuming $\alpha \in (0, 1 - \varepsilon D_{max})$ and $\varepsilon \in \left(0, \frac{1}{2D_{max}}\right)$ it follows that by construction the eigenvalues defined in (31) are such that $\lambda_i \in (0,1),\ i \in \{1, \ldots, n\}$. Thus the asymptotic stability of the dynamical system in (30) follows. Notably, the same reasoning applies to the remaining $m-1$ stages, being the dynamical matrix identical for all stages. In particular, the steady state equilibrium for the the first system (stage) is

$$x^{1,\star} = (\alpha I + \varepsilon L)^{-1}\,\alpha\,\mathbf{u}. \tag{32}$$

Since the system is a cascade, for $s > 1$ it holds that

$$x^{s,\star} = (\alpha I + \varepsilon L)^{-1}\,\alpha\,x^{s-1,\star}. \tag{33}$$

Thus, from eq. (32) and (33) we can easily compute the equilibrium point of the s-th stage as

$$x^{s,\star} = (\alpha I + \varepsilon L)^{-s}\,\alpha^s\,\mathbf{u}.$$

Thus proving the statement. □

7.2 **Proof of Theorem 2**

To prove the result, let us recall that from Theorem 1 we know that the steady-state equilibrium for the s-th stage is

$$x^{s,\star} = (\alpha I + \varepsilon L)^{-s}\,\alpha^s\,\mathbf{u}. \tag{34}$$

The eigenvalues of the matrix $(\alpha I + \varepsilon L)^{-s}$ are:

$$\hat{\lambda}_i = \left(\frac{1}{\alpha + \varepsilon\,\lambda_{L,i}}\right)^s, \quad \lambda_{L,i} \in \sigma(L),$$

where $\lambda_{L,i}$ denotes the i-th eigenvalue of matrix L, v_i the corresponding eigenvector and $\sigma(L)$ the spectrum of matrix L. If graph $\mathcal{G}$ is undirected and connected, L is symmetric and has rank $n-1$ with a single one null eigenvalue with unitary geometric multiplicity. The eigenvector corresponding to the null eigenvalue has identical elements. At this point, by recalling that for a symmetric matrix L the eigenvectors $V = [v_1, \ldots, v_n]$ are orthonormal, i.e., $V\,V^T = I$, and thus the matrix L can be expressed as a linear combination of 1-dimensional projections as $L = V\Lambda_L V^T = \sum_{i=1}^n \lambda_L v_i\,v_i^T$ with $\Lambda_L = \mathrm{diag}[\lambda_{L,1}, \ldots, \lambda_{L,n}]$, we have that (34) can be rewritten as

$$\begin{aligned} x^{s,\star} &= (\alpha I + \varepsilon L)^{-s}\,\alpha^s\,\mathbf{u} \\ &= (\alpha V\,I\,V^T + \varepsilon V\Lambda_L V^T)^{-s}\,\alpha^s\,\mathbf{u} \\ &= \big(V\,(\alpha I + \varepsilon\Lambda_L)V^T\big)^{-s}\,\alpha^s\,\mathbf{u} \\ &= V\,(\alpha I + \varepsilon\Lambda_L)^{-s}\,V^T\,\alpha^s\,\mathbf{u} \\ &= \sum_{i=1}^n\left(\frac{1}{\alpha + (\varepsilon\,\lambda_{L,i})}\right)^s v_i v_i^T\,\alpha^s\,\mathbf{u} \\ &= \sum_{i=1}^n\left(\frac{\alpha}{\alpha + \varepsilon\,\lambda_{L,i}}\right)^s v_i v_i^T\,\mathbf{u} \\ &= \bar{u}\,\mathbf{1} + \sum_{i=2}^n\left(\frac{\alpha}{\alpha + \varepsilon\,\lambda_{L,i}}\right)^s v_i v_i^T\,\mathbf{u}, \end{aligned} \tag{35}$$

where the fact $(\alpha I + \varepsilon L)^{-1}\mathbf{1} = \alpha^{-1}\mathbf{1}$ has been used.

By substituting the expression given in (35) for the steady state equilibrium of the m-th stage as given in (35) into $\|\bar{u}(k)\,\mathbf{1} - x^{m,\star}(k)\|_2$ and noticing that $\hat{\mathbf{u}}(k) = (\mathbf{u}(k) - \bar{u}(k)\,\mathbf{1})$ it holds

$$\begin{aligned} \|\bar{u}(k)\,\mathbf{1} - x^{m,\star}\|_2 &= \left\|\sum_{i=2}^n\left(\frac{\alpha}{\alpha + \varepsilon\,\lambda_{L,i}}\right)^m v_i v_i^T\,\mathbf{u}(k)\right\|_2 \\ &\leq \left(\frac{\alpha}{\alpha + \varepsilon\,\lambda_{L,2}}\right)^m\left\|\sum_{i=2}^n v_i v_i^T\,\mathbf{u}(k)\right\|_2, \end{aligned} \tag{36}$$

At this point, by recalling that the matrix L is symmetric, and thus the eigenvectors corresponding to different eigenvalues must be orthogonal to each other, it follows that $v_i^T\,v_1 = 0$ for $i \in 2, \ldots, n$ with $v_1 = \mathbf{1}$. Thus we obtain

$$\|\bar{u}(k)\,\mathbf{1} - x^{m,\star}\|_2 \le \left(\frac{\alpha}{\beta}\right)^m \|\hat{\mathbf{u}}(k)\|_2\,, \tag{37}$$

thus proving the statement.

□

7.3 *Proof of Proposition 1*

To prove the statement of this proposition we manipulate the error at time $k+1$ as follows

$$\begin{aligned} e(k+1) &= x(k+1) - x^\star(k+1) \\ &= Ax(k) + Bu(k) - x^\star(k+1) \\ &= Ax(k) + Ax^\star(k) - Ax^\star(k) + Bu(k) \\ &\quad - x^\star(k+1) \\ &= Ae(k) + Bu(k) + Ax^\star(k) - x^\star(k+1). \end{aligned} \tag{38}$$

Since vector $x^\star(k)$ represents the equilibrium point for constant inputs and for any given $u(k)$, it satisfies by definition the next equality

$$x^\star(k) = Ax^\star(k) + Bu(k),$$

it follows that eq. (38) can be further simplified as

$$\begin{aligned} e(k+1) &= Ae(k) + Ax^\star(k) + Bu(k) - x^\star(k+1) \\ &= Ae(k) + x^\star(k) - x^\star(k+1) \\ &= Ae(k) + \delta^\star(k). \end{aligned}$$

thus proving the statement □

7.4 *Proof of Theorem 3*

Regarding the natural response of the error dynamics, due to Fact 1, it holds

$$\begin{aligned} e_n^m(k) &= C\,A^k e(0) \\ &= \sum_{s=0}^{m-1} \binom{k}{s} S^{k-s}\alpha^s e^{m-s}(0) \\ &= \sum_{s=0}^{m-1} \binom{k}{s} (I-Q)^{k-s}\alpha^s e^{m-s}(0), \end{aligned} \tag{39}$$

Now, let us define

$$\begin{aligned} e_{max}(0) &= \max_{s\in\{0,\dots,m-1\}} \left\{\|e^{m-s}(0)\|_2\right\} \\ &= \max_{s\in\{1,\dots,m\}} \left\{\|e^{s}(0)\|_2\right\}, \end{aligned} \tag{40}$$

it holds

$$\begin{aligned} \|e_n^m(k)\|_2 &= \left\| \sum_{s=0}^{m-1} \binom{k}{s} (I-Q)^{k-s}\alpha^s e^{m-s}(0) \right\|_2 \\ &\le \sum_{s=0}^{m-1} \binom{k}{s} \left\|(I-Q)^{k-s}\alpha^s e^{m-s}(0)\right\|_2 \\ &\le \frac{k^{m-1}}{(m-1)!} \sum_{s=0}^{m-1} \left\|(I-Q)^{k-s}\right\|_2 \alpha^s \left\|e^{m-s}(0)\right\|_2 \\ &\le \frac{k^{m-1}}{(m-1)!} \sum_{s=0}^{m-1} (1-\alpha)^{k-s}\alpha^s \left\|e^{m-s}(0)\right\|_2 \\ &\le \frac{k^{m-1}}{(m-1)!} e_{max}(0) \sum_{s=0}^{m-1} (1-\alpha)^{k-s}\alpha^s \\ &\le (1-\alpha)^k \frac{k^{m-1}}{(m-1)!} \frac{1-\left(\frac{\alpha}{1-\alpha}\right)^m}{1-\frac{\alpha}{1-\alpha}} e_{max}(0), \end{aligned}$$

thus proving the statement of the theorem. □

7.5 *Proof of Theorem 4*

Since $\alpha \in (0, \varepsilon D_{max})$ and $\varepsilon \in \left(0, \frac{1}{2D_{max}}\right)$, the considered cascade of systems is asymptotically stable. Thus, let us now consider the forced response of the error dynamics in eq. (9)

$$e(k) = \sum_{l=0}^{k-1} A^{k-l-1}\delta^\star(l).$$

Thus, we have that $e^m(k)$ can be written as

$$e^m(k) = \sum_{l=0}^{k-1} C\,A^{k-l-1}\delta^\star(l). \tag{41}$$

Due to **Fact 1** from the Appendix, eq. (41) becomes

$$e^m(k) = \sum_{l=0}^{k-1}\sum_{s=0}^{m-1} \binom{k-l-1}{s} (I-Q)^{k-l-1-s}\alpha^s \delta^{m-s,\star}(l).$$

To simplify the notation, let $\ell = k-l-1$. We substitute the summation in l for $l = 0,\dots,k-1$ with the summation in ℓ for $\ell = 0,\dots,k-1$ as follows

$$e^m(k) = \sum_{\ell=0}^{k-1}\sum_{s=0}^{m-1} \binom{\ell}{s} (I-Q)^{\ell-s}\alpha^s \delta^{m-s,\star}(k-\ell-1). \tag{42}$$

In order to analyze the average error value $e^m(k)$ and the disagreement error vector $\hat{e}^m(k)$, let us now recall that $x^{s,\star}(k)$ can be written as

$$x^{s,\star}(k) = Q^{-s}\alpha^s\mathbf{u}(k),$$

with $s = 1, \ldots, m$ the index of the stage of the cascade consensus filter. It follows that $\delta^{s,\star}(k)$ can be equivalently expressed as

$$\begin{aligned}
\delta^{s,\star}(k) &= Q^{-s}\alpha^s \mathbf{u}(k) - Q^{-s}\alpha^s \mathbf{u}(k+1) \\
&= Q^{-s}\alpha^s \left(\mathbf{u}(k) - \mathbf{u}(k+1)\right) \\
&= Q^{-s}\alpha^s \delta_u(k) \\
&= Q^{-s}\alpha^s \left(\bar{\delta}_u(k)\mathbf{1} + \hat{\delta}_u(k)\right) \\
&= \alpha^{-s}\alpha^s \bar{\delta}_u(k)\mathbf{1} + Q^{-s}\alpha^s \hat{\delta}_u(k) \\
&= \bar{\delta}_u(k)\mathbf{1} + Q^{-s}\alpha^s \hat{\delta}_u(k),
\end{aligned} \tag{43}$$

with $\delta_u(k) = \mathbf{u}(k) - \mathbf{u}(k+1)$ and $\delta_u(k) = \bar{\delta}_u(k)\mathbf{1} + \hat{\delta}_u(k)$ Regarding the average error value $\bar{e}^m(k)$, from (42) it follows that it can be written as

$$\bar{e}^m(k) = \frac{\mathbf{1}^T}{n}\sum_{\ell=0}^{k-1}\sum_{s=0}^{m-1}\binom{\ell}{s}(I-Q)^{\ell-s}\alpha^s \delta^{m-s,\star}(k-\ell-1).$$

Now, since $\mathbf{1}^T(I-Q) = (1-\alpha)\mathbf{1}^T$ it holds

$$\begin{aligned}
\bar{e}^m(k) &= \sum_{\ell=0}^{k-1}\sum_{s=0}^{m-1}\binom{\ell}{s}(1-\alpha)^{\ell-s}\alpha^s \frac{\mathbf{1}^T\delta^{m-s,\star}(k-\ell-1)}{n} \\
&= \sum_{\ell=0}^{k-1}\sum_{s=0}^{m-1}\binom{\ell}{s}(1-\alpha)^{\ell-s}\alpha^s \bar{\delta}_u(k-\ell-1).
\end{aligned}$$

where eq. (43) has been used. At this point, let $|\bar{\delta}_u(k-\ell-1)| < \bar{\delta}_{u,max}$ for all $k, \ell = 1, \ldots, \infty$, the previous equation can be rewritten as

$$|\bar{e}^m(k)| \leq \bar{\delta}_{u,max}\sum_{s=0}^{m-1}\sum_{\ell=0}^{k-1}\binom{\ell}{s}(1-\alpha)^{\ell-s}\alpha^s. \tag{44}$$

Thus by eq. (44), we can establish that the maximum error is bounded by the error for k at infinity, that is

$$|\bar{e}^m(k)| \leq \bar{\delta}_{u,max}\sum_{s=0}^{m-1}\sum_{\ell=0}^{\infty}\binom{\ell}{s}(1-\alpha)^{\ell-s}\alpha^s. \tag{45}$$

At this point, by resorting to Lemma 1, we can compute an upper bound for $|\bar{e}^m|$ as

$$|\bar{e}^m(k)| \leq \frac{m}{\alpha}\,\bar{\delta}_{u,max}. \tag{46}$$

Regarding the disagreement error vector $\hat{e}^m(k)$, by substituting eq. (42) within eq. (11) and by noticing that $P(I-Q) = (I-Q)P$ and $P^2 = P$, we obtain

$$\hat{e}^m(k) = \sum_{s=0}^{m-1}\sum_{\ell=0}^{k-1}\binom{\ell}{s}(I-Q)^{\ell-s}\alpha^s Q^{-(m-s)}\alpha^{m-s} \hat{\delta}_u(k-\ell-1), \tag{47}$$

where eq. (43) has been used again along with the fact that $P\delta_u(k) = \hat{\delta}_u(k)$. Furthermore, by noticing that for any p, $q \in \mathbb{R}^+$ the following holds

$$\|(I-Q)^p\, Q^{-q}\, \hat{\delta}_u(\ell)\|_2 \leq (1-\alpha-\varepsilon\lambda_{L,2})^p\, (\alpha+\varepsilon\lambda_{L,2})^{-q}\, \hat{\delta}_{u,max},$$

by letting $\beta = \alpha + \varepsilon\lambda_{L,2}$, we have that eq. (47) can be bounded as

$$\|\hat{e}^m(k)\| \leq \hat{\delta}_{u,max}\sum_{s=0}^{m-1}\sum_{\ell=0}^{k-1}\binom{\ell}{s}(1-\beta)^{\ell-s}\alpha^s \alpha^{m-s}\beta^{-(m-s)}, \tag{48}$$

Thus by eq. (47), we can establish that the maximum error is bounded by the error for k at infinity, that is

$$\|\hat{e}^m(k)\| \leq \hat{\delta}_{u,max}\sum_{s=0}^{m-1}\sum_{\ell=0}^{\infty}\binom{\ell}{s}(1-\beta)^{\ell-s}\alpha^s \alpha^{m-s}\beta^{-(m-s)}, \tag{49}$$

At this point, since due to Lemma 1 it holds

$$\sum_{\ell=0}^{\infty}\sum_{s=0}^{m-1}\binom{\ell}{s}(1-\beta)^{\ell-s}\alpha^s\alpha^{m-s}\beta^{-(m-s)} = \frac{m}{\beta}\left(\frac{\alpha}{\beta}\right)^m, \tag{50}$$

we find that an upper bound for $|\hat{e}^m|$ is as follows

$$|\hat{e}^m(k)| \leq \frac{m}{\beta}\left(\frac{\alpha}{\beta}\right)^m \hat{\delta}_{u,max}. \tag{51}$$

7.6 *Proof of Theorem 5*

We can represent each state update of Algorithm 2 as follows:

$$\begin{aligned}
x(k+1) &= A(k)x(k) + B(k)u(k) \\
x^m(k) &= Cx(k),
\end{aligned}$$

where $x(k) = \left[x^m(k)^T, \ldots, x^1(k)^T\right]^T$, $A(k) \in \mathbb{R}^{nm\times nm}$ is defined as (given that at iteration k agent i selects agent j for one iteration)

$$A(k) = A_{ij} = \left[\begin{array}{c:c:c:c:c} W_{ij} & \dfrac{\hat{\alpha}}{D_i} e_i e_i^T & \mathbf{0} & \dots & \mathbf{0} \\ \hdashline \mathbf{0} & W_{ij} & \dfrac{\hat{\alpha}}{D_i} e_i e_i^T & \dots & \mathbf{0} \\ \hdashline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hdashline \mathbf{0} & \dots & \mathbf{0} & W_{ij} & \dfrac{\hat{\alpha}}{D_i} e_i e_i^T \\ \hdashline \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & W_{ij} \end{array}\right],$$

matrix $B(k) \in \mathbb{R}^{nm \times n}$ is defined as (given that agent i is updating its state at iteration k)

$$B(k) = B_i = \left[\begin{array}{c} \mathbf{0} \\ \hdashline \vdots \\ \hdashline \mathbf{0} \\ \hdashline \dfrac{\hat{\alpha}}{D_i} e_i e_i^T \end{array}\right],$$

and matrix $C \in \mathbb{R}^{n \times nm}$ defined as

$$C = \left[\begin{array}{c:c:c:c} I & \mathbf{0} & \dots & \mathbf{0} \end{array}\right].$$

Since the edges (i,j) have uniform probability p to be chosen, such that $\sum_{(i,j)\in E} p = 1$, it holds $p = \frac{1}{|E|}$. We now compute

$$\mathbb{E}\left[A(k)\right] = \frac{1}{|E|} \sum_{(i,j)\in E} A_{ij}.$$

First, we note that

$$\sum_{(i,j)\in E} e_i e_i^T = D,$$

and, by exploiting eq. (18), it holds

$$\begin{aligned} \frac{1}{|E|} \sum_{(i,j)\in E} W_{ij} &= \frac{1}{|E|} \sum_{(i,j)\in E} I + \frac{e_i e_j^T}{2} - \frac{(1 + 2\hat{\alpha}/D_i) e_i e_i^T}{2} \\ &= I - \frac{1}{|E|}\left(\frac{1}{2} L + \hat{\alpha} I\right). \end{aligned}$$

Then, by denoting $\varepsilon = \dfrac{1}{2|E|}$, $\alpha = \dfrac{\hat{\alpha}}{|E|}$ and

$$Q = \alpha\, I + \varepsilon\, L, \tag{52}$$

it holds

$$\mathbb{E}\left[A(k)\right] = \left[\begin{array}{c:c:c:c:c} I-Q & \alpha\, I & \mathbf{0} & \dots & \mathbf{0} \\ \hdashline \mathbf{0} & I-Q & \alpha\, I & \dots & \mathbf{0} \\ \hdashline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hdashline \mathbf{0} & \dots & \mathbf{0} & I-Q & \alpha\, I \\ \hdashline \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & I-Q \end{array}\right].$$

Thus, the dynamics of protocol (16) is in expectation formally equivalent to that of protocol (1). If both matrix A and $\mathbb{E}\left[A(k)\right]$ are Shur stable it holds that

$$\begin{aligned} \mathbb{E}\left[x(k+1)\right] &= \mathbb{E}\left[A(k)\right] \mathbb{E}\left[x(k)\right] + \mathbb{E}\left[B(k)\right] u(k) \\ &= A\mathbb{E}\left[x(k)\right] + Bu(k). \end{aligned}$$

At this point, let us recall that the graph $\mathcal{G}$ is assumed to be undirected and connected, thus matrix L and $I - Q$ are symmetric positive semi-definite. Therefore, if $\varepsilon\lambda_n + \alpha < 1$, the system is Shur stable, i.e., with eigenvalues strictly inside the unit circle. To prove this, notice that with $\varepsilon = \frac{1}{2|E|}$ and $\alpha = \frac{\hat{\alpha}}{|E|}$, since $|E| = \text{trace}(L) = \sum_{i=1}^{n} \lambda_i \leq \lambda_n$, it holds that $\varepsilon\lambda_n \leq \frac{1}{2}$. Thus, if $|E| > 2$ by choosing $\hat{\alpha} \in (0, 0.5)$ the inequality $\varepsilon\lambda_n + \alpha < 1$ is always satisfied. □

7.7 ***Proof of Theorem 6***

To prove that $x^m(k)$ almost surely converges in distribution to a unique invariant random variable $\mathbf{x}_\infty$ we exploit the results in Ravazzi et al. (2015). Briefly, in that paper the authors characterized convergence in distribution for general randomized affine dynamics. Algorithm 2 is a randomized version of Algorithm 1 which is modeled by discrete time, Shur stable, affine dynamics and the sequence of selected edges is i.i.d. with uniform distribution. In the particular case of constant inputs, the conditions of in Theorem 1 in Ravazzi et al. (2015) are satisfied, thus $x^m(k)$ converges in distribution to a random variable x^m_∞ and this distribution is unique and it holds that

$$\lim_{k\to\infty} E\left[x^m(k)\right] = E\left[x^m_\infty\right] = x^{m,\star}.$$

□

References

R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

M. Franceschelli, A. Giua, and C. Seatzu. Consensus on the average on arbitrary strongly connected digraphs based on broadcast gossip algorithms. In *1st IFAC Workshop on Estimation and Control of Networked Systems*, pages 66–71, Sep 2009.

K. Cai and H. Ishii. Average consensus on general strongly connected digraphs. *Automatica*, 48(11): 2750–2761, 2012.

Alejandro D Domínguez-García and Christoforos N Hadjicostis. Distributed strategies for average consensus in directed graphs. In *IEEE 50th Conference on Decision and Control and European Control Conference*, pages 2124–2129, 2011.

Eduardo Montijano, Andrea Gasparri, Attilio Priolo, and Carlos Sagues. Average consensus on strongly connected weighted digraphs: A generalized error bound. *Automatica*, 58(0):1 – 4, 2015.

Wenwu Yu, Guanrong Chen, and Ming Cao. Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems. *Automatica*, 46(6):1089 – 1095, 2010.

D. Bauso, L. Giarré, and R. Pesenti. Non-linear protocols for optimal distributed consensus in networks of dynamic agents. *Systems & Control Letters*, 55(11): 918 – 928, 2006.

F. S. Cattivelli and A. H. Sayed. Diffusion strategies for distributed kalman filtering and smoothing. *IEEE Transactions on Automatic Control*, 55(9):2069–2084, 2010.

M. Franceschelli and A. Gasparri. Gossip-based centroid and common reference frame estimation in multiagent systems. *IEEE Transactions on Robotics*, 30(2):524–531, 2014.

R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Optimal synchronization for networks of noisy double integrators. *IEEE Transactions on Automatic Control*, 56(5):1146–1152, 2011.

Emanuele Garone, Andrea Gasparri, and Francesco Lamonaca. Clock synchronization protocol for wireless sensor networks with bounded communication delays. *Automatica*, 59:60 – 72, 2015.

Demetri P Spanos, Reza Olfati-Saber, and Richard M Murray. Dynamic consensus on mobile networks. In *IFAC world congress*, pages 1–6, 2005.

Shahram Nosrati, Masoud Shafiee, and Mohammad Bagher Menhaj. Dynamic average consensus via nonlinear protocols. *Automatica*, 48(9):2262 – 2270, 2012.

Solmaz S. Kia, Jorge Cortés, and Sonia Martínez. Distributed event-triggered communication for dynamic average consensus in networked systems. *Automatica*, 59:112 – 119, 2015a.

Minghui Zhu and Sonia Martínez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322 – 329, 2010.

Eduardo Montijano, Juan Ignacio Montijano, Carlos Sagüés, and Sonia Martínez. Robust discrete time dynamic average consensus. *Automatica*, 50(12):3131 – 3138, 2014.

R. A. Freeman, P. Yang, and K. M. Lynch. Stability and convergence properties of dynamic average consensus estimators. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 338–343, Dec 2006.

H. Bai, R. A. Freeman, and K. M. Lynch. Robust dynamic average consensus of time-varying inputs. In *49th IEEE Conference on Decision and Control*, pages 3104–3109, Dec 2010.

Bryan Van Scoy, Randy A. Freeman, and Kevin M. Lynch. A fast robust nonlinear dynamic average consensus estimator in discrete time. *5th IFAC Workshop on Distributed Estimation and Control in Networked Systems NecSys 2015*, 48(22):191 – 196, 2015.

S. S. Kia, J. Cortés, and S. Martínez. Dynamic average consensus under limited control authority and privacy requirements. *International Journal of Robust and Nonlinear Control*, 25(13):1941–1966, 2015b.

S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52 (6):2508–2530, 2006.

A.G. Dimakis, S. Kar, J.M.F. Moura, M.G. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, Nov 2010.

G. Habibi, Z. Kingston, Z. Wang, M. Schwager, and J. McLurkin. Pipelined consensus for global state estimation in multi-agent systems. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, volume 2, pages 1315–1323, 2015.

Chiara Ravazzi, Paolo Frasca, Roberto Tempo, and Hideaki Ishii. Ergodic randomized algorithms and dynamics over networks. *IEEE Transactions on Control of Network Systems*, 2(1):78–87, 2015.

M. Franceschelli, A. Gasparri, and A. Pisano. Coordination of electric thermal systems for distributed demand-side management: A gossip-based cooperative approach. In *16th annual European Control Conference*, pages 623–630, June 2016.

M. Franceschelli, A. Pilloni, and A. Gasparri. A heuristic approach for online distributed optimization of multiagent networks of smart sockets and thermostatically controlled loads based on dynamic average consensus. In *17th annual European Control Conference*, June 2018.