Dynamic Min and Max Consensus and Size Estimation of Anonymous Multi-Agent Networks

D. Deplano, *Member, IEEE*, M. Franceschelli, *Member, IEEE*, and A. Giua, *Fellow, IEEE*

***Abstract*—In this paper we propose two distributed control protocols for discrete-time multi-agent systems (MAS), which solve the dynamic consensus problem on the max value. In this problem each agent is fed an exogenous reference signal and has the objective to estimate and track the instantaneous and time-varying value of the maximum among all the signals fed to the network by exploiting only local and anonymous interactions among the agents. The first protocol achieves bounded steady-state and tracking errors which can be traded-off for convergence time. The second protocol achieves zero steady-state error and requires knowledge of an upper bound to the diameter of the graph representing the network. Modified versions of both protocols are provided to solve the dual dynamic min-consensus problem. These protocols are then exploited to solve a distributed size estimation problem in a network of anonymous agents in a dynamic setting where the size of the network is time-varying during the execution of the estimation algorithm. Numerical simulations are provided in order to corroborate the characterization of the proposed protocols.**

***Index Terms*—Dynamic consensus, max consensus, distributed estimation, multi-agent systems, network size estimation, anonymous networks.**

I. Introduction

In the past decade there has been a significant interest in the design of distributed algorithms to solve the consensus or agreement problem over networked systems.

Problem of interest and motivation. In a consensus problem the agents agree upon a state value by making only use of local information coming from neighboring agents. In its simplest formulation, the consensus problem considers autonomous multi-agent systems, i.e., the agents are required to converge to a state value which is function of the initial state of the network. On the contrary, in the dynamic consensus problem the agents are assumed to be non-autonomous and are required to converge to a state value which is a function of the time-varying reference signals given as input to the agents, most commonly the average [23] and the median value [32], [37], or, not yet explored, the minimum and maximum value. The *dynamic max/min-consensus problem* is thus the focus of this paper. We develop dynamic consensus protocols which achieve consensus and track the value of the maximum or, alternatively, the minimum value among the set of reference signals. Existing applications of max/min-consensus protocols involve monitoring and optimization [21]; distributed synchronization, such as time-synchronization [10] and target tracking [31]; network parameter estimation, such as cardinality [24], diameter and radius [12], as well as highest/lowest node degree [5].

Related literature. The so-called max-consensus problem has been thoroughly investigated. Its objective is to make the states of a network of agents converge to the maximum of their initial states [30]. First protocols solving the max-consensus problem have been proposed by Cortes [9] and by Tahbaz and Jadbabaie [34], in continuous-time and discrete-time frameworks, respectively. The popular discrete-time protocol in [34] was originally formulated for the min-consensus problem, which was later studied by Nejad et al [27] for the max-consensus problem in a max-plus algebraic setting, providing the characterization of the convergence rate. Its functioning, as most of other methods, consists in flooding the system with the biggest observed value, which may be outdated so far. Other approaches include soft-max estimators [34], [40], gossip based or randomized approaches [1], [21], nonlinear Perron-Frobenius theory [11], [13]. The max-consensus problem has been addressed in more general scenarios. Convergence results for time-varying networks with synchronous switching topologies have been investigated in [28]. Asynchronous updates and communications affected by time-delays have been considered in [18], while a stochastic framework for asynchronous updates has been proposed in [21]. The effect of noise in communications among the agents has been characterized in [26], [39]. Finally, in the context of open multi-agent systems where the size and composition of the network is time-varying, a gossip algorithm has been proposed in [1].

Instead, the literature has focused significantly on the dynamic average-consensus problem (also known as distributed average tracking problem); an insightful tutorial has been provided by Kia et al. in [23]. Spanos' et al. work was pioneering in the continuous-time framework, by considering the derivative of the reference signals to design the dynamics of the estimator [33], while the discrete-time case has been

This work was partially supported by Region Sardinia (RAS) with project MOSIMA, RASSR05871, FSC 2014-2020, Annualità 2017, Area Tematica 3, Linea d'Azione 3.1. and by the Fondazione Banco di Sardegna with the grant “Formal Methods and Technologies for the Future of Energy Systems”, cup F72F20000350007.

D. Deplano, M. Franceschelli and A. Giua are with DIEE, University of Cagliari, 09123 Cagliari, Italy. Emails: {diego.deplano,mauro.franceschelli,giua}@unica.it

D. Deplano gratefully acknowledges Sardinian Regional Government for the financial support of his PhD scholarship (P.O.R. Sardegna F.S.E. - Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2014-2020 - Axis III Education and training, Thematic goal 10, Investment Priority 10ii), Specific goal 10.5.

addressed by Zhu and Martinez some years later by considering discrete-time derivatives of the reference signals [41]; both algorithms require a specific initialization of the network and cannot handle noise in the communication or link failures. To overcome these limitations several approaches have been been proposed both in continuous-time [14], [17], [22] and in discrete-time [11], [16], [25].

The dynamic consensus problem naturally arises in the framework of *open multi-agent systems*, where the composition of the network and its size change over time. In this recent topic of research, interesting contributions can be found in [14], [15], [19], [20], [36] where the authors formulate consensus and dynamic consensus problems for networks of time-varying size. The works in [19], [20], [36] consider stochastic arrivals and departures of agents in the network while [14], [15] does not consider a model for agent arrivals and departures from the network.

Main contributions. This paper provides two protocols to solve the dynamic max-consensus problem:

– The first protocol is proved to achieve bounded steady-state and tracking error which can be traded-off for improved convergence time by tuning the protocol parameters;

– The second protocol is proved to achieve zero steady-state error and bounded tracking error, while it requires knowledge of an upper bound to the diameter of the graph representing the network to be executed;

– The bounds on the steady-state and tracking error for both protocols are theoretically characterized;

– The dual version of the proposed protocols to solve the dynamic min-consensus problem are derived;

– The proposed protocols are employed in the scenario of open multi-agent systems where agents can join and leave the network during the algorithm execution, solving for the first time the distributed size estimation problem for an anonymous network with time-varying size.

The latter problem was previously addressed by means of the popular max-consensus protocol for the time-invariant scenario by Varagnolo et al. [35]. We exploit our novel dynamic max-consensus protocols to enable the tracking of the size of a time-varying network, avoiding the need for network wide re-initialization of the distributed estimation procedure.

Structure of the paper. In Section II some preliminaries and the notation used in this paper are presented. In Section III the dynamic max/min-consensus problem is formalized along with the main working assumptions. In Section IV the proposed protocols are presented in detail and their performance is theoretically characterized. In Section V the dynamic max-consensus problem is applied to solve the distributed size estimation problem in open and anonymous multi-agent networks. In Section VI numerical simulations corroborating the theoretical analysis are provided. Finally, concluding remarks and future research directions are given in Section VII.

II. Notation and preliminaries

We denote by $\mathbb{R}$ and $\mathbb{N}$ the sets of real numbers and positive integer numbers, respectively.

Maximum and minimum of a vector $v = [v_1, \ldots, v_m]^\top$, with $m \in \mathbb{N}$, are denoted by

$$\bar{v} = \max_{i=1,\ldots,m} v_i, \qquad \underline{v} = \min_{i=1,\ldots,m} v_i. \tag{1}$$

A multi-agent system (MAS) consists of n agents modeled as dynamical systems interacting among each other. The undirected graph $G = (V, E)$ describes the pattern of bidirectional interactions among the agents; $V \subset \mathbb{N}$ is the set of nodes modeling the agents and $E \subseteq (V \times V)$ is the set of communication channels between them.

The state and the input of the i-th agent at time $k \in \mathbb{N}$ are denoted with $x_i(k) \in \mathbb{R}^m$ and $u_i(k) \in \mathbb{R}$, respectively. Agents i and j are said to be neighbors if there exists an edge between i and j, i.e., $(i, j) \in E$. A set of neighbors N_i is associated to the i-th agent and it is defined as $N_i = \{j \in V : (i, j) \in E\}$, which represents the agents in the graph sharing a point-to-point communication channel with agent i. For sake of simplicity we denote $N_i^+ = N_i \cup \{i\}$.

Communications among the agents are assumed to be bidirectional, and thus the graph G is considered as undirected. A path between two nodes i and j in a graph is a sequence of consecutive edges $\pi_{ij} = (i, k), (k, r), \ldots, (s, t), (t, j)$ where each consecutive edge shares a node with its predecessor. An undirected graph is said to be *connected* if there exists a path π_{ij} between any pair of nodes $i, j \in V$. The *diameter* of a connected graph, denoted as $\delta(G)$, is defined as the longest among the shortest paths among any pair of nodes $i, j \in V$. For all connected undirected graphs it holds $\delta(G) \le n - 1$ where n denotes the number of active agents in the network, which is defined as the cardinality of the set V, i.e., $n = |V|$. In this paper we also consider graphs with time-varying set of edges $E(k)$ and time varying set of nodes $V(k)$. We describe the interconnections at time $k \in \mathbb{N}$ among the $n(k)$ active agents with a time-varying graph $G(k) = (V(k), E(k))$.

III. Problem statement

Consider a network of n agents modeled as discrete-time dynamical systems with state $x_i = [x_i^1, \cdots, x_i^m] \in \mathbb{R}^m$, each of which has access to time-varying reference signal $u_i \in \mathbb{R}$ and interacts with other agents according to a graph $G = (V, E)$ and a local interaction protocol

$$x_i(k) = f_i(u_i(k), x_j(k-1) : j \in N_i^+). \tag{2}$$

The *dynamic max/min-consensus problem* consists in the design of a local interaction protocol $f_i(\cdot)$ which steers the agents to track the maximum $\bar{u}(k) \in \mathbb{R}$ or the minimum $\underline{u}(k) \in \mathbb{R}$ among the time-varying reference signals. In this work, we provide two different local interaction protocols (2) that solve for the first time in the current literature this problem.

Since the dynamic min-consensus problem can be reformulated as a dynamic max-consensus problem by replacing all the reference signals by the negative of their values, we present and characterize only the protocols for the maximum seeking, from which the dual protocols for the minimum seeking can be readily derived. For the sake of consistency, such reformulation and the derivation of the dual protocols are made explicit in Section IV-D.

TABLE I

CHARACTERIZATION OF THE DMC AND EDMC PROTOCOLS IN A CONNECTED GRAPH $\mathcal{G}$ WITH REFERENCE SIGNALS $\boldsymbol{u_i}$ UNDER ASSUMPTION 1. $\geq \mathbf{0}$: MAXIMUM ABSOLUTE VARIATION OF SIGNALS $\boldsymbol{u_i}$. $_G \in \mathrm{N}$: DIAMETER OF GRAPH $\mathcal{G}$. $>$ $\mathbf{0}$, $\boldsymbol{m} \in \mathrm{N} \setminus \{\mathbf{0}\}$: TUNING PARAMETERS.

Protocol	State	Tuning Parameters	Local interaction rule	Bound on the tracking error	Convergence time
DMC	$x_i \in \mathrm{R}$	$>$	$x_i(k) = \max_{j \, 2N_i} \{x_j(k-1) - ; u_i(k)\}$	$e(k) \leq (\ + \ ) \ _G;$	$T_c \leq \max \ _G; \frac{\overline{x}(k) - \overline{u}(k)}{-}$
EDMC	$x_i \in \mathrm{R}^m$	$m \geq \ _G$	$x_i^0(k) = u_i(k)$ $x_i^{`}(k) = \max_{j \, 2N_i} \left\{ x_j^{` \ 1}(k-1) \right\}; \quad ` \in [1; m]:$	$e(k) \leq m \ ;$	$T_c \leq m$

We characterize the performance of the proposed protocols in terms of convergence time as well as tracking error

$$e(k) = \max_{i2V} jx_i^m(k) \quad \overline{u}(k)j: \tag{3}$$

A. Working assumptions

The variation of each reference signal over the time window $[k \quad T; k]$, with $T \; 2 \; \mathrm{N}$ and $k \quad T$, is defined as

$$u_i(k; T) = u_i(k) \quad u_i(k \quad T); \quad 8i \; 2 \; V: \tag{4}$$

and, in a similar way, the variation of the maximum among the reference signals is defined as

$$\overline{u}(k; T) = \overline{u}(k) \quad \overline{u}(k \quad T): \tag{5}$$

Next, we state our first assumption concerning the boundedness of the reference signals' variation, which is a common assumption in the dynamic consensus literature.

Assumption 1. *The maximum absolute variation*[1] *of the reference signals in one step,* $T = 1$, *is bounded by a constant* 0, *i.e.,*

$$j \quad u_i(k; 1)j \qquad ; \quad 8i \; 2 \; V; \quad 8k \quad 0: \tag{6}$$

In the main application of this paper, we also deal with the scenario of an open multi-agent system (OMAS), wherein agents may leave or join and the communication pattern among them may change over time. Therefore a change in the network may involve a variation of the number of active agents and a variation of the active communication channels. These changes in the network are encoded into a time-varying graph $G(k) = (V(k); E(k))$ describing the interconnection among the $n(k)$ active agents. As soon as such a change occurs, the new agents' reference signals can be possibly much larger or smaller than those of the agents previously connected to the network. Thus, to address open networks we assume that the frequency at which the agents can join or leave the network is bounded as formalized next.

Assumption 2. *There exists a minimum dwell time* $2 \; \mathrm{N}$ *between two consecutive changes of the graph* $G(k)$.

[1]Note that if the reference signals are sampled versions of continuous-time signals, then by increasing the sampling frequency their variation in one iteration is reduced. Thus, for any signal with bounded variation there exists a sampling frequency such that Assumption 1 is also satisfied.

IV. PROPOSED DYNAMIC MAX/MIN-CONSENSUS PROTOCOLS

In this section we present and characterize two distributed protocols which solve the dynamic max-consensus problem. The first one, presented in Section IV-B and called *Dynamic Max-Consensus (DMC) Protocol*, achieves bounded steady-state and tracking error without requiring any further information about the network topology. The second one, presented in Section IV-C and called *Exact Dynamic Max-Consensus (EDMC) Protocol*, achieves zero steady-state error and bounded tracking error by requiring the knowledge of an upper bound to the network diameter. For the convenience of the reader, we anticipate in Table I the local interaction rules employed by the DMC and EDMC Protocols, along with the characterization of the convergence time and tracking error provided in the reminder of this section.

We first introduce the popular protocol solving the max-consensus problem over time-invariant signals and show its bias if applied in the case of time-varying signals, which is instrumental to understand the functioning of the protocols presented in the subsequent sections.

A. Bias of the max-consensus protocol with time-varying signals

The popular protocol which solves the (static) max-consensus problem [18], [26], [27], [34] makes use of the following local interaction rule,

$$x_i(k) = \max_{j2N_i} f x_j(k \quad 1)g; \tag{7}$$

with $x_i \; 2 \; \mathrm{R}$. This protocol enables the agents' states to converge to the maximum among the initial states. Thus, assuming a set of constant reference signals $u_i(k) = u_i(0) \; 2 \; \mathrm{R}$ for $k \; 2 \; \mathrm{N}$, the protocol enables to estimate their maximum by requiring the following initialization step

$$x_i(0) = u_i(0): \tag{8}$$

A naive generalization of this protocol to deal with time-varying reference signals could be

$$x_i(k) = \max_{j2N_i} f x_j(k \quad 1); u_i(k)g: \tag{9}$$

In the case of constant reference signals, the protocol in eq. (9) is equivalent to the one in eq. (7); in the case of time-varying signals, it is biased since it provides a monotonic estimation.

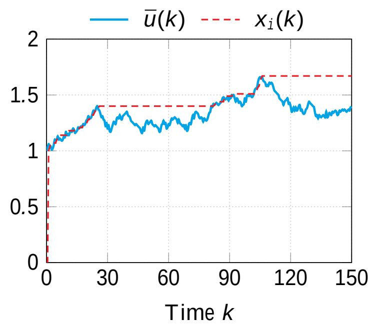

Fig. 1. Biased behavior of conventional max-consensus protocol in eq. (9) with time-varying reference signals.

This is major drawback that prevents the protocol in eq. (9) from tracking max value in the case it is non-monotonic.

As an example of this biased behavior, Figure 1 shows the evolution of a random network of agents executing the protocol in eq. (9) with time-varying reference signals and without any re-initialization logic: as one can notice, the tracking is lost every time the maximum among the reference signals decreases below the agents' states. Therefore, only a re-initialization of the protocol execution in the whole network can mitigate such an estimation bias for time-varying reference signals, which is a significant drawback for the implementation of a distributed algorithm in large-scale networks.

The above discussion is illustrative of the complexity of the problem under study. In the following, we provide two different strategies to modify the conventional protocol in order to overcome the issue of re-initializing the network and thus allowing the tracking of time-varying reference signals.

B. Dynamic Max-Consensus Protocol

Consider a network of agents with scalar state $x_i \in \mathbb{R}$. The strategy proposed in this section suggests to equip the generalized version of the max-consensus protocol, given in eq. (9) and discussed in the previous section, with an additive tuning parameter $\; > 0$, resulting in the following local interaction rule

$$x_i(k) = \max_{j \in N_i} \{x_j(k \quad 1) \quad ; u_i(k)\}: \qquad (10)$$

the DMC Protocol details the employment of this local interaction rule, while in the next theorem we provide its characterization.

Theorem 1. *Consider a MAS executing the DMC Protocol under Assumption 1 and consider a generic initial instant of time $k_0 \in \mathbb{N}$. If graph G is connected and if*

$$> \; ; \qquad (11)$$

then there exists a convergence time $T_c \quad 0$ such that the tracking error is bounded for $k \quad k_0 + T_c$ by

$$e(k) \qquad _G + \overline{\quad u}(k; \; _G) \quad (\; + \;) \; _G; \qquad (12)$$

where $_G$ is the diameter of graph G, while $e(\;)$ and $\overline{\quad u}(\;)$ are given in eq. (3) and (5), and it holds

$$T_c \quad \max \quad _G; \frac{\overline{x}(k_0) \quad \overline{u}(k_0)}{} \quad : \qquad (13)$$

DMC Protocol: Dynamic Max-Consensus

Input: Tuning parameter $\; > 0$.
Initialization: $x_i(0) \in \mathbb{R}$ for $i \in V$.
Init. for opt. conv. time: $x_i(0) = u_i(0)$ for $i \in V$.
Output: $x_i(k) \in \mathbb{R}$ for $i \in V$.
for $k = 1; 2; \ldots$ **each node** i **does**
Gather $x_j(k \quad 1)$ from each neighbor $j \in N_i$
Update the current state according to

$$x_i(k) = \max_{j \in N_i} \{x_j(k \quad 1) \quad ; u_i(k)\}$$

Proof: Given a generic time $k_0 \in \mathbb{N}$, we are going to prove that the maximum and minimum among the agents' state for $k \quad k_0 + \max\{\; _G; T'\}$ satisfy

$$\overline{x}(k) = \overline{u}(k); \qquad (14)$$

$$\underline{x}(k) \quad \overline{u}(k) + \overline{\quad u}(k; \; _G) \qquad _G; \qquad (15)$$

where T' is given by

$$T' = \frac{\max\{\overline{x}(k_0) \quad \overline{u}(k_0); 0\}}{}: \qquad (16)$$

Proof of eq. (14). Let $T \quad 1$, then it holds

$$\begin{aligned}\overline{x}(k_0 + T) &= \max_{i \in V} x_i(k_0 + T)\\ &= \max_{i \in V} \max_{j \in N_i} \{x_j(k_0 + T \quad 1) \quad ; u_i(k_0 + T)\}\\ &= \max_{i \in V} \{x_i(k_0 + T \quad 1) \quad ; u_i(k_0 + T)\}\\ &= \max\{\overline{x}(k_0 + T \quad 1) \quad ; \overline{u}(k_0 + T)\}\\ &= \max\{\overline{x}(k_0 + T \quad 2) \quad 2 \; ; \overline{u}(k_0 + T)\}\\ &= \quad \vdots\\ &= \max\{\overline{x}(k_0) \quad T \; ; \overline{u}(k_0 + T)\} \end{aligned} \qquad (17)$$

and since by Assumption 1 it holds $|\overline{\quad u}(k; T)| \quad T$, we further conclude that

$$\begin{aligned}\overline{x}(k_0 + T) &= \max\{\overline{x}(k_0) \quad T \; ; \overline{u}(k_0 + T)\}\\ &= \max\{\overline{x}(k_0) \quad T \; ; \overline{u}(k_0) + \overline{\quad u}(k; T)\}\\ & \quad \max\{\overline{x}(k_0) \quad T \; ; \overline{u}(k_0) \quad T \; \} \end{aligned} \qquad (18)$$

Due to the assumption in eq. (11), there exists a time T' such that the lower bound to the maximum state is smaller than the lower bound to the maximum input at time $k_0 + T'$, i.e.,

$$\overline{x}(k_0) \quad T' \; < \overline{u}(k_0) \quad T' \; :$$

The smallest value of T' which solves the above inequality corresponds to the time given eq. (16). From eq. (17), it follows that for $T \quad T'$ it holds

$$\overline{x}(k_0 + T) = \overline{u}(k_0 + T);$$

thus showing that eq. (14) holds for $k \quad k_0 + T'$.

Proof of eq. (15). We define the set

$$V_0 = \{i \in V : x_i(k_0) = \overline{x}(k_0)\}$$

denoting the set of agents whose state at time k_0 is the maximum among all others. We now consider the set of one-hop neighbors of nodes in set V_0 and denote it with V_1, which is formally defined next

$$V_1 = \{i \in V : (i,j) \in E,\ j \in V_0\}.$$

By induction on $\ell = 1, 2, \ldots$, we also define the sets

$$V_\ell = \left\{ i \in V : (i,j) \in E, j \in \bigcup_{s=0}^{\ell-1} V_s \right\},$$

for which it holds $V_\ell \subseteq V_{\ell+1}$. Now, the state of agents in V_1 at time $k_0 + 1$ can be lower bounded as follows

$$\begin{aligned} x_i(k_0+1) &= \max_{j \in N_i} \{x_j(k_0) - \alpha, u_i(k_0+1)\} \\ &= \max\{\overline{x}(k_0) - \alpha, u_i(k_0+1)\} \\ &\geq \overline{x}(k_0) - \alpha, \quad \forall i \in V_1 \end{aligned}$$

where the second step follows from the fact that each agent $i \in V_1$ has at least a neighbor $j \in V_0$ with state value at time k_0 is $x_j(k_0) = \overline{x}(k_0)$, by definition. For the sake of clarity, let us consider another step and compute a lower bound to the state of agents in V_2 at time $k_0 + 2$,

$$\begin{aligned} x_i(k_0+2) &= \max_{j \in N_i} \{x_j(k_0+1) - \alpha, u_i(k_0+2)\} \\ &\geq \max\{\overline{x}(k_0) - 2\alpha, u_i(k_0+2)\} \\ &\geq \overline{x}(k_0) - 2\alpha, \quad \forall i \in V_2. \end{aligned}$$

By induction, we conclude that for $\ell = 1, 2, \ldots$

$$x_i(k_0 + \ell) \geq \overline{x}(k_0) - \ell\alpha, \quad \forall i \in V_\ell.$$

The above lower bound holds for all agents $i \in V$ with $\ell \geq \delta_G$ since the longest shortest path between two nodes in a connected graph is at most equal to its diameter δ_G and thus $V_{\delta_G} = V$. We conclude that

$$x_i(k_0 + \delta_G) \geq \overline{x}(k_0) - \alpha\delta_G, \quad \forall i \in V.$$

Notice that the lower bound for $\ell = \delta_G$ is tight, in fact for $\ell = \delta_G + T$ with $T > 0$ one can verify that

$$\begin{aligned} x_i(k_0 + \delta_G + T) &\geq \overline{x}(k_0) - \alpha(\delta_G + 1) \\ &\geq \overline{x}(k_0 + T) + \pi T - \alpha(\delta_G + T) \\ &\geq \overline{x}(k_0 + T) - \alpha\delta_G, \quad \forall i \in V, \end{aligned}$$

since it always holds that $\overline{x}(k_0) \geq \overline{x}(k_0 + T) + \pi T$. We conclude that for all $k \geq k_0 + \delta_G$ it holds

$$x_i(k) \geq \overline{x}(k - \delta_G) - \alpha\delta_G, \quad \forall i \in V.$$

Finally, if also eq. (14) holds, then the next chain of inequalities holds for $k \geq k_0 + \max\{\delta_G, T^0\}$,

$$\begin{aligned} \underline{x}(k) &= \min_{i \in V} x_i(k) \\ &\geq \overline{x}(k - \delta_G) - \alpha\delta_G \\ &\geq \overline{u}(k - \delta_G) - \alpha\delta_G \\ &\geq \overline{u}(k) + \overline{\Delta u}(k; \delta_G) - \alpha\delta_G, \end{aligned}$$

thus showing that eq. (15) holds for $k \geq k_0 + \max\{\delta_G, T^0\}$.

The bounds in eq. (14) and eq. (15) are both satisfied for $k \geq k_0 + \max\{\delta_G, T^0\}$. This proves that the convergence time T_c is upper bounded as in eq. (13). Finally, the bound on the tracking error can be derived for $k \geq k_0 + T_c$ as follows

$$\begin{aligned} e(k) &= \max_{i \in V} |x_i(k) - \overline{u}(k)| \\ &= \max\{|\overline{x}(k) - \overline{u}(k)|, |\underline{x}(k) - \overline{u}(k)|\}. \end{aligned}$$

Exploiting the upper bound in eq. (14) and the lower bound in eq. (15) we derive

$$\begin{aligned} e(k) &= \max\{|\overline{x}(k) - \overline{u}(k)|, |\underline{x}(k) - \overline{u}(k)|\} \\ &= |\underline{x}(k) - \overline{u}(k)| \\ &\leq -\overline{\Delta u}(k; \delta_G) + \alpha\delta_G. \end{aligned}$$

Furthermore, by exploiting Assumption 1, it holds that $\overline{\Delta u}(k; \delta_G) \in [-\pi\delta_G, \pi\delta_G]$, and therefore

$$e(k) \leq (\pi + \alpha)\delta_G,$$

completing the proof. ■

In the next corollaries we make explicit the convergence time in the case of optimal initialization and the steady-state error in the case of constant reference signals.

Corollary 1. *The convergence time for Theorem 1 is minimized by the initialization* $x_i(k_0) = u_i(k_0)$, *which gives*

$$T_c \leq \delta_G. \tag{19}$$

Corollary 2. *The estimation error for Theorem 1, in the case all reference signals remain constant for an interval* $T^0 > T_c$, *satisfies the following condition*

$$e(k) \leq \alpha\delta_G, \quad \forall k \in [k_0 + T_c, k_0 + T^0]. \tag{20}$$

It is clear that the parameter α plays a fundamental role in the proposed protocol. The first main consideration concerns the fact that it is strictly positive, which allows one to avoid the usual centralized initialization of the max-consensus protocol. In fact, for $\alpha = 0$ the proposed protocol becomes the naive generalization discussed in Section IV-A, which has been shown to be biased and unsuitable to solve a tracking problem. Thus, for $\alpha = 0$ the results in Theorem 1 do not hold, even in the case of constant reference signals, i.e., $\pi = 0$: the requirement $\alpha > 0$ avoids the initialization in eq. (8).

A second important consideration is about the role that the parameter α plays in the trade-off between estimation error and convergence time. On one hand, to minimize the tracking error one needs to choose α as small as possible, according to the design condition in eq. (11). On the other hand, the choice of α also affects the convergence time T_c, with smaller values of α giving a larger convergence time. A pragmatic design criterion for α is to first fix the desired tracking error and then choose the largest α satisfying the error performance constraint, while minimizing the convergence time.

A last remark concerns the homogeneity in the choice of the parameter α among the agents. This restriction can be relaxed by allowing the agents to use different α_i still satisfying condition (11) without affecting the validity of the protocol. Following the same proof steps of Theorem 1, one can derive that if each agent selects its own α_i, the error remains bounded as in eq. (12), when α is replaced by $\overline{\alpha} = \max_{i \in V} \alpha_i$.

C. Exact Dynamic Max-Consensus Protocol

Consider a network of agents with vector state $x_i = [x_i^0, x_i^1, \dots, x_i^m]^\top \in \mathbb{R}^{m+1}$ where $m \in \mathbb{N}$ is an upper bound on the diameter of the underlying communication network, i.e., $m \geq \delta_G$. The strategy proposed in this section suggests to replicate the initialization step in eq. (8) of the conventional protocol at each instant of time in the first element x_i^0 of the state vector and then cascade the conventional protocol in eq. (7) over the remaining state variables: the estimate of each agent is the last state x_i^m. The proposed local interaction rule is formalized next

$$
\begin{aligned}
x_i^0(k) &= u_i(k) \\
x_i^\ell(k) &= \max_{j \in N_i} \left(x_j^{\ell-1}(k-1) \right), \quad \ell = 1, \dots, m.
\end{aligned}
\tag{21}
$$

The EDMC Protocol details the employment of this local interaction rule, while in the next theorem we provide its characterization.

Theorem 2. *Consider a MAS executing the EDMC Protocol under Assumption 1 and consider a generic initial instant of time $k_0 \in \mathbb{N}$. If graph G is connected and if*

$$m \geq \delta_G; \tag{22}$$

then there exists a convergence time $T_c \geq 0$ such that the tracking error is bounded for $k \geq k_0 + T_c$ by

$$e(k) \leq \overline{\Delta u}(k; m) \leq m\Pi; \tag{23}$$

where $e(\cdot)$ and $\overline{\Delta u}(\cdot)$ are given in (3) *and* (5)*, and it holds*

$$T_c \leq m. \tag{24}$$

Proof: At time k_0, we define the set

$$V_0 = \left\{ i \in V : x_i^0(k_0) = \max_{j \in V} x_j^0(k_0) \right\}.$$

Since by the EDMC Protocol it holds $x_i^0(k) = u_i(k)$, then

$$V_0 = \left\{ i \in V : x_i^0(k_0) = \bar{u}(k_0) \right\}.$$

Let us now consider time the set V_1 of one-hop neighbors of nodes in set V_0. Formally,

$$V_1 = \{ i \in V : (i,j) \in E,\ j \in V_0 \}.$$

The update rule (21) of the state x_i^1 for agents belonging to this set reduces to

$$x_i^1(k_0+1) = \bar{u}(k_0), \quad \forall i \in V_1$$

because all agents $i \in V_1$ have a neighbor $j \in V_0$ with state value $x_j^0(k_0) = \bar{u}(k_0 - 1)$. By induction, for $\ell \geq 1$ define

$$V_\ell = \left\{ i \in V : (i,j) \in E, j \in \bigcup_{s=0}^{\ell-1} V_s \right\};$$

and for all agents in these sets the update rule (10) of the state x_i^ℓ reduces to

$$x_i^\ell(k_0 + \ell) = \bar{u}(k_0).$$

By noticing that $V_m \supseteq V_{\delta_G} \equiv V$, we infer that for all $i \in V$ and for any time $k \geq k_0 + m$, it holds

$$x_i^m(k) = \bar{u}(k - m); \tag{25}$$

EDMC Protocol: Exact Dynamic Max-Consensus

Input: Network's diameter upper bound $m \in \mathbb{N}$.
Initialization: $x_i(0) \in \mathbb{R}^m$ for $i \in V$.
Output: $x_i^m(k) \in \mathbb{R}$ for $i \in V$.
for $k = 1, 2, \dots$ **each node** i **does**
  Gather $x_j(k-1)$ from each neighbor $j \in N_i$
  Update the current state according to
  $x_i^0(k) = u_i(k)$
  $x_i^\ell(k) = \max_{j \in N_i} \left(x_j^{\ell-1}(k-1) \right), \quad \ell = 1, \dots, m$

which proves that the convergence time T_c is at most equal to the upper bound m as in eq. (24). Furthermore, by Assumption 1 it holds $\bar{u}(k) = \bar{u}(k - m) + \overline{\Delta u}(k; m)$ and exploiting (25) we conclude that for any $k \geq m$ the bound on the tracking error given in (23) is correct since

$$e(k) = \max_{i \in V} |x_i^m(k) - \bar{u}(k)| \leq \overline{\Delta u}(k; m) \leq m\Pi;$$

where the last inequality is due to Assumption 1. ∎

In the next corollary we make explicit the steady-state error in the case of constant reference signals.

Corollary 3. *The estimation error for Theorem 2, in the case all reference signals remain constant for an interval $T' > T_c + \delta_G$, satisfies the following condition*

$$e(k) = 0, \quad \forall k \in [k_0 + T_c + \delta_G, k_0 + T']. \tag{26}$$

It is clear that the parameter m, which determines the dimension of the state vector, plays a fundamental role in the proposed protocol. Increasing the value of m involves a greater memory burden and communication complexity. However, since the precise value of the network diameter is usually unknown, it is not possible to make the best choice $m = \delta_G$. Therefore, m should be set to be equal to the sharpest available upperbound on the network diameter, ensuring in this way the functioning of the protocol and the minimization of the memory requirement.

D. Min-consensus Protocols

In the next remark we state the equivalence between the dynamic max-consensus problem and the dynamic min-consensus problem up to a change of variables.

Remark 1. *The dynamic min-consensus problem in a MAS where the agents have state $x_i(k)$ and have access to signals $u_i(k)$ is equivalent to a dynamic max-consensus problem in a MAS where the agents have state $y(k)$ and have access to signals $v_i(k)$ defined as follows*

$$y(k) = -x_i(k); \qquad v_i(k) = -u_i(k). \tag{27}$$

The remark can be understood by noticing that

$$v(k) = -u(k) \Rightarrow \bar{v}(k) = -\underline{u}(k);$$

from which it follows that the problem of steering the agents' state $x_i(k) = -y(k)$ to $\underline{u}(k) = -\bar{v}(k)$ is equivalent to the problem of steering the agents' state $y(k)$ to $\bar{v}(k)$.

DSE Protocol: Dynamic Size-Estimation

Input: Number of random numbers $p \in \mathbb{N}$.
Initialization: $u_{ij}(0) \leftarrow U(0,1)$ for $j = 1,\dots,p$
Output: $\hat{n}_i(k) = -\frac{p}{\sum_{j=1}^{p} \log(y_{ij}(k))}$ for $i \in V(k)$.
for $k = 1,2,\dots$ **each node** i **does**
  for $j = 1,\dots,p$ **do**
    Generate a random number if just joined
    if $i \in V(k) \setminus V(k-1)$ **then**
      $u_{ij}(k) \leftarrow U(0,1)$
    Execute either the DMC or the EDMC Protocol with reference signals $[u_{1j}(k), u_{2j}(k), \dots]$
    Store the estimated value in $y_{ij}(k)$

By means of the above discussion, one can derive the dual local interaction rules to solve the dynamic min-consensus problem. In particular, from the local interaction rule in eq. (10) of the DMC Protocol we derive

$$x_i(k) = \min_{j \in \mathcal{N}_i} \{x_j(k-1) + \delta, u_i(k)\}. \tag{28}$$

and, similarly, from the local interaction rule in eq. (21) of the EDMC Protocol we derive

$$\begin{aligned} x_i^0(k) &= u_i(k) \\ x_i^\ell(k) &= \min_{j \in \mathcal{N}_i} x_j^{\ell-1}(k-1), \quad \ell = 1,\dots,m. \end{aligned} \tag{29}$$

Their characterization can be readily borrowed from Theorem 1 and Theorem 2 by means of the equivalence given in Remark 1.

E. Robustness to re-initialization

A main feature of the protocols we have provided and characterized is that they don't need to be re-initialized if the network changes. This feature entails that if any change in the network occurs, such as an unexpected variation of the reference signals, a discontinuity of an agent's state due to a fault, or even a change of the topology or the number of active nodes, there is no need to restart the algorithm at each agent, as opposed to the popular max-consensus protocol.

This feature enables the employment of these protocols in open networks, as in the case of the application we discuss in the next section. When nodes can leave/join an open network in an arbitrary fashion, a large convergence time and tracking error may result. On the contrary, if the join/leave event can be locally controlled, such degradation of the performances can be avoided. For instance, if an agent joins or leaves under the following circumstances, the results given in Theorems 1-2 and Corollaries 1-2-3 still hold:

- An agent that joins the network with a reference signal lying in the convex hull of its neighbors' states and initializes its state to its reference signal;
- An agent that leaves the network with a reference signal lower than its own state.

V. Size Estimation of Open Networks

In this section we focus on the problem of estimating the size of an open network, i.e., the number of active nodes in it: we describe the interconnections at time $k \in \mathbb{N}$ among the $n(k)$ active agents with a time-varying graph $G(k)$.

We consider the framework of anonymous networks [38] wherein the agents cannot be identified within the network, thus guaranteeing security and privacy of the nodes but hindering their cooperation, and each node only knows its neighbors and has not information on the topology, or at most only a little information such as a bound on the network diameter. This problem counts an high number of interesting applications, e.g., maintenance purposes in ad-hoc wireless sensor networks [7], optimization of query access plans in internet-scale data networks [29], coordination of robotic agents [6], and so on.

A. Estimation methodology

In this section we describe and characterize our protocol for estimating the time-varying network's size. Our methodology extends the one proposed by Varagnolo et al. [35] to networks where the agents are free to join or leave at any time, thus resulting in a time-varying network's size $n(k)$ to be estimated by the agents. The implementation of the strategy is given in the DSE Protocol, while its characterization is given in the next section.

The methodology is based on statistical inference concepts and can be outlined in three main steps: generation, estimation and inference, described next.

1) (Generation) When a node i joins the network, it generates $p \in \mathbb{N} \setminus \{0\}$ independent random numbers[2] $u_{ij} \in [0,1]$ from a uniform distribution, i.e., $u_{ij} \leftarrow U(0,1)$ with $j = 1,\dots,p$;
2) (Estimation) The $n(k)$ active nodes execute either the DMC or the EDMC Protocol, thus each node i computes p estimates y_{ij} of the maximum value $\bar{u}_j$ among each local set $[u_{1j}, u_{2j}, \dots]$, with $j = 1,\dots,p$;
3) (Inference) Each node i infers the estimate $\hat{n}_i(k)$ of t he the network size $n(k)$ by maximum likelihood estimation from its own set of estimations $\mathbf{y}_i = [y_{i1},\dots,y_{ip}]$.

The likelihood function measures the fitness of a statistical model to a data sample (in our case, the maximum values $\bar{u}_j$), for given values of the unknown parameters (in our case the dimension of the network n). By noticing that the values $\bar{u}_j$ are the n-th order statistics of the sets $[u_{1j}, u_{2j}, \dots]$ for any $j = 1,\dots,p$ and by noticing that they are independent and identically distributed random variables forming the sample $\bar{\mathbf{u}} = \{\bar{u}_1,\dots,\bar{u}_p\}$, one can compute the likelihood function, cfr. [35, Section III],

$$L(n|\bar{\mathbf{u}}) = n^p \prod_{j=1}^{p} \bar{u}_j^{\,n-1}.$$

For further details about the derivation of the likelihood function we refer the reader to the proof of Theorem 3. The

[2] The number p of generations is a design parameter: the higher is the value of p, the better is the estimation at steady-state but the slower is the convergence rate.

value $\hat{n}$ maximizing the likelihood function is the maximum likelihood estimator of parameter n, that is given by

$$\hat{n} = -\frac{p}{\sum_{j=1}^{p} \ln(\bar{u}_j)}. \tag{30}$$

However, the i-th agent does not have the exact knowledge of the values $\bar{u}_j$, but it only knows its own estimates y_{ij}, therefore the best it can do is to use the values y_{ij} instead of $\bar{u}_j$, thus computing the estimate

$$\hat{n}_i = -\frac{p}{\sum_{j=1}^{p} \ln(y_{ij})}, \quad \forall i \in V, \tag{31}$$

which, in fact, is the output of the DSE Protocol.

B. Characterization of the DSE Protocol

Whenever an agent leaves or joins the network, the set of reference signals changes, and so do their maximum values. The DMC Protocol and the EDMC Protocol guarantee the tracking of such time-varying signal thanks to their robustness to the initial condition discussed in Section IV-E. Intuitively, the rate at which the agents leave or join the network is correlated to the variation of the maximum values to be estimated and thus some critical scenarios may happen.

Here, we just make the assumption that our protocols can run a sufficiently high number of iterations such that a stady state is reached after each change of the network: the minimum dwell time τ between two changes of the network ensured by Assumption 2 is required to be greater or equal than the convergence time T_c of the employed protocol. In the next theorem, we provide a characterization of the estimation error in the above described scenario with a sufficiently large dwell-time, even though we remark that the proposed protocol tracks the correct value, with possibly larger worst case error, when the assumption on the dwell time is not satisfied.

Theorem 3. *Consider an OMAS executing the DSE Protocol under Assumption 2 and consider a generic initial instant of time $k_0 \in \mathbb{N}$ at which the network changes, i.e., $\mathcal{G}(k_0-1) \neq \mathcal{G}(k_0)$. Assume that the dwell time is greater than the convergence time of the employed protocol[3], i.e., $\tau \geq T_c$.*

For the DMC Protocol, the expected value of the estimations $\hat{n}_i(k)$ for $k \in [k_0+T_c, k_0+\tau]$ is given by

$$\mathrm{E}[\hat{n}_i(k)] = \epsilon^{p-1} e^{\epsilon np} (\epsilon p)^p \, \Gamma(1-p, \epsilon np), \tag{32}$$

where $\epsilon = \delta_G$ is the upperbound to the tracking error, and $\Gamma(\cdot)$ denotes the upper incomplete gamma function[4].

For the EDMC Protocol, the expected value of the estimations $\hat{n}_i(k)$ for $k \in [k_0+T_c, k_0+\tau]$ is given by

$$\mathrm{E}[\hat{n}_i(k)] = \frac{np}{p-1}. \tag{33}$$

Proof: By Assumption 2, the network remains unchanged for $k \in [k_0, k_0+\tau]$. Thus, by the DSE Protocol the reference signals and their maximum are constant in this interval of time and Corollaries 2-3 hold. In the remaining of the proof we consider the steady-state for $k \in [k_0+T_c, k_0+\tau]$ and omit the time dependence (k).

Consider the samples of numbers $u_{1j}, \ldots, u_{nj}$ for any $j = 1, \ldots, p$. Each of these numbers is randomly generated with probability distribution function $P(a) = a$ for $a \in [0, 1]$ and $P(a) = 0$ otherwise. The maximum value of the sample

$$\bar{u}_j = \max_{j \in V} u_{ij}, \quad \forall j = 1, \ldots, p.$$

is the the n-th order statistics of the sample. Consider now the sample obtained by the n-th order statistics of each random number generated by the agents, i.e.,

$$\bar{\mathbf{u}} = \{\bar{u}_1, \ldots, \bar{u}_p\}.$$

All variables in the sample are i.i.d. random variables with probability density function $p_n(a) = nP^{n-1}(a)$ depending on the parameter n. Thus, the likelihood function $L(n|\bar{\mathbf{u}})$ can be computed as the product of the probability density functions,

$$L(n|\bar{\mathbf{u}}) = \prod_{j=1}^{p} p_n(\bar{u}_j) = n^p \prod_{j=1}^{p} \bar{u}_j^{\,n-1}.$$

In practice, it is often convenient to work with the natural logarithm of the likelihood function, called the log-likelihood

$$\begin{aligned} \mathcal{L}(n|\bar{\mathbf{u}}) &= \ln(L(n|\bar{\mathbf{u}})) = \ln\left(n^p \prod_{j=1}^{p} \bar{u}_j^{\,n-1}\right) \\ &= p\ln(n) + (n-1)\sum_{j=1}^{p} \ln \bar{u}_j. \end{aligned}$$

By computing the value $\hat{n}$ maximizing the log-likelihood function one obtains the best estimate of the size n of the network, which is given by eq. (30). However, variables $\bar{u}_j$ are not known exactly at each node, and instead they know their estimate y_{ij}. Therefore, the best estimation $\hat{n}_i$ an agent can do is to implement eq. (31). It is necessary to understand how the error arising from the use of $y_{ij} \neq \bar{u}_j$ affects the estimation of $\hat{n}$. We start our discussion taking into consideration the employment of the DMC Protocol for which a non-null error is reached at steady-state. Then, as a special case for zero error, we derive the result for the EDMC Protocol.

Discussion for the DMC Protocol. By Corollary 2 the steady-state error for the DMC Protocol in the estimating of $\bar{u}_j$ is bounded by the following

$$e_j = \max_{i \in V} |y_{ij} - \bar{u}_j| \leq \delta_G = \epsilon. \tag{34}$$

A fundamental consideration, resulting from the constructing proof of Theorem 1 and Corollary 2, is that at steady-state the estimation y_{ij} of agent i of the quantity $\bar{u}_j$ is always an underestimation, i.e., $y_{ij} \leq \bar{u}_j$ for all $i \in V$. With this consideration in mind, it is easy to realize that the worst case is when at least one agent underestimates all variables $\bar{u}_j$ with maximum error $\epsilon = \delta_G$. Thus, we consider such a worst case scenario by assuming that

$$\exists i \in V: \quad y_{ij} = \bar{u}_j - \epsilon, \quad \forall j = 1, \ldots, p. \tag{35}$$

[3] The dwell time expressed in absolute time units can be arbitrarily reduced by simply increasing the frequency of the algorithm iterations.

[4] The upper incomplete gamma function $\Gamma(a, x)$ is defined as follows $\Gamma(a, x) = \int_x^{\infty} t^{a-1} e^{-t} dt$. There does not exist a closed form of this function, but it is usually implemented in programming platforms. For example, with MATLAB it can be computed with the command `igamma(a,x)`. For further details we refer the reader to [4].

Under condition (35) we obtain a lower bound $\hat{n}$ on the estimation $\hat{n}_i$ of each agent, which is obtained as follows

$$\begin{aligned}\hat{n}_i &= \frac{-p}{\sum_{j=1}^{p} \ln\left(\overline{u}_j - \varepsilon\right)} = \frac{-p}{\sum_{j=1}^{p}\left[\ln \overline{u}_j + \ln\left(1-\frac{\varepsilon}{\overline{u}_j}\right)\right]} \\ &\geq \frac{-p}{\sum_{j=1}^{p}\left[\ln \overline{u}_j + \ln(1-\varepsilon)\right]} \geq \frac{-p}{\sum_{j=1}^{p}\left(\ln \overline{u}_j - \varepsilon\right)} \\ &\geq \frac{p}{\sum_{j=1}^{p}(-\ln \overline{u}_j) + p\varepsilon} \geq \frac{1}{\frac{1}{p}\sum_{j=1}^{p}(-\ln \overline{u}_j) + \varepsilon} = \hat{n} \qquad (36)\end{aligned}$$

At the denominator of (36) we can recognize the term

$$\gamma = -\frac{1}{p}\sum_{j=1}^{p} \ln \overline{u}_j. \qquad (37)$$

Now, consider the following conceptual steps:

1) The variables $\overline{u}_j$ are beta random variables with shape parameters equal to $(n,1)$, since they are the n-th order statistics of a sample of n random numbers drawn from a continuous distribution;
2) The variables $-\ln \overline{u}_j$ are exponential random variables with rate n due to the equivalence to the beta distribution with parameters $(n,1)$;
3) The variable γ as in eq. (37) is a gamma random variable with shape p and rate pn since they are the averaged sum of exponential functions.

Therefore, by means of the *law of the unconscious statistician*, we can calculate the expected value of $\hat{n}$ in eq. (36) as follows

$$\mathrm{E}[\hat{n}] = \int_0^{1} f(x)g(x)dx; \qquad (38)$$

where $f(x) = 1/(x+\varepsilon)$ is the relation between $\hat{n}$ and the gamma variable γ, while $g(x)$ is the probability density function of the gamma variable γ, i.e.,

$$g(x) = \frac{(np)^p}{(p-1)!}x^{p-1}e^{-npx}$$

Solution to (38) can be computed through several solver (we have used Wolfram|Alpha Pro engine) and it is giving by

$$\mathrm{E}[\hat{n}] = \varepsilon^{p-1}e^{\varepsilon np}(np)^p\,\Gamma(1-p, \varepsilon np);$$

where $\Gamma(p, x)$ is known as the upper incomplete gamma function. We point out that this expression holds for $n, p \in \mathbb{N}$ and $\varepsilon \in \mathbb{R}$ such that $n \geq 1$, $p > 1$ and $\varepsilon \geq 0$.

Discussion for the EDMC Protocol. By Corollary 3 the steady-state error in the estimating of $\overline{u}_j$ is null. Solution to (38) for $\varepsilon = 0$ is given by the following

$$\mathrm{E}[\hat{n}] = \frac{np}{p-1}.$$

We point out that this expression holds for $n, p \in \mathbb{N}$ and $\varepsilon \in \mathbb{R}$ such that $n \geq 1$, $p > 1$ and $\varepsilon \geq 0$. ■

VI. NUMERICAL SIMULATIONS

To illustrate the performance of the proposed protocols, simulation results are given in this section. First, we substantiate the results for the DMC and the EDMC Protocols about their stability and error bounds by simulating a worst-case scenario network with line topology. Second, we simulate the DMC Protocol tracking a sinusoidal reference signal for different choices of the design parameters, showing convergence time and tracking error can be traded-off. Third, we apply these protocols in the context of distributed size estimation of open networks considering the case of scale-free networks with approximately fixed diameter.

A. Example 1: comparison of the DMC and the EDMC Protocols

We simulate a network of $n = 10$ agents with line topology. The choice of the line topology is instrumental to run simulations in the worst case scenario. In fact, for line graphs the information takes exactly $\delta_G = n - 1 = 9$ steps to flow through the network, thus maximizing the error for a fixed number of agents.

Figures 2-3 show the evolution of the output variables (dashed red lines) and of the maximum among the time-varying reference signals (solid blue line) when the DMC Protocol or the EDMC Protocol are run over the MAS, respectively. The agents are uniformly initialized in the interval $[0, 1.5]$ and the reference signals are set to be equal to -1 for all nodes but the 6-th one, which is initialized at $u_6(0) = 0.5$ to be the maximum. All reference signals remain constant except for the 6-th component, the maximum, which is time-varying with respect to eq. (39), given in the next page, with initial condition $u_6(0) = 0$ and $\theta = 0.02$ being the absolute variation according to Assumption 1,

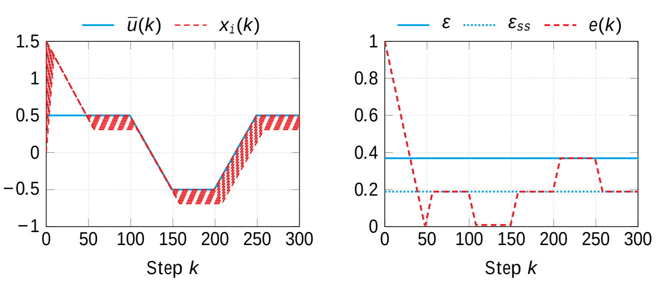

Fig. 2. *Example 1*: Evolution of a MAS, with $n = 10$ agents in a line configuration with diameter $\delta_G = 9$, running the DMC Protocol. The signal variation is bounded by $\theta = 0.02$ and the protocol is designed with $\alpha = 0.021$. ε denotes the bound on the tracking error as in eq. (12) and ε_{ss} denotes the bound on the steady-state error as in eq. (20).

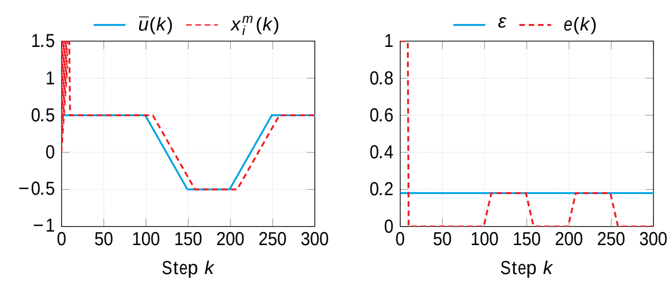

Fig. 3. *Example 1*: Evolution of a MAS, with $n = 10$ agents in a line configuration with diameter $\delta_G = 9$, running the EDMC Protocol. The signal variation is bounded by $\theta = 0.02$ and the protocol is designed with $\alpha = 0.021$. ε denotes the bound on the tracking error as in eq. (23) and the steady-state error is zero according to eq. (26)

TABLE II
Example 1: DESIGN OF THE DMC AND THE EDMC PROTOCOLS AND THEIR CHARACTERIZATION DUE TO THEOREMS 1-2.

Protocol	Input	Conv. Time	Bound on the errors (tracking)	(steady-state)
DMC	$= 0.021$	$T_c = 32$	$\varepsilon = 0.37$	$\varepsilon_{ss} = 0.19$
EDMC	$m = 9$	$T_c = 9$	$\varepsilon = 0.18$	$\varepsilon_{ss} = 0$

$$u_6(k+1) = \begin{cases} u_6(k) & \text{if } k < 100 \\ u_6(k) & \text{if } k \in [100, 150) \\ u_6(k) & \text{if } k \in [150, 200) \\ u_6(k) + & \text{if } k \in [200, 250) \\ u_6(k) & \text{if } k \quad 250 \end{cases} \qquad (39)$$

The design for the DMC Protocol and its characterization provided by Theorem 1 are given in Table II. These simulations show how the protocols steer the agents to track the time-varying maximum value $\bar{u}(k)$ among the reference signals, corroborating the convergence times and the bound on the errors given in Theorems 1-2 and Corollaries 2-3.

B. Example 2: design trade-off for the DMC Protocol

As a second simulation we consider the same network and initialization of Section VI-A. The time-varying reference signal $u_6(k)$ is a sinusoidal signal given by

$$u_6(k) = u(k \quad 1) + 0.2 \sin \frac{k}{10} ; \qquad k \quad 1;$$

with initial condition $u_6(0) = 0$, which is also the maximum signal to be tracked since all other reference signals stay constant at 1. Notice that for this signal the variation is bounded by $= 1 \;.99$.

Fig. 4 shows the evolution of the output variables (dashed red lines) and the maximum time-varying reference signal (solid blue line). In particular, in the above plots the DMC Protocol has been designed with $= 0.2$, while in the bottom plots with $= 0.5$. We notice that the design of $= 0.2$ provides a greater convergence time $T_c = 70$ and a smaller tracking error $e(k) \quad 0.36$ compared to the design $= 0.5$ which gives a convergence time $T_c = 35$ and a tracking error $e(k) \quad 0.63$. Thus, the design of provides a trade-off between convergence time and tracking error.

C. Example 3: dynamic estimation in large networks

In this third simulation, we are going to increase the number of agents in the network up to $n = 10^4$ while maintaining the diameter fixed to $_G = 9$, as in the previous two examples. In Fig 5 we report the simulations of a large network executing the DMC Protocol (above) and the EDMC Protocol (below) where the reference signals randomly vary with a variation bounded by $= 0 \;.02$ and $= 0.021$.

This simulation is instrumental to emphasize two main strengths of the proposed protocols: 1) The bound on the tracking error of both protocols does not increase proportionally to the dimension of the network: in fact, it can be verified that these bounds correspond to the one computed in the Example 1 for the case of a small line network with $n = 10$ agents; 2) The memory capability required by each agent in the network does not increase proportionally to the dimension of the network. In particular, the memory burden for each agent for the DMC Protocol remains unaffected by the increase of the network size, while for the EDMC Protocol the memory burden only increases with the diameter of the network.

Fig. 4. Example 2: Different evolution of a MAS, with $n = 10$ agents in a line configuration with diameter $_G = 9$, running the DMC Protocol with different designs. While the signal variation is bounded by $= 0.0199$, the top figures show the dynamic tracking with $= 0.02$ and the bottom figures show the dynamic tracking with $= 0.05$.

Fig. 5. Example 3: Different evolution of a large MAS, with $n = 10^4$ agents in a random configuration with diameter $_G = 9$. The signal variation is bounded by $= 0 \;.02$. The top figures show the evolution when DMC Protocol is employed, while the bottom figures show the evolution when the EDMC Protocol is employed.

D. Example 4: dynamic size estimation

We choose to run simulations of the DSE Protocol over scale-free networks [2], [3]. Such networks are known to be ultrasmall [8], meaning that their diameter scales very slow with the dimension of the network, behaving as $d \approx \ln(\ln(N))$.

We randomly generate a scale-free network by means of Barabási–Albert (BA) model [2], which iteratively constructs a random scale-free networks using a preferential attachment mechanism given an initial small network, not necessarily scale-free. We use as initial network a line network of 5 nodes, and then we run the algorithm until a network of $n = 100$ nodes is generated whose diameter is of the order of the original small network, i.e., $d \approx 5$. In order to simulate nodes leaving and joining the network without losing the connectivity and the scale-free structure of the graph, we randomly deactivate or activate some of the last 25 nodes added to the network by the algorithm every $\tau = 5 \cdot 10^2$ steps. This choice does not have any impact on the simulation results: every nodes who leaves the network does not keep any information from the last active period. In fact when a node rejoins the network, it generates a new random number and initializes its state to this number. We remark that the log in and log out actions occur without any coordination among the agents, thus dealing with fault conditions such as a node which suddenly stops working.

Fig. 6 shows the estimation of the size of a network by means of the DSE Protocol which makes use of either the DMC Protocol or the EDMC Protocol. For the sake of clarity, we have decided to plot the worst estimation among the agents. We also recall that this constitutes a generalization of the method proposed in [35] to open networks.

Fig. 6. Example 4: Dynamic size estimation of an open and time-varying network by means of the DSE Protocol.

VII. Conclusions

In this paper we provided two distributed protocols, namely the DMC and the EDMC Protocols, enabling the agents of an anonymous network to track the time-varying maximum value of a set of reference signals given as inputs to the agents. The difficulty of the problem, which so far was an open problem in the current literature, arises from the necessity to allow the estimates to decrease, whereas most methods consist in flooding techniques which saturates the estimations with the biggest observed value, which may be outdated by now. Dual protocols are derived to solve the minimum seeking problem.

A main feature of the proposed protocols is their robustness to re-initialization, which allows their extension to open networks, where the agents are allowed to log in and out of the network. This motivated our our main application, i.e., the dynamic tracking of the number of active agents in open networks, a problem which goes by the name of cardinality or size estimation. By means of the dynamic max-consensus protocols, we extended the strategy proposed in [35] and solved for the first time a distributed size estimation problem for an anonymous and time-varying networks.

Future work will further investigate the use of the DMC and the EDMC Protocols to solve other distributed estimation problems in networks which change in both topology and size and possible application of the algorithms in the context of distributed optimization.

References

[1] M. Abdelrahim, J. Hendrickx, and W. Heemels, "Max-consensus in open multi-agent systems with gossip interactions," in *IEEE 56th Conference on Decision and Control (CDC)*, 2017, pp. 4753–4758.

[2] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, 2002.

[3] L. Amaral, A. Scala, M. Barthélémy, and H. Stanley, "Classes of small-world networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 21, pp. 11 149–11 152, 2000.

[4] G. B. Arfken, H. J. Weber, and F. E. Harris, "Gamma function," in *Mathematical Methods for Physicists*, 7th Edition, Cambridge, MA, USA: Academic Press, 2012, ch. 13, pp. 599–641.

[5] T. Borsche and S. A. Attia, "On leader election in multi-agent control systems," in *22th Chinese Control and Decision Conference (CCDC)*, 2010, pp. 102–107.

[6] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton, NJ, USA: Princeton University Press, 2009.

[7] J. Cichoń, J. Lemiesz, and M. Zawada, "On cardinality estimation protocols for wireless sensor networks," in *10th International Conference on Ad-Hoc Networks and Wireless*, 2011, pp. 322–331.

[8] R. Cohen and S. Havlin, "Scale-free networks are ultrasmall," *Physical review letters*, vol. 90, no. 5, pp. 058 701–1/4, 2003.

[9] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.

[10] Z. Dengchang, A. Zhulin, and X. Yongjun, "Time synchronization in wireless sensor networks using max and average consensus protocol," *International Journal of Distributed Sensor Networks*, vol. 9, no. 3, pp. 192 128–1/10, 2013.

[11] D. Deplano, M. Franceschelli, and A. Giua, "A nonlinear perron–frobenius approach for stability and consensus of discrete-time multi-agent systems," *Automatica*, vol. 118, p. 109 025, 2020.

[12] ——, "Distributed tracking of graph parameters in time-varying anonymous networks," in *IEEE Conference on Decision and Control (CDC)*, 2021.

[13] ——, "Lyapunov-free analysis for consensus of nonlinear discrete-time multi-agent systems," in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 2525–2530.

[14] M. Franceschelli and P. Frasca, "Proportional dynamic consensus in open multi-agent systems," in *IEEE 57th Conference on Decision and Control (CDC)*, 2018, pp. 900–905.

[15] ——, "Stability of open multi-agent systems and applications to dynamic consensus," *IEEE Transactions on Automatic Control (Early Access)*, pp. 1–1, 2020.

[16] M. Franceschelli and A. Gasparri, "Multi-stage discrete time and randomized dynamic average consensus," *Automatica*, vol. 99, pp. 69–81, 2019.

[17] J. George and R. Freeman, "Robust dynamic average consensus algorithms," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4615–4622, 2019.

[18] S. Giannini, A. Petitti, D. Di Paola, and A. Rizzo, "Asynchronous max-consensus protocol with time delays: Convergence results and applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 2, pp. 256–264, 2016.