An Asynchronous Consensus-Based Algorithm for Estimation From Noisy Relative Measurements

Andrea Carron, Marco Todescato, Ruggero Carli, and Luca Schenato, *Member, IEEE*

***Abstract*—In this paper, we address the problem of optimal estimating the position of each agent in a network from relative noisy vectorial distances with its neighbors by means of only local communication and bounded complexity, independent of network size and topology. We propose a consensus-based algorithm with the use of local memory variables which allows asynchronous implementation, has guaranteed exponential convergence to the optimal solution under simple deterministic and randomized communication protocols, and requires minimal packet transmission. In the randomized scenario, we then study the rate of convergence in expectation of the estimation error and we argue that it can be used to obtain upper and lower bound for the rate of converge in mean square. In particular, we show that for regular graphs, such as Cayley, Ramanujan, and complete graphs, the convergence rate in expectation has the same asymptotic degradation of memoryless asynchronous consensus algorithms in terms of network size. In addition, we show that the asynchronous implementation is also robust to delays and communication failures. We finally complement the analytical results with some numerical simulations, comparing the proposed strategy with other algorithms which have been recently proposed in the literature.**

***Index Terms*—Wireless-sensor networks (WSNs), distributed localization algorithms, consensus algorithms.**

I. Introduction

THE proliferation of relatively inexpensive devices capable of communicating, computing, sensing, interacting with the environment, and storing information is promising an unprecedented number of novel applications throughout the cooperation of these devices toward a common goal. These applications include swarm robotics, wireless-sensor networks (WSNs), smart energy grids, smart traffic networks, and smart camera networks. These applications also pose new challenges, of which *scalability* is one of the major ones. Scalability is intended as the ability for an application to continue functioning without any dramatic performance degradation even if the number of devices involved keeps increasing. In particular, an application is scalable if it is not necessary to increase hardware resources nor to adopt more complex software algorithms in each device even if the total number of devices increases.

Manuscript received September 20, 2013; revised December 20, 2013; accepted April 9, 2014. Date of publication August 28, 2014; date of current version September 16, 2014. This work was supported by the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement 257462 HYCON2 Network of Excellence. Recommended by Associate Editor F. Fagnani.

The authors are with the Department of Information Engineering, University of Padova, Padova 35131, Italy (e-mail: carronan@dei.unipd.it; todescat@dei.unipd.it; carlirug@dei.unipd.it; schenato@dei.unipd.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCNS.2014.2353512

In this paper, we address the problem of designing algorithms that are capable of reconstructing the optimal estimate of the location of a device based on noisy relative measurements with respect to its neighbors in a connected network. In particular, we want to design distributed algorithms that allow each device to reconstruct its own position only from exchanging information with its neighbors, regardless of the size of the network. Moreover, these algorithms must be scalable, that is, their computational complexity, bandwidth, and memory requirements should be independent of the network size. Finally, the estimate provided should asymptotically converge to the solution of a centralized optimization problem.

Distributed optimization has being attracting ever-growing attention in the past years since many problems in the large-scale network have been cast as convex optimization problems. In particular, the problem at hand in this paper can be cast as the following unconstrained optimization problem:

$$\min_{x_1,\dots,x_N} \sum_{(i,j)\in\mathcal{E}}^{|\mathcal{E}|} f_{ij}(x_i - x_j) \tag{1}$$

where $x_i \in \mathbb{R}^\ell$, $\mathcal{E}$ represents all of the pairs of nodes for which relative measurements are available and f_{ij} are convex functions. Many problems can be written in this framework, such as sensor localization [1], [2], sensor calibration [3], clock synchronization [4], and camera localization [5], [6]. For example, in the context of localization from vectorial relative distances in a plane, the cost functions f_{ij} are given by

$$f_{ij}(x_i - x_j) = \|x_i - x_j - z_{ij}\|^2$$

where $z_{ij} \in \mathbb{R}^\ell$ is the noisy measurement of the relative (vector) distance of node i from node j. As a consequence, the optimization problem in (1) becomes a distributed least-square problem. Several scalable distributed solutions to this problem are already available in the literature. In [1] and [2], the authors propose a distributed Jacobi solution based on a synchronous implementation, which was later extended to account for asynchronous communication and packet losses [7]. The same approach has been independently proposed in [8] in the context of distributed time synchronization in WSNs. Differently, in [3], a broadcast consensus-based algorithm, which is suitable for asynchronous implementation, is proposed but the local estimates do not converge and exhibit oscillatory behavior around the optimal value. A similar approach has been proposed in [9] and [10], where the local ergodic average of the gossip asynchronous algorithm is proved to converge to

the optimal value as $1/k$, where k is the number of iterations. An alternative approach based on the Kaczmarz method for the solution of general linear systems has been suggested in [11]; however, a practical asynchronous implementation for distributed localization from relative measurements, which satisfies the specific edge and node activation probabilities dictated by the algorithm, is not given; moreover, no robustness analysis in terms of delays is provided.

The main contribution of this paper is to propose a novel asynchronous algorithm whose main idea consists of casting the estimation problem as a consensus problem under some suitable changes of coordinates, and then to add some extra memory variables at each node to keep track of the estimated location of its neighbors, that is, the nodes from which they collected the relative distance measurements. Estimates of these local variables eventually converge to the estimates of the neighbors, thus guaranteeing the convergence of the entire algorithm, at the price of some delay. This strategy has several relevant advantages, namely: 1) it is scalable; 2) it has proven the exponential rate of convergence under mild assumptions; 3) is robust to packet losses and delays; and 4) requires the transmission of a single communication packet per iteration. This last feature is particularly relevant for WSN applications since agents have a limited energy budget and communication is more expensive than computation from an energy standpoint. We also study the performance of the proposed algorithm in terms of the convergence rate. This task is particularly challenging since the proposed algorithm turns out to be a higher order consensus algorithm, for which little analytic tools are available. In fact, the few works available in the literature, which address the rate of convergence of randomized higher order consensus algorithms, are limited to the convergence in expectation [12]. In this paper, we exactly compute the rate of convergence in expectation of our algorithm for regular graphs, and through extensive numerical simulations, we conjecture that it also provides an upper bound for the rate of convergence in mean square. Moreover, we show that, asymptotically, for many types of regular graphs, such as Cayley, Ramanujan, and complete graphs, such a rate of convergence in expectation is reduced by a factor N, where N is the number of nodes, which is the same of standard memoryless asynchronous consensus algorithms, thus implying that asymptotically in N, the reduction of rate of convergence due to memory is negligible.

We also prove the convergence of the proposed algorithm when bounded delays and packet losses are present, thus making it particularly suitable for applications using lossy wireless communication. We finally complement the theoretical results with some numerical simulations which show that the proposed algorithm has performance in terms of the rate of convergence per iteration which is slightly slower than the fastest algorithms available in the literature. However, it greatly outperforms them if the rate of convergence is computed in terms of the number of exchanged messages, that is, the estimation error obtained by sending a fixed number of packets is much lower for our proposed algorithm than the other algorithms available in the literature.

This paper is organized as follows. In Section II, we formulate the problem. In Section III, we introduce the synchronous consensus-based algorithm (denoted as s-CL). In Section IV, we propose a more realistic asynchronous implementation of the s-CL algorithm (denoted as a-CL). In Section V, we establish the convergence of the a-CL algorithm and we provide some bounds on the rate of convergence in mean square. In Section VI, we show that the a-CL algorithm is robust to delays and communication failures. In Section VII, we provide some numerical results comparing the a-CL algorithm to other strategies recently proposed in the literature. Finally, in Section VIII, we gather our conclusions.

A. Mathematical Preliminaries

Before proceeding, we collect some useful definitions and notations. In this paper, $\mathcal{G} = (V, \mathcal{E})$ denotes a directed graph, where $V = \{1, \ldots, N\}$ is the set of vertices and $\mathcal{E}$ is the set of directed edges, that is, a subset of $V \times V$. More precisely, the edge (i, j) is incident on node i and node j and is assumed to be directed away from i and directed toward j. The graph $\mathcal{G}$ is said to be bidirected if $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$.

Given a directed graph $\mathcal{G} = (V, \mathcal{E})$, a directed path in $\mathcal{G}$ consists of a sequence of vertices $(i_1, i_2, \ldots, i_r)$ such that $(i_{j+1}) \in \mathcal{E}$ for every $j \in \{1, \ldots, r-1\}$. An undirected path in $\mathcal{G}$ consists of a sequence of vertices $(i_1, i_2, \ldots, i_r)$ such that either $(i_j, i_{j+1}) \in \mathcal{E}$ or $(i_{j+1}, i_j) \in \mathcal{E}$ for every $j \in \{1, \ldots, r-1\}$.[1] The directed graph $\mathcal{G}$ is said to be *strongly connected* (respectively, *weakly connected*) if for any pair of vertices (i, j) a directed path (respectively, an undirected path) exists, connecting i to j. Given the directed graph $\mathcal{G}$, the set of neighbors of node i, denoted by $\mathcal{N}_i$, is given by $\mathcal{N}_i = \{j \in V \mid (i, j) \in \mathcal{E}\}$. A directed graph is said to be *regular* if all of the nodes have the same number of neighbors.

Given a directed graph $\mathcal{G} = (V, \mathcal{E})$ with $|\mathcal{E}| = M$, let the *incidence matrix* $A \in \mathbb{R}^{M \times N}$ of $\mathcal{G}$ be defined as $A = [a_{ei}]$, where $a_{ei} = 1$, -1, 0, if edge e is incident on node i and directed away from it, is incident on node i and directed toward it, or is not incident on node i, respectively.

Let $\mathbf{1}_N$ be the N-dimensional column vector with all components equal to one. If there is no risk of confusion, we will drop the subscript N. Given a matrix B, we denote with $B^\dagger$ its pseudoinverse. Given a vector v with v^T, we denote its transpose. A matrix $P \in \mathbb{R}^{N \times N}$ is said to be stochastic if all of its elements are non-negative and $P\mathbf{1} = \mathbf{1}$. Moreover, it is said to be doubly stochastic if it is stochastic and, in addition, $\mathbf{1}^T P = \mathbf{1}^T$. A stochastic matrix P is primitive if it has only one eigenvalue equal to 1 and all other eigenvalues are strictly inside the unitary circle. With the symbol $\rho_{\text{ess}}(P)$, we denote the essential spectral radius of P (see [13]), namely, the second largest eigenvalue of P in absolute value.

The symbol $\mathbb{E}$ denotes the expectation operator. Given two functions $f, g : \mathbb{N} \longmapsto \mathbb{R}$, we say that $f \in o(g)$ if $\lim_{n\to\infty}(f(n)/g(n)) = 0$.

[1] Basically, an *undirected path* is a path from a node to another node that does not respect the orientation of the edges.

II. Problem Formulation

The problem we deal with is that of estimating N variables $x_1, \dots, x_N$ from noisy measurements of the form

$$z_{ij} := x_i - x_j + n_{ij}, \quad i, j \in \{1, \dots, N\} \tag{2}$$

where n_{ij} is its zero-mean measurement noise. Although all results in this work apply to general vector-valued variables, for the sake of simplicity, in this paper, we assume that $x_i \in \mathbb{R}$, $i \in \{1, \dots, N\}$. This estimation problem can be naturally associated with a *measurement graph* $\mathcal{G}_m = (V; \mathcal{E}_m)$. The vertex set V of the measurement graph consists of the set of nodes $V = \{1, \dots, N\}$, where N is the number of nodes, while its edge set $\mathcal{E}_m$ consists of all of the ordered pairs of nodes (i, j) such that a noisy measurement of the form (2) between i and j is available to node i. The measurement errors on distinct edges are assumed uncorrelated. The measurement graph $\mathcal{G}_m$ is a directed graph since $(i, j) \in \mathcal{E}_m$ implies that the measurement z_{ij} is available to node i, while $(j, i) \in \mathcal{E}_m$ implies the measurement z_{ji} is available to node j, and these two are, in general, distinct.

Next, we formally state the problem we aim at solving. Let $\mathbf{x} \in \mathbb{R}^N$ be the vector obtained by stacking together all of the variables $x_1, \dots, x_N$, that is, $\mathbf{x} = [x_1, \dots, x_N]^T$, and let $\mathbf{z} \in \mathbb{R}^M$ and $\mathbf{n} \in \mathbb{R}^M$, where $M = |\mathcal{E}_m|$ are the vectors obtained by stacking all of the measurements together z_{ij} and the noises n_{ij}, respectively. In addition, let $R_{ij} > 0$ denote the covariance of the zero mean error n_{ij}, that is, $R_{ij} = \mathbb{E}[n_{ij}^2]$, and let $R \in \mathbb{R}^{M \times M}$ be the diagonal matrix collecting in its diagonal the covariances of the noises n_{ij}, $(i, j) \in \mathcal{E}_m$, that is, $R = \mathbb{E}[\mathbf{n}\mathbf{n}^T]$. Observe that (2) can be rewritten in a vector form as

$$\mathbf{z} = A\mathbf{x} + \mathbf{n}.$$

Now define the set

$$\chi := \arg\min_{\mathbf{x} \in \mathbb{R}^N} (\mathbf{z} - A\mathbf{x})^T R^{-1} (\mathbf{z} - A\mathbf{x}).$$

The goal is to construct an optimal estimate $\mathbf{x}_{\text{opt}}$ of $\mathbf{x}$ in a least square sense, namely, to compute

$$\mathbf{x}_{\text{opt}} \in \chi. \tag{3}$$

Assume the measurement graph $\mathcal{G}_m$ to be *weakly connected*, then it is well known (see [2]) that

$$\chi = \left\{ (A^T R^{-1} A)^\dagger A^T R^{-1} \mathbf{z} + \alpha \mathbf{1} \right\}.$$

Moreover let

$$\mathbf{x}^*_{\text{opt}} = (A^T R^{-1} A)^\dagger A^T R^{-1} \mathbf{z}$$

then $\mathbf{x}^*_{\text{opt}}$ is the minimum norm solution of (3), i.e.,

$$\mathbf{x}^*_{\text{opt}} = \min_{\mathbf{x}_{\text{opt}} \in \chi} \|\mathbf{x}_{\text{opt}}\|.$$

The matrix $A^T R^{-1} A$ is called in literature the *Weighted Generalized Grounded Laplacian* [2].

Remark II.1: Observe that just with relative measurements, determining the x_i's is only possible up to an additive constant. This ambiguity might be avoided by assuming that a node (say node 1) is used as reference node, i.e., $x_1 = 0$.

III. Synchronous Distributed Consensus-Based Solution

To compute an optimal estimate $\mathbf{x}_{\text{opt}}$ directly, one needs all of the measurements and their covariances $(\mathbf{z}, R)$, and the topology of the measurement graph $\mathcal{G}_m$. In this section, the goal is to compute the optimal solution in a distributed fashion, employing only local communications. In particular, we assume that a node i and another node j can communicate with each other if either $(i, j) \in \mathcal{E}_m$ or $(j, i) \in \mathcal{E}_m$. Accordingly, we introduce the *communication graph* $\mathcal{G}_c(V, \mathcal{E}_c)$, where $(i, j) \in \mathcal{E}_c$ if either $(i, j) \in \mathcal{E}_m$ or $(j, i) \in \mathcal{E}_m$. Observe that, if $(i, j) \in \mathcal{E}_c$, then also $(j, i) \in \mathcal{E}_c$, namely, $\mathcal{G}_c$ is a bidirected graph. From now on, $\mathcal{N}_i$ denotes the set of neighbors of node i in the communication graph $\mathcal{G}_c(V, \mathcal{E}_c)$.

In what follows, we introduce a distributed solution which is based on a standard linear consensus algorithm. A discussion of the linear consensus algorithm can be found in the review papers [14], [15]; hence, we refrain from describing it here. Instead, we make the presentation of the algorithm self-contained. First, we assume that the communications among the nodes are synchronous, namely, all nodes perform their transmissions and updates at the same instant, and design the algorithm for this scenario. We refer to this algorithm as the *synchronous consensus-based localization* algorithm (denoted hereafter as a s-CL algorithm). In Section IV, we will modify the s-CL algorithm to make it suitable for asynchronous communications. We assume that before running the s-CL algorithm, the nodes exchange with their neighbors their relative measurements as well as the associated covariances. So every node has access to the measurements on the edges that are incident to it, whether the edge is directed to or away from it. Each node uses the measurements obtained initially for all future computations. The s-CL algorithm is formally described as follows.

Processor states: For $i \in \{1, \dots, N\}$, node i stores in memory the measurements $\{z_{ij}, (i, j) \in \mathcal{E}_m\}$, and $\{z_{ji}, (j, i) \in \mathcal{E}_m\}$, and the associated covariances $\{R_{ij}, (i, j) \in \mathcal{E}_m\}$ and $\{R_{ji}, (j, i) \in \mathcal{E}_m\}$. Moreover, node i stores in memory an estimate $\hat{x}_i$ of x_i.

Initialization: For $i \in \{1, \dots, N\}$, node i initializes its estimate $\hat{x}_i(0)$ to any arbitrary value.

Transmission iteration: For $k \in \mathbb{N}$, at the start of the $(k+1)$th iteration of the algorithm, node i transmits its estimate $\hat{x}_i(k)$ to all of its neighbors. It also gathers the kth estimates of its neighbors $\hat{x}_j(k)$, $j \in \mathcal{N}_i$.

Update iteration: For $k \in \mathbb{N}$, node i, $i \in \{1, \dots, N\}$, based on the information received from its neighbors, updates its estimate as follows:

$$\hat{x}_i(k+1) := p_{ii} \hat{x}_i(k) + \sum_{j \in \mathcal{N}_i} p_{ij} \hat{x}_j(k) + b_i$$

where

$$b_i = \epsilon \sum_{(i,j) \in \mathcal{E}_m} R_{ij}^{-1} z_{ij} - \epsilon \sum_{(j,i) \in \mathcal{E}_m} R_{ji}^{-1} z_{ji}$$

and where

$$p_{ij} = \begin{cases} \epsilon\left(R_{ij}^{-1} + R_{ji}^{-1}\right), & \text{if } (i,j) \in \mathcal{E}_m \text{ and } (j,i) \in \mathcal{E}_m \\ \epsilon R_{ij}^{-1}, & \text{if } (i,j) \in \mathcal{E}_m \text{ and } (j,i) \notin \mathcal{E}_m \\ \epsilon R_{ji}^{-1}, & \text{if } (j,i) \in \mathcal{E}_m \text{ and } (i,j) \notin \mathcal{E}_m \end{cases}$$

and

$$p_{ii} = 1 - \sum_{j \in \mathcal{N}_i} p_{ij}$$

with ϵ being a positive constant *a priori* assigned to the nodes.

Now, let $P \in R^{N \times N}$ be the matrix defined by the weights p_{ij} introduced before. One can see that this matrix P is equal to

$$P = I - \epsilon A^T R^{-1} A.$$

Moreover, let

$$b = \epsilon A^T R^{-1} \mathbf{z}$$

and let $\hat{\mathbf{x}}(k) = [\hat{x}_1(k), \ldots, \hat{x}_N(k)]^T$. Then, the s-CL algorithm can be written in a compact form as

$$\hat{\mathbf{x}}(k+1) = P\hat{\mathbf{x}}(k) + b.$$

To characterize the convergence properties of the s-CL algorithm, we next introduce two definitions and a crucial property of the matrix P. First, let $d_{\max} = \max\{|\mathcal{N}_i|, i \in \{1, \ldots, N\}\}$. Second, let $R_{\min} = \min\{R_{ij}, (i,j) \in \mathcal{E}_m\}$. Observe that if $0 < \epsilon < 1/(2d_{\max}R_{\min}^{-1})$, then the matrix P is stochastic. If, in addition, the measurement graph $\mathcal{G}_m$ is weakly connected and, consequently, if communication graph $\mathcal{G}_c$ is strongly connected, then the matrix P is primitive. Under these assumptions, we have the following Proposition:

Proposition III.1: Consider the s-CL algorithm running over a *weakly connected* measurement graph $\mathcal{G}_m$. Let ϵ be such that $0 < \epsilon < 1/(2d_{\max}R_{\min}^{-1})$. Moreover, let $\hat{x}_i$, $i \in \{1, \ldots, N\}$ be initialized to any real number. Then, the following two facts hold true:

- the evolution $k \to \hat{\mathbf{x}}(k)$ asymptotically converges to an optimal estimate $\mathbf{x}_{\text{opt}} \in \chi$, that is, there exists $\alpha \in \mathbb{R}$, such that

$$\lim_{k \to \infty} \hat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt}}^* + \alpha \mathbf{1}$$

where α linearly depends on $\hat{\mathbf{x}}(0)$.
- the convergence is exponential, namely, there exists $C > 0, \rho_{\text{ess}} < 1$ such that

$$\|\hat{\mathbf{x}}(k) - \left(\mathbf{x}_{\text{opt}}^* + \alpha\mathbf{1}\right)\| \leq C\rho_{\text{ess}}^k(P)\|\hat{\mathbf{x}}(0) - \left(\mathbf{x}_{\text{opt}}^* + \alpha\mathbf{1}\right)\|.$$

Proof: We start by proving item i). Let us define the change of variable $\xi = \hat{\mathbf{x}} - \mathbf{x}_{\text{opt}}^*$. Since $\mathbf{x}_{\text{opt}}^* = P\mathbf{x}_{\text{opt}}^* + b$, it is possible to write

$$\begin{aligned} \hat{\mathbf{x}}(k+1) - x_{\text{opt}}^* &= P\hat{\mathbf{x}}(k) + b - \mathbf{x}_{\text{opt}}^* \\ &= P\hat{\mathbf{x}}(k) + b - \left(P\mathbf{x}_{\text{opt}}^* + b\right) \\ &= P\left(\hat{\mathbf{x}}(k) - \mathbf{x}_{\text{opt}}^*\right) \end{aligned}$$

and, in turn, $\xi(k+1) = P\xi(k)$. This equation describes the iteration of the classical consensus algorithm. Since P is a primitive doubly stochastic matrix, we have that

$$\xi(k) \to \frac{\mathbf{1}\mathbf{1}^T}{N}\xi(0)$$

where $\xi(0) = \hat{\mathbf{x}}(0) - \mathbf{x}_{\text{opt}}^*$. This implies that

$$\hat{\mathbf{x}}(k) \to \mathbf{x}_{\text{opt}}^* + \frac{\mathbf{1}\mathbf{1}^T}{N}\hat{\mathbf{x}}(0) - \frac{\mathbf{1}\mathbf{1}^T}{N}\mathbf{x}_{\text{opt}}^*.$$

The fact that $(\mathbf{1}\mathbf{1}^T/N)\mathbf{x}_{\text{opt}}^* = 0$ concludes the proof of item i).

Concerning item ii), it is well known [13] that the convergence rate of a consensus algorithm ruled by a primitive matrix P is exponential, whose delay coefficient is given by the essential spectral radius $\rho_{\text{ess}}(P)$. ■

Remark III.2: The s-CL algorithm is similar to the algorithm proposed in [9]. However, in [9], the measurement graph is assumed to be undirected, namely, both measurements z_{ij} and z_{ji} are available to node i and j under the additional assumption that $z_{ij} = -z_{ji}$.

Remark III.3: The authors in [16] solved the problem formulated in (3) by proposing a synchronous algorithm that implements the Jacobi iterative method. The performance of this algorithm, in terms of the rate of convergence to the optimal solution, is similar for many families of measurement graphs, to the performance of the synchronous consensus-based algorithm introduced in this section.

IV. Asynchronous Implementation of the Distributed Consensus-Based Solution

The distributed algorithm illustrated in the previous section has an important limitation: it is applicable only to sensor networks with synchronized and reliable communication. Indeed, the s-CL algorithm requires that there exists a predetermined common communication schedule for all nodes and, at each communication round, each node must simultaneously and reliably communicate its information. The aim of this section is to reduce the communication requirements of the s-CL algorithm, in particular, in terms of synchronization. To do so, we next introduce the *asynchronous Consensus-based Localization* algorithm (denoted as a-CL hereafter). This algorithm is based on an asymmetric broadcast communication protocol. Different from the s-CL, at each iteration of the a-CL, there is only one node transmitting information to all of its neighbors. Since the actual value of neighboring estimates is not available at each iteration, we assume that each node stores in its local memory a copy of the neighbors' variables recorded from the last communication received. For $j \in \mathcal{N}_i$, we denote by $\hat{x}_j^{(i)}(k)$ the estimate of x_j kept in i's local memory at the end of the kth iteration. If node j performed its last transmission to node i during the hth iteration, $h \leq k$, then $\hat{x}_j^{(i)}(k) = \hat{x}_j(h)$.

The a-CL algorithm is formally described as follows.

Processor states: For $i \in \{1, \ldots, N\}$, node i stores in memory the measurements z_{ij}, z_{ji} and the covariances R_{ij}, R_{ji} for all $j \in \mathcal{N}_i$. Moreover, node i stores in memory also the estimate $\hat{x}_i$ of x_i and, for $j \in \mathcal{N}_i$, an estimate $\hat{x}_j^{(i)}$ of $\hat{x}_j$.

Initialization: Every node i initializes its estimate $\hat{x}_i$ and the variables $\hat{x}_j^{(i)}$, $j \in \mathcal{N}_i$ to arbitrary values.

Transmission iteration: For $k \in \mathbb{N}$, at the start of the $(k+1)$th iteration of the algorithm, there is only one node, say i, which transmits information to its neighbors; precisely, node i sends the value of its estimate $\hat{x}_i(k)$ to node j, $j \in \mathcal{N}_i$.

Update iteration: For $j \in \mathcal{N}_i$, node j performs the following actions in order

i) it sets $\hat{x}_i^{(j)}(k+1) = \hat{x}_i(k)$, while for $s \in \mathcal{N}_j \setminus \{i\}$, $\hat{x}_s^{(j)}$ is left unchanged, that is, $\hat{x}_s^{(j)}(k+1) = \hat{x}_s^{(j)}(k)$;

ii) it updates $\hat{x}_j$ as

$$\hat{x}_j(k+1) := p_{jj}\hat{x}_j(k) + \sum_{h \in \mathcal{N}_j} p_{jh}\hat{x}_h^{(j)}(k+1) + b_j. \quad (4)$$

Clearly for $s \notin \mathcal{N}_i$, $\hat{x}_s$ is left unchanged during the $(k+1)$th iteration of the algorithm, that is, $\hat{x}_s(k+1) = \hat{x}_s(k)$.

Remark IV.1: Observe that the aforementioned algorithm has been described assuming that the communication channels are reliable, that is, no packet losses occur, and that the communication delays are negligible, that is, when node i performs a transmission, the estimate $\hat{x}_i$ is instantaneously used by its neighbors. We will come back on these nonidealities in Section VI.

Next, we rewrite the updating step of the a-CL in a more compact way. Observe preliminarily that under the assumption of reliable communications and by denoting $\bar{k}$ as the first iteration after which all nodes have transmitted at least once, then the estimate of node x_i stored in the neighbors of node i is always the same, that is, for all $k \geq \bar{k}$ and $\ell, j \in \mathcal{N}_i$, we have $\hat{x}_i^{(\ell)}(k) = \hat{x}_i^{(j)}(k)$. Moreover, if we denote $t_i'(k)$ as the iteration during which node i has performed its last transmission up to iteration k of the a-CL (that is, $\hat{x}_i(t_i'(k))$ is the value of $\hat{x}_i$ at its last communication round), then for $j \in \mathcal{N}_i$, $\hat{x}_i^{(j)}(t'') = \hat{x}_i(t_i'(k))$ for all t'' such that $t_i'(k) < t'' \leq k$.

Now let us define $x_i'(k) = \hat{x}_i(k)$ and $x_i''(k) = \hat{x}_i(t_i'(k))$ and, accordingly, let $\mathbf{x}'(k) = [x_1'(k), \ldots, x_N'(k)]^T$ and $\mathbf{x}''(k) = [x_1''(k), \ldots, x_N''(k)]^T$. Moreover, let $Q_i \in \mathbb{R}^{2N \times 2N}$ be defined as

$$Q_i = \begin{bmatrix} Q_{11}^{(i)} & Q_{12}^{(i)} \\ Q_{21}^{(i)} & Q_{22}^{(i)} \end{bmatrix} \quad (5)$$

where

$$Q_{11}^{(i)} = \sum_{h \notin \mathcal{N}_i} e_h e_h^T + \sum_{j \in \mathcal{N}_i} \left(p_{jj} e_j e_j^T + p_{ji} e_j e_i^T \right)$$

$$Q_{12}^{(i)} = \sum_{j \in \mathcal{N}_i} e_j \left(\sum_{h \in \mathcal{N}_j / i} p_{jh} e_h^T \right)$$

$$Q_{21}^{(i)} = e_i e_i^T$$

$$Q_{22}^{(i)} = I - e_i e_i^T$$

with e_ℓ, $\ell \in \{1, \ldots, N\}$ being the N-dimensional vector having all of the components equal to zero except the ℓth component which is equal to 1. Observe that for $i \in \{1, \ldots, N\}$, Q_i is a $2N$-dimensional stochastic matrix. Finally, let

$$B_i = \begin{bmatrix} \sum_{j \in \mathcal{N}_i} e_j^T b \\ 0_N \end{bmatrix}.$$

Assume, w.l.o.g., that node i is the node performing the transmission during the $(k+1)$th iteration of the a-CL. Hence, the updating step of a-CL can be written in vector form as

$$\begin{bmatrix} \mathbf{x}'(k+1) \\ \mathbf{x}''(k+1) \end{bmatrix} = Q_i \begin{bmatrix} \mathbf{x}'(k) \\ \mathbf{x}''(k) \end{bmatrix} + B_i, \quad k \geq \bar{k}. \quad (6)$$

Now let us introduce the auxiliary variable

$$\xi(k) = \begin{bmatrix} \mathbf{x}'(k) \\ \mathbf{x}''(k) \end{bmatrix} - \begin{bmatrix} \mathbf{x}_{\text{opt}}^* \\ \mathbf{x}_{\text{opt}}^* \end{bmatrix}.$$

By exploiting the fact that, for $i \in \{1, \ldots, N\}$

$$\begin{bmatrix} \mathbf{x}_{\text{opt}}^* \\ \mathbf{x}_{\text{opt}}^* \end{bmatrix} = Q_i \begin{bmatrix} \mathbf{x}_{\text{opt}}^* \\ \mathbf{x}_{\text{opt}}^* \end{bmatrix} + B_i \quad (7)$$

we have that the variable ξ satisfies the following $2N$-dimensional recursive equation

$$\xi(k+1) = Q_i \xi(k). \quad (8)$$

Observe that $\hat{\mathbf{x}}(k) \to \mathbf{x}_{\text{opt}}^* + \alpha \mathbf{1}$ if and only if $\xi(k) \to \alpha \mathbf{1}$. Moreover, since Q_i is a stochastic matrix for any $i \in \{1, \ldots, N\}$, we have that (8) represents a $2N$-dimensional time-varying consensus algorithm.

In the following sections, we analyze the convergence properties and the robustness to delays and packet losses of the a-CL algorithm by studying system (8) resorting to the mathematical tools developed in the literature of the consensus algorithms. In particular, we will provide our results considering two different scenarios which are formally described in the following definitions.

Definition IV.2 (Randomly Persistent Communicating Network): A network of N nodes is said to be a *randomly persistent communicating network* if there exists a N-upla $(\beta_1, \ldots, \beta_N)$ such that $\beta_i > 0$, for all $i \in \{1, \ldots, N\}$, and $\sum_{i=1}^N \beta_i = 1$, and such that, for all $k \in \mathbb{N}$

$$\mathbb{P}[\text{the transmitting node at iteration } k \text{ is node } i] = \beta_i.$$

Definition IV.3 (Uniformly Persistent Communicating Network): A network of N nodes is said to be a *uniformly persistent communicating network* if there exists a positive integer number τ such that, for all $k \in \mathbb{N}$, each node transmits the value of its estimate to its neighbors at least once within the time interval $[k, k+\tau)$.

V. Performance Analysis of a-CL Algorithm Under Randomly Persistent Communications

The following result characterizes the convergence properties of the a-CL when the network is randomly persistent communicating.

Proposition V.1: Consider a *randomly persistent communicating network* of N nodes running the a-CL algorithm over a weakly connected measurement graph $\mathcal{G}_m$. Let ϵ be such that $0 < \epsilon < 1/(2d_{\max}R_{\min}^{-1})$. Moreover, let $\hat{x}_i$, $i \in \{1,\dots,N\}$, $\hat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true

i) the evolution $k \to \hat{\mathbf{x}}(k)$ converges almost surely to an optimal solution $\mathbf{x}_{\text{opt}} \in \chi$, that is, there exists $\alpha \in \mathbb{R}$ such that

$$\mathbb{P}\left[\lim_{k\to\infty} \hat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt}}^* + \alpha\mathbf{1}\right] = 1$$

ii) the evolution $k \to \hat{\mathbf{x}}(k)$ is exponentially convergent in the mean-square sense, that is, there exists $C > 0$ and $0 \le \rho < 1$ such that

$$\lim_{k\to\infty} \mathbb{E}\left[\left\|\hat{\mathbf{x}}(k) - \left(\mathbf{x}_{\text{opt}}^* + \alpha\mathbf{1}\right)\right\|^2\right] \le C\rho^k \mathbb{E}\left[\left\|\hat{\mathbf{x}}(0) - \left(\mathbf{x}_{\text{opt}}^* + \alpha\mathbf{1}\right)\right\|^2\right].$$

Proof: The proof of Proposition V.1 is based on proving the convergence to a consensus of (8) using the mathematical tools developed in [17]. Let σ be the random process such that $\sigma(k)$ denotes the node performing the transmission action at the beginning of the $k+1$th iteration. Clearly, in the randomized scenario we are considering, we have that, for $i \in \{1,\dots,N\}$, $\mathbb{P}[\sigma(k) = i] = \beta_i$ for all k. Let

$$S(k) = \prod_{h=0}^{k} Q_{\sigma(h)}.$$

Observe that $S(k)$ inherits the same block structure of the matrices $\{Q_i\}_{i=1}^N$; namely, we can write

$$S(k) = \begin{bmatrix} S_{11}(k) & S_{12}(k) \\ S_{21}(k) & S_{22}(k) \end{bmatrix}.$$

As a consequence of [17, Theor. 3.1] the a-CL reaches almost surely consensus if and only if, for every i and j in V

$$\mathbb{P}[\mathcal{E}_{ij}] = 1 \tag{9}$$

where

$$\mathcal{E}_{ij} = \{\exists \ell, \exists k | S_{i\ell}(k) S_{j\ell}(k) > 0\}.$$

Now observe that since the measurement graph is weakly connected, then the communication graph is a connected undirected graph. This fact, together with the fact that the diagonal elements of $Q_{11}^{(i)}$ are all positive for any $i \in \{1,\dots,N\}$, implies that there exists almost surely $\bar{k}$ such that, for all $k' \ge \bar{k}$, all elements of the matrix $S_{11}(k')$ are strictly greater than 0. Assume now, without loss of generality, that $\sigma(k') = i$, for $k' \ge k$. Then, since the ith row of $S_{21}(k'+1)$ is equal to $e_i e_i^T S_{11}(k')$, it turns out that all elements of the ith row of $S_{21}(k'+1)$ are strictly greater than 0. Moreover, it is easy to see that they will remain strictly greater than 0 also for any $k'' \ge k'$. Hence, we can argue that there exists almost surely, also a $\bar{k}'$ such that for all $k' \ge \bar{k}'$, all of the elements of the matrix $S_{21}(k')$ are strictly greater than 0. It follows that the property stated in (9) is satisfied for any $k \ge \bar{k}'$ and for any $\ell \in \{1,\dots,N\}$. This concludes the proof of item i).

Concerning item ii), we again resort to the results in [17]. Let $\Omega = I - (1/2N)\mathbf{1}\mathbf{1}^T$ where in this expression we assume that I is the $2N$-dimensional identity matrix and the vector $\mathbf{1}$ is $2N$-dimensional. From the results in [17], it follows that to study the rate of convergence of $\mathbb{E}[\|\xi(k) - \alpha\mathbf{1}\|^2]$, it is equivalent to study the convergence rate of $\mathbb{E}\|\Omega\xi(k)\|^2$ and, in particular, of the linear recursive system

$$\Delta(t+1) = \mathbb{E}\left[Q_{\sigma(0)}^T \Delta(t) Q_{\sigma(0)}\right]$$

where $\Delta(0) = \Omega$. Observe that $\Delta(t)$ is the evolution of a linear dynamical system which can be written in the form

$$\Delta(t+1) = \mathcal{L}(\Delta(t))$$

where $\mathcal{L} : \mathbb{R}^{2N\times 2N} \to \mathbb{R}^{2N\times 2N}$ is given by

$$\mathcal{L}(M) = \mathbb{E}\left[Q_{\sigma(0)}^T M Q_{\sigma(0)}\right].$$

As highlighted in [17], the linear operator $\mathcal{L}$ can be represented by the matrix $\mathbf{L} = \mathbb{E}[Q_{\sigma(0)} \otimes Q_{\sigma(0)}]^T$, where $\otimes$ denotes the Kronecker product of matrices. Following the proof of [17, Prop. 4.3], one can see that $\mathbf{L}^T$ is a primitive stochastic matrix which, therefore, has the eigenvalue 1 with algebraic multiplicity 1. Moreover, $\mathbf{L}^T(\mathbf{1}\otimes\mathbf{1}) = (\mathbf{1}\otimes\mathbf{1})$ and $(\mathbf{1}\otimes\mathbf{1})(\Omega\otimes\Omega) = 0$, from which it follows that $\mathbb{E}\|\Omega\xi(k)\|^2 \le C\rho_{\text{ess}}(\mathbf{L}^T)\mathbb{E}\|\Omega\xi(0)\|^2$, where $\rho_{\text{ess}}(\mathbf{L}^T)$ denotes the essential spectral radius of $\mathbf{L}^T$. ■

A. Bounds on the Convergence Rate of the a-CL Algorithm

In this section, we provide some insights on the convergence rate of the a-CL algorithm in the randomly persistent communicating scenario. To do so, we consider (8), whose performance in terms of rate of convergence to the consensus can be analyzed by following again the treatment in [17]. Typically, one would like to study the convergence rate of a randomized consensus algorithm by providing a mean-square analysis of the behavior of the distance between the state and the asymptotic consensus point, namely, by analyzing the rate of convergence of the quantity $\mathbb{E}[\|\xi - \alpha\mathbf{1}\|^2]$. Unfortunately, this is not a trivial task in general. To overcome this difficulty, we study the evolution of $\Omega\xi$. The first consequence of the results obtained in [17] is that the quantities $\mathbb{E}[\|\xi - \alpha\mathbf{1}\|^2]$ and $\mathbb{E}[\|\Omega\xi\|^2]$ have the same exponential convergence rate to zero, or, more formally, given any initial condition $\xi(0)$

$$\limsup_{k\to\infty} \mathbb{E}\left[\|\xi(k) - \alpha\mathbf{1}\|^2\right]^{\frac{1}{k}} = \limsup_{k\to\infty} \mathbb{E}\left[\|\Omega\xi(k)\|^2\right]^{\frac{1}{k}}.$$

For this reason, in what follows, we study the right-hand expression, which turns out to be simpler to analyze. In order to have a single performance metric not dependent on the initial condition, we focus on this worst case exponential rate of convergence

$$R = \sup_{\xi(0)} \limsup_{k\to\infty} \mathbb{E}\left[\|\Omega\xi(k)\|^2\right]^{\frac{1}{k}}.$$

It has been proved in [17, Prop. 4.4] that

$$\left[\rho_{\text{ess}}(\bar{Q})\right]^2 \leq R \leq sr\left(\mathbb{E}\left(Q_i^T \Omega Q_i\right)\right) \tag{10}$$

where $\bar{Q}$ is the average consensus matrix, namely, $\bar{Q} = \mathbb{E}[Q_i] = \sum_{i=1}^N \beta_i Q_i$ and where $sr(\mathbb{E}(Q_i^T\Omega Q_i))$ denotes the spectral radius of the semidefinite positive matrix $\mathbb{E}(Q_i^T\Omega Q_i)$, that is, its largest eigenvalue. Unfortunately, it turns out from a numerical inspection over significant families of graphs, such as Cayley graphs (see [13]) and random geometric graphs, that the upper bound $sr(\mathbb{E}(Q_i^T\Omega Q_i))$ is greater than 1, that is, it is not informative for our analysis. However, we have run a number of MonteCarlo simulations randomized over graphs of different topology and size and over different initial conditions, and it always resulted that $\limsup_{k\to\infty} \mathbb{E}[\|\Omega\xi(k)\|^2]^{1/k} \leq \rho_{\text{ess}}(\bar{Q})$. Based on this experimental evidence, we formulate the following conjecture.

Conjecture V.2: The quantity $\rho_{\text{ess}}(\bar{Q})$ is an upper bound for the exponential convergence rate R, i.e.,

$$R \leq \rho_{\text{ess}}(\bar{Q}).$$

The aforementioned conjecture and the fact that $[\rho_{\text{ess}}(\bar{Q})]^2 \leq R$ motivates to study $\rho_{\text{ess}}(\bar{Q})$.

Remark V.3: Notice that (8) describes a higher order consensus algorithm, for which few analytic tools are available. In fact, the few works available in the literature which address the rate of convergence of randomized higher order consensus algorithms are limited to the convergence in expectation [12].

B. Rate Analysis of a-CL Algorithm for Regular Graphs

In this section, we assume that the measurements graph $\mathcal{G}_m = (V, \mathcal{E}_m)$ is a strongly connected bidirected regular graph such that for $i \in \{1, \ldots, N\}$, $|\mathcal{N}_i| = \nu$. Moreover, we assume the following properties.

Assumption V.4: We have that:

i) the error measurements covariances are all identical, that is, $R_{ij} = R$ for all $(i, j) \in \mathcal{E}_m$;
ii) $\epsilon = R/(2(\nu + 1))$;
iii) the probabilities $\{\beta_1, \ldots, \beta_N\}$ are uniform, that is, $\beta_1 = \ldots = \beta_N = 1/N$.

Observe that, from properties i) and ii) of Assumption V.4, it turns out that the matrix $P = I - \epsilon A^T R^{-1} A$, associated with the s-CL algorithm, is symmetric and such that $P_{ij} = 1/(\nu + 1)$ for $j \in \mathcal{N}_i \cup \{i\}$. Let $\lambda_1(P) = 1 > \lambda_2(P) \geq \ldots \geq \lambda_N(P)$ be the eigenvalues of P. Then, $\rho_{\text{ess}}(P) = \max\{|\lambda_2(P)|, |\lambda_N(P)|\}$. The following Lemma illustrates how the $2N$ eigenvalues of $\bar{Q}$ are related to those of P.

Lemma V.5: Consider the a-CL algorithm running over a bidirected regular graph $\mathcal{G}_m = (V, \mathcal{E}_m)$ such that, for $i \in \{1, \ldots, N\}$, $|\mathcal{N}_i| = \nu$. Assume Assumption V.4 holds true. Then, the $2N$ eigenvalues of $\bar{Q}$ are the solutions of the following N second-order equations:

$$f(s; \lambda_i, N, \nu) = s^2 + (a + b)s + (ab + c) \tag{11}$$

where

$$a = -\left[\frac{N-\nu}{N} + \frac{\lambda_i}{N} + \frac{\nu - 1}{N(\nu+1)}\right]$$
$$b = -\frac{N-1}{N}, \quad c = -\frac{\nu - 1}{N^2}\left(\lambda_i - \frac{1}{\nu+1}\right).$$

Now let $s_1^{(i)}$ and $s_2^{(i)}$ denote the two solutions of $f(s; \lambda_i, N, \nu)$. It easy to see that $s_1^{(1)} = 1$ and $s_2^{(1)} = 1 - ((\nu^2 + 1)/N(\nu + 1))$. The following result restricts the search of $\rho_{\text{ess}}(\bar{Q})$ among the values $|s_1^{(2)}|$, $|s_2^{(2)}|$, and $1 - ((\nu^2 + 1)/N(\nu + 1))$.

Theorem V.6: Consider the a-CL algorithm running on a bidirected regular graph $\mathcal{G}_m = (V, \mathcal{E}_m)$ such that for $i \in \{1, \ldots, N\}$, $|\mathcal{N}_i| = \nu$. Assume Assumption V.4 holds true. Moreover let, as shown in the equation at the bottom of the page, then

i) if $1 - \rho_{\text{ess}}(P) \leq \gamma^* \Longrightarrow \rho_{\text{ess}}(\bar{Q}) = \max(|s_1^{(2)}|, |s_2^{(2)}|)$;
ii) if $1 - \rho_{\text{ess}}(P) > \gamma^* \Longrightarrow \rho_{\text{ess}}(\bar{Q}) = s_2^{(1)} = 1 - ((\nu^2 + 1)/N(\nu + 1))$.

The proofs of Lemma V.5 and Theorem V.6 follow from standard algebraic manipulations. Due to space constraints, we do not include them here, but we refer the interested reader to the document [18].

We provide now an asymptotic result on $\rho_{\text{ess}}(\bar{Q})$. To do so, consider a sequence of connected undirected regular graphs $\mathcal{G}_N$ of increasing size N, and fixed degree ν. Assume Assumption V.4 holds true for any $\mathcal{G}_N$. Then, to any $\mathcal{G}_N$, we can associate a stochastic matrix P_N such that $(P_N)_{ij} = 1/(\nu + 1)$ for all $j \in \mathcal{N}_i \cup \{i\}$. Let us assume the following property.

Assumption V.7: Consider the sequence of matrices P_N associated with the sequence of graphs $\mathcal{G}_N$ that were described before and assume that

$$\rho_{\text{ess}}(P_N) = 1 - \varepsilon(N) + o\left(\varepsilon(N)\right) \tag{12}$$

where $\varepsilon : \mathbb{N} \to \mathbb{R}$ is a positive function such that $\varepsilon(N) \to 0$ as $N \to \infty$.

Important families of matrices satisfying the above assumption (12) are given by the matrices built over the d-dimensional

$$\gamma^* = \frac{\nu - 1 + N(\nu+1) - \sqrt{N^2(\nu+1)^2 - 2N(\nu^3 + \nu + 2) + (\nu-1)^2 + (\nu^2+1)^2}}{\nu + 1}$$

tori and the Cayley graphs (see [13]). It is worth remarking that the tori and the Cayley graphs have been shown to exhibit important spectral similarities with the random geometric graphs [19], which is a family of graphs that during the last few years, and have been successfully used to model wireless communication in many applications [20]. Now, let the matrix $\bar{Q}_N$ represent the average matrix associated with the a-CL algorithm running over $\mathcal{G}_N$. The following result characterizes the asymptotic behavior of $\rho_{\text{ess}}(\bar{Q}_N)$, with respect to $\rho_{\text{ess}}(P_N)$.

Proposition V.8: Consider the sequence of graphs $\mathcal{G}_N$ described before. Consider the a-CL algorithm running over $\mathcal{G}_N$. Assume Assumption V.4 and Assumption V.7 hold true. Then

$$\rho_{\text{ess}}(\bar{Q}_N) = 1 - \frac{\nu(\nu+1)}{N(\nu^2+1)}\varepsilon(N) + o\left(\frac{\varepsilon(N)}{N}\right). \qquad (13)$$

Proof: Let $\gamma_i = 1 - \lambda_i$, then we can rewrite (11) as

$$f(s;\lambda_i,N,\nu) = d(s;N,\nu) + \gamma_i n(s;N,\nu) \triangleq g(s;\gamma_i,N,\nu)$$

so that g is an explicit function of γ_i, where

$$d(s;N,\nu) = s^2 - \frac{2N(\nu+1)-(\nu^2+1)}{N(\nu+1)}s + 1 - \frac{\nu^2+1}{N(\nu+1)}$$

$$n(s;sN,\nu) = \frac{s}{N} + \frac{\nu-N}{N^2}.$$

Note that

$$\lim_{N\to\infty} \gamma^*(\nu,N) = \frac{\nu^2+1}{(\nu+1)^2}.$$

Therefore, according to Theorem V.6 and assumption V.7, for N sufficiently large, $\rho_{\text{ess}}(\bar{Q}_N)$ is given by i). Since $\gamma_2 = 1 - \lambda_2 + \epsilon(N) + o(\epsilon(N))$, then $s_1^{(2)} = \bar{s}_1^{(2)} + \alpha\varepsilon(N) + o(\varepsilon(N))$ and $s_2^{(2)} = \bar{s}_2^{(2)} + \beta\varepsilon(N) + o(\varepsilon(N))$ for some scalar α, β, where $\lambda_2 = \rho_{\text{ess}}(P_N)$ and $\bar{s}_1^{(2)}, \bar{s}_2^{(2)}$ are the solutions of second-order equation $g(s;0,N,\nu) = 0$. It is easy to verify that $\bar{s}_1^{(2)} = 1$ and $\bar{s}_2^{(2)} = 1 - ((\nu^2+1)/N(\nu+1))$. Since $|\bar{s}_1^{(2)}| > |\bar{s}_2^{(2)}|$, then for N sufficiently large and by continuity, we have $\rho_{\text{ess}}(\bar{Q}_N) = |s_1^{(2)}|$. We are therefore interested in explicitly computing the scalar α. Since

$$\begin{cases} g(1;0,N,\nu) = 0 \\ \left.\frac{\partial g}{\partial s}\right|_{(1,0,N,\nu)} \neq 0 \end{cases}$$

it is possible to exploit the implicit function theorem that allows us to write

$$\begin{aligned} s_1^{(2)} &= 1 - \frac{\partial g}{\partial \gamma_i}\left(\frac{\partial g}{\partial s}\right)^{-1}\Bigg|_{(1,0,N,\nu)} (\epsilon(N) + o(\epsilon(N))) \\ &= 1 - \frac{\nu(\nu+1)}{N(\nu^2+1)}\epsilon(N) + o\left(\frac{\epsilon(N)}{N}\right) \end{aligned}$$

which means that $\rho_{\text{ess}}(\bar{Q})$ can be expressed as

$$\rho_{\text{ess}}(\bar{Q}) = 1 - \frac{\nu(\nu+1)}{N(\nu^2+1)}\epsilon(N) + o\left(\frac{\epsilon(N)}{N}\right). \qquad (14)$$

■

Thanks to [17], we know that the rates of convergence are lower bounded by $\rho_{\text{ess}}(\bar{Q}_N)^2$, while we recall we conjecture that $R \leq \rho_{\text{ess}}(\bar{Q}_N)$. From the aforementioned proposition, we can obtain the following result which compares the convergence rate R of the a-CL algorithm with respect to the convergence rate of the s-CL algorithm.

Corollary V.9: Consider a sequence of graphs $\mathcal{G}_N$ as in Proposition V.8. Consider the a-CL algorithm running over $\mathcal{G}_N$. Assume Assumption V.4 and Assumption V.7 hold true. Then, for $N \gg 1$

$$\frac{1-\rho_{\text{ess}}(\bar{Q}_N)}{1-\rho_{\text{ess}}(P_N)} \simeq \frac{\nu(\nu+1)}{N(\nu^2+1)}$$

and

$$\frac{1-\left[\rho_{\text{ess}}(\bar{Q}_N)\right]^2}{1-\rho_{\text{ess}}(P_N)} \simeq 2\frac{\nu(\nu+1)}{N(\nu^2+1)}.$$

Namely, assuming Conjecture V.2 holds true, the a-CL algorithm slows down of a factor $1/N$ with respect to the synchronous implementation.

Observe the fact that the rate of convergence in expectation is reduced by a factor N, and it is not surprising because in the a-CL, there is only one node transmitting information at each iteration.

Remark V.10: It is worth remarking that also standard memoryless asynchronous consensus algorithms based on asymmetric broadcast communication protocols, slow down their convergence rate by a factor $1/N$ with respect to the standard synchronous consensus implementations, see [21]. In other words, the presence of memory storage in the a-CL does not further deteriorate the convergence rate with respect to standard memoryless asynchronous consensus algorithms.

Remark V.11: Similar analysis can be provided also for other relevant families of regular graphs, such as the complete graphs and more, in general, the Ramanujan graphs [22]. Let us recall the asymptotic lower bound proved by Alon and Boppana for doubly stochastic matrices built over ν-regular bidirected graphs. Specifically, if A denotes the adjacency matrix of a ν-regular bidirected graph, let P be the doubly stochastic matrix defined as $P = \nu^{-1}A$, then

$$\liminf_{N\to\infty} \rho_{\text{ess}}(P) \geq \frac{2\sqrt{\nu-1}}{\nu}$$

where the lim inf is taken along the family of all ν-regular bidirected graphs having N vertices. Ramanujan graphs are those ν-regular bidirected graphs which achieves the previous bound, that is, such that $\rho_{\text{ess}}(P) = 2\sqrt{\nu-1}/\nu$. Hence, through the Ramanujan graphs, it is possible to keep the essential spectral radius bounded away from 1, while keeping the degree fixed. Exploiting Theorem V.6, it is possible to prove that for the a-CL algorithm running over Ramanujan graphs, it holds $\rho_{\text{ess}}(\bar{Q}) = 1 - \alpha(\nu)/N$, where $\alpha(\nu) \leq 1$ depends only on the degree ν, and, in turn

$$\frac{1-\rho_{\text{ess}}(\bar{Q})}{1-\rho_{\text{ess}}(P)} \simeq \frac{\alpha(\nu)\nu}{N\sqrt{\nu+1}}.$$

In other words also for the Ramanujan graphs, the a-CL algorithm is slowed down by a factor of $1/N$ with the respect to the synchronous implementation. Concerning the complete graphs, we have $\rho_{\text{ess}}(P) = 0$ and, again from Theorem V.6, that $\rho_{\text{ess}}(\bar{Q}) = 2(N-1)/N$. Hence, it follows that for $N \gg 1$

$$\frac{1-\rho_{\text{ess}}(\bar{Q})}{1-\rho_{\text{ess}}(P)} \simeq 1.$$

Namely, the a-CL algorithm is not slowed down by a factor N. This is due to the fact that when a complete graph is employed, the number of neighbors of each node linearly increases with the size of the graph. Due to space constraints, we do not include here all of the details of the analysis related to the Ramanujan graphs which, however, can be found in [18].

It is worth remarking that even though there are plenty of Ramanujan graphs, it is still an open problem if for any pair N and ν there exist Ramanujan graphs with N vertices and of degree ν. Moreover, even if they exist, their construction is quite complex, thus making them of marginal interest from an application standpoint.

Remark V.12: Following Remark III.3, it is worth stressing that also the Jacobi-like strategy introduced in [2] is amenable of asynchronous implementation, see [7]. However, to the best of our knowledge, no theoretical analysis of the rate of convergence of the asynchronous version, introduced in [7], has been proposed in the literature.

VI. Robustness Properties of the a-CL Algorithm With Respect to Packet Losses and Delays

In Section IV, we have introduced the a-CL algorithm assuming that the communication channels are reliable, that is, no packet losses occur, and that the transmission delays are negligible. In this section, we relax these assumptions and we show that the a-CL algorithm still converges, provided that the network is uniformly persistent communicating and the transmission delays and the frequencies of communication failures satisfy mild conditions which we formally describe next.

Assumption VI.1 (Bounded Packet Losses): There exists a positive integer L such that the number of consecutive communication failures between every pair of neighboring nodes in the communication graph $\mathcal{G}_c$ is less than L.

Assumption VI.2 (Bounded Delay): Assume node i broadcasts its estimate to its neighbors at the beginning of iteration k and assume that the communication link (i,j) does not fail. Then, there exists a positive integer D such that the information $\hat{x}_i(k)$ is used by node j to perform its local update no later than iteration $k+D$.

Loosely speaking, Assumption VI.1 implies that there can be no more than L consecutive packet losses between any pair of nodes i,j belonging to the communication graph. Differently, Assumption VI.2 considers the scenario where the received packets are not used instantaneously, but are subject to some delay no greater than D iterations.

Clearly, in this more realistic scenario, it turns out that the implementation of the a-CL is slightly different from the description provided in Section IV. Specifically, consider the kth iteration of the a-CL algorithm and, without loss of generality, assume node i is the transmitting node during this iteration. Due to the presence of packet losses and delays, the set of updating nodes might be, in general, different from the set $\mathcal{N}_i$. In fact, for $j \in \mathcal{N}_i$, node j does not perform any update since the packet $\hat{x}_i(k)$ from node i is lost or simply because the update is delayed. Moreover, there might be a node $h \notin \mathcal{N}_i$ which, during iteration k, decides to perform an update since it received a packet $\hat{x}_s$, $s \in \mathcal{N}_h$, within the last D iterations. This scenario can be formally represented by the set of nodes $V'(k) \subseteq V$ which decide to perform an update at iteration k. Then (4) can be rewritten as

$$\hat{x}_j(k+1) := p_{jj}\hat{x}_j(k) + \sum_{h \in \mathcal{N}_j} p_{jh}\hat{x}_h\left(k'_h\right) + b_j \qquad (15)$$

for all $j \in V'(k)$, where $k-(\tau L+D) \le k'_h \le k$, thta is, loosely speaking when an update is performed, the local estimate of the neighboring nodes cannot be older than $\tau L + D$ iterations.[2] Indeed, if $L = D = 0$, then we recover the standard a-CL algorithm where $V'(k) = \mathcal{N}_i$.

The following result characterizes the convergence properties of the a-CL in the presence of delays, packet losses, and when the network is uniformly persistent communicating.

Proposition VI.3: Consider a *uniformly persistent communicating network* of N nodes running the a-CL algorithm over a weakly connected measurement graph $\mathcal{G}_m$. Let Assumptions VI.1 and VI.2 be satisfied. Let ϵ be such that $0 < \epsilon < 1/(2d_{\max}R_{\min}^{-1})$. Moreover, let $\hat{x}_i$, $i \in \{1,\dots,N\}$, $\hat{x}_j^{(i)}$, $j \in \mathcal{N}_i$ be initialized to any real number. Then, the following facts hold true:

i) the evolution $k \to \hat{\mathbf{x}}(k)$ asymptotically converges to an optimal estimate $\mathbf{x}_{\text{opt}} \in \chi$, that is, there exists $\alpha \in \mathbb{R}$ such that

$$\lim_{k\to\infty} \hat{\mathbf{x}}(k) = \mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1};$$

ii) the convergence is exponential, namely, $C > 0$ and $0 \le \rho < 1$ exists such that

$$\left\|\hat{\mathbf{x}}(k) - \left(\mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1}\right)\right\| \le C\rho^k \left\|\hat{\mathbf{x}}(0) - \left(\mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1}\right)\right\|.$$

Proof: The proof follows from the statement of [23, Prop. 1] which for convenience has been reported in the Appendix as Proposition A.5. To apply Proposition A.5, we show that the a-CL algorithm in the presence of delays and packet losses can be rewritten as a consensus with delays that satisfies Assumptions A.1, A.2, A.3, and A.4 reported in the Appendix.

To this aim, let $\delta_j(k) = \hat{x}_j(k) - [\mathbf{x}^*_{\text{opt}}]_j$, where $[\mathbf{x}^*_{\text{opt}}]_j$ denotes the jth component of the vector $\mathbf{x}^*_{\text{opt}}$. Recalling that $\mathbf{x}^*_{\text{opt}} = P\mathbf{x}^*_{\text{opt}} + b$ and, according to (15), we have that, if $j \in V'(k)$

$$\delta_j(k+1) := p_{jj}\delta_j(k) + \sum_{h \in \mathcal{N}_j} p_{jh}\delta_h\left(k'_h\right). \qquad (16)$$

[2]Recall that we are assuming the network is uniformly persistent communicating, namely, for all $k \in \mathbb{N}$, each node performs at least one transmission within the time interval $[k, k+\tau)$.

Otherwise

$$\delta_j(k+1) = \delta_j(k).$$

The above equations describe a consensus algorithm on the variables $\delta_1, \ldots, \delta_N$ which satisfies Assumptions A.1, A.2, A.3, A.4 reported in the Appendix. Indeed, Assumption A.1 on the weights is trivially satisfied. Assumption A.2 follows from the facts that the communication graph $\mathcal{G}_c$ is connected, the network is uniformly persistent communicating and Assumptions VI.1 and VI.2. Assumption A.3 is a consequence of the fact that the network is uniformly persistent communicating and Assumption VI.1; in our setup, we have $B = L\tau$. Finally, Assumption A.4 follows from Assumption VI.2 and (16). Hence, the variables $\delta_1, \ldots, \delta_N$ converge exponentially to a consensus value α which, in turn, implies that $\hat{\mathbf{x}}$ converges exponentially to $\mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1}$. ■

Remark VI.4: We believe that the analysis of the robustness to packet losses of the a-CL algorithm might be performed also in the randomized scenario considered in Section V assuming that each transmitted packet might be lost with a certain probability. We leave this analysis as future research. However, in the numerical section, specifically in Example VII.1, we show the effectiveness of the a-CL algorithm also in the presence of random communication failures when the network is randomly persistent communicating.

Remark VI.5: Also, the Jacobi-like strategy has been shown to be robust to packet losses, see [7]. Instead, concerning the other algorithms recently proposed in the literature, see [8] and [11]. To the best of our knowledge, no analysis considering the nonidealities introduced in this section has been proposed in the literature.

VII. Numerical Results

In this section, we provide some simulations implementing the localization consensus-based strategy introduced in this paper.

Example VII.1: In this example, we consider a random geometric graph generated by choosing $N = 100$ points randomly placed in the interval [0,1]. Two nodes are connected and take measurements if they are sufficiently close, that is, more specifically, measurements z_{ij} and z_{ji} are available provided that $|x_i - x_j| \leq 0.15$. This choice resulted in networks with an average number of neighbors per node of about 7. Every measurement was corrupted by Gaussian noise with covariance $\sigma^2 = 10^{-4}$. In this example, we assumed that the network is randomly persistent communicating with uniform communication probabilities $(\beta_1, \ldots, \beta_N)$, namely, $\beta_1 = \ldots = \beta_N = 1/N$. Moreover, the possibility of communication failure is taken into account. Specifically, supposing node i is transmitting, each node $j \in \mathcal{N}_i$ with a certain probability, that is, p_f, cannot receive the sent packet.

In Fig. 1, we plotted the behavior of the error

$$J(k) = \log\left(\|A\left(\hat{x}(k) - x^*\right)\|\right)$$

for different values of the failure probability p_f.

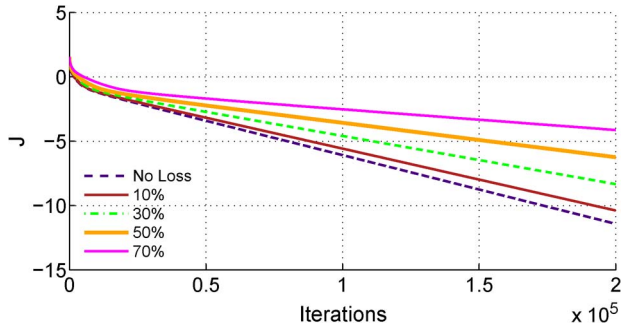

Fig. 1. Behavior of J for a randomly persistent communicating network in a random geometric graph, for different values of the probability failure p_f.

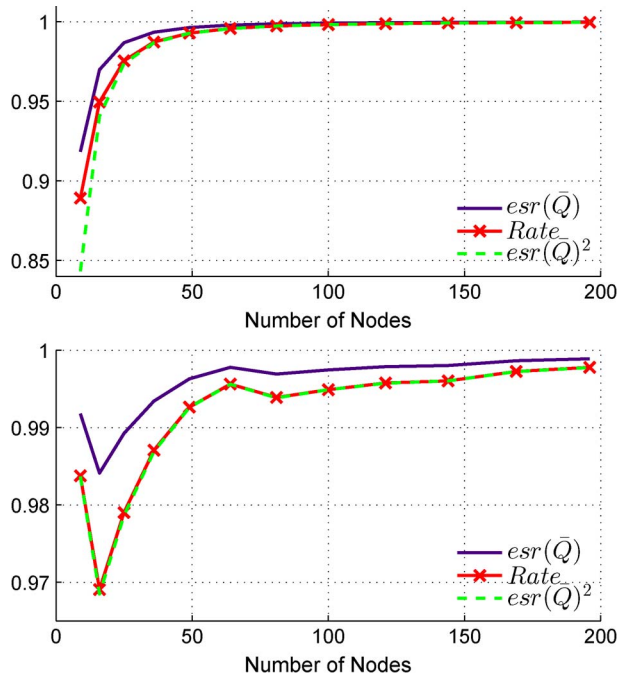

Fig. 2. Trend of the rate of convergence, of the $esr(\bar{Q})^2$, and of the $esr(\bar{Q})$ for $2-d$ torus (*top panel*) and for random geometric graphs (*bottom panel*) of increasing size N.

The plot reported is the result of the average of more than 1000 Monte Carlo runs, randomized with respect to the measurement graph[3] and the initial conditions. Observe that the trajectory of J decreases exponentially.

Example VII.2: In this example, we tested the validity of conject. V.2. In Fig. 2 (*top panel*) the simulation considering a set of 2-D torus graphs of increasing size N is shown, while in Fig. 2 (*bottom panel*) the simulation is performed considering a family of random geometric graphs. What we show is a comparison between the empirical rate of convergence of the algorithm, its lower bound, represented by $esr(\bar{Q})^2$, and the $esr(\bar{Q})$.

Example VII.3: In this example, we provide a numerical comparison with some well-known algorithms proposed in the literature which, for the sake of completeness, we briefly recall (Table I). The first algorithm considered, hereafter called

[3]In performing our average, we kept only the random geometric graphs which resulted in being connected.

TABLE I
ALGORITHMS COMPARISON

Algorithm	Sent packets per iteration
a-CL	1
a-GL	1
BC	$\lvert N_i\rvert + 1$
REK	$\lvert N_j\rvert + 5$

Number of sent packets per iteration for each algorithm.

a-GL algorithm, is proposed in [3]. Similar to the a-CL algorithm, during its kth iteration one node, say h transmits its variable $\hat{x}_h$ to all of its neighbors. For $l \in \mathcal{N}_h$, node l, based on the information received from node h, performs the following update:

$$\begin{aligned}\hat{x}_l(k+1) &= 1/2\,(\hat{x}_l(k) + \hat{x}_h(k) + 1/2(z_{lh} - z_{hl})) \\ &= \hat{x}_l(k) + 1/2\,(\hat{x}_l(k) - \hat{x}_h(k) + 1/2(z_{lh} - z_{hl}))\end{aligned}$$

while for $l \notin \mathcal{N}_h$, the state remains unchanged, that is, $\hat{x}_l(k+1) = \hat{x}_l(k)$. Note that just one packet is transmitted at each iteration. Moreover, since this algorithm is known to reach mean square convergence [9], then its ergodic mean has been proposed as a possible estimator of the state.

The second algorithm, denoted hereafter as the **BC algorithm**, is proposed in [8]. It requires a coordinated broadcast communication protocol meaning that during the kth iteration, one node, say h, asks the variable $\hat{x}_l$ to all of its neighbors $l \in \mathcal{N}_h$. When it receives the current state values, it performs the following greedy local optimization based on the current status of the network:

$$\begin{aligned}\hat{x}_h(k+1) &:= \arg\min_{\hat{x}_h} \sum_{(i,j)\in\mathcal{E}}^{|\mathcal{E}|} \|\hat{x}_i(k) - \hat{x}_j(k) - z_{ij}\|^2 \\ &= \frac{1}{2|\mathcal{N}_h|} \sum_{l\in\mathcal{N}_h} (2\hat{x}_l(k) - z_{lh} + z_{hl})\,.\end{aligned}$$

Note that the number of communications performed during one iteration are $|\mathcal{N}_h| + 1$, since there is a broadcast packet sent by node h, and $|\mathcal{N}_h|$ packets sent by all of its neighbors. We stress the fact that the Jacobi-like algorithm proposed [7] is indeed the same algorithm proposed in [8]. The last algorithm that we considered is the Randomized Extended Kaczmarz, hereafter called **REK algorithm**, presented in [11], consisting of two different update steps. The first step is an orthogonal projection of the noisy measurements onto the column space of the incidence matrix A in order to bound the measurements error. The second step is similar to the standard Kaczmarz update. Since a distributed implementation is not formally presented in [11], we propose the following algorithm. More specifically, let $s \in \mathbb{R}^M$ be the current projection of the noisy measurements onto the column space of A. Similarly as before, we denote with a little abuse of notation the eth entry of s with the corresponding edge, that is, $s_e = s_{ij}$. Then, the REK algorithm proposed in [11] for general least-squares problems, reduces in our setting to randomly and independently select a node h and an edge (i,j) at each iteration k according to the following probabilities:

$$p_h = \frac{|\mathcal{N}_h| + 1}{2M}; \quad p_{ij} = \frac{1}{M}$$

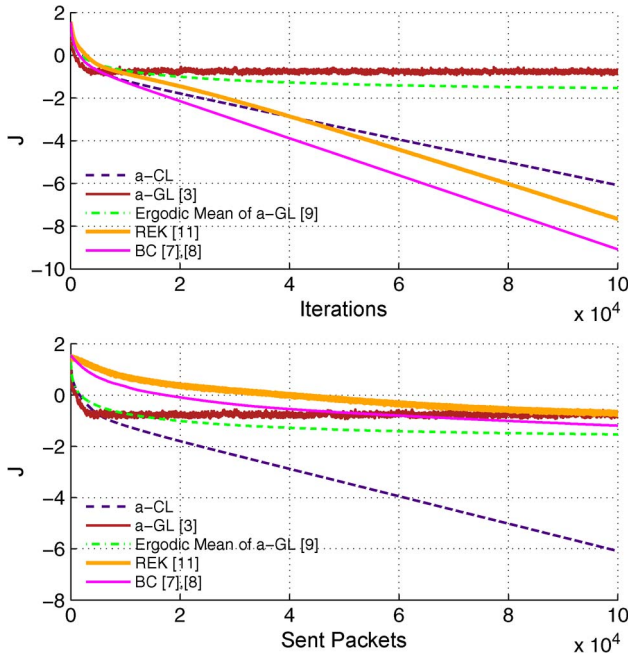

Fig. 3. Comparison of various algorithms considering the number of iteration (*top panel*) and number of sent packets (*bottom panel*).

and then to perform the following local updates:

$$\begin{aligned}s_{\ell h}(k+1) &= s_{\ell h}(k) + \frac{\sum_{m\in\mathcal{N}_h}(s_{hm}(k) - s_{mh}(k))}{|\mathcal{N}_h| + 1},\ \forall \ell \in \mathcal{N}_h \\ s_{h\ell}(k+1) &= s_{h\ell}(k) - \frac{\sum_{m\in\mathcal{N}_h}(s_{hm}(k) - s_{mh}(k))}{|\mathcal{N}_h| + 1},\ \forall \ell \in \mathcal{N}_h \\ \hat{x}_i(k+1) &= \hat{x}_i(k) + \frac{z_{ij} - s_{ij}(k) - (\hat{x}_i(k) - \hat{x}_j(k))}{2} \\ \hat{x}_j(k+1) &= \hat{x}_j(k) - \frac{z_{ij} - s_{ij}(k) - (\hat{x}_i(k) - \hat{x}_j(k))}{2}.\end{aligned}$$

We point out that since in the updating step only local information is required, the algorithm is implemented in a distributed fashion and it exactly requires $|N_j| + 5$ communication rounds to perform an iteration. Specifically, the first $|N_j| + 2$ are due to the update of the variable s and the last three are needed to update $\hat{x}$.

In this example, we consider a random geometric measurement graph $\mathcal{G}$ built as in the previous example. In Fig. 3, we plot the behavior of J with respect to the number of iterations and sent packets. From these simulations, we observe that from an energy point of view, the a-CL algorithm is the most convenient since the effective number of sent packets to achieve a certain estimation error is lower. On the other hand, if no energy constraint is imposed, then BC is the fastest algorithm, although not substantially faster than REK and a-CL.

As observed in [3], the local estimates of the a-GL algorithm do not converge to the optimal solution, but they oscillate around it. However, a-GL exhibits the fastest transient among all algorithms and it is also energetically efficient. In our recent work, we thus propose combining the a-CL algorithm with the a-GL algorithm in order to have fast transients as well as guaranteed exponential asymptotic convergence by using suitable switching strategies [24].

VIII. Conclusion

In this paper, we introduced a consensus-based strategy to solve the problem of optimally estimating the position of each agent in a network from relative noisy vector distances with its neighbors. We first introduced the algorithm in its synchronous version showing that it exponentially converges to the optimal centralized least-squares solution. We then proposed a more realistic asynchronous implementation which is still shown to be exponentially convergent under simple communication protocols. In the randomized scenario, we performed a theoretical analysis of the rate of convergence in mean square, providing general lower and upper bounds. A more detailed analysis has been performed that is restricted to communication graphs which are regular graphs. In addition, we showed that our novel asynchronous algorithm is robust to packet losses and delays. Finally, through numerical simulations, we tested the effectiveness of our approach compared to other strategies recently proposed in the literature.

Appendix
Consensus Protocols in the Presence of Delays

In this appendix, we review the result stated in [23, Prop. 1]. In [23], the authors consider the following consensus algorithm with delays[4]

$$x^i(k+1) = \sum_{j=1}^{m} a_j^i(k) x^j\left(k - t_j^i(k)\right) \tag{17}$$

where x^i denotes the state of node i, $i \in \{1, \ldots, M\}$, the scalar $t_j^i(k)$ is non-negative and it represents the delay of a message from agent j to agent i, while the scalar $a_j^i(k)$ is a non-negative weight that agent i assigns to a delayed estimate $x^j(s)$ arriving from agent j at time k. It is assumed that the weights $a_j^i(k)$ satisfy the following assumption.

Assumption A.1: There exists a scalar η, $0 < \eta < 1$ such that

i) $a_i^i(k) \geq \eta$ for all $k \geq 0$;
ii) $a_j^i(k) \geq \eta$ for all $k \geq 0$, and all agents j whose (potentially delayed) information $x^j(s)$ reaches agent i during the kth iteration;
iii) $a_j^i(k) = 0$ for all $k \geq 0$ and j otherwise;
iv) $\sum_{j=1}^{m} a_j^i(k) = 1$ for all i and k.

For any k, let the information exchange among the agents may be represented by a directed graph $(\mathcal{V}, \mathcal{E}_k)$, where $V = \{1, \ldots, m\}$ with the set $\mathcal{E}_k$ of directed edges given by $\mathcal{E}_k = \{(j,i) | a_j^i(k) > 0\}$. The authors impose a connectivity assumption on the agent system, which is stated as follows.

Assumption A.2: The graph $(V, \mathcal{E}_\infty)$ is connected, where $\mathcal{E}_\infty$ is the set of edges (j,i) representing agent pairs communicating directly infinitely many times, that is, $\mathcal{E}_\infty = \{(j,i) | (j,i) \in \mathcal{E}_k \text{ for infinitely many indices } k\}$.

In addition, it is assumed that the intercommunication intervals are bounded for those agents that communicate directly. Specifically:

[4]We adopt the notations of paper [23].

Assumption A.3: There exists an integer $B \geq 1$ such that for every $(j,i) \in \mathcal{E}_\infty$, agent j sends information to its neighbor i at least once every B consecutive iterations.

Finally, it is assumed that the delays $t_j^i(k)$ in delivering a message from an agent j to any neighboring agent i are uniformly bounded at all times. Formally:

Assumption A.4: Let the following hold:

i) $t_i^i(k) = 0$ for all agents i and all $k \geq 0$.
ii) $t_j^i(k) = 0$ for all agents j communicating with agent i directly and whose estimates x^j are not available to agent i during the kth iteration.
iii) There is an integer B_1 such that $0 \leq t_j^i(k) \leq B_1 - 1$ for all agents i, j, and all k.

The result illustrated in [23, Prop. 1] is recalled in the following proposition.

Proposition A.5: Let Assumption A.1, A.2, A.3, and A.4 hold. Then, the sequences $\{x^{(i)}(k)\}$, $i = 1, \ldots, m$, generated by (17) converge exponentially to a consensus.

References

[1] P. Barooah and J. P. Hespanha, "Distributed estimation from relative measurements in sensor networks," in *Proc. 2nd Int. Conf. Intell. Sens. Inf. Process.*, Dec. 2005, pp. 226–231.
[2] P. Barooah, "Estimation and control with relative measurements: algorithms and scaling laws," Ph.D. dissertation, Department of Electrical and Computer Engineering, Univ. California, Santa Barbara, CA, USA, 2007.
[3] S. Bolognani, S. D. Favero, L. Schenato, and D. Varagnolo, "Consensus-based distributed sensor calibration and least-square parameter identification in WSNs," *Int. J. Robust Nonlinear Control*, vol. 20, no. 2, pp. 176–193, 2010.
[4] R. Solis, V. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proc. IEEE 45th Conf. Dec. Control*, San Diego, CA, USA, Dec. 2006, pp. 2734–2739.
[5] D. Borra, E. Lovisari, R. Carli, F. Fagnani, and S. Zampieri, "Autonomous calibration algorithms for networks of cameras," in *Proc. Amer. Control Conf.*, Jul. 2012, pp. 5126–5131.
[6] R. Tron and R. Vidal, "Distributed image-based 3-d localization of camera sensor networks," in *Proc. IEEE 49th Conf. Dec. Control*, 2009, pp. 901–908.
[7] P. Barooah and J. Hespanha, "Distributed estimation from relative measurements in sensor networks," in *Proc. 3rd Int. Conf. Intell. Sens. Inf. Process.*, 2005, pp. 226–231.
[8] A. Giridhar and P. Kumar, "Distributed clock synchronization over wireless networks: Algorithms and analysis," in *Proc. IEEE 45th Dec. Control Conf.*, 2006, pp. 4915–4920.
[9] W. Rossi, P. Frasca, and F. Fagnani, "Transient and limit performance of distributed relative localization," in *Proc. IEEE 51st Annu. Conf. Dec. Control* , 2012, pp. 2744–2748.
[10] C. Ravazzi, P. Frasca, H. Ishii, and R. Tempo, "A distributed randomized algorithm for relative localization in sensor networks," in *Proc. Eur. Control Conf.*, 2013, pp. 1776–1781.
[11] A. Zouzias and N. Freris, "Randomized extended Kaczmarz for solving least-squares," Univ. Toronto, Toronto, ON, Canada, 2013, Tech. Rep.
[12] J. Liu, B. Anderson, M. Cao, and A. S. Morse, "Analysis of accelerated gossip algorithms," in *Proc. IEEE Dec. Control Conf.*, 2009, pp. 871–876.
[13] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *Automatica*, vol. 44, no. 3, pp. 671–684, 2008.
[14] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–913, Jan. 2007.
[15] F. Garin and L. Schenato, "A survey on distributed estimation and control applications using linear consensus algorithms," in *Networked Control Systems*, vol. 406. New York, USA: Springer, 2011, ser. Lect. Notes Control Inf. Sci., pp. 75–107.
[16] P. Barooah and J. Hespanha, "Estimation on graphs from relative measurements: Distributed algorithms and fundamental limith," *IEEE Control Syst. Mag.*, vol. 27, no. 4, pp. 57–74, Aug. 2007.

[17] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 634–649, May 2008.
[18] R. Carli, A. Carron, L. Schenato, and M. Todescato, An exponential-rate consensus-based algorithms for estimation from relative measurements: implementation and performance analysis, University of Padova, Padova, Italy, Tech. Rep. [Online]. Available: http://automatica.dei.unipd.it/tl_files/utenti2/carron/Articoli/consensus-based_localization_tech_note.pdf
[19] S. Boyd, A. Ghosh, D. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
[20] M. Franceschetti and R. Meester, Eds., *Random Networks for Communication*. Cambridge, U.K.: Cambridge University Press, 2007.
[21] F. Fagnani and P. Frasca, "Broadcast gossip averaging: Interference and unbiasedness in large Abelian Cayley networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 866–875, Aug. 2011.
[22] M. R. Murthy, "Ramanujan graphs," *J. Ramanujan Math. Soc.*, vol. 18, no. 6, pp. 1–20, 2003.
[23] A. Nedic and A. Ozdaglar, "Convergence rate for consensus with delays," *J. Global Optimiz.*, vol. 47, no. 3, pp. 437–456, 2008.
[24] A. Carron, M. Todescato, R. Carli, and L. Schenato, "Adaptive consensus-based algorithms for fast estimation from relative measurements," in *Proc. 4th IFAC Workshop Distrib. Estimation Control Netw. Syst.*, 2013, pp. 234–239.

Andrea Carron was born in Padova, Italy, in 1988. He received the Dr. Eng. Bachelor's and Master's degrees in control engineering from the University of Padova, Padova, Italy, in 2010 and 2012, respectively, where he is currently pursuing the Ph.D. degree.

Since 2013, he has been with the Department of Information Engineering, University of Padova. His interests include multiagent robotics, distributed optimization, and nonparametric estimation.

Marco Todescato was born in Vicenza, Italy, in 1988. He received the Dr. Eng. Bachelor's degree in information engineering and the Dr. Eng. Master's degree in control engineering from the University of Padova, Padova, Italy, in 2010 and 2012, respectively, where he is currently pursuing the Ph.D. degree.

Since 2013, he has been with the Department of Information Engineering, University of Padova. His interests include smart grid, distributed optimization, and nonparametric estimation.

Ruggero Carli received the Laurea Degree in computer engineering and the Ph.D. degree in information engineering from the University of Padova, Padova, Italy, in 2004 and 2007, respectively.

From 2008 through 2010, he was a Postdoctoral Fellow with the Department of Mechanical Engineering, University of California at Santa Barbara, Santa Barbara, CA, USA. Currently, he is an Assistant Professor with the Department of Information Engineering, University of Padova. His research interests include control theory and, in particular, control under communication constraints, cooperative control, and distributed estimation. He serves as a reviewer for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON SIGNAL PROCESSING, *Automatica, Systems and Control Letters*, *SIAM Journal on Control and Optimization*, and *International Journal of Robust and Nonlinear Control*.

Luca Schenato (M'07) received the Laurea degree from the University of Padova, Padova, Italy, in 1999 and the Ph.D. degree in electrical engineering from the University of California at Berkeley in 2003.

Currently, he is an Associate Professor with the University of Padova. He currently holds a visiting position at University of California at Berkeley and was Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL. His research interests include distributed control, estimation and optimization for multiagent systems, control subject to communication constraints in networked control systems, and the control of biomimetic locomotion.

Prof. Schenato was the recipient of the Italian Professorship "Returning Brains" in 2004, the Eli Jury Award from the EECS Department of UC Berkeley in 2006, and of the EUCA European Control Award in 2014. He was one of the organizers of the Conference on Robotics, Communication and Coordination (Robocomm) and the Workshop on Estimation and Control of Networked Systems (Necsys). He is a member of the IFAC Technical Committee on "Networked Systems."