An interpolatory algorithm for distributed set membership estimation in asynchronous networks

Francesco Farina, *Member, IEEE,* Andrea Garulli, *Fellow, IEEE,* Antonio Giannitrapani, *Member, IEEE*

***Abstract*—This paper addresses distributed estimation problems over asynchronous networks in a set membership framework. The agents in the network asynchronously collect and process measurements, communicate over a possibly time-varying and unbalanced directed graph and may have non negligible computation times. Measurements are affected by bounded errors, so that they define feasible sets containing the unknown parameters to be estimated. The proposed algorithm requires each agent to compute a weighted average of its estimate and those of its neighbors and to project it onto a local feasible set. By assuming convexity of the measurement sets, the local estimates are shown to converge to a common point belonging to the global feasible set.**

***Index Terms*—Set membership estimation, Distributed estimation**

I. Introduction

DISTRIBUTED estimation is receiving increasing attention in recent years. Motivated by the widespread diffusion of sensor networks and, more generally, of networks of agents equipped with sensing, computing and communication capabilities, researchers have proposed a number of techniques allowing the agents to cooperate within a common estimation task. A wide variety of solutions is now available for many different problems and settings (see [1]–[7] and references therein). These approaches consider different types of underlying graph topology, but they typically refer to synchronous networks. The more realistic setting of asynchronous networks has also received considerable attention, leading to several different approaches for tackling distributed optimization [8]–[13] and consensus [14].

Although most contributions on distributed estimation have been proposed in a stochastic framework, significant attention has been devoted to deterministic approaches as well, like set membership estimation. The set membership (or bounded error) framework assumes that the measurements provided by the sensors are affected by bounded noise (see e.g. [15], [16]). This implies that the quantities to be estimated (the *parameter* vector) can be constrained within the so-called *feasible measurement set*. The overall knowledge available to the network about such quantities is summarized by the intersection of all the measurement sets, usually referred to as *feasible parameter set*, which quantifies the uncertainty associated with the estimate. This has led to the development of a number of set membership estimation techniques and applications in different contexts [17]–[22]. In particular, there is a vast literature on the application of the set membership estimation paradigm in the fields of signal processing and sensor networks (see e.g. [23], [24]). Set membership algorithms based on affine projections are particularly appealing to perform adaptive filtering with power constrained devices such as mobile stations [25]. In [26], the set-membership framework has been proven to reduce both the information exchange and the computational burden in distributed parameter estimation performed by wireless sensor networks. In this context, the error bound usually plays the role of a tuning parameter, to trade-off estimation performance and convergence speed. In [27], for example, it is used to guarantee that the computational load is uniformly distributed among the nodes of network. Distributed set-valued estimators for networks of sensors affected by bounded noise have been proposed in [28], [29].

In a distributed estimation setting, each agent can perform the intersection only of the local measurement sets, thus achieving a conservative knowledge of the unknown parameter. Clearly, uncertainty can be reduced by sharing such knowledge with other nodes of the network. On the other hand, sharing the entire information on the local sets with the neighboring nodes may dramatically increase the communication and computation load. A feasible compromise consists in sharing only pointwise local estimates, obtained by computing a weighted average of the neighbor estimates and then projecting it onto the local feasible set. Such an approach has been proposed in [30] for a synchronous static network, showing that all the node local estimates converge to a common vector, belonging to the global feasible parameter set.

In this note, a more challenging setting is considered with respect to the one in [30]. First, measurements are collected and processed in a fully asynchronous fashion by the network nodes. Moreover, the communication graph has a time-varying topology, accounting for temporary node unavailability or transmission failures. The main result shows that the node estimates converge to the same vector and are *asymptotically interpolatory*, i.e., their distance to the feasible parameter set tends to zero. This is achieved under mild assumptions on the network topology: directed unbalanced graphs are considered and it is sufficient that the sequence of graphs resulting form the asynchronous measurement processing are jointly

† F. Farina is with the Artificial Intelligence and Machine Learning group, GlaxoSmithKline, TW8 9GS, London, United Kingdom. This work was carried out while the author was at the Department of Electrical, Electronic and Information Engineering, Università di Bologna, 40136 Bologna, Italy. e-mail: francesco.x.farina@gsk.com

‡A. Garulli and A. Giannitrapni are with the Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università di Siena, Siena, Italy. e-mail: {garulli,giannitrapani}@diism.unisi.it

This work was supported in part by the Italian Ministry for Research in the framework of the 2017 Program for Research Projects of National Interest (PRIN) under Grant 2017YKXYXJ.

strongly connected over finite time intervals. The proposed approach can be seen as an extension of the well-studied constrained consensus problem [31]–[33], in which each node performs projections indefinitely over the same local set. On the contrary, in the framework considered in this paper, an infinite sequence of different constraint sets is generated in real time, corresponding to the measurements collected by the sensors. A preliminary version of this work has been presented in [34]. The main novelties of this contribution with respect to [34] are a rigorous formulation and analysis of asynchronous implementation of the proposed algorithm for general time-varying graphs, and the analysis of the effect of computation times.

The paper is organized as follows. In Section II, the considered set membership estimation framework and the proposed asynchronous distributed algorithm are introduced. The main contribution of the paper is presented in Section III, where it is shown that the proposed algorithm generates sequences of estimates converging to a common point of the feasible parameter set. The case of a stationary underlying graph is addressed in Section IV, while the effect of non-negligible computation times is accounted for in Section V. A numerical example is reported in Section VI and some conclusions are drawn in Section VII.

Notation: Given a matrix W we denote by w_{ij} the element in the i-th row and j-th column. Given a point $p \in \mathbb{R}^n$ and a closed set $Z \subset \mathbb{R}^n$, we denote by $P_Z[p]$ the projection of p on Z, defined as $P_Z[p] = \arg\min_{z\in Z} \|p - z\|$, where $\|\cdot\|$ denotes the 2-norm in $\mathbb{R}^n$.

II. Asynchronous set membership estimation

Consider a network of N peer nodes which collaborate in order to estimate an unknown parameter $x \in \mathbb{R}^n$. The structure of the network at time $t \in \mathbb{R}_+$ is modeled by means of a weighted directed graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t), W(t))$, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of nodes, $\mathcal{E}(t) \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges and $W(t) \in \mathbb{R}^{N\times N}$ is the associated weighted adjacency matrix. If $(j, i) \in \mathcal{E}(t)$, then node j communicates to node i at time t (we say that j is a neighbor of i at time t) and $w_{ij}(t) > 0$. Conversely, if $(j, i) \notin \mathcal{E}(t)$, necessarily $w_{ij}(t) = 0$. We denote by $\mathcal{N}_i(t) = \{j \mid (j, i) \in \mathcal{E}(t)\} \cup \{i\}$ the set of neighbors of node i at time t (including node i itself).

We consider a fully asynchronous setting in which agents compute and communicate independently without any coordination. When a local timer triggers, node i wakes up, performs computations and, then, returns in idle mode. For the sake of presentation, in Sections II and III it is assumed that the local computation times are negligible. However, in Section V, it will be shown how to deal with finite computation times.

In order to estimate the unknown parameter x, each node i, when awake, takes a noisy measurement $y_i \in \mathbb{R}^{p_i}$ of a known function of x. Let us call $t_h^{(i)} \in \mathbb{R}_+$ the time instant at which node i wakes up for the h-th time. Then, the measurement equation is given by

$$y_i(t_h^{(i)}) = m_i(x) + \varepsilon_i(t_h^{(i)}), \tag{1}$$

where $m_i(x)$ is the noise-free measurement function of agent i and the measurement noise ε_i is assumed to be unknown-but-bounded (UBB), i.e.

$$\varepsilon_i(t_h^{(i)}) \in \mathcal{S}_i(t_h^{(i)}) \tag{2}$$

for all $h \geq 0$. The set $\mathcal{S}_i(t_h^{(i)}) \subset \mathbb{R}^{p_i}$ is assumed to be a known (possibly time-varying) compact set. Under the UBB assumption, the measurement taken at time $t_h^{(i)}$ by node i produces a *feasible measurement set*

$$\mathcal{M}_i(t_h^{(i)}) = \{x \in \mathbb{R}^n \mid \Big(y_i(t_h^{(i)}) - m_i(x)\Big) \in \mathcal{S}_i(t_h^{(i)})\}, \tag{3}$$

which contains all the possible values of the unknown parameter x that are compatible with the collected measurement. Consistently, let $\mathcal{X}_i(t_h^{(i)})$ be the *feasible parameter set* of node i containing the values of x that are compatible with the measurements taken by node i up to time $t_h^{(i)}$, i.e.,

$$\mathcal{X}_i(t_h^{(i)}) = \bigcap_{\tau=1}^{h} \mathcal{M}_i(t_\tau^{(i)}).$$

By construction, for each i, the sequence $\{\mathcal{X}_i(t_h^{(i)})\}$, $h = 1, 2, \ldots$, is a *nonincreasing sequence of sets*. Hence, according to the definition of limit of a sequence of sets [35], it converges to the set

$$\mathcal{X}_i = \lim_{h\to\infty} \mathcal{X}_i(t_h^{(i)}). \tag{4}$$

The nodes aim at cooperatively finding an estimate of the unknown parameter which is compatible with all the measurements they collect, i.e. an estimate belonging to the *global asymptotic feasible parameter set*

$$\mathcal{X} = \bigcap_{i=1}^{N} \mathcal{X}_i. \tag{5}$$

In order to solve such a problem, we propose an asynchronous version of a distributed algorithm studied in [30], in which the nodes produce sequences of estimates eventually converging to the global asymptotic feasible parameter set $\mathcal{X}$. The algorithm works as follows. Each node i in the network maintains a local state variable x_i which represents its local estimate of x. When waking up at time $t_h^{(i)}$, node i takes a new noisy measurement of x and produces a feasible measurement set $\mathcal{M}_i(t_h^{(i)})$. Then, it reads the estimates x_j of its neighbors and computes a weighted average z_i of them (including its own one), according to the weights $w_{ij}(t_h^{(i)})$. Finally, it projects z_i on the current feasible parameter set $\mathcal{X}_i(t_h^{(i)})$. It is assumed that when a new estimate is computed, it is immediately made available to all neighbors (including those not awake), which can collect it by means of a buffering mechanism. In this way, when a node awakes it can read the most recent estimates of all its neighbors from the corresponding buffer. A pseudocode of the algorithm is reported in Algorithm 1.

In the following, the convergence of the Asynchronous Distributed Interpolatory Algorithm is studied by recasting it as a synchronous algorithm on a suitable time-varying graph. From a global perspective, it is possible to associate an iteration $k \in \mathbb{N}$ to each awakening in the network. Let $\mathcal{G}(k)$ denote the communication graph at iteration k. Since,

Algorithm 1 Asynchronous Distributed Interpolatory Algorithm (node i)

Initialization: x_i, $\mathcal{X}_i(t_0^{(i)}) = \mathbb{R}^n$

Iteration h (time $t_h^{(i)}$)

TAKE MEASUREMENT $y_i(t_h^{(i)})$ and produce $\mathcal{M}_i(t_h^{(i)})$
READ x_j for all $j \in \mathcal{N}_i(t_h^{(i)})$
UPDATE

$$\mathcal{X}_i(t_h^{(i)}) \leftarrow \mathcal{X}_i(t_{h-1}^{(i)}) \cap \mathcal{M}_i(t_h^{(i)}) \tag{6a}$$

$$z_i \leftarrow \sum_{j \in \mathcal{N}_i(t_h^{(i)})} w_{ij}(t_h^{(i)}) x_j \tag{6b}$$

$$x_i \leftarrow P_{\mathcal{X}_i(t_h^{(i)})}[z_i] \tag{6c}$$

in principle, it is possible for two or more nodes to awake exactly at the same time instant, let us denote by $\mathcal{A}(k)$ the set of nodes which are awake at iteration k. Moreover, we call $x_i(k)$ the value of the local state variable of node i at iteration k. Let us define

$$M_i(k) = \begin{cases} \mathbb{R}^n, & \text{if } i \notin \mathcal{A}(k) \\ \mathcal{M}_i(t_{\kappa_i(k)}^{(i)}), & \text{otherwise} \end{cases} \tag{7}$$

where $\kappa_i(k)$ denotes the number of times that node i has been awake in the first k iterations. Similarly, let the weights at iteration k satisfy

$$w_{ij}(k) = \begin{cases} 1, & \text{if } i \notin \mathcal{A}(k) \text{ and } i = j \\ 0, & \text{if } i \notin \mathcal{A}(k) \text{ and } i \neq j \\ w_{ij}(t_{\kappa_i(k)}^{(i)}) & \text{if } i \in \mathcal{A}(k). \end{cases} \tag{8}$$

Then, Algorithm 1 can be rewritten as the following equivalent synchronous algorithm

$$X_i(k+1) = X_i(k) \cap M_i(k+1), \tag{9a}$$

$$z_i(k) = \sum_{j=1}^{N} w_{ij}(k) x_j(k), \tag{9b}$$

$$x_i(k+1) = P_{X_i(k+1)}[z_i(k)], \tag{9c}$$

for all i and all $k \geq 0$, with $X_i(0) = \mathbb{R}^n$. Notice that (7) and (8) ensure that if a node is not awake at iteration k, its estimate and its feasible parameter set are not changed when running the k-th iteration of the synchronous algorithm (9). Moreover, if we define

$$X_i = \lim_{k \to \infty} X_i(k), \tag{10}$$

$$X = \bigcap_{i=1}^{N} X_i, \tag{11}$$

we have that $X_i = \mathcal{X}_i$ and hence the global asymptotic feasible parameter set corresponds to the intersection of all the sets X_i, i.e. $X = \mathcal{X}$. In order to prove that the estimates generated by Algorithm 1 converge to a common point lying in the global feasible parameter set $\mathcal{X}$, in the following we will show that all the sequences of estimates $\{x_i(k)\}$ generated by the synchronous algorithm (9) converge to a common point lying in X.

III. CONVERGENCE ANALYSIS

The following assumptions are made on the local measurement sets, the global feasible set, the communication graphs and the weights.

Assumption 1. *The feasible measurement sets $M_i(k)$, $k = 1, 2, \ldots,$ are closed convex sets.* □

Assumption 2. *The global asymptotic feasible parameter set X is not empty.* □

Assumption 3. *There exists a $K > 0$ and an infinite sequence of iteration indexes $\{k_l \in \mathbb{N}\}$, $l = 0, 1, \ldots,$ such that for all l: i) $k_{l+1} - k_l < K$, and ii) the union of graphs $\mathcal{G}(k_l)$, $\mathcal{G}(k_l + 1), \ldots,$ $\mathcal{G}(k_{l+1} - 1)$ is strongly connected.* □

Assumption 4. *For all $i, j = 1, \ldots, N$, and for all k, the weights $w_{ij}(k) \geq 0$ in* (9b) *satisfy*

- *$w_{ii}(k) > 0$;*
- *if $i \neq j$, $w_{ij}(k) > 0$ if and only if $(j, i) \in \mathcal{E}(k)$;*
- *$\sum_{j=1}^{N} w_{ij}(k) = 1$;*
- *if $w_{ij}(k) > 0$, then $w_{ij}(k) > \eta$, for some $\eta > 0$.* □

Under Assumption 1, the local feasible parameter sets $X_i(k)$, as well as their limit sets X_i, are closed convex sets, being the intersection of (possibly infinitely many) closed convex sets. In the considered framework, Assumption 2 is always satisfied if the measurement noise does not violate the UBB constraints (2). Assumption 3 requires that the union of the communication graphs is strongly connected over finite time intervals. In particular, this assumption rules out the possibility that the network eventually becomes disconnected or that a node remains idle indefinitely. Assumption 4 ensures that the weights are compliant with the network topology and none of them vanishes over time. Assumptions 3 and 4 are typically made to ensure consensus achievement in the presence of time-varying graph topology (e.g., see [32], [33]).

The following lemmas, whose proof can be found in [30], are instrumental to prove the convergence of the estimates generated by algorithm (9).

Lemma 1. *Let $\{Z(k)\}$, $k = 1, 2, \ldots,$ be a nonincreasing sequence of closed convex subsets of $\mathbb{R}^n$ and denote by Z its limit. Consider an arbitrary point $p \in \mathbb{R}^n$ and let $q(k) = P_{Z(k)}[p]$ be its projection on $Z(k)$. Then, for any $z \in Z$ and $k = 1, 2, \ldots,$ it holds*

$$(p - q(k))^\top (z - q(k)) \leq 0. \tag{12}$$

□

Lemma 2. *Let $\{Z(k)\}$, $k = 1, 2, \ldots,$ be a nonincreasing sequence of closed convex subsets of $\mathbb{R}^n$ and denote by Z its limit. Let $\{z(k) \in Z(k)\}$ be a sequence admitting a convergent subsequence $\{z(k_j)\}$, $j = 1, 2, \ldots,$ to a point $\hat{z}$. Then, $\hat{z} \in Z$.* □

By defining

$$u_i(k) = P_{X_i(k+1)}[z_i(k)] - z_i(k), \tag{13}$$

expressions (9b)-(9c) can be written as

$$x_i(k+1) = \sum_{i=1}^{N} w_{ij}(k) x_j(k) + u_i(k), \tag{14}$$

for $i = 1, \ldots, N$. Denote by

$$\Psi(r, s) = W(r-1) \ldots W(s+1) W(s), \quad r > s \tag{15}$$

be the state transition matrix of system (14) from iteration s to iteration r, with the convention $\Psi(r, r) = I$. The following lemma summarizes some results related to stochastic, irreducible and aperiodic matrices.

Lemma 3. *Let Assumptions 3 and 4 hold. Let $\{k_l\}$ be a sequence of iterations defined as in Assumption 3. Then, there exist a scalar $\delta > 0$ and an absolute probability sequence $\{v(k_l)\}$ such that for $i = 1, \ldots, N$ and all k_l:*

- $v_i(k_l) > \delta$
- $\sum_{i=1}^{N} v_i(k_l) = 1$
- $\sum_{i=1}^{N} v_i(k_{l+1}) \Psi_{ij}(k_{l+1}, k_l) = v_j(k_l)$

where $\Psi(k_{l+1}, k_l)$ is defined in (15)*.*

Proof. Under Assumptions 3 and 4, $\Psi(k_{l+1}, k_l)$ is a row-stochastic, irreducible and aperiodic matrix, for all k_l. Hence, $\{\Psi(k_{l+1}, k_l)\}$, $l = 0, 1, \ldots$, is a sequence of matrices admitting an absolute probability sequence $\{v(k_l)\}$

$$v^\top(k_l) = v^\top(k_{l+1}) \Psi(k_{l+1}, k_l) \tag{16}$$

satisfying $v_i(k_l) > \delta$, for all $i = 1, \ldots, N$ and all k_l, for some $\delta > 0$ (see [33]). □

Lemma 4. *Let Assumptions 1-4 hold. Let $u_i(k)$, $i = 1, \ldots, N$, $k = 1, 2, \ldots$, be defined as in* (13)*, where $z_i(k)$ are computed according to* (9)*. Then,*

$$\lim_{k \to \infty} \|u_i(k)\| = 0, \quad i = 1, \ldots, N.$$

Proof. Let $\{k_l\}$ be a sequence of iterations defined as in Assumption 3 and let $\bar{x}$ be any point in X. From (9b),(9c), one has that for $i = 1, \ldots, N$

$$\begin{aligned} \|x_i(k_{l+1}) - \bar{x}\|^2 &\leq \|z_i(k_{l+1}-1) - \bar{x}\|^2 \\ &\leq \sum_{j=1}^{N} w_{ij}(k_{l+1}-1) \|x_j(k_{l+1}-1) - \bar{x}\|^2, \end{aligned} \tag{17}$$

where the first inequality comes from the properties of projections on convex sets and the second one from Jensen's inequality. By iterating (17) backwards, and recalling the definition of $\Psi(k_{l+1}, k_l)$ in (15), one gets

$$\begin{aligned} \|x_i(k_{l+1}) - \bar{x}\|^2 &\leq \|z_i(k_{l+1}-1) - \bar{x}\|^2 \\ &\leq \sum_{j=1}^{N} \Psi_{ij}(k_{l+1}, k_l) \|x_j(k_l) - \bar{x}\|^2. \end{aligned} \tag{18}$$

Let the vectors $v(k_l) \in \mathbb{R}^N$ be defined as in Lemma 3. Inequality (18) implies that

$$\begin{aligned} &\sum_{i=1}^{N} v_i(k_{l+1}) \|x_i(k_{l+1}) - \bar{x}\|^2 \\ &\quad \leq \sum_{i=1}^{N} v_i(k_{l+1}) \sum_{j=1}^{N} \Psi_{ij}(k_{l+1}, k_l) \|(x_j(k_l) - \bar{x})\|^2 \\ &\quad = \sum_{i=1}^{N} v_i(k_l) \|(x_i(k_l) - \bar{x})\|^2, \end{aligned} \tag{19}$$

where the last equality comes from Lemma 3. By (19), for $i = 1, \ldots, N$, the sequence

$$\left\{ \sum_{i=1}^{N} v_i(k_j) ||x_j(k_j) - \bar{x}||^2, \sum_{i=1}^{N} v_i(k_{j+1}) ||z_i(k_{j+1}-1) - \bar{x}||^2 \right\}_j$$

is bounded and nonincreasing. Hence, it converges to some $\bar{d} \geq 0$, i.e.

$$\begin{aligned} &\lim_{l \to \infty} \sum_{i=1}^{N} v_i(k_l) \|x_i(k_l) - \bar{x}\|^2 \\ &\quad = \lim_{l \to \infty} \sum_{i=1}^{N} v_i(k_l) \|z_i(k_l - 1) - \bar{x}\|^2 = \bar{d}. \end{aligned} \tag{20}$$

Moreover, from (9c) one has

$$\begin{aligned} &\|z_i(k_{l+1}-1) - \bar{x}\|^2 \\ &= \|z_i(k_{l+1}-1) - x_i(k_{l+1}) + x_i(k_{l+1}) - \bar{x}\|^2 \\ &= \|z_i(k_{l+1}-1) - x_i(k_{l+1})\|^2 + \|x_i(k_{l+1}) - \bar{x}\|^2 \\ &\quad + 2(z_i(k_{l+1}-1) - x_i(k_{l+1}))^\top (x_i(k_{l+1}) - \bar{x}) \\ &\geq \|z_i(k_{l+1}-1) - x_i(k_{l+1})\|^2 + \|x_i(k_{l+1}) - \bar{x}\|^2, \end{aligned}$$

where the inequality follows from Lemma 1. Hence,

$$\begin{aligned} \|x_i(k_{l+1}) - \bar{x}\|^2 \leq &\|z_i(k_{l+1}-1) - \bar{x}\|^2 \\ &- \|x_i(k_{l+1}) - z_i(k_{l+1}-1)\|^2. \end{aligned} \tag{21}$$

By multiplying (21) by $v_i(k_{l+1})$, summing over i and exploiting (20), it follows

$$\lim_{l \to \infty} \sum_{i=1}^{N} v_i(k_{l+1}) \|x_i(k_{l+1}) - z_i(k_{l+1}-1)\|^2 = 0,$$

which, being $v_i(k_{l+1}) > \delta > 0$ from Lemma 3, implies that

$$\begin{aligned} &\lim_{l \to \infty} \|x_i(k_{l+1}) - z_i(k_{l+1}-1)\|^2 \\ &\quad = \lim_{l \to \infty} \|u_i(k_{l+1}-1)\|^2 = 0. \end{aligned} \tag{22}$$

So far, we have proved that each subsequence $\|u_i(k_l - 1)\|$, $i = 1, \ldots, N$, converges to zero. To show that the whole sequence $\{u_i(k)\}$ actually goes to zero, we notice that the set of all $k \geq k_0$ can be obtained as the union of $2K$ subsequences of the form $k_h^{(q)} = k_0 + q + 2Kh$, for $q = 0, \ldots, 2K-1$ and $h = 0, 1, \ldots$. By construction, each subsequence $\{k_h^{(q)}\}$ has the same properties as $\{k_l\}$ in Assumption 3, although for different values of K. Hence, using arguments similar to those leading to (22), it can be shown that each subsequence $\{u_i(k_h^{(q)} - 1)\}$ tends to zero, thus implying that $\lim_{k\to\infty} \|u_i(k)\| = 0$, for $i = 1, \dots, N$. □

Lemma 5. *Let Assumptions 1-4 hold. Let* $\{x_i(k)\}$, $i = 1, \dots, N$, $k = 1, 2, \dots$, *be the sequences of estimates computed according to* (9). *Define*

$$y(k) = \frac{1}{N} \sum_{i=1}^{N} x_i(k). \tag{23}$$

Then, $\lim_{k\to\infty} \|y(k) - x_i(k)\| = 0$, $i = 1, \dots, N$. □

The proof of Lemma 5 is similar to that of Lemma 9 in [32] and hence it is omitted. It consists in studying the evolution of system (14) when driven by a vanishing input, i.e. when $\lim_{k\to\infty} \|u_i(k)\| = 0$.

We can now prove the main result of the paper.

Theorem 1. *Let Assumptions 1-4 hold. Let* $\{x_i(k)\}$, $i = 1, \dots, N$, $k = 1, 2, \dots$, *be the sequences of estimates computed according to* (9). *Then, there exists a point* $\hat{x}$ *such that*

$$\lim_{k\to\infty} x_i(k) = \hat{x} \in X, \quad i = 1, \dots, N.$$

Proof. Let $y(k)$ be defined as in Lemma 5 and define $q_i(k) = P_{X_i(k)}[y(k)]$. Since $x_i(k) \in X_i(k)$ for $i = 1, 2, \dots, N$ and all k, we have that

$$\sum_{i=1}^{N} \|y(k) - q_i(k)\| \le \sum_{i=1}^{N} \|y(k) - x_i(k)\|. \tag{24}$$

From Lemma 5, (24) implies that

$$\lim_{k\to\infty} \sum_{i=1}^{N} \|y(k) - q_i(k)\| = 0$$

and thus

$$\lim_{k\to\infty} \|y(k) - q_i(k)\| = 0 \tag{25}$$

for $i = 1, \dots, N$. From (25), it follows

$$\lim_{h\to\infty} \|q_i(h) - q_j(h)\| = 0 \tag{26}$$

for $i, j = 1, \dots, N$. Since the sequences $\{x_i(k)\}$ are bounded due to (17), so is the sequence $\{y(k)\}$. Consequently, from (25), the sequences $\{q_i(k)\}$ are bounded too and admit a converging subsequence $\{q_i(k_h)\}$, i.e.

$$\lim_{h\to\infty} q_i(k_h) = \hat{x}_i. \tag{27}$$

By Lemma 2, the limit points $\hat{x}_i \in X_i$. But, from (26) this implies that $\hat{x}_1 = \dots = \hat{x}_N \triangleq \hat{x} \in X$. From (25) and Lemma 5, it follows

$$\lim_{h\to\infty} x_i(k_h) = \hat{x} \tag{28}$$

for $i = 1, \dots, N$. By the definition of limit, for any $\epsilon > 0$ there exists a $\hat{h}$ such that $\|x_i(k_h) - \hat{x}\| < \epsilon$ for all $h \ge \hat{h}$ and

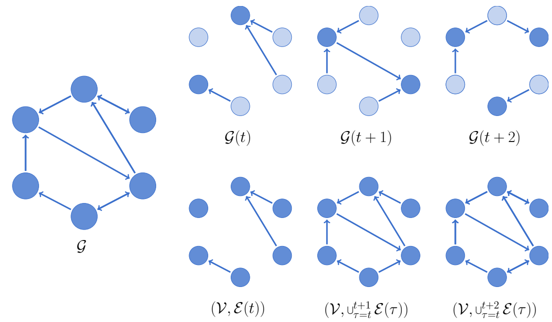

Fig. 1: Time varying graph induced by node awakenings. The underlying graph $\mathcal{G}$ is depicted on the left.

$i = 1, \dots, N$. Using the same arguments leading to (18), one has that for $i = 1, \dots, N$ and all $k \ge k_{\hat{h}}$

$$\begin{aligned} \|x_i(k) - \hat{x}\|^2 &\le \sum_{j=1}^{N} \Psi_{ij}(k, k_{\hat{h}}) \left\|x_j(k_{\hat{h}}) - \hat{x}\right\|^2 \\ &\le \max_j \left\|x_j(k_{\hat{h}}) - \hat{x}\right\|^2 \le \epsilon^2, \end{aligned}$$

where the second inequality comes from the row-stochasticity of matrices $\Psi(k, k_{\hat{h}})$, from which the thesis follows. □

Remark 1. The convergence analysis is carried out in a very general framework. It applies to distributed asynchronous implementation and time-varying communication graphs. We stress that the considered setup encompasses changes in the graph topology (e.g., due to temporary link failures), as well as changes in the weight coefficients. Assumptions 3 and 4 are similar to those adopted in related work on constrained consensus such as [32], [33]. In particular, unbalanced graphs are allowed and the strongly connectedness property must hold only over finite-length time intervals. A peculiarity of the considered recursive estimation setting is that the sets on which the estimates are projected are infinitely many and each of them is processed only once in general. As a consequence, convergence results from previous works on constrained consensus cannot be directed applied. Theorem 1 shows that convergence is indeed guaranteed in this setting as well, provided that the sequence of sets is nonincreasing. □

When the asynchronous Algorithm 1 is run over an underlying communication graph with static topology, Assumption 3 can be replaced by a simple condition about the awakenings of the nodes, as shown next.

IV. Stationary underlying communication graph

In this section, we assume that the network has static topology and it is modeled by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$. The communication graph $\mathcal{G}$ is supposed to be strongly connected and the weight matrix W satisfies Assumption 4. Similarly to what done in Section III, the asynchronous execution of Algorithm 1 is equivalent to the execution of the synchronous algorithm (9) over a suitable time-varying graph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k), W(k))$. Denote by $\mathcal{A}(k)$ the subset of nodes awake at iteration k. Let $D(k) \in \mathbb{R}^{N\times N}$ be a diagonal matrix whose i-th element of the diagonal is set to 1 if node $i \in \mathcal{A}(k)$ and to

0 otherwise and define $D_-(k) = I - D(k)$. Then, the weighted adjacency matrix associated with $\mathcal{E}(k)$ can be written as

$$W(k) = D(k)W + D_-(k). \tag{29}$$

The graph $\mathcal{G}(k)$ is induced from $\mathcal{G}$ (see Figure 1 for a graphical representation) and definition (29) guarantees that if a node is not awake at iteration k it does not change its estimate and its feasible parameter set. Notice that, if the matrix W satisfies Assumption 4, then also the matrices $W(k)$ satisfy it for all k.

In an asynchronous setup like the one we are considering, some nodes may wake up more often than others. The following technical assumption regarding the local triggers ensures that a node cannot remain idle forever or have an infinite frequency of awakenings.

Assumption 5. *For each node $i \in \mathcal{V}$ there exist constants $T_m^{(i)}, T_M^{(i)}$, with $0 < T_m^{(i)} \leq T_M^{(i)} < \infty$, such that*

$$T_m^{(i)} \leq t_{h+1}^{(i)} - t_h^{(i)} \leq T_M^{(i)}, \quad \text{for all } h$$

i.e., node i stays in idle mode for at most $T_M^{(i)}$ seconds, but not less than $T_m^{(i)}$ seconds. □

The following lemma shows that the union of the graphs $\mathcal{G}(k)$ over a sufficiently long time interval has the same topology as the underlying graph $\mathcal{G}$.

Lemma 6. *Let Assumption 5 hold. Then, for all k,*

$$\bigcup_{\tau=k}^{k+K} \mathcal{E}(\tau) = \mathcal{E}$$

with $K = \sum_{i=1}^{N} \left\lfloor \frac{\max_j T_M^{(j)}}{T_m^{(i)}} \right\rfloor + 1$.

Proof. Assumptions 5 implies that, for all $i \in \mathcal{V}$ node i is triggered at least once every $T_M^{(i)}$ seconds and no more frequently than once every $T_m^{(i)}$ seconds. Hence, all nodes are awake at least once every K iterations, thus completing the proof. □

Thanks to Lemma 6, Assumption 5, along with the assumption of strongly connected underlying graph $\mathcal{G}$, guarantees that Assumption 3 is satisfied. Hence, in this setting, the convergence result of Theorem 1 holds true if one replaces Assumption 3 with Assumption 5. This is summarized in the following corollary.

Corollary 1. *Let the underlying communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ be strongly connected. Let Assumptions 1,2,4 and 5 hold. Let $\{x_i(k)\}$, $i = 1, \ldots, N$, $k = 1, 2, \ldots$, be the sequences of estimates computed according to* (9). *Then, there exists a point $\hat{x}$ such that*

$$\lim_{k\to\infty} x_i(k) = \hat{x} \in X, \quad i = 1, \ldots, N.$$

V. Finite computation times

In this section we consider the case in which the computation time associated with the processing taking place at each node is not negligible. In this case, we assume that if node i is asked for the value of its estimate x_i while computing, it will send to its neighbors the value that x_i had at the beginning of the computation. As a consequence, in this set-up the nodes may use outdated information from their neighbors. Nonetheless, we will show that under the assumption of bounded computation times, convergence of Algorithm 1 can still be proved.

Assumption 6. *For each node $i \in \mathcal{V}$ there exists a finite constant $C_i \geq 0$ such that, at each awakening, the node performs the local computations in at most C_i seconds.*

In this scenario, when node i wakes up, it reads the local estimate of its neighbors and, then, it takes up to C_i seconds in order to update x_i. However, in the mean time, its neighbors may have updated their local estimates one or more times. Thus, if at iteration k node i is awake (i.e., $i \in \mathcal{A}(k)$), instead of using $x_j(k)$, $j \in \mathcal{N}_i(k)$ to produce $x_i(k+1)$, it may use outdated versions of them. This means that $x_i(k)$ is computed by using $x_j(k - d_{ij}(k))$, for some integer variables $d_{ij}(k) \geq 0$. In other words, the presence of non negligible computation times induces communication delays in the time varying communication graph. Hence, the equivalent synchronous algorithm (9) becomes

$$X_i(k+1) = X_i(k) \cap M_i(k+1), \tag{30a}$$

$$z_i(k) = \sum_{j=1}^{N} w_{ij}(k) x_j(k - d_{ij}(k)), \tag{30b}$$

$$x_i(k+1) = P_{X_i(k+1)}[z_i(k)]. \tag{30c}$$

Notice that $d_{ii}(k) = 0$ for all i and all k. The value of $d_{ij}(k)$ is upper bounded for all k and all $i, j \in \mathcal{V}$ in the following result.

Lemma 7. *Let Assumptions 5 and 6 hold. Then, for any $i \in \mathcal{V}$,*

$$d_{ij}(k) \leq D_i = \sum_{\ell=1}^{N} \left\lfloor \frac{C_i}{T_m^{(\ell)}} \right\rfloor + 1$$

for all k and all $j \in \mathcal{N}_i(k)$.

Proof. The result follows by using the same arguments as in the proof of Lemma 6. □

By resorting to arguments similar to those used in [32], it can be shown that algorithm (30a)-(30c) can be put in the form of (9) (i.e., without communication delays) by adding suitable fictitious nodes to the communication graph, corresponding to the states $x_i(k-1), x_i(k-2), \ldots, x_i(k-D)$, with $D = \max_i D_i$. Being the delays $d_{ij}(k)$ bounded, according to Lemma 7, the number of such fictitious nodes is bounded as well. Hence, the following result holds.

Corollary 2. *Let Assumptions 1,2,4, 5 and 6 hold. Let $\{x_i(k)\}$, $i = 1, \ldots, N$, $k = 1, 2, \ldots$, be the sequences of* *estimates computed according to* (30)*. Then, there exists a point $\hat{x}$ such that*

$$\lim_{k\to\infty} x_i(k) = \hat{x} \in X, \quad i = 1, \ldots, N.$$

VI. Numerical Example

In order to show how the proposed algorithm works and to validate the provided theoretical results, an example involving a localization problem in a sensor network is presented. Consider N agents which are deployed in a two dimensional region and collaborate in order to estimate the unknown position $x^\star \in \mathbb{R}^2$ of a certain target. Let $c_i \in \mathbb{R}^2$ denote the position of agent i. Each agent is equipped with a sensor for measuring the bearing angle from the target. In particular, when node i wakes up at time $t_h^{(i)}$, it takes a noisy measurement

$$\phi_i(t_h^{(i)}) = \text{atan2}(x^\star - c_i) + \xi_i(t_h^{(i)}). \tag{31}$$

In (31), with some abuse of notation, given $r = [r_1, r_2]^\top$ we denote $\text{atan2}(r) = \text{atan2}(r_2, r_1)$. The measurement error $\xi_i(t_h^{(i)}) \in \mathbb{R}$ is uniformly distributed in $[-\epsilon_i, \epsilon_i]$, with $\epsilon_i \geq 0$. The measurement (31) defines the local feasible measurement set of node i at time $t_h^{(i)}$ as

$$\mathcal{M}_i(t_h^{(i)}) = \{x \in \mathbb{R}^2 \mid |\phi_i(t_h^{(i)}) - \text{atan2}(x - c_i)| \leq \epsilon_i\}. \tag{32}$$

Therefore, from (6a) and (32), the local feasible parameter set can be explicitly computed by storing the maximum and minimum value of the measurements at each iteration as

$$\mathcal{X}_i(t_h^{(i)}) = \Big\{x \in \mathbb{R}^2 \mid \max_{k\leq h} \phi_i(t_k^{(i)}) - \epsilon_i \leq \text{atan2}(x - c_i) \leq \min_{k\leq h} \phi_i(t_k^{(i)}) + \epsilon_i\Big\}.$$

The proposed algorithm has been implemented with DISROPT [36]. The considered scenario involves $N = 10$ agents interconnected through a stationary communication graph as in Section IV, generated as a binomial random graph with edge probability $p = 0.3$. The associated static weight matrix W is generated with the Metropolis-Hastings rule. Idle times are uniformly distributed in the interval $[T_m^{(i)}, T_M^{(i)}]$ and computation times are bounded by D_i for all $i = 1, \ldots, 10$. The target is located at $x^\star = [0.6, -0.1]^\top$ and we set $\epsilon_i = 0.2$ for all i. The initial estimates $x_i(t_0^{(i)})$ and the agent positions c_i are uniformly randomly generated in $[-1.5, 1.5]^2$. We let agents run two instances of Algorithm 1 for $T = 300$ seconds. In the first run we set $T_m^{(i)} = 10^{-4}$, $T_M^{(i)} = 0.1$, and $D_i = 0.1$, while in the second one we set $T_m^{(i)} = 10^{-4}$, $T_M^{(i)} = 0.5$, and $D_i = 0.2$. The evolution of the distance of the local estimates x_i from $\mathcal{X}_T = \cap_{i=1}^N \mathcal{X}(T)$ (i.e., the global feasible set at the end of the simulation) is depicted in Figure 2 (blue lines). As expected, it can be observed that the estimates converge to the global feasible set. Moreover, Figure 2 shows that higher computation and idle times result in a slower convergence rate.

To emphasize the benefits of computing the local feasible parameter set $\mathcal{X}_i(t_h^{(i)})$, we have analyzed the convergence behavior when (6a) is not performed and the local estimates are projected on the local measurement sets, i.e., when in Algorithm 1 (6c) is replaced with

$$x_i \leftarrow P_{\mathcal{M}_i(t_h^{(i)})}[z_i]. \tag{33}$$

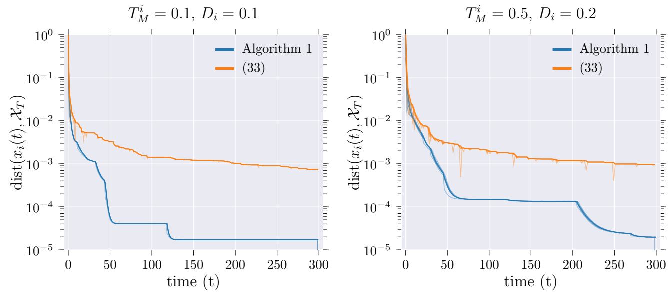

Fig. 2: Evolution of the distance between the local estimates x_i and the global feasible set at the end of the simulation: Comparison with (33).

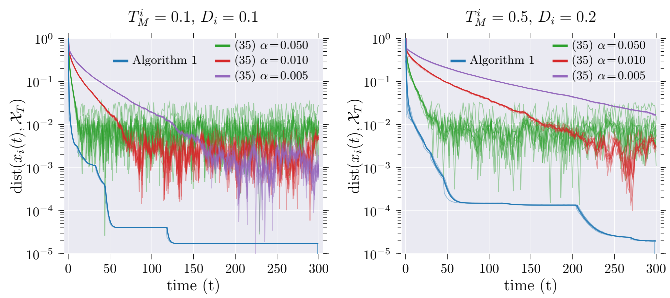

Fig. 3: Evolution of the distance between the local estimates x_i and the global feasible set at the end of the simulation: Comparison with (35).

This can be interpreted as a constrained consensus algorithm applied to the current measurement sets. The results are reported in Figure 2 in terms of the distance of the local estimates x_i from $\mathcal{X}_T$ (orange lines). It can be appreciated that such distances decrease at a significantly slower rate. Moreover, it is easy to construct counterexamples in which the local estimates generated according to (33) converge to a point which does not belong to the global feasible set.

Finally, the proposed algorithm has been compared to a distributed gradient descent algorithm solving the unconstrained localization problem

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{N} \sum_{h\geq 0} \|\phi_i(t_h^{(i)}) - \text{atan2}(x - c_i)\|^2. \tag{34}$$

Techniques for solving such a problem in a distributed and asynchronous way have been widely studied (see e.g. [37]). By setting $f_h^i(x) = \|\phi_i(t_h^{(i)}) - \text{atan2}(x - c_i)\|^2$, the distributed gradient descent algorithm for agent i is obtained by skipping (6a) and replacing (6c) with

$$x_i \leftarrow z_i - \alpha \nabla f_h^i(z_i) \tag{35}$$

in Algorithm 1. Results are reported in Figure 3 for different values of the step size α. It is apparent that a proper choice of α is crucial for obtaining a satisfactory behavior. Moreover, different values of α may be needed for different problem instances, i.e. whenever $T_M^{(i)}$ or D_i change. Conversely, Algorithm 1 is guaranteed to provide estimates that converge to the asymptotic feasible set without the need to tune any parameter.

VII. Conclusion

A set membership estimation problem has been addressed in a distributed way, within a fully asynchronous network, accounting for node or link failures and finite computation times. Convergence of the node local estimates to the global feasible parameter set has been proven, under two main assumptions: strong connectedness of the union of graphs over finite time intervals and convexity of the feasible measurement sets. While the former is standard in networked estimation and control problems, keeping track of generic convex sets might require a high computational burden at the node level. The use of set approximations and the analysis of its impact on the convergence properties of the distributed estimation scheme will be the subject of future research. The application of the proposed technique to set membership state estimation problems is also under investigation.

References

[1] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, 2005, pp. 63–70.

[2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215–233, 2007.

[3] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2009.

[4] M. Farina, G. Ferrari-Trecate, and R. Scattolini, "Distributed moving horizon estimation for linear constrained systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2462–2475, 2010.

[5] Q. Zhang and J.-F. Zhang, "Distributed parameter estimation over unreliable networks with markovian switching topologies," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2545–2560, 2012.

[6] S. Kar, J. M. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575–3605, 2012.

[7] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[8] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[9] F. J. Vazquez-Abad, C. G. Cassandras, and V. Julka, "Centralized and decentralized asynchronous optimization of stochastic discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 5, pp. 631–655, 1998.

[10] M. Zhong and C. G. Cassandras, "Asynchronous distributed optimization with event-driven communication," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2735–2750, 2010.

[11] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011.

[12] I. Notarnicola and G. Notarstefano, "Asynchronous distributed optimization via randomized dual proximal gradient," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2095–2106, 2017.

[13] F. Farina, A. Garulli, A. Giannitrapani, and G. Notarstefano, "A distributed asynchronous method of multipliers for constrained nonconvex optimization," *Automatica*, vol. 103, pp. 243–253, 2019.

[14] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.

[15] D. P. Bertsekas and I. B. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *IEEE Transactions on Automatic Control*, vol. 16, pp. 117–128, 1971.

[16] M. Milanese and A. Vicino, "Optimal estimation theory for dynamic systems with set membership uncertainty: an overview," *Automatica*, vol. 27, no. 6, pp. 997–1009, 1991.

[17] M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino, "Simultaneous localization and map building for a team of cooperating robots: A set membership approach," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 238–249, 2003.

[18] G. Calafiore, "Reliable localization using set-valued nonlinear filters," *IEEE Transactions on systems, man, and cybernetics-part A: systems and humans*, vol. 35, no. 2, pp. 189–197, 2005.

[19] I. Kolmanovsky, I. Sivergina, and J. Sun, "Simultaneous input and parameter estimation with input observers and set-membership parameter bounding: theory and an automotive application," *International Journal of Adaptive Control and Signal Processing*, vol. 20, no. 5, pp. 225–246, 2006.

[20] C. Novara, F. Ruiz, and M. Milanese, "Direct identification of optimal SM-LPV filters and application to vehicle yaw rate estimation," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 1, pp. 5–17, Jan 2011.

[21] M. Canale, L. Fagiano, and C. Novara, "A DVS-MHE approach to vehicle side-slip angle estimation," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 2048–2055, 2014.

[22] V. Cerone, J. B. Lasserre, D. Piga, and D. Regruto, "A unified framework for solving a general class of conditional and robust set-membership estimation problems," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2897–2909, Nov 2014.

[23] J. R. Deller, "Set membership identification in digital signal processing," *IEEE ASSP Magazine*, vol. 6, no. 4, pp. 4–20, 1989.

[24] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Processing Letters*, vol. 5, no. 5, pp. 111–114, 1998.

[25] P. S. Diniz, "Convergence performance of the simplified set-membership affine projection algorithm," *Circuits, Systems, and Signal Processing*, vol. 30, no. 2, pp. 439–462, 2011.

[26] S. Werner, Y.-F. Huang, M. L. De Campos, and V. Koivunen, "Distributed parameter estimation with selective cooperation," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 2849–2852.

[27] H. Shiri, M. A. Tinati, M. Codreanu, and S. Daneshvar, "Distributed sparse diffusion estimation based on set membership and affine projection algorithm," *Digital Signal Processing*, vol. 73, pp. 47–61, 2018.

[28] T. R. Nelson and R. A. Freeman, "Set-valued estimation for mobile sensor networks," in *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009, pp. 2681–2686.

[29] J.-B. Léger and M. Kieffer, "Guaranteed robust distributed estimation in a network of sensors," in *Proceedings of the 2010 IEEE International Conference on Acoustics Speech and Signal Processing*, 2010, pp. 3378–3381.

[30] F. Farina, A. Garulli, and A. Giannitrapani, "Distributed interpolatory algorithms for set membership estimation," *IEEE Transactions on Automatic Control*, vol. 64, no. 9, pp. 3817 – 3822, 2019.

[31] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[32] P. Lin and W. Ren, "Constrained consensus in unbalanced networks with communication delays," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 775–781, 2014.

[33] A. Nedic and J. Liu, "On convergence rate of weighted-averaging dynamics for consensus problems," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 766–781, 2017.

[34] F. Farina, A. Garulli, and A. Giannitrapani, "Distributed set membership estimation with time-varying graph topology," in *Proceedings of the 58th IEEE Conference on Decision and Control*, 2019, pp. 4121–4126.

[35] S. Resnick, *A probability path*. Birkäuser, 2013.

[36] F. Farina, A. Camisa, A. Testa, I. Notarnicola, and G. Notarstefano, "DISROPT: a Python Framework for Distributed Optimization," *arXiv e-prints*, p. arXiv:1911.02410, 2019.

[37] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.