Distributed Asymptotic Minimization of Sequences of Convex Functions by a Broadcast Adaptive Subgradient Method

Renato L. G. Cavalcante, *Member, IEEE*, Alex Rogers, Nicholas R. Jennings, *Fellow, IEEE*, and Isao Yamada, *Senior Member, IEEE*

***Abstract*—We propose a non-hierarchical decentralized algorithm for the asymptotic minimization of possibly time-varying convex functions. In our method, each agent in a network has a private, local (possibly time-varying) cost function, and the objective is to minimize asymptotically the sum of these local functions in every agent (this problem appears in many different applications such as, among others, motion planning, acoustic source localization, and environmental modeling). The algorithm consists of two main steps. First, to improve the estimate of a minimizer, agents apply a particular version of the adaptive projected subgradient method to their local functions. Then the agents exchange and mix their estimates using a communication model based on recent results of consensus algorithms. We show formally the convergence of the resulting scheme, which reproduces as particular cases many existing methods such as gossip consensus algorithms and recent decentralized adaptive subgradient methods (which themselves include as particular cases many distributed adaptive filtering algorithms). To illustrate two possible applications, we consider the problems of acoustic source localization and environmental modeling via network gossiping with mobile agents.**

***Index Terms*—Adaptive projected subgradient method, decentralized estimation and detection via network gossiping, decentralized optimization via network gossiping, gossip algorithms for decentralized adaptive filtering.**

I. Introduction

In many applications involving systems of autonomous interacting agents, the objective is to solve an optimization problem in which the cost function (hereafter termed the global function) can be decomposed as the sum of local cost functions, each of which is known by only one agent in a network. Applications that can be posed as such optimization problems include, among others, motion planning in multiagent systems [1], [2], acoustic source localization [3]–[5], and distributed adaptive filtering [6]–[8]. Typically, in these problems centralized approaches are not desirable because of physical limitations (the central agent may not have a direct connection with all other agents) or because of robustness issues (the system may fail if the central agent collapses). Therefore, a great deal of effort has been devoted to the development of non-hierarchical distributed optimization algorithms [1]–[7], [9]–[11]. In particular, here we focus on decentralized subgradient methods where agents can work massively in parallel and exchange information with point-to-multipoint links [1], [5], [6], [9], [11]. These approaches often give rise to low-complexity iterative optimization algorithms that are suitable for large-scale systems using a simple communication model among agents.

To date, the majority of distributed subgradient methods have focused on static systems where the cost function does not change during the iterations of the algorithm [1], [9], [11]. Formally, once these algorithms start running, agents have to wait until a good estimate of the minimizer of the global function is obtained in every agent. This process can take many iterations in large-scale systems, and in several applications agents may need to change frequently the local functions (and, consequently, the global function) to drop outdated information or to add new information gathered from the environment. For example, in estimation problems involving mobile sensor networks, agents may need to estimate a parameter of interest (e.g., the position of a target) by minimizing a global function that is built based on measurements, obtained at different locations, of a physical phenomenon (e.g., the sound intensity). As a result, if agents keep taking measurements of the environment after the optimization algorithm starts running, they may be able to improve the estimate of the parameter of interest by incorporating the new available information into new local cost functions.[1] Unfortunately, most studies on distributed subgradient methods do not characterize the behavior of the algorithms in such dynamic systems.

A distributed algorithm that considers time-varying cost functions has been proposed in [5], [6]. The algorithm is based on the adaptive projected subgradient method [12], [13] (which itself is an extension of Polyak's algorithm [17] to handle time-varying

[1]See [5], [6], [12]–[16] for applications of centralized optimization algorithms involving time-varying functions.

Manuscript received June 19, 2010; revised November 22, 2010; accepted January 25, 2011. Date of publication February 14, 2011; date of current version July 20, 2011. This work was supported by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the U.K. Ministry of Defence. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Anna Scaglione.

R. L. G. Cavalcante is with the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, 10587 Berlin, Germany (e-mail: renato.cavalcante@hhi.fraunhofer.de).

A. Rogers and N. R. Jennings are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: acr@ecs.soton.ac.uk; nrj@ecs.soton.ac.uk).

I. Yamada is with the Department of Communications and Integrated Systems, Tokyo Institute of Technology, Tokyo 152-8550, Japan (e-mail: isao@comm.ss.titech.ac.uk).

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/JSTSP.2011.2114325

cost functions), and thus it can be applied to problems where the environment is nonstationary or where incoming data from sensors has to be processed online and in real time. This algorithm uses a network model in which links are considered deterministic, but recent results in consensus algorithms [18]–[24] and also in distributed optimization problems with fixed cost functions [10], [11] have shown that modeling the network links among agents as random links is highly desirable for flexibility purposes. In particular, random links can easily model wireless networks in which agents communicate asynchronously with simple broadcast channels where simultaneous information exchange is not possible [20]. In such networks, the assumptions used in the analysis of the algorithm in [5], [6] are not satisfied, and thus the results in [5], [6] cannot be formally applied to important classes of multiagent-systems systems using wireless networks (in particular, the algorithms in Section IV cannot be derived from the method in [5] and [6] owing to the assumptions on the communication model). An addition limitation of the analysis in [5] and [6] is that it only shows conditions for the asymptotic minimization of the cost functions, which neither guarantees the convergence of the algorithm nor characterizes the convergence point.

To address the shortcomings of the above-mentioned schemes, we develop an iterative optimization algorithm that can deal with both time-varying cost functions and random links among agents. In the first step, as in [6], each agent improves its own local estimate of the minimizer of the (possibly time-varying) global function by applying a particular version of the adaptive projected subgradient method [12], [13] to its local cost function. In the second step of the algorithm, unlike [1], [5], [6], [9], agents communicate through possibly random links. More specifically, here we adopt a general communication model that includes as particular examples the methods used in recent algorithms for consensus via network gossiping [18]–[23]. Our approach has convergence guarantees in dynamic systems and can reproduce and extend, within a unified framework, many existing distributed algorithms. We can, for example, address the limitations of existing batch and adaptive algorithms by changing the cost functions (e.g., to consider the presence of mobile agents) and/or by choosing a different communication model. Convergence properties of those modified algorithms follow directly from the analysis of our general framework—they do not need to be studied separately for each possible scenario. In particular, we show how to derive, from the general method developed here, adaptive algorithms for environmental modeling (decentralized adaptive filtering) and for acoustic source localization with mobile agents. Note, however, that our algorithm in its most general form is by no means restricted to applications in these particular domains.[2]

The structure of the paper is as follows. Section II outlines basic tools in convex analysis and reviews a class of problems with many applications in multiagent systems. Section III introduces and analyzes the proposed algorithm, which solves the problem in Section II. In Section IV, we show two possible applications of the proposed method: acoustic source localization and environmental modeling. The appendices contain the proof of lemmas and theorems.

[2]A short version of this paper appeared in [25]. Unlike the study in [25], here we show the full proof of our main results, additional convergence properties of the proposed algorithm, and also new algorithms for acoustic source localization and environmental modeling.

II. Preliminaries

A. Basic Tools in Convex Analysis

In this section we give a number of results and definitions that are extensively used in the discussion that follows. In particular, we denote by $\lfloor x \rfloor$ the largest integer not exceeding x. The component of the ith row and jth column of a matrix $\boldsymbol{X}$ is given by $[\boldsymbol{X}]_{ij}$. For every vector $\boldsymbol{v} \in \mathbb{R}^N$, we define the norm of $\boldsymbol{v}$ by $\|\boldsymbol{v}\| := \sqrt{\boldsymbol{v}^T\boldsymbol{v}}$, which is the norm induced by the Euclidean inner product $\langle \boldsymbol{v}, \boldsymbol{y} \rangle := \boldsymbol{v}^T\boldsymbol{y}$ for every $\boldsymbol{v}, \boldsymbol{y} \in \mathbb{R}^N$. For a matrix $\boldsymbol{X} \in \mathbb{R}^{M\times N}$, its spectral norm is $\|\boldsymbol{X}\|_2 := \max\{\sqrt{\lambda} \,|\, \lambda \text{ is an eigenvalue of } \boldsymbol{X}^T\boldsymbol{X}\}$, which satisfies $\|\boldsymbol{X}\boldsymbol{y}\| \leq \|\boldsymbol{X}\|_2\|\boldsymbol{y}\|$ for any vector $\boldsymbol{y}$ of compatible size. In the sequel, $(\Omega, \mathcal{F}, \mathcal{P})$ always denotes probability spaces, where Ω is the sure event, $\mathcal{F}$ is the σ-field of events, and $\mathcal{P}$ is the probability measure. To avoid tedious repetition, we often omit the underlying probability spaces. Unless otherwise stated, we always use the Greek letter $\omega \in \Omega$ to denote a particular outcome. Thus, by $\boldsymbol{x}_\omega$ ($\boldsymbol{X}_\omega$), we denote an outcome of the random vector $\boldsymbol{x}$ (matrix $\boldsymbol{X}$). We also often drop the qualifier "almost surely" (or "with probability one") in equations involving random variables.

A set C is said to be *convex* if $\boldsymbol{v} = \nu\boldsymbol{v}_1 + (1-\nu)\boldsymbol{v}_2 \in C$ for every $\boldsymbol{v}_1, \boldsymbol{v}_2 \in C$ and $0 \leq \nu \leq 1$. If, in addition to being convex, C contains all its boundary points, then C is a *closed convex set* [26], [27]. The *metric projection* $P_C : \mathbb{R}^N \to C$ of a closed convex set C maps $\boldsymbol{v} \in \mathbb{R}^N$ to the uniquely existing vector $P_C(\boldsymbol{v}) \in C$ satisfying $\|\boldsymbol{v} - P_C(\boldsymbol{v})\| = \min_{\boldsymbol{y}\in C}\|\boldsymbol{v} - \boldsymbol{y}\| =: d(\boldsymbol{v}, C)$.

A function $\Theta : \mathbb{R}^N \to \mathbb{R}$ is said to be *convex* if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ and $\forall \nu \in [0,1]$, $\Theta(\nu\boldsymbol{x} + (1-\nu)\boldsymbol{y}) \leq \nu\Theta(\boldsymbol{x}) + (1-\nu)\Theta(\boldsymbol{y})$ (in this case Θ is continuous at every point in $\mathbb{R}^N$). The *c-sublevel set* of a function $\Theta : \mathbb{R}^N \to \mathbb{R}$ is defined by $\text{lev}_{\leq c}\Theta := \{\boldsymbol{h} \in \mathbb{R}^N \,|\, \Theta(\boldsymbol{h}) \leq c\}$, which is a closed convex set for every $c \in \mathbb{R}$ if Θ is convex [27]. Convex functions are not necessarily differentiable everywhere, so subgradients play a special role in the results that follow. In more detail, if $\Theta : \mathbb{R}^N \to \mathbb{R}$ is a convex function, then the *subdifferential* of Θ at $\boldsymbol{y}$, denoted by $\partial\Theta(\boldsymbol{y})$, is the nonempty closed convex set of all *subgradients* of Θ at $\boldsymbol{y}$:

$$\partial\Theta(\boldsymbol{y}) := \{\boldsymbol{a} \in \mathbb{R}^N \,|\, \Theta(\boldsymbol{y}) + \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{a} \rangle \leq \Theta(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^N\}. \tag{1}$$

In particular, if Θ is differentiable at $\boldsymbol{y}$, then the only subgradient in the subdifferential is the gradient, i.e., $\partial\Theta(\boldsymbol{y}) = \{\nabla\Theta(\boldsymbol{y})\}$.

We end this subsection with results that we use to simplify the analysis of the proposed algorithm.

Fact 1 ([13], Claim 2): Let $C \subset \mathbb{R}^N$ be a closed convex set having a point $\tilde{\boldsymbol{u}} \in C$ such that $\emptyset \neq \{\boldsymbol{h} \in \mathbb{R}^N \,|\, \|\boldsymbol{h} - \tilde{\boldsymbol{u}}\| \leq \rho\} \subset C$ for some $\rho > 0$. If for given $\boldsymbol{v} \in \mathbb{R}^N \backslash C$ and $t \in (0,1)$ we have $\boldsymbol{u}_t := (1-t)\tilde{\boldsymbol{u}} + t\boldsymbol{v} \notin C$, then $d(\boldsymbol{v}, C) > \rho(1-t)/(t) = \rho(\|\boldsymbol{u}_t - \boldsymbol{v}\|)/(\|\boldsymbol{u}_t - \tilde{\boldsymbol{u}}\|) > 0$.

Theorem 1: Assume that random vectors $\{\boldsymbol{x}[i]\}$ $(i = 0, 1, \ldots)$ with $E[\|\boldsymbol{x}[0]\|^2] < \infty$ are defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$. Suppose that, for a given set $C \subset \mathbb{R}^M$ and for any $\boldsymbol{x}^\star \in C$, we have

$$E[\|\boldsymbol{x}[i+1] - \boldsymbol{x}^\star\|^2 \,|\, \boldsymbol{x}[i], \ldots, \boldsymbol{x}[0]] \leq \|\boldsymbol{x}[i] - \boldsymbol{x}^\star\|^2 - y[i] + z[i]$$

where $y[i]$ and $z[i]$ are sequences of non-negative random variables that are functions of $\boldsymbol{x}[0], \ldots, \boldsymbol{x}[i]$. If $\sum_{i=0}^{\infty} E[z[i]] < \infty$, which also implies that $\sum_{i=0}^{\infty} z[i] < \infty$ with probability one [28, p. 60], then we have the following properties:

1) the sequence $\{\|\boldsymbol{x}[i] - \boldsymbol{x}^\star\|\}$ converges almost surely (or with probability one) for any $\boldsymbol{x}^\star \in C$, and $E[\|\boldsymbol{x}[i] - \boldsymbol{x}^\star\|^2] < \infty$;
2) the set of accumulation points $\{\boldsymbol{x}_\omega[i]\}$ is not empty for almost every $\omega \in \Omega$;
3) if two accumulation points $\boldsymbol{x}'_\omega$ and $\boldsymbol{x}''_\omega$ of the sequence $\{\boldsymbol{x}_\omega[i]\}$ are such that $\boldsymbol{x}'_\omega, \boldsymbol{x}''_\omega \notin C$, then the set C lies in a hyperplane equidistant from the points $\boldsymbol{x}'_\omega$ and $\boldsymbol{x}''_\omega$, or, in other words, $\|\boldsymbol{x}'_\omega - \boldsymbol{x}^\star\|^2 = \|\boldsymbol{x}''_\omega - \boldsymbol{x}^\star\|^2$ for every $\boldsymbol{x}^\star \in C$;
4) with probability one, $\sum_{i=0}^{\infty} y[i] < \infty$.

See [29, Theorem 1] for the proof of the first three properties. For the last property, see [30, Prop. 4.2] and the references therein.

B. Problem Formulation

In this paper, we consider a multiagent optimization problem related to that in [5] and [6]. In more detail, at time i, we represent a network with N agents by a (random) directed graph denoted by $\mathcal{G}[i] := (\mathcal{N}, \mathcal{E}[i])$, where $\mathcal{N} = \{1, \ldots, N\}$ is the set of agents and $\mathcal{E}[i] \subseteq \mathcal{N} \times \mathcal{N}$ is the edge set [23]. The edges of the graph indicate possible communication between two agents. More precisely, if agent k can send information to agent l at time i, then $(k, l) \in \mathcal{E}[i]$ (we assume that $(k, k) \in \mathcal{E}[i]$). Inward neighbors of agent k are denoted by $\mathcal{N}_k[i] = \{l \in \mathcal{N} \,|\, (l, k) \in \mathcal{E}[i]\}$ (i.e., $l \in \mathcal{N}_k[i]$ is an agent that can send information to agent k at time i).

We assume that each agent k has knowledge of a local convex cost function $\Theta_k[i] : \mathbb{R}^M \to [0, \infty)$ $(i \in \mathbb{N})$. Note that $\Theta_k[i]$ is non-negative, possibly time-varying, and not necessarily differentiable. Now, define the global cost function $\Theta[i] : \mathbb{R}^M \to [0, \infty)$ of the network by

$$\Theta[i](\boldsymbol{h}) = \sum_{k \in \mathcal{N}} \Theta_k[i](\boldsymbol{h}) \tag{2}$$

which is the function that the agents try to minimize at every time instant i. We also assume that each agent has its own estimate $\boldsymbol{h}_k[i]$ $(k \in \mathcal{N})$ of a minimizer of (2) and that $\Theta_k[i]$ is private information of agent k. With these assumptions, if we also impose that all agents should reach consensus on a minimizer of (2), we obtain the following optimization problem at time i:

$$\begin{aligned} \text{minimize} \quad & \sum_{k \in \mathcal{N}} \Theta_k[i](\boldsymbol{h}_k[i]) \\ \text{subject to} \quad & \boldsymbol{h}_k[i] = \boldsymbol{h}_l[i], \quad \forall k, l \in \mathcal{N}. \end{aligned} \tag{3}$$

Unfortunately, solving (3) at every time instant i is difficult if the communication among agents is limited because in such a case agents have only partial information of the problem. Conventional iterative decentralized algorithms that are able to find an approximate solution of (3) (for fixed i) may require many iterations, but in many real-time applications the optimization problem can change as often as every iteration of the algorithm (cf. Section IV). To handle such dynamic scenarios, we devise an algorithm that allows the local functions to change during the iterative process and that solves (3) asymptotically (a precise definition will be given soon). Here we mostly focus on the case where the problem in (3) has spatially and temporally related local cost functions, as defined below. (This class of problems appears in many practical applications [6].)

Definition 1: (*Spatially related local functions*) If, for any time index i, the sets of minimizers of the local cost functions $\Theta_k[i]$ $(k \in \mathcal{N})$ have nonempty intersection, we say that the local functions $\Theta_k[i]$ $(k \in \mathcal{N})$ are *spatially related*. More precisely, the time-varying local functions $\Theta_k[i]$ $(k \in \mathcal{N})$ are spatially related if the following holds for every $i \in \mathbb{N}$:

$$\Upsilon[i] := \bigcap_{k \in \mathcal{N}} \Upsilon_k[i] \neq \emptyset \tag{4}$$

where

$$\Upsilon_k[i] := \{\boldsymbol{h} \in \mathbb{R}^M \,|\, \Theta_k[i](\boldsymbol{h}) = \Theta_k^\star[i] := \inf_{\boldsymbol{h} \in \mathbb{R}^M} \Theta_k[i](\boldsymbol{h})\}. \tag{5}$$

Note that, in particular, if the local functions are spatially related, then $\Upsilon[i]$ is a set of minimizers of the global function $\Theta[i]$.

Definition 2: (*Temporally related local functions*) If the functions $\Theta_k[i]$ $(k \in \mathcal{N})$ are such that the resulting global functions $\Theta[i]$ $(i \in \mathbb{N})$ have a common set of minimizers, we say that the local functions $\Theta_k[i]$ $(k \in \mathcal{N})$ are temporally related. In other words, the local functions $\Theta_k[i]$ $(k \in \mathcal{N})$ are temporally related if $\bigcap_{i \in \mathbb{N}} \Upsilon_G[i] \neq \emptyset$, where $\Upsilon_G[i]$ is the set of minimizers of the global function at time i:

$$\Upsilon_G[i] := \{\boldsymbol{h} \in \mathbb{R}^M \,|\, \Theta[i](\boldsymbol{h}) = \inf_{\boldsymbol{h} \in \mathbb{R}^M} \Theta[i](\boldsymbol{h})\}.$$

The optimization problem in (3) can be seen as a sequence of optimization problems indexed by i. If the local functions $\Theta_k[i]$ are both spatially and temporally related,[3] there is a point in $\mathbb{R}^M$ that is a minimizer of every local cost function $\Theta_k[i]$ and every global function $\Theta[i]$ for every $i \in \mathbb{N}$. As a result, (3) has at least one solution that does not depend on the time index i, so we should seek those solutions that solve (3) for as many time indices i as possible (ideally, for all $i \in \mathbb{N}$). Unfortunately, except in special cases, computing a time-invariant solution of (3) (a solution that does not depend on $i \in \mathbb{N}$) can be only possible with *a priori* knowledge of $\Theta_k[i]$ also for every $i \in \mathbb{N}$ and $k \in \mathcal{N}$, which is a very strong assumption in online algorithms because the functions $\Theta_k[i]$ are dispersed throughout the network and are constructed as information is obtained. Nonetheless, with some mild additional assumptions, we can devise a low-complexity algorithm that guarantees mean square asymptotic consensus among agents and that, with probability one, minimizes asymptotically all local cost functions. (Note: if all

[3] If the cost functions are not spatially related, but the local functions are time-invariant, we can use, for example, the algorithms in [1], [9], [11] to solve (approximately) the resulting optimization problem.

local functions are minimized and agents are in consensus at time i, we have a solution to the problem in (3) at time i.) In addition, the algorithm also guarantees that, with probability one, all sequences $\{\boldsymbol{h}_k[i]\}$ $(k \in \mathcal{N})$ converge to a (random) vector that, loosely speaking, is "sufficiently" close to the set of points that minimize all but a finite number of global functions $\Theta[i]$, $i \in \mathbb{N}$. This last property shows that the time structure of the problem in (3) is also exploited. Before showing the algorithm, we need to formalize what we mean by "asymptotic minimization of time-varying functions" and by "asymptotic consensus."

Definition 3: (Asymptotic minimization [13]) Let $\Theta[i] : \mathbb{R}^M \to [0,\infty)$ be any given time-varying function, and denote by $\boldsymbol{h}[i] \in \mathbb{R}^M$ an estimate of a minimizer of $\Theta[i]$, where i is the time index. Assume that, for every $i \in \mathbb{N}$, there is a time-invariant scalar $\Theta^\star \in [0,\infty)$ such that $\Theta^\star = \inf_{\boldsymbol{h}\in\mathbb{R}^M} \Theta[i](\boldsymbol{h})$. We say that an algorithm minimizes asymptotically $\Theta[i]$ if the algorithm produces a (not necessarily convergent) sequence $\{\boldsymbol{h}[i]\}$ satisfying

$$\lim_{i\to\infty} \Theta[i](\boldsymbol{h}[i]) = \Theta^\star.$$

Definition 4: (Asymptotic consensus [6]) We say that agents reach asymptotic consensus if the estimates $\boldsymbol{h}_1[i],\ldots,\boldsymbol{h}_N[i]$ satisfy

$$\lim_{i\to\infty} [(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i]] = \boldsymbol{0} \tag{6}$$

where $\boldsymbol{\psi}[i] := [\boldsymbol{h}_1[i]^T \ldots \boldsymbol{h}_N[i]^T]^T$, $\boldsymbol{J} := \boldsymbol{B}\boldsymbol{B}^T \in \mathbb{R}^{MN\times MN}$, $\boldsymbol{B} := [\boldsymbol{b}_1 \ldots \boldsymbol{b}_M] \in \mathbb{R}^{MN\times M}$, $\boldsymbol{b}_k = (\boldsymbol{1}_N \otimes \boldsymbol{e}_k)/\sqrt{N} \in \mathbb{R}^{MN}$, $\boldsymbol{1}_N \in \mathbb{R}^N$ is the vector of ones, $\boldsymbol{e}_k \in \mathbb{R}^M$ $(k = 1,\ldots,N)$ is the standard basis vector, and $\otimes$ denotes the Kronecker product. (The convergence of $\boldsymbol{h}_k[i]$ is not a requirement.)

In the last definition, note that $\boldsymbol{J}$ is the orthogonal projection matrix onto the consensus subspace

$$\mathcal{C} := \operatorname{span}\{\boldsymbol{b}_1,\ldots,\boldsymbol{b}_M\}. \tag{7}$$

Therefore, if $\boldsymbol{\psi}[i] \in \mathcal{C}$, then $(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i] = \boldsymbol{0}$ and all local estimates $\boldsymbol{h}_k[i]$ $(k \in \mathcal{N})$ are equal, i.e., we have consensus at time i : $\boldsymbol{h}_k[i] = \boldsymbol{h}_j[i]$ for every $k, j \in \mathcal{N}$.

III. Proposed Algorithm

To find in every agent a common point that minimizes all but finitely many global functions $\Theta[i]$ $(i \in \mathbb{N})$, we use a simple algorithm that consists of two steps, each of which exploits directly the assumption that the local functions are spatially related.

In the first step of the algorithm, agents use the spatial relation assumption from a local perspective; each agent exploits the fact that there is a minimizer of *its own local function* that is also a minimizer of the global function. In more detail, each agent k improves its estimate $\boldsymbol{h}_k[i]$ of the minimizer of the global function by finding a point that is also an improved estimate of its own local cost function $\Theta_k[i]$. Mathematically, as in [5], [6], each agent k applies a particular version of the adaptive projected subgradient method [13] to its local function $\Theta_k[i]$:

$$\boldsymbol{h}'_k[i+1] = \boldsymbol{h}_k[i] - \mu_k[i]\frac{(\Theta_k[i](\boldsymbol{h}_k[i]) - \Theta^\star_k[i])}{(||\Theta'_k[i](\boldsymbol{h}_k[i])||^2 + \delta_k[i])}\Theta'_k[i](\boldsymbol{h}_k[i]) \tag{8}$$

where $\boldsymbol{h}'_k[i+1]$ is the resulting estimate after the subgradient update; $\Theta'_k[i](\boldsymbol{h}_k[i]) \in \partial\Theta_k[i](\boldsymbol{h}_k[i])$ [see (1)] is a subgradient of $\Theta_k[i]$ at $\boldsymbol{h}_k[i]$; $\mu_k[i] \in (0,2)$ is a step size; $\Theta^\star_k[i] := \inf_{\boldsymbol{h}\in\mathbb{R}^M} \Theta_k[i](\boldsymbol{h})$, $k \in \mathcal{N}$; $\delta_k[i]$ is arbitrarily chosen from $0 < \delta_k[i] \le L$ if $\Theta'_k[i](\boldsymbol{h}_k[i]) = \boldsymbol{0}$ or $\delta_k[i] = 0$ otherwise; $L > 0$ is an arbitrarily chosen upper bound for the choice of $\delta_k[i]$; and $\boldsymbol{h}_k[0]$ is an arbitrary initial (deterministic) estimate of a minimizer of the global function $\Theta[0]$.[4]

In the second step of the algorithm, agents use the spatial relation assumption from a global perspective; they use the fact that a point that minimizes *every local function* is also a minimizer of the global function. The main idea is that agents should try to reach consensus on their estimates in the hope that they agree on a point that minimizes every local function. Mathematically, for a network represented by a graph $\mathcal{G}[i]$, agents exchange information locally with consensus algorithms similar to those in [18]–[23]

$$\boldsymbol{h}_k[i+1] = \sum_{j\in\mathcal{N}_k[i]} \boldsymbol{W}_{kj}[i]\boldsymbol{h}'_j[i+1], \quad k = 1,\ldots,N \tag{9}$$

where $\boldsymbol{W}_{kj}[i] : \Omega \to \mathbb{R}^{M\times M}$ is a random weight matrix that agent k assigns to the edge (j,k) at time i ($\boldsymbol{W}_{kj}[i] = \boldsymbol{0}$ if $(j,k) \notin \mathcal{E}[i]$). We can rewrite (9) in the equivalent form $[\boldsymbol{h}_1[i+1]^T \ldots \boldsymbol{h}_N[i+1]^T]^T = \boldsymbol{P}[i][\boldsymbol{h}'_1[i+1]^T \ldots \boldsymbol{h}'_N[i+1]^T]^T$, where $\boldsymbol{P}[i] : \Omega \to \mathbb{R}^{MN\times MN}$ is given by

$$\boldsymbol{P}[i] = \begin{bmatrix} \boldsymbol{W}_{11}[i] & \ldots & \boldsymbol{W}_{1N}[i] \\ \vdots & \ddots & \vdots \\ \boldsymbol{W}_{N1}[i] & \ldots & \boldsymbol{W}_{NN}[i] \end{bmatrix}. \tag{10}$$

Here we assume that, periodically (cf. Theorem 2), $\boldsymbol{P}[i]$ is an ϵ-random consensus matrix conditioned on $[\boldsymbol{h}_1[i]^T \ldots \boldsymbol{h}_N[i]^T]^T$ as defined below.

Definition 5: (ϵ-random consensus matrix) For given $\epsilon \in (0,1]$ and graph $\mathcal{G}(\mathcal{N},\mathcal{E}[i])$, we define an ϵ-random consensus matrix conditioned on a random vector $\boldsymbol{\psi}$ at time i as a random matrix $\boldsymbol{P}[i] : \Omega \to \mathbb{R}^{MN\times MN}$ satisfying the following properties:

1) $||E[\boldsymbol{P}[i]^T(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{P}[i]\,|\,\boldsymbol{\psi}]||_2 \le (1-\epsilon)$;
2) $||E[\boldsymbol{P}[i]^T\boldsymbol{P}[i]\,|\,\boldsymbol{\psi}]||_2 = 1$;
3) $\boldsymbol{P}[i]\boldsymbol{v} = \boldsymbol{v}$ for every $\boldsymbol{v} \in \mathcal{C}$ [see (7)].
4) If $\boldsymbol{P}[i]$ is decomposed as in (10), then $\boldsymbol{W}_{kj}[i] = \boldsymbol{0}$ if $(j,k) \notin \mathcal{E}[i]$.

Note that ϵ-random consensus matrices are a simple extension of conventional consensus matrices [18]–[23] to the case where consensus has to be reached over vectors. Therefore, we can use many different techniques [18]–[23] to build these matrices. Section IV has one example of such a technique.

Before showing the main properties of the algorithm, we introduce the following lemma, which is used to simplify the analysis.

[4]Note that (8) requires knowledge of the minimum value attained by the local functions. Although this information may seem to be a strong assumption, in many practical problems the minimum value attained by functions is available. Examples of such problems include all those where the objective is to find a point in the intersection of closed convex sets [13], and they are frequently found in, among other application domains, communication, optics, and signal and image processing [26]. In Section IV, we show two examples of applications.

Lemma 1: Assume that $\{a[i]\}$ is a real sequence satisfying $\lim_{i\to\infty} a[i] = 0$. In addition, let $\{\lambda[i]\}$ be a non-negative sequence such that $0 \leq \lambda[i] \leq 1$ for every i. If there exist $I \in \mathbb{N}$ and $\beta \in [0,1)$ such that, for every $l \in \mathbb{N}$, we have $0 \leq \lambda[i] \leq \beta$ for at least one time instant $i \in [l, l+I]$, then

$$\lim_{i\to\infty} \sum_{j=0}^{i} \prod_{n=0}^{i-j} \lambda[i-n]a[j] = 0.$$

Proof: The proof is shown in Appendix I. ∎

We now summarize and analyze the proposed algorithm.

Theorem 2 (Broadcast Adaptive Subgradient Method): Consider the problem in Section II-B and assume that, for every $i \in \mathbb{N}$ and conditioned on $\boldsymbol{\psi}[i] = [\boldsymbol{h}_1[i]^T \dots \boldsymbol{h}_N[i]^T]^T$, $\boldsymbol{P}[i] : \Omega \to \mathbb{R}^{MN\times MN}$ satisfies properties 2), 3), and 4) in Definition 5. To solve the problem described in Section II-B, we use the following sequence (which is obtained by combining (8), (9), and (10) in a single equation):

$$\boldsymbol{\psi}[i+1] = \boldsymbol{P}[i]\left(\boldsymbol{\psi}[i] - \begin{bmatrix} \mu_1[i]\alpha_1[i]\Theta_1'[i](\boldsymbol{h}_1[i]) \\ \vdots \\ \mu_N[i]\alpha_N[i]\Theta_N'[i](\boldsymbol{h}_N[i] \end{bmatrix}\right) \quad (11)$$

where $\alpha_k[i] = (\Theta_k[i](\boldsymbol{h}_k[i]) - \Theta_k^\star[i])/(\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2 + \delta_k[i])$ and $\boldsymbol{\psi}[0]$ is considered deterministic. (NOTE: other parameters have been defined after (8).) The algorithm satisfies the following:

1) *(Mean square monotone approximation)*:
Suppose that the local functions $\Theta_k[i]$ are spatially related, and let the step sizes be within the interval $\mu_k[i] \in (0,2)$. Then, for every $\boldsymbol{h}^\star[i] \in \Upsilon[i]$ [$\Upsilon[i]$ is defined in (4)]

$$E[\|\boldsymbol{\psi}[i+1] - \boldsymbol{\psi}^\star[i]\|^2] \leq E[\|\boldsymbol{\psi}[i] - \boldsymbol{\psi}^\star[i]\|^2] \quad (12)$$

where $\boldsymbol{\psi}^\star[i] := [\boldsymbol{h}^\star[i]^T \dots \boldsymbol{h}^\star[i]^T]^T \in \mathbb{R}^{MN}$.

2) *(Almost sure asymptotic minimization of the local cost functions):*
Assume the following.
   a) The step size in every agent is bounded away from zero and two, i.e., there exist $\epsilon_1, \epsilon_2 > 0$ such that $\mu_k[i] \in [\epsilon_1, 2-\epsilon_2] \subset (0,2)$.
   b) $\Theta_k^\star[i] =: \Theta_k^\star \in \mathbb{R}, i = 0, 1, \dots$.
   c) $\Upsilon := \bigcap_{i\geq 0} \Upsilon[i] \neq \emptyset$ (i.e., the local functions are spatially and temporally related).
   d) There exists some $\mathcal{M} > 0$ satisfying $\|\Theta_k'[i](\boldsymbol{h}_k[i])\| < \mathcal{M}$ for every $k \in \mathcal{N}$ and $i = 0, 1, \dots$.

   (Assumption c) guarantees $\emptyset \neq C^\star := \{[\boldsymbol{h}^T \dots \boldsymbol{h}^T]^T \in \mathbb{R}^{MN} \,|\, \boldsymbol{h} \in \Upsilon\}$.) Then, with probability one and for any $\boldsymbol{\psi}^\star \in C^\star$, the sequence $\{\|\boldsymbol{\psi}[i] - \boldsymbol{\psi}^\star\|^2\}$ converges, and the local cost functions are asymptotically minimized, i.e., $\mathcal{P}(\lim_{i\to\infty} \Theta_k[i](\boldsymbol{h}_k[i]) = \Theta_k^\star) = 1$.

3) *(Mean square asymptotic consensus):*
In addition to the assumptions above, for some fixed $\epsilon > 0$ and $I \in \mathbb{N}$, assume that in every interval $[l, l+I]$ $(l \in \mathbb{N})$ there exists $i \in [l, l+I]$ such that the matrix $\boldsymbol{P}[i]$ is an ϵ-random consensus matrix conditioned on $\boldsymbol{\psi}[i]$. Then we have asymptotic mean square consensus, i.e.,

$$\lim_{i\to\infty} E[\|(\boldsymbol{I} - \boldsymbol{J})\boldsymbol{\psi}[i]\|^2] = 0. \quad (13)$$

4) (*Almost sure convergence of* $\boldsymbol{\psi}[i]$):
If the assumptions in part 3) hold and $C^\star$ does not lie in a hyperplane, then, with probability one, $\boldsymbol{\psi}[i]$ converges to a random vector $\hat{\boldsymbol{\psi}}$ that satisfies $(\boldsymbol{I} - \boldsymbol{J})\hat{\boldsymbol{\psi}} = \boldsymbol{0}$ (i.e., the estimates $\boldsymbol{h}_k[i]$ of every agent converge, and the agents reach consensus asymptotically).

5) (*Characterization of* $\hat{\boldsymbol{\psi}}$):
Assume that the conditions of 4) hold. By 2) and 4), for almost every $\omega \in \Omega$, we know that $\boldsymbol{\psi}_\omega[i] = [\boldsymbol{h}_{1,\omega}[i]^T \dots \hat{\boldsymbol{h}}_{N,\omega}[i]^T]^T$ converges to a vector $\hat{\boldsymbol{\psi}}_\omega = [\hat{\boldsymbol{h}}_\omega^T \dots \hat{\boldsymbol{h}}_\omega^T]^T \in \mathbb{R}^{MN}$ $(\hat{\boldsymbol{h}}_\omega^T \in \mathbb{R}^M)$ and that $\lim_{i\to\infty} \Theta_k(\boldsymbol{h}_{k,\omega}[i]) = 0$ for every $k \in \mathcal{N}$. If, for an interior $\tilde{\boldsymbol{u}}$ of Υ (which has an interior point because $C^\star$ does not line in a hyperplane), we have that

$$(\forall \epsilon > 0, \forall r > 0, \exists \xi_\omega > 0)$$
$$\inf_{i\in\mathcal{S}_\omega} \sum_{k\in\mathcal{N}} \Theta_k[i](\boldsymbol{h}_{k,\omega}[i]) \geq \sum_{k\in\mathcal{N}} \Theta_k^\star + \xi_\omega$$

where $\mathcal{S}_\omega := \{i \in \mathbb{N} \,|\, \sum_{k\in\mathcal{N}} d(\boldsymbol{h}_{k,\omega}[i], \text{lev}_{\leq\Theta_k^\star}\Theta_k[i]) > \epsilon$ and $\sum_{k\in\mathcal{N}} \|\tilde{\boldsymbol{u}} - \boldsymbol{h}_{k,\omega}[i]\| \leq r\}$, then $\hat{\boldsymbol{h}}_\omega$ satisfies $\hat{\boldsymbol{h}}_\omega \in \overline{\liminf_{i\to\infty} \Upsilon[i]}$, where $\liminf_{i\to\infty} \Upsilon[i] := \cup_{i=0}^{\infty} \cap_{n\geq i} \Upsilon[n]$ and the overbar operator denotes the closure of a set.

Proof: The proof builds on the results in [6], [13], [16] and is given in Appendix II. ∎

Recall that, when the local functions are spatially related, the problem in (3) is solved when the following properties are satisfied: 1) every local function is minimized and 2) the agents are in consensus ($\boldsymbol{h}_1[i] = \dots = \boldsymbol{h}_N[i]$). These two properties are satisfied asymptotically when we apply the proposed algorithm. More precisely, the local cost functions are asymptotically minimized with probability one [Theorem 2(b)] and agents reach mean square consensus [Theorem 2(c)]. In addition, Theorem 2(d)–(e) shows that agents reach consensus not only in the mean square sense, but also with probability one, and their estimates $\boldsymbol{h}_{k,\omega}[i]$ $(k \in \mathcal{N})$ converge to a point in $\overline{\liminf_{i\to\infty} \Upsilon[i]}$, which is the closure of the set of minimizers of all but finitely many global functions $\Theta[i]$ $(i \in \mathbb{N})$. This last property shows that the algorithm exploits the temporal structure of the sequence of optimization problems in (3). (Theorem 2(a) says that, if the local function is only spatially related at time i, in the mean square sense, the Euclidean distance of $[\boldsymbol{h}_1[i]^T \dots \boldsymbol{h}_N[i]^T]^T$ to a solution of (3) does not increase.)

Remark 1 (On Theorem 2):

1) The algorithm in Theorem 2 cannot be analyzed with the deterministic approach in [6] because the mapping $T : \mathbb{R}^{MN} \to \mathbb{R}^{MN}$ defined by $T(\boldsymbol{\psi}) = \boldsymbol{P}_\omega[i]\boldsymbol{\psi}$ is not necessarily nonexpansive, i.e., $\|T(\boldsymbol{x}) - T(\boldsymbol{y})\| \leq \|\boldsymbol{x} - \boldsymbol{y}\|$ does not necessarily hold. Nonexpansive mappings play a crucial role in the analysis of the algorithm in [6], and the scheme in Theorem 2 includes as a particular case the method in [6].

2) (*Asynchronous operation*) For simplicity, assume that agents want to minimize a time-invariant global function $\Theta(\boldsymbol{h}) = \sum_{k\in\mathcal{N}}\Theta_k(\boldsymbol{h})$, where $\Theta_1,\ldots,\Theta_N$ are spatially related. Suppose that agents are not synchronized and they do not necessarily apply the first step of the algorithm, the subgradient updates in (8), all at the same time. To model this scenario, we can consider that agent k only applies the subgradient method at time instants $i \in \mathcal{I}_k \subset \mathbb{N}$, where $\mathcal{I}_k$ is an infinite set. In doing so, we can consider that agents are using the following sequence of spatially and temporally related local functions:

$$\tilde{\Theta}_k[i](\boldsymbol{h}) = \begin{cases} \Theta_k(\boldsymbol{h}), & i \in \mathcal{I}_k \\ \Theta_k^\star, & \text{otherwise.} \end{cases}$$

Note that, with the time-varying local functions defined above, for every $l \in \mathbb{N}$, the set of minimizers of the original function Θ can be equivalently expressed as $\Upsilon = \cap_{i\geq l}\Upsilon[i]$, where $\Upsilon[i]$ is the set of minimizers of $\tilde{\Theta}[i]$. Likewise, if agents also do not want to exchange information [as in (9)] whenever a subgradient update is applied, we can use the following sequence of random matrices:

$$\tilde{\boldsymbol{P}}[i] = \begin{cases} \boldsymbol{P}[i], & i \in \mathcal{I}_{\boldsymbol{P}} \\ \boldsymbol{I}, & \text{otherwise} \end{cases}$$

where $\mathcal{I}_{\boldsymbol{P}} \subset \mathbb{N}$ is an infinite set that shows time instants in which information is exchanged among agents, and $\boldsymbol{P}[i]$ is the random matrix corresponding to the communication scheme. By applying $\tilde{\Theta}[i]$ and $\tilde{\boldsymbol{P}}[i]$ to the algorithm in Theorem 2, we conclude that, with probability one, agents produce sequences $\{\boldsymbol{h}_k[i]\}$, $k \in \mathcal{N}$, that converge to a common point in $\cup_{i=0}^{\infty}\cap_{n\geq i}\Upsilon[n] = \cup_{i=0}^{\infty}\Upsilon = \Upsilon$, which, as discussed above, is the set of minimizers of the global function Θ.

3) (*Adding constraints*) Constraints can also be easily added by considering time-varying cost functions. For example, with the assumptions in Theorem 2(b), let $\Theta_k : \mathbb{R}^M \to [0,\infty)$ be a (fixed) cost function known by agent k. Suppose that the agent has knowledge of a set C such that $\Upsilon \subset C$. Then we can use the following time-varying cost function instead of the original function $\Theta_k : \mathbb{R}^M \to [0,\infty)$:

$$\tilde{\Theta}_k[i](\boldsymbol{h}) = \begin{cases} \Theta_k(\boldsymbol{h}), & i \text{ odd} \\ d(\boldsymbol{h}, C) + \Theta_k^\star, & i \text{ even} \end{cases}$$

Applying the proposed method to $\tilde{\Theta}_k[i]$ and using similar arguments to those in Remark 1.2, we conclude that every agents will find a common point that satisfies all constraints and minimizes every local function.

IV. Possible Applications

In this section, we specialize the scheme in Theorem 2 to derive new distributed algorithms for acoustic source localization (Section IV-A) and for environmental modeling (Section IV-B). In the acoustic source localization problem, we show that batch incremental methods such as those in [3] can be easily modified to become adaptive, parallel algorithms operating with gossip networks and with mobile sensors. In the environmental modeling problem, we show that existing distributed set-theoretic adaptive filters can also be straightforwardly extended to gossip networks. In both applications, in ideal scenarios, the convergence properties of these particular cases of our general optimization algorithm follow directly from Theorem 2. (This is in stark contrast with many existing distributed adaptive algorithms, which are typically devised to solve specific problems, such as, for example, system identification with linear filters.) We also show that, in practice, these particular cases of our general method can have good performance even when many assumptions of Theorem 2 are just rough approximations.

A. Coordinated Acoustic Source Localization

1) Problem Description and Existing Solutions: The objective is to estimate the unknown location $\boldsymbol{x}^\star \in \mathbb{R}^2$ of an acoustic source with N agents distributed at spatial locations $\boldsymbol{x}_k \in \mathbb{R}^2$ $(k = 1,\ldots,N)$. (We later extend this problem to the case where agents are mobile.) Each agent knows its own position $\boldsymbol{x}_k$ and the acoustic source power A.[5] In addition, agents are equipped with an acoustic sensor, so they also know the sound intensity at their position. (With this information, the agents can estimate the range of the acoustic source from the received volume, but not the direction.) In more detail, the acoustic power perceived by agent k can be modeled as [31]

$$y_k = \frac{A}{||\boldsymbol{x}_k - \boldsymbol{x}^\star||^2} + n_k \tag{14}$$

where n_k is a noise sample. For mathematical simplicity, noise is often modeled as Gaussian, even though this assumption is unrealistic because y_k should be always positive. Nonetheless, algorithms using this unrealistic assumption often give good performance when deployed in real-world scenarios [31].

Given the statistical distribution of the noise, we can estimate the position of the target with the maximum-likelihood approach [31]. However, in this application the likelihood function is not a concave/convex function, so computing a global maximum/minimum may not be an easy task. To devise simple decentralized algorithms, we can consider the following convex optimization problem [3]:

$$\boldsymbol{x}_{\text{opt}} \in \arg\min_{\boldsymbol{h}\in\mathbb{R}^2} \sum_{k=1}^{N} d(\boldsymbol{h}, D_k) \tag{15}$$

where D_k is given by $D_k := \{\boldsymbol{h} \in \mathbb{R}^2 \,|\, ||\boldsymbol{h} - \boldsymbol{x}_k|| \leq \sqrt{A/y_k}\}$. When noise is not present, the solution set of the optimization problem in (15) is $\cap_{k=1}^{N} D_k \ni \boldsymbol{x}^\star$. If the acoustic source position $\boldsymbol{x}^\star$ lies in the convex hull of the agents' locations, i.e., $\boldsymbol{x}^\star \in H$ where

$$H = \left\{ \boldsymbol{x} \in \mathbb{R}^2 \,\middle|\, \boldsymbol{x} = \sum_{k=1}^{N} \alpha_k \boldsymbol{x}_k,\ \alpha_k \geq 0, \sum_{k=1}^{N} \alpha_k = 1 \right\} \tag{16}$$

then the unique point in $\cap_{k=1}^{N} D_k$, the solution to the problem in (15), is $\boldsymbol{x}^\star = \boldsymbol{x}_{\text{opt}}$ [3]. The incremental projection-onto-convex-sets (POCS) algorithm [3], a low-complexity, decentralized algorithm, can thus be used to solve (15) in this scenario (this method has some parameters that can be adjusted to deal with noise). However, this algorithm requires the definition of a path visiting all agents in the system, which is a difficult task in

[5] We can use the same techniques developed in [3] to extend the proposed algorithm to the case where A is unknown. For brevity, we do not consider such extensions here.

large-scale networks. Furthermore, during the iteration process, new measurements can be available to the agents, but the incremental POCS algorithm does not use such information. An additional limitation of this algorithm is that it does not consider mobile agents.

2) Proposed Algorithm: To derive our proposed algorithm, we first start by introducing the time-varying cost function that we minimize asymptotically. We start by assuming that agents are mobile and that they constantly take new samples of the acoustic sound intensity. Therefore, to model this dynamic scenario, we replace the model in (14) by

$$y_k[i] = \frac{A}{||\boldsymbol{x}_k[i] - \boldsymbol{x}^\star||^2} + n_k[i] \tag{17}$$

where $y_k[i]$, $\boldsymbol{x}_k[i]$, and $n_k[i]$ are, respectively, the acoustic sound intensity, the position of the kth agent, and the noise sample of agent k, all at time i. Agents take samples of the acoustic sound intensity at different positions, so they have access to samples with varying signal-to-noise ratio (SNR) (which is high in positions close to the acoustic source). Therefore, as many samples are available to estimate the position of the acoustic source in every agent, here we use those with potentially high SNR. In more detail, we keep in the memory of each agent only the largest observed sample $y_k[i]$ and the corresponding position $\boldsymbol{x}_k[i]$ (up to time i).[6] The index of this sample can be mathematically expressed by $l_k[i] = \arg\max_{l\in\{0,1,\ldots,i\}} y_k[l]$. For notational simplicity, hereafter we denote $l_k[i]$ by l_k, and the dependence of l_k with i is implicit. Now, consider the following (time-varying) set in agent k:

$$D_k[i] := \begin{cases} \mathbb{R}^2 & \text{if } y_k[l_k] \le c_k[i], \\ \left\{\boldsymbol{h} \in \mathbb{R}^2 \,|\, ||\boldsymbol{h} - \boldsymbol{x}_k[l_k]|| \le \sqrt{\frac{A}{y_k[l_k]-c_k[i]}}\right\} & \text{otherwise} \end{cases},$$

where $c_k[i]$ is a parameter used to increase the reliability of the sphere $D_k[i]$ when noise is present. In the noiseless case, we can use the same arguments used after (15) to conclude that the position of the acoustic source satisfies $\boldsymbol{x}^\star \in \cap_{k\in\mathcal{N}} D_k[i]$ for every $i \in \mathbb{N}$ and $c_k[i] \in [0,\infty)$. In this scenario, at time i, the set $\cap_{k\in\mathcal{N}} D_k[i] \ni \boldsymbol{x}^\star$ is also the solution set of $\arg\min_{\boldsymbol{h}\in\mathbb{R}^2} \sum_{k\in\mathcal{N}} \Theta_k[i](\boldsymbol{h})$, where the local (time-varying) functions are given by

$$\Theta_k[i](\boldsymbol{h}) = ||\boldsymbol{h} - P_{D_k[i]}(\boldsymbol{h})||. \tag{18}$$

Therefore, in ideal scenarios, the local functions in (18) are spatially and temporally related because $\Theta[i](\boldsymbol{x}^\star) = 0$ for every $i \in \mathbb{N}$. In particular, if $c_k[i] = 0$ and $\boldsymbol{x}^\star$ belongs to the convex hull defined by the positions $\boldsymbol{x}_k[l_k]$ ($k \in \mathcal{N}$), then we also have that $\boldsymbol{x}^\star$ is the only point in the intersection $\cap_{k\in\mathcal{N}} D_k[i]$. If noise is present, we can increase the radius of the spheres $D_k[i]$ by increasing the parameter $c_k[i]$ to guarantee that $\Theta[i](\boldsymbol{x}^\star) = 0$ (or, equivalently, $\boldsymbol{x}^\star \in \cap_{k\in\mathcal{N}} D_k[i]$) with high probability. However, later we show that in practice the resulting algorithm works well even with $c_k[i] = 0$ in the presence of noise. The main idea of the proposed method for acoustic source localization is thus to use the scheme in Theorem 2 to minimize asymptotically $\Theta[i]$ and to find a fixed point that minimizes as many global functions $\Theta[i]$ as possible. Such a solution is expected to be a good estimate of $\boldsymbol{x}^\star$ because $\boldsymbol{x}^\star$ is a minimizer of every global function at any time instant, i.e., $\Theta[i](\boldsymbol{x}^\star) = 0$ for every $i \in \mathbb{N}$.

Having defined the sequence of global functions to be minimized asymptotically, we now turn our attention to the communication model. Owing to the nature of wireless channels, if agent k broadcasts an estimate $\boldsymbol{h}_k[i]$, all other agents within a certain distance are able to receive this information. To exploit this physical characteristic of wireless channels, we use the communication model in [20]. In more detail, we assume that, at each iteration, only agent k, selected uniformly at random, broadcasts its estimate $\boldsymbol{h}_k[i]$. Then all agents within range R, i.e., all agents in the set

$$\mathcal{N}_k[i] := \{j \in \mathcal{N} \,|\, ||\boldsymbol{x}_k[i] - \boldsymbol{x}_j[i]|| \le R\} \tag{19}$$

mix their estimates with that received from agent k. To be more precise, given that agent k has been selected at time i in realization ω, we express this communication model as in (9) by using the following matrix $\boldsymbol{W}_{kj,\omega}[i]$:

$$\boldsymbol{W}_{jl,\omega}[i] = \begin{cases} \boldsymbol{I}, & j \notin \mathcal{N}_k[i]\backslash\{k\} \text{ and } j = l \\ \gamma\boldsymbol{I}, & j \in \mathcal{N}_k[i]\backslash\{k\} \text{ and } j = l \\ \boldsymbol{I} - \gamma\boldsymbol{I}, & j \in \mathcal{N}_k\backslash\{k\} \text{ and } l = k \\ \boldsymbol{0}, & \text{otherwise} \end{cases} \tag{20}$$

where $\gamma \in (0,1)$ is a mixing parameter. If the communication range R is long enough so that the graphs $\mathcal{G}[i]$ with the neighboring rule in (19) are (strongly) connected, then the matrices $\boldsymbol{P}[i]$ ($i \in \mathbb{N}$), which are random block matrices having $\boldsymbol{W}_{kj}[i]$ as submatrices [see (10)], are ϵ-random matrices for some $\epsilon > 0$. This fact can be proven with the results in [20] and the references therein. We omit the details for brevity.

Applying the local cost functions in (18) and the communication model in (20) to the scheme in (11), we arrive at the following algorithm:[7]

Algorithm 1: (*Proposed algorithm*)**:**

1) Initialize the estimates $\boldsymbol{h}_k[i]$ with an arbitrary $\boldsymbol{h}_k[0] \in \mathbb{R}^2$.
2) Move all agents and take new samples of the acoustic sound intensity.
3) Keep in the memory of each agent the largest sample observed so far and its corresponding position (the sample and position are denoted by, respectively, $y_k[l_k]$ and $\boldsymbol{x}_k[l_k]$).
4) Agents apply the subgradient update defined in (8): for all $k \in \mathcal{N}$

$$\boldsymbol{h}'_k[i+1] = \boldsymbol{h}_k[i] + \mu_k[i]\left(P_{D_k[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\right)$$

where $\mu_k[i] \in (0,2)$ is the step size and

$$P_{D_k[i]}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} & \text{if } \boldsymbol{h} \in D_k[i] \\ \boldsymbol{x}_k[l_k] + \sqrt{\frac{A}{y_k[l_k]-c_k[i]}}\frac{(\boldsymbol{h}-\boldsymbol{x}_k[l_k])}{||\boldsymbol{h}-\boldsymbol{x}_k[l_k]||} & \text{otherwise.} \end{cases}$$

5) Choose $m \in \mathcal{N}$ uniformly at random.
6) Agent m broadcasts $\boldsymbol{h}'_m[i+1]$.

[6] We could easily derive variations where only the largest sample within an interval is used. This idea could be useful to track mobile acoustic sources.

[7] A subgradient of $\Theta_k[i]$ at $\boldsymbol{h}$ is $\Theta'_k[i](\boldsymbol{h}) = (\boldsymbol{h} - P_{D_k[i]}(\boldsymbol{h}))/||\boldsymbol{h} - P_{D_k[i]}(\boldsymbol{h})||$ [13].

7) Agents within distance R to agent m mix the received estimate $\boldsymbol{h}'_m[i+1]$ with their own estimates $\boldsymbol{h}_j[i]$:

$$\boldsymbol{h}_j[i+1] = \begin{cases} \gamma \boldsymbol{h}'_j[i+1] + (1-\gamma)\boldsymbol{h}'_m[i+1], \\ \qquad \text{if } j \in \mathcal{N}_m[i] \backslash \{m\}, \\ \boldsymbol{h}'_j[i+1], \qquad \text{otherwise} \end{cases}$$

where $\gamma \in (0,1)$ is a mixing parameter common to all agents.

8) Increment i and go to step 2.

Note that Algorithm 1 requires neither simultaneous information exchange nor agents to be aware of the position or the number of their neighbors. In addition, unlike incremental methods, we also do not need to define a path visiting all agents.

3) Numerical Simulations: In a 100 m $\times$ 100 m field, at each realization of the simulation we randomly distribute 36 agents and place an acoustic source with $A = 100$ at $\boldsymbol{x}^\star = [50\ 50]^T$. Each agent measures the acoustic power at their own locations according to (17). The noise samples $n_k[i]$ are i.i.d. and drawn from a Gaussian distribution with variance $\sigma_k = 1$ and mean zero. For simplicity, to obtain the samples $y_k[i]$ at time i, agents choose positions $\boldsymbol{x}_k[i]$ uniformly at random within the region of interest.

We simulate two different versions of the proposed algorithm (Proposed-1 and Proposed-2) that differ in the choice of the parameter $c_k[i]$. In more detail, Proposed-1 uses $c_k[i] = 0$, and Proposed-2 uses $c_k[i] = 4\sigma_k$ (this last value guarantees that $\boldsymbol{x}^\star \in D_k[i]$ with high probability and that the radius of the sphere $D_k[i]$ is not excessively increased when samples $y_k[i]$ are taken close to the acoustic source location). Other parameters are equal in both Proposed-1 and Proposed-2: $\mu_k[i] = 1$, $R = 30$, and $\gamma = 0.5$. We compare the proposed method with the incremental POCS algorithm [3], which is the algorithm we build on to derive the proposed adaptive method. The incremental POCS algorithm uses fixed agents (i.e., $r_k[i] = r_k[0]$ for all i and k) and just a single sample of acoustic sound intensity to estimate the acoustic source location. In this algorithm, agents are activated using a greedy rule: from all agents not previously selected in a cycle, the next agent in the cycle is the one closest to the current agent.[8] To mitigate noise, we set the step size of the incremental POCS algorithm to 0.2.

The performance of interest is the average mean square error (MSE) of the agents

$$\text{MSE}[i] = E\left[\frac{1}{N}\sum_{k=1}^{N} \|\boldsymbol{h}_k[i] - \boldsymbol{x}^\star\|^2\right].$$

We compute expectations by averaging the results of 100 realizations of the simulation. Fig. 1 shows the simulation results.

We can see that both proposed algorithms greatly decrease the estimation error compared to the incremental POCS algorithm. The superior performance of the proposed methods is explained by two facts: 1) agents are mobile, so they can take samples close the acoustic source; and 2) agents can choose a suitable cost function as data becomes available.[9]

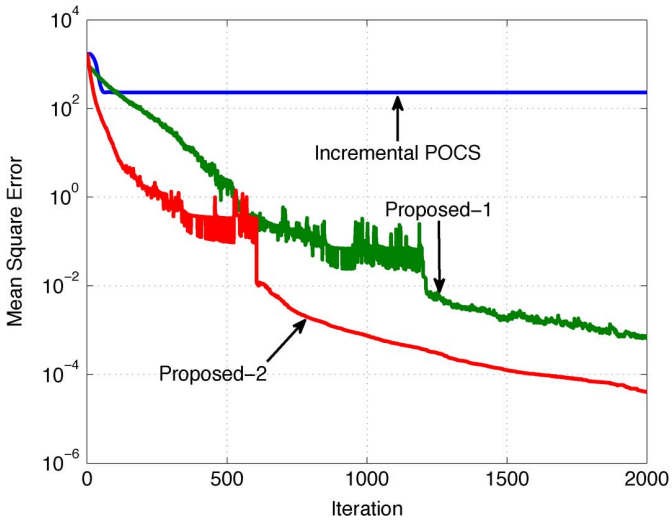

Fig. 1. Transient performance of the algorithms.

An additional good feature of the proposed algorithm is that it does not require the definition of a path visiting all agents in the system. Agents are randomly selected, broadcast their estimates, and only those agents within the communication range mix estimates. No feedback is necessary, so agents can ignore the position and the number of neighbors. In many applications, this communication model could be enough to justify the use of the proposed method over incremental methods (even if the performance of the proposed method were inferior to that of incremental methods). The reason is that acquiring a path visiting all agents is a difficult problem to solve in large-scale networks, and the proposed method does not need to solve such problems.

The performance of Proposed-2 is better than that of Proposed-1 because the former expands the sets $D_k[i]$, thus increasing the probability that $\boldsymbol{x}^\star \in \cap_{k\in\mathcal{N}} D_k[i]$. Note that the parameter $c_k[i]$ in Proposed-2 does not unduly increase the "size" of $\cap_{k\in\mathcal{N}} D_k[i]$. (Sets $\cap_{k\in\mathcal{N}} D_k[i]$ that are too large can give poor estimates because not all points in these sets are necessarily close to $\boldsymbol{x}^\star$.) The jumps and the initial unsteady behavior shown by the MSE curves of the proposed algorithms are explained by the fact that agents take samples at random locations. Therefore, until samples with sufficiently high SNR are obtained in every agent, the sets $D_k[i]$ are not reliable, and the subgradient updates can unduly increase the estimation error. Note that the sets used by Proposed-2 are always more reliable than those used by Proposed-1 because of the larger expansion factor $c_k[i]$, and this fact explains why the unsteady behavior of Proposed-1 lasts longer than that of Proposed-2.

B. Environmental Modeling

1) Problem Description: Suppose that a physical phenomenon (e.g., temperature, salinity, density of adversarial agents,

[8]If in the simulation we have that $y_k < 0$ (not physically possible, but it can happen in the simulation because of the acoustic model we adopted), then the corresponding agent simply sends the estimate of the previous agent of the cycle to the next agent in the cycle.

[9]In contrast, batch methods, such as the incremental POCS algorithms, consider fixed sets/cost functions, so, formally, they cannot incorporate new information obtained by taking samples at different positions if the algorithm has already started to run.

etc. [32]) in a region of interest $\mathcal{D} \subset \mathbb{R}^D$ is expressed by a function $g : \mathcal{D} \to \mathbb{R}$ that can be well approximated by $f : \mathcal{D} \to \mathbb{R}$ defined as follows:

$$f(\boldsymbol{x}) := \boldsymbol{\alpha}^T \boldsymbol{\Phi}(\boldsymbol{x}) \approx g(\boldsymbol{x}) \tag{21}$$

where $\boldsymbol{x} \in \mathcal{D}$ is a spatial coordinate, $\boldsymbol{\Phi}(\boldsymbol{x}) := [\phi_1(\boldsymbol{x}) \dots \phi_L(\boldsymbol{x})]^T \in \mathbb{R}^L$, $\phi_n : \mathbb{R}^D \to \mathbb{R}$ is the nth basis function (e.g., Fourier series, wavelets, radial basis functions, etc.), $\boldsymbol{\alpha} := [\alpha_1 \dots \alpha_L]^T \in \mathbb{R}^L$, and α_n is the coefficient associated with the nth basis function (see also [32]). If, for example, we use a large enough number of properly selected radial basis functions to build $\boldsymbol{\Phi}(\boldsymbol{x})$, the universal approximation theorem [33, Sec. 20.6] justifies the approximation in (21). We assume that the bases $\phi_n : \mathbb{R}^D \to \mathbb{R}$ $(n = 1, \dots, L)$ are fixed and known by all agents, which form a network associated with a graph $\mathcal{G}[i] = (\mathcal{N}, \mathcal{E}[i])$ as described in Section II-B. In addition, we also assume that agent k can observe noisy samples $y_k[i] \in \mathbb{R}$

$$y_k[i] := g(\boldsymbol{x}_k[i]) + n_k[i] \approx f(\boldsymbol{x}[i]) + n_k[i] \tag{22}$$

where $\boldsymbol{x}_k[i] \in \mathcal{D}$ and $n_k[i] \in \mathbb{R}$ are, respectively, the position and the noise sample of agent k at time i. The environmental modeling problem amounts to estimating $\boldsymbol{\alpha}$ in (21) in all agents from the samples $y_k[i]$ $(k \in \mathcal{N})$, which are dispersed throughout the network. Note that, by knowing $\boldsymbol{\alpha}$, agents have complete information about the physical phenomenon in the region of interest.

Having described the estimation problem, we now turn to the proposed distributed algorithm. In our method, agents communicate asynchronously and do not have access to the location, number, or samples $y_k[i]$ of their neighbors.

2) Set-Theoretic Adaptive Algorithms for Environmental Modeling: We start by considering an ideal scenario; suppose that there exists $\boldsymbol{\alpha} \in \mathbb{R}^L$ such that $\boldsymbol{\alpha}^T \boldsymbol{\Phi}(\boldsymbol{x}) = g(\boldsymbol{x})$ for every coordinate $\boldsymbol{x} \in \mathcal{D}$, and no noise is present in the measurements $y_k[i]$ $(k \in \mathcal{N}, i \in \mathbb{N})$. As a result, we have that, for every $k \in \mathcal{N}$, $i \in \mathbb{N}$, and $\boldsymbol{x}_k[i] \in \mathcal{D}$

$$\boldsymbol{\alpha} \in F_k[i] := \{\boldsymbol{h} \in \mathbb{R}^L \,|\, \boldsymbol{h}^T \boldsymbol{\Phi}(\boldsymbol{x}_k[i]) = y_k[i]\}.$$

Therefore, if $\boldsymbol{\alpha}$ is to be estimated in this ideal scenario, a good estimate should also belong to $F_k[i]$ for any $k \in \mathcal{N}$, $i \in \mathbb{N}$, $\boldsymbol{x}_k[i] \in \mathcal{D}$. To handle non-ideal scenarios, we can use the following relaxation $G_k[i]$ of $F_k[i]$:

$$G_k[i] := \{\boldsymbol{h} \in \mathbb{R}^L \,|\, |\boldsymbol{h}^T \boldsymbol{\Phi}(\boldsymbol{x}_k[i]) - y_k[i]| \leq \xi_k[i]\} \tag{23}$$

where $\xi_k[i] \geq 0$ is a suitably chosen relaxation parameter of agent k at time i (to avoid clutter, we omit the dependence of $G_k[i]$ with $\xi_k[i]$). In more detail, the parameter $\xi_k[i]$ serves two purposes. First, it increases the probability that $\boldsymbol{\alpha} \in G_k[i]$ in noisy environments. Second, it is used to take into account the fact that the existence of $\boldsymbol{\alpha} \in \mathbb{R}^L$ satisfying the equality $\boldsymbol{\alpha}^T \boldsymbol{\Phi}(\boldsymbol{x}) = g(\boldsymbol{x})$ for every $\boldsymbol{x} \in \mathcal{D}$ is questionable because, in the domain $\mathcal{D}$, the function g may not be equivalently expressed by a linear combination of the basis functions $\phi_1, \dots, \phi_L$. In such a case, we could, for example, redefine the desired estimand $\boldsymbol{\alpha}$ as any vector in $\mathbb{R}^L$ such that f in (21) reproduces g with an uniform tolerance $\epsilon > 0$ in the region of interest, i.e., $\boldsymbol{\alpha} \in \{\boldsymbol{h} \in \mathbb{R}^L \,|\, -\epsilon \leq \boldsymbol{h}^T \boldsymbol{\Phi}(\boldsymbol{x}) - g(\boldsymbol{x}) \leq \epsilon, \boldsymbol{x} \in \mathcal{D}\}$ (this set is nonempty provided that ϵ is large enough). Therefore, if the relaxation parameter $\xi_k[i]$ is sufficiently large, we have that $\boldsymbol{\alpha} \in G_k[i]$ (in the simulations we show that the algorithm can work well even with $\xi_k[i] = 0$ in non-ideal scenarios). At time index i, reasonable estimates of $\boldsymbol{\alpha}$ should then belong to

$$\mathcal{C}[i] := \bigcap_{n \in \mathcal{I}[i]} \bigcap_{k \in \mathcal{N}} G_k[n] \ni \boldsymbol{\alpha} \tag{24}$$

where $\mathcal{I}[i]$ is a properly chosen subset of time indices of available measurements $y_k[i]$ (i.e., $\mathcal{I}[i] \subset \{0, 1, \dots, i\}$). Intuitively, $\mathcal{C}[i]$ is the set of estimates of $\boldsymbol{\alpha}$ that are consistent with all measurements $y_k[n]$, $k \in \mathcal{N}$ and $n \in \mathcal{I}[i]$. The set $\mathcal{C}[i]$ can be time-varying because $\mathcal{I}[i]$ is allowed to change from iteration to iteration. This time-varying property of $\mathcal{I}[i]$ (and, consequently, of $\mathcal{C}[i]$) can be used to incorporate information gained by measurements $y_k[i]$ (represented by sets $G_k[i]$) as they become available. The choice of $\mathcal{I}[i]$ should take into account the desired complexity of the algorithm and the time in which the environment, described by the function g, can be considered approximately static. Having defined $\mathcal{C}[i]$ in (24) as the set of reasonable estimates of $\boldsymbol{\alpha}$ at time i, we now proceed to construct convex cost functions having $\mathcal{C}[i]$ as the set of minimizers, and then we apply the scheme in Theorem 2 to derive low-complexity algorithms that minimize these time-varying cost functions asymptotically.

The parameter $\boldsymbol{\alpha}$ in (21) can be seen as a linear filter [33], [34], so we can use the cost functions of existing set-theoretic linear adaptive filters (e.g., the affine projection algorithm [13], [35]–[37], the normalized least-mean-square algorithm [13], [37], [38], etc.) to estimate $\boldsymbol{\alpha}$. In doing so, we can extend these approaches to distributed networks with random links. In particular, here we use the following local cost function [13]:

$$\Theta_k[i](\boldsymbol{h}) = \sum_{j=0}^{q[i]-1} c_k[i,j] \left\| \boldsymbol{h} - P_{G_k[i-j]}(\boldsymbol{h}) \right\| \tag{25}$$

where $q[i] \in \mathbb{N}$ is the number of the most recent samples $y_k[i]$ used by the agents, $c_k[i,j]$ is a constant given by[10]

$$c_k[i,j]] = \begin{cases} \frac{w_k[i,j]}{L_k[i]} \left\| \boldsymbol{h}_k[i] - P_{G_k[i-j]}(\boldsymbol{h}_k[i]) \right\|, & \text{if} \quad L_k[i] = 0 \\ 0, & \text{otherwise} \end{cases}$$

$L_k[i]$ is defined by $L_k[i] := \sum_{j=0}^{q[i]-1} w_k[i,j] \|\boldsymbol{h}_k[i] - P_{G_k[i-j]}(\boldsymbol{h}_k[i])\|$, and $w_k[i,j] > 0$ is a weighting factor of the set $G_k[i-j]$ and should satisfy $\sum_{j=0}^{q[i]-1} w_k[i,j] = 1$. Note that, if $\mathcal{C}[i] \neq \emptyset$ with $\mathcal{I}[i] = \{i - q[i] + 1, \dots, i\}$, then $\Theta[i](\tilde{\boldsymbol{\alpha}}) = \sum_{k \in \mathcal{N}} \Theta_k[i](\tilde{\boldsymbol{\alpha}}) = 0$ for any $\tilde{\boldsymbol{\alpha}} \in \mathcal{C}[i]$ and for any of the possible choices of weights $w_k[i,j]$. In particular, in the ideal scenario described above, $\Theta[i](\boldsymbol{\alpha}) = 0$, which shows that the local functions are both spatially and temporally related. Therefore, we see that good estimates of $\boldsymbol{\alpha}$ should minimize as many global functions $\Theta[i]$ as possible (ideally, for all i) because $\boldsymbol{\alpha}$ is a point that minimizes every global function.

[10]The constant $c_k[i,j]$ is not necessarily the same at different time instants because $c_k[i,j]$ depends on the current estimate $\boldsymbol{h}_k[i]$.

The set of minimizers of $\Theta[i]$ may not depend on the possible choices of weights in an ideal scenario, so this fact may suggest that we should not pay any special attention to the choice of $w_k[i,j]$. However, by noticing that the environment g can be time-varying in real-world scenarios, in practice we may need to give large weights to sets $G_k[i]$ based on more recent samples $y_k[i]$. In doing so, by using the scheme in Theorem 2 with the local functions $\Theta_k[i]$ in (25), agents move their estimates $\boldsymbol{h}_k[i]$ to points closer to sets based on recent measurements $y_k[i]$ (i.e., sets $G_k[i]$ with large weight $w_k[i,j]$) than to sets based on old measurements. In addition, as shown in [6], [14], particular choices of weights $w_k[i,j]$ yield subgradient updates [defined in (8)] that are easy to implement even when the memory of the algorithm, represented by the parameter $q[i]$, grows unboundedly.

Having defined the cost functions to be minimized asymptotically, we now need to choose a communication model. For this application, we again use the simple communication model applied to the acoustic sensor localization problem. Briefly, we assume that agents within range R from each other are neighboring agents, and only one agent, selected uniformly at random, broadcasts its estimate $\boldsymbol{h}_k[i]$ to neighboring agents. Details have already been provided in the discussion before Algorithm 1. Applying this communication model with the local cost functions in (25) to the scheme in Theorem (2), we arrive at the algorithm described below.

Algorithm 2:

1) Initialize the estimates $\boldsymbol{h}_k[i]$ with an arbitrary $\boldsymbol{h}_k[0] \in \mathbb{R}^L$. Choose $q[i]$, which is the number of sets $G_k[i]$ used at each iteration of the algorithm, the expansion parameter $\xi_k[i]$ of $G_k[i]$, and the weights $w_k[i,j]$ $(j = 0, \ldots, q[i]-1)$.
2) Move all agents and take new samples $y_k[i]$.
3) Agents apply the subgradient update defined in (8):[11] for all $k \in \mathcal{N}$

$$\boldsymbol{h}'_k[i+1] = \boldsymbol{h}_k[i] + \bar{\mu}_k[i]\left(\sum_{j=0}^{q[i]-1} \omega_k[i,j] P_{G_k[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\right) \quad (26)$$

where $\bar{\mu}_k[i] \in [0, 2\mathcal{M}_k[i]]$ is the step size

$$\mathcal{M}_k[i] := \begin{cases} \dfrac{\sum_{j=0}^{q[i]-1} \omega_k[i,j] \left\| P_{G_k[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i] \right\|^2}{\left\| \sum_{j=0}^{q[i]-1} \omega_k[i,j] P_{G_k[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i] \right\|^2}, \\ \qquad \text{if } \boldsymbol{h}_k[i] \notin \bigcap_{j=0}^{q[i]-1} G_k[i-j] \\ 1, \qquad \text{otherwise,} \end{cases}$$

(NOTE: $\mathcal{M}_k[i] \geq 1$) and ([26, p. 99]

$$P_{G_k[i]}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} & \text{if } \boldsymbol{h} \in G_k[i] \\ \boldsymbol{h} + \dfrac{(y_k[i] - \xi_k[i]) - \boldsymbol{h}^T \boldsymbol{\Phi}(\boldsymbol{x}_k[i])}{\|\boldsymbol{\Phi}(\boldsymbol{x}_k[i])\|^2} \boldsymbol{\Phi}(\boldsymbol{x}_k[i]) \\ \qquad \text{if } \boldsymbol{h}^T \boldsymbol{\Phi}(\boldsymbol{x}_k[i]) < y_k[i] - \xi_k[i] \\ \boldsymbol{h} + \dfrac{(y_k[i] + \xi_k[i]) - \boldsymbol{h}^T \boldsymbol{\Phi}(\boldsymbol{x}_k[i])}{\|\boldsymbol{\Phi}(\boldsymbol{x}_k[i])\|^2} \boldsymbol{\Phi}(\boldsymbol{x}_k[i]) \\ \qquad \text{if } \boldsymbol{h}^T \boldsymbol{\Phi}(\boldsymbol{x}_k[i]) > y_k[i] + \xi_k[i]. \end{cases}$$

[11]The details of the derivation of (26), obtained by applying the subgradient update to the local cost function in (25), is shown in ([13], Example 3).

4) Choose $m \in \mathcal{N}$ uniformly at random.
5) Agent m broadcasts $\boldsymbol{h}'_m[i+1]$.
6) Agents within distance R to agent m mix the received estimate $\boldsymbol{h}'_m[i+1]$ with their own estimates $\boldsymbol{h}_j[i]$:

$$\boldsymbol{h}_j[i+1] = \begin{cases} \gamma \boldsymbol{h}'_j[i+1] + (1-\gamma)\boldsymbol{h}'_m[i+1], & \text{if } j \in \mathcal{N}_m[i] \backslash \{m\} \\ \boldsymbol{h}'_j[i+1], & \text{otherwise} \end{cases}$$

where $\gamma \in (0,1)$ is a mixing parameter common to all agents.
7) Increment i and go to step 2.

3) Numerical Simulations: In the simulation, we drop the assumption of static environments, and agents estimate the dynamic environment described by

$$g[i](\boldsymbol{x}) = \sin\left(2\pi \frac{x_1}{100} + 2\pi \frac{i}{2500}\right) + \cos\left(2\pi \frac{x_2}{100} + 2\pi \frac{i}{2500}\right)$$

where $\boldsymbol{x} := [x_1 x_2]^T \in \mathbb{R}^2$, $i \in \mathbb{N}$ is the discrete-time index, and $x_1, x_2 \in [0, 100]$ are spatial coordinates of the region of interest (a 100×100 square). (We use this particular function $g[i]$ to illustrate a scenario where the approximation in (21) is a rough approximation due to the choice of basis functions.) Agents use Gaussian radial basis functions

$$\phi_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_j\|^2}{2\sigma_w^2}\right), \quad j = 1, \ldots, L$$

where $L = 16$, $\boldsymbol{c}_j \in \mathbb{R}^2$ $(j = 1, \ldots, L)$ are centers distributed in the region of interest, and σ_w is the width of the radial basis functions. We subdivide the region of interest into 16 squares of the same area, and we place one agent in each subdivision $(\mathcal{N} = \{1, \ldots, L\})$. Each center $\boldsymbol{c}_k$ is located at the center of each subdivision, and we set $\sigma_w = 30/\sqrt{2}$, which is a value chosen to avoid basis functions that are too peaked or too flat in the region of interest. At time i, each agent k takes samples $y_k[i]$ according to (22), where the noise samples $n_k[i]$ are i.i.d. and drawn from a zero-mean Gaussian distribution with variance $\sigma_k[i]^2 = 0.3$, and $\boldsymbol{x}_k[i]$ is a position selected uniformly at random in the subdivision into which agent k is placed.

The parameters of the proposed algorithm are as follows: Proposed-1 $(q[i] = 1, \bar{\mu}_k[i] = 0.2, \gamma = 0.5, \xi_k[i] = 0, w_k[i,j] = 1)$, Proposed-2 $(q[i] = 1, \bar{\mu}_k[i] = 1, \gamma = 0.5, \xi_k[i] = \sigma_k[i], w_k[i,j] = 1)$, Proposed-3 $(q[i] = 8, \bar{\mu}_k[i] = 0.5, \gamma = 0.5, \xi_k[i] = 0, w_k[i,j] = 1/8)$, and Proposed-4 $(q[i] = 8, \bar{\mu}_k[i] = 1, \gamma = 0.5, \xi_k[i] = \sigma_k[i], w_k[i,j] = 1/8)$. Proposed-1 and Proposed-3 mitigate the effects of noise and modeling errors by using a relatively small step size $\bar{\mu}_k[i]$, whereas Proposed-2 and Proposed-4 mitigate those effects by increasing $\xi_k[i]$ (i.e., by increasing the reliability of the sets $G_k[i]$). Note that, in particular, Proposed-1 is an extension of the celebrated normalized least-mean-square algorithm [13], [33], [34], [37], [38] to distributed gossip networks. The communication range R of the agents in all proposed algorithms is $R = 50$. In these algorithms, agents use the same set of parameters, but such a choice is not a requirement. Agents using different sets of parameters can be useful in scenarios where the memory and computational power of the agents are different.

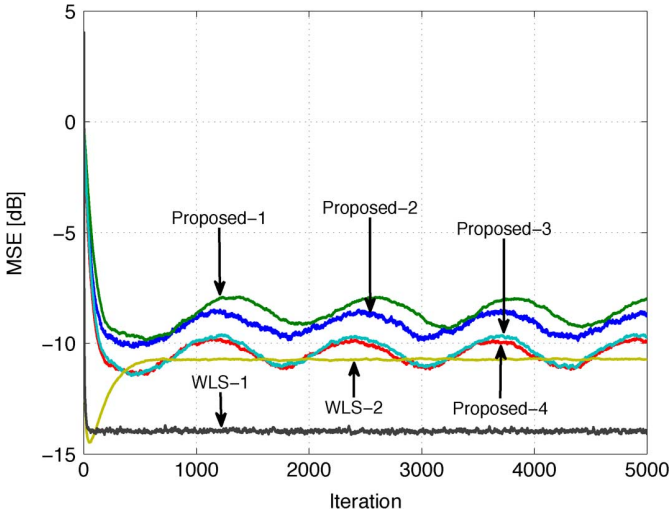

Fig. 2. Tracking performance of the algorithms.

We compare the proposed algorithms with a method where all agents use the solution of the following weighted least-squares fit problem:

$$\boldsymbol{h}_{\mathrm{LS}}[i] \in \arg\min_{\boldsymbol{h}\in\mathbb{R}^2} \left(\sum_{n=1}^{i} \beta_{\mathrm{FF}}^{i-n} \sum_{k\in\mathcal{N}} (y_k[n] - \boldsymbol{h}^T \boldsymbol{\Phi}(\boldsymbol{x}_k[n]))^2 + \delta_R \|\boldsymbol{h}\|^2 \right)$$

where $\beta_{\mathrm{FF}} \in (0,1]$ is a forgetting factor used to take into account the dynamic nature of the environment, and δ_R is a regularization factor. This algorithm, hereafter denominated weighted least-squares (WLS) algorithm, can be implemented if there is an all-to-all communication among agents in every iteration, or if all agents have a bi-directional link with a center fusion. Therefore, the WLS algorithm is ignoring the assumptions of the multiagent system, which we require to be non-hierarchical and to have sparse communication among agents. In the simulations we use two versions of the WLS algorithm: WLS-1 ($\beta_{\mathrm{FF}} = 0.92, \delta_R = 10^{-6}$) and WLS-2 ($\beta_{\mathrm{FF}} = 0.99, \delta_R = 10^{-6}$).

The goal of every agent is to estimate the time-varying function $g[i]$ in the region of interest (the 100×100 field); thus, given the estimates $\boldsymbol{h}_k[i]$ ($k \in \mathcal{N}$) at time i, we use as the performance metric a normalized sum of the mean-square error (MSE) of the agents

$$\frac{\sum_{k\in\mathcal{N}} E\left[\int_0^{100}\int_0^{100} |g[i](\boldsymbol{x}) - \boldsymbol{h}_k[i]^T \boldsymbol{\Phi}(\boldsymbol{x})|^2 \mathrm{d}x_1 \mathrm{d}x_2\right]}{|\mathcal{N}| \cdot \int_0^{100}\int_0^{100} |g[i](\boldsymbol{x})|^2 \mathrm{d}x_1 \mathrm{d}x_2}$$

where expectations are computed from ensemble averages of 100 realizations of the simulation, and integrals are evaluated numerically. (In practice, computing the filter that minimizes the MSE is not possible because perfect knowledge of $g[i]$ is required.) Fig. 2 shows the performance of the algorithms.

The two versions of the WLS algorithm have the best performance because the WLS algorithm can be considered as a centralized method, and, as such, it should be used only as a reference of the best performance that can be achieved by the proposed algorithm. The performance of WLS-2 is inferior to that of WLS-1 because WLS-2 weights heavily old measurements $y_k[i]$ (the parameter β_{FF} of WLS-2 is larger than that of WLS-1) and the environment is dynamic. Proposed-1 and Proposed-2 use only the most recent measurement $y_k[i]$ at every iteration, so it is not surprising that they have the worst performance. However, these two algorithms have the lowest computational complexity of all compared algorithms. The computational complexity of Proposed-1 and Proposed-2 is $\mathcal{O}(L)$ (per agent), and the better performance of the latter is due to the larger relaxation parameter $\xi_k[i]$, which mitigates the detrimental effects of noise and modeling errors. Proposed-3 and Proposed-4 have better performance than Proposed-1 and Proposed-2 because Proposed-3 and Proposed-4 use more information at each iteration (measurements $y_k[i]$). The slightly superior performance of Proposed-4 compared to that of Proposed-3 is again explained by the larger relaxation parameter $\xi_k[i]$ of Proposed-4. In terms of computational complexity, note that the subgradient updates in Proposed-3 and Proposed-4 can be parallelized in operations of complexity $\mathcal{O}(L)$ (per agent) [39].

V. Conclusion

We have developed a non-hierarchical algorithm that minimizes asymptotically a global function defined by the sum of convex functions. Each term in this sum is a local cost function known by an agent in a network, and we assume that the sets of optimizers of the local functions have nonempty intersection. Unlike existing optimization methods, the local cost functions can be time-varying, and agents exchange information locally via network gossiping. This mechanism for information exchange enable us to relax the assumption of simultaneous information exchange among agents, a common assumption in the analysis of multiagent algorithms using subgradient methods. We showed conditions to guarantee almost sure asymptotic minimization of the local cost functions, consensus among agents, and convergence. We provided examples of applications where the algorithm in its most general form was specialized to handle specific problems. In more detail, we applied the proposed method to derive new adaptive algorithms for acoustic source localization and for environmental modeling. In the former application, agents estimate the position of the acoustic source directly; in the latter application, agents estimate a physical phenomenon (temperature, salinity, density of adversarial agents, etc.) by trying to reach consensus on coefficients that define the environment. These applications show techniques that can be applied when the assumptions of Theorem 2 are rough approximations, and they also show how to extend existing adaptive or batch projection-based methods to distributed networks with random links.

Appendix I
Proof of Lemma 1

For $i \geq j$ with $j \in \mathbb{N}$, there are at least $\lfloor (i-j)/(I+1) \rfloor$ distinct indices k within the interval $[j, i]$, where $0 \leq \lambda[k] \leq \beta$;

hence, $\prod_{n=0}^{i-j}\lambda[i-n] \leq \beta^{\lfloor (i-j)/(I+1)\rfloor}$. We also know that $\{\sum_{j=0}^{i}\prod_{n=0}^{i-j}\lambda[i-n]\}$ is bounded because, for every $i\in\mathbb{N}$

$$0\leq \sum_{j=0}^{i}\prod_{n=0}^{i-j}\lambda[i-n]\leq \sum_{j=0}^{i}\beta^{\lfloor (i-j)/(I+1)\rfloor}\leq \frac{I+1}{1-\beta}=:S. \tag{27}$$

From this moment the proof resembles that of Mertens' theorem [40]. The convergence of $\{a[i]\}$ to zero implies that, for every $\epsilon>0$, there exists $K\in\mathbb{N}$ such that for $i\geq K$ we have

$$|a[i]|<\frac{\epsilon}{2S} \tag{28}$$

where S is as defined in (27). For this K and any $j\leq K$ ($j\in\mathbb{N}$), we conclude from $\lim_{i\to\infty}\prod_{n=0}^{i-j}\lambda[i-n]=0$ that there exists $L>K$ such that, for every $i\geq L$

$$\prod_{n=0}^{i-j}\lambda[i-n]<\frac{\epsilon}{2KB} \tag{29}$$

where $B>0$ can be any (nonzero) upper bound of the convergent sequence $\{|a[i]|\}$ (e.g., $B=\sup_i(|a[i]|)+1$). Therefore, for $i\geq L$, by (27), (28), (29), and the triangle inequality

$$\begin{aligned}&\left|\sum_{j=0}^{i}a[j]\prod_{n=0}^{i-j}\lambda[i-n]\right|\\&\quad\leq\sum_{j=0}^{i}\left|a[j]\prod_{n=0}^{i-j}\lambda[i-n]\right|\\&\quad=\sum_{j=0}^{K-1}|a[j]|\prod_{n=0}^{i-j}\lambda[i-n]+\sum_{j=K}^{i}|a[j]|\prod_{n=0}^{i-j}\lambda[i-n]\\&\quad<\epsilon\frac{\sum_{j=0}^{K-1}|a[j]|}{2KB}+\epsilon\frac{\sum_{j=K}^{i}\prod_{n=0}^{i-j}\lambda[i-n]}{2S}<\epsilon\end{aligned}$$

which concludes the proof.

Appendix II
Proof of Theorem 2

(Proof of Part (a)): The initial part of the proof builds on results in [6], [13], with the main difference that we now have to deal with random matrices $\boldsymbol{P}[i]$. For notational simplicity define

$$\boldsymbol{\Phi}[i]:=\begin{bmatrix}\mu_1[i]\frac{(\Theta_1[i](\boldsymbol{h}_1[i])-\Theta_1^\star[i])}{\|\Theta_1'[i](\boldsymbol{h}_1[i])\|^2+\delta_1[i]}\Theta_1'[i](\boldsymbol{h}_1[i])\\ \vdots\\ \mu_N[i]\frac{(\Theta_N[i](\boldsymbol{h}_N[i])-\Theta_N^\star[i])}{\|\Theta_N'[i](\boldsymbol{h}_N[i]\|^2+\delta_N[i]}\Theta_N'[i](\boldsymbol{h}_N[i]\end{bmatrix}. \tag{30}$$

From the properties of the matrix $\boldsymbol{P}[i]$, we can verify that $\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star[i]\|^2=\|\boldsymbol{P}[i](\boldsymbol{\psi}[i]-\boldsymbol{\Phi}[i]-\boldsymbol{\psi}^\star[i])\|^2$. Taking conditional expectation with respect to $\boldsymbol{\psi}[i]$ in this last equality, we deduce

$$\begin{aligned}&E[\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star[i]\|^2|\boldsymbol{\psi}[i]]\\&\quad=E[\|\boldsymbol{P}[i](\boldsymbol{\psi}[i]-\boldsymbol{\Phi}[i]-\boldsymbol{\psi}^\star[i])\|^2|\boldsymbol{\psi}[i]]\\&\quad=(\boldsymbol{\psi}[i]-\boldsymbol{\Phi}[i]-\boldsymbol{\psi}^\star[i])^T\\&\qquad E[\boldsymbol{P}[i]^T\boldsymbol{P}[i]\,|\,\boldsymbol{\psi}[i]](\boldsymbol{\psi}[i]-\boldsymbol{\Phi}[i]-\boldsymbol{\psi}^\star[i])\\&\quad\leq\|E[\boldsymbol{P}[i]^T\boldsymbol{P}[i]\,|\,\boldsymbol{\psi}[i]]\|_2\|\boldsymbol{\psi}[i]-\boldsymbol{\Phi}[i]-\boldsymbol{\psi}^\star[i]\|^2.\end{aligned}$$

We can now use Definition 5.2 to get

$$\begin{aligned}&E[\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star[i]\|^2\,|\,\boldsymbol{\psi}[i]]\\&\quad\leq\|\boldsymbol{\psi}[i]-\boldsymbol{\Phi}[i]-\boldsymbol{\psi}^\star[i]\|^2\\&\quad=\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star[i]\|^2-2\boldsymbol{\Phi}[i]^T(\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star[i])+\|\boldsymbol{\Phi}[i]\|^2\\&\quad\leq\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star[i]\|^2\\&\qquad-2\sum_{k\in\mathcal{N}}\mu_k[i]\left(\frac{\Theta_k[i](\boldsymbol{h}_k[i])-\Theta_k^\star[i]}{\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i]}\right)\\&\qquad\cdot[\Theta_k'[i](\boldsymbol{h}_k[i])^T(\boldsymbol{h}_k[i]-\boldsymbol{h}^\star[i])]\\&\qquad+\sum_{k\in\mathcal{N}}\mu_k[i]^2\frac{(\Theta_k[i](\boldsymbol{h}_k[i])-\Theta_k^\star[i])^2}{\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i]}\end{aligned}$$

where the last inequality comes from the definition of $\boldsymbol{\Phi}[i]$ and the fact that $\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2/(\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i])^2\leq 1$. By the definition of subgradients given in (1), we have $\Theta_k'[i](\boldsymbol{h}_k[i])^T(\boldsymbol{h}_k[i]-\boldsymbol{h}^\star[i])\geq\Theta_k[i](\boldsymbol{h}_k[i])-\Theta_k^\star[i]\geq 0$ for every $k\in\mathcal{N}$; therefore,

$$\begin{aligned}&E[\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star[i]\|^2|\boldsymbol{\psi}[i]]\\&\quad\leq\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star[i]\|^2\\&\qquad-2\sum_{k\in\mathcal{N}}\mu_k[i]\frac{(\Theta_k[i](\boldsymbol{h}_k[i])-\Theta_k^\star[i])^2}{\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i]}\\&\qquad+\sum_{k\in\mathcal{N}}\mu_k[i]^2\frac{(\Theta_k[i](\boldsymbol{h}_k[i])-\Theta_k^\star[i])^2}{\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i]}\\&\quad=\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star[i]\|^2\\&\qquad-\sum_{k\in\mathcal{N}}\mu_k[i](2-\mu_k[i])\left[\frac{(\Theta_k[i](\boldsymbol{h}_k[i])-\Theta_k^\star[i])^2}{\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i]}\right].\end{aligned} \tag{31}$$

By recalling that $E[x]=E_y[E_x[x\,|\,y]]$ for any two random variables x and y [41, p. 105], it holds that $E[\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star[i]\|^2]=E[E[\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star[i]\|^2|\boldsymbol{\psi}[i]]]$. Applying this result to (31) and using the fact that $\mu_k[i]\in(0,2)$, we arrive at the desired inequality $E[\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star[i]\|^2]-E[\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star[i]\|^2]\geq 0$.

[Proof of Part (b)]: By (31) and the allowed range of the step size, we get the supermartingale

$$\begin{aligned}E[\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star\|^2\,|\,\boldsymbol{\psi}[i]]&\leq\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star\|^2\\&\quad-\sum_{k\in\mathcal{N}}\epsilon_1\epsilon_2\left[\frac{(\Theta_k[i](\boldsymbol{h}_k[i])-\Theta_k^\star[i])^2}{\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i]}\right]\end{aligned} \tag{32}$$

where $\boldsymbol{\psi}^\star\in C^\star$. Applying Theorem 1 to (32) with $z[i]=0$ and $y[i]$ being the series in the right-hand side of (32), we verify that, with probability one, $\{\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star\|^2\}$ converges and

$$\sum_{j=0}^{\infty}\sum_{k\in\mathcal{N}}\frac{(\Theta_k[j](\boldsymbol{h}_k[j])-\Theta_k^\star)^2}{\|\Theta_k'[j](\boldsymbol{h}_k[j]\|^2+\delta_k[j]}<\infty. \tag{33}$$

In particular, the convergence of the series in (33) implies, that, with probability 1

$$\lim_{j\to\infty}\frac{(\Theta_k[j](\boldsymbol{h}_k[j])-\Theta_k^\star)^2}{(\|\Theta_k'[j](\boldsymbol{h}_k[j]\|^2+\delta_k[j])}=0,\quad k\in\mathcal{N}.$$

The above limit, together with the assumption that the sequence $\{\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i]\}$ is bounded, shows that $\mathcal{P}(\lim_{i\to\infty}\Theta_k[i](\boldsymbol{h}_k[i])=\Theta_k^\star)=1$.

[Proof of Part (c)]: Before proceeding with the proof of mean square consensus, we list some simple properties that will ease the analysis.

Claim 1: Consider the vectors $\boldsymbol{\psi}[i],\boldsymbol{\Phi}[i],\tilde{\boldsymbol{\psi}}[i]:=(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i]$, and $\tilde{\boldsymbol{\Phi}}[i]:=(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\Phi}[i]$. The following holds:

1) $E[\|\boldsymbol{\psi}[i]\|^2]$ is bounded (also implying the boundedness of $E[\|\tilde{\boldsymbol{\psi}}[i]\|^2]$);
2) $\lim_{i\to\infty}E[\|\boldsymbol{\Phi}[i]\|^2]=0$ (thus $\lim_{i\to\infty}E[\|\tilde{\boldsymbol{\Phi}}[i]\|^2]=0$);
3) $\lim_{i\to\infty}E[\tilde{\boldsymbol{\psi}}[i]^T\tilde{\boldsymbol{\Phi}}[i]]=0$.

Proof:

1) We know that the sequence $\{E[\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star\|^2]\}$ is monotone non-increasing (thus bounded) for every $\boldsymbol{\psi}^\star\in C^\star$ [this result follows from part (a) of Theorem 2 and the assumptions in part (b)]. Since

$$E[\|\boldsymbol{\psi}[i]\|^2]=E[\|\boldsymbol{\psi}^\star\|^2]+E[\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star\|^2]+2E[(\boldsymbol{\psi}^\star)^T(\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star)]$$

we only need to show that the last term in the previous equation is bounded to prove our claim, and this result follows from the Cauchy–Schwartz inequality applied to the inner product $\langle\boldsymbol{x},\boldsymbol{y}\rangle:=E[\boldsymbol{x}^T\boldsymbol{y}]$:

$$(E[(\boldsymbol{\psi}^\star)^T(\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star)])^2\le E[\|\boldsymbol{\psi}^\star\|^2]E[\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star\|^2].$$

2) Taking expectation on both sides of (32) and after some simple manipulations, we obtain

$$\begin{aligned}&E[\|\boldsymbol{\Phi}[i]\|^2]\\&=\sum_{k\in\mathcal{N}}\mu_k[i]^2\\&\quad\cdot E\left[\frac{(\Theta_k[i](\boldsymbol{h}_k[i])-\Theta_k^\star[i])^2\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2}{(\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i])^2}\right]\\&\le(2-\epsilon_2)^2\sum_{k\in\mathcal{N}}E\left[\frac{(\Theta_k[i](\boldsymbol{h}_k[i])-\Theta_k^\star[i])^2}{\|\Theta_k'[i](\boldsymbol{h}_k[i])\|^2+\delta_k[i]}\right]\\&\le\frac{(2-\epsilon_2)^2}{\epsilon_1\epsilon_2}(E[\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star\|^2]\\&\quad-E[\|\boldsymbol{\psi}[i+1]-\boldsymbol{\psi}^\star\|^2])\to0\end{aligned}\tag{34}$$

as $i\to\infty$ because $E[\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star\|^2]\ge0$ converges.

3) By parts 1) and 2) of the claim and the Cauchy–Schwartz inequality applied once again to the inner product $\langle\boldsymbol{x},\boldsymbol{y}\rangle:=E[\boldsymbol{x}^T\boldsymbol{y}]$, we obtain $(E[\tilde{\boldsymbol{\psi}}[i]^T\tilde{\boldsymbol{\Phi}}[i]])^2\le E[\|\tilde{\boldsymbol{\psi}}[i]\|^2]E[\|\tilde{\boldsymbol{\Phi}}[i]\|^2]\le B_{\tilde{\boldsymbol{\psi}}}E[\|\tilde{\boldsymbol{\Phi}}[i]\|^2]\to0$ as $i\to\infty$, where $B_{\tilde{\boldsymbol{\psi}}}<\infty$ is an upper bound of the sequence $\{E[\|\tilde{\boldsymbol{\psi}}[i]\|^2]\}$ (which is well defined because of Claim 1.1).

■

Now we proceed with the main proof. Left-multiplying both sides of the iteration $\boldsymbol{\psi}[i]=\boldsymbol{P}[i](\boldsymbol{\psi}[i]-\boldsymbol{\Phi}[i])$ by $(\boldsymbol{I}-\boldsymbol{J})$ and using the fact that $(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{J}=\boldsymbol{0}$ and $\boldsymbol{P}[i]\boldsymbol{J}=\boldsymbol{J}$ (property 3 in Definition 5), we have $(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i+1]=(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{P}[i](\boldsymbol{\psi}[i]-\boldsymbol{\Phi}[i])=(\boldsymbol{P}[i]-\boldsymbol{J}\boldsymbol{P}[i])(\boldsymbol{I}-\boldsymbol{J})(\boldsymbol{\psi}[i]-\boldsymbol{\Phi}[i])$. We can use this property to verify that

$$\|(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i+1]\|^2=(\tilde{\boldsymbol{\psi}}[i]-\tilde{\boldsymbol{\Phi}}[i])^T\boldsymbol{Y}[i]^T\boldsymbol{Y}[i](\tilde{\boldsymbol{\psi}}[i]-\tilde{\boldsymbol{\Phi}}[i])$$

where $\boldsymbol{Y}[i]:=\boldsymbol{P}[i]-\boldsymbol{J}\boldsymbol{P}[i]$, $\tilde{\boldsymbol{\psi}}[i]:=(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i]$, and $\tilde{\boldsymbol{\Phi}}[i]:=(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\Phi}[i]$. Taking expectation on both sides, we obtain

$$\begin{aligned}&E[\|\tilde{\boldsymbol{\psi}}[i+1]\|^2]\\&\quad=E[(\tilde{\boldsymbol{\psi}}[i]-\tilde{\boldsymbol{\Phi}}[i])^TE[\boldsymbol{Y}[i]^T\boldsymbol{Y}[i]|\boldsymbol{\psi}[i]](\tilde{\boldsymbol{\psi}}[i]-\tilde{\boldsymbol{\Phi}}[i])]\\&\quad\le E[\|E[\boldsymbol{Y}[i]^T\boldsymbol{Y}[i]|\boldsymbol{\psi}[i]]\|_2\|\tilde{\boldsymbol{\psi}}[i]-\tilde{\boldsymbol{\Phi}}[i]\|^2]\\&\quad\le\lambda[i](E[\|\tilde{\boldsymbol{\psi}}[i]\|^2]-2E[\tilde{\boldsymbol{\psi}}[i]^T\tilde{\boldsymbol{\Phi}}[i]]+E[\|\tilde{\boldsymbol{\Phi}}[i]\|^2])\end{aligned}\tag{35}$$

where $\lambda[i]=(1-\epsilon)$ if i is a time index where an ϵ-random consensus matrix is present or $\lambda[i]=1$ otherwise. Expanding recursively the resulting inequality, we get

$$\begin{aligned}E[\|\tilde{\boldsymbol{\psi}}[i+1]\|^2]&\le\prod_{n=0}^{i}\lambda[i-n]\|\tilde{\boldsymbol{\psi}}[0]\|^2\\&\quad-2\sum_{j=0}^{i}\prod_{n=0}^{i-j}\lambda[i-n]a[j]+\sum_{j=0}^{i}\prod_{n=0}^{i-j}\lambda[i-n]b[j]\end{aligned}\tag{36}$$

where $a[i]:=E[\tilde{\boldsymbol{\psi}}[i]^T\tilde{\boldsymbol{\Phi}}[i]]$ and $b[i]:=E[\|\tilde{\boldsymbol{\Phi}}[i]\|^2]$ for $i>0$ ($a[0]:=\tilde{\boldsymbol{\psi}}[0]^T\tilde{\boldsymbol{\Phi}}[0]$ and $b[0]:=\|\tilde{\boldsymbol{\Phi}}[0]\|^2$ are deterministic because $\boldsymbol{\psi}[0]$ is deterministic). The first term of the right-hand side of (36) in the second inequality converges to zero because $\prod_{n=0}^{i-j}\lambda[i-n]\le(1-\epsilon)^{\lfloor(i-j)/(I+1)\rfloor}$ (there is at least one ϵ-random consensus matrix in every interval in the form $[l,l+I]$, $l\in\mathbb{N}$). Using Claim 1.2 and 1.3 together with Lemma 1, we verify that the last two terms of the right-hand side of (36) also converge to zero, and thus $\lim_{i\to\infty}E[\|\tilde{\boldsymbol{\psi}}[i]\|^2]=0$.

[Proof of Part (d)]: The proof of almost sure convergence is essentially a rephrasing of the proof of Theorem 1.3 (see [29] and the references therein). First apply Theorem 1 to (32) to conclude that $\{\boldsymbol{\psi}_\omega[i]\}$ is bounded and has an accumulation point for almost every $\omega\in\Omega$. Therefore, to prove convergence, we only need to show that the accumulation point is unique. Assume the contrary, and suppose that $\boldsymbol{\psi}_\omega'$ and $\boldsymbol{\psi}_\omega''$ are distinct accumulation points when $C^\star$ does not lie in a hyperplane. The sequence $\{\|\boldsymbol{\psi}[i]-\boldsymbol{\psi}^\star\|^2\}$ converges with probability one for every $\boldsymbol{\psi}^\star\in C^\star$ (proved in part (b) of the theorem), and thus, for almost every $\omega\in\Omega$, $0=\|\boldsymbol{\psi}_\omega'-\boldsymbol{\psi}^\star\|^2-\|\boldsymbol{\psi}_\omega''-\boldsymbol{\psi}^\star\|^2=\|\boldsymbol{\psi}_\omega'\|-\|\boldsymbol{\psi}_\omega''\|-2(\boldsymbol{\psi}_\omega'-\boldsymbol{\psi}_\omega'')^T\boldsymbol{\psi}^\star$ which contradicts the fact that $C^\star$ does not lie in a hyperplane. Therefore, $\boldsymbol{\psi}[i]$ converges with probability one to a random vector $\hat{\boldsymbol{\psi}}$.

Consensus follows from part (c) of the theorem because, by Mann–Wald's theorem and Fatou's lemma [41], $E[\|(\boldsymbol{I}-\boldsymbol{J})\hat{\boldsymbol{\psi}}\|^2]=E[\lim_{i\to\infty}\|(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i]\|^2]\le\liminf_{i\to\infty}E[\|(\boldsymbol{I}-\boldsymbol{J})\boldsymbol{\psi}[i]\|^2]=0$, which implies that $\|(\boldsymbol{I}-\boldsymbol{J})\hat{\boldsymbol{\psi}}\|=0$ with probability one, i.e., the agents reach consensus.

[Proof of Part (e)]: Now that we know that the sequence $\{\boldsymbol{\psi}[i]\}$ converges with probability one to a point in the consensus subspace $\mathcal{C}$ defined in (7), we can mimic the proof of [[13], Theorem 2(d)] or that of ([16], Theorem 3.1.4) to finish ours.

Assume the contrary of the statement we want to prove; suppose that, for almost every $\omega \in \Omega$, $\hat{\boldsymbol{h}}_\omega$ satisfies $\hat{\boldsymbol{h}}_\omega \notin \overline{\liminf_{i\to\infty} \Upsilon[i]}$. Since $\tilde{\boldsymbol{u}}$ is an interior point of Υ, there exists ρ such that $\{\boldsymbol{v} \in \mathbb{R}^M | \|\boldsymbol{v} - \tilde{\boldsymbol{u}}\| \leq \rho\} \in \Upsilon$. In addition, there exists $t_\omega \in [0,1]$ such that $\boldsymbol{u}_{t_\omega} := t_\omega \hat{\boldsymbol{h}}_\omega + (1-t_\omega)\tilde{\boldsymbol{u}} \notin \overline{\liminf_{i\to\infty} \Upsilon[i]} \supset \liminf_{i\to\infty} \Upsilon[i]$. By $\lim_{i\to\infty} \boldsymbol{h}_{k,\omega}[i] = \hat{\boldsymbol{h}}_\omega$ $(k \in \mathcal{N})$, we know that for some $N_{1,\omega} \in \mathbb{N}$ it holds that $\|\boldsymbol{h}_{k,\omega}[i] - \hat{\boldsymbol{h}}_\omega\| \leq \rho(1-t_\omega)/(2t_\omega)$ for every $i \geq N_{1,\omega}$ and $k \in \mathcal{N}$. Therefore, by $\boldsymbol{u}_{t_\omega} \notin \liminf_{i\to\infty} \Upsilon[i]$, for any $L_{1,\omega} > N_{1,\omega}$, there exists $i_1 = i_1(L_{1,\omega}) \geq L_{1,\omega}$ satisfying $\boldsymbol{u}_{t_\omega} \notin \Upsilon[i_1] = \bigcap_{k\in\mathcal{N}} (\mathrm{lev}_{\leq\Theta_k^\star} \Theta_k[i_1])$. As a result, there exists $j_\omega \in \mathcal{N}$ such that $\boldsymbol{u}_{t_\omega} \notin \mathrm{lev}_{\leq\Theta_{j_\omega}^\star} \Theta_{j_\omega}[i_1]$. For this j_ω, by $\Upsilon \subset \Upsilon[i] \subset \mathrm{lev}_{\leq\Theta_{j_\omega}^\star} \Theta_{j_\omega}[i_1]$ and Fact 1,

$$\begin{aligned} &d(\boldsymbol{h}_{j_\omega,\omega}[i_1], \mathrm{lev}_{\leq\Theta_{j_\omega}^\star} \Theta_{j_\omega}[i_1]) \\ &\quad \geq d(\hat{\boldsymbol{h}}_\omega, \mathrm{lev}_{\leq\Theta_{j_\omega}^\star} \Theta_{j_\omega}[i_1]) - \|\boldsymbol{h}_{j_\omega,\omega}[i_1] - \hat{\boldsymbol{h}}_\omega\| \\ &\quad \geq \rho\frac{1-t_\omega}{t_\omega} - \frac{\rho}{2}\frac{1-t_\omega}{t_\omega} = \frac{\rho}{2}\frac{1-t_\omega}{t_\omega} =: \epsilon_\omega > 0 \end{aligned}$$

which shows that $\sum_{k\in\mathcal{N}} d(\boldsymbol{h}_{k,\omega}[i_1], \mathrm{lev}_{\leq\Theta_k^\star} \Theta_k[i_1]) \geq \epsilon_\omega$. By the triangle inequality, we obtain

$$\begin{aligned} \|\tilde{\boldsymbol{u}} - \boldsymbol{h}_{k,\omega}[i]\| &\leq \|\tilde{\boldsymbol{u}} - \hat{\boldsymbol{h}}_\omega\| + \|\boldsymbol{h}_{k,\omega}[i] - \hat{\boldsymbol{h}}_\omega\| \\ &\leq \|\tilde{\boldsymbol{u}} - \hat{\boldsymbol{h}}_\omega\| + \frac{\rho}{2}\frac{1-t_\omega}{t_\omega} \quad (k \in \mathcal{N}) \end{aligned}$$

and thus

$$\begin{aligned} \sum_{k\in\mathcal{N}} \|\tilde{\boldsymbol{u}} - \boldsymbol{h}_{k,\omega}[i_1]\| &\leq N\|\tilde{\boldsymbol{u}} - \hat{\boldsymbol{h}}_\omega\| + N\frac{\rho}{2}\frac{1-t_\omega}{t_\omega} \\ &=: r_\omega > 0. \end{aligned}$$

We can now fix $L_{2,\omega} > i_1$ and repeat the above to find $i_2 = i_2(L_{2,\omega}) \geq L_{2,\omega}$ such that $\sum_{k\in\mathcal{N}} d(\boldsymbol{h}_{k,\omega}[i_2], \mathrm{lev}_{\leq\Theta_k^\star} \Theta_k[i_2]) \geq \epsilon_\omega$ and $\sum_{k\in\mathcal{N}} \|\tilde{\boldsymbol{u}} - \boldsymbol{h}_{k,\omega}[i_2]\| \leq r_\omega$, which shows that we can construct a subsequence $\{i_l\}$ $(l \geq 1)$ satisfying

$$\begin{aligned} &\sum_{k\in\mathcal{N}} d(\boldsymbol{h}_{k,\omega}[i_l], \mathrm{lev}_{\leq\Theta_k^\star} \Theta_k[i_l]) \geq \epsilon_\omega \text{ and} \\ &\sum_{k\in\mathcal{N}} \|\tilde{\boldsymbol{u}} - \boldsymbol{h}_{k,\omega}[i_l]\| \leq r_\omega. \end{aligned}$$

As a result, for almost every $\omega \in \Omega$, we can construct a subsequence $\{i_l\}$ as above and, using the assumptions of the theorem, we can find $\xi_\omega > 0$ such that $\sum_{k\in\mathcal{N}} \Theta_k[i_l](\boldsymbol{h}_{k,\omega}[i_l]) \geq \sum_{k\in\mathcal{N}} \Theta_k^\star + \xi_\omega$ for every $l \geq 1$. This contradicts $\lim_{i\to\infty} \Theta_k[i](\boldsymbol{h}_{k,\omega}[i]) = \Theta_k^\star$, which is proved in part (b) of the theorem. Therefore, $\hat{\boldsymbol{h}}_\omega \in \overline{\liminf_{i\to\infty} \Upsilon[i]}$, and the proof is complete.

REFERENCES

[1] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *Proc. 47th IEEE Conf. Decision Control (CDC'08)*, Cancun, Mexico, Dec. 2008, pp. 4185–4190.

[2] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson, "On decentralized negotiation of optimal consensus," *Automatica*, vol. 44, no. 4, pp. 1175–1179, Apr. 2008.

[3] D. Blatt and A. O. Hero, III, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3614–3619, Sep. 2006.

[4] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM J. Optim.*, vol. 18, no. 1, pp. 29–51, Feb. 2007.

[5] R. L. G. Cavalcante, I. Yamada, and B. Mulgrew, "Learning in diffusion networks with an adaptive projected subgradient method," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 2853–2856.

[6] R. L. G. Cavalcante, I. Yamada, and B. Mulgrew, "An adaptive projected subgradient approach to learning in diffusion networks," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2762–2774, Jul. 2009.

[7] N. Takahashi and I. Yamada, "Incremental adaptive filtering over distributed networks using parallel projection onto hyperslabs," IEICE Tech. Rep. SIP2008-38, 2008, pp. 17–22.

[8] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 1035–1048, May 2010.

[9] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[10] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Asynchronous gossip algorithms for stochastic optimization," in *Proc. 48th IEEE Conf. Decision Control, 2009 Held Jointly With the 2009 28th Chinese Control Conf. CDC/CCC 2009.*, Shanghai, China, Dec. 2009, pp. 3581–3586.

[11] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Trans. Autom. Control*, vol. 56, 2011, to be published.

[12] I. Yamada, "Adaptive projected subgradient method: A unified view for projection based adaptive algorithms," (in Japanese) *J. IEICE*, vol. 86, no. 8, pp. 654–658, Aug. 2003.

[13] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numer. Funct. Anal. Optimizat.*, vol. 25, no. 7/8, pp. 593–617, 2004.

[14] R. L. G. Cavalcante and I. Yamada, "Multiaccess interference suppression in orthogonal space-time block coded MIMO systems by adaptive projected subgradient method," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1028–1042, Mar. 2008.

[15] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Signal Processing*, vol. 56, pp. 2781–2796, Jul. 2008.

[16] K. Slavakis, I. Yamada, and N. Ogura, "The adaptive projected subgradient method over the fixed point set of strongly attracting nonexpansive mappings," *Numer. Funct. Anal. Optimizat.*, vol. 27, no. 7–8, pp. 905–930, Dec. 2006.

[17] B. T. Polyak, "Minimization of unsmooth functionals," *USSR Comput. Math. Phys.*, vol. 9, pp. 14–29, 1969.

[18] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[19] 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.

[20] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2748–2761, Jul. 2009.

[21] D. Ustebay, B. O. , M. Coates, and M. Rabbat, "Rates of convergence for greedy gossip with eavesdropping," in *Proc. Commun., Control, Comput. 46th Annual Allerton Conf.*, Monticello, IL, Sep. 2008, pp. 367–374.

[22] D. Ustebay, M. Coates, and M. Rabbat, "Greedy gossip with eavesdropping," in *Proc. 3rd Int. Symp. Wireless Pervasive Comput., ISWP'08.*, May 2008, pp. 759–763.

[23] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[24] S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3315–3326, Jul. 2008.

[25] R. L. G. Cavalcante, A. Rogers, N. Jennings, and I. Yamada, "Distributed multiagent learning with a broadcast adaptive subgradient method," in *Proc. 9th Int. Conf. Auton. Agents Multiagent Syst. (AAMAS'10)*, 2010.
[26] H. Stark and Y. Yang, *Vector Space Projections—A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics*. New York: Wiley, 1998.
[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[28] D. Williams, *Probability With Martingales*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
[29] Y. Ermoliev, "Stochastic quasigradient methods and their application in systems optimization," *Stochastics*, no. 9, pp. 1–36, 1983.
[30] D. P. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.
[31] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 44–53, Jan. 2005.
[32] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Trans. Robotics*, vol. 24, no. 3, pp. 710–724, Jun. 2008.
[33] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.
[34] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Hoboken, NJ: Wiley, 2003.
[35] T. Hinamoto and S. Maekawa, "Extended theory of learning identification," *Trans. IEE Japan*, vol. 95, no. 10, pp. 227–234, 1975.
[36] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *IEICE Trans.*, vol. 67-A, no. 5, pp. 126–132, Feb. 1984.
[37] I. Yamada, K. Slavakis, and K. Yamada, "An efficient robust adaptive filtering algorithm based on parallel subgradient projection techniques," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1091–1101, May 2002.
[38] J. Nagumo and J. Noda, "A learning method for system identification," *IEEE Trans. Automat. Control*, vol. 12, no. 3, pp. 282–287, Jun. 1967.
[39] M. Yukawa, R. L. G. Cavalcante, and I. Yamada, "Efficient blind MAI suppression in DS/CDMA systems by embedded constraint parallel projection techniques," *IEICE Trans. Fundamentals*, vol. E88-A, no. 8, pp. 2062–2071, Aug. 2005.
[40] G. Pedrick, *A First Course in Analysis*. New York: Springer-Verlag, 1994.
[41] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 3rd ed. Oxford, U.K.: , 2005.

Renato L. G. Cavalcante (M'06) received the electronics engineering degree from the Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, Brazil, in 2002, and the M.E. and Ph.D. degrees in communications and integrated systems from the Tokyo Institute of Technology, Tokyo, Japan, in 2006 and 2008, respectively.

He is currently a Research Fellow with the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin, Germany. Previously, he held appointments as a Research Fellow with the University of Southampton, Southampton, U.K., and as a Research Associate with the University of Edinburgh, Edinburgh, U.K. His current interests are in signal processing for distributed systems, multiagent systems, and wireless communications.

Dr. Cavalcante received the Excellent Paper Award from the IEICE in 2006 and the IEEE Signal Processing Society (Japan Chapter) Student Paper Award in 2008. From April 2003 to April 2008, he was a recipient of the Japanese Government (MEXT) Scholarship.

Alex Rogers received the B.Sc. degree (honors) in physics from the University of Durham, Durham, U.K., and the Ph.D. degree in computer science from the University of Southampton, Southampton, U.K.

He is currently a Reader in the School Electronics and Computer Science at the University of Southampton. His current research interests are developing decentralized algorithms to control open agent-based systems.

Nicholas R. Jennings (M'92–SM'03–F'09) received the B.Sc. (honors) and Ph.D. degrees in computer science (First Class) from the Queen Mary, University of London, London, U.K., in 1988 and 1992, respectively.

Currently, he is a Professor of computer Science at the University of Southampton, Southampton, U.K., and a Chief Scientific Advisor to the U.K. Government. Previously, he was a Professor at Queen Mary, University of London. He has published nearly 500 articles in the areas of multi-agent systems and artificial intelligence.

Prof. Jennings has received a number of international awards for his research: the Computers and Thought Award, the ACM Autonomous Agents Research Award, and an IEE Achievement Medal. He is a Fellow of the Royal Academy of Engineering, the British Computer Society, the Institution of Engineering and Technology (formerly the IEE), the Association for the Advancement of Artificial Intelligence (AAAI), the Society for the Study of Artificial Intelligence and Simulation of Behavior (AISB), and the European Artificial Intelligence Association (ECCAI) and a member of Academia Europaea and the UK Computing Research Committee (UKCRC). He was the founding Editor-in-Chief of the *International Journal of Autonomous Agents and Multi-Agent Systems*, is a member of the scientific advisory board of the German AI Institute (DFKI) and a founding director of the International Foundation for Autonomous Agents and Multi-Agent Systems. He has also led teams that have won competitions in the areas of: the Iterated Prisoners' Dilemma (the 20th Anniversary competitions in 2004 and 2005), RoboCup Rescue (the Infrastructure competition in 2007), Agent Trust and Reputation (the ART competitions in 2006 and 2007), the Lemonade Stand Game (2009 and 2010), and Market Design (the TAC CAT competition in 2007).

Isao Yamada (M'96–SM'06) received the B.E. degree in computer science from the University of Tsukuba, Ibaraki, Japan, in 1985 and the M.E. and Ph.D. degrees in electrical and electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1987 and 1990, respectively.

Currently, he is a Professor in the Department of Communications and Integrated Systems, Tokyo Institute of Technology. From August 1996 to July 1997, he was a Visiting Associate Professor at Pennsylvania State University, State College. His current research interests are in mathematical signal processing, nonlinear inverse problem and optimization theory.

Dr. Yamada is a member of the American Mathematical Society (AMS), the Society for Industrial and Applied Mathematics (SIAM), the Japan Society for Industrial and Applied Mathematics (JSIAM), and the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan. He has been an Associate Editor for several journals, including the *International Journal on Multidimensional Systems and Signal Processing*, Springer (since 1997), the *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* (2001–2005), and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: FUNDAMENTAL THEORY AND APPLICATIONS (2006–2007). Currently, he is a member of the IEEE Signal Processing Theory and Methods Technical Committee as well as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is also the chairman of the IEICE Technical Group on Signal Processing (2011). He received Excellent Paper Awards in 1991, 1995, 2006, and 2009; the Young Researcher Award in 1992 and the Achievement Award in 2009 from the IEICE; the ICF Research Award from the International Communications Foundation in 2004; the Docomo Mobile Science Award (Fundamental Science Division) from Mobile Communication Fund in 2005; and the Fujino Prize from Tejima Foundation in 2008.