A Distributed Technique for Localization of Agent Formations From Relative Range Measurements

Giuseppe C. Calafiore, Luca Carlone, and Mingzhu Wei

Abstract—Autonomous agents deployed or moving on land for the purpose of carrying out coordinated tasks need to have good knowledge of their absolute or relative position. For large formations, it is often impractical to equip each agent with an absolute sensor such as GPS, whereas relative range sensors measuring interagent distances are cheap and commonly available. In this setting, this paper considers the problem of autonomous distributed estimation of the position of each agent in a networked formation using noisy measurements of interagent distances. The underlying geometrical problem has been studied quite extensively in various fields, ranging from molecular biology to robotics, and it is known to lead to a hard nonconvex optimization problem. Centralized algorithms do exist that work reasonably well in finding local or global minimizers for this problem (e.g., semidefinite programming relaxations). Here, we explore a fully decentralized approach for localization from range measurements, and we propose a computational scheme based on a distributed gradient algorithm with Barzilai–Borwein stepsizes. The advantage of this distributed approach is that each agent may autonomously compute its position estimate, exchanging information only with its neighbors, without need of communicating with a central station and without needing complete knowledge of the network structure.

Index Terms—Consensus, distributed algorithms, network localization, range measurements.

I. INTRODUCTION

DISTRIBUTED and networked systems (e.g., portable phones, smart dust, environmental sensors, mobile robots) are bound to become increasingly intertwined with human life and play an important role in many technological fields due to their capability of acquiring information over wide areas in a decentralized and autonomous way. Advances in wireless communication further increased the potentiality of multiagent networks in applications such as mobile robotics, target tracking, environmental monitoring, and surveillance. Most of the mentioned applications require that each node has precise knowledge of its geometric position since actions and observations are often location dependent. For example, monitoring and surveillance become meaningless if it is not possible to associate to the observation in the corresponding geographic information; formation control requires knowledge of agent positions; location-aware routing benefits from the position information for optimizing the flow of information through the network, etc. Although the agents' positions can be obtained through direct measures (e.g., GPS), in many cases, this solution is not feasible due to technological constraints, such as cost and power consumption. As a consequence, research effort has been recently oriented toward designing indirect methods for reconstructing the absolute node positions from relative and partial measurements. The problem of determining node positions is generically known as network localization: the absolute node positions (with respect to a local or global reference frame) need be estimated from partial relative measurements between nodes, that is, each node may measure the relative position from a set of neighbor nodes, and the global absolute positions of all nodes need be deduced from this information.

Localization problems for planar networks can be classified depending on the type of available relative measurements. A recent survey on both technological and algorithmic aspects can be found in [18]. If relative coordinates between nodes are measured (or, equivalently, relative angles and distances are measured), network localization can be formulated as a linear estimation problem and efficiently solved using a least-square approach (see, e.g., [1], [2], and the references therein). The case in which angle-only measurements are available to the nodes was pioneered by Stanfield [32] and further generalized in [8]. The latter is often referred to as bearing localization and can be attacked via maximum likelihood estimation, as described in [18]. The most common setup is probably the one in which each node can measure only distances from a subset of other nodes in the formation. This case, which we will name range localization, has been treated in various settings (see, for instance, [7] and [28]). Range localization naturally leads to a strongly NP-hard nonlinear and nonconvex optimization problem (see [27]) in which convergence to a global solution cannot in general be guaranteed. Convex relaxation methods based on semidefinite programming (SDP) have recently been proposed; in the noiseless case, these approaches provably converge to the global solution (see, e.g., [4]). However, the speed of the SDP approach is not satisfactory for practical applications with medium/high number of nodes, and in presence of measurement noise, the estimate of SDP-based solutions can be inaccurate (i.e., far from a global optimum). This motivates the use of steepest-descent-based local optimization techniques to improve the accuracy of an SDP-based solution. The distributed gradient algorithm described in this paper can also be applied in such refinement phase (see Section V-C).

G. C. Calafiore and M. Wei are with the Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Torino, Italy (e-mail: giuseppe.calafiore@polito.it; mingzhu.wei@polito.it).

L. Carlone is with the CSPP, Laboratorio di Meccatronica, Politecnico di Torino, 10129 Torino, Italy (e-mail: luca.carlone@polito.it).

The actual reconstruction of a unique network configuration from range measurements is possible only under particular hypotheses on the topology of the networked formation (graph rigidity, see [7]). Moreover, localization in an absolute reference frame requires that a subset of the nodes (anchor nodes or beacons) already knows its exact location in the external reference frame. Otherwise, localization is possible only up to an arbitrary roto-translation. This latter setup is referred to as anchor-free localization (see, e.g., [23]).

If all relative measurements are gathered to some "central elaboration unit" that performs estimation over the whole network, the corresponding localization technique is said to be centralized. This is the approach that one implicitly assumes when writing and solving an optimization problem: all the data that are relevant for problem description are available to the problem solver. However, such an assumption may be unrealistic in situations where data and information are actually stored locally at the agents, and where the computations can also be executed locally at the individual agents' level. Approaching such problems in a centralized way would require that each agent sends its information to the central elaboration unit, who solves the (usually large-scale) problem and transmits back the results to each agent. This may of course be highly undesirable due to the intensive communication load among the central units and agents. Moreover, since all the computation is performed by a single unit, for large networks, the computational effort can be just too intensive. In addition, the system is fragile, since failure in the central elaboration unit or in communication would compromise the functioning of the whole network. According to these considerations, distributed approaches are desirable for solving network localization. In a distributed setup, each node communicates only with its neighbors and performs local computations to obtain an estimate of its own position. As a consequence, the communication burden is equally spread over the network, and the computation is decentralized and entrusted to each agent, improving both efficiency and robustness of the estimation process. First attempts to reduce the computational effort of localization by breaking the problem into smaller subproblems trace back to [11], in which a divide-and-conquer algorithm is proposed. Similar considerations are drawn in [19], where clustering is applied to the network to properly reconstruct network configuration. Recent approaches to the problem include [14] and [35]. The former faces the problem in the anchor-free setting, whereas the latter makes use of barycentric coordinates for localizing the nodes, under the hypothesis that nonanchor nodes lie in the convex hull of anchors. The distributed approach described in [14] guarantees global convergence under the additional hypothesis that each node can increase his radius of communication until it gets contained in the convex hull of its neighbors. Reference [15] adds further convergence analysis on previous work of the authors, although the use of barycentric coordinates still requires assumptions on network topology. In [13], the authors extend their previous work by discussing a general framework for distributed sensing and control. Distributed solutions are also investigated in [30], where the authors propose distributed greedy optimization algorithms and a second-order cone programming-based approach. Networks

with mobile agents are also considered in the recent literature, including [16] and [29].

In this paper, we consider the problem of distributed network localization with range-only measurements and develop a distributed gradient algorithm that uses Barzilai–Borwein (BB) stepsizes, where the stepsizes are computed in the network in a decentralized way through consensus iterations. The idealized version of this algorithm has two time scales: per each outer gradient iteration, a (ideally infinite) series of inner communication iterations is performed among nodes to reach consensus on the stepsize parameter. In practice, consensus is reached at a geometric rate; hence, a finite and limited number of inner iterations suffice to compute the stepsize. Once the network topology is known, the required number of inner iterations can actually be fixed a priory for the given desired accuracy level. This technique is completely decentralized, and it appears to be scalable for large networks. Moreover, it presents advantages over state-of-the-art techniques in terms of computational and communication requirements. As it is common to all gradient-based methods (either centralized or distributed), convergence of the algorithm can only be guaranteed for initial conditions that are close enough to the actual solution. For this reason, the proposed approach appears to be well suited as a refinement technique for centralized methods (such as SDP relaxations) or in dynamic situations where the node location is iteratively obtained as a small perturbation of a previously known location.

The rest of this paper is organized as follows. The problem setup is presented in Section II. A centralized version of the gradient scheme is reviewed in Section III. The main contribution of this paper is contained in Section IV, where the distributed range localization method is presented. Numerical experiments are reported in Section V, and conclusions are drawn in Section VI.

Notation: I_n denotes the $n \times n$ identity matrix, $\mathbf{1}_n$ denotes a (column) vector of all ones of dimension n, $\mathbf{0}_n$ denotes a vector of all zeros of dimension n, and $e_i \in \mathbb{R}^n$ denotes a vector with all zero entries, except for the ith position, which is equal to one. We denote with $\lfloor x \rfloor$ the largest integer smaller than or equal to x. Subscripts with dimensions may be omitted when they can easily be inferred from context. For a matrix X, X_{ij} denotes the element of X in row i and column j, and $X^\top$ denotes the transpose of X. $X > 0$ (respectively $X \geq 0$) denotes a positive (respectively nonnegative) matrix, that is, a matrix with all positive (respectively nonnegative) entries. $\|X\|$ denotes the spectral (maximum singular value) norm of X, or the standard Euclidean norm, in case of vectors. For a square matrix $X \in \mathbb{R}^{n,n}$, we denote with $\sigma(X) = \{\lambda_1(X), \ldots, \lambda_n(X)\}$ the set of eigenvalues, or spectrum, of X, and with $\rho(X)$ the spectral radius: $\rho(X) \doteq \max_{i=1,\ldots,n} |\lambda_i(X)|$, where $\lambda_i(X)$, $i = 1, \ldots, n$, are the eigenvalues of X ordered with decreasing modulus, i.e., $\rho(X) = |\lambda_1(X)| \geq |\lambda_2(X)| \geq \cdots \geq |\lambda_n(X)|$.

II. PROBLEM FORMULATION

Let $\mathcal{V} = \{v_1, \ldots, v_n\}$ be a set of n nodes (representing sensors, agents, robots, vehicles, etc.), and let $\mathcal{P} = \{p_1, \ldots, p_n\}$ denote a corresponding set of positions on the Cartesian plane, where $p_i = [x_i \ y_i]^\top \in \mathbb{R}^2$ are the coordinates of the ith node.

We will call $\mathcal{P}$ a configuration of nodes. Suppose that some pairs of nodes, say, nodes (i,j), have the possibility of measuring the relative distance between them, i.e.,

$$d_{ij}^2 = \|p_i - p_j\|^2 + \epsilon_{ij}$$

where ϵ_{ij} represents the residual error between the true (squared) distance and its measure d_{ij}^2. We denote with $\mathcal{E}$ the set of unordered node pairs (i,j) such that a distance measurement exists between i and j. Our objective is to determine a node configuration $\{p_1, \ldots, p_n\}$ that minimizes a least-squares goodness of fit criterion

$$f(p) = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} \left(\|p_i - p_j\|^2 - d_{ij}^2\right)^2. \qquad (1)$$

When the global minimum of f is zero, we say that exact matching is achieved, that is, a configuration $\{p_1, \ldots, p_n\}$ is found that exactly matches the given distance measurements

$$\|p_i - p_j\|^2 = d_{ij}^2, \quad \forall (i,j)\in\mathcal{E}.$$

Otherwise, no geometric node configuration can exactly match the given range data, and we say that approximate matching is achieved by the optimal configuration.

Actual recovery of the absolute geometric node positions from distance measurements is only possible if the node configuration is generically globally rigid (ggr) [7]. In this case, the objective function in (1) has a unique global minimum if the positions of at least three nodes is known and fixed in advance (anchor nodes, or beacons), or it has several equivalent global minima corresponding to congruence transformations (roto-translations) of the configuration if no anchors are specified. If the configuration is not ggr, instead, there exist many different geometric configurations (also called flexes) that match exactly or approximately the distance data and that correspond to the equivalent global minima of the cost f.

In this paper, we focus on a distributed numerical technique to compute a local or global minimum of f in the neighborhood of some given initial position estimate. In our approach, we treat under the same framework both anchor-based and anchor-free localization problems. In particular, when anchor nodes are specified at fixed positions, we just set the respective node position variables to the given values and eliminate these variables from the optimization.

In the next section, we review a centralized gradient-based approach for locally solving the localization problem (1). Then, Section IV contains the main contribution of this paper, providing a distributed version of the localization algorithm.

III. CENTRALIZED GRADIENT-BASED ALGORITHM

Let $p \doteq [p_1^\top \; p_2^\top \; \cdots \; p_n^\top]^\top$, where $p_i^\top = [x_i \; y_i]$ denotes the vector of node positions. The minimization objective (1) is rewritten as

$$f(p) = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} g_{ij}^2(p), \quad g_{ij}(p) \doteq \|p_i - p_j\|^2 - d_{ij}^2 \qquad (2)$$

and we let $p^{(0)}$ denote the vector of initial position estimates. This vector of initial guess can be available either as the result of a preliminary optimization phase (e.g., via SDP relaxation) or as the output of our position estimation algorithm itself at a previous time period (this may be the case, for instance, in the context of iterative dynamic tracking of a moving formation).

We next describe a centralized iterative method to determine a local minimum of the cost function starting from $p^{(0)}$. We remark that in centralized methods, the whole vector p must be stored and updated by some "central computing unit" that also has full knowledge of all the distance measurements.

A. Gradient-Based Method

The most basic iterative method for finding a local minimizer of $f(p)$ is the so called gradient algorithm. Let $p^{(\tau)}$ be the configuration computed by the algorithm at iteration τ, being $p^{(0)}$ the given initial configuration: at each iteration, the solution is updated according to the rule

$$p^{(\tau+1)} = p^{(\tau)} - \alpha_\tau \nabla f\left(p^{(\tau)}\right) \qquad (3)$$

where α_τ is the step length, which may be computed at each iteration via exact or approximate line search, and where

$$\nabla f(p) = \sum_{(i,j)\in\mathcal{E}} g_{ij}(p) \nabla g_{ij}(p) \qquad (4)$$

where gradient ∇g_{ij} is a column vector of n blocks, with each block composed of two entries, thus $2n$ entries in total, and with the only nonzero terms corresponding to the blocks in positions i and j, i.e.,

$$\nabla g_{ij}(p) = 2 \left[\mathbf{0}_2^\top \cdots \mathbf{0}_2^\top \; (p_i - p_j)^\top \; \mathbf{0}_2^\top \cdots \right.$$
$$\left. \mathbf{0}_2^\top \; (p_j - p_i)^\top \; \mathbf{0}_2^\top \cdots \mathbf{0}_2^\top\right]^\top.$$

The gradient method is guaranteed to converge to a local minimizer whenever $\{p : f(p) \le f(p^{(0)})\}$ is bounded and the step lengths satisfy the Wolfe conditions (see, e.g., [20]). Although the rate of convergence of the gradient method can be poor, we are interested in this method here since it requires first-order only information (no Hessian has to be computed), and it can suitably be adapted to a distributed implementation, as discussed in Section IV-B.

a) BB Scheme: A critical part of the gradient algorithm is the computation of suitable stepsizes α_τ. Exact line search prescribes to compute the stepsize by solving the unidimensional optimization problem

$$\min_\alpha f\left(p^{(\tau)} - \alpha \nabla f\left(p^{(\tau)}\right)\right).$$

Determining the optimal α can, however, be costly in terms of evaluations of objective and gradient. Moreover, an approach based on exact or approximate line search is not suitable for the decentralized implementation that we are seeking.

Barzilai and Borwein [3] proposed an alternative simple and effective technique for the selection of stepsize, which requires

few storage and inexpensive computations. The BB approach prescribes to compute the stepsize according to the formula

$$\alpha_\tau = \frac{\left\| p^{(\tau)} - p^{(\tau-1)} \right\|^2}{(p^{(\tau)} - p^{(\tau-1)})^\top \left(\nabla f(p^{(\tau)}) - \nabla f\left(p^{(\tau-1)}\right) \right)} \quad (5)$$

hence, no line searches or matrix computations are required to determine α_τ. The gradient algorithm with BB stepsizes is particularly well suited for large scale problems, and it is guaranteed to converge to the global optimum in the convex quadratic case (see [9] and [24]). For the nonconvex situation, as it is the case for any algorithm, global convergence cannot in general be guaranteed. As we will show in Section IV-B, the use of BB stepsizes permits a fully distributed implementation of the gradient method.

IV. DISTRIBUTED LOCALIZATION METHOD

We will use graph formalism to describe the distributed structure of the localization problem and of the corresponding algorithm. The next section introduces some useful concepts and definitions related to graphs.

A. Graph Preliminaries

A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is induced by the set of nodes $\mathcal{V} = \{v_1, \ldots, v_n\}$ representing graph vertices and by pairs of nodes (v_i, v_j) such that a relative distance measurement exists between v_i and v_j, representing graph edges in the edge set $\mathcal{E}$. Throughout this paper, we will assume that whenever a distance measurement exists between two nodes, then these two nodes can communicate bidirectionally. Therefore, graph $\mathcal{G}$ describes both the measurement as well as the communication structure of the node formation. For each $i \in \mathcal{V}$, the set of neighbors $\mathcal{N}_i$ of node i is defined as the set of nodes with which node i can exchange information, that is, $\mathcal{N}_i = \{j \in \mathcal{V}, j \neq i : (i, j) \in \mathcal{E}\}$.

We recall that graph $\mathcal{G}$ is said to be connected if a path (i.e., a sequence of edges) exists between any pair of nodes of $\mathcal{G}$ (see standard references on graph theory, such as [6] and [10]). There is a natural way of associating a nonnegative matrix A to a graph $\mathcal{G}$ by considering matrices whose (i, j) entry is positive whenever an edge exists between nodes (i, j) in the corresponding graph, and it is zero otherwise. Let us thus introduce the following set of nonnegative matrices with positive diagonal entries:

$$\mathcal{M} \doteq \{A \in \mathbb{R}^{n,n} : A \geq 0, \ A_{ii} > 0, \ i = 1, \ldots, n\}.$$

We have the following definition.

Definition 1: For $A \in \mathcal{M}$, we say that the matrix/graph pair $(A, \mathcal{G}(\mathcal{V}, \mathcal{E}))$ is compatible if $A_{ij} > 0 \Leftrightarrow (i, j) \in \mathcal{E}$. $\square$

The notion of connectedness of a graph is related to the notion of primitiveness of a matrix compatible with that graph. A square matrix $A \geq 0$ is said to be primitive if there exist an integer $m \geq 1$ such that $A^m > 0$. The least integer m such that $A^m > 0$ is called the index of primitivity of A. If A is primitive, then $\rho(A)$ is an algebraically simple eigenvalue of A, and the eigenspace associated with this eigenvalue has dimension one.

The following theorem can readily be established (using, for instance, [12]).

Theorem 1: Let $A \in \mathcal{M}$ such that $(A, \mathcal{G}(\mathcal{V}, \mathcal{E}))$ is a compatible pair. Then, A is primitive if and only if $\mathcal{G}$ is connected.

Consider a subset of $\mathcal{M}$ composed of matrices in which the sum over each row is equal to 1 (such matrices are usually called (row) stochastic), i.e., $\mathcal{M}_s \doteq \{A \in \mathcal{M} : A\mathbf{1} = \mathbf{1}\}$. For $A \in \mathcal{M}_s$, we have that 1 is an eigenvalue of A. Observe that the spectral radius of a matrix is no larger than any norm of the matrix (see [12]); hence, by taking the ℓ_∞-induced norm, we have that for $A \in \mathcal{M}_s$, it holds that

$$\rho(A) \leq \|A\|_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^{n} |a_{ij}| = 1.$$

Since 1 is an eigenvalue of A, it therefore must be $\rho(A) = 1$. It also follows that if $A \in \mathcal{M}_s$ is primitive, then A has a unique (i.e., an algebraically simple) eigenvalue at 1; hence, all the other eigenvalues have modulus strictly smaller than 1, and the fixed-point subspace $\mathcal{I}(A) \doteq \{x \in \mathbb{R}^n : Ax = x\}$ is one dimensional. Now, consider the subset of $\mathcal{M}_s$ formed by symmetric matrices $\mathcal{M}_{ss} \doteq \{A \in \mathcal{M}_s : A = A^\top\}$ (a matrix $A \in \mathcal{M}_{ss}$ is symmetric, nonnegative with strictly positive diagonal elements and doubly stochastic, that is, $A\mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top A = \mathbf{1}^\top$). For $A \in \mathcal{M}_{ss}$, let $\lambda_1, \ldots, \lambda_n$ be the (real) eigenvalues ordered with nonincreasing modulus; then, we clearly have that $\rho(A) = \lambda_1(A) \equiv \|A\| = 1$, and A can be written in diagonally factored form as

$$A = \frac{1}{n}\mathbf{1}\mathbf{1}^\top + Z, \quad Z = VDV^\top, \quad D = \text{diag}(\lambda_2, \ldots, \lambda_n) \quad (6)$$

where $V \in \mathbb{R}^{n,n-1}$ is such that

$$V^\top V = I_{n-1}, \quad VV^\top = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top, \quad \mathbf{1}^\top V = 0. \quad (7)$$

The following key lemma relates the convergence rate of iterations $z_{k+1} = Az_k$ to the second-largest modulus eigenvalue of matrix A. This result is quite standard in consensus literature (see, e.g., [21] in Section II-D), but we restate and prove it here in our specific context for the purpose of clarity.

Lemma 1: Let $A \in \mathcal{M}_{ss}$ such that $(A, \mathcal{G}(\mathcal{V}, \mathcal{E}))$ is a compatible pair. If $\mathcal{G}$ is connected, then for any $z \in \mathbb{R}^n$ and any integer $k \geq 0$, it holds that

$$\|A^k z - \mathbf{1}\zeta\| \leq |\lambda_2(A)|^k \cdot \|z\|, \quad |\lambda_2(A)| < 1$$

where ζ is the average of the entries in vector z, i.e.,

$$\zeta = \frac{1}{n}\sum_{i=1}^{n} z_i.$$

Proof: Since $A \in \mathcal{M}_{ss}$ implies $A \in \mathcal{M}$, by Theorem 1, we have that $\mathcal{G}$ connected implies that A is primitive; hence, $\lambda_1(A) = 1$, and all the other eigenvalues $\lambda_2, \ldots, \lambda_n$ have modulus strictly smaller than 1. Then, since A is symmetric, we can factorize it according to (6), and using (7), it can easily

be verified that $A^k = (1/n)\mathbf{1}\mathbf{1}^\top + VD^kV^\top$. Now, note that $(1/n)\mathbf{1}\mathbf{1}^\top z = \mathbf{1}\zeta$, hence

$$\left\| A^k z - \frac{1}{n}\mathbf{1}\mathbf{1}^\top z \right\| = \|A^k z - \mathbf{1}\zeta\| = \|VD^kV^\top z\|$$
$$\leq \|VD^kV^\top\| \cdot \|z\| \leq \|D^k\| \cdot \|z\| = |\lambda_2|^k \cdot \|z\|$$

which concludes the proof. ∎

B. Distributed Gradient Algorithm With BB Stepsizes

We here describe a distributed implementation of the update step (3) with BB stepsizes. The approach is divided into two phases: in phase 1, the nodes perform a distributed consensus algorithm to converge to the value (5) of the stepsize α_τ; and in phase 2, each node locally updates its position according to the gradient step (3). The algorithm thus works on two time scales: an outer iteration scale involving the gradient updates, and an inner iteration scale involving the consensus fusion among nodes. This inner phase requires, ideally, an infinite number of consensus iterations for nodes to converge to the true value of the stepsize parameter. However, convergence occurs at geometric rate; hence, a reasonable finite number of iterations suffices in practice for the nodes to approximately agree on the stepsize value. This issue is further discussed in Remark 1. Numerical experiments, using also very coarse stepsize computation (i.e., very few consensus iterations per outer step) support the idea that the algorithm is quite resilient to the choice of stepsize parameter.

1) Distributed Computation of Stepsize: Suppose that each node i knows its own estimated position at the current iteration τ and at the previous one $\tau - 1$, along with the local block of gradient $\nabla_i f(p)$ [see (13)] at the same iterations, with the convention that $p^{(-1)} = 0$, $\nabla_i f(p^{(-1)}) = 0$. At "time" $\tau(0)$, each node i, $i = 1, \ldots, n$, initializes two scalar values

$$\rho_i(\tau(0)) = \left\| p_i^{(\tau)} - p_i^{(\tau-1)} \right\|^2 \tag{8}$$

$$\psi_i(\tau(0)) = \left(p_i^{(\tau)} - p_i^{(\tau-1)} \right)^\top$$
$$\times \left(\nabla_i f\left(p^{(\tau)} \right) - \nabla_i f\left(p^{(\tau-1)} \right) \right) \tag{9}$$

and then starts a series of consensus iterations, exchanging data with its neighbors

$$\rho_i(\tau(t+1)) = W_{ii}\rho_i(\tau(t))$$
$$+ \sum_{j \in \mathcal{N}_i} W_{ij}\rho_j(\tau(t)) \tag{10}$$

$$\psi_i(\tau(t+1)) = W_{ii}\psi_i(\tau(t))$$
$$+ \sum_{j \in \mathcal{N}_i} W_{ij}\psi_j(\tau(t)), \quad t = 0, 1, \ldots \tag{11}$$

where $W \in \mathcal{M}_{\mathrm{ss}}$ is a symmetric doubly stochastic matrix compatible with graph $\mathcal{G}$. The following proposition holds.

Proposition 1: If graph $\mathcal{G}$ is connected, then

$$\lim_{t \to \infty} \frac{\rho_i(\tau(t))}{\psi_i(\tau(t))} = \alpha_\tau, \quad \text{for all } i = 1, \ldots, n.$$

Proof: First, notice that the numerator of α_τ in (5) is given by $\sum_{i=1}^n \rho_i(\tau(0))$, where $\rho_i(\tau(0))$ is defined in (8), and that the denominator of α_τ is given by $\sum_{i=1}^n \psi_i(\tau(0))$, where $\psi_i(\tau(0))$ is defined in (9). We next show that iterations (10) and (11) converge at geometric rate to the average of the $\rho_i(\tau(0))$ values and to the average of the $\psi_i(\tau(0))$ values, respectively; hence, the ratio $\rho_i(\tau(t))/\psi_i(\tau(t))$ converges to α_τ, as claimed. To prove this fact, notice that iteration (10) can be written in compact vector form as follows: $\rho(\tau(t+1)) = W\rho(\tau(t))$ where $\rho(\tau) = [\rho_1(\tau) \cdots \rho_n(\tau)]^\top$. Given the initial vector $\rho(\tau(0))$, the preceding recursion generates at generic iteration $t = 1, 2, \ldots$ the vector $\rho(\tau(t)) = W^t\rho(\tau(0))$. Similarly, (11) generates at iteration t the vector $\psi(\tau(t)) = W^t\psi(\tau(0))$. Since $W \in \mathcal{M}_{\mathrm{ss}}$ is compatible with connected graph $\mathcal{G}$, by Lemma 1, we have that

$$\left\| \rho(\tau(t)) - \frac{1}{n}\sum_{j=1}^n \rho_j(\tau(0))\mathbf{1} \right\| \leq \gamma^t \|\rho(\tau(0))\|$$

where $\gamma < 1$ is the modulus of the second largest modulus eigenvalue of W. Recalling the vector norm inequality $\|x\|_\infty \leq \|x\|$, we also have that

$$\left\| \rho(\tau(t)) - \frac{1}{n}\sum_{j=1}^n \rho_j(\tau(0))\mathbf{1} \right\|_\infty \leq \gamma^t \|\rho(\tau(0))\|$$

that is

$$\left| \rho_i(\tau(t)) - \frac{1}{n}\sum_{j=1}^n \rho_j(\tau(0)) \right| \leq \gamma^t \|\rho(\tau(0))\|, \quad \forall i = 1, \ldots, n$$

and similarly, we obtain that

$$\left| \psi_i(\tau(t)) - \frac{1}{n}\sum_{j=1}^n \psi_j(\tau(0)) \right| \leq \gamma^t \|\psi(\tau(0))\|, \quad \forall i = 1, \ldots, n$$

which shows that, as t tends to infinity, $\rho_i(\tau(t))$, $\psi_i(\tau(t))$ converge at geometric rate to the averages of the entries in vectors $\rho(\tau(0)), \psi(\tau(0))$, respectively. From this, it follows that:

$$\lim_{t \to \infty} \frac{\rho_i(\tau(t))}{\psi_i(\tau(t))} = \frac{\sum_{j=1}^n \rho_j(\tau(0))}{\sum_{j=1}^n \psi_j(\tau(0))} = \alpha_\tau$$

thus, proving the claim. ∎

Remark 1: From Proposition 1, it follows that, at inner iteration t, each node can locally construct the ratio

$$\alpha_i(\tau(t)) \doteq \frac{\rho_i(\tau(t))}{\psi_i(\tau(t))} \tag{12}$$

and use it as an approximation of the stepsize α_τ. The numerator and denominator of $\alpha_i(\tau(t))$ geometrically converge to the average of the entries in vectors $\rho(\tau(0)), \psi(\tau(0))$, respectively, and the convergence rate is dictated by γ, the modulus of the

second largest modulus eigenvalue of matrix W. We next show that, for t larger than some finite $\bar{t}$, also the convergence of $\alpha_i(\tau(t))$ to α_τ is at least geometric and still dictated by γ. It follows that whenever the network topology is known and γ can be computed *a priori*, then fixing a maximum number T of inner iterations induces a predictable error in the estimation of the stepsize parameter. This error can be made as small as desired by a suitable choice of (finite) T. Indeed, inspecting the proof of Proposition 1 and recalling Lemma 1, we see that $\rho_i(\tau(t)) = \bar{\rho} + e_i^\top V D^t V^\top \rho(\tau(0))$, $\psi_i(\tau(t)) = \bar{\psi} + e_i^\top V D^t V^\top \psi(\tau(0))$, where e_i is the ith unit vector, and we defined $\bar{\rho} = (1/n)\sum_{j=1}^n \rho_j(\tau(0))$, $\bar{\psi} = (1/n)\sum_{j=1}^n \psi_j(\tau(0))$. Recalling that $\alpha_\tau = \bar{\rho}/\bar{\psi}$, we next construct the relative convergence error. Let $\eta > 0$ be a small constant, and let $\bar{t}$ be the smallest integer larger than $(\log \eta - \log \|\psi(\tau(0))\|)/\log \gamma$. Then

$$
\begin{aligned}
\varepsilon_i(t) &= \frac{|\alpha_i(\tau(t)) - \alpha_\tau|}{|\alpha_\tau|} \\
&= \frac{\left|e_i^\top V D^t V^\top \left(\alpha_\tau^{-1}\rho(\tau(0)) - \psi(\tau(0))\right)\right|}{\left|\bar{\psi} + e_i^\top V D^t V^\top \psi(\tau(0))\right|} \\
&\leq \frac{\|e_i\| \cdot \|V D^t V^\top\| \cdot \left\|\alpha_\tau^{-1}\rho(\tau(0)) - \psi(\tau(0))\right\|}{|\bar{\psi}| - \|e_i\| \cdot |V D^t V^\top| \cdot \|\psi(\tau(0))\|} \\
&\leq \frac{\gamma^t \left\|\alpha_\tau^{-1}\rho(\tau(0)) - \psi(\tau(0))\right\|}{|\bar{\psi}| - \gamma^t \|\psi(\tau(0))\|} \leq \text{const.}\gamma^t, \quad \text{for } t \geq \bar{t}
\end{aligned}
$$

where const. $= \|\alpha_\tau^{-1}\rho(\tau(0)) - \psi(\tau(0))\|/|\bar{\psi}| - \eta$, which shows that the modulus of relative error decreases at least at geometric rate. A practical implementation of the algorithm works by fixing *a priori* a sufficiently large T and running iterations (10) and (11) up to $t \leq T$. An analysis on the effect of the choice of T on the algorithm performance is presented via numerical simulations in Section V-B.

a) Choice of weights: Note that, to being able to update its estimate, node i needs to know its current neighbors' weights W_{ij}. These coefficients may be imposed *a priori* on the nodes (but this would require a centralized *a priori* knowledge of the network topology) or, more practically, negotiated autonomously online by the nodes. A standard and effective rule for building the weights autonomously is the so-called Metropolis rule (see [33] and [34]), which prescribes weights as follows:

$$
W_{ij} = \begin{cases} \frac{1}{\max(|\mathcal{N}_i|,|\mathcal{N}_j|)}, & \text{if } (i,j) \in \mathcal{E}, \ i \neq j \\ 1 - \sum_{j \in \mathcal{N}_i \backslash i} W_{ij}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}
$$

where $|\mathcal{N}_i|$ denotes the cardinality of $\mathcal{N}_i$. This choice of weights is indeed well suited for distributed implementation since each node only needs to know the number of its neighbors and exchange information with them. With these weights, iterations (10) and (11) are based on local-only information, which means that each node can execute the update without need of knowing the global structure of the network or even the number n of nodes composing the network.

2) Distributed Gradient Update: Let $\nabla_i f(p)$ denote the ith 1×2 block in the gradient $\nabla f(p)$ in (4). Given the specific structure of $\nabla g_{ij}(p)$, which is nonzero only for the blocks in position i and j that are equal to $(p_i - p_j)$ and $(p_j - p_i)$, respectively, it is immediate to verify that

$$
\nabla_i f(p) = \sum_{j \in \mathcal{N}_i} (p_i - p_j)g_{ij}(p) \tag{13}
$$

where $\mathcal{N}_i \doteq \{j : (i,j) \in \mathcal{E}\}$ is the set of neighbors of node i in graph $\mathcal{G}$. Observe next from (2) that $g_{ij}(p)$ is actually a function of $p_i - p_j$ only, which represents the current mismatch between $\|p_i - p_j\|^2$ and d_{ij}^2. It follows that the portion of gradient $\nabla_i f(p)$ can be computed individually by node i by simply querying the neighbors for their current estimated positions.

Assuming that the stepsize α_τ is known at each node, each node would be able to locally update its estimated position according to the distributed gradient rule

$$
\begin{aligned}
p_i^{(\tau+1)} &= p_i^{(\tau)} - \alpha_\tau \nabla_i f\left(p^{(\tau)}\right) \\
&= p_i^{(\tau)} - \alpha_\tau \sum_{j \in \mathcal{N}_i} (p_i - p_j)^\top g_{ij}(p), \quad i = 1,\dots,n. \tag{14}
\end{aligned}
$$

In practice, the value of α_τ is known only approximately at each node as a result of the finite number T of consensus iterations. Summarizing, the practical distributed localization algorithm works as follows.

Algorithm 1 (Distributed localization algorithm):

---

0. (**Initialization**) Fix T (maximum number of consensus iterations), τ_{wup} (maximum number of warm-up iterations), $\tau_{\max}$ (maximum number of gradient updates), and η_{abstol} (absolute update tolerance). Let $\tau = 0$ and set the position estimates p_i^τ, for $i = 1,\dots,n$ to be the given initial guess for optimization. Mark all nodes as "active".
1. (**Warm-up**)
   (1.a) Fix $\alpha_\tau = \tilde{\alpha}$ (a small number, say $\tilde{\alpha} = 10^{-6}$, equal for all nodes);
   (1.b) Update all node position estimates according to (14), and let $\tau = \tau + 1$;
   (1.c) If $\tau < \tau_{\text{wup}}$, then goto (1.a), else goto 2.
2. (**Consensus iterations**)
   (2.a) Let $t = 0$, and initialize $\rho_i(\tau(t))$ and $\psi_i(\tau(t))$ as in (8), (9);
   (2.b) Update ρ_i, ψ_i according to (10), (11); let $t = t + 1$;
   (2.c) If $t < T$, then goto (2.a), else goto 3.
3. (**Gradient update**) At each active node:
   (3.a) Compute $\alpha_i(\tau(t))$ according to (12);
   (3.b) Update position estimate according to (14);
   (3.c) If $\|p_i^{(\tau)} - p_i^{(\tau-1)}\| \leq \eta_{\text{abstol}}$, then mark node i as "inactive";
4. (**Stopping criterion**) Let $\tau = \tau + 1$. If $\tau \geq \tau_{\max}$, or all nodes are inactive, then exit; otherwise goto 2.

It is worth noticing that the warm-up is only useful to have reasonable values for the correction in the first update ($\tau = 0$): since $p^{(-1)}$, $\nabla_i f(p^{(-1)})$ are chosen to be zero by convention (see Section IV-B-1), the first iteration can produce arbitrary values of the stepsize and can lead the local estimate outside of the region of attraction of the global minimum of the objective function. In practice, numerical tests suggest that it often suffices to fix $\tau_{\mathrm{wup}} = 1$ to have good convergence properties for the algorithm (see Section V-B).

V. NUMERICAL EXPERIMENTS

We now present some numerical tests on decentralized network localization with the proposed distributed gradient method. In the first experiment (Section V-A), we verify the correctness of our theoretical analysis on a small formation, showing the convergence of the stepsizes, computed locally at each node, to a common value, which is actually the centralized stepsize α_τ. In the second experiment (Section V-B), we investigate the performance of the distributed gradient method on a large networked system, showing the advantages of integrating a distributed averaging technique in the localization process; moreover, we analyze the scalability of the proposed approach. Finally, in Section V-C, we apply the technique to a realistic setup in which nodes in a sensor network use (eventually without any synchronization) the distributed gradient method for refining an inaccurate position guess, obtained through a preliminary semidefinite optimization-based localization method.

A. Example 1

Let us consider a first setup that exemplifies the case of a team of autonomous vehicles or mobile robots moving in formation where few agents are equipped with GPS, whereas others use dead reckoning for position estimation. Using dead reckoning progressively decreases the localization accuracy; hence, the agents may periodically stop and use the relative range measurements to improve their position estimate. Distance measurements are typically obtained through acoustic beacons [22], radio frequency devices [36], or other "time of flight" sensors.

For a numerical example, we considered the case of $n = 10$ agents located on terrain according to the configuration shown in Fig. 1(b). This actual configuration should be estimated autonomously by the agents using as prior knowledge the ideal position configuration shown in Fig. 1(a), which is assumed as the initial position guess by the agents. Three anchor nodes are selected at the external vertices of the formation.

To measure the localization effectiveness, we define the local positioning error $\phi_i(\tau)$ at node i as the Euclidean distance between the estimated position $p_i^{(\tau)}$ at iteration τ and the true position p_i of the node. The global positioning error $\Phi(\tau)$ is further defined as the mean value of the local positioning errors of all the nodes in the network

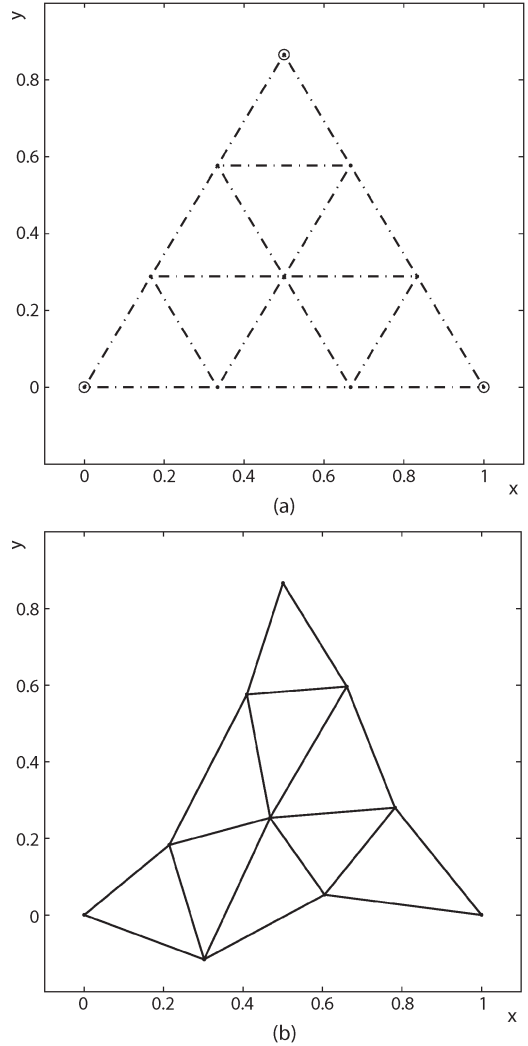$$\Phi(\tau) = \frac{1}{n} \sum_{i=1}^{n} \left\| p_i - p_i^{(\tau)} \right\|.$$

Fig. 1. Triangle formation: (a) initial position guess and (b) actual node configuration ($n = 10$).

At each update iteration, the position estimate of the ith node is corrected according to the recursion (14), initialized with initial position estimate $p_i^{(0)}$ corresponding to the configuration in Fig. 1(a). Before each update, at iteration τ, each node starts inner consensus iterations $\tau(t)$, $t = 0, 1, \ldots, T$, to compute the BB stepsize α_τ in a decentralized fashion.

Fig. 2(a) shows, at some given iteration τ, how local stepsizes

$$\alpha_i \left(\tau(t) \right) = \frac{\rho_i \left(\tau(t) \right)}{\psi_i \left(\tau(t) \right)}$$

computed according to the scheme presented in Section IV-B-1 converge to the desired value α_τ equal for all nodes.

It is possible to notice that, after a small and fixed number of consensus iterations, the nodes agree on a common value for the stepsizes (see Algorithm 1). Further tests on the influence of T on the performances of the algorithm are reported in Section V-B. Fig. 2(b) reports the local positioning error for each nonanchor node using $T = 20$. As expected, in a noiseless setup, the estimated positions converge to the true location in a reasonable number of τ iterations. In practice, each node becomes inactive after a threshold condition is met (see
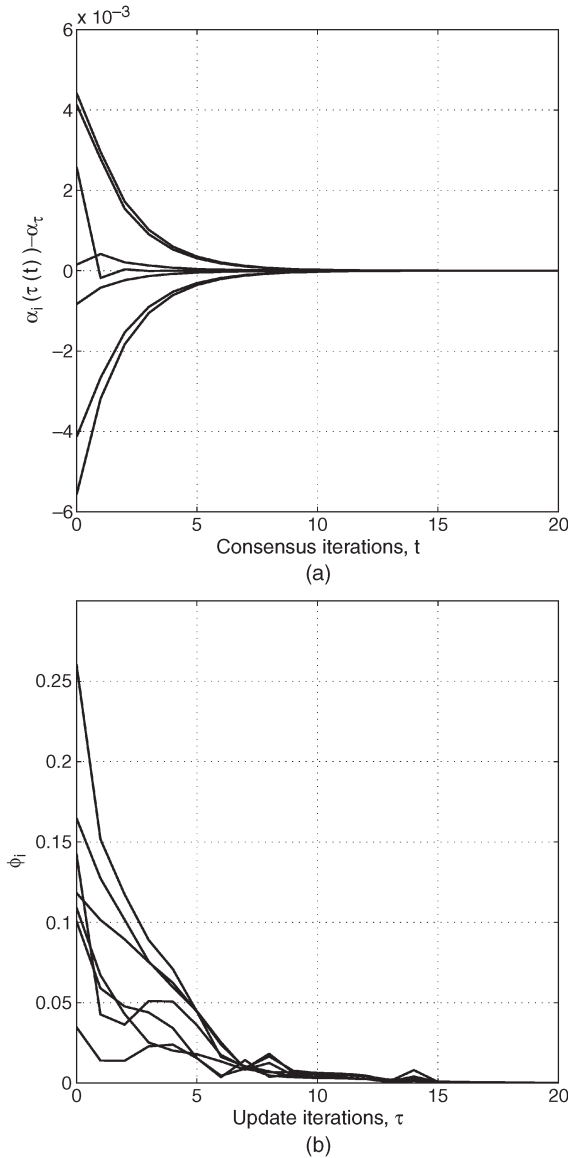
Fig. 2. (a) Local stepsizes $\alpha_i(\tau(t))$ of the agents compared with the centralized stepsizes α_τ at some given iteration τ. (b) Local positioning errors for the nonanchor nodes in the formation versus update iterations count τ; the number of consensus iterations is $T = 20$.

Algorithm 1); hence, the approach can be used with no need of synchronization among nodes: each node interrogates the neighbors T times for each update, and it may stop to correct its position estimate autonomously, with no need of global information (further details can be found in Section V-C).

B. Example 2

For the second experiment, we consider a graph with n nodes disposed in lattice configuration on the unit square $[0, 1] \times [0, 1]$. Four anchor nodes are selected on the vertices of the unit square. The actual network configuration to be estimated is taken as the regular lattice configuration [Fig. 3(a)], whereas the initial position guess is generated by randomly perturbing the position of each node [see Fig. 3(b)]. Diagonal edges are added to the lattice structure for guaranteeing rigidity of the
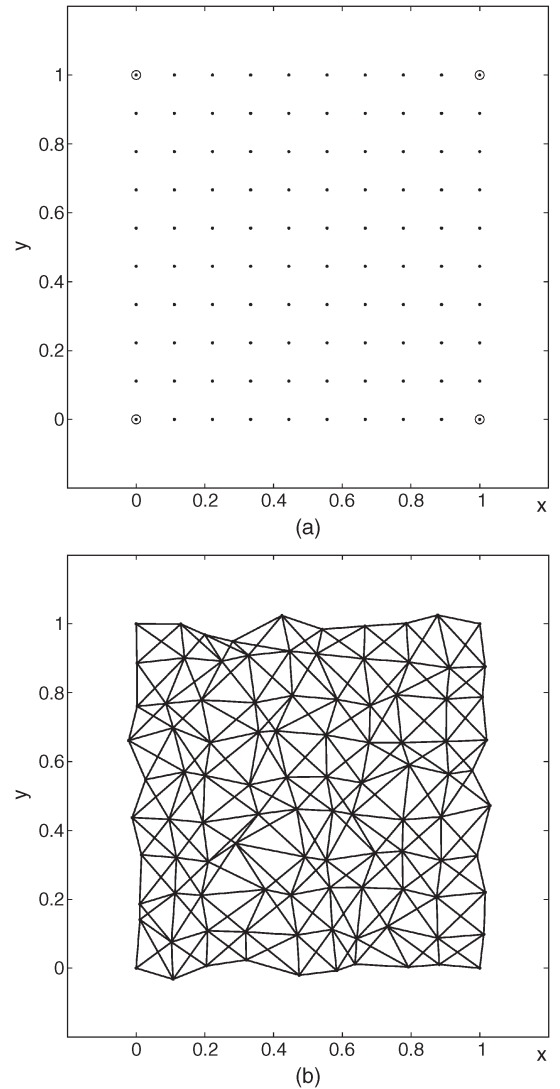

Fig. 3. Lattice configuration: (a) actual node configuration and (b) initial position guess ($n = 100$).

underlying graph, hence enabling the nodes to retrieve the correct configuration.

To evaluate the advantages of the proposed method, we compare it with other gradient-based techniques for network localization. The first compared technique simply employs a fixed stepsize for gradient update: each node communicates with the neighbors to compute the local gradient, and the update (3) is performed using a stepsize $\alpha_\tau = \bar{\alpha}$, $\bar{\alpha} > 0$. The second technique can be seen as a limit condition of the proposed approach, in which the agents may perform no consensus at all ($T = 0$), and each node computes its own stepsize according to the local information

$$\alpha_{\tau,i} = \frac{\left\| p_i^{(\tau)} - p_i^{(\tau-1)} \right\|^2}{\left(p_i^{(\tau)} - p_i^{(\tau-1)} \right)^\top \left(\nabla_i f\left(p_i^{(\tau)} \right) - \nabla_i f\left(p_i^{(\tau-1)} \right) \right)}. \tag{15}$$

Fig. 4 shows the global positioning error as a function of update iterations τ, comparing the proposed approach (with $T = 20$) with the two gradient techniques, for a network of $n = 100$
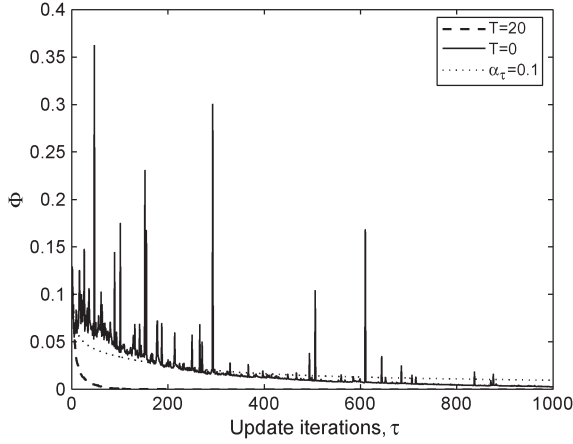
Fig. 4. (a) Global positioning error versus iterations τ for the limit case without consensus (solid line), for the proposed distributed gradient method with $T = 20$ (dashed line), and for the gradient method with fixed stepsizes $\alpha_\tau = 0.1$ (dotted line).
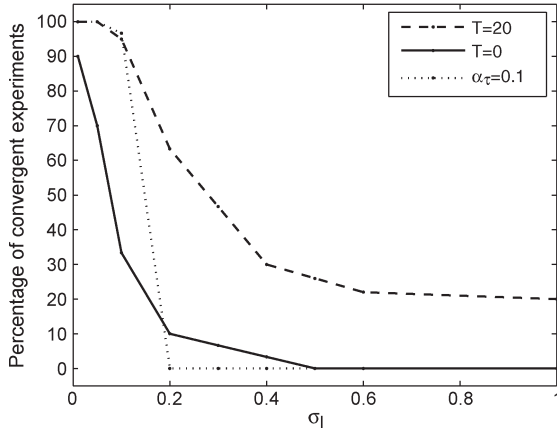

Fig. 5. Percentage of convergent experiments for the case with no consensus (solid line), for the proposed distributed gradient method with $T = 20$ (dashed line), and for the gradient method with fixed stepsizes $\alpha_\tau = 0.1$ (dotted line).

nodes. It can be seen how the use of distributed averaging improves the convergence rate of the gradient method (results are averaged over 50 Monte Carlo runs). The use of local stepsizes, instead, causes the presence of spikes in the localization error since there is no agreement on the correction to be applied at a single node, whereas the use of fixed stepsizes corresponds to long tails in the localization errors and poor convergence rates. The three compared techniques are local since they require a suitable initial guess for attaining the global minimum of the nonlinear objective function. In Fig. 5, we report the percentage of convergent experiments for the three techniques, with different initial guesses. The initial guess of each node is drawn from a normal distribution with mean corresponding to the actual node position and covariance equal to $P = \sigma_I^2 I_2$; hence, for increasing values of σ_I (x-axis in the figure), the quality of the initial guess worsens. The figure highlights how the use of the proposed technique presents further advantages in terms of convergence properties with respect to the other compared gradient-based approaches.

We now want to evaluate the total number of update iterations, say, τ_{end}, required for network localization using the aforementioned techniques. The total number of iterations is

TABLE I
COMPUTATIONAL EFFORT AND COMMUNICATION BURDEN FOR THE LIMIT CASE WITHOUT CONSENSUS, FOR THE PROPOSED DISTRIBUTED GRADIENT METHOD WITH $T = 20$, FOR GRADIENT METHOD WITH FIXED STEPSIZES, AND FOR GRADIENT METHOD WITH SPECIFIED STEP LENGTHS. RESULTS ARE AVERAGED OVER 50 MONTE CARLO RUNS. (*) ITERATIONS ARE STOPPED BECAUSE THE MAXIMUM NUMBER OF ALLOWED UPDATES ($\tau_{\max}$) IS REACHED

	Required update iterations (τ_{end})	Communication rounds
Distributed gradient method with $T = 20$	262	5240
Gradient method with local stepsizes ($T = 0$)	6071	6071
Gradient method with stepsizes $\alpha_\tau = 0.1$	23491	23491
Gradient method with stepsizes $\alpha_\tau = 0.01$	10^5(*)	10^5(*)
Gradient method with step length $k = 10/\sqrt{\tau}$	10^5(*)	10^5(*)

connected to the amount of computation and communication burden required to the nodes; hence, it is crucial in low-cost low-power applications. We consider the value $\eta_{abstol} = 10^{-10}$ as threshold for a node to stop the update iterations, and $\tau_{\max} = 10^5$ (see Algorithm 1). In Table I, we report the values of τ_{end} when using the proposed technique, a gradient method with local BB stepsizes (15), and a gradient method with fixed stepsizes ($\alpha_\tau = 0.1$ and $\alpha_\tau = 0.01$). We further include in the analysis a gradient method with specified step length, i.e., with a stepsize $\alpha_\tau = k/\|\nabla_i f(p^{(\tau)})\|$, $k > 0$. This last strategy can be employed with a constant $k = \bar{k}$ or with a diminishing sequence of step lengths, like $k = \bar{k}/\tau$ or $k = \bar{k}/\sqrt{\tau}$ (with $\bar{k} > 0$) (see [31]). Obviously, due to the stopping criterion for the update iterations, the number of updates performed using specified step lengths can be known *a priori*; when using $k = \bar{k}/\sqrt{\tau}$, for instance, the iterations will stop when $\tau \geq (\bar{k}/\eta_{abstol})^2$.

In Table I, we also report the communication burden for the compared techniques.

The proposed approach requires each node to communicate T times with the neighbors for each update (each communication allows a generic node i to broadcast the local information to the neighbors). On the other hand, the compared gradient methods (local stepsizes, fixed stepsizes, specified step lengths) require nodes to communicate with their neighbors only one time for each update to compute the local gradient $\nabla_i f(p^{(\tau)})$. It is possible to observe that, despite the communication load due to consensus, the proposed approach is still advantageous over the compared techniques.

In the previous experiments, we considered $T = 20$ consensus iterations for each update. Now, we discuss such choice, reporting some results on how the total number of update iterations varies in function of the number of consensus iterations T. In Fig. 6, we show the value of τ_{end} for different values of T and for different network sizes. It is worth noticing that, also for small values of T, the introduction of the consensus phase contributes to remarkably reduce the total amount of updates, and there is no need to consider a large number of consensus iterations (convergence is exponential, see Remark 1).

As a conclusion of this section, we discuss the scalability of the proposed approach. Scalability issues are particularly
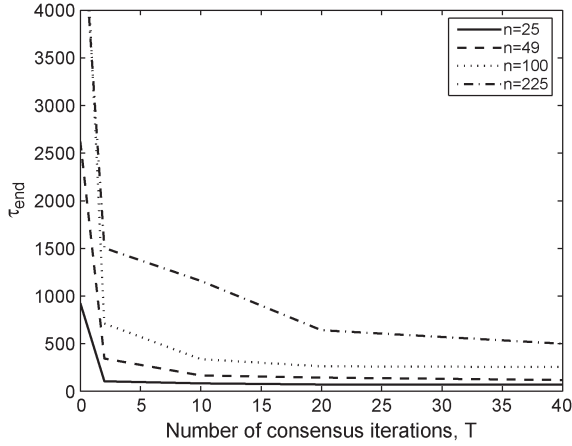
Fig. 6. Number of update iterations τ_{end} required for network localization using different values of T. Results are provided for lattice configurations with $n = 25, 49, 100, 225$ nodes.
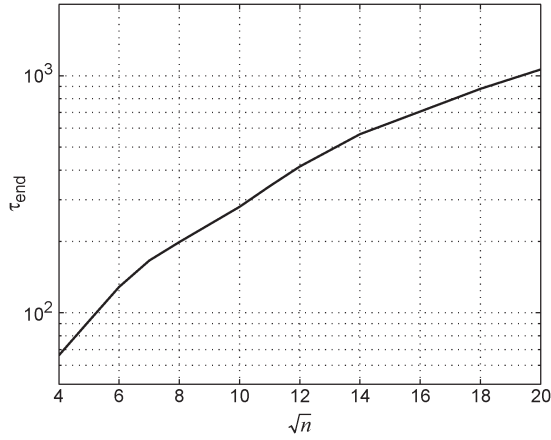

Fig. 7. Total number of update iterations τ_{end} for increasing network sizes.

TABLE II
GLOBAL LOCALIZATION ERROR WHEN USING SDP AND SDP WITH DISTRIBUTED GRADIENT REFINEMENT (SDP-DG); THE TABLE SHOWS RESULTS FOR DIFFERENT VALUES OF THE NOISE FACTOR ν_d AND FOR DIFFERENT NUMBER OF ANCHOR NODES n_a

			ν_d		
			0.05	0.1	0.15
n_a	20	SDP	0.010	0.017	0.023
		SDP+DG	0.006	0.013	0.018
	10	SDP	0.020	0.033	0.040
		SDP+DG	0.013	0.023	0.032
	3	SDP	0.094	0.145	0.170
		SDP+DG	0.046	0.127	0.156

smaller than a sensing radius R. It has been proved in [7] that if $R > 2\sqrt{2}\sqrt{\log(n)/n}$, the graphs produced by the previous technique are ggr with high probability.

We consider the configuration generated as previously described as the "true" configuration (which is of course unknown in practice), and then, we use the distance measurements from this configuration as the data for the numerical tests; distance measurements are affected by noise in the form

$$\bar{d}_{ij} = \left| d_{ij} + \epsilon_{ij}^d \right| \ \forall \, (i,j) \in \mathcal{E} \tag{16}$$

where d_{ij} is the true distance among nodes i and j, $\bar{d}_{ij}$ is the corresponding measured quantity, and ϵ_{ij}^d is a zero mean white noise with standard deviation σ_{ij}^d. According to [5], we choose $\sigma_{ij}^d = \nu_d d_{ij}$, where the constant ν_d, which is often referred to as noise factor, defines how measurement accuracy decreases with the actual distance. The sensing radius is $R = 0.3$, and n_a anchor nodes are randomly selected among the nodes. The parameters used for the distributed gradient method are $T = 20$, $\eta_{\mathrm{abstol}} = 10^{-15}$, and $\tau_{\mathrm{max}} = 10^5$.

In Table II, we report the global localization error for different ν_d considering the network in Fig. 8. It is possible to see that the introduction of distributed gradient refinement improves the localization accuracy, leading to an error reduction of $15 \div 50\%$. In particular, the distributed gradient method can drastically improve the localization performance in case few anchor nodes are present in the formation or when the anchors lie in the interior on the formation: in such situation, the SDP localization is known to be inaccurate, and the local refinement is essential for practical use (see Fig. 8 for a meaningful example).

It is worth noticing that the proposed distributed technique assumes that some sort of synchronization among the nodes in the network exists, i.e., there is a global clock, say, t^g, such that for $t^g = 0$ all the nodes start the warm-up phase, and at $t^g = kT$, $k = 1, 2, \ldots$, they all update local estimates, iterating consensus at each discrete time in the interval $[(k-1)T, kT]$, $k = 1, 2, \ldots$ Although such underlying assumption is common to several distributed techniques (see [2], for instance), we now want to evaluate what happens when relaxing this hypothesis. For this purpose, we consider a setup in which each node may start iterations at a different time t_i^g and alternates consensus and update phases based on a local clock t_i^l; after the warm-up, each node will interrogate neighbors T times (during which, possibly, some neighbors may change their estimates) and then apply a local gradient update, according to (14). Before intro-
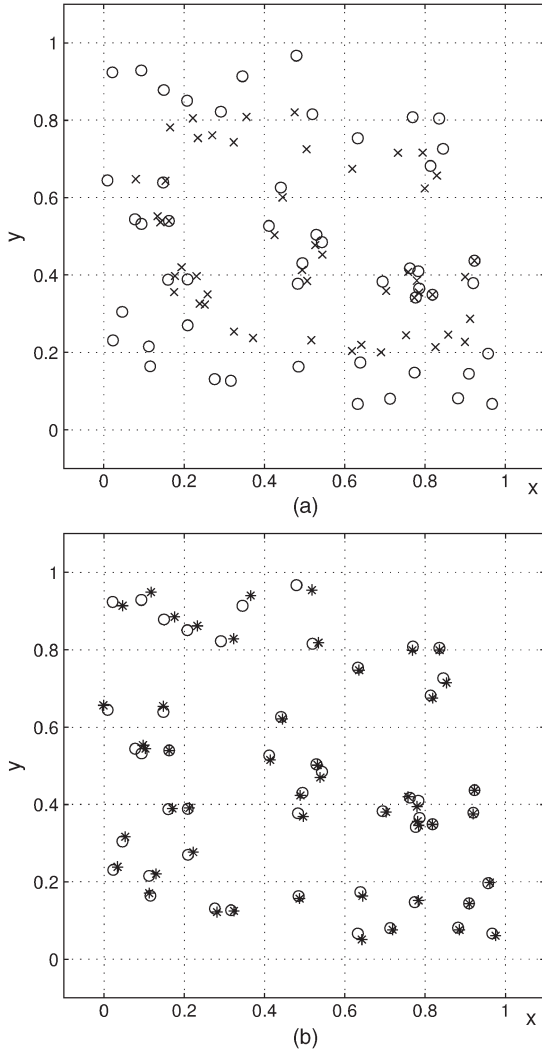
relevant when dealing with sensor networks in which a large number of nodes have to perform localization with limited bandwidth and reduced computational resources. Moreover, this issue is relevant when considering swarms of interacting robots, as discussed, for instance, in [25]. For evaluating the scalability of the approach, we tested the distributed gradient algorithm for different network sizes. Fig. 7 shows the total number of iterations τ_{end} for networks with n nodes and $T = 20$.

C. Example 3

In this section, we evaluate the use of the proposed technique as a refinement phase after SDP is used for computing a rough estimate of node positions. It is well known that the SDP relaxation can be inaccurate in the presence of measurements noise (see [4] and [5]); hence, we use the output of SDP as initial guess for a distributed gradient-based refinement. The idea of a local refinement of a global solution is not new (see [17] and [26]), although our approach has the advantages of being fully distributed and effective in terms of communication and computational requirements.

To work on more realistic networks, we now consider random geometric graphs, i.e., graphs in which nodes are deployed at random in the unit square, and an edge exists between a pair of nodes if and only if their geometrical distance is

	Δt_{max}				
	10	20	30	50	100
Φ for SDP	0.096	0.095	0.099	0.097	0.094
Φ for SDP+DG	0.053	0.053	0.055	0.060	0.063
τ_{end}	1681	1915	1859	1787	1882

Table III reports the global localization error and the average number of updates for the distributed gradient method. It is possible to see that there is no remarkable degradation in performance, hence confirming that the algorithm is suitable for decentralized asynchronous operation.

VI. CONCLUSION

In this paper, we presented a distributed gradient method for solving the network localization problem from relative distance measurements. First, the localization problem is formulated as minimization of a nonlinear cost, and the formulation of a centralized gradient method is recalled, focusing the attention on the use of BB stepsizes. Then, a distributed scheme is proposed for allowing each node to autonomously correct its position estimate by communicating only with its neighbors. The decentralized technique iterates two phases: 1) a consensus phase and an 2) update phase. In the consensus phase, each node iteratively consults and exchanges information with its neighbors to converge approximately to some parameter value (the stepsize α_τ in the gradient method) that must be common and available to all nodes for an update step to be possible. In the update phase, this common parameter is used to actually update the current position estimate. The distributed gradient method is proved to converge to the same solution of its centralized counterpart, while providing the benefits of a fully decentralized scheme that can be implemented autonomously by the networked agents. Moreover, extensive numerical experiments highlighted the advantages of using the proposed technique in terms of convergence properties, computational cost, and communication burden for the nodes in the network. The algorithm is also suitable for the case in which no synchronization exists among the nodes, and it is shown to be scalable in the network size since it requires inexpensive computation and low memory storage.

Fig. 8. (a) Estimated versus actual configuration for SDP localization (circles denote actual node positions, whereas crosses correspond to SDP estimates). (b) Estimated versus actual configuration for SDP localization with distributed gradient refinement (circles denote actual node positions, whereas stars correspond to estimates after refinement). The reported estimates are obtained in the same Monte Carlo run with $\nu_d = 0.05$, $T = 20$, and $\eta_{abstol} = 10^{-15}$.

ducing our numerical tests, we define the maximum clock skew $\Delta t_{max} = \max_{i,j} |t_i^g - t_j^g|$. Obviously, the case of $\Delta t_{max} = 0$ corresponds to a perfect synchronization among the nodes. On the other hand, for increasing values of the maximum clock skew, the delays between the nodes in the network can be relevant.

For numerical tests, we studied the behavior of the distributed gradient method considering different maximum skew times as follows: given Δt_{max}, the starting time t_i^g of the ith node is randomly sampled in the interval $[0; \Delta t_{max}]$, and the local clock is initialized accordingly [when the local clock starts the node becomes active). Before starting consensus iterations, a node verifies that also the neighbors are active; otherwise, it remains in the warm-up phase (such procedure assures that the results of the consensus are meaningful, since inactive nodes cannot properly define the quantities (8) and (9)]. In the reported tests, each node uses the value $T = 20$ for the consensus iterations and autonomously stops when the correction of the local estimates drops below $\eta_{abstol} = 10^{-15}$.

REFERENCES

[1] P. Barooah, "Estimation and control with relative measurements: Algorithms and scaling laws," Ph.D. dissertation, Univ. California, Santa Barbara, CA, 2007.
[2] P. Barooah and J. P. Hespanha, "Estimation on graphs from relative measurements," *IEEE Control Syst. Mag.*, vol. 27, no. 4, pp. 57–74, Aug. 2007.
[3] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.*, vol. 8, no. 1, pp. 141–148, 1988.
[4] P. Biswas, T. Liang, K. Toh, T. Wang, and Y. Ye, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, Oct. 2006.
[5] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proc. 3rd Int. Symp. IPSN*, 2004, pp. 2673–2684.

[6] R. Diestel, *Graph Theory*. New York: Springer-Verlag, 2005.

[7] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization," in *Proc. IEEE INFOCOM*, 2004, vol. 4, pp. 2673–2684.

[8] W. H. Foy, "Position-location solutions by Taylor-series estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-12, no. 2, pp. 187–194, Mar. 1976.

[9] A. Friedlander, J. M. Martinez, and M. Raydan, "A new method for large-scale box constrained convex quadratic minimization problems," *Optim. Methods Softw.*, vol. 5, pp. 57–74, 1995.

[10] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York: Springer-Verlag, 2001.

[11] B. Hendrickson, "The molecule problem: Exploiting structure in global optimization," *SIAM J. Optim.*, vol. 5, no. 4, pp. 835–857, 1995.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[13] M. D. Ili, L. Xie, U. A. Khan, and J. M. F. Moura, "Modeling of future cyber-physical energy systems for distributed sensing and control," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 4, pp. 825–838, Jul. 2010.

[14] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 2000–2016, May 2009.

[15] U. A. Khan, S. Kar, and J. M. F. Moura, "Diland: An algorithm for distributed sensor localization with noisy distance measurements," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1940–1947, Mar. 2010.

[16] R. V. Kulkarni and G. K. Venayagamoorthy, "Bio-inspired algorithms for autonomous deployment and localization of sensor nodes," *IEEE Trans. Syst., Man, Cybern. , Part C*, vol. 40, no. 6, pp. 663–675, Nov. 2010.

[17] T. C. Liang, T. C. Wang, and Y. Ye, "A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization," Dept. Manage. Sci. Eng., Stanford Univ., Stanford, CA, Tech. Rep., 2004.

[18] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, Jul. 2007.

[19] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. 2nd ACM Conf. SenSys*, 2004, pp. 50–61.

[20] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer-Verlag, 2006.

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

[22] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *IEEE J. Ocean. Eng.*, vol. 31, no. 4, pp. 949–958, Oct. 2006.

[23] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sens. Syst.*, 2003, pp. 340–341.

[24] M. Raydan, "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem," *SIAM J. Optim.*, vol. 7, no. 1, pp. 26–33, 1997.

[25] J. A. Rothermich, M. I. Ecemis, and P. Gaudiano, "Distributed localization and mapping with a robotic swarm," in *Proc. Swarm Robot.*, 2004, pp. 58–69.

[26] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proc. USENIX Annu. Tech. Conf.*, 2002, pp. 317–327.

[27] J. B. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," in *Proc. 17th Allerton Conf. Commun., Control Comput.*, 1979, pp. 480–489.

[28] I. Shames, B. Fidan, and B. D. O. Anderson, "Minimization of the effect of noisy measurements on localization of multi-agent autonomous formations," *Automatica*, vol. 45, no. 4, pp. 1058–1065, Apr. 2009.

[29] J. Sheu, W. Hu, and J. Lin, "Distributed localization scheme for mobile sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 4, pp. 516–526, Apr. 2010.

[30] Q. Shi, C. He, H. Chen, and L. Jiang, "Distributed wireless sensor network localization via sequential greedy optimization algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3328–3340, Jun. 2010.

[31] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. New York: Springer-Verlag, 1985, ser. Springer Series in Computational Mathematics.

[32] R. G. Stanfield, "Statistical theory of DF finding," *J. IEE*, vol. 94, no. 5, pp. 762–770, 1947.

[33] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65–78, 2004.

[34] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Int. Conf. IPSN*, 2005, pp. 63–70.

[35] C. Xunxue, S. Zhiguan, and L. Jianjun, "Distributed localization for anchor-free sensor networks," *J. Syst. Eng. Electron.*, vol. 19, no. 3, pp. 405–418, 2008.

[36] J. Zhou and J. Shi, "Performance evaluation of object localization based on active radio frequency identification technology," *Comput. Ind.*, vol. 60, no. 9, pp. 669–676, Dec. 2009.

Giuseppe C. Calafiore received the "Laurea" degree in electrical engineering and the Ph.D. degree in information and system theory from Politecnico di Torino, Torino, Italy, in 1993 and 1997, respectively.

He held several visiting positions at international institutions: at Information Systems Laboratory, Stanford University, Stanford, CA, in 1995; at Ecole Nationale Supérieure de Techniques Avancées (ENSTA), Paris, France, in 1998; and at the University of California at Berkeley, Berkeley, in 1999, 2003, and 2007; at the Institute of Pure and Applied Mathematics, University of California at Los Angeles, Los Angeles, in 2010; and at the John Von Neumann Institute, Vietnam National University, in 2011. Since 1998, he has been with the faculty of Dipartimento di Automatica e Informatica, Politecnico di Torino, where he currently serves as an Associate Professor. He is the author of more than 120 journal and conference proceedings papers and of seven books. His research interests are in the fields of convex optimization, randomized algorithms, identification, and control of uncertain systems, with applications ranging from finance and economic systems to robust control, pattern recognition, and robotics.

Dr. Calafiore is an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS (T-SMC), the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING (T-ASE), and the *Journal Européen des Systèmes Automatisés*.

Luca Carlone received the Bachelor degree in mechatronic engineering in 2006 from Politecnico di Torino, Torino, Italy, and the M.S. degree in mechatronic engineering and the M.S. degree in automation engineering in 2008 from Politecnico di Milano, Milan, Italy, where he is currently working toward the Ph.D. degree.

In 2010, he was a Visiting Student with the University of Zaragoza, Zaragoza, Spain, and in 2011, he held a Visiting Student position with the University of California, Santa Barbara. His research interests include localization, mapping, and SLAM algorithms for mobile robots navigation.

Mingzhu Wei received the M.S. degree in instrument science and technology from Harbin Institute of Technology, Harbin, China, in 2008 and the B.S. degree in instrument science and technology from Yanshan University, Qinhuangdao, China, in 2006. He is currently working toward the Ph.D. degree with Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy.

Her research interests include cooperative distributed techniques for estimation and optimization over a network of communicating agents.