Distributed Localization from Relative Noisy Measurements: a Robust Gradient Based Approach

Marco Todescato Andrea Carron Ruggero Carli Luca Schenato

***Abstract*— In this work we address the problem of optimal estimating the position of each agent in a network from relative noisy vectorial distances with its neighbors. Although the problem can be cast as a standard least-squares problem, the main challenge is to devise scalable algorithms that allow each agent to estimate its own position by means of only local communication and bounded complexity, independently of the network size and topology. We propose a gradient based algorithm that is guaranteed to have exponentially convergence rate to the optimal centralized least-square solution. Moreover we show the convergence also in presence of bounded delays and packet losses. We finally provide numerical results to support our work.**

I. Introduction

The recent technological progress in MEMS systems, wireless communications and digital electronic allowed the development of small and inexpensive devices capable of communicating, computing, sensing, interacting with the environment and storing information. These devices are promising an unprecedented number of new applications as swarm robotics, wireless sensor networks, smart energy grid, smart traffic networks and smart camera networks, which pose significant technical challenges, among them, scalability is one of the major. The scalability propriety, for a network, is intended as the ability to handle a growing number of nodes without requiring to increase the hardware resources and to adapt the software algorithms.

In this work we address the problem of designing a scalable and distributed algorithm that is capable to reconstruct the optimal estimate of the location of a network of devices from relative noisy measurements. In particular by *scalable* we mean that the computational complexity, bandwith, memory requirements should be independent of the network size. By *distributed* we mean that the algorithm must take into account the advantages of sharing its data with other devices but also it has to consider the limited communication capabilities, i.e., a node is allowed to exchange information only with its neighbors regardless the size of the network.

The problem at hand in this work can be casted as the following unconstrained optimization problem

$$\min_{x_1,\ldots,x_N} \sum_{(i,j)\in\mathcal{E}}^{|\mathcal{E}|} \parallel x_i - x_j - z_{ij} \parallel^2 \tag{1}$$

where $x_i, z_{ij} \in \mathbb{R}^\ell$ are the unknown position and the relative noisy measurement, respectively, and $\mathcal{E}$ represents all the pair of nodes for which are available relative measurements.

The solution of this optimization problem becomes a least-square problem. Several distributed solutions are available in the literature. In [1], [2] the authors propose a distributed Jacobi solution which requires a synchronous implementation. Similarly, in [3] the authors propose a coordinate descent strategy which is suitable for asynchronous implementation but requires the updating node to receive all the estimated positions of its neighbors. Differently, in [4] a broadcast consensus-based algorithm is proposed but the local estimates exhibits an oscillatory behavior around the true value. A similar approach has been proposed in [5] where the local ergodic average of the gossip asynchronous algorithm is proved to converge to the optimal value as $1/k$, where k is the number of iterations. An alternative approach based on the Kaczmarz method for the solution of linear systems has been suggested in [6], however the proposed algorithms either oscillate or converge to the optimal value as $1/k$.

The contribution of this work is to provide an asynchronous algorithm which is scalable, robust to delays and have proven exponential convergence rate under mild assumption. The algorithm is based on a standard gradient descent strategy. To compute the gradient each node is required to store in memory only a copy of the estimate of all its neighbors. The proposed algorithm is similar to the algorithm presented in [7], which requires bidirectional communication among nodes; on the contrary our strategy is based on broadcast protocols which require no acknowledge from the neighbors. Thorough numerical simulations our solution is shown to outperform the performance of the other algorithms presented in the literature in terms of speed of convergence to the optimal solutions and in terms of number of packets required to be transmitted.

II. Problem Formulation

The problem we consider in this paper is that of estimating N variables $x_1, \ldots, x_N$ from noisy measurements of the form

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

This work is supported by the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n. 257462 HYCON2 Network of excellence.

R. Carli, A. Carron, L. Schenato and M. Todescato are with the Department of Information Engineering, University of Padova, Via Gradenigo 6/a, 35131 Padova, Italy {carlirug|carronan|schenato|todescat}@dei.unipd.it.

where n_{ij} is zero-mean measurement noise. Though the variables are often vector-valued, for simplicity, in this paper we assume that $x_i \in \mathbb{R}$, $i \in \{1,\ldots,N\}$.

This estimation problem can be naturally associated with a undirected *measurement graph* $\mathcal{G} = (V;\mathcal{E})$ where

(i) V denotes the set of nodes which are labeled 1 through N, being N the number of nodes, i.e., $V = \{1,\ldots,N\}$;

(ii) $\mathcal{E}$ is the edge set and consists of all the pairs of nodes (i,j) such that a noisy measurement of the form (2) between i and j is available to both node i and node j.

In the sequel it is convenient to assume that, if z_{ij} is the measurement available at node i then $z_{ji} = -z_{ij}$ is the measurement available at node j. Basically we are assuming that the measurements are symmetrical, meaning that both agents of a pair know the measurement, with a reverse sign.

Assume that there are M available measurements, i.e., $|\mathcal{E}| = m$ and assume that the measurements errors on distinct edges are uncorrelated.

Next we formally state the problem we aim at solving. To do so we first need some preliminary definitions. Let $\mathbf{x} \in \mathbb{R}^N$ be the vector obtained stacking together all the variables $x_1,\ldots,x_N$, i.e., $\mathbf{x} = [x_1,\ldots,x_N]^T$, where given a vector v with v^T we denote its transpose, and let $\mathbf{z} \in \mathbb{R}^M$ and $\mathbf{n} \in \mathbb{R}^M$ be the vectors obtained stacking together all the measurements z_{ij} and the noises n_{ij}, respectively. Additionally, let $R_{ij} > 0$ denote the covariance of the zero mean error n_{ij}, i..e, $R_{ij} = \mathbb{E}[n_{ij}^2]$, where $\mathbb{E}$ denotes the expectation operator, and let $R \in \mathbb{R}^{M\times M}$ be the diagonal matrix collecting in its diagonal the covariances of the noises n_{ij}, $(i,j) \in \mathcal{E}$, i.e., $R = \mathbb{E}[\mathbf{n}\mathbf{n}^\mathbf{T}]$. Finally let $\mathbf{1}$ be the column vector with all components equal to one.

Now, on each edge, let us choose an orientation, that is, let us define a starting node and an ending node, in order to encode the measurements by using the *incidence matrix* $A \in \mathbb{R}^{M\times N}$ of $\mathcal{G}$ defined as $A = [a_{ei}]$, where $a_{ei} = 1, -1, 0$, if edge e is incident on node i and directed away from i, is incident on node i and directed toward it, or is not incident on node i, respectively. Observe that equation (2) can be rewritten in a vector form as

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

Consider the function $J : \mathbb{R}^{N+M} \to \mathbb{R}$, defined as

$$J(\mathbf{x},\mathbf{z}) = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} \frac{(x_i - x_j - z_{ij})^2}{R_{ij}}.$$

Observe that

$$J(\mathbf{x},\mathbf{z}) = \frac{1}{2}(\mathbf{z} - A\mathbf{x})^T R^{-1}(\mathbf{z} - A\mathbf{x}).$$

Define the set

$$\chi := \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{argmin}}\ J(\mathbf{x},\mathbf{z}).$$

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

$$\mathbf{x}^* \in \chi \tag{3}$$

Assume the measurement graph $\mathcal{G}$ to be *connected*, then it is well known that

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

Moreover let

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

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

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

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

III. An asynchronous gradient-based localization algorithm

To compute an optimal estimate $\mathbf{x}^*$ directly, one needs all the measurements and their covariances and the topology of the measurement graph $\mathcal{G}$. In this section the goal is to compute the optimal solution in a distributed fashion, employing only local communication. In particular we assume that a node i and another node j can communicate with each other only if $(i,j) \in \mathcal{E}$. Accordingly a node i is said to be a neighbor of another node j (and viceversa) if $(i,j) \in \mathcal{E}$. For $i \in \{1,\ldots,N\}$, by $\mathcal{N}_i$ we denote the set of neighbors of node i, namely,

$$\mathcal{N}_i = \{j \in V \text{ such that } (i,j) \in \mathcal{E}\}.$$

In this paper we are interested into solutions with the following two features:

(i) They are *distributed* as opposed to centralized solutions, namely, there is no a central unit gathering all the measurements z_{ij}, having global knowledge of the graph $\mathcal{G}$ and computing $\mathbf{x}^*$ directly; instead each node has at its disposal computational and memory resources and is allowed to communicate only with its neighbors in the graph $\mathcal{G}$.

(ii) They are *asynchronous*, as opposed to synchronous solutions, namely, there is no a common reference time which keeps all the updating and transmitting actions synchronized among all the nodes.

In what follows we introduce a distributed algorithm which is based on a standard gradient descent strategy and which employs an *asynchronous broadcast* communication protocol; specifically during each iteration of the algorithm there is only node which transmits information to all its neighbors in the graph $\mathcal{G}$. We refer to this algorithm as the *asynchronous gradient-based localization* algorithm (denoted hereafter as a-GL algorithm).

We assume that every node has access to the measurements on the edges that are incident to it, as well as the associated covariances. Additionally we assume that node i, $i \in V$, stores in memory an estimate $\hat{x}_i$ of x_i and, for $j \in \mathcal{N}_i$, an estimate $\hat{x}_j^{(i)}$ of x_j.

Next we formally describes the a-GL algorithm. Let $t_0, t_1, t_2, \ldots$ be the time instants in which the iterations of the a-GL algorithm occur. Assume at time t_h node i is activated. The following actions are performed in order.

(i) Node i updates its estimate $\hat{x}_i$ in the following way

$$\hat{x}_i \leftarrow \hat{x}_i - \alpha_i \sum_{j \in \mathcal{N}_i} \frac{\hat{x}_i - \hat{x}_j^{(i)} - z_{ij}}{R_{ij}}$$

where α_i is a suitable positive real number;

(ii) Node i broadcasts the updated value of the estimate $\hat{x}_i$ to all its neighbors j, $j \in \mathcal{N}_i$;

(iii) Node j, $j \in \mathcal{N}_i$, updates the estimate $\hat{x}_i^{(j)}$ setting it equal to the value $\hat{x}_i$ it has received from node i, i.e.,

$$\hat{x}_i^{(j)} \leftarrow \hat{x}_i$$

Some explanations are now in order. Observe that the quantity $\sum_{j \in \mathcal{N}_i} \left(\hat{x}_i - \hat{x}_j^{(i)} - z_{ij} \right) / R_{ij}$ represents the gradient of the function

$$J_i = \frac{1}{2} \sum_{j \in N_i} \frac{\left(\hat{x}_i - \hat{x}_j^{(i)} - z_{ij} \right)^2}{R_{ij}}.$$

Basically, node i updates the value of $\hat{x}_i$ moving along a descent direction of the function J_i. Notice that J_i does not increase if

$$0 < \alpha_i \leq \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1}$$

and, in particular, if $\alpha_i = \left(\sum_{j \in \mathcal{N}_i} R_{ij} \right)^{-1}$ then the minimum of J_i is attained. Indeed in this case we have that

$$\hat{x}_i \leftarrow \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1} \left(\sum_{j \in \mathcal{N}_i} \frac{\hat{x}_j^{(i)} + z_{ij}}{R_{ij}} \right)$$

which corresponds to the unique solution of the problem

$$\underset{\hat{x}_i}{\operatorname{argmin}} \frac{1}{2} \sum_{j \in N_i} \frac{\left(\hat{x}_i - \hat{x}_j^{(i)} - z_{ij} \right)^2}{R_{ij}}.$$

Next we provide a convenient vector form description of the a-GL algorithm. To do so, we introduce the following definitions. Let $\hat{x}_i(h)$ and $\hat{x}_i^{(j)}(h)$, $j \in \mathcal{N}_i$, denote the estimates that node i has of x_i and of x_j, $j \in \mathcal{N}_i$, respectively, just before time instant t_h. Since we are assuming that there are no communication delays and packet losses, it follows that $\hat{x}_i^{(j)}(h) = \hat{x}_j(h)$, $j \in \mathcal{N}_i$. Then

$$\hat{x}_i(h+1) = \left(1 - \alpha_i \sum_{j \in \mathcal{N}_i} R_{ij} \right) \hat{x}_i(h) + \alpha_i \sum_{j \in \mathcal{N}_i} R_{ij} \left(\hat{x}_j(h) + z_{ij} \right)$$

while $\hat{x}_k(h+1) = \hat{x}_k(h)$, $k \neq i$. Let us rewrite the above equation as

$$\hat{x}_i(h+1) = p_{ii} \hat{x}_i(h) + \sum_{j \in \mathcal{N}_i} p_{ij} \hat{x}_j(h) + u_i \qquad (4)$$

where

$$p_{ij} = \begin{cases} 1 - \alpha_i \sum_{j \in \mathcal{N}_i} R_{ij} & \text{if } j = i \\ \alpha_i R_{ij} & \text{if } j \neq i, j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}$$

and where

$$u_i = \alpha_i \sum_{j \in \mathcal{N}_i} R_{ij} z_{ij}.$$

Let $P \in \mathbb{R}^{N \times N}$ be the matrix defined by the weights p_{ij} above defined. Then the updating step at time t_h can be written in vector form as

$$\hat{\mathbf{x}}(h+1) = \left(I + e_i e_i^T (P - I) \right) \hat{\mathbf{x}}(h) + U_i \qquad (5)$$

where the vector $U_i \in \mathbb{R}^N$ is defined as $U_i = u_i e_i$. Let

$$Q_i = I + e_i e_i^T (P - I),$$

and observe that, if

$$0 < \alpha_i \leq \left(\sum_{j \in \mathcal{N}_i} \frac{1}{R_{ij}} \right)^{-1}, \qquad \forall i \in V$$

then the matrix Q_i is a stochastic matrix for all $i \in V$. Indeed all the elements of Q_i are nonnegative and it is easy to see that $Q_i \mathbf{1} = \mathbf{1}$.

Now let us introduce the auxiliary variable

$$\xi(h) = \hat{\mathbf{x}}(h) - \hat{\mathbf{x}}^*_{\text{opt}}$$

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

$$\mathbf{x}^*_{\text{opt}} = Q_i \mathbf{x}^*_{\text{opt}} + U_i \qquad (6)$$

we have that the variable ξ satisfies the following recursive equation

$$\xi(h+1) = Q_i \xi(h). \qquad (7)$$

Observe that $\hat{\mathbf{x}}(h) \to \mathbf{x}^*_{\text{opt}} + \gamma \mathbf{1}$ if and only if $\xi(h) \to \gamma \mathbf{1}$. Moreover, since Q_i is a stochastic matrix for any $i \in \{1, \ldots, N\}$, we have that (7) represents a N-dimensional time-varying consensus algorithm.

Remark III.1 It is worth to highlight that our algorithm has been inspired by the algorithm proposed in [7]. The main differences are related to the communication protocol. Specifically, in [7] when a node is activated, say i, firstly it interrogates its neighbors to obtain their estimates $\{\hat{x}_j\}_{j \in \mathcal{N}_i}$, secondly, based on the information received, it updates its own estimate $\hat{x}_i$. This implies that during this iteration of the algorithm there are $|\mathcal{N}_i|+1$ transmitted packets (one packet is related to the broadcast request by node i while the other $|\mathcal{N}_i|$ packets are related to $\{\hat{x}_j\}_{j \in \mathcal{N}_i}$ responses). Instead in the a-GL algorithm, there is just one packet broadcasted during each iteration of the algorithm. This leads to a lighter, faster and energy-saving solution. Additionally in [7] there is no robustness analysis against packet losses.

In next sections, we analyze the convergence properties and the robustness to delays and packet losses of the a-GL algorithm by studying system (7) resorting to the mathematical tools developed in the literature of the consensus algorithms. In particular we will provide our results considering two different scenarios which are formally described in the following definitions.

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

$$\mathbb{P}\left[\mathcal{A}_{i,h}\right] = \beta_i,$$

where $\mathcal{A}_{i,h}$ is the event

$$\mathcal{A}_{i,h} = \{\textit{the node performing steps 1) and 2) of the a-GL algorithm at iteration } h \textit{ is node } i\}$$

Definition III.3 (Uniformly persistent comm. network) *A network of N nodes is said to be a* uniformly persistent communicating network *if there exists a positive integer number τ such that, for all $h \in \mathbb{N}$, each node perform steps 1) and 2) of the a-GL algorithm at least once within the iteration-interval $[h, h+\tau)$.*

IV. Convergence analysis in the randomly persistent communicating scenario

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

Proposition IV.1 *Consider a* randomly persistent communicating network *of N nodes running the a-GL algorithm over a connected measurement graph $\mathcal{G}$. Assume the weights α_i are such that*

$$0 < \alpha_i \leq \left(\sum_{j\in\mathcal{N}_i} \frac{1}{R_{ij}}\right)^{-1}, \qquad \forall\, i \in V,$$

and assume that $\hat{x}_i$, $i \in \{1, \ldots, N\}$, $\hat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true

(i) *the evolution $h \to \hat{\mathbf{x}}(h)$ converges almost surely to an optimal solution $\mathbf{x}_{opt} \in \chi$, i.e., there exists $\gamma \in \mathbb{R}$ such that*

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

(ii) *the evolution $h \to \hat{\mathbf{x}}(h)$ is exponentially convergent in mean-square sense, i.e., there exist $C > 0$ and $0 \leq \rho < 1$ such that*

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

The proofs of the above proposition and of the following one are reported in the technical note in [8].

V. Robustness to packet losses and delays in the uniformly persistent communicating scenario

In section III we have introduced the a-GL algorithm under the assumptions that

- the communication channels are reliable, i.e., no packet losses occur; and
- the transmission delays are negligible.

In this section we consider a more realistic scenario where the above two assumptions are relaxed. We are still able to prove that the a-GL algorithm converges to an optimal solution provided that the network is uniformly persistent communicating and the transmission delays and the frequencies of communication failures satisfy mild conditions which we formally describe next.

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

Assumption V.2 (Bounded delay) *Assume node i broadcasts its estimate to its neighbors during iteration h, and, assume that, the communication link (i, j) does not fail. Then, there exists a positive integer D such that the information $\hat{x}_i(h+1)$ is used by node j to perform its local update not later than iteration $h + D$.*

Loosely speaking Assumption V.1 implies that there can be no more than L consecutive packet losses between any pair of nodes i, j belonging to the communication graph. Differently, Assumption V.2 consider the scenario where the received packets are not used instantaneously, but are subject to some delay no greater than D iterations.
This implies that in general $\hat{x}_i^{(j)}(h) = \hat{x}_j(h'_{ij})$ for some h'_{ij} such that $h - (\tau L + D) \leq h'_{ij} \leq h$. It turns out that the equation update (4) is, in general, modified as

$$\hat{x}_i(h+1) = p_{ii}\hat{x}_i(h) + \sum_{j\in\mathcal{N}_i} p_{ij}\hat{x}_j(h'_{ij}) + u_i.$$

The following proposition characterizes the convergence proprieties in presence of bounded packet losses and bounded delay.

Proposition V.3 *Consider a* uniformly persistent communicating network *of N nodes running the a-GL algorithm over a connected measurement graph $\mathcal{G}$. Let Assumptions V.1 and V.2 be satisfied. Assume the weights α_i are such that*

$$0 < \alpha_i < \left(\sum_{j\in\mathcal{N}_i} \frac{1}{R_{ij}}\right)^{-1}, \qquad \forall\, i \in V,$$

and assume that $\hat{x}_i$, $i \in \{1,\dots,N\}$, $\hat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true

(i) *the evolution $h \to \hat{\mathbf{x}}(h)$ asymptotically converges to an optimal estimate $\mathbf{x}_{opt} \in \chi$, i.e., there exists $\gamma \in \mathbb{R}$ such that*

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

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

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

Observe that in Proposition V.3 it is assumed that α_i is strictly smaller than $\left(\sum_{j\in\mathcal{N}_i} \frac{1}{R_{ij}}\right)^{-1}$, while the result in Proposition IV.1 holds true also if the equality is satisfied. Next we provide a example showing that, if $\alpha_i = \left(\sum_{j\in\mathcal{N}_i} \frac{1}{R_{ij}}\right)^{-1}$, then the optimal solution is not reached in presence of constant positive delays.

Example V.4 Consider a network with $N = 2$ agents and the following cost function $f = \frac{1}{2R}\left(\hat{x}_1 - \hat{x}_2 - z\right)^2$, where z is the noisy measurement. Setting $\alpha = R$ we have the following two update rules:

$$\begin{cases} \hat{x}_1(h) &= \hat{x}_2^{(1)}(h) + z \\ \hat{x}_2(h) &= \hat{x}_1^{(2)}(h) - z \end{cases} \tag{8}$$

where

$$\begin{cases} \hat{x}_1^{(2)}(h) &= \hat{x}_1(h-1) \\ \hat{x}_2^{(1)}(h) &= \hat{x}_2(h-1) \end{cases}$$

which means that the state of the other agent is known with one step delay. During the odd iterations node 1 makes the update, while during even iterations node 2 updates. Starting from initial conditions equal to zero and assuming that at the first iteration the agent 1 consider the state of the agent 2 equal to zero, following the update rule (8) we get

h	1	2	3	4	5	...
$\hat{x}_1$	0	z	z	z	z	...
$\hat{x}_2$	0	0	z	z	0	...

It can be seen from the above table that there is a three steps oscillatory behavior. The value of the function f keeps jumping from 0 to $\frac{z^2}{2R}$ so there is not convergence to the optimal set of solutions.

VI. Numerical Results

In this section we provide some simulations implementing and comparing three different algorithms.

Example VI.1 In the example we consider a random geometric graph generated with $N = 50$ nodes randomly placed in the interval $[0, 3]$. Two nodes can be considered connected, and consequently they can share their estimates, if they are sufficiently close. More specifically, in our scenario, two nodes are connected if $|x_i - x_j| \le 0.5$. With this choice the average number of neighbors per node results to be of about 10.

Every measurement was corrupted by Gaussian noise with variance $\sigma^2 = 10^{-2}$. In this example we assumed that the network was randomly persistent communicating with the following probabilities to select a node or an edge (when it was required):

$$\beta_k = \frac{|\mathcal{N}_k| + 1}{2M}; \quad \beta_{ij} = \frac{1}{M}$$

With these kind of communication probabilities the Randomized Extended Kaczmarz Algorithm, hereafter called **REK algorithm**, get the best performances, so to provide a fair comparison we did not choose the uniform communication probabilities.

The first algorithm that we considered is the REK presented in [9], consisting of two different update steps. The first step is an orthogonal projection of the noisy measurements onto the column space of the incidence matrix A in order to bound the measurements error. The second step is similar to the standard Kaczmarz update. Since a distributed implementation is not formally presented in [9], we propose the following. More specifically, let $s \in \mathbb{R}^M$ be the current projection of the noisy measurements onto the column space of A. Similarly as above, we denote with a little abuse of notation the e-th entry of s with the corresponding edge, i.e. $s_e = s_{ij}$. Then, the REK algorithm proposed in [9] for general least-squares problems, performs the following local updates:

$$s_{\ell k}(h+1) = s_{\ell k}(h) + \frac{\sum_{m\in\mathcal{N}_k}(s_{km}(h) - s_{mk}(h))}{|\mathcal{N}_k|+1}, \quad \forall \ell \in \mathcal{N}_k$$

$$s_{k\ell}(h+1) = s_{k\ell}(h) - \frac{\sum_{m\in\mathcal{N}_k}(s_{km}(h) - s_{mk}(h))}{|\mathcal{N}_k|+1}, \quad \forall \ell \in \mathcal{N}_k$$

$$\hat{x}_i(h+1) = \hat{x}_i(h) + \frac{z_{ij} - s_{ij}(h) - (\hat{x}_i(h) - \hat{x}_j(h))}{2}$$

$$\hat{x}_j(h+1) = \hat{x}_j(h) - \frac{z_{ij} - s_{ij}(h) - (\hat{x}_i(h) - \hat{x}_j(h))}{2}$$

We point out that, since in the updating step only local information is required, the algorithm is implemented in a distributed fashion and it exactly requires $|N_j| + 5$ communication rounds to perform an iteration. Specifically the first $|N_j| + 2$ are due to the update of the variable s and the last 3 are needed to update $\hat{x}$.

The second algorithm, hereafter called a **a-CL algorithm**, is proposed in [10]. Since the actual value of neighboring estimates are not available at each iteration, we assume that each node stores in its local memory a copy of the neighbors' variables recorded from the last communication received. For $j \in \mathcal{N}_i$, we denote by $\hat{x}_j^{(i)}(h)$ the estimate of x_j kept in i's local memory at the end of the h-th iteration. During the $h-th$ iteration a node, say i, broadcast its estimate to all its neighbors $j \in \mathcal{N}_i$, so node j performs the following actions in order

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

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

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

where p_{jj}, p_{jk} are the elements of the matrix defined as $P = I - \epsilon A^T R^{-1} A$ and b_j is the j-th component of the vector $\epsilon A^T R^{-1}$. The parameter ϵ is chosen such that $0 < \epsilon < 1/(2d_{max}R_{min}^{-1})$, where $R_{min} = \min\{R_{ij}; (i,j) \in \mathcal{E}\}$ and $d_{max} = \max\{|\mathcal{N}_i|, i \in V\}$.

Note that just one packet is transmitted at each iteration of the a-CL algorithm. The following table summarizes the number of packets transmitted during each iteration of the algorithms we are considering.

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

Number of sent packets per iteration for each algorithm.

In Figure 1 we plotted the behavior of the error

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

Observe that the trajectory of J decreases exponentially. From the simulation we observe that the a-GL algorithm,

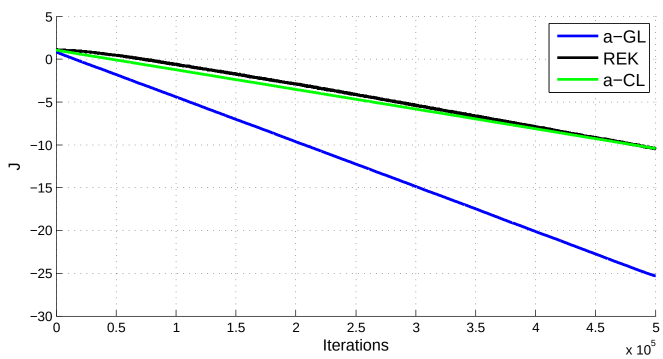

Fig. 1: Comparison of various algorithm in a random geometric graph.

together with the a-CL, is the most convenient from the energy point of view. Moreover the a-GL is also the fastest algorithm.

Example VI.2 In this example we use the same framework of the previous example but we consider a circular graph. We can see that the results are the same of the previous example.

Example VI.3 In this example we assume the same framework of the example VI.1 with the difference that here we are verifying the capability of the a-GL algorithm to converge also if the packets received are delayed. However from the Figure 3 we can see that the algorithm still converge to the optimal solution but with a slower convergence rate.

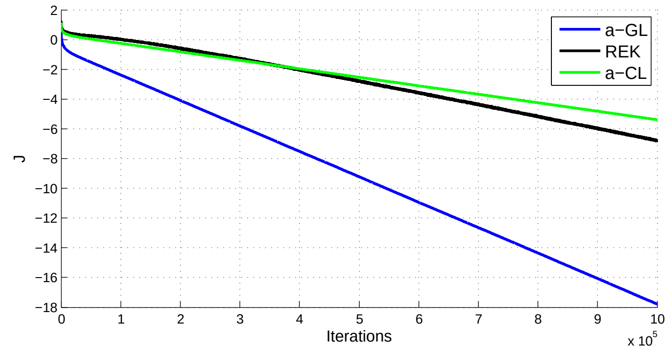

Fig. 2: Comparison of various algorithm in a circular graph.

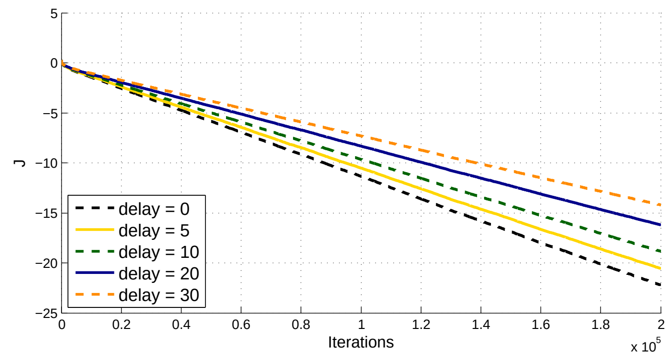

Fig. 3: Comparison of various algorithm in a circular graph.

Example VI.4 In this example we compare the performances of the algorithm proposed [7], hereafter called BC, and the a-GL algorithm. We consider a random geometric graph with $N = 20$ nodes. In Figure 4 we plot the behavior of J respect to the number of sent packets. As we can see the a-GL is much more faster then the BC, so can be considered the most convenient from an energy point of view.

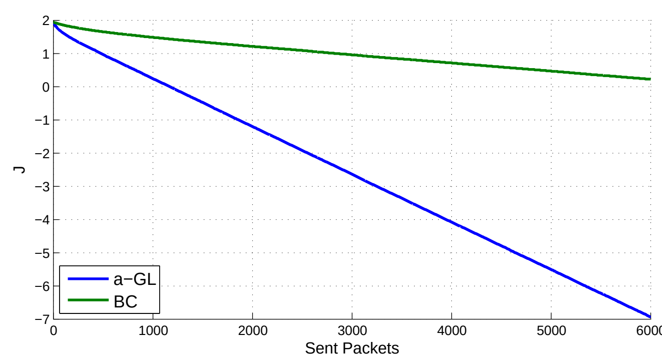

Fig. 4: Comparison of a-GL and BC w.r.t. the number of sent packets.

VII. CONCLUSIONS

In this paper we consider the problem of optimally estimating the position of each agent in a network from relative noisy distances. After having formulated the problem in a least-square framework, we proposed a revisited and more efficient version of the algorithm presented in [7]. We proved that the trajectories generated by the algorithm converges to the optimal solution exponentially and that the algorithm is robust against packet losses and bounded delays.

REFERENCES

[1] P. Barooah and J. P. Hespanha, “Distributed estimation from relative measurements in sensor networks,” in *Proceedings of the 2nd International Conference on Intelligent Sensing and Information Processing*, Dec. 2005.

[2] P. Barooah, “Estimation and control with relative measurements: Algorithms and scaling laws,” Ph.D. dissertation, University of California, Santa Barbara, 2007.

[3] A. Giridhar and P. R. Kumar, “Towards a theory of in-network computation in wireless sensor networks,” *IEEE Communications Magazine*, vol. 44, no. 4, pp. 98–107, April 2006.

[4] S. Bolognani, S. D. Favero, L. Schenato, and D. Varagnolo, “Consensus-based distributed sensor calibration and least-square parameter identification in wsns,” *International Journal of Robust and Nonlinear Control*, vol. 20, no. 2, pp. 176–193, 2010.

[5] C. Ravazzi, P. Frasca, H. Ishii, and R. Tempo, “A distributed randomized algorithm for relative localization in sensor networks,” in *Proceedings of European Conference on Control (ECC’13)*, 2013.

[6] N. M. Freris and A. Zouzias, “Fast distributed smoothing of relative measurements,” in *to appear in IEEE Conference on Decision and Control (CDC 2012)*, 2012.

[7] A. Giridhar and P. Kumar, “Distributed clock synchronization over wireless networks: Algorithms and analysis,” in *Decision and Control, 2006 45th IEEE Conference on*, Dec 2006, pp. 4915–4920.

[8] R. C. A. Carron, M. Todescato and L. Schenato, “Distributed localization from relative noisy measurements: a packet losses and delays robust approach,” 2013, [Online] Available at http://automatica.dei.unipd.it/people/carron/publications.html.

[9] A. Zouzias and N. Freris, “Randomized extended Kaczmarz for solving least-squares,” University of Toronto, Tech. Rep., 2013.

[10] A. Carron, M. Todescato, R. Carli, and L. Schenato, “Adaptive consensus-based algorithms for fast estimation from relative measurements,” in *4th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys’13)*, March 2013.