Robust Distributed Estimation for Localization in Lossy Sensor Networks

N. Bof [*] **M. Todescato** [*] **R. Carli** [*] **L. Schenato** [*]

[*] *Department of Information Engineering, Padova, 35131, Italy.*
(e-mail: bofnicol|todescat|carlirug|schenato @ dei.unipd.it).

Abstract: In this paper we address the problem of fault resilient estimation for large-scale systems, where the measurements are possibly corrupted due to faults of low-cost sensors. As a toy application, we consider the problem of localization in Sensor Networks (SN). We propose a distributed solution based on a recently developed generalized descent algorithm. To cope with real-world applications, the algorithm we propose is suitable for an asynchronous implementation and is numerically robust to non ideal communications, i.e., packet-losses. Under mild assumptions, theoretical convergence of the algorithm is shown. The algorithm is compared with a recently developed ADMM-based algorithm for robust state estimation.

Keywords: Multi-agent systems, Robust Estimation, Robust Statistics, Asynchronous Broadcast, Distributed Convex Optimization, Sensor Networks, WSN

1. INTRODUCTION

Nowadays, large-scale and distributed cyber-physical systems, consisting of a multitude of sensors and "smart agents" equipped with mild computational, communication and actuation capabilities, permeate our lives. Because of the size of the systems, low-cost sensors are typically used. However the latter are more prone to random failures, and consequently, one important challenge to face is the systematic quantitative monitoring of the system. Indeed, by affecting the collected measurements, these failures eventually compromise the knowledge of the system' state, usually used for management and control. In order to avoid this issue, two strategies can be followed: (i) the development of suitable fault diagnosis algorithms (see Paradis and Han (2007) for a survey on the topic), consisting in detection, isolation and identification of the fault; (ii) the design of fault resilient state estimation procedures which are able to produce accurate outcomes by automatically filtering out the outliers. These two approaches, which may eventually complement each other, become necessary to implement reliable systems. However, the possibly large scale of these systems makes central monitoring strategies difficult and sometimes impossible to implement. Thus, distributed solutions must be addressed.

Fault detection and bad data analysis have been largely studied in the past. A lot of work has been done on the static analysis of faults. The main idea behind static analysis is to process the measurements residuals through suitable hypothesis tests in order to detect the source of the fault. In Chen et al. (2006) a distributed belief propagation approach is proposed for WSN. With specific applications to electrical power systems, in Korres (2011) a distributed bad data analysis and detection procedure is shown, which is based on the normalized residual test. Choi and Xie (2011) propose a reduced model for distributed wide area monitoring and a bad data analysis based on the χ^2-test. A more recent branch of research regards the development of fault diagnosis strategies for general networks of dynamical systems using sensors networks. In Franco et al. (2006) a distributed hypothesis testing method, based on a belief consensus technique to perform fault diagnosis, is presented. Consensus is exploited in Boem et al. (2011) as well, where the authors propose a distributed strategy which is based on the combination of local fault estimators to reach a common agreement on the fault detection. More recently, Boem et al. (2013) propose a method based on Pareto optimization. Finally, in Keliris et al. (2015) the authors present a distributed scheme for the detection of process and sensors faults for a certain class of nonlinear discrete-time systems.

Regarding distributed state estimation, a vast amount of literature can be found. However, historically, state estimation does not deal with the presence of outliers. In order to deal with bad data analysis, the standard approach consists of two iterative steps: first, state estimation is performed; second, hypothesis tests on the measurements residuals are applied as done in Korres (2011); Choi and Xie (2011). If a bad datum is detected, this is deleted from the data-set and state estimation is performed again. Hypothesis test on the new residuals can confirm or belie the detection. In this sense, this approach iteratively combines standard state estimation with static fault detection procedures, to eventually lead to a fault resilient state estimator.
A different approach is followed in Kekatos and Giannakis (2013), where the authors propose an iterative distributed strategy based on the classical ADMM algorithm to simultaneously solve the state estimation and the fault localization in power systems.

In this work we are interested in developing a fault resilient state estimator rather than a fault detection scheme. Conversely to what is done in Kekatos and Giannakis (2013), where the problem is solved using a least square approach with the introduction of an additional variable to take into account the presence of outliers, we exploit ideas coming from robust statistical analysis (Bloomfield and Steiger, 2012; Huber, 2011) to formulate a suitable convex problem. In particular, the choice of a "$1-$norm"-based cost function, let us automatically filter out potential outliers in the measurements caused by sensors faults. Inspired by the recent result in Todescato et al. (2015), we provide a distributed algorithm to solve the problem. Starting from a synchronous algorithm which assumes perfect and ideal communications among sensor nodes, we modify it to deal with communication non idealities. This is an important aspect since, in real-world large-scale systems, ideal synchronous communications are not likely. The algorithm we propose is based on an asynchronous broadcast communication protocol. Numerically, the algorithm is shown to be robust to communication non-idealities. Under additional mild assumptions on the type of communication non-idealities and on the curvature of our prescribed cost function, convergence of the algorithm is theoretically proven.

We apply the proposed algorithm in the framework of sensors networks localization, even if the strategy applies to a more general setup. Because of the well known performance of the ADMM algorithm, we decide to compare the algorithm with the strategy recently proposed in Kekatos and Giannakis (2013). Since neither asynchronous nor robust implementation of the algorithm in Kekatos and Giannakis (2013) is provided, we suggest one. As shown by the numerical simulations, compared to the ADMM, our robust algorithm has the following features: (i) comparable steady state estimation accuracy; (ii) in scenarios of highly connected graphs, the algorithm is characterized by a faster behavior for both the asymptotic and the transient convergence rate; (iii) in general, conversely to the ADMM, the transient evolution of our algorithm is monotonically decreasing.

1.1 Mathematical Preliminaries

In this paper, $\mathcal{G}\,(\mathcal{V},\mathcal{E})$ denotes a directed graph, where $\mathcal{V} = \{1,\dots,N\}$ is the set of vertices and $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ is the set of directed edges. The graph $\mathcal{G}$ is said to be bidirected if $(i,j)\in\mathcal{E}$ implies $(j,i)\in\mathcal{E}$. An undirected path in $\mathcal{G}$ consists of a sequence of vertices $(i_1, i_2, \dots, i_r)$ such that either $(i_j, i_{j+1})\in\mathcal{E}$ or $(i_{j+1}, i_j)\in\mathcal{E}$ for every $j\in\{1,\dots,r-1\}$. The bidirected graph $\mathcal{G}$ is said to be *connected* if for any pair of vertices (i,j) there exists a undirected path connecting i to j. Given the directed graph $\mathcal{G}$, the set of neighbors of node i, denoted by $\mathcal{N}_i$, is given by $\mathcal{N}_i = \{j\in\mathcal{V}\,|\,(i,j)\in\mathcal{E}\}$. Moreover, $\mathcal{N}_i^+ = \mathcal{N}_i \cup \{i\}$. Given a directed graph $\mathcal{G}\,(\mathcal{V},\mathcal{E})$ with $|\mathcal{E}| = M$, let the *incidence matrix* $\mathcal{A}\in\mathbb{R}^{M\times N}$ of $\mathcal{G}$ be defined as $\mathcal{A} = [a_{ei}]$, where $a_{ei} = 1, -1, 0$, if edge e is incident on node i and directed away from it, is incident on node i and directed toward it, or is not incident on node i, respectively. Given a vector v, with v^T we denote its transpose and with $diag(v)$ the diagonal matrix where the i-th diagonal element corresponds to the i-th element of the vector v. The symbol I denotes the identity matrix of suitable dimension.

2. PROBLEM FORMULATION

In the following, we consider a *localization-type* problem in Sensor Networks (Mao et al., 2007) where, starting from a set of noisy measurements, the agents' goal is to estimate their absolute positions. We want to develop a distributed strategy where the agents are allowed to exchange information locally, i.e., between neighbors. Moreover, for real-world applications, the algorithm must be robust to communication non idealities, e.g., packet dropouts, while being resilient to faulty measurements due to possible sensors failures.
Consider a set of N agents/sensors $\mathcal{V} = \{1,\dots,N\}$, where each agent is described by a state vector $x_i\in\mathbb{R}^{n_i}$. For our purpose and for ease of notation, we restrict the analysis to the scalar case where $n_i = 1$, $\forall i\in\mathcal{V}$.By exploiting graph theoretical tools, we model the SN by means of a bidirected connected measurement graph $\mathcal{G}(\mathcal{V},\mathcal{E})$.
In the following we introduce the measurement model used and we formally state the problem at hand.

2.1 Measurement Model & Fault Resilient Estimation

Assume that each agent collects a certain number of measurements according to the measurement graph $\mathcal{G}$. More specifically, only two types of measurements can be collected. The first are noisy relative distance measurements with respect to neighboring agents, that is, for each $i\in\mathcal{V}$ and $j\in\mathcal{N}_i$, node i measures

$$b_{ij} = x_i - x_j + w_{ij}\,, \quad w_{ij}\sim\mathcal{N}(0,\sigma_{ij}^2)\ .$$

where σ_{ij}^2 denotes the relative measurement noise variance. The second type of measurements is a noisy absolute measurement of the form

$$b_i = x_i + w_i\,, \quad w_i\sim\mathcal{N}(0,\sigma_i^2)\ .$$

where σ_i^2 is the absolute measurement noise variance. By collecting all the state variables in the vector $\mathbf{x} := [x_1,\dots,x_N]^T$ and by defining the measurement matrix H and the vectors of measurements, $\mathbf{b}$, and noises, $\mathbf{w}$, respectively as

$$H := \begin{bmatrix} I \\ \mathcal{A} \end{bmatrix}, \quad \mathbf{b} := \begin{bmatrix} \{b_i\}_{i\in\mathcal{V}} \\ \{b_{ij}\}_{(i,j)\in\mathcal{E}} \end{bmatrix}, \quad \mathbf{w} := \begin{bmatrix} \{w_i\}_{i\in\mathcal{V}} \\ \{w_{ij}\}_{(i,j)\in\mathcal{E}} \end{bmatrix},$$

the overall measurement model[1] can be rewritten in compact form as

$$\mathbf{b} = H\mathbf{x} + \mathbf{w}\,, \qquad \mathbf{w}\sim\mathcal{N}(0,R)\,, \tag{1}$$

where $R := diag(\{\sigma_i^2\}_{i\in\mathcal{V}}\,,\,\{\sigma_{ij}^2\}_{(i,j)\in\mathcal{E}})$ denotes the noise variance matrix.
In presence of outliers, however, some of the measurements may be corrupted by an extra term, which has a probability distribution that highly differs from that of the expected gaussian noise. By collecting these outliers in the sparse vector $\mathbf{o}$, the measurement model (1) becomes

$$\mathbf{b} = H\mathbf{x} + \mathbf{w} + \mathbf{o}. \tag{2}$$

Remark 1. (Measurement model). We underline that the more general case of multidimensional positions can be easily derived assuming independent measurements along each dimension. Moreover, all the following analysis seamlessly applies to more general measurement model in which the measurements are linear combinations of the states of neighboring nodes. For instance, this is the case for the state estimation in smart electric grids.

As above mentioned, we are willing to design a distributed state estimation procedure which is fault resilient, that is which is able to produce a reliable estimation by automatically filtering out the outliers. Conversely to classical least squares estimation, where the objective is to minimize the weighted squared norm of the residuals, here we follow an approach which is inspired from robust statistical analysis (Bloomfield and Steiger, 2012; Huber, 2011), i.e., *least absolute estimation*. The main idea is to make use of suitable convex costs which, differently to the classical quadratic costs, are locally quadratic only around the origin while they become linear away from it. Thanks to this, small residuals are weighted quadratically as in the classical least squares. On the contrary, big residuals, which usually identify the presence of sensors faults, are weighted linearly. Consequently, the estimator weights and "trusts" more the measurements corresponding to small

[1] We underline the fact that we do not require all the nodes to collect absolute positioning measurements. However, for absolute positioning we require that at least one agent measures it. Conversely, only relative localization is performed.

residuals. Precisely, we consider a modified 1-norm defined as (Argaez et al., 2011)

$$\|\cdot\|_{1,\epsilon}\ : \mathbb{R}^n \mapsto \mathbb{R}\ : x \mapsto \|x\|_{1,\epsilon} := \sum_{i=1}^{n} \sqrt{x_i^2 + \epsilon}\,, \tag{3}$$

where the parameter ϵ is used to tune the point where the function changes its behavior from quadratic to linear. Observe that this modified 1-norm is differentiable and thus suitable for gradient based approaches.

By making use of the modified 1-norm (3), we are now ready to formulate our problem of interest. This reads as

$$\mathcal{P}_1\ :\ \min_{\mathbf{x}\in\mathbb{R}^N} \underbrace{\|(\mathbf{b} - H\mathbf{x})\|_{1,\epsilon}}_{J(\mathbf{x})}\,. \tag{4}$$

Observe that efficient convex solvers might be used to solve $\mathcal{P}_1$. However, these require global knowledge of the network model as well as of the measurements.

In the next section, before presenting the proposed algorithm, we briefly recall the particular communication architecture exploited.

2.2 Partition-based Communication Architecture

We assume the network is partitioned into p non-overlapping areas A_h, where $h \in \mathcal{V}_c := \{1, \dots, p\}$. Each area contains a certain number s_h of agents $i \in \mathcal{V}$ and is monitored by a local master node which has complete knowledge of the intra-area communications and measurement model. This node collects all the intra-area states, can collect all the intra-area measurements, can process the data, and can communicate with the master nodes in charge to monitor the neighboring areas. According to this partitioning, it is possible to define a new bidirected communication graph $\mathcal{G}_c(\mathcal{V}_c, \mathcal{E}_c)$, where the edge set $\mathcal{E}_c \subseteq \mathcal{V}_c \times \mathcal{V}_c$ consists of the pairs (h, k) for which there exist agent $i \in A_h$ and agent $j \in A_k$ such that $(i, j) \in \mathcal{E}$. Moreover, it is possible to block partition the state, the measurement, the noise vectors as well as the measurement matrix as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_p \end{bmatrix},\ \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_p \end{bmatrix},\ \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_p \end{bmatrix},\ H = \begin{bmatrix} H_{11} & \cdots & H_{1p} \\ & \ddots & \\ H_{p1} & \cdots & H_{pp} \end{bmatrix},$$

where, for $h \in \mathcal{V}_c$, $\mathbf{x}_h \in \mathbb{R}^{s_h}$ is the intra-area state vector, while $\mathbf{b}_h, \mathbf{w}_h \in \mathbb{R}^{m_h}$ ($m_h = \sum_{i\in A_h} |\mathcal{N}_i^+|$) contain the measurements and noises of the agents owned by area A_h. Similarly, the block $H_{hk} \in \mathbb{R}^{m_h \times s_k}$ consists of the elements of the matrix H connecting the agents contained in area A_h with those contained in area A_k. Observe that $H_{hk} \neq 0$ if and only if $(h, k) \in \mathcal{E}_c$. Finally, the noise variance matrix becomes $R = blkdiag(R_1, \dots, R_p)$, $R_h \in \mathbb{R}^{m_h \times m_h}$.

Observe that, according to the partition-based architecture, it is possible to rewrite Problem $\mathcal{P}_1$ as

$$\mathcal{P}_1\ :\ \min_{\mathbf{x}_1,\dots,\mathbf{x}_p} \sum_{h\in\mathcal{V}_c} \underbrace{\|\mathbf{b}_h - \sum_{k\in\mathcal{N}_h^+} H_{hk}\mathbf{x}_k\|_{1,\epsilon}}_{J_h(\mathbf{x}_h, \{\mathbf{x}_k\}_{k\in\mathcal{N}_h})}\,, \tag{5}$$

highlighting the separability structure of the cost function, which is now written as sum of "local" costs.

3. DISTRIBUTED SOLUTIONS

3.1 Synchronous Fault Resilient Estimation

Here, we present a distributed solution for Problem (5) which exploits the partition-based communciation architecture of Section 2.2. This algorithm represents the starting point for an asynchronous robust version which will be presented in the next section. The proposed algorithm, which we refer to as *Distributed Fault Resilient Estimation* algorithm (hereafter denoted as DFRE), is inspired on the *Block Jacobi Algorithm* recently developed in Todescato et al. (2015). The algorithm is essentially a generalized gradient descent of the type

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \rho D^{-1}(t)\nabla J(t)\,,$$

where, at iteration t, $\nabla J(t)$ is the gradient of the cost function evaluated in the current state estimate $\mathbf{x}(t)$, $D(t)$ is a block diagonal matrix which is used to accelerate the convergence of the algorithm and ρ is the gradient step size. In particular, from the cost in (5), we have that the h-th block of the gradient vector, computed by the master node of area A_h is equal to

$$[\nabla J(t)]_h = -\sum_{k\in\mathcal{N}_h^+} \underbrace{H_{kh}^T \left((diag(\mathbf{z}_k(t)))^2 + \epsilon I \right)^{-1/2} \mathbf{z}_k(t)}_{\mathbf{g}_k(t)}\,, \tag{6}$$

where $\mathbf{z}_h(t) := \mathbf{b}_h - \sum_{k\in\mathcal{N}_h^+} H_{hk}\mathbf{x}_k(t)$ represents the vector of current estimation residuals. Observe that to compute $\mathbf{z}_h(t)$, the master node h has to receive the states $\mathbf{x}_k$ from nodes $k \in \mathcal{N}_h$, and then to compute (6) it needs to receive the vector $\mathbf{g}_k(t)$ from $k \in \mathcal{N}_h$. Regarding the $D(t)$ matrix, as done in Todescato et al. (2015), the idea is to use the second order information of the cost. In particular, we set the h-th diagonal block of $D(t)$ equal to the h-th diagonal block of the cost Hessian. In the aforementioned paper, the adoption of a quadratic cost function led to a constant second derivative term, that is $D(t) = \bar{D}$, $\forall t \geq 0$. However, in our case, $D(t)$ changes over time and is state dependent. Precisely, the h-th block of $D(t)$ is equal to

$$D_h(t) = \epsilon H_{hh}^T \left((diag(\mathbf{z}_h(t)))^2 + \epsilon I \right)^{-3/2} H_{hh}\,, \tag{7}$$

and can be computed by the master node h using only intra-area information. Finally, each master node can update its current estimate as

$$\mathbf{x}_h(t+1) = \mathbf{x}_h(t) - \rho D_h^{-1}(t)[\nabla J(t)]_h\,. \tag{8}$$

Algorithm 1 formally describes the DRFE algorithm. Note that, in order to compute the $\mathbf{g}$'s and update the $\mathbf{x}$'s, Algorithm 1 requires two communication rounds per iteration.

Algorithm 1 DFRE

```
for $t \in \mathbb{N}$ each $h \in \mathcal{V}_c$ do
    sends $\mathbf{x}_h(t-1)$ to $k \in \mathcal{N}_h$;
    receives $\mathbf{x}_k(t-1)$ from $j \in \mathcal{N}_k$;
    computes $\mathbf{g}_h(t)$ as defined in (6);
    sends $\mathbf{g}_h(t)$ to $k \in \mathcal{N}_h$;
    receives $\mathbf{g}_k(t)$ from $k \in \mathcal{N}_k$;
    computes $\mathbf{x}_h(t)$ by using (8);
end for
```

Remark 2. (On the gradient computation).
Observe that to compute the gradient and, in particular, $\mathbf{z}_h$, $h \in \mathcal{V}_c$, the exchange of $\{\mathbf{x}_k\}_{k\in\mathcal{N}_h}$ among neighbors is required. However, it is worth noticing that only the entries of $\mathbf{x}_k$ corresponding to columns of H_{hk} which are different from the vector of all zeros, are needed. Consequently, the information exchange can be reduced.

3.2 Asynchronous, Robust, Fault Resilient Estimation

Algorithm 1 is designed for the scenario of synchronous ideal communications where neither packet losses nor delays occur. Here, inspired on the *Robust Block Jacobi* algorithm proposed in Todescato et al. (2015), we eventually generalize Algorithm 1 to the case of asynchronous and non ideal lossy communications. We refer to this modified version as *Robust Distributed Fault Resilient Estimation* algorithm (denoted hereafter as r-DFRE). In particular, we exploit an *asynchronous broadcast communication protocol* where one master node, say $h \in \mathcal{V}_c$, wakes up, updates its variables and, at the end of the computations, sends them to all its neighbors. Observe that the protocol requires only one communication round per iteration since, conversely to DFRE, we can send at the same time the state $\mathbf{x}_h$ and its gradient-related variable $\mathbf{g}_h$. However, in order to perform its local updates, it is necessary that each node stores in its local memory the following variables:

- $\mathbf{x}_h$: estimate of its state;
- $\mathbf{x}_k^{(h)}$, $k \in \mathcal{N}_h$: local estimate of the state of the neighboring area k (note that $\mathbf{x}_h^{(h)} \equiv \mathbf{x}_h$);
- $\mathbf{g}_h$: gradient-related variable regarding intra-area h information;
- $\mathbf{g}_k^{(h)}$, $k \in \mathcal{N}_h$: gradient-related variable regarding neighboring k information (note that $\mathbf{g}_h^{(h)} \equiv \mathbf{g}_h$).

Thanks to this additional memory not only we can use an asynchronous communication protocol, but we are also able to deal with packet losses in the communication. The latter can be conveniently modeled using the indicator function

$$\gamma_k^{(h)}(t) = \begin{cases} 1 \text{ if } h \text{ received the packet from } k \\ 0 \text{ otherwise} \end{cases}$$

as done in Todescato et al. (2015). Indeed, if node $k \in \mathcal{N}_h$ does not receive the packet that node h sends to it, then it simply does not update its memory. Namely,

$$\begin{aligned} \mathbf{x}_h^{(k)}(t) &= \begin{cases} \mathbf{x}_h(t) & \text{if } \gamma_h^{(k)} = 1 \\ \mathbf{x}_h^{(k)}(t-1) & \text{otherwise} \end{cases} \\ \mathbf{g}_h^{(k)}(t) &= \begin{cases} \mathbf{g}_h(t) & \text{if } \gamma_h^{(k)} = 1 \\ \mathbf{g}_h^{(k)}(t-1) & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

Consequently, when node k will wake up, it will use information regarding node h which are possibly out of date. Specifically Eq.(6) becomes

$$[\nabla J(t)]_h = -\sum_{k \in \mathcal{N}_h^+} \mathbf{g}_k^{(h)}(t)\,. \tag{10}$$

This algorithm, which uses memory, is our r-DFRE algorithm. Its formal description can be found in Algorithm 2.

Algorithm 2 r-DFRE

```
for $t \in \mathbb{N}$ do
    assume node $h \in \mathcal{V}_c$ wakes up;
    using (7) and (10), updates $\mathbf{x}_h(t)$ as in (8);
    computes $\mathbf{g}_h(t)$;
    sends $\mathbf{x}_h(t), \mathbf{g}_h(t)$ to $k \in \mathcal{N}_h$;
    if $\gamma_h^{(k)}(t) = 1$ then node $k$
        receives $\mathbf{x}_h(t)$ and $\mathbf{g}_h(t)$;
        updates $\mathbf{x}_h^{(k)}(t)$ and $\mathbf{g}_h^{(k)}(t)$ as in (9);
    end if
end for
```

3.3 Convergence Analysis of the r-DFRE Alogirthm

Here we present a convergence result regarding the r-DFRE algorithm described in Algorithm 2. In order to state our result, which is based on the convergence analysis of *partially asynchronous algorithms* provided in Bertsekas and Tsitsiklis (1989), we introduce the following assumptions concerning our overall communication system.

Assumption 3. (Persistent activation).
There exists a positive integer D such that each master node $h \in \mathcal{V}_c$ wakes up and performs its update at least once within any interval $[t, t+D]$.

Assumption 4. (Bounded packet-losses).
There exists a positive integer L such that the number of consecutive communication failures between every pair of neighboring nodes in the communication graph $\mathcal{G}_c$ is less than L.

Finally, since we need a technical assumption on the curvature of the prescribed cost function, we consider a slightly modified cost, that is

$$\|x\|_{1,\epsilon} := \sum_{i \in \overline{\mathcal{S}}} \sqrt{x_i^2 + \epsilon} + \sum_{i \in \mathcal{S}} (\alpha x_i^2 + \gamma) \tag{11}$$

where $\mathcal{S} := \{i \in \mathcal{V}_c : |x_i| > \overline{x}\}$ with $\overline{x}$ an arbitrary large scalar which acts as “saturation” point, $\overline{\mathcal{S}} := \mathcal{V}_c/\mathcal{S}$ and α and γ are chosen to ensure twice continuous differentiability of the cost at $\overline{x}$.

Basically, by choosing the cost of Eq.(11), we ensure the function to be strongly convex. Observe that, from a practical point of view, the saturation value $\overline{x}$ can be any arbitrarily large finite value. Thus, this modification does not practically influence the r-DFRE algorithm.
The next proposition characterizes the convergence[2] of Algorithm 2.

Proposition 5. (Convergence of r-DFRE algorithm).
Consider Problem 5 with the cost function of Equation (11), solved using the r-DFRE Algorithm 2. Let Assumptions 3 and 4 hold. Then, there exists $\overline{\rho}$ such that, for all $0 < \rho < \overline{\rho}$, it holds $\lim_{t\to\infty} \nabla J(x(t)) = 0$.

The proof can be found in Bof et al. (2016a).

4. SIMULATIONS

In this section we compare the r-DFRE algorithm with a “robustified” version of the ADMM algorithm proposed in Kekatos and Giannakis (2013), for which we use an asynchronous broadcast communication protocol and we assume that each master node can store in memory the last received information from its neighbors, similarly to what is done in Section 3.2. We consider a one dimensional environment of length $L = 200$[m] equally partitioned in $p = 20$ areas, each of them supervised by a master node. In each area the number of agents varies from 5 to 10. Inside each area, we assume the nodes are positioned and connected according to a bidirected *random geometric* measurement graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with connectivity radius r. Moreover we assume that each node is able to measure its absolute position as well. Regarding the inter-area connections, since we are working on a line, we assume that $\mathcal{N}_h = \{h-1, h+1\}$ for $h \in \{2, \ldots, p-1\}$, while $\mathcal{N}_1 = \{2\}$ and $\mathcal{N}_p = \{p-1\}$. To enforce connectivity, for each pair of adjacent areas we take m relative measurements corresponding to the m closest pairs of nodes. We assume the measurements are characterized by the same prior distribution. In particular, we set $\sigma_i = \sigma_{ij} = \sigma = 0.1$[m], $\forall i \in \mathcal{V}$,

[2] Interestingly, assuming the presence of bounded delays in the communication, the r-DFRE algorithm remains provably convergent. Indeed, this is true since the presence of bounded delays, together with the specific broadcast communication protocol chosen, implies the persistent activation of each node.

r-DFRE	ADMM	LS
0.0612	0.0594	0.2290

Table 1. Steady state estimation accuracy. Values of $ARMSE$, computed over $M = 1000$ Monte Carlo runs, for the solutions of problems $\mathcal{P}_1$, $\mathcal{P}_2$ and of LS, respectively, for fixed values of $r = 3$[m] and $m = 5$.

$\forall (i,j) \in \mathcal{E}$. Concerning the outliers, we assume that 10% of the measurements are corrupted by an additive noise, whose absolute amplitude is uniformly distributed in the range $[1.5, 2]$[m]. For consistency, in case the outlier corrupts a relative distance measure among agents of the same area, the measurement is saturated to the communication radius r characterizing the intra-areas random geometric graph $\mathcal{G}$. A similar approach is followed for the case of corrupted inter-areas relative measurements. Concerning ϵ, its value influences the robustness of $\|\cdot\|_{1,\epsilon}$ to the presence of outliers and eventually the accuracy of the solution, but at the same time, by controlling the smoothness of $\|\cdot\|_{1,\epsilon}$, it also influences the rate of convergence of our proposed gradient-based algorithm. In the next simulations, we heuristically set $\epsilon = \frac{1}{4}\sigma^{8/3}$, which enforces the cost function $\|\cdot\|_{1,\epsilon}$ to behave quadratically within $[-\sigma, \sigma]$. Preliminary simulations show that this value represents a good trade-off between accuracy and rate of convergence. For the choice of λ, following Kekatos and Giannakis (2013), we set $\lambda = 1.34\sigma$. [3]

We compare the two algorithms in terms of averaged root mean squared error ($ARMSE$). Given M Monte Carlo runs for different graph realizations, denote with $\mathbf{x}_{\{i\}}(t)$ the estimate at time t given by one of the two algorithms in the $i-$th Monte Carlo run. Then, $ARMSE$ reads as

$$ARMSE(t) := \frac{1}{M}\sum_{i=1}^{M} RMSE(\mathbf{x}_{\{i\}}(t)) \ , \qquad (12)$$

where $RMSE(\mathbf{x}) := \|\mathbf{x} - \mathbf{x}_{\text{true}}\|/\sqrt{N}$ represents the root mean squared error between the true nodes positions, $\mathbf{x}_{\text{true}}$, and the estimate, $\mathbf{x}$. We first use ARMSE to compare the steady state accuracy of the estimates obtained with the r-DFRE and the ADMM, that is we evaluate $ARMSE$ as $t \to \infty$. Moreover, we also compare the algorithm with the classical least squares approach (LS) to verify its effectiveness. Table 1 highlights the complete inaccuracy of LS. Conversely, the other two approaches are fault resilient and comparable, even though ADMM gives, in general, slightly better results.

We also compare the r-DFRE and the ADMM with respect to their convergence rate [4]. To do so, we need to choose the parameters ρ for the r-DFRE and c for the ADMM (which represents the penalty in the augmented Lagrangian). We follow the common approach of selecting the values $\rho = \rho^*$ and $c = c^*$, respectively, corresponding to the fastest asymptotic convergence rate [5]. This is done in order to minimize the number of iterations to converge toward to optimal solution of the corresponding problem.

Figure 1 shows, in the upper panel, the evolution of the ARMSE between the solution of the algorithms at time t, $\mathbf{x}(t)$, and the solution of the respective problem, while, in the lower panel, it shows the ARMSE between $\mathbf{x}(t)$ and the true positions $\mathbf{x}_{\text{true}}$. For the ADMM, it is useful to report

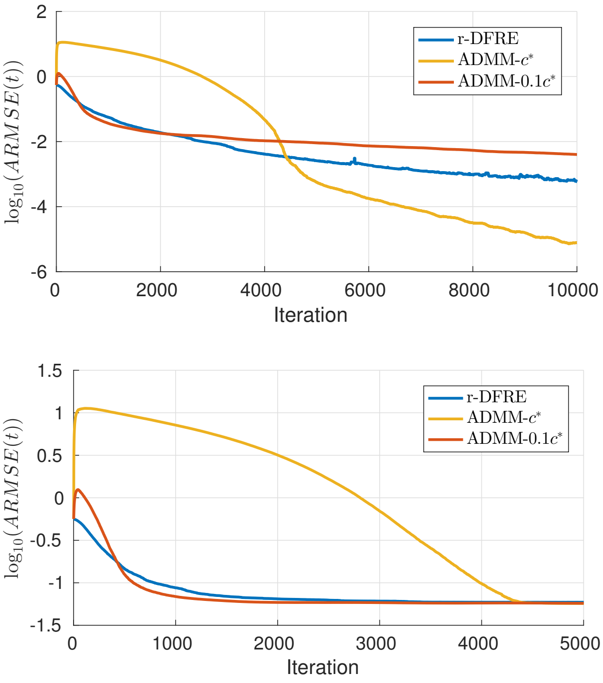

Fig. 1. Upper panel: ARMSE, over $M = 100$ Monte Carlo runs, with respect to the optimal solution of the respective problem. Lower panel: ARMSE with respect to the true positions. The parameter used for the simulations are $m = 5$ and $r = 3$ [m].

$\mathbf{m = 5}$	$r = 3$[m]	$r = 5$[m]	$r = 8$[m]
r-DFRE	2593	1089	811
ADMM 0.1c^*	1643	1597	1526
ADMM c^*	4361	4886	4896

Table 2. Number of iterations, averaged over $M = 100$ runs, required to reach a 95% accuracy from the optimal solution, for increasing intra-area communication radius r and for $m = 5$ inter-area connections.

the simulations obtained not only for $c = c^*$ but also for $c = c^*/10$. This choice is due to the different convergence behavior of the ADMM. Indeed, from the upper panel of Figure 1 it can be seen that, by choosing $c = c^*$, the ADMM is characterized by an undesirable transient but converges to its corresponding optimum faster than the r-DFRE. Conversely, by choosing $c = c^*/10$, the ADMM shows a better transient but slows down its asymptotic behavior, becoming even slower than the r-DFRE. However, by comparing the algorithms in terms of estimation accuracy with respect to the true positions, as shown in the lower panel of Figure 1, a different behavior emerges. In particular, for $c = c^*$ the ADMM is much slower than the r-DFRE to reach the same level of accuracy. Conversely, by choosing $c = c^*/10$, the ADMM and the r-DFRE are characterized by a similar behavior both in terms of transient and asymptotic evolution. A final remarkable feature of the r-DFRE compared to the ADMM, is that the transient evolution is monotonically decreasing.

We recall that the possible discrepancy between transient and asymptotic behavior is a known fact, see Fagnani and Zampieri (2008). As highlighted by the previous analysis, this translates in a non trivial procedure to find the optimal value of c for the ADMM. Conversely, for the r-DFRE, the extensive simulations performed suggest the fact that the optimal ρ leads to an optimal transient and asymptotic behavior simultaneously.

To highlight the dependency of the algorithms on the

[3] In Kekatos and Giannakis (2013), $\lambda = 1.34$ since the measurements are normalized by their standard deviation. Equivalently $\sigma = 1$.

[4] In all these simulations, both algorithms are initialized to the same initial conditions, which correspond to the absolute measurements.

[5] To give a quantitative idea, in all the simulation performed, the order of magnitude of ρ^* and c^* are -4 and -2, respectively.

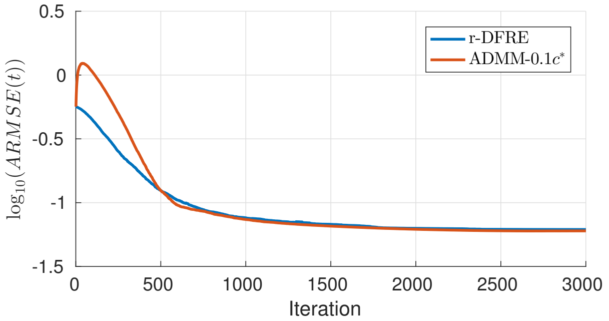

Fig. 2. ARMSE, averaged over $M = 100$ runs, between the estimate of the positions of the agents and their true positions, with a packet loss probability of 50%. The parameter used for the simulations are $m = 5$ and $r = 3$ [m].

sensors network connectivity, we perform simulations for different values of intra-area communication radius r. Table 2 shows, for a number $m = 5$ of inter-area connections and increasing values of the communication radius r, the number of iterations needed to reach an estimate withing a 95% level of accuracy from the optimal solution of the corresponding problem. Observe that increasing r, the r-DFRE highly improves, the ADMM corresponding to $c = c^*/10$ slightly improves while for $c = c^*$ moderately degrades. The analysis just performed suggests that the connectivity of the underlying graph affects the behavior of the algorithm and in general it is a quantity that must be taken into account. Remarkably, even if in a totally different scenario, the connection between highly connected graphs and deteriorating performance of ADMM (as happens when choosing c^*) has been shown in the recent Bof et al. (2016b) as well.

Finally, Figure 2 shows a simulation in the presence of packet losses. In particular, a packet loss probability of 50% is chosen. The plot shows that both algorithms are robust to packet losses. However, as stated in Proposition 5, convergence of the r-DFRE is theoretically proven. Conversely, a similar rigorous result for the ADMM is not available.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this work we presented an asynchronous and robust algorithm which can be used to perform fault resilient estimation in presence of outliers. We applied it to a localization-type problem and showed that its results, in terms of accuracy and convergence behavior, are comparable with those of a (slightly modified) ADMM algorithm recently proposed by Kekatos and Giannakis (2013). Through simulations, we showed that both the parameters choices and the connectivity of the underlying graph play a fundamental role in the convergence behavior of the algorithms. In particular, our algorithm behaves consistently in the transient and in steady state, thus alleviating the parameter tuning phase, and for highly connected graphs, our solution outperforms the ADMM.

As future directions, we are interested in better understanding the dependencies of the algorithm on the parameter choice as well as on the graph connectivity related quantities. Moreover, a deeper analysis on the robustness of the algorithm represent an interesting research avenue. Finally, detection of faults and extensions to general nonlinear measurements models are interesting topics.

REFERENCES

Argaez, M., Ramirez, C., and Sanchez, R. (2011). An ℓ_1-algorithm for underdetermined systems and applications. In *Fuzzy Information Processing Society (NAFIPS), 2011 Annual Meeting of the North American*, 1–6. IEEE.

Bertsekas, D.P. and Tsitsiklis, J.N. (1989). *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ.

Bloomfield, P. and Steiger, W. (2012). *Least absolute deviations: Theory, applications and algorithms*, volume 6. Springer Science & Business Media.

Boem, F., Ferrari, R.M., and Parisini, T. (2011). Distributed fault detection and isolation of continuous-time non-linear systems. *European Journal of Control*, 17(5), 603–620.

Boem, F., Xu, Y., Fischione, C., and Parisini, T. (2013). Distributed fault detection using sensor networks and pareto estimation. In *Control Conference (ECC), 2013 European*, 932–937. IEEE.

Bof, N., Todescato, M., Carli, R., and Schenato, L. (2016a). Proofs of robust estimation for localization in lossy sn. Technical report. URL `http://automatica.dei.unipd.it/people/todescato/publications.pdf`.

Bof, N., Carli, R., and Schenato, L. (2016b). On the performance of consensus based versus lagrangian based algorithms for quadratic cost functions. In *Control Conference (ECC), 2016 European*. IEEE.

Chen, J., Kher, S., and Somani, A. (2006). Distributed fault detection of wireless sensor networks. In *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, 65–72. ACM.

Choi, D.H. and Xie, L. (2011). Fully distributed bad data processing for wide area state estimation. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, 546–551. IEEE.

Fagnani, F. and Zampieri, S. (2008). Randomized consensus algorithms over large scale networks. *Selected Areas in Communications, IEEE Journal on*, 26(4), 634–649.

Franco, E., Olfati-Saber, R., Parisini, T., and Polycarpou, M.M. (2006). Distributed fault diagnosis using sensor networks and consensus-based filters. In *Decision and Control, 2006 45th IEEE Conference on*, 386–391. IEEE.

Huber, P.J. (2011). *Robust statistics*. Springer.

Kekatos, V. and Giannakis, G. (2013). Distributed robust power system state estimation. *Power Systems, IEEE Transactions on*, 28(2), 1617–1626.

Keliris, C., Polycarpou, M.M., and Parisini, T. (2015). Distributed fault diagnosis for process and sensor faults in a class of interconnected input–output nonlinear discrete-time systems. *International Journal of Control*, 88(8), 1472–1489.

Korres, G.N. (2011). A distributed multiarea state estimation. *Power Systems, IEEE Transactions on*, 26(1), 73–84.

Mao, G., Fidan, B., and Anderson, B.D. (2007). Wireless sensor network localization techniques. *Computer networks*, 51(10), 2529–2553.

Paradis, L. and Han, Q. (2007). A survey of fault management in wireless sensor networks. *Journal of Network and Systems Management*, 15(2), 171–190.

Todescato, M., Cavraro, G., Carli, R., and Schenato, L. (2015). A robust block-jacobi algorithm for quadratic programming under lossy communications. *IFAC-PapersOnLine*, 48(22), 126–131.