4th IFAC Workshop on Distributed Estimation
and Control in Networked Systems
September 25-26, 2013. Rhine-Moselle-Hall, Koblenz, Germany

Adaptive consensus-based algorithms for fast estimation from relative measurements ⋆

Carron Andrea * **Todescato Marco** * **Carli Ruggero** *
Schenato Luca *

* *Univeristy of Padova, Department of Information Engineering, via Gradenigo 6/B,*
email: [carlirug, carronan, schenato, todescat]@dei.unipd.it

Abstract: In this work we propose a novel asynchronous algorithm to solve the problem of optimal estimating the position of each agent in a network from relative noisy vectorial distances with its neighbours. This algorithm is based on the combination of two asynchronous algorithms which have been recently proposed in the literature for localization problems. Specifically, we analyze various switching strategies from an asynchronous algorithm which exhibits a fast transient but does not convergence to an optimal solution, to another asynchronous algorithm which is slower in the transient but reaches an optimal solution exponentially. We provide a large set of simulations to compare the different switching strategies and to verify which one performs better.

Keywords: Adaptive Algorithms; Distributed Control; Optimization; Relative Measurements Localization

1. INTRODUCTION

The proliferation of relatively inexpensive devices capable of communicating, computing, sensing, interacting with the environment and storing information is promising an unprecedented number of novel applications throughout the cooperation of these devices toward a common goal. These applications include swarm robotics, wireless sensor networks, smart energy grids, smart traffic networks, smart camera networks. These applications also pose new challenges, of which *scalability* is one of the major ones. Scalability is intended as the ability for an application to continue functioning without any dramatic performance degradation even if the number of devices involved keep increasing. In particular, an application is scalable if it is not necessary to increase either HW or SW resources in each device even if the total number of devices increases.

In this work we address the problem of designing algorithms that are capable to reconstruct the optimal estimate of the location of a device from noisy relative measurements from its neighbors in a connected network. In particular, combining two existing algorithms we want to design a novel distributed strategy that, exploiting their main features, allows each device to reconstruct its own position only from exchanging information with its neighbors, regardless of the size of the network. Moreover, this novel algorithm must be scalable, i.e. its computational complexity, bandwidth and memory requirements should be independent of the network size. This algorithm will be obtained as the solution of an optimization problem.

Distributed optimization has been attracting ever growing attention in the past years since many problems in large scale network have been cast as convex optimization problem. In particular, a large class of problem can be cast as the solution of the following optimization problem

$$\begin{aligned} x^* = \text{argmin}_{x_1,\dots,x_N} & \sum_{i=1}^{N} f_i(x_i) \\ s.t. \quad & x_i = x_j, \forall i,j \end{aligned} \tag{1}$$

where N is the number of nodes, $x_i \in \mathbb{R}^m$, and f_i are convex functions. The function f_i represents local cost of each agent i, but each agent must compute the minimizer x^* of the sum all local costs. The local cost functions are separable but the additional contraint that all local variables x_i must be the same, makes the optimization problem coupled. Many problems can be cast in this terms such as distributed least squares (Xiao et al. [2005], Bolognani et al. [2010]), map building (Schwager et al. [2009]), network utility maximization (Johansson et al. [2006]), distributed learning and support vector machines (Boyd et al. [2011]). Several approaches have been proposed such as the distributed subgradient methods (SDMs) (Nedic and Ozdaglar [2009]), the alternating direction method of multipliers (ADMM)(Boyd et al. [2011]), the Newton-Raphson consensus (Zanella et al. [2011]) and the control-based methods (Wang and Elia [2010]).

The problem at hand in this work is of a different type and can be cast as the following unconstrained optimization problem:

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

where $x_i \in \mathbb{R}^\ell$, $\mathcal{E}$ represents all the pair of nodes for which are available relative measurements and f_{ij} are convex functions. Differently from the previous optimization problem, the local variables are unconstrained, but the cost functions are now not separable, and therefore the problem is once again coupled. Many problems can be cast in this framework such as sensor localization (Barooah and Hespanha [2005], Barooah [2007]), sensor calibration

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

(Bolognani et al. [2010]), clock synchronization (Solis et al. [2006]) and camera localization (Borra et al. [2012], Tron and Vidal [2009]). For example, in the context of localization from vectorial relative distance in a plane, the cost function f_{ij} are given by:

$$f_{ij}(x_i - x_j) = ||x_i - x_j - z_{ij}||^2$$

where $z_{ij} \in \mathbb{R}^\ell$ is the noisy measurement of the relative (vector) distance of node i from node j. As a consequence, the optimization problem in Eqn. (2) becomes a distributed least-square problem that in principle could be cast as the optimization problem in Eqn. (2). However, in this case, each node will need to compute the location of all other nodes, i.e. the size of the local variable becomes of size $m = N\ell$, which according to our objective is not scalable. There exist alternative approaches that try to exploit the special structure of the problem, but they either show oscillatory behavior as in consensus-based with constant weights (Bolognani et al. [2010]), or the convergence rate decreases only as $1/k$ as in the Randomized Kaczmarz with Under-Relaxation (Freris and Zouzias [2012]). This slow convergence rate is mainly due to the fact that these algorithms adopt asynchronous gradient-based algorithms whose step-size decreases to zero as progress.

The contribution of this work is to provide an adaptive strategy based on two asynchronous algorithms recently proposed in the literature. The first, based on a modified implementation of the algorithm proposed in Bolognani et al. [2010], presents a fast transient, but it does not converge to the optimal estimate. The second, presented in Carli et al. [2013], on the contrary, presents a relatively slow transient, but it converges to an optimal estimate exponentially. The aim will be to appropriately combine the algorithms to ensure an higher performances strategy.

2. MATHEMATICAL PRELIMINARIES

We introduce now some preliminary notations. The symbols $\mathbb{R}^N$ and $\mathbb{R}^{M\times N}$ denote the vector space of N dimensional column vectors and $M \times N$ matrices respectively, where all the entries are real.
With the superscript ^{T} we denote the transpose of a matrix. The symbol $\mathbf{1}$ represents a column vector where all the entries are equal to one, whereas e_i, $i = \{1, \dots, N\}$ denotes the column vector in R^N having all entries equal to zero except a 1 in postion i. The symbol I represents the identity matrix. We denote with the symbol $\| \cdot \|$ the 2-norm on $\mathbb{R}^N$ and the induced norm in $\mathbb{R}^{N\times N}$. Given a matrix $A \in \mathbb{R}^{N\times N}$ the symbol $sr(A)$ denotes the spectral radius of A. Given a stochastic matrix $P \in \mathbb{R}^{N\times N}$ the symbol $esr(P)$ denotes the essential spectral radius of P, i.e., its second largest eigenvalue in absolute value.
A directed graph $\mathcal{G}$ is defined as a pair $(V, \mathcal{E})$ where $V = \{1, \dots, N\}$ and $\mathcal{E} \subseteq V \times V$. The set V is called set of vertices and the set $\mathcal{E}$ is called the set of edges. If $\mathcal{E}$ is such that $(i, j) \in \mathcal{E}$ then also $(j, i) \in \mathcal{E}$, the associated graph is called bidirected. With $| \mathcal{E} | = M$ we denote the cardinality of the set $\mathcal{E}$. We also introduce the incidence matrix $A \in \mathbb{R}^{M\times N}$ 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. Given a directed graph, a path from a node to another node that does not respect the orientation of the edges is called an *undirected path*. A directed graph is said to be *weakly connected* if there is a undirected path form any node to any other node. The expectation operator is denoted with the symbol $\mathbb{E}$. Given a set $\mathcal{A}$ with a finite number of elements, by $| \mathcal{A} |$ we denote its cardinality.

3. PROBLEM FORMULATION

The problem we aim at solving is that of estimating N variables $x_1, \dots, x_N$ from relative measurements of the form

$$z_{ij} := x_i - x_j + n_{ij}, \qquad i, j \in \{1, \dots, N\}, \tag{3}$$

corrupted by zero-mean measurement noise, n_{ij}. In this paper we assume that $x_i \in \mathbb{R}$, $i \in \{1, \dots, N\}$, though the variables are often vector-valued.

It is natural to associate this estimation problem with a *measurement graph* $\mathcal{G} = (V; \mathcal{E})$ whose the vertex set V consists of the set of nodes $V = \{1, \dots, N\}$ where N is the number of nodes, while its edge set $\mathcal{E}$ consists of all of the ordered pairs of nodes (i, j) such that a noisy measurement of the form (3) between i and j is available to node i. The measurement errors on distinct edges are assumed uncorrelated. Moreover, the measurement graph $\mathcal{G}$ is a directed graph because the measurements z_{ij} and z_{ji} are available at both the nodes i and j, and the measurements are in general different.

We firstly introduce some preliminary definitions. Now, let $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{z} \in \mathbb{R}^M$ and $\mathbf{n} \in \mathbb{R}^M$ be the vector obtained stacking together all the variables $x_1, \dots, x_N$, i.e., $\mathbf{x} = [x_1, \dots, x_N]^T$, all the measurements z_{ij} and all the noises n_{ij}, respectively. Additionally, we denote as $R_{ij} > 0$ the covariance of the zero mean error n_{ij}, i.e., $R_{ij} = \mathbb{E}[n_{ij}^2]$, 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., $R = \mathbb{E}[\mathbf{n}\mathbf{n}^\mathbf{T}]$.

Observe that equation (3) can be rewritten in a vector form as

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

Define the set

$$\chi := \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{argmin}} (\mathbf{z} - A\mathbf{x})^T R^{-1} (\mathbf{z} - A\mathbf{x}).$$

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

$$x^* \in \chi \tag{4}$$

Assume the measurement graph $\mathcal{G}$ to be *weakly connected*, then it is well known (see Barooah [2007]) that

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

Moreover let

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

then x^*_{opt} is the minimum norm solution of (4), i.e.,

$$x^*_{\text{opt}} = \min_{x^* \in \chi} \| x^* \|$$

Remark 3.1. 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., $x_1 = 0$.

4. ASYNCHRONOUS DISTRIBUTED CONSENSUS ALGORITHMS

To compute an optimal estimate x^*, one needs all the measurements and their covariances (z, R), and the topology of the measurement graph $\mathcal{G}$. In this paper we consider

algorithms that compute an 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 if either $(i,j) \in \mathcal{E}$ or $(j,i) \in \mathcal{E}$. Accordingly a node i is said to be a neighbor of node j (and vicecersa) if either $(i,j) \in \mathcal{E}$ or $(j,i) \in \mathcal{E}$. For $i \in \{1,\dots,N\}$, by $\mathcal{N}_i$ we denote the set of neighbors of node i, namely

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

In this section we review two asynchronous algorithms which are based on a standard consensus strategy. In these algorithms the nodes are assumed to communicate with each other through an *asymmetric broadcast communication protocol*; more precisely, at each iteration of the algorithms, there is only one node which wakes up and communicates its estimate to all its neighbors that, based on the received information, perform their state update. The transmitting node is selected in a randomly way. In particular in the rest of the paper we refer to the scenario described by the following definition.

Definition 4.1. A network of N nodes is said to be a *randomly persistent communicating network* if there exists a N-upla of probabilities $(\beta_1,\dots,\beta_N)$ such that $\beta_i > 0$, for all $i \in \{1,\dots,N\}$, and $\sum_{i=1}^{N} \beta_i = 1$, and such that, for all $k \in \mathbb{N}$,

$$\mathbb{P}[\text{the transmitting node at iteration } k \text{ is node } i] = \beta_i.$$

4.1 Memory-less asynchronous consensus algorithm

The first algorithm we review is based on a slightly modification of the algorithm proposed in Bolognani et al. [2010]. In Bolognani et al. [2010] the authors adopted a *symmetric gossip communication protocol* in place of a *asymmetric broadcast communication protocol*. This algorithm is a memory-less algorithm (denoted hereafter as ML-alg), in the sense that, each node keeps in memory only an estimate of its own position and not a copy of the estimates of other nodes. A formal description of the ML-alg, adopting an asymmetric broadcast communication protocol, is provided in Algorithm 1.

Remark 4.2. As stressed by the authors in Bolognani et al. [2010] and supported by several simulations, the trajectory of $\hat{x}$ does not converge to an optimal solution x^* but just exhibits an oscillatory behavior around x^*. However, the ML-alg exhibits a fast transient after which it starts to oscillate within a neighborhood of x^*.

4.2 Memory-based asynchronous consensus algorithm

The second algorithm we review is a memory based asynchronous algorithm (denoted as MB-alg hereafter) that has been first proposed in Carli et al. [2013]. It is mainly based on the idea that since the actual values of neighboring estimates are not always available, each node stores in its local memory a copy of the neighbors' variables recorded from the last communication its neighbors performed. Specifically, for $j \in \mathcal{N}_i$, we denote by $\hat{x}_j^{(i)}(k)$ the estimate of x_j kept in i's local memory at the end of the k-th iteration. If node j performed its last transmission to node i during h-th iteration, $h \le k$, then $\hat{x}_j^{(i)}(k) = \hat{x}_j(h)$.

The MB-alg is formally described in algorithm 2

where

$$b_i = \delta \sum_{(i,j)\in\mathcal{E}} R_{ij}^{-1} z_{ij} - \delta \sum_{(j,i)\in\mathcal{E}} R_{ji}^{-1} z_{ji}$$

Algorithm 1 ML-alg

Require: $\forall i \in V$: node i stores in memory the measurements $\{z_{ij}, z_{ji}; R_{ij}, R_{ji};\ j \in \mathcal{N}_i\}$ with associated covariances. Moreover initializes an estimate $\hat{x}_i(0)$ of x_i.

Transmission
for k = 1,2,... **do**
  one node, say i, with probability β_i wakes up and transmits its estimate to all its neighbors $j \in \mathcal{N}_i$
end for

Update
for $j \in V$ **do**
  if $j \in \mathcal{N}_i$ **then**
    node j updated its state as follows:

$$\hat{x}_j(k+1) = \frac{1}{2}\left(\hat{x}_i(k) + \hat{x}_j(k)\right) + \frac{1}{4}\left(R_{ij}^{-1} z_{ij} - R_{ji}^{-1} z_{ji}\right) \tag{5}$$

  else
    $\forall j \notin \mathcal{N}_i$ node j remains unchanged, i.e.

$$\hat{x}_j(k+1) = \hat{x}_j(k)$$

  end if
end for

Algorithm 2 MB-alg

Require: $\forall i \in \{1,\dots,N\}$ every node i stores in memory the measurements $\{z_{ij}, R_{ij}; (i,j) \in \mathcal{E}\}$ and $\{z_{ji}, R_{ji}; (j,i) \in \mathcal{E}\}$ with the associeted covariances. Moreover node i stores also an estimate $\hat{x}_i$ of x_i and, for $j \in \mathcal{N}_i$, an estimate $\hat{x}_j^{(i)}$ of $\hat{x}_j$ initialized to arbitrary values.

Transmission
for $k = 1,2,\dots$ **do**
  only one node, say i, with probability β_i transmits its estimate $\hat{x}_i(k)$ to all its neighbors $j \in \mathcal{N}_i$.
end for

Update
for $j \in \mathcal{N}_i$ **do**
  node j performs the following actions in order:
  (i) it sets $\hat{x}_i^{(i)}(k+1) = \hat{x}_i(k)$
  (ii) it leaves $\hat{x}_s^{(j)}(k+1) = \hat{x}_s^{(j)}(k)\ \forall s \in \mathcal{N}_j/\{i\}$
  (iii) it updates $\hat{x}_j$ as follows

$$\hat{x}_j(k+1) := p_{jj}\hat{x}_j(k) + \sum_{h\in\mathcal{N}_j} p_{jh}\hat{x}_h^{(j)}(k+1) + b_j \tag{6}$$

end for

for $s \notin \mathcal{N}_i$ **do**
  node s left unchanged, i.e. $\hat{x}_s(k+1) = \hat{x}_s(k)$
end for

and where

$$p_{ij} = \begin{cases} \delta(R_{ij}^{-1} + R_{ji}^{-1}) & \text{if } (i,j) \in \mathcal{E} \text{ and } (j,i) \in \mathcal{E} \\ \delta R_{ij}^{-1} & \text{if } (i,j) \in \mathcal{E} \text{ and } (j,i) \notin \mathcal{E} \\ \delta R_{ji}^{-1} & \text{if } (j,i) \in \mathcal{E} \text{ and } (i,j) \notin \mathcal{E} \end{cases}$$

and

$$p_{ii} = 1 - \sum_{j\in\mathcal{N}_i} p_{ij}$$

being δ a positive constant *a-priori* assigned to the nodes. Now, let $P \in \mathbb{R}^{N\times N}$ be the matrix defined by the weights p_{ij} above introduced. One can see the matrix P is equal to

$$P = I - \delta A^T R^{-1} A$$

For sufficiently small values of δ, P is a stochastic matrix.

Remark 4.3. The MB-alg is shown to converge exponentially to an optimal solution x^*, (the proof of this fact can be found in Carli et al. [2013]) provided that the following two conditions are satisfied:

(i) the measurements graph $\mathcal{G}$ is weakly connected,
(ii) $0 < \delta < 1/(2d_{max}R_{min}^{-1})$,

where $d_{max} = \max\{|\ \mathcal{N}_i\ |, i \in \{1,\dots,N\}\}$ and let $R_{min} = \min\{R_{ij}, (i,j) \in \mathcal{E}\}$.

5. HYBRID STRATEGY

In this section we introduce an hybrid algorithm (denoted hereafter as H-alg) which is based on the combination of MB-alg and ML-alg. The idea is to exploit the better transient performance of ML-alg and the asymptotic exponential convergence of MB-alg. Loosely speaking, the main features of the H-alg are the following two:

(i) For $i \in \{1,\dots,N\}$, node i runs ML-alg during the transient;
(ii) For $i \in \{1,\dots,N\}$, node i opportunely switches from ML-alg to MB-alg as soon as its estimate $\hat{x}_i$ starts to exhibit oscillating behavior.

We stress that, in the distributed scenario we deal with, the H-alg assumes every node to work standalone. In other words, every node will switch from the ML-alg to the MB-alg independently from all the other nodes. Recall line 9 of algorithm 1 and line 7 of algorithm 2. Then the H-alg updating law is defined as follows

$$\hat{x}_j(k+1) = \frac{w_j(k)}{2}\left(\hat{x}_i(k) + \hat{x}_j(k) + \frac{1}{2}\left(R_{ij}^{-1}z_{ij} - R_{ji}^{-1}z_{ji}\right)\right) + (1-w_j(k))\Big(p_{jj}\hat{x}_j(k) + \sum_{h\in\mathcal{N}_j} p_{jh}\hat{x}_h^{(j)}(k+1) + b_j\Big) \tag{7}$$

where $w_j(k) \in [0,1]$ represents the chosen weight for combining the two strategy at the k-th iteration.

The H-alg is formally described in Algorithm 3.

Now we propose three strategies for choosing $\{w(k)\}_{k\in\mathbb{N}}$.

Heuristic 1: Let $\bar{M} = \mathbb{E}[M_i] = \sum_{i=1}^N \beta_i M_i$ the average ML-alg matrix. As shown in Carli et al. [2013], $\bar{M}$ is a stochastic matrix. Since the algorithm has an exponential transient evolution one can approximate it as $e^{-\frac{t}{\tau}}$ where the time constant τ is equal to

$$\tau = -\frac{1}{\log(\rho)}$$

where $\rho = esr(\mathbb{E}[\bar{M}])$. The strategy requires all nodes to store in their memories the value of τ. Moreover, for $i \in \{1,\dots,N\}$, node i keeps in memory also a counter $c_i(k)$ which counts how many updates node i has performed up to the k-th iteration of the H-alg. As soon as $c_i(k) > \tau|\mathcal{N}_i|/N$, node i switches from the ML-alg to the MB-alg, namely, w_i is set definitively equal to 0.

Algorithm 3 H-alg

Require: $\forall i \in \{1,\dots,N\}$ every node i stores in memory the measurements $\{z_{ij}, R_{ij}; (i,j) \in \mathcal{E}\}$ and $\{z_{ji}, R_{ji}; (j,i) \in \mathcal{E}\}$ with the associated covariances. Moreover node i stores also an estimate $\hat{x}_i$ of x_i and, for $j \in \mathcal{N}_i$, an estimate $\hat{x}_j^{(i)}$ of $\hat{x}_j$ initialized to arbitrary values.

Transmission
for $k = 1, 2, \dots$ **do**
  only one node, say i, transmits its estimate $\hat{x}_i(k)$ to all its neighbors $j \in \mathcal{N}_i$.
end for

Update
for $k = 1, 2, \dots$ **do**
  if $j \in \mathcal{N}_i$ **then**
    Choose heuristic 1, 2 or 3 and check it, and accordingly assign the value to $w_j(k)$ then, opportunely update $\hat{x}_j(k+1)$ according to equation (7).
  else
    $\forall s \notin \mathcal{N}_i$ node s remains unchanged, i.e.
$$\hat{x}_s(k+1) = \hat{x}_s(k)$$
  end if
end for

Heuristic 1 Hard Switch

Require: Given a node $i \in \{1,\dots,N\}$, the weight $w_i(k)$ is initialized to one.
if $c_i(k) > \tau|\mathcal{N}_i|/N$ **then**
  $w_i(k) = 0$
end if

Remark 5.1. Observe that, if the probabilities β_i, $i \in \{1,\dots,N\}$, are uniform, i.e., $\beta_1 = \dots = \beta_N = 1/N$, then the nodes will switch around the same iteration.

Remark 5.2. The computation of τ require an a priori knowledge of the network which could not be ensure.

Heuristic 2: Observe that, in the previous heuristic, the parameters w_i abruptly changes from 1 to 0. Instead in this heuristic we assume that w_i is a smooth decreasing function of the counter $c_i(k)$; specifically $w_i(k) = e^{-\frac{c_i(k)N}{\tau|\mathcal{N}_i|}}$, where, again, $\tau = -\frac{1}{\log(\rho)}$.

Heuristic 2 Smooth Switch

Require: Given a node $i \in \{1,\dots,N\}$, the weight $w_i(k)$ is initialized to one.
$w_i(k) = e^{-\frac{c_i(k)N}{\tau|\mathcal{N}_i|}}$

In this heuristic, as in the previous one, the parameter τ, represents an appropriate time constant of the network, tries to "predict" the behaviour of the state convergence.

Heuristic 3: The final strategy we present is based on a statistical behavior of the state evolution. Given $n \in \mathbb{N}$, assume that, for $i \in \{1,\dots,N\}$, node i keeps in memory the values of the last $2n$ updates of its own estimate it has performed. Without loss of generality, assume that these updates have occured during iterations $k_1 < k_2 < \dots < k_{2n}$. Let us define $\bar{x}_{i_{new}} = \frac{1}{n}\sum_{t=k_{n+1}}^{k_{2n}} x_i(t)$, $\bar{x}_{i_{old}} = \frac{1}{n}\sum_{t=k_1}^{k_n} x_i(t)$ and $\hat{\sigma}_i = \sqrt{\frac{1}{n}\sum_{t=k_{n+1}}^{k_{2n}}(x_i(t) - \bar{x}_{i_{new}})^2}$. Observe that $\bar{x}_{i_{new}}$ and $\bar{x}_{i_{old}}$ represent the sample mean of the state update over the two windows of length n, $[k_1, k_n]$ and

$[k_{n+1}, k_{2n}]$, respectively, while $\hat{\sigma}_i$ represents the sample standard deviation over $[k_{n+1}, k_{2n}]$. Then, if the difference between $\bar{x}_{i_{new}}$ and $\bar{x}_{i_{old}}$ is less than the normalized sample standard deviation $\hat{\sigma}_i/\sqrt{n}$, the node switches from the ML-alg to the MB-alg.

Heuristic 3 Statistical Switch

Require: $\forall$ node $i \in \{1, \dots, N\}$, initialize the weight $w_i(k)$ to one. Moreover a window of dimension n is given.

1: $\bar{x}_{i_{new}} = \frac{1}{n}\sum_{t=k_{n+1}}^{k_{2n}} x_i(t)$
2: $\bar{x}_{i_{old}} = \frac{1}{n}\sum_{t=k_1}^{k_n} x_i(t)$
3: $\hat{\sigma}_i = \sqrt{\frac{1}{n}\sum_{t=k_{n+1}}^{k_{2n}} (x_i(t) - \bar{x}_{i_{new}})^2}$
4: **if** $|\bar{x}_{i_{new}} - \bar{x}_{i_{old}}| < \hat{\sigma}_i/\sqrt{n}$ **then**
5: $\quad w_i(k) = 0$
6: **end if**

This strategy take advantage of the fact that the oscillatory behavior of the state within a neighbor of an optimal solution suggests the process to be steady state stationary. Moreover it does not require any a priori knowledge of the network.

Remark 5.3. Note that all the strategies are completely local since every node does not need any global counter.

6. SIMULATION RESULTS

Now we show some simulation results to show how the different switches strategy affect the hybrid implementation. All simulations are carried out over a random geometric bidirected connected graph of 50 nodes uniformly distributed in a squared area of side $\sqrt{50}$. Specifically, nodes $i, j \in V$ are considered connected if their relative distance is less than or equal to 1.8 , i.e. $|x_i - x_j| \leq 1.8$. In the simulations have been taken into account only configurations where the generated graphs resulted to be connected. Relative measurements are corrupted with zero mean gaussian noise with standard deviation equal to 0.01. The simulations report the behavior of the cost function

$$J(k) = \log(\| A\hat{x}(k) - Ax^*_{opt} \|)$$

All the numerical results we report have been obtained implementing the randomized scenario of definition 4.1 with uniform probability namely, $\beta_1 = \ldots = \beta_N = 1/N$.

Example 6.1. In this example we show a typical behavior of the H-alg's using the "Hard" switching strategy. Figure 1 shows how, for $c_i(k) > \tau|\mathcal{N}_i|/N$, the H-alg abruptly switches from the ML-alg to the MB-alg, causing an evident change in the convergence. The plot reports three realizations of the H-alg for different values of τ, specifically for τ, 0.1τ and 10τ. As previously mentioned, the strategy needs an a priori knowledge of the network to compute τ. Since this could not be the case, one can estimate or approximate it within a certain range. The plot shows that the strategy is not robust to change of the parameter and that needs to be tuned appropriately.

Example 6.2. In this example we report the H-alg's behavior corresponding to the "Smooth" switching strategy. Thanks to the combination weight, the algorithm gradually switches between the two algorithms.

Figure 2 shows three realizations corresponding to 0.1τ, τ and 10τ. The strategy, similarly to the "Hard" switch, needs, for τ's computation, an a priori knowledge. The plot highlights the performance's variability due to parameter

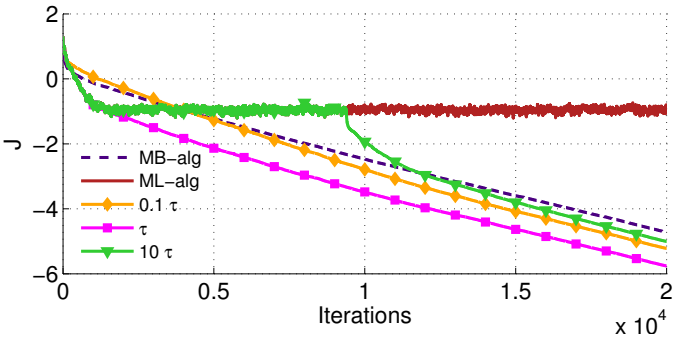

Fig. 1. Comparison between the ML-alg (red solid line), the MB-alg (blue dashed line) and the H-alg (orange-magenta-green lines) with "Hard" switching strategy

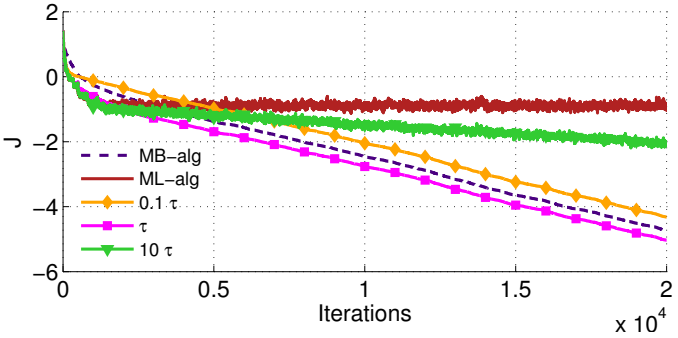

Fig. 2. Comparison between the ML-alg (red solid line), the MB-alg (blue dashed line) and the H-alg (orange-magenta-green lines) with "Smooth" switching strategy

bad approximations. Note that for the overestimated parameter 10τ the H-alg almost follows the ML-alg for too many iterations.

Example 6.3. In this example we present the last heuristic based on the "Statistical" switching strategy. We plot two different simulations each of which reporting three realizations corresponding to a parameter variation. Specifically, Figure 3 shows the realizations corresponding to $0.1(\hat{\sigma}_i/\sqrt{n})$, $\hat{\sigma}_i/\sqrt{n}$ and $10(\hat{\sigma}_i/\sqrt{n})$. In Figure 4 is reported the simulation for different dimension of the iteration window, in particular $n = 1$, 10, 100.

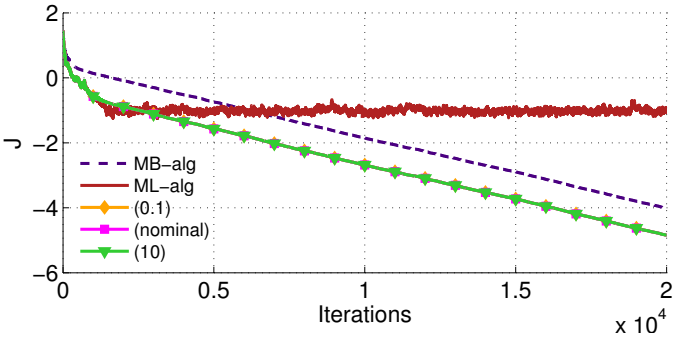

Fig. 3. Comparison between the ML-alg (red solid line), the MB-alg (blue dashed line) and the H-alg (orange-magenta-green lines) with "Statistical" switching strategy for different values of $0.1(\hat{\sigma}_i/\sqrt{n})$, $\hat{\sigma}_i/\sqrt{n}$ and $10(\hat{\sigma}_i/\sqrt{n})$

It comes clear from the first image that the realizations are almost equivalent suggesting the robustness of the strategy wrt the threshold variation. On the contrary, Figure 4

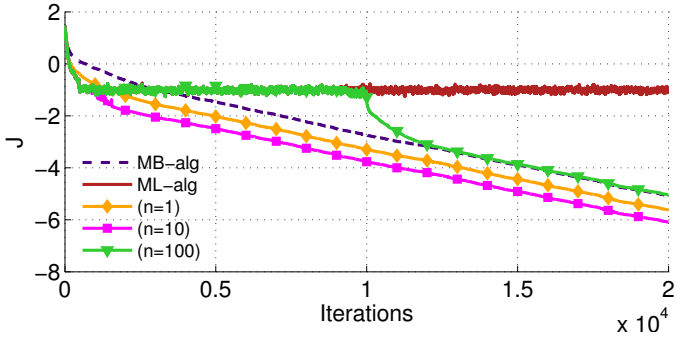

Fig. 4. Comparison between the ML-alg (red solid line), the MB-alg (blue dashed line) and the H-alg (orange-magenta-green marked lines) with "Statistical" switching strategy for $n = 1, 10, 100$.

shows that the method is affected by large values of the window' size.

Example 6.4. Finally, we present a brief comparison of the different hybrid strategies corresponding to nominal values of the parameter.

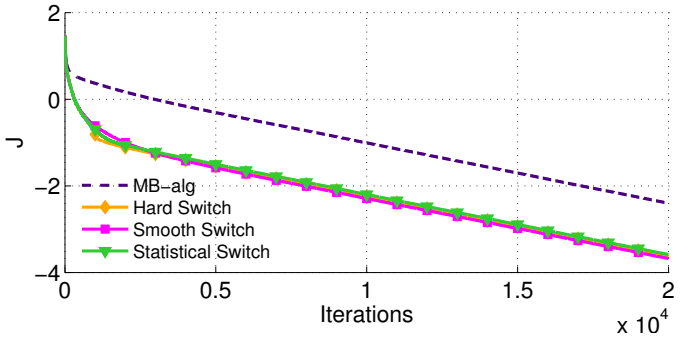

Fig. 5. Comparison between the different hybrid strategies: "Hard" (diamonds line), "Smooth" (squared line), "Statistical" (triangled line)

Figure 5 shows the averaging over 100 MonteCarlo runs. It appears that the three strategies have the same average behavior. Anyway, if there is not some a priori knowledge of the network, the "Statistical" Switching heuristic can be the best choice because it is easily tunable.

Remark 6.5. Notice that for networks with a small number of neighbours, the difference in performance between the ML-alg and the MB-alg are almost negligible and since the MB-alg converges to the optimal solution it is recommended to straightforward used the MB-alg.

7. CONCLUSION

In this work we studied the behaviour of various hybrid algorithms that combine the algorithm in Bolognani et al. [2010], also called ML-alg along the article, and the recently proposed randomized asynchronous algorithm presented in Carli et al. [2013], also called MB-alg. Three switching strategies have been proposed: two of them, heuristics (1) and (2), are easy to implement and tune, but an a priori knowledge on the communication graph is needed, to be more precise, the essential spectral radius of the communication matrix has to be known. The other strategy, (3), more complicated to implement, it is insensible to the graph parameters variations and it is not largely affected by a parameter variation. Future works are directed in proving the convergence of the hybrid algorithm with the various switching strategies.

REFERENCES

P. Barooah. *Estimation and Control with Relative Measurements: Algorithms and Scaling Laws*. PhD thesis, University of California, Santa Barbara, 2007.

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.

S. Bolognani, S. Del 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*, 20(2):176–193, 2010.

D. Borra, E. Lovisari, R. Carli, F. Fagnani, and S. Zampieri. Autonomous calibration algorithms for networks of cameras. In *Proceedings of the Americal Control Conference, ACC'12.*, July 2012.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Mathematical decomposition techniques for distributed cross-layer optimization of data networks. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

R. Carli, A. Carron, L. Schenato, and M. Todescato. An exponential-rate consensus-based algorithms for estimation from relative measurements: implementation and performance analysis. Technical report, University of Padova, March 2013. [Online] Available at `http://automatica.dei.unipd.it/people/schenato/publications.html`

Nikolaos M. Freris and Anastasios Zouzias. Fast distributed smoothing of relative measurements. In *to appear in IEEE Conference on Decision and Control (CDC 2012)*, 2012.

Bj orn Johansson, Pablo Soldati, and Mikael Johansson. Mathematical decomposition techniques for distributed cross-layer optimization of data networks. *IEEE Journal on Selected Areas in Communication*, 24(8):1535–1547, August 2006.

A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):46–61, January 2009.

M. Schwager, D. Rus, and J. J. Slotine. Decentralized, adaptive coverage control for networked robots. *International Journal of Robotics Research*, 28(3):357–375, March 2009.

R. Solis, V. Borkar, and P. R. Kumar. A new distributed time synchronization protocol for multihop wireless networks. In *45th IEEE Conference on Decision and Control (CDC'06)*, pages 2734–2739, San Diego, December 2006.

Roberto Tron and René Vidal. Distributed image-based 3-d localization of camera sensor networks. In *Proceedings of the 49th IEEE Conference on Decision and Control CDC'09*, pages 901–908, 2009.

Jing Wang and Nicola Elia. Control approach to distributed optimization. In *Annual Allerton Conference on Communication, Control, and Computing*, pages 557–561, 2010.

L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN05*, pages 63–70, 2005.

F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato. Newton-raphson consensus for distributed convex optimization. In *IEEE Conference on Decision and Control (CDC 2011)*, pages 5917–5922, 2011.