Peer-to-peer Estimation over Wireless Sensor Networks via Lipschitz Optimization [*]

Carlo Fischione
ACCESS Linnaeus Center
Royal Institute of Technology
100-44, Stockholm, Sweden
carlofi@ee.kth.se

Alberto Speranzon
United Technology Research Center
CT 06108, USA
alberto.speranzon@utrc.utc.com

Karl Henrik Johansson
ACCESS Linnaeus Center
Royal Institute of Technology
100-44, Stockholm, Sweden
kallej@ee.kth.se

Alberto Sangiovanni-Vincentelli
University of California at Berkeley
CA 94720, USA
alberto@eecs.berkeley.edu

ABSTRACT

Motivated by a peer-to-peer estimation algorithm in which adaptive weights are optimized to minimize the estimation error variance, we formulate and solve a novel nonconvex Lipschitz optimization problem that guarantees global stability of a large class of peer-to-peer consensus-based algorithms for wireless sensor network. Because of packet losses, the solution of this optimization problem cannot be achieved efficiently with either traditional centralized methods or distributed Lagrangian message passing. We prove that the optimal solution can be obtained by solving a set of nonlinear equations. A fast distributed algorithm, which requires only local computations, is presented for solving these equations. Analysis and computer simulations illustrate the algorithm and its application to various network topologies.

[*]C. Fischione and A. Speranzon acknowledge the support of the San Francisco Italian Institute of Culture by the Science & Technology Attaché T. Scapolla. The work by C. Fischione and K. H. Johansson was partially funded by the Swedish Foundation for Strategic Research, the Swedish Research Council, the Swedish Governmental Agency for Innovation System, and EU integrated project FeedNetBack. A. Sangiovanni-Vincentelli wishes to acknowledge the support of the NSF ITR CHESS, the GSRC, the COMBEST European Project and the ArtistDesign Network of Excellence.

IPSN'09, April 13–16, 2009, San Francisco, California, USA.
Copyright 2009 ACM 978-1-60558-371-6/09/04 ...$5.00.

Categories and Subject Descriptors

C.2.1 [**Computer Systems Organization**]: Computer-Communication Networks—*Network Architecture and Design.*

General Terms

Algorithms, Design, Performance, Theory.

Keywords

Lipschitz Optimization; Parallel and Distributed Computation; Wireless Sensor Networks; Distributed Estimation.

1. INTRODUCTION

Wireless sensor networks are equipped with wireless communication and sensing capabilities for communication, control and monitoring purposes, see [7] and references therein. Given the small dimensions of the sensing devices and their inaccessibility when deployed in the environment, the operations of these networks are often characterized by the absence of a central control unit and by limited communication capabilities.

The absence of central coordination is the strength of consensus-based estimation algorithms, where each node locally produces accurate estimates by filtering data received only by neighboring nodes. The challenge of these estimators is that local processing must be carefully designed to avoid local errors escalating throughout the network. For a new class of consensus-based distributed estimators [1, 4, 16, 18, 22, 19, 12, 15], the filter weights are chosen at each node so that the estimation error of the entire network is bounded. In these peer-to-peer algorithms, it is necessary to guarantee the stability of a global matrix collecting the weights that all nodes use to fuse information received from neighbors [10, 18]. This is a difficult constraint to satisfy when there is not central coordination and a cost function needs to be optimized.

The main contribution of this paper is the formulation and the development of an efficient distributed strategy to solve a novel Lipschitz optimization problem, whose solution guarantees that locally computed weights yield local estimation errors that decrease when estimates are exchanged throughout the entire network. In particular, given a network of N nodes, we show that a positive linear combination of N positive decision variables, to be maximized under N Lipschitz constraints, can be solved with a fast decentralized algorithm. We show that the optimal solution is provided by a system of nonlinear equations, and then we investigate a method to distribute quickly the computation of the optimal solution.

A strategy where the computation of the optimal solution is centralized, namely where nodes provide local information to a central unit, demands a large amount of communication resources, such as radio power, bandwidth, routing, etc., which is a major drawback. Message passing algorithms have been developed to distribute the computation by exchanging series of Lagrange multipliers associated to local constraints, see [2, 11] and references therein. However, these approaches require a large number of iterations to reach convergence. This is a major limitation, since a large amount of data exchanged by the wireless nodes has strong effects on the battery lifetime. Furthermore, the presence of packet losses may increase significantly the number of iterations.

We show that our distributed computation of the solution of the Lipschitz optimization problem provides a method to adapt the filter weights so that the estimation error variance is minimized and, at the same time, the error propagation is bounded. The main advantage of our method is that we do not need to rely on fixed (sub-optimal) Laplacian matrix associated to the communication graph or to the Metropolis weights to design the filter weights, as done in [1, 4, 22]. In contrast to our earlier work [17], where we considered the special case of packet losses i.i.d. across the network, here we assume a much more general model of the packet loss distribution. A substantial novel original analysis is therefore developed to characterize the estimator in the presence of these packet losses. Hence, we show that our method allows to build a peer-to-peer estimator that outperforms significantly other solutions from the literature even in the presence of severe packet losses.

The paper is organized as follows: In Section 2, we show that the Lipschitz optimization problem investigated here is of major relevance in distributed estimation. Section 3 presents the optimization problem in detail. In Section 4, an efficient algorithm is presented to compute the optimal solution of the optimization problem. In Section 5, we apply the algorithm to an estimation problem. We characterize in Section 5.1 performance of the estimator by incorporating the distributed algorithm developed in the previous section. Monte Carlo simulations illustrate the analysis for various packet loss probabilities in Section 6. Finally, conclusions are given in Section 7.

1.1 Notation

Given a stochastic variable x, $\mathbb{E}\,x$ denoted its expected value. $\mathbb{E}_{\,y}x$ denotes that the expected value is taken with respect to the probability density function of y. $\|\cdot\|$ denotes the ℓ^2-norm of a vector or the spectral norm of a matrix. Given a matrix $\mathbf{A}$, its largest singular value is denoted by $\gamma(\mathbf{A})$. We denote the element (i, j) of $\mathbf{A}$ with a_{ij} and with $\mathbf{a}_i$ the ith row of $\mathbf{A}$. Given the matrix $\mathbf{B}$, $\mathbf{A} \circ \mathbf{B}$ is the Hadamard (element-wise) product between $\mathbf{A}$ and $\mathbf{B}$. We denote with $\mathbf{a} \preceq \mathbf{b}$ and $\mathbf{a} \succeq \mathbf{b}$ element-wise inequalities. $\mathbf{I}$ and $\mathbf{1}$ denote the identity matrix and the vector $(1, \dots, 1)^T$, respectively. Let $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. To keep the notation light, the time dependence of the variables and parameters is disregarded when the meaning is clear from the context.

2. PRELIMINARIES

Consider a network of $N > 1$ nodes located at fixed positions. We model the network as a weighted graph. At time t, the graph is $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the vertex set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. A weighting function $\mathcal{W} : \mathcal{E} \times N_0 \to \mathbb{R} : (i, j)(t) \mapsto g_{ij}(t)$ assigns a weight to each edge of the graph. The set of neighbors of node $i \in \mathcal{V}$ plus node i is denoted as

$$\mathcal{N}_i(t) = \{j \in V : (j, i) \in \mathcal{E}\} \cup (i, i)\,.$$

Namely, $\mathcal{N}_i(t)$ represents the neighbors a node i has, including itself.

We assume that each node of the network updates its state $z_i(t)$ at time $t \in \mathbb{N}_0$ by a linear combination of states and inputs of the neighboring nodes:

$$z_i(t) = \sum_{j \in \mathcal{N}_i(t)} k_{ij}(t)\phi_{ij}(t) z_j(t-1) + \sum_{j \in \mathcal{N}_i(t)} h_{ij}(t)\phi_{ij}(t) u_j(t)\,, \tag{2.1}$$

where $k_{ij}(t)$ and $h_{ij}(t)$ are weighting coefficients and $\phi_{ij}(t)$ is a binary random variable describing the packet loss process:

$$\begin{aligned}
&\Pr(\phi_{ij}(t) = \boldsymbol{\varphi}_{ij|t} = 1) = p_{ij}\,,\\
&\Pr(\phi_{ij}(t) = \boldsymbol{\varphi}_{ij|t} = 0) = 1 - p_{ij} = q_{ij}\\
&\Pr(\phi_{ii}(t) = \boldsymbol{\varphi}_{ii|t} = 1) = 1\,.
\end{aligned}$$

The probability p_{ij} denotes the successful packet reception at the receiver of the link from node i to node j. Thus a node j is a neighbor of i by sending successfully ($\boldsymbol{\varphi}_{ij|t} = 1$) its state $z_j(t-1)$ and input $u_j(t)$. Note that $\boldsymbol{\varphi}_{ij|t}$ is a realization of the packet loss process $\phi_{ij}(t)$. Eq. (2.1) can be written for all nodes of the network as

$$\mathbf{z}(t) = (\mathbf{K}(t) \circ \mathbf{\Phi}(t))\,\mathbf{z}(t-1) + (\mathbf{H}(t) \circ \mathbf{\Phi}(t))\,\mathbf{u}(t)\,, \tag{2.2}$$

where $\mathbf{z}(.) \in R^N$, $\mathbf{K}(t) = [k_{ij}] \in R^{N\times N}$, $\mathbf{H}(t) = [h_{ij}] \in R^{N\times N}$, and $\mathbf{\Phi}(t) = [\psi_{ij}(t)] \in R^{N\times N}$.

The difference equations in (2.2) are common in the area of distributed consensus [5, 10] or distributed estimation and data fusion [4, 18, 23]. In particular, Eq. (2.2) is used in average consensus problems [10, 21], when $\mathbf{H}(t) = 0$, or when $\mathbf{H}(t) = \mathbf{I}$ and u_{ij} are Gaussian zero mean i.i.d. random variables [23].

Eq. (2.2) models also consensus-based estimation of a common scalar signal $d(t)$, where $\mathbf{u}(t)$ assumes the meaning of the input vector of noisy measurements [4, 14, 18]. In these estimators, $\mathbf{K}(t)$ is designed such that the expected estimation error $\mathbf{e}(t) = \mathbf{z}(t) - d(t)\mathbf{1}$ is bounded

$$\lim_{t\to+\infty} \|\,\mathbb{E}\,\mathbf{e}(t)\| \leq \alpha\,, \tag{2.3}$$

where $\alpha \geq 0$. It is possible to show that the state converges to a neighborhood of the origin if $((\mathbf{K}(t)+\mathbf{H}(t)-\mathbf{I})\circ\mathbf{\Phi}(t))\mathbf{1} = 0$ and $\|\mathbf{K}(t)\circ\mathbf{\Phi}(t)\| < 1$ for any packet loss realization [18, 6]. Therefore, accurate estimates can be achieved by solving the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{K}(t)} \quad & \mathbb{E}\,\mathbf{e}(t)^T\mathbf{e}(t) && (2.4)\\
\text{s.t.} \quad & ((\mathbf{K}(t) + \mathbf{H}(t) - \mathbf{I}) \circ \mathbf{\Phi}(t))\,\mathbf{1} = 0\\
& \|\mathbf{K}(t) \circ \mathbf{\Phi}(t)\| \leq \gamma_{\max} < 1
\end{aligned}$$

This optimization problem cannot be solved by a central coordination unit, because we are assuming that no such unit is available for the network. Besides, if a central unit were used, it should have been able to know all information in the network at each time instant, which is not possible because of packet losses. Therefore, the problem must be solved by a distributed strategy.

The objective functions and the first constraint in problems (2.4) can be easily distributed, since the former is given by a sum of node's estimation variance, whereas the elements of the constraint vector include only local information. By contrast, the last constraint is difficult to distribute efficiently among the nodes. One could use the max-norm or the infinity-norm, which give simple local conditions to ensure global stability. Unfortunately, these norms make the optimization problem (2.4) infeasible. As a result, to ensure that a matrix has a bounded norm by local conditions is a challenging task. We approach this problem in the following.

3. A LIPSCHITZ OPTIMIZATION PROBLEM

Let us define the set $\Theta_{\boldsymbol{\varphi}_i} = \{j \neq i : \mathcal{N}_i(t) \cup \mathcal{N}_j(t) \neq \emptyset\}$, for $i = 1, \ldots, N$, where $\boldsymbol{\varphi}_i$ is a realization at time t of the process $\boldsymbol{\phi}_i(t)$. The set $\Theta_{\boldsymbol{\varphi}_i}$ is the collection of communicating subsystems located at two-hops distance from the subsystem i, plus the neighbors of i, at time t. Then we have the following result.

PROPOSITION 3.1 ([18]). *Let $\mathbf{K} = [\mathbf{k}_i] \in \mathbb{R}^{N\times N}$, where $\mathbf{k}_i \in \mathbb{R}^{1\times N}$. Let $0 < \gamma_{\max} < 1$. Suppose there exists a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)^T \succ 0$, such that*

$$x_i + \sqrt{x_i} \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} \sqrt{x_j} \leq \gamma_{\max}\,, \tag{3.1}$$

for all $i = 1, \ldots, N$. If $\|\mathbf{k}_i\|^2 \leq x_i$, $i = 1, \ldots, N$, then $\|\mathbf{K}\|_2 = \gamma(\mathbf{K}) \leq \gamma_{\max}$.

This proposition suggests that, given some thresholds $x_i > 0$ satisfying a set of non-linear inequalities, then as long as the norms of the rows $\mathbf{k}_i$ are not above the thresholds x_i, for $i = 1, \ldots, N$, the matrix $\mathbf{K}$ is stable. Obviously, the condition on $\|\mathbf{k}_i\|^2 \leq x_i$ leaves much freedom in choosing the single elements in the vector $\mathbf{k}_i$. It can be shown that the estimation cost function of problem (2.4) decreases as $\|\mathbf{k}_i\|$ increases. Therefore, we need to solve the following problem:

$$\begin{aligned}
\max_{\boldsymbol{x}} \quad & \mathbf{1}^T\boldsymbol{x} && (3.2)\\
\text{s.t.} \quad & x_i + \sqrt{x_i} \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} \sqrt{x_j} \leq \gamma_{\max} \quad i = 1, \ldots, N && (3.3)\\
& \boldsymbol{x} \succ 0\,.
\end{aligned}$$

Such a problem is non-linear and non-convex. Writing the constraints in the canonical form, it becomes a Lipschitz optimization problem [9]. No closed form solution is available. The solution can be computed via standard centralized numerical approaches, but the presence of packet losses introduces a very large delay as information must be transmitted from all the nodes to a central one, which would compute the solution and send it back to local nodes. We show in the following that by exploiting that the problem is Lipschitz, it is possible to find a decentralized algorithm to compute the solution to the problem.

4. OPTIMAL SOLUTION

The distributed computation of the solution of (3.2) could be performed by parallelization and decomposition techniques, as in [2]. However, the convergence speed may be prohibitive, particularly due to the presence of packet losses.

The fact that in (3.3) the only information from two-hop neighboring nodes is required, and not of the entire network, allows us to develop a decentralized algorithm to compute the optimal solution. This is obtained in two steps. First we show that the optimal solution satisfies the inequality constraints with equality. Second, we build on this to distribute the computation among nodes to obtain the optimal solution. We provide details in the sequel.

4.1 Equality constraints

In this section, we show that there is a global optimal solution of (3.2) that satisfies the inequality constraints (3.3) with equality. In particular we have the following important result:

THEOREM 4.1. *Problem* (3.2) *admits a global optimum* $\boldsymbol{x}^*$*, which is the solution of the following set of nonlinear equations:*

$$x_i^* + \sqrt{x_i^*} \sum_{j \in \Theta_{\boldsymbol{\varphi}_i}} \sqrt{x_j^*} - \gamma_{\max} = 0 \,, \qquad (4.1)$$

with $i = 1, \ldots, N$ *and* $\Theta_{\boldsymbol{\varphi}_i} = \{j \neq i : \mathcal{N}_i \cup \mathcal{N}_j \neq \emptyset\}$.

PROOF. See Appendix A.1. □

We use Theorem 4.1 in the next sections to develop a strategy for the distributed computation of the optimal solution.

4.2 Distribution of the Computation

From the previous section, we see that the thresholds required in Proposition 3.1 are the solution of the set of nonlinear equations (4.1). Unfortunately, an explicit solution is not available. Numerical techniques have to be used. In the following, we present a quick decentralized algorithm with guaranteed convergence.

Let $y_i^2 = x_i$ for $i = 1, \ldots, N$. Define the class of functions parameterized in the scalar $\beta_i > 0$, $i = 1, \ldots, N$,

$$f_i(\mathbf{y}) = y_i - \beta_i \left(y_i^2 + y_i \sum_{j \in \Theta_{\boldsymbol{\varphi}_i}} y_j - \gamma_{\max} \right) \,, \qquad (4.2)$$

and let $\mathbf{f}(\mathbf{y}) = (f_1(\mathbf{y}), \ldots, f_N(\mathbf{y}))^T$. Note that the solution $\mathbf{y}^*$ to the system of nonlinear equations $\mathbf{y} = \mathbf{f}(\mathbf{y})$ is related to the solution of the system (4.1) by ${y_i^*}^2 = \psi_i^*$, as explained in the proof of Lemma (A.2). When $\mathbf{f}(\mathbf{y})$ is contractive, then it is easy to show that the fixed point of the mapping $\mathbf{y}(k+1) = \mathbf{f}(\mathbf{y}(k))$ is the solution of (4.1) [2, Pag.191]. Furthermore, the convergence speed can be tuned at a local node i by the parameter β_i. The following result provides us with the best β_i:

PROPOSITION 4.2. *Let*

$$\beta_i^*(k) = \frac{2y_i(k) + \sum_{j \in \Theta_{\boldsymbol{\varphi}_i}} y_j(k)}{\sum_{j \in \Theta_{\boldsymbol{\varphi}_i}} y_j^2(k) + \Big(2y_i(k) + \sum_{j \in \Theta_{\boldsymbol{\varphi}_i}} y_j(k)\Big)^2} \,. \qquad (4.3)$$

Then $y_i(k+1) = f_i(\mathbf{y}(k))$ *is a contraction map having the largest convergence speed among the mappings* (4.2) *with respect to the* 2*-norm.*

PROOF. See Appendix A.2. □

From previous proposition, the overall mapping $\mathbf{y} = \mathbf{f}(\mathbf{y})$ is a contraction. The component solution method [2, Pag.187] can be applied, so that the solution of (3.2) is given by the algorithm

$$y_i(k+1) = f_i(\mathbf{y}(k)) \,. \qquad (4.4)$$

Using the $\beta_i^*(k)$ given by Proposition 4.2, the mapping converges quickly. From Monte Carlo simulations, discussed in Section 6, we observed that the algorithm converges in about $5 - 10$ iterations.

4.3 Computation of the Thresholds

The distributed computation of the thresholds x_i requires that the neighboring nodes communicate the instantaneous values of the local threshold, until (4.4) converges. Clearly, the thresholds are transmitted over the same channel used for broadcasting the nodes' state and inputs, and thus they are subject to packet losses. These losses may happen during the phase between the beginning of the iterations (4.4) and the convergence. As a result, no convergence may be reached. In the following, we develop a strategy to cope with this problem.

First, notice that the optimization problem is not sensitive to perturbations of the constraints. In other words, if $\boldsymbol{x}^*$ is the solution of the system of non-linear equations (4.1), then $\boldsymbol{x}^*$ is not significantly perturbed by packet losses. We can see this from the proof of Theorem 4.1, form where we know that the optimal solution is such that $\mathbf{J}(\boldsymbol{x}^*)^T \boldsymbol{\xi}^* = \mathbf{1}$, with $\mathbf{J}(\boldsymbol{x}^*)$ being the Jacobian of the constraints and $\boldsymbol{\xi}^*$ the Lagrange multipliers associated to the dual problem of (3.2). Specifically, the i-th equation of $\mathbf{J}(\boldsymbol{x}^*)^T \boldsymbol{\xi}^* = \mathbf{1}$ is given by

$$\xi_i^* \left(1 + \frac{1}{2\sqrt{x_i^*}} \sum_{j \in \Theta_{\boldsymbol{\varphi}_i}} \sqrt{x_j^*} \right) + \sum_{j \in \Theta_{\boldsymbol{\varphi}_i}} \xi_j^* \frac{\sqrt{x_j^*}}{2\sqrt{x_i^*}} = 1 \,. \qquad (4.5)$$

It follows that $\xi_i^* < 1$ for $i = 1, \ldots, N$, because such a system of equations has positive coefficients, $\boldsymbol{\xi}^* \succeq 0$

(for strong duality holds), and the coefficient of ξ_i^* in Eq. (4.5) is strictly greater than 1. Then, $\boldsymbol{\xi}^* < 1$ implies that the optimal solution is not sensitive to perturbations of the constraints [3, pag. 249].

Since a change in the number of two-hops neighbors of a node, caused by packet losses, can be regarded as a perturbation of the constraints, we conclude that the optimal solution of the problem (3.2) is not much sensitive to the packet losses. By this argument, we can compute just once the optimal solution. In particular, we assume that the nodes compute the optimal thresholds before the updating (2.1) starts by considering the maximum number of neighbors. This is accomplished by using high transmission radio powers and a retransmission protocol that guarantee a successful packet reception. Such a preliminary phase is very short, since from Proposition 4.2 the computation of the thresholds according to (4.4) requires few iterations to converge. During the estimation phase, if the packet loss probability is very high, the perturbation might be large, resulting in a significant change of the optimum. However, simulations reported in Section 6 show that the solution we adopt for the threshold computation is robust to rather intense packet losses.

5. APPLICATION: PEER-TO-PEER ESTIMATION

We show in the following that having a distributed algorithm for the solution of the Lipschitz optimization problem (3.2) is instrumental for designing a peer-to-peer accurate estimator of time-varying signals.

Let us consider the problem of estimating a scalar signal $d(t)$ from noisy measurements

$$u_i(t) = d(t) + v_i(t)\,, \quad t \in \mathbb{N}_0$$

for all $i = 1, \dots, N$. We assume that $v_i(t) \sim \mathcal{N}(0, \sigma^2)$ for all i and that $\mathbb{E}\, v_i(t) v_j(t) = 0$ for all $t \in \mathbb{N}_0$, and that $|d(t) - d(t-1)| \le \Delta$. We remark here that we do not assume any model of the signal to track, in contrast to, e.g., [13]. No central coordination point is present either, in contrast to [20], since we are interested in peer-to-peer solutions.

We consider an estimator where each node i computes an estimate $z_i(t)$ of $d(t)$ by taking a linear combination of its own and of its neighbors' estimates and measurements, as described in Eq. (2.1). Defining the estimator error as $\mathbf{e}(t) = \hat{\mathbf{d}}(t) - d(t)\mathbf{1}$ we have that its dynamics are described by

$$\begin{aligned} \mathbf{e}(t) &= (\mathbf{K}(t) \circ \mathbf{\Phi}(t))\, \mathbf{e}(t-1) + (\mathbf{H}(t) \circ \mathbf{\Phi}(t))\, \mathbf{u}(t) \\ &\quad + (\mathbf{K}(t) \circ \mathbf{\Phi}(t))\, d(t-1)\mathbf{1} - d(t)\mathbf{1}\,. \end{aligned} \tag{5.1}$$

Under the conditions that $((\mathbf{K}(t) + \mathbf{H}(t) - \mathbf{I}) \circ \bar{\Phi}(t))\mathbf{1} = 0$ for any realization $\bar{\Phi}(t)$ of the stochastic process $\Phi(t)$, we obtain

$$\mathbb{E}_v \mathbf{e}(t) = (\mathbf{K}(t) \circ \mathbf{\Phi}(t))\, \mathbb{E}_v \mathbf{e}(t-1) - \delta(t)(\mathbf{K}(t) \circ \mathbf{\Phi}(t))\mathbf{1}\,. \tag{5.2}$$

To design a minimum variance estimator of $d(t)$ we need to impose that the estimation error converges, which is ensured if $\|K(t)\| \le \gamma_{\max} < 1$ [6].

The optimal choice of the filter coefficients are given by solving the optimization problem (2.4). As a consequence of the main results of this paper, Theorem 4.1 and the distributed algorithms described in Section 4, we can replace the global constraint (2.4) with a local one, given by $\|\mathbf{k}_i \circ \boldsymbol{\phi}_i\|^2 \le x_i$. The value of x_i is obtained by the distributed iterations presented in Section 4.3. Therefore, the global optimization problem can be decomposed into local optimization problems:

$$\begin{aligned} \min_{\mathbf{k}_i(t), \mathbf{h}_i(t)} \quad & \mathbf{k}_i^T(t) \left(\mathbf{P}_i(t-1) \circ (\boldsymbol{\varphi}_{i|t} \boldsymbol{\varphi}_{i|t}^T) \right) \mathbf{k}_i(t) \\ & + \sigma^2 \mathbf{h}_i^T(t) \boldsymbol{\varphi}_{i|t} \boldsymbol{\varphi}_{i|t}^T \mathbf{h}_i(t) \qquad (5.3) \\ \text{s.t.} \quad & \left((\mathbf{k}_i(t) + \mathbf{h}_i(t))^T \circ \boldsymbol{\varphi}_{i|t} \right) \mathbf{1} = 1 \\ & \|\mathbf{k}_i(t) \circ \boldsymbol{\varphi}_{i|t}\|^2 \le \boldsymbol{x}_i\,. \end{aligned}$$

where $\bar{\mathbf{\Phi}}(t)$ is a realization of the packet loss process $\mathbf{\Phi}(t)$ and $P(t-1) = \mathbb{E}\,(\mathbf{e}(t-1) - \mathbb{E}\,\mathbf{e}(t-1))(\mathbf{e}(t-1) - \mathbb{E}\,\mathbf{e}(t-1))^T$.

The optimization problem 5.3 is a Quadratically Constrained Quadratic Problem [3]. It can be numerically solved efficiently as shown in [6]. Therefore, the optimal local weights $\mathbf{k}_i(t)$ and $\mathbf{h}_i(t)$ that minimize the estimation error variance at each time instant can be computed locally at each node:

PROPOSITION 5.1. *For a given covariance matrix* $\mathbf{P}_i(t-1)$ *and a realization* $\boldsymbol{\varphi}_{i|t}$ *of* $\boldsymbol{\phi}_i(t)$*, the values of* $\mathbf{k}_i(t)$ *and* $\mathbf{h}_i(t)$ *that minimizes* (5.3) *are*

$$\mathbf{k}_i(t) = \tag{5.4}$$

$$\frac{\left((\mathbf{P}_i(t-1) + \lambda_i(t)\mathbf{I}) \circ \boldsymbol{\varphi}_{i|t} \boldsymbol{\varphi}_{i|t}^T \right)^\dagger \boldsymbol{\varphi}_{i|t}}{\boldsymbol{\varphi}_{i|t}^T \left(\left((\mathbf{P}_i(t-1) + \lambda_i(t)\mathbf{I}) \circ \boldsymbol{\varphi}_{i|t} \boldsymbol{\varphi}_{i|t}^T \right)^\dagger + \sigma^{-2}\mathbf{I} \right) \boldsymbol{\varphi}_{i|t}}\,,$$

$$\mathbf{h}_i(t) = \tag{5.5}$$

$$\frac{\left(\boldsymbol{\varphi}_{i|t} \boldsymbol{\varphi}_{i|t}^T \right)^\dagger \boldsymbol{\varphi}_{i|t}}{\sigma^2 \boldsymbol{\varphi}_{i|t}^T \left(\left((\mathbf{P}_i(t-1) + \lambda_i(t)\mathbf{I}) \circ \boldsymbol{\varphi}_{i|t} \boldsymbol{\varphi}_{i|t}^T \right)^\dagger + \sigma^{-2}\mathbf{I} \right) \boldsymbol{\varphi}_{i|t}}\,,$$

with the optimal Lagrange multiplier

$$\lambda_i(t) \in \left[0, \max\left(0, \sigma^2 / \sqrt{|\mathcal{N}_{\boldsymbol{\varphi}_i}| \psi_i(t)} - \ell_m(\mathbf{\Gamma}_i(t-1)) \right) \right]\,.$$

PROOF. The proof is similar to the proof of Proposition III.2 in [18]. □

Remark 5.2. *The modeling of the packet losses by the Hadamard product allows us to obtain weights having a similar form to those we obtained in the case of no packet loss [18]. However, this result is not a straightforward application of [18] because* (5.4) *and* (5.5) *are obtained by exploitation of the Hadamard product and the Moore-Penrose pseudo-inverse in the computation of the Lagrange dual function and the KKT conditions. Therefore, the previous proposition generalizes our earlier result for any given realization of the packet loss process. In the special case when $\boldsymbol{\varphi}_{i|t} = \mathbf{1}$, namely when there are no packet losses, we reobtain the result in [18].*

Previous proposition provides us with an interval within which the optimal λ_i is located. Simple search algorithms can be considered to solve numerically the KKT condition $(\mathbf{k}_i(t) \circ \boldsymbol{\varphi}_{i|t})^T(\mathbf{k}_i(t) \circ \boldsymbol{\varphi}_{i|t}) - \psi_i = 0$ for λ_i, such as, for example, the bisection algorithm.

We can now summarize the analysis so far developed and describe the estimator: At time t, node i makes a measurement $u_i(t)$, receives estimates and measurements that neighboring nodes send successfully, and builds the estimate by Eq. (2.1). In such an equation, node i uses the coefficients $\mathbf{k}_i(t)$ and $\mathbf{h}_i(t)$ given by Proposition 5.1 and the thresholds $\boldsymbol{x}$ computed by algorithm (4.4) as descried in Subsection 4.3. We recall that algorithm (4.4) requires simple calculations, whereas the matrix inversions needed in Proposition 5.1 can be easily computed either by pre-stored equations or by well-known numerical algorithm [2], by considering that the number of neighboring nodes is not large.

Performance of the estimator described above is characterized in the next subsection.

5.1 Performance Analysis

In this section we characterize the performance of our peer-to-peer estimator by investigating the variance of the estimation error. We have the following results:

Proposition 5.3. *For any packet loss realization $\boldsymbol{\varphi}_{i|t}$ of $\boldsymbol{\phi}_i(t)$, the optimal value of $\mathbf{k}_i(t)$ and $\mathbf{h}_i(t)$ are such that the error variance at node i satisfies*

$$\mathbb{E}_{\,v}(e_i^2 - \mathbb{E}\, e_i^2 | \boldsymbol{\phi}_i(t) = \boldsymbol{\varphi}_{i|t})^2 < \frac{\sigma^2}{|\mathcal{N}_i|}\,.$$

Proof. See [6]. □

Notice that previous proposition guarantees that the instantaneous estimation error in each node is always upper-bounded by the variance of the estimator that just takes the averages of the received $u_i(t)$.

The previous results can be made more tight and dependent directly on the packet losses by taking the average over the packet loss distribution. To show this, we need an intermediate technical result:

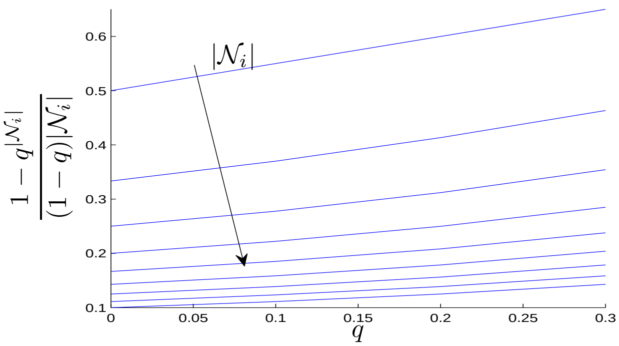


Figure 1: Eq. (5.9) (which is the second factor of (5.8) for packet losses i.i.d.) as function of q for increasing values of $|\mathcal{N}_i|$ ranging from 2 to 10. The factor is always less than 1. The smallest values are achieved when q is small and $|\mathcal{N}_i|$ is large. This is explained by that the packet loss probability has a decreasing negative effect when the number of neighbors of a node increases, which translates into a smaller value of the coefficient.

Lemma 5.4.

$$\mathbb{E}_{\,\boldsymbol{\phi}}[\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i]^{-1} = \sum_{k=0}^{|\mathcal{N}_i|-1} \frac{\chi(k)}{k+1}\,, \tag{5.6}$$

where

$$\chi(k) = \sum_{\ell=1}^{\binom{|\mathcal{N}_i|-1}{k}} \left(\prod_{n=1}^{k} q_{is(n)} \times \prod_{m=k+1}^{|\mathcal{N}_i|-1} p_{is(m)} \right)\,, \tag{5.7}$$

and the function $s: \{1, 2, \ldots, |\mathcal{N}_i|-1\} \to \{1, 2, \ldots, |\mathcal{N}_i|-1\}$ is a permutation. Namely the k-th coefficient of the polynomial is the sum of $\binom{|\mathcal{N}_i|-1}{k}$ terms in which there are k factors q_{ij} and $|\mathcal{N}_i| - 1 - k$ factors p_{ir} with $j \neq r$.

Proof. See [6]. □

Proposition 5.5. *It holds*

$$\mathbb{E}_{\,\boldsymbol{\phi}}\,\mathbb{E}_{\,v}(e_i^2 - \mathbb{E}_{\,v} e_i^2) \leq \frac{(\sqrt{5}-1)\sqrt{\gamma_{\max}} + 2N}{2(\sqrt{5}-1)\sqrt{\gamma_{\max}} + 2N} \sum_{k=0}^{|\mathcal{N}_i|-1} \frac{\chi(k)}{k+1}\,\sigma^2\,. \tag{5.8}$$

Proof. See Appendix A.3. □

Observe that the estimation error variance given by the previous proposition depends on the packet loss probabilities q_{ij}, on the maximum number of neighbors for each node $|\mathcal{N}_i|$, the total number of nodes in the networks N, and the largest singular value of the matrix $\mathbf{K}(t)$. The first factor of the coefficient of (5.8) is always less than 1. The smallest values are achieved when $\gamma_{\max}$ is large and N small. The second factor in (5.8) depends clearly on the value attained by the various q_{ij}. If we consider the simple case when $q_{ij} = q$ for all i, j, then

$$\sum_{k=0}^{|\mathcal{N}_i|-1} \frac{\chi(k)}{k+1} = \frac{1-q^{|\mathcal{N}_i|}}{(1-q)|\mathcal{N}_i|}\,. \tag{5.9}$$

This function decreases very fast as the maximum number of neighbors of a node increases, for all values of q, as we show in Fig 1. This is rather intuitive, since as the number of neighbors increases packet losses have less impact on the estimation and thus better performance are achieved. Notice also that the value of the function (5.9) for $q = 0$ is $1/|\mathcal{N}_i|$. Thus in presence of non-identical packet loss probabilities the degradation in performance is not remarked. In particular even when the first factor of (5.8) is very close to 1 with a packet loss of $q = 0.3$ we have that the product of the two coefficients does not exceed 0.65 and it is just 30% higher than the case when no packet losses are present.

Corollary 5.6. *Consider as benchmark the estimator computing the estimates by the instantaneous average of the available measurements, namely the estimator for which the weights are chosen to be* $k_{ij} = 0$ *and* $h_{ij} = 1/|\mathcal{N}_i|$, *for all* $i = 1, \dots, N$, *and* $j \in \mathcal{N}_i$. *Then,* $\lim_{t\to+\infty} \mathbb{E}_v e_i(t) = 0$ *and the variance is*

$$\mathbb{E}_\phi \mathbb{E}_v e_i^2 = \mathbb{E}_\phi \frac{\sigma^2}{\boldsymbol{\phi}_i^T \boldsymbol{\phi}_i} = \sigma^2 \sum_{k=0}^{|\mathcal{N}_i|-1} \frac{\chi(k)}{k+1}\,. \tag{5.10}$$

From this corollary we see that the difference in the expected performance between the proposed estimator, given by (5.8), and the unbiased estimator that does an arithmetic average, given by (5.10), is on the first coefficient of (5.8). Clearly, the proposed estimator outperforms the latter, since the first factor in (5.8) is always less than one.

6. SIMULATIONS AND NUMERICAL RESULTS

In this section we illustrate the theoretical analysis carried out in the previous sections, and show the benefits of the distributed computation of the Lipschitz optimization problem.

An example of the distributed computation as obtained with mapping (4.2) is reported in Fig. 2 for $N = 10$ nodes, with an average number of 5 neighbors per node. The algorithm converges quickly. Specifically, we assumed that convergence is reached when $|x_i(k+1) - x_i(k)| \leq \varpi$, with ϖ being the desired accuracy. From Monte Carlo simulations, we observed that the algorithm converges in about $5 - 10$ iterations on average by setting $\varpi = 10^{-8}$, which is a quite small value if compared to the order of magnitude of the optimal solution (10^{-2}). We remark that in general the worst case convergence behavior is dependent on the average connectivity of the network and it does not depend on the total number of nodes of the network. The reason is that the convergence speed is given by the Lipschitz constant of the mapping (4.2), (see the proof of Proposition 4.2 and [2] for further details). Such a constant depends on the local connectivity $\Theta_{\boldsymbol{\varphi}_i}$, and since $0 \prec \boldsymbol{x} \prec \mathbf{1}$ regardless the number of nodes of the network, it follows that it can be upper bounded by just a function of the local connectivity. It is also interesting to observe that possible rounding errors in the computations have a small effect since these errors can be modelled by constraints perturbations, and the Lipschitz optimization problem investigated in this paper is not sensitive to perturbations, as studied in Section 4.3.

In Fig. 3, we report the test signals $d_1(t), \dots, d_4(t)$ to estimate, which are used to assess our estimator. We compared our estimator with three other solutions. We

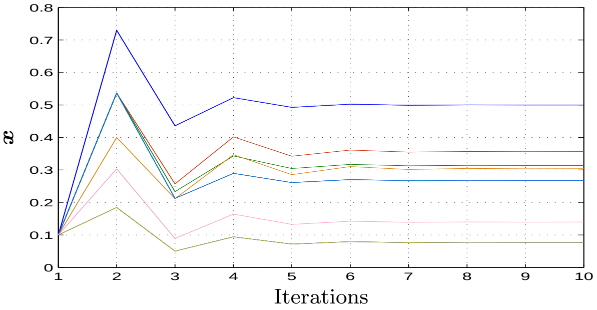


Figure 2: Qualitative convergence behavior of the contraction mapping (4.2) for $N = 10$ nodes. The convergence is reached quickly, in this case with less than 6 iterations. The iterations are initialized with $1/N$.

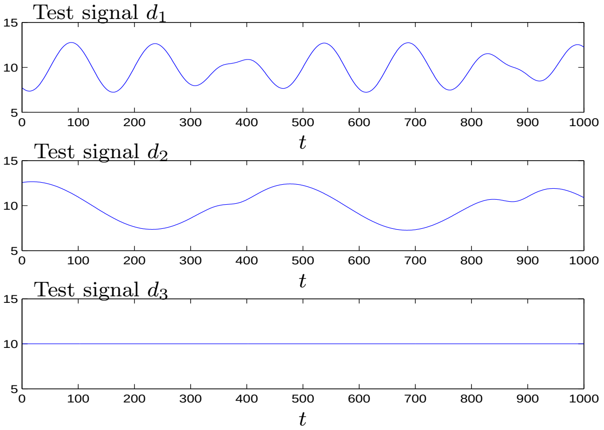


Figure 3: Test signals used to compare performance with various estimator. The signals are generated accordingly to $d_i(t) = 3(\sin(2\pi\omega_i t/300)) + 10 - 0.5\sin(2\pi\omega_i t/300)(1-\exp(-t/300))/(\sin(2\omega_i\pi t/500) + 1.2)$ with $\omega_1 = 1$, $\omega_2 = 0.5$ and $\omega_3 = 0$.

considered the estimator that computes the average of the measurements received at each node (which we define estimator E_1), the estimator that uses Eq. (2.1) with coefficients given by the weights associated to the Laplacian of the graph (which we define estimator E_2), and our peer-to-peer estimator (which we define estimator E_p). In the simulations, we set $\gamma_{\max} = 0.995$, $\Delta = 3.25 \times 10^{-2}$, and $\sigma^2 = 1.5$. These values are chosen so that the noise variance is very high if compared to the signal to estimate. The value of Δ used in the computation of $\gamma_{\max}$ was considered about 3% larger then the real value, as to simulate an imperfect a-priori knowledge on the signal to estimate.

We first considered the case when packet losses are i.i.d. processes. Fig. 4 shows the Mean Square Error (MSE) for the three estimators under consideration, E_1, E_2 and E_p, for a network with 10 nodes and for four packet loss probabilities: $q = 0\%$, $q = 10\%$, $q = 20\%$ and $q = 30\%$. As a performance metric, we used the average and variance over 30 simulation of the relative MSE:

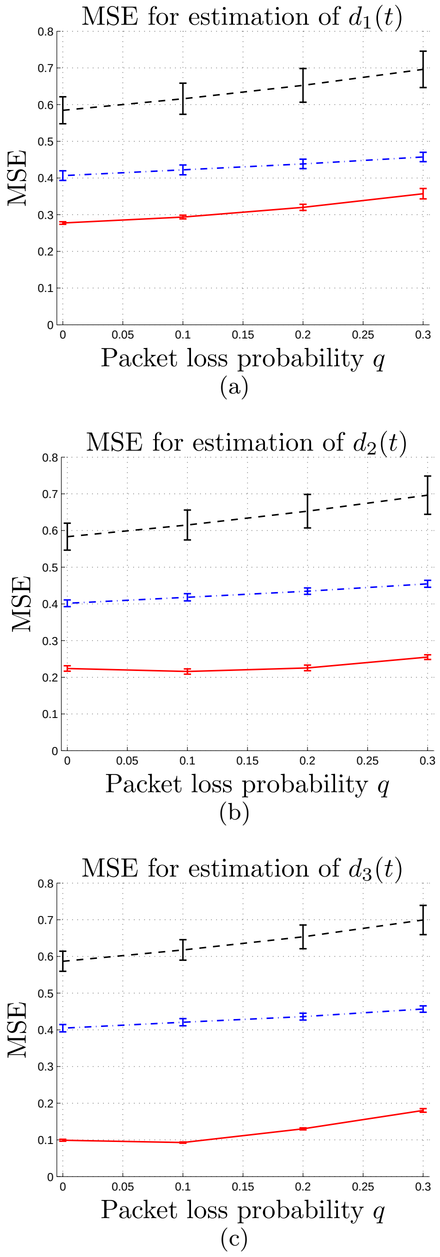
$$\text{MSE}_{rel} = \frac{\text{MSE}(E_i) - \text{MSE}(E_p)}{\text{MSE}(E_i)}\,.$$

As it can be seen the proposed estimator outperforms the other ones. We remark that as the packet loss rate grows, performance of E_p approaches E_2, but it is always better for packet losses below 50%. Notice also that when the signal is faster then MSE is higher since Δ is larger.

Fig. 5 shows the estimates computed by 30 nodes for three different estimators with non-i.i.d. packet loss probabilities $q_{ij} = 0\%, 10\% \pm 5\%, 20\% \pm 5\%$ and $30\% \pm 5\%$. The first plot shows the actual measurements, the second shows the estimates computed by estimator E_1, the third plot shows the estimates computed estimator E_2, and the last plot shows the estimates obtained by our estimator E_p. The simulations reported in Fig. 5 are obtained with $q_{ij} = 20\% \pm 5\%$ and the signal $d_2(t)$, shown in Fig. 3, has to be tracked. By using our estimator, nodes are able to reconstruct $d_2(t)$ with very low error even in presence of high measurement noise and with a high packet loss probability. Similar results as those obtained for the estimation of $d_2(t)$ are achieved for the estimation of $d_1(t)$ and $d_3(t)$. The difference is that the quality of the estimator is slightly reduced for $d_1(t)$, as already discussed about Fig. 4. Analogously, if we had used the signal $d_3(t)$, we would have a better quality of the estimates.

Numerical results obtained when packet losses are non-i.i.d. random variables are collected in Tab. 1. We see that our peer-to-peer estimator outperforms significantly the other solutions in all considered cases.

We carried out numerical simulations to show how the distributed solution of the Lipschitz optimization prob-



(a)

(b)

(c)

Figure 4: Mean Square Error (MSE) performance comparison among estimators for various i.i.d. packet loss probabilities q for a network with $N = 10$ nodes. Each plot is associated to one of the three test signals $d_1(t), \ldots, d_3(t)$, see Fig. 3. The dashed line refers to the estimator E_1, the dashed-dotted line refers to the estimator E_2 and the solid line to the proposed estimator E_p. The vertical bars represent the variance of the MSE computed for the 20 simulations.

Estimator Type	MSE_{rel}							
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
	Signal test d_1							
	$q_{ij} = 0\%$		$q_{ij} = 10\% \pm 5\%$		$q_{ij} = 20\% \pm 5\%$		$q_{ij} = 30\% \pm 5\%$	
Average of measurements (E_1)	0.596	0.183	0.628	0.201	0.665	0.214	0.710	0.236
Laplacian based (E_2)	0.478	0.161	0.434	0.128	0.452	0.102	0.472	0.083
Proposed Estimator (E_p)	0.277	0.045	0.295	0.048	0.323	0.065	0.362	0.081
	Signal test d_2							
	$q_{ij} = 0\%$		$q_{ij} = 10\% \pm 5\%$		$q_{ij} = 10\% \pm 5\%$		$q_{ij} = 10\% \pm 5\%$	
Average of measurements (E_1)	0.596	0.146	0.628	0.159	0.665	0.177	0.709	0.201
Laplacian based (E_2)	0.414	0.167	0.431	0.128	0.449	0.102	0.469	0.082
Proposed Estimator (E_p)	0.214	0.077	0.215	0.058	0.226	0.042	0.260	0.067
	Signal test d_3							
	$q_{ij} = 0\%$		$q_{ij} = 10\% \pm 5\%$		$q_{ij} = 10\% \pm 5\%$		$q_{ij} = 10\% \pm 5\%$	
Average of measurements (E_1)	0.596	0.149	0.631	0.167	0.667	0.178	0.711	0.193
Laplacian based (E_2)	0.415	0.159	0.432	0.119	0.449	0.097	0.469	0.076
Proposed Estimator (E_p)	0.098	0.041	0.087	0.049	0.126	0.044	0.174	0.045

Table 1: Comparison of the performance of the proposed estimator with two other estimators, the Laplacian based and the Average, in a network with 30 nodes. The first one uses fixed weights which are associated to the Laplacian of the graph, the second uses $\mathbf{K}(t) = 0$ and all the weights in $\mathbf{H}(t)$ equal to $1/|\mathcal{N}_i|$. We compare the estimators under various packet loss conditions and for the three different test signals of Fig. 3.

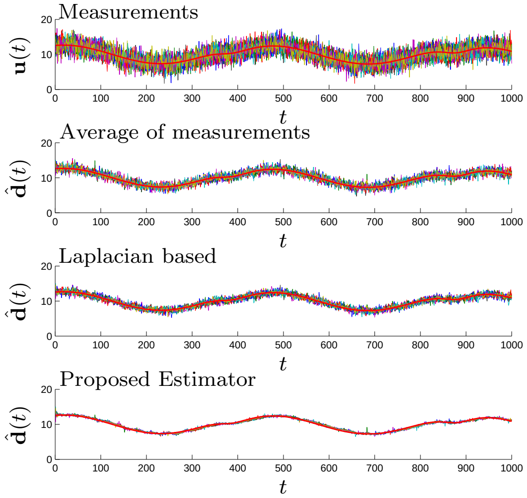


Figure 5: Comparison between the measurements taken by $N = 30$ nodes about the signal $d(t)$, shown in thin dashed red line, and the estimate computed by the estimator discussed in Section 5.

lem (3.2) guarantees $\gamma(\mathbf{K}(t) \circ \mathbf{\Phi}(t)) \le \gamma_{\max}$ in the estimation problem of Section 5. Simulations are referred to non-i.i.d. packet loss probabilities $q_{ij} = 0\%, 10\% \pm 5\%$, $20\% \pm 5\%$ and $30\% \pm 5\%$ for a network of $N = 30$ nodes. Fig. 6 shows the maximum value of $\gamma(\mathbf{K}(t) \circ \mathbf{\Phi}(t))$. Such a value is the maximum obtained in 30 Monte Carlo simulations, in which the network topology was maintained constant, but with difference realizations of measurement noise and packet loss process. The values of $\gamma(\mathbf{K}(t) \circ \mathbf{\Phi}(t))$ is always below the limit $\gamma_{\max}$ (dashed line), for various packet losses probabilities. The gap between the value $\gamma_{\max}$ and the actual value of $\gamma(\mathbf{K}(t) \circ \mathbf{\Phi}(t))$ is mainly caused by that the condition (4.1) is derived without using any a-priori knowledge on the network topology, which yields a conservative bound.

7. CONCLUSIONS AND FUTURE WORK

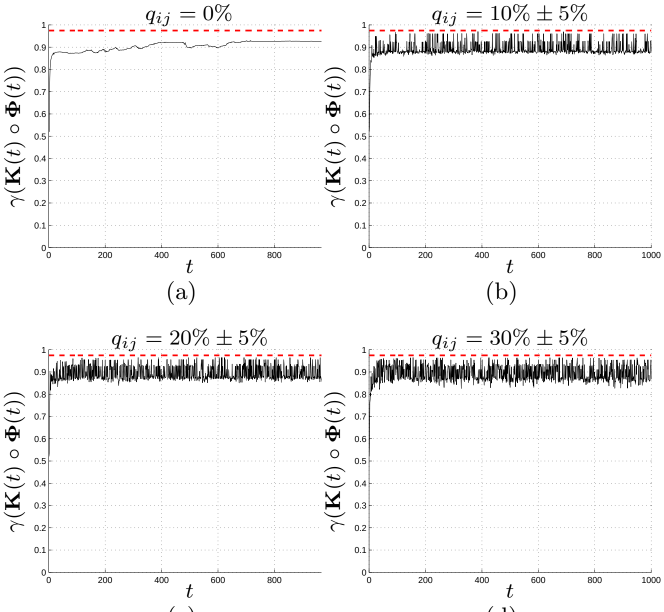
We presented a novel strategy for the distributed computation of the solution of a Lipschitz optimization problem. Specifically, we showed that the problem arises peer-to-peer consensus based estimation, where the network lacks of central coordination.

We showed that the optimization problem is very useful for decentralized tracking of time-varying signals. Our approach allows designing an estimator that runs locally in each node of the network and that does not require a central processing unit. Numerical results illustrate the validity of our analysis.

Future work will be devoted to the extension of the method presented here to problems for distributed resource control in wireless communication systems.

8. REFERENCES

[1] P. Alrikson and A. Rantzer. Experimental evaluation of a distributed kalman filter algorithm. In *In Proceedings of IEEE CDC*, 2007.



(a) (b)

(c) (d)

Figure 6: The plots show the maximum value of $\gamma(\mathbf{K}(t) \circ \mathbf{\Phi}(t))$ over 30 Monte Carlo simulations. We considered a fixed network with $N = 30$ nodes for different realization of the measurement noise and packet loss probability $q_{ij} = 0\%, 10\% \pm 5\%, 20\% \pm 5\%$, and $30\% \pm 5\%$, where $\pm 5\%$ is the maximum difference between the packet loss on one link and any other link. The dashed line show the value of $\gamma_{\max}$ chosen in the simulation. As it can be seen $\gamma(\mathbf{K}(t) \circ \mathbf{\Phi}(t))$ is always below the value $\gamma_{\max}$.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Athena Scientific, 1997.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[4] R. Carli, A. Chiuso, L. Schenato, and A. Zampieri. Distributed kalman filtering using consensus strategies. In *In Proceedings of IEEE CDC*, 2007.

[5] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44(3), 2008.

[6] C. Fischione, A. Speranzon, K. H. Johansson, and A. Sangiovanni-Vincentelli. Distributed estimation over wireless sensor networks with packet losses. *Online http://arxiv.org/abs/0810.3715*, 2008.

[7] H. Gharavi and P. R. Kumar, editors. *Proceedings of IEEE: Special Issue on Sensor Networks and Applications*, volume 91, 2003.

[8] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[9] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization, Nonconvex Optimization and its Applications.* Kluwer Academic Publisher, 1995.

[10] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[11] B. Johansson. *On Distributed Optimization in Networked Systems.* PhD thesis, KTH, 2009.

[12] Y. Kim, D. Gu, and I. Postlethwaite. Fault-tolerant cooperative target tracking in distributed uav networks. In *IFAC World Congress*, 2008.

[13] E. J. Msechu, A. Ribeiro, S. I. Roumeliotis, and G. B. Giannakis. Distributed Kalman filtering based on quantized innovations. In *Proceedings of IEEE ICASSP*, 2007.

[14] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. *In Proceedings of IEEE CDC*, 2005.

[15] L. Shi. *Resource optimization for networked estimator with guaranteed estimation quality.* PhD thesis, Caltech, 2008.

[16] D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. In *In Proceedings of IEEE CDC*, 2005.

[17] A. Speranzon, C. Fischione, B. Johansson, and K. Johansson. Adaptive distributed estimation over wireless sensor networks with packet losses. In *In Proceedings of IEEE CDC*, 2007.

[18] A. Speranzon, C. Fischione, K. H. Johansson, and A. Sangiovanni-Vincentelli. A distributed minimum variance estimator for sensor networks. *IEEE JSAC, Special Issue on Control and Communication*, 2008.

[19] S. Stankovic, M. Stankovic, and D. Stipanovic. Decentralized parameter estimation by consensus based stochastic approximation. *In Proceedings of IEEE CDC*, 2007.

[20] J. J. Xiao and Z.-Q. Luo. Universal decentralized estimation in a bandwidth-constrained sensor network. *IEEE Transactions on Signal Processing*, 2005.

[21] J. J. Xiao, A. Riberio, Z.-Q. Luo, and G. B. Giannakis. Distributed compression-estimation using wirless sensor netowrks. *IEEE Signal Processing Magazine*, 2006.

[22] L. Xiao, S. Boyd, and S. J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 2006.

[23] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. *In Proceedings of IEEE IPSN*, 2005.

APPENDIX

A.1 Proof of Theorem 4.1

To prove Theorem 4.1, we need some intermediate technical results:

Lemma A.1. *Problem* (3.2) *admits a non-trivial feasible solution* $\boldsymbol{x}^\ell = (x_1^\ell, \dots, x_i^\ell, \dots, x_N^\ell)^T \succ 0$ *where*

$$x_i^\ell = \frac{\gamma_{\max}}{4}\left(\sqrt{|\Theta_{\boldsymbol{\varphi}_i}|^2 + 4} - |\Theta_{\boldsymbol{\varphi}_i}|\right)^2 \qquad i = 1, \dots, N\,. \tag{A.1}$$

Proof. See [6]. □

This lemma is useful, because it allows us to establish the existence of an optimal solution:

Lemma A.2. *Problem* (3.2) *admits an optimal solution* $\boldsymbol{x}^*$*, which is the solution of the following set of nonlinear equations:*

$$x_i^* + \sqrt{x_i^*}\sum_{j\in\Theta_{\boldsymbol{\varphi}_i}}\sqrt{x_j^*} - \gamma_{\max} = 0 \qquad i = 1, \dots, N\,.$$

Proof. The proof is based on a useful rewriting of the optimization problem and by a *reductio ad absurdum* argument.

Let $y_i^2 = x_i$ for $i = 1, \dots, N$. Then, the optimization problem (3.2) can be rewritten as follows

$$\max_{\mathbf{y}} \quad \mathbf{y}^T\mathbf{y} \tag{A.2}$$

$$\text{s.t.} \quad \mathbf{y} - \mathbf{f}(\mathbf{y}) \preceq 0 \tag{A.3}$$

$$\mathbf{y} \succ 0\,.$$

where $\mathbf{f}(\mathbf{y}) = (f_1(\mathbf{y}), \dots, f_N(\mathbf{y}))^T$ and

$$f_i(\mathbf{y}) = y_i - \beta\left(y_i^2 + y_i\sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} y_j - \gamma_{\max}\right)\,,$$

with β being any positive scalar. This problem and (3.2) are obviously equivalent: for all $\beta > 0$, $\mathbf{S}(\boldsymbol{x}) \preceq 0$ if and only if $\mathbf{y} - \mathbf{f}(\mathbf{y}) \preceq 0$. Let $\mathbf{y}^*$ be an optimal solution of (A.2), then $x_i^* = {y_i^*}^2$. Problem (A.2) admits optimal solutions, since from Lemma A.1 the problem is feasible. We show next that the optimal solutions satisfy the constraints at the equality.

Let $\mathbf{y}^*$ be an optimal solution. Suppose by contradiction that there is constraint i that is satisfied at a strict inequality, namely $y_i^* < f_i(\mathbf{y}^*)$, while suppose $y_j^* \leq f_j(\mathbf{y}^*)$ for $i \neq j$. In the following, we show that from $\mathbf{y}^*$ we can construct a feasible solution $\mathbf{t}^*$ such that ${\mathbf{t}^*}^T\mathbf{t}^* > {\mathbf{y}^*}^T\mathbf{y}^*$, so that it is not possible that $\mathbf{y}^*$ be an optimal solution.

Since β is arbitrary, we can select a convenient value that makes the Lipschitz constant of the constraints small enough so that we can construct a feasible solution $\mathbf{t}^*$ such that ${\mathbf{t}^*}^T\mathbf{t}^* > {\mathbf{y}^*}^T\mathbf{y}^*$, as we show later. Let

$$\beta \leq \bar{\beta} < \min_{0\prec\mathbf{y}\preceq 1}\frac{1}{2y_i + \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} y_j} = \frac{1}{2 + \Theta_{\boldsymbol{\varphi}_i}}\,.$$

This choice of β makes $f_i(\mathbf{y})$ being an increasing function of y_i, and a decreasing function of y_j, for $j \neq i$. Indeed

$$\nabla_i f_i(\mathbf{y}^*) = 1 - \beta\left(2y_i^* + \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} y_j^*\right) > 0\,,$$

$$\nabla_j f_i(\mathbf{y}^*) = \begin{cases} -\beta y_i^* < 0, & \text{if } j \in \Theta_{\boldsymbol{\varphi}_i} \\ 0, & \text{if } j \notin \Theta_{\boldsymbol{\varphi}_i}, j \neq i \end{cases}$$

and $\nabla_i^2 f_i(\mathbf{y}^*) = -2\beta < 0$, $\nabla_j^2 f_i(\mathbf{y}^*) = 0$ if $j \neq i$. Let $\mathbf{v} \in \mathbb{R}^N$ such that $v_i \in (0, 1]$. We have

$$\begin{aligned} f_i(\mathbf{v}) =& f_i(\mathbf{y}^*) + \nabla f_i(\mathbf{y}^*)(\mathbf{v} - \mathbf{y}^*)^T \\ &+ \frac{1}{2}(\mathbf{v} - \mathbf{y}^*)^T\nabla^2 f_i(\mathbf{y}^*)(\mathbf{v} - \mathbf{y}^*)\,, \end{aligned} \tag{A.4}$$

because the third order derivatives are zero. Then, we chose a small positive scalar $0 < \varepsilon \leq f_i(\mathbf{y}^*) - \mathbf{y}^*$ so that $\mathbf{v}$ be an augmented vector of $\mathbf{y}^*$, with $v_i = \varepsilon + y_i^*$, $v_j = y_j^*$ for $j = 1, \dots, N$, $j \neq i$, and $v_i = y_i^* + \varepsilon \leq f_i(\mathbf{y}^*) < f_i(\mathbf{v})$. The last inequality is allowed by that $f_i(\mathbf{y})$ is an increasing function of y_i. From (A.4) it follows

$$\begin{aligned} f_i(\mathbf{v}) &= f_i(\mathbf{y}^*) + \nabla_i f_i(\mathbf{y}^*)\varepsilon + \frac{1}{2}\varepsilon^2\nabla_i^2 f_i(\mathbf{y}^*) \\ &= f_i(\mathbf{y}^*) + \left[1 - \beta\left(2y_i^* + \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} y_j^*\right)\right]\varepsilon - \beta\varepsilon^2 \\ &\triangleq f_i(\mathbf{y}^*) + \Delta f_i\,, \\ f_j(\mathbf{v}) &= f_j(\mathbf{y}^*) + \nabla_i f_j(\mathbf{y}^*)\varepsilon + \frac{1}{2}\varepsilon^2\nabla_i^2 f_j(\mathbf{y}^*) \\ &= f_j(\mathbf{y}^*) - \beta y_j^*\varepsilon \triangleq f_j(\mathbf{y}^*) - \Delta f_j \quad \text{if} \quad j \in \Theta_{\boldsymbol{\varphi}_i}\,, \\ f_\ell(\mathbf{v}) &= f_\ell(\mathbf{y}^*) \quad \text{otherwise}\;. \end{aligned}$$

By using ε and Δf_j, $j \neq i$, we can define a vector $\mathbf{t}^*$ such that $t_i^* = y_i^* + \varepsilon$, $t_j^* = y_j^* - \Delta f_j$ if $j \in \Theta_{\boldsymbol{\varphi}_i}$, and $t_\ell^* = y_\ell^*$ otherwise. Notice that $\mathbf{f}(\mathbf{v}) \preceq \mathbf{f}(\mathbf{t}^*)$ since $\mathbf{t}^* \preceq \mathbf{v}$. The solution $\mathbf{t}^*$ is feasible for problem (A.2), namely $\mathbf{t}^* \preceq \mathbf{f}(\mathbf{t}^*)$, because $t_i^* = y_i^* + \varepsilon = v_i \leq f_i(\mathbf{v}) \leq f_i(\mathbf{t}^*)$, $t_j^* = y_j^* - \Delta f_j \leq f_j(\mathbf{y}^*) - \Delta f_j = f_j(\mathbf{v}) \leq f_j(\mathbf{t}^*)$ if $j \in \Theta_{\boldsymbol{\varphi}_i}$ and $t_\ell^* = y_\ell^* \leq f_\ell(\mathbf{y}^*) = f_\ell(\mathbf{t}^*)$ if $\ell \neq i$ and

$l \notin \Theta_{\boldsymbol{\varphi}_i}$. Now, observe that

$$\begin{aligned}\mathbf{t}^{*T}\mathbf{t}^* - \mathbf{y}^{*T}\mathbf{y}^* &= \varepsilon^2 + \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}}^{N} \Delta f_j^2 + 2y_i^*\varepsilon + 2\sum_{j\in\Theta_{\boldsymbol{\varphi}_i}}^{N} y_j^*\Delta f_j \\ &= \varepsilon^2 + \beta^2\varepsilon^2 \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} {y_j^*}^2 + 2y_i^*\varepsilon - 2\beta\varepsilon \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} {y_j^*}^2 \,.\end{aligned}$$

The last right-hand side of previous equation is always positive, provided that one chooses

$$\varepsilon < \frac{2\beta \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} {y_j^*}^2}{1 + \beta^2 \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} {y_j^*}^2} \,.$$

This implies that $\mathbf{t}^{*T}\mathbf{t}^* > \mathbf{y}^{*T}\mathbf{y}^*$, namely that $\mathbf{t}^*$ is a feasible solution of (A.2) with higher cost function than $\mathbf{y}^*$, which is a contradiction because $\mathbf{y}^*$ was assumed to be an optimal solution. It follows that optimal solutions must satisfy all the constraints at the equality. □

The previous lemma guarantees that there are optimal solutions satisfying the constraints at the equality. However, we do not know yet if there is a global optimal solution. If there were multiple optimal solutions, we would have to chose the most fair for all nodes. Recall that a small x_i^* means smaller estimation quality. To establish the uniqueness of the optimal solution, we need the following lemma, which will be used for the proof of Theorem 4.1:

Lemma A.3. *Let* $\mathbf{J}(\boldsymbol{x}) = \nabla\mathbf{S}(\boldsymbol{x})$ *be the Jacobian of* $\mathbf{S}(\boldsymbol{x})$. *Then* $\mathbf{J}(\boldsymbol{x})$ *is a nonsingular matrix.*

Proof. See [6]. □

We are now in the position of proving Theorem 4.1. From Lemma A.2, we know that there is an optimal solution satisfying the constraints at the equality. We show next that such a solution is unique, thus proving Theorem 4.1.

Proof of Theorem 4.1. The proof of the uniqueness of the optimal solution is based on the Lagrange dual theory. First, observe that from Lemma A.2 the optimization problem admits optimal solutions. The optimization problem is non-convex, since the constraints (3.3) are not convex. The Lagrange dual theory for non-convex non-linear optimization problems can be applied. A qualification constraint from [9, pag. 25] states that strong duality holds if the optimization problem is feasible and the Jacobian of $\mathbf{S}(\boldsymbol{x})$ is non-singular, which we know from Lemma A.1 and Lemma A.3, respectively. Therefore, the optimal solution of the problem can be investigated via the Lagrange dual function $L(\boldsymbol{\xi}, \boldsymbol{x}) = -\boldsymbol{x}^T\mathbf{1} + \boldsymbol{\xi}^T\mathbf{S}(\boldsymbol{x})$, where $\boldsymbol{\xi} \succeq \mathbf{0}$ is the Lagrangian multiplier. From the KKT conditions it follows that $\mathbf{J}(\boldsymbol{x})^T\boldsymbol{\xi} = \mathbf{1}$. We see that previous equality trivially holds also for the optimal solution $\boldsymbol{x}^*$, namely $\mathbf{J}(\boldsymbol{x}^*)^T\boldsymbol{\xi}^* = \mathbf{1}$. From Lemma A.3, we know that the Jacobian is non-singular. It follows that there is a unique solution to the previous system of equations, namely $\boldsymbol{\xi}^* = \mathbf{J}(\boldsymbol{x}^*)^{-T}\mathbf{1}$, and since strong duality holds, we conclude that the optimal solution given by (4.1) is unique. □

A.2 Proof of Proposition 4.2

Given $\mathbf{y} \in \mathbb{R}^N$ and $\mathbf{z} \in \mathbb{R}^N$ such that $y_i \in (0, 1]$ and $z_i \in (0, 1]$, from the proof of Lemma A.2 we have that

$$\begin{aligned}f_i(\mathbf{z}) =& f_i(\mathbf{y}) + \nabla f_i(\mathbf{y})(\mathbf{z}-\mathbf{y})^T + \frac{1}{2}(\mathbf{z}-\mathbf{y})^T\nabla^2 f_i(\mathbf{y}) \\ &\times (\mathbf{z}-\mathbf{y}) \le f_i(\mathbf{y}) + \nabla f_i(\mathbf{y})(\mathbf{z}-\mathbf{y})^T \,, \qquad \text{(A.5)}\end{aligned}$$

because the third order derivatives of $f_i(\mathbf{y})$ are zero, whereas $\nabla^2 f_i(\mathbf{y}) \preceq 0 \ \forall \beta_i > 0$. It follows that

$$\|f_i(\mathbf{z}) - f_i(\mathbf{y})\|_2 \le \|\nabla f_i(\mathbf{y})\|_2 \|\mathbf{z}-\mathbf{y}\|_2 \,,$$

where

$$\|\nabla f_i(\mathbf{y})\|_2^2 = \left[1 - \beta_i\left(2y_i + \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} y_j\right)\right]^2 + \beta_i^2 \sum_{j\in\Theta_{\boldsymbol{\varphi}_i}} y_j^2 \,,$$

It is evident that $\|\nabla f_i(\mathbf{y})\|$ is the Lipschitz constant of the mapping $y_i = f_i(\mathbf{y})$. The value of β_i that minimizes the Lipschitz constant, while keeping such a constant strictly less than 1, maximizes the convergence speed of the mappings (4.2). Simple algebraic computations show that (4.3) is such an optimal value.

A.3 Proof of Proposition 5.5

By using the filter weights given by Proposition 5.1, the i-th component of the expectation of (5.1) can be upper bounded as follows:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\phi}}\,\mathbb{E}_v(e_i^2 - \mathbb{E}_v e_i^2) \qquad \text{(A.6)}\\ &\le \mathbb{E}_{\boldsymbol{\phi}} \frac{\sigma^2}{\boldsymbol{\phi}_i^T\left((\mathbf{P}(t-1) + \lambda_i(t)\mathbf{I}) \circ \boldsymbol{\phi}_i\boldsymbol{\phi}_i^T\right)^\dagger \boldsymbol{\phi}_i + \boldsymbol{\phi}_i^T\boldsymbol{\phi}_i} \,.\end{aligned}$$

The previous inequality follows from that the expectation is taken on a positive argument having a positive distribution, thus the sign of the argument is maintained [8, pag.392]. From Lemma V.2 in [6] it follows

$$\begin{aligned}&\boldsymbol{\phi}_i^T\left((\mathbf{P}(t-1) + \lambda_i(t)\mathbf{I}) \circ \boldsymbol{\phi}_i\boldsymbol{\phi}_i^T\right)^\dagger \boldsymbol{\phi}_i \\ &\quad \ge \boldsymbol{\phi}_i^T\boldsymbol{\phi}_i\left[\sigma^2\left(1 + \frac{2N}{(\sqrt{5}-1)\sqrt{\gamma_{\max}}}\right)\right]^{-1} \,.\end{aligned}$$

By using previous inequality in (A.6), we have

$$\mathbb{E}_{\boldsymbol{\phi}}\,\mathbb{E}_v(e_i^2 - \mathbb{E}_v e_i^2) \le \frac{(\sqrt{5}-1)\sqrt{\gamma_{\max}} + 2N}{2(\sqrt{5}-1)\sqrt{\gamma_{\max}} + 2N}\,\mathbb{E}_{\phi}\frac{\sigma^2}{\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i} \,.$$

The proposition follows by invoking Lemma 5.4.