A Distributed Estimation Method for Sensor Networks Based on Pareto Optimization

Francesca Boem, Yuzhe Xu, Carlo Fischione, and Thomas Parisini

Abstract—A novel distributed estimation method for sensor networks is proposed. The goal is to track a time-varying signal that is jointly measured by a network of sensor nodes despite the presence of noise: each node computes its local estimate as a weighted sum of its own and its neighbors' measurements and estimates and updates its weights to minimize both the variance and the mean of the estimation error by means of a suitable Pareto optimization problem. The estimator does not rely on a central coordination: both parameter optimization and estimation are distributed across the nodes. The performance of the distributed estimator is investigated in terms of estimation bias and estimation error. Moreover, an upper bound of the bias is provided. The effectiveness of the proposed estimator is illustrated via computer simulations and the performances are compared with other distributed schemes previously proposed in the literature. The results show that the estimation quality is comparable to that of one of the best existing distributed estimation algorithms, guaranteeing lower computational cost and time.

I. INTRODUCTION

Because of the increasing scientific interest for distributed, networked and large scale systems, distributed estimation algorithms have become extremely useful tools. In fact, due to the limited power, computation and communication capabilities of typical sensor networks, it is important for each node to be able to estimate the values of some signals, without the help of a centralized structure. This is relevant, for example, when monitoring large-scale systems. The state of the art proposes some algorithms for distributed estimation that are based on minimum variance techniques (for example, [1], [2]), even if the estimation error can be affected also by bias term when dealing with decentralized scenarios. In these cases, a trade-off between the variance of the estimation error and its bias occurs and it would be useful to minimize both of them, at the same time.

In this paper, we propose to formulate a distributed estimation problem in the framework of Pareto optimization, by solving a multi-objective minimization problem. A distributed estimator to track in a sensor network an unknown time-varying signal corrupted by noise is proposed. The centralized solution of the problem needs a central data-fusion center for the processing, but this approach clearly suffers from scalability issues and from the need of a large amount of energy and communication resources, with the further disadvantage of being non-robust to failures. On the other hand, the methodology proposed in this paper is scalable, since each node operates using only local information, although cooperation between neighboring nodes improves the estimation quality considerably.

Our approach refers to a class of in-network distributed estimation algorithms that have been proposed in the context of distributed adaptive filtering. More specifically, in [3] (and in the references of the same authors cited therein), diffusion-based and incremental adaptive solutions for the Recursive Least Squares and Least-Mean Squares approaches are presented. Moreover, in [2], a minimum variance estimator for distributed tracking of a noisy time-varying signal is proposed. See also [4] and the references cited therein for a survey on distributed estimation and control applications using linear consensus algorithms. Many of these papers focus on diffusion mechanisms so that each node of the network obtains the average of the initial samples of the network nodes. Consensus algorithms for distributed state estimation based on Kalman filters have recently been proposed, for example in [5]–[8]. This approach consists on a combination of the diffusion mechanism with distributed Kalman filtering [9]: the estimates of the local Kalman filters are communicated to neighboring nodes and then averaged using a diffusion strategy. In [10] and [11] an architecture for distributed estimation and motion control of mobile sensor networks for collaborative target tracking is proposed, by means of a continuous Kalman-Consensus filtering algorithm, introduced in [6] and [7]. In [12] and in previous works by the same authors, a distributed average computation of a time-varying signal is analyzed. A convex optimization problem is defined to compute the weights used by each node to minimize the least mean square deviation of the estimates. In [13], the problem of decentralized estimation as the solution of multiple constrained convex optimization subproblems is dealt with: a maximum likelihood estimator is formulated and the problem is decomposed using the alternating-direction method of multipliers in conjunction with a block coordinate descent approach. In [14], an algorithm is proposed exploiting the concept of moving horizon estimation (assuming the knowledge of the model, as in the case of Kalman filtering), and here assuming that the observed process has linear dynamics. A further approach to distributed estimation is based on nonlinear filters using self-synchronization and coupling functions: see, for instance, [15] where the estimate of each node is provided by the
state of a suitable nonlinear dynamical system and this latter system is coupled to some of the other nodes by a suitable static coupling function.

The main contribution of the present paper is the introduction of a novel distributed estimator for sensor networks that minimizes, at the same time, both the variance and the bias of the estimation error. Each node measures a time-varying noisy signal and computes an estimate as a weighted sum of its own and its neighbors’ measurements and estimates. At each step, the filter weights are updated locally as the solution of a Pareto optimization problem able to minimize both the variance and the mean of the estimation error. This allows a distributed implementation of the problem. A bound on the estimation error bias, linear with respect to the measurement noise variance and decaying with the number of neighboring nodes, that can be defined by suitably setting the Pareto parameter, is presented. Analogously with the estimator proposed in [2], our methodology differs from the related works in the literature because it tracks a time-varying signal, while the works cited in [4] are limited to averaging initial samples, and it does not rely on a model of the system that generates the signal to track, in contrast to model-based approaches like Kalman filtering methods. Indeed, the novelty of the present paper is the design of a distributed estimator able to minimize both the variance and the bias of the estimator error, by defining time-varying filter coefficients in a distributed way. In this way, a solution to an open challenge of the estimation problem is proposed: the trade-off between variance and bias of the estimation error.

To the best of author’s knowledge, this is the first time that the Pareto optimization problem has been used for the time-varying signal estimation by trading off estimation error bias and variance. Pareto optimization has been used in the field of sensor networks for other purposes: some examples are [16], [17] and [18]. In [17], the optimization of wireless sensor network layout is analyzed in order to determine the best sensor placement, where the competing objectives considered are the total sensor coverage and the lifetime of the network. On the other hand, in [18], an optimal sensor network design for multi rate systems is proposed, evaluating trade-off solutions between the quality of the state estimation and the total measurement cost. In both these last papers, the problem is solved by genetic algorithms. [16] deals with the localization problem, studying the optimal fusion of ranging information with speed and absolute orientation information.

The paper is organized as follows: after introducing a few useful notations, in Section III, the problem dealt with is formulated and the distributed estimator is proposed. In Section IV, the choice of the Pareto parameter is addressed while, in Section V, a bound to the bias is given. Then, the structure and the implementation of the estimator is discussed in Section VI and in Section VII, the effectiveness of the proposed algorithms is shown also including a comparison with other methods available in the literature. Finally, some concluding remarks are drawn in Section VIII.

II. Notations

By diag(\ldots) we denote a diagonal matrix having the elements on the diagonal as the one in brackets. By \|\cdot\| we denote the cardinality of the argument and by \|\cdot\| the spectral norm of a matrix. Given a stochastic variable \(x\), we denote by \(Ex\) its expected value. Finally, by \(1\) and \(I\) we denote the vector \((1,\ldots,1)^T\) and the identity matrix, respectively.

III. Problem Formulation

Let us consider \(N > 1\) sensor nodes, measuring a common scalar signal \(d(t)\) affected by additive noise:

\[ u_i(t) = d(t) + v_i(t), \quad i = 1,\ldots,N, \]

where \(v_i(t)\) is a zero-mean white noise. Collecting the variables in vectors, it is possible to write:

\[ u(t) = d(t)1 + v(t). \]

We assume the covariance matrix \(\Sigma\) of \(v(t)\) to be diagonal:

\[ \Sigma = \text{diag}(\sigma_1^2,\ldots,\sigma_N^2). \]

As in [2], the communication network is modeled as an undirected graph \(G = (V,E)\), where \(N_i = \{j \in V : (j,i) \in E \cup \{i\}\}\) is the set of neighbors of node \(i \in V\) plus the node itself. It is assumed that there are no message losses.

Each node \(i\) computes an estimate \(x_i(t)\) of the signal \(d(t)\), taking a linear combination of neighboring estimates and measurements:

\[ x_i(t) = \sum_{j \in N_i} k_{i,j}(t) x_j(t-1) + \sum_{j \in N_i} h_{i,j}(t) u_j(t). \]

In vector notation, the above becomes

\[ x(t) = K(t)x(t-1) + H(t)u(t), \quad (1) \]

where \(K\) and \(H\) can be seen as the adjacency matrices of two graphs with time-varying weights. The algorithm is initialized with \(x_j(0) = u_j(0), \quad j \in N_i\).

The estimation error \(e(t) = x(t) - d(t)\) can be computed:

\[ e(t) = K(t)e(t-1) + d(t)(K(t) + H(t) - I)1 - \delta(t)K(t)1 + H(t)v(t), \quad (2) \]

where \(\delta(t) = d(t) - d(t-1)\). Moreover, the expected value of the estimation error dynamics with respect to the stochastic variable \(v(t)\) is given by

\[ Ex(t) = K(t)Ex(t-1) + d(t)(K(t) + H(t) - I)1 - \delta(t)K(t)1. \quad (3) \]

Now, we introduce the following

Assumption 1: We assume that \((K(t) + H(t))1 = 1\).

Assumption 1 is needed to guarantee the convergence properties of the centralized estimation error, that are derived in the paper of Speranzon et al. (see [2], section B) and that hold likewise in our case: if, in addition, \(H(t)1 = 1\), then, the expected value of the estimation error converges to 0 and so the estimation error is unbiased; otherwise, if \(d(t)\) is
slowly varying (that is, \(\delta(t)\) is bounded), then also \(\|Ee(t)\|\)
tends to a bounded value in the neighborhood of the origin.
In this way, if we consider the \(i\)-th node, the expected value of
the estimation error can be computed in a distributed way as:
\[
Ee_i(t) = \kappa_i^T(t)Ee_i(t-1) - \kappa_i^T(t)\delta(t),
\]
(4)
where \(e_i(t)\) collects the estimation errors available at node \(i\),
derived according to their indices:
\[
e_i = (e_{i1}, \ldots, e_{iM_i})^T, \quad i_1 < \cdots < i_{M_i},
\]
with \(M_i = |\mathcal{N}_i|\) the number of neighbors of node \(i\) plus \(i\) itself,
and we introduced \(\kappa_i^T(t)\) and \(\eta_i^T(t)\) that correspond
to the non-zero elements of the \(i\)-th row of matrices \(K(t)\)
and \(H(t)\) respectively. Note that Assumption 1 is equivalent
to require \((\kappa_i(t) + \eta_i(t))^T\mathbf{1} = 1\), that can be computed in a
distributed way. Now, we are able to compute the variance of
the estimation error:
\[
E(e_i(t) - Ee_i(t))^2 = \kappa_i^T(t)\Gamma_i(t)(1 - \kappa_i(t)) + \sigma^2 \eta_i^T(t)\eta_i(t),
\]
(5)
where \(\Gamma_i(t) = E(e_i(t) - Ee_i(t))(e_i(t) - Ee_i(t))^T\) is the
error covariance matrix. When considering the noise co-
variance matrix \(\Sigma = \text{diag}([\sigma_1^2, \ldots, \sigma_{M_i}^2])\), it is sufficient
in the following to replace \(\sigma^2\) with an appropriate defined local matrix \(Q_i =
\text{diag}(\sigma_1^2, \ldots, \sigma_{M_i}^2)\).

We want to determine \(\kappa_i^T(t)\) and \(\eta_i^T(t)\) at each step, si-
multaneously minimizing both the variance of the estimation
error and the bias. We remark that the approach proposed in
this paper differs substantially from the one proposed in [2]
because in the latter paper only the variance of the estimation
error is minimized. We propose to formulate the problem as
a Pareto Optimization problem:
\[
\min_{\kappa_i^T, \eta_i^T} (1 - \rho_i)P_2 + \rho_iP_1^2
\]
\[
\text{s.t. } (\kappa_i(t) + \eta_i(t))^T\mathbf{1} = 1,
\]
(6)
where \(0 \leq \rho_i \leq 1\), \(P_1 = E\xi(t)\) is the bias term of the
estimation error and \(P_2 = E(e_i(t) - Ee_i(t))^2\) is the variance
term. This problem can be solved in a distributed way by
each node \(i\). It can be rewritten as:
\[
\min_{\kappa^T_i, \eta^T_i} \kappa^T_i(1 - \rho_i)\sigma^2 \eta_i^T(t)\eta_i(t)
\]
\[
\text{s.t. } (\kappa_i(t) + \eta_i(t))^T\mathbf{1} = 1,
\]
(7)
where
\[
\Theta_i(t) = (1 - \rho_i)\Gamma_i(t-1) + \rho_i\Lambda_i(t),
\]
with \(\Lambda_i(t) = (E\xi_i(t) - \delta(t))(E\xi_i(t-1) - \delta(t))\mathbf{1}^T\).
The problem is convex since the cost function and the constraint
are convex: \(\Gamma_i(t-1)\) is positive definite since it represents
the covariance matrix, \(\Lambda_i(t)\) is positive semi-definite, so that
the linear combination of these two matrices with non-negative
coefficients is positive definite, as well. In addition, Slater’s
conditions are satisfied, so strong duality holds [19].
We derived optimal values of \(\kappa_i^T(t)\) and \(\eta_i^T(t)\) for a fixed \(\rho_i\).

**Proposition 3.1:** For a given positive definite matrix
\(\Theta_i(t)\), the solution to the optimization problem is:
\[
\kappa_i(t) = \frac{(1 - \rho_i)\sigma^2 \Theta_i^{-1}\mathbf{1}}{(1 - \rho_i)\sigma^2 \Theta_i^{-1} + M_i},
\]
(8)
\[
\eta_i(t) = \frac{1}{(1 - \rho_i)\sigma^2 \Theta_i^{-1} + M_i},
\]
(9)

**Proof:** Since the problem is convex and Slater’s condition holds,
the Karush-Kuhn-Tucker (KKT) conditions are both necessary
and sufficient for optimality:
\[
(\kappa_i^* + \eta_i^*)^T\mathbf{1} - 1 = 0,
\]
\[
2\Theta_i \kappa_i^* + \nu_i^* = 0,
\]
\[
2\sigma^2 (1 - \rho_i) \nu_i^* + \nu_i^* \mathbf{1} = 0,
\]
where \((\kappa_i^*, \eta_i^*)\) are the primal optimal points and \(\nu_i^*\) is the
dual optimal variable. The last two KKT conditions derive from
\(\nabla \kappa_i, L(\kappa_i, \eta_i, \nu_i)\) and \(\nabla \eta_i, L(\kappa_i, \eta_i, \nu_i)\) with \(L\) being the
Lagrangian form
\[
L(\kappa_i, \eta_i, \nu_i) = \kappa_i \Theta_i \kappa_i + (1 - \rho_i)\sigma^2 \eta_i^T \eta_i + \nu_i((\kappa_i + \eta_i)^T \mathbf{1} - 1).
\]
It is possible to provide the solution in a closed form, simply
by solving the system of equations derived by the KKT
conditions.

**IV. CHOICE OF THE PARETO PARAMETER**

In the literature, the best value of \(\rho_i\) is determined by
building the Pareto trade–off curve and selecting the “knee–
point” of this curve, that is, choosing \(\rho_i^*\) such that \(P_1\) and \(P_2\),
computed with the values \(\kappa_i^T(\rho_i^*)\) and \(\eta_i^T(\rho_i^*)\), are \(P_2 =
P_1^2\). This can be obtained by solving the following further problem:
\[
\min_{\rho_i} (P_2(\kappa_i^T(\rho_i), \eta_i^T(\rho_i)) - P_1^2(\kappa_i^T(\rho_i), \eta_i^T(\rho_i)))^2.
\]
This problem is highly non–linear. Numerical methods can
be used to compute the optimal value and, in the literature,
genetic algorithms are often used (see, for instance, [18],
[17]). In this paper, we tested different approaches for the
definition of the Pareto parameter. Specifically, we chose to
compute it locally, using the Nelder-Mead simplex algorithm
as described in [20], to minimize the cost function \((1 - \rho)P_2 + \rho P_1^2\)
with the values \(P_1\) and \(P_2\) obtained at the previous step. Note that such values are function of \(\rho\).
The Nelder-Mead algorithm is one of the most widely used
methods for nonlinear unconstrained optimization problem
adopting a direct search method that allows to avoid the
computation of numerical or analytic gradients, which are
difficult to obtain in our case due to the presence of \(\Theta^{-1}\).

In this connection, it is worth noting that, by setting the
Pareto parameter as \(\rho_i = 1\), it turns out that only the bias is
minimized and the optimal cost function is given by
\[
P_1^2 = 0, \quad P_2 = \frac{\sigma^2}{M_i}.
\]
(10)
On the other hand, by setting \(\rho_i = 0\), the variance is
minimized and we obtain
\[ P_1^2 = \left( \frac{\sigma^2 \Gamma_i^{-1}}{\sigma^2 \Gamma_i^{-1} + M_i} \right)^\top \Lambda_i \left( \frac{\sigma^2 \Gamma_i^{-1}}{\sigma^2 \Gamma_i^{-1} + M_i} \right), \]
\[ P_2 = \left( \frac{\sigma^2 \Gamma_i^{-1}}{\sigma^2 \Gamma_i^{-1} + M_i} \right)^\top \Gamma_i \left( \frac{\sigma^2 \Gamma_i^{-1}}{\sigma^2 \Gamma_i^{-1} + M_i} \right) + \frac{\sigma^2}{\sigma^2 \Gamma_i^{-1} + M_i} I. \]

Pareto parameter can be set depending on the required features. In the next Section, it will be shown how to define a bound on the bias by appropriately setting this parameter.

V. BOUNDS ON THE BIAS

As proposed in [2], it is possible to bound the bias by defining the following global constraint:
\[ \gamma_{\max}(K(t)) \leq f(\Delta, \mathcal{Y}), \]
where \( \gamma_{\max}(K(t)) \) denotes the largest singular value of matrix \( K(t) \), \( \mathcal{Y} \) denotes the Signal-to-Noise Ratio, \( \Delta \) is a bound on the derivative of the signal and
\[ f(\Delta, \mathcal{Y}) = \frac{\sqrt{\mathcal{Y}}}{\sqrt{\mathcal{Y}} + \Delta}. \]

In [2], it is shown that this global constraint holds when the following local constraint holds:
\[ \|\kappa_i\|^2 \leq \psi_i, \]
where \( \psi_i > 0 \) is a suitable constant scalar that can be computed locally (for more details, see [2]). This new constraint ensures the stability of the estimation error even if it leads to a distributed solution which is in general different from the centralized one. Problem (7) can be reformulated, taking into account the bound on the bias, as follows:
\[ \min_{\kappa_i, \eta_i} \kappa_i^\top(t) \Theta_i(t) \kappa_i(t) + (1 - \rho_i) \sigma^2 \eta_i^\top(t) \eta_i(t) \quad \text{s.t.} \quad (\kappa_i(t) + \eta_i(t))^\top 1 = 1 \]
\[ \|\kappa_i\|^2 \leq \psi_i. \]

The solution of this problem cannot be computed in a closed form. Because of the convexity of the new constraint, the problem is convex. Strong duality holds, and the KKT conditions are necessary and sufficient for optimality: the primal and dual \((\kappa_i^*, \eta_i^*)\) and \((\xi_i^*, \nu_i^*)\) have to satisfy:
\[ (\kappa_i^*)^\top \kappa_i^* - \psi_i \leq 0, \quad (\kappa_i^* + \eta_i^*)^\top 1 = 1 = 0, \]
\[ (\xi_i^* + \nu_i^*)^\top \kappa_i^* = 0, \quad (\xi_i^* + \nu_i^*)^\top \eta_i^* = 0, \]
\[ 2(\Theta_i + \xi_i^* I) \kappa_i^* + \nu_i^* 1 = 0, \quad 2\sigma^2 (1 - \rho_i) \eta_i^* + \nu_i^* 1 = 0. \]

where last two conditions are obtained from the Lagrangian
\[ L(\kappa_i, \eta_i, \xi_i, \nu_i) = \kappa_i^\top \Theta_i \kappa_i + (1 - \rho_i) \sigma^2 \eta_i^\top \eta_i + \xi_i^\top \kappa_i^* - \psi_i + \nu_i^\top (\kappa_i + \eta_i)^\top 1 - 1. \]

By combining these two KKT conditions with the second one, we obtain the optimal values
\[ \kappa_i(t) = \frac{(1 - \rho_i) \sigma^2 (\Theta_i + \xi_i I)^{-1}}{(1 - \rho_i) \sigma^2 I^\top (\Theta_i + \xi_i I)^{-1} + M_i}, \]
\[ \eta_i(t) = \frac{1}{(1 - \rho_i) \sigma^2 I^\top (\Theta_i + \xi_i I)^{-1} + M_i}. \]

The fourth KKT condition establishes that either \( \xi^* = 0 \) or \( (\kappa_i^*)^\top \kappa_i^* = \psi_i \). Comparing the results with the previous ones (8), (9), it is possible to see that choosing the case \( \xi^* > 0 \), the optimal solutions are the same of the problem (7) (where there is no bias constraint), if the constraint \((\kappa_i^*)^\top \kappa_i^* \leq \psi_i \) holds. By observing that, in this case,
\[ (\kappa_i)^\top \kappa_i = \frac{(1 - \rho_i)^2 \sigma^4 I^\top (\Theta_i)^{-1} I}{((1 - \rho_i)^2 \sigma^4 I^\top (\Theta_i)^{-1} I + M_i)^2} \leq \frac{(1 - \rho_i)^2 \sigma^4 (\Theta_i^{-1})^2}{M_i}, \]
and by choosing
\[ \psi_i = \frac{(1 - \rho_i)^2 \sigma^4 (\Theta_i^{-1})^2}{M_i}, \]
the fourth KKT condition is satisfied. We can see that, in this way, it is possible to define an appropriate bound to the bias, by setting an appropriate value of the Pareto parameter \( \rho_i \) and maintaining the results of Equations (8) and (9).

VI. ESTIMATOR STRUCTURE

Now, we analyze how to implement the estimator. Some of the estimates of the needed quantities are derived in [2]. In our case, we considered a signal that is component-wise quasi-stationary. Since the time-varying linear system in (1) is uniformly bounded-input bounded-output stable, then also \( x(t) \) is quasi-stationary (see [2] for more details) and hence the mean \( \mathbb{E}\xi_i = m_{\xi_i}(t) \) and the covariance matrix \( \Gamma_i(t) \) can be estimated from the samples as follows:
\[ \hat{m}_{\xi_i}(t) = \frac{1}{T} \sum_{\tau=0}^{T} \hat{\xi}_i(\tau) \]
\[ \hat{\Gamma}_i(t) = \frac{1}{T} \sum_{\tau=0}^{T} (\hat{\xi}_i(\tau) - \hat{m}_{\xi_i}(\tau))(\hat{\xi}_i(\tau) - \hat{m}_{\xi_i}(\tau))^\top, \]
where \( \hat{\xi}_i(t) \) is an estimate of the error, that is obtained in [2], taking into account both \( x_i(t) \) and \( u_i(t) \), by solving a regularized linear least squares problem:
\[ \min_{\mathbf{x}, \mathbf{u}} \left\| \begin{bmatrix} \mathbf{x}_i \\ \mathbf{u}_i \end{bmatrix} - A \left( \frac{d}{\beta} \right) \right\|^2 + \nu \left\| B \left( \frac{d}{\beta} \right) \right\|^2, \]
where the vector \( \mathbf{x}_i(t-1) := (x_i(t-1), \ldots, x_{iM_i}(t-1))^\top \), with \( \{i_1, \ldots, i_{M_i}\} \in \mathcal{N}_i \), collects estimates, \( \mathbf{u}_i(t) := (u_{i_1}(t), \ldots, u_{iM_i}(t))^\top \) collects neighboring measurements,
\[ A = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \mathbb{R}^{2M_i \times M_i+1}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix} \in \mathbb{R}^{M_i \times M_i+1} \]
and \( \nu > 0 \) is a parameter, which can be chosen using the Generalized Cross-Validation method (for more details, see
[2], where a sub-optimal result is presented: \( \nu = \arg \min \| (A^T A + \nu B^T B)^{-1} A^T (x^T - u^T) \| \). The solution is
\[
(\hat{d}, \hat{\beta}^T) = (x^T, u^T) A (A^T A + \nu B^T B)^{-1},
\]
where \( \hat{\beta} \) is an estimate of \( \epsilon_i(t) \) and \( \hat{d} \) can be used to estimate \( \delta(t) = d(t) - d(t - 1) \). Finally, we propose the following solution to estimate \( \Lambda_i(t) \):
\[
\hat{\Lambda}_i(t) = \frac{1}{t} \sum_{\tau=1}^{t} (\hat{m}_{\epsilon_i}(\tau - 1) - \hat{\delta}(\tau) 1)(\hat{m}_{\epsilon_i}(\tau - 1) - \hat{\delta}(\tau) 1)^T.
\]

(18)

A. Distributed Estimation Algorithm

In this subsection, the implementation of the proposed algorithm is addressed. Each node has to implement the estimator given by Algorithm 1. Notice that \( \rho_i \) is computed locally by using the Nelder-Mead simplex algorithm [20] in Line 13. Once the parameter has been calculated, the optimal weights \( \kappa_i(t) \) and \( \eta_i(t) \) are computed (Lines 14 - 15) and the local estimate of the signal \( x_i(t) \) can be obtained (Line 16). After that, the values of the estimates of \( \hat{F}_i(t) \), \( \hat{\Lambda}_i(t) \) and \( \hat{m}_{\epsilon_i}(t) \) can be updated using new signal estimates and measurements (Lines 17 - 24).

B. Computational complexity

The computational complexity of the proposed estimator \( E_p \) is given mainly by three components: the computational complexity of a matrix inverse, the one of a Nelder-Mead simplex algorithm, and that needed for the estimation of the covariance matrix. The computation of a matrix inverse is required to compute the optimal weights \( \kappa_i(t) \) and \( \eta_i(t) \). It has complexity \( O(|N_i|^2) \). The computation of a Nelder-Mead simplex algorithm is required to compute the optimal Pareto parameter \( \rho_i \). This computation has complexity \( O(N_{iter} |N_i|^2) \), where \( N_{iter} \) is a number of iterations. In the simulation, the typical \( N_{iter} \) are less than 20. The computation of the covariance matrix is required to compute the approximate estimates \( \hat{\Gamma}_i(t) \) and \( \hat{\Lambda}_i(t) \); the complexity is \( O(\text{Tablesize} \times \log(\text{Tablesize})) \), where the Tablesize is the size of a look-up table used to speed up the computation of the quadratically constrained least-square problem [2] given in (16). We set Tablesize = 100.

VII. SIMULATION RESULTS

In this section, numerical simulations are described to show the effectiveness of the proposed distributed estimator and to compare the performances with estimators available in the literature. We consider the following approaches:

- **E1:** \( K = 0 \) and \( H = [\eta_{ij}] \) with \( \eta_{ij} = 1/M_i \) if node \( i \) and \( j \) communicate, and \( \eta_{ij} = 0 \) otherwise, resulting in the average of the measurements.
- **E2:** \( K = [\kappa_{ij}] \), where \( \kappa_{ii} = 1/2M_i, \kappa_{ij} = 1/M_i \) if node \( i \) and \( j \) communicate, \( \kappa_{ij} = 0 \) otherwise, whereas \( H = [\eta_{ij}] \) with \( \eta_{ii} = 1/2M_i \) and \( \eta_{ij} = 0 \) elsewhere. This results in the average of the old estimates and node’s single measurement.
- **E3:** \( K = H \) with \( \kappa_{ij} = 1/2M_i \) if node \( i \) and \( j \) communicate, and \( \kappa_{ij} = 0 \) otherwise. This is the average of the old estimates and all local new measurements.
- **E4:** The estimator proposed in [2], which minimizes only the variance of the estimation error. 
- **E_p:** The estimator proposed in this paper.

A 35-nodes network is considered, as in [2], by distributing the nodes randomly over a squared area of size \( N/3 \) and the graph is then obtained by letting two nodes communicate if their relative distance is less than \( \sqrt{N} \). One example can be seen in Figure 1. Figure 2 shows the signal that has to be tracked in simulations, that we used to benchmark the estimators. The test signals used in simulation are highly non-linear and generated so that the signal presents intervals in which is very slowly varying and some where it is faster. The noises in the measurements are generated randomly for each node: in the simulation each

---

**Algorithm 1 Estimation algorithm for node \( i \)**

1. \( t := 0 \)
2. \( \hat{m}_{\epsilon_i}(0) := 0 \)
3. \( \hat{\Gamma}_i(0) := \sigma^2 I \)
4. \( \hat{\Lambda}_i(0) := \sigma^2 I \)
5. Compute \( \rho_i \)
6. \( \hat{\Theta}_i(t) := (1 - \rho_i)\hat{\Gamma}_i(t) + \rho_i\hat{\Lambda}_i(t) \)
7. \( x_i(0) := u_i(0) \)
8. **while** forever **do**
9. \( M_i := |N_i| \)
10. \( t := t + 1 \)
11. Collect estimates \( \hat{x}_i(t - 1) := (x_{i1}(t - 1), ..., x_{iM_i}(t - 1))^T \) where \( \{i_1, ..., i_{M_i}\} \in N_i \)
12. Collect measurements \( \hat{u}_i(t) := (u_{i1}(t), ..., u_{iM_i}(t))^T \) where \( \{i_1, ..., i_{M_i}\} \in N_i \)
13. Compute \( \rho_i \)
14. \( \kappa_i(t) := \frac{(1 - \rho_i)^2 \hat{\Theta}_i(t) \hat{\Theta}_i(t) + (1 + \rho_i)/2}{(1 - \rho_i)^2 \hat{\Theta}_i(t) + (1 + \rho_i)/2} \)
15. \( \eta_i(t) := \frac{(1 - \rho_i)^2 \hat{\Theta}_i(t) \hat{\Theta}_i(t) + (1 + \rho_i)/2}{(1 - \rho_i)^2 \hat{\Theta}_i(t) + (1 + \rho_i)/2} \)
16. \( x_i(t) := \kappa_i(t)^T \hat{x}_i(t - 1) + \eta_i(t)^T \hat{u}_i(t) \)
17. \( \hat{\delta}(t) := d(t) - d(t - 1) \)
18. \( \hat{\epsilon}_i(t) := \hat{\delta}(t) - \hat{\delta}(t - 1) \)
19. \( \hat{\delta}(t) := \hat{d}(t) + \hat{d}(t - 1) \)
20. Compute \( \rho_i \)
21. \( \hat{m}_{\epsilon_i}(t) := \frac{1}{t} \hat{m}_{\epsilon_i}(t - 1) + \frac{1}{\hat{\delta}(t)} \hat{\epsilon}_i(t) \)
22. \( \hat{\Gamma}_i(t) := \frac{1}{t} \hat{\Gamma}_i(t - 1) + \frac{1}{\hat{\delta}(t)} (\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t)) (\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t))^T \)
23. \( \hat{\Lambda}_i(t) := \frac{1}{t} \hat{\Lambda}_i(t - 1) + \frac{1}{\hat{\delta}(t)} (\hat{m}_{\epsilon_i}(t) - \hat{\delta}(t) 1)^T \)
24. \( \hat{\Theta}_i(t) := (1 - \rho_i)\hat{\Gamma}_i(t) + \rho_i\hat{\Lambda}_i(t) \)
25. **end while**

---

779
node has its own measurement noise. In order to compare the performances obtained by the five estimators, we analyze the mean square error of the estimates of each node, averaging the mean square error over all nodes of the network and obtaining what we denote MSE. After that, we averaged the MSE values calculated from 50 different simulations, with different networks and different measurements noises. The results are presented in Table I.

**TABLE I**

SIMULATIONS RESULTS: MEAN MSE OVER 50 SIMULATIONS

<table>
<thead>
<tr>
<th>Estimator</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>0.6082</td>
</tr>
<tr>
<td>$E_2$</td>
<td>0.0542</td>
</tr>
<tr>
<td>$E_3$</td>
<td>0.1771</td>
</tr>
<tr>
<td>$E_4$</td>
<td>0.0400</td>
</tr>
<tr>
<td>$E_p$</td>
<td>0.0317</td>
</tr>
</tbody>
</table>

Measurements and the resulting estimates for all nodes for each considered estimator are shown in Figure 3 for one experiment, but they are similar in all the tested cases, with different networks and different measurements noise. As we can see, all estimators are able to track the signal, but the quality of the estimates is different. The proposed algorithm performs much better than methods $E_1$, $E_2$ and $E_3$ and has results similar to the one obtained with the minimum variance estimator of [2], presenting a lower averaged value of MSE. Moreover, one drawback of this last method $E_4$ is the computational cost, while our approach has lower computational complexity, because $E_1$ requires in addition the distributed computation of a constraint that permits to guarantee the boundedness of the bias. Furthermore, it is important to remark that the computational time required by our estimator is lower: the proposed approach saves about 20% time to track the signal in simulations run under the same conditions on a Intel i7 processor (2.80GHz, 4 Cores, 8 Logical Processors). As example, in the presented simulation case, the time needed for the estimate computation in average for each node is 0.0053s for our estimator $E_p$ and 0.0067s for estimator $E_4$.

**VIII. CONCLUSIONS**

In this paper, a novel distributed estimator for tracking an unknown noisy time-varying signal in a sensor network was proposed. The filter coefficients were updated locally in order to minimize both the variance and the mean of the estimation error by means of a Pareto optimization problem. This is a novelty of the proposed approach, since usually estimation methods only minimize the variance of the estimation error, whereas in this paper we minimized simultaneously both the variance of the estimation error and the bias. We discussed the analytical results and derived an upper bound on the estimation error bias, depending on the Pareto parameter, which is linear in the measurement noise variance and decays with the number of neighboring nodes. The simulation results showed the effectiveness of the proposed methodology, guaranteeing a lower computational cost than previous works having similar estimation quality.

**REFERENCES**


Fig. 3. In the first graph, the signal to be tracked and the measurements realized by all the \( N = 35 \) nodes. In the following graphs, the estimates obtained by the different five estimators for each node. The tracked signal is represented by the thick blue curve; measurements and estimates have different colors for each node.


