A Distributed Estimation Algorithm for Tracking over Wireless Sensor Networks

Alberto Speranzon, Carlo Fischione, Karl Henrik Johansson
School of Electrical and Engineering
Royal Institute of Technology, Stockholm, Sweden 100-44
Email: {albspe|carlofi|kallej}@ee.kth.se

Abstract—A new distributed estimation algorithm for tracking using a wireless sensor network is presented. We investigate how to track a time varying signal, noisily sensed by the nodes of the network. The algorithm is distributed, meaning that it does not require a central coordination among the nodes. Moreover, the proposed approach is scalable with respect to the network size, which means that its complexity does not grow with respect to the total number of nodes. The algorithm designed turns out to be composed by a cascade structure. Local constraints are determined to guarantee the global asymptotic stability of the estimation error. The algorithm can be applied e.g., for the position estimation, temporal synchronization, as well as tracking of signals. Performance is illustrated by simulations, where our filter is shown to behave better than other distributed schemes proposed in the literature.

Index Terms—Distributed filtering; Wireless sensor networks; Cooperative Communication.

I. INTRODUCTION

Estimation using wireless networks of sensors (WSNs) has a growing domain of applications in diverse areas such as communication, environmental monitoring, industrial automation, surveillance, and transportation ([1], [2]). Essential features of these systems are that they are widely distributed and that their operations rely on limited computational and communication resources, as discussed in [3]. Furthermore, central coordination is often impractical or impossible, since the nodes communicate with low data rate and short range. This is the case of large networks where the nodes have to track or estimate a common time-varying signal (as, e.g., a clock or the intensity of a physical phenomenon commonly sensed).

There is a recent interest in a class of simple and highly scalable distributed estimation algorithms. Here we limit the discussion to some recent and relevant contributions. In [4], the problem of distributed estimation of an average by a wireless sensor network is presented. It is assumed that nodes take a set of initial samples, and then iteratively exchange the averages of the samples locally collected. Each node reaches asymptotically the global average. In [5], a more general approach is investigated. The authors propose the consensus of the average of a common time-varying signal measured by

each sensor, when the signal is affected by a zero-mean noise. The same linear filter was also considered in [6], where a faster coefficient computation is investigated. A related consensus filter for distributed low-pass sensor fusion has been proposed in [7]. In [8], we have proposed an heuristic algorithm for cooperative estimation in WSNs. It is an extension to the algorithms in [4]–[7], in that the filter is adaptive and thereby it is better in tracking time-varying signals. We showed that the filter performs well in practice.

The specific contribution of this paper is an extension of the algorithm presented in [8]. A mathematical framework is proposed to design a cascade structured filter. The filter is supposed to run locally in each node. It firstly computes the optimal estimate, and then updates the statistical parameters necessary to the new stage of estimation. A constrained quadratic optimization problem is addressed to suitably adapt the filter coefficients. The solution of the problem provides a slightly biased estimate but with a low variance. In order to characterize the statistical inputs for the new estimation stage, a constrained and weighted sub-optimal least square is investigated to provide an unbiased estimate. The approach presents a natural tradeoff between tracking performance and computational complexity.

Compared to recent relevant work [4]-[7], our approach is original because we adopt a more general model of the filter structure since we do not rely to the common heuristic of the Laplacian associated to the communication graph. Moreover, our approach differ from [5] and [6], since we are interested to investigate distributed solutions, whereas in such papers centralized algorithms are used to compute the filter coefficients. Furthermore, with respect to [8], we provide a better theoretical characterization of the estimate of the error covariance matrix. The characterization enables to design a filter with enhanced performance when compared to the findings proposed in [8].

The outline of the paper is as follows. After the estimation problem has been formulated in Section II, a discussion on unbiased versus biased estimation is carried out in Section III. In Section IV, the optimal design of the decentralized estimation is presented. An implementation of the distributed algorithm is proposed in section V, and numerical results are given in Section VI. Finally, section VII concludes the paper with future perspectives.

Notation: Given a stochastic variable x we denote with $\mathbb{E}\,x$

its expected value. With $\|.\|$ we denote the ℓ^2-norm of a vector or the spectral norm of a matrix. Given a matrix K we refer to its largest singular value with the symbol $\bar{\gamma}(K)$. We say that $a \propto b$ if $a = k\,b$ with $k > 1$. With I and $\mathbb{1}$ we denote the identity matrix and the vector $(1, \ldots, 1)^T$, respectively, of appropriate dimensions.

II. PROBLEM FORMULATION

Consider $N > 1$ sensor nodes randomly placed on the space. We assume that each node can measure a common scalar signal $d(t)$ corrupted by additive noise:

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

with $t \in \mathbb{N} \cup \{0\}$ and where $v_i(t)$ is zero-mean white noise. Let us collect measurements and noise variables in vectors, $u(t) = (u_1(t), \ldots, u_N(t))^T$ and $v(t) = (v_1(t), \ldots, v_N(t))^T$, so that we can rewrite previous equation in the following compact form

$$u(t) = d(t)\,\mathbb{1} + v(t), \qquad t = 0, 1, \ldots.$$

The covariance matrix of $v(t)$ is supposed to be diagonal $\Sigma = \sigma^2 I$, so that $v_i(t)$ and $v_j(t)$, for $i \neq j$, are uncorrelated. With these assumptions it is clear that the additive noise can be averaged out only if nodes communicate measurements and estimates. Note that the communication rate of the measurements and estimates should be just fast enough to track the variations of the signal $d(t)$, but it is not required to be faster for performance improvement. Indeed, reducing the sampling time is, in general, not beneficial because measurements may be affected by an auto-correlated noise.

It is convenient to model the communication network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the vertex set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the edge set. We will assume that if $(i, j) \in \mathcal{E}$ then $(j, i) \in \mathcal{E}$, namely the graph is undirected. The graph $\mathcal{G}$ is said to be connected if there is a sequence of edges in $\mathcal{E}$ that can be traversed to go from any vertex to any other vertex. In the following we will denote the set of neighbors of node $i \in \mathcal{V}$ as

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

The estimation algorithm we propose is such that a node i computes an estimate $x_i(t)$ of $d(t)$ by taking a linear combination of neighboring estimates and measures

$$x_i(t) = \sum_{j \in \mathcal{N}_i} k_{ij}(t) x_j(t-1) + \sum_{j \in \mathcal{N}_i} h_{ij}(t) u_j(t). \quad \text{(II.1)}$$

For node i, the algorithm is initialized with $x_j(0) = u_i(0)$ for all $j \in \mathcal{N}_i$. In vector notation, we have

$$x(t) = K(t)x(t-1) + H(t)u(t). \quad \text{(II.2)}$$

Note that the matrices $K(t)$ and $H(t)$ can be interpreted as the adjacency matrices of two weighted time-varying graphs compatible with a given $\mathcal{G}$ representing the underlying communication network.

Given a wireless sensor network modelled by a connected graph $\mathcal{G}$ we consider the following design problem: find time-varying matrices $K(t)$ and $H(t)$, compatible with $\mathcal{G}$, such that

the signal $d(t)$ is consistently estimated and the variance of the estimate is minimized. Moreover, the solution should be distributed in the sense that the computation of $k_{ij}(t)$ and $h_{ij}(t)$ should be performed by node i. In the next sections we will review some of the results presented in a previous work by the authors of [8] and extend the analysis of the estimator.

III. UNBIASED VERSUS BIASED ESTIMATOR

Let us consider the estimation error $e(t) = x(t) - d(t)\,\mathbb{1}$. We have

$$e(t) = K(t)e(t-1) + K(t)d(t-1)\,\mathbb{1} + (H(t) - I)d(t)\,\mathbb{1} + H(t)v(t).$$

Let us assume that $d(t) = d(t-1) + \delta(t)$. Taking the expected value with respect to the stochastic variable $v(t)$, we obtain

$$\mathbb{E}\,e(t) = K(t)\,\mathbb{E}\,e(t-1) + d(t-1)(K(t) + H(t) - I)\,\mathbb{1} + \delta(t)(H(t) - I)\,\mathbb{1}. \quad \text{(III.1)}$$

We then have the following result.

Proposition 3.1: Let us consider the system (III.1). Assume that $(K(t) + H(t))\,\mathbb{1} = \mathbb{1}$ and $\bar{\gamma}(K(t)) < 1$ for all $t \geq 0$. Then we have that:

i) If $\delta(t) < \infty$ and $H(t)\,\mathbb{1} = \mathbb{1}$, for any $t \geq 0$, the system is uniformly exponentially stable.

ii) For $|\delta(t)| < \Delta$, for all $t \geq 0$, the system trajectories uniformly exponentially converge to a neighborhood of the origin whose size is proportional to Δ, namely $\mathbb{E}\,e(t) \to \mathcal{I}(\Delta)$ for $t \to +\infty$ where $\mathcal{I}(\Delta) = \{y \in \mathbb{R} : |y| \propto \Delta\}$.

Proof: If $(K(t) + H(t))\,\mathbb{1} = \mathbb{1}$ then the system equation reduces to

$$\mathbb{E}\,e(t) = K(t)\,\mathbb{E}\,e(t-1) + \delta(t)(H(t) - I)\,\mathbb{1}. \quad \text{(III.2)}$$

i) If also it holds that $H(t)\,\mathbb{1} = \mathbb{1}$ then (III.2) simply becomes $\mathbb{E}\,e(t) = K(t)\,\mathbb{E}\,e(t-1)$, since by hypothesis $\delta(t) < \infty$ for all t. Let us consider the Lyapunov function $V(t-1) = \|\mathbb{E}\,e(t-1)\|$. Then we have that

$$V(t) - V(t-1) = \|\mathbb{E}\,e(t)\| - \|\mathbb{E}\,e(t)\|$$
$$\leq (\|K(t)\| - 1)\|\mathbb{E}\,e(t-1)\|$$

which is negative if and only if $\|K(t)\| = \bar{\gamma}(K(t)) < 1$.

ii) In this case we have that $H(t)\,\mathbb{1} - \mathbb{1} = -K(t)\,\mathbb{1}$ and thus the system equation becomes

$$\mathbb{E}\,e(t) = K(t)\,\mathbb{E}\,e(t-1) - \delta(t)K(t)\,\mathbb{1}.$$

Let us consider again the Lyapunov function $V(t-1) = \|\mathbb{E}\,e(t-1)\|$. We have

$$V(t) - V(t-1) \leq (\|K(t)\| - 1)\|\mathbb{E}\,e(t-1)\| + \|K(t)\|N\Delta,$$

where we have used the fact that $|\delta(t)| \leq \Delta$. It is easy to see that, in this case, $V(t) - V(t-1) < 0$ if

$$\|\mathbb{E}\,e(t-1)\| > \frac{\bar{\gamma}(K(t))}{1 - \bar{\gamma}(K(t))}N\Delta \quad \text{(III.3)}$$

with $\bar{\gamma}(K(t)) < 1$. Thus as $t \to +\infty$ then $\mathbb{E}(t-1)$ tends to the interval $\mathcal{I}(\Delta) = N\Delta\bar{\gamma}(K(t))/(1 - \bar{\gamma}(K(t)))$. ∎

If we want to design an asymptotic unbiased linear estimator then the conditions in 3.1(i) need to be satisfied. Under this conditions, it is possible to show that the variance is minimized if $K(t) = 0$ and with $H(t)$ such that

$$h_{ij}(t) = h_{ji}(t) = \begin{cases} \dfrac{1}{|\mathcal{N}_i|} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise}. \end{cases}$$

Notice that in this case the nodes do not use any previous information about the signal $d(t)$ and that the error variance at each node is proportional to its neighborhood size.

However, if the signal $d(t)$ is slowing varying, then, under the assumptions of 3.1(ii), it is possible to guarantee that $\|\mathbb{E}\,e(t-1)\|$ tends to a neighborhood of the origin. As we can notice in (III.3) the size of this cumulative bias is small if the maximum signal variation is such that

$$\Delta < \frac{1 - \bar{\gamma}(K(t))}{\bar{\gamma}(K(t))N}.$$

We remark that the bound is rather conservative and one could try to find a different Lyapunov function which would yield a tighter bound. However, we show in the next sections how we can determine the matrices $K(t)$ and $H(t)$ so that both bias and variance are minimized.

IV. DESIGN OF THE DECENTRALIZED ESTIMATOR

A. Variance minimization

In order to design a decentralized estimator to track the signal $d(t)$, we consider the error covariance matrix

$$P(t) = \mathbb{E}\,(e(t) - \mathbb{E}\,e(t))(e(t) - \mathbb{E}\,e(t))^T,$$

where we assume that $x(t-1)$ and $u(t)$ are independent stochastic variables. Using the state equation (III.1) we have that the covariance update can be computed as

$$P(t) = K(t)P(t-1)K(t)^T + \sigma^2 H(t)H(t)^T. \quad \text{(IV.1)}$$

The problem is then to determine $K(t)$ and $H(t)$ so that, given the covariance matrix $P(t-1)$, the covariance $P(t)$ is minimized. A centralized solution can be obtained through a semi-definite program, as discussed more in [8], but it is not interesting here since our goal is to find a distributed algorithm.

Let M_i denote the number of neighbors of node i, i.e., M_i is the cardinality of $\mathcal{N}_i = \{i_1, \ldots, i_{M_i}\}$. Collect the estimation errors available at node i in the vector $\epsilon_i \in \mathbb{R}^{M_i}$. The elements of ϵ_i are ordered according to the node indices:

$$\epsilon_i = (e_{i_1}, \ldots, e_{i_{M_i}})^T, \qquad i_1 < \cdots < i_{M_i}.$$

Similarly, we introduce vectors $\kappa_i(t), \eta_i(t) \in \mathbb{R}^{M_i}$ corresponding to the non-zero elements of row i of the matrices $K(t)$ and $H(t)$, respectively, and ordered according to node indices. It follows from (IV.1) that the variance of $e_i(t)$ can be evaluated as

$$\mathbb{E}\,e_i(t)^2 = \kappa_i(t)^T\Gamma_i(t-1)\kappa_i(t) + \sigma^2\eta_i(t)^T\eta_i(t), \quad \text{(IV.2)}$$

where $\Gamma_i(t) = \mathbb{E}\,\epsilon_i(t)\epsilon_i(t)^T$. To minimize the variance of the estimation error in each node, we propose that $\kappa_i(t)$ and $\eta_i(t)$ are chosen to minimize (IV.2). In order to minimize the variance at each node and have convergence as well, the following optimization problem should be solved at each time t and in each node i:

$$\min_{\kappa_i(t),\eta_i(t)} \quad \kappa_i(t)^T\Gamma_i(t)\kappa_i(t) + \sigma^2\eta_i(t)^T\eta_i(t) \quad \text{(IV.3)}$$

$$\text{s.t.} \quad (\kappa_i(t) + \eta_i(t))^T \mathbb{1} = 1 \quad \text{(IV.4)}$$

$$\bar{\gamma}(K(t)) < 1. \quad \text{(IV.5)}$$

Note, however, that the inequality constraint (IV.5) is global, since $K(t)$ depends on all $\kappa_i(t)$, $i = 1, \ldots, N$. To get a distributed solution, we consider the following relaxed optimization problem

$$\min_{\kappa_i(t),\eta_i(t)} \quad \kappa_i(t)^T\Gamma_i(t)\kappa_i(t) + \sigma^2\eta_i(t)^T\eta_i(t) + \beta\kappa_i(t)^T\kappa_i(t)$$

$$\text{s.t.} \quad (\kappa_i(t) + \eta_i(t))^T \mathbb{1} = 1$$

where we have removed the constraint (IV.5), but we have modified the cost function. The parameter $\beta > 0$ is a parameter that enforce $\|\kappa_i(t)\|^2$ to be small. Since the size of this norm is related to the larger singular value of $\bar{\gamma}(K(t))$ by Gershgorin's theorem and this, on its turn, to the cumulative bias, as in (III.3), then the β parameter, roughly speaking, allows a tradeoff between variance and bias.

The optimization problem we posed is tractable since the cost function is convex and the constraint is linear. Strong duality holds and each node i can compute the optimal weights as

$$\kappa_i(t) = \frac{(\Gamma_i(t-1) + \beta I)^{-1}\mathbb{1}}{\sigma^{-2}M_i + \mathbb{1}^T(\Gamma_i(t-1) + \beta I)^{-1}\mathbb{1}} \quad \text{(IV.6)}$$

$$\eta_i(t) = \frac{\mathbb{1}}{M_i + \sigma^2\mathbb{1}^T(\Gamma_i(t-1) + \beta I)^{-1}\mathbb{1}}. \quad \text{(IV.7)}$$

Note that the covariance matrix $\Gamma_i(t-1)$ needs to be known at node i so that it can compute the weights. Since we assume that the nodes can exchange only estimates and measurements the covariance matrix needs to be estimated from data.

The estimator developed so far is shown as block diagram on the left of Figure 1. Notice that the filter has as input the measurement vector $u(t)$ and that it uses the previous error covariance in order to adapt the filter gains.

B. Estimation of error covariance

Since the estimator is a discrete linear time-varying system and the stochastic process $x(t)$ and thus $e(t)$ is not stationary, the estimation of the error covariance is not an easy task. However, if we consider the signals in the quasi-stationary sense, estimation based on samples guarantees to give good results. In particular we give the following definition.
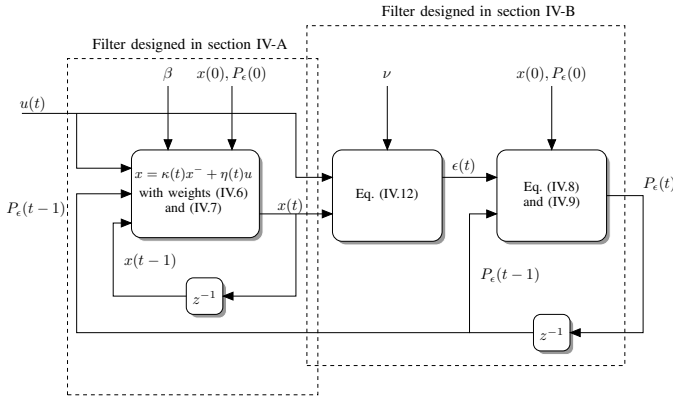
Fig. 1. Block diagram of the proposed estimator. Notice that it is a cascade of two sub-systems. The one on the left is adaptive and it produces a good estimate of the $d(t)$ with low variance and small bias. The sub-system on the right, instead, estimates the error covariance.

Definition 4.1 ([9, pag. 34]): A signal $s(t)$ is said to be quasi-stationary if it is such that

(i) $\mathbb{E}\, s(t) = m_s(t)$, $|m_s(t)| \leq C$ for all t
(ii) $\mathbb{E}\, s(t)s(r) = R_s(t,r)$, $|R_s(t,r)| \leq C$ for all t and

$$\lim_{N \to +\infty} \frac{1}{N} \sum_{t=1}^{N} R_s(t, t-\tau) = R_s(\tau)$$

for all τ, and where $C \in \mathbb{R}$ is constant.

It turns out that if a quasi-stationary signal is the input of a uniformly exponentially stable time-varying linear system then its output is also quasi-stationary [10]. In particular in our case the input signal is the measurement signal $u(t)$ which is (component-wise) stationary and ergodic and thus also quasi-stationary. This implies that also $x(t)$ is quasi-stationary (component-wise) since it is the output of a uniformly exponentially stable time-varying linear system. We can estimate the error covariance using the sample covariance. In particular we have that the sample mean $\hat{m}_{\epsilon_i}(t)$ and covariance estimate of $\hat{\Gamma}_i(t)$ can then be computed as

$$\hat{m}_{\epsilon_i}(t) = \frac{1}{t} \sum_{i=0}^{t} \hat{\epsilon}_i(t) \qquad \text{(IV.8)}$$

$$\hat{\Gamma}_i(t) = \frac{1}{t} \sum_{i=0}^{t} (\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t))(\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t))^T \qquad \text{(IV.9)}$$

where $\hat{\epsilon}_i(t) = x^i(t) - \hat{d}$. Thus the problem reduces to design an estimate of $\hat{d}(t)$ from data. A node i in the network has available the estimates $x_{i_j}(t)$ and measurements $u_{i_j}(t)$ with $i_j \in \mathcal{N}_i$. Let $x^i(t)$ and $u^i(t)$ be the collection of all these values. From the design of the estimator described in the previous section we can model this data set as

$$x^i(t) = d(t)\,\mathbb{1} + \xi(t) + w(t)$$
$$u^i(t) = d(t)\,\mathbb{1} + v(t)\,,$$

where $\xi(t) \in \mathbb{R}^{M_i}$ models the bias of the estimates and $w(t)$ is zero-mean Gaussian noise modelling the variance of the

estimator. Roughly speaking, node i has available $2M_i$ data values in which half of the data are corrupted by a small biased term $\xi(t)$ and a low variance noise $w(t)$ and the other half is corrupted by zero-mean Gaussian noise $v(t)$ with rather high variance.

It is clear that using only $u^i(t)$ to estimate $\hat{d}(t)$ we can obtain an unbiased estimate of $d(t)$, however its covariance is rather large since M_i is typically small. Using only measurements we then over-estimate the error covariance and this results in poor performance. On the other hand using $x^i(t)$ only determines an under-estimate of the covariance, which rapidly makes the weights $\eta_i(t)$ vanish. In this case the signal measurements are discarded and thus tracking becomes impossible. In order to use both $x^i(t)$ and $u^i(t)$ we pose a least square problem as follows[1]:

$$\min_{\hat{d},\hat{\xi}} \quad \left\| \begin{pmatrix} x^i \\ u^i \end{pmatrix} - A \begin{pmatrix} \hat{d} \\ \hat{\xi} \end{pmatrix} \right\|^2$$
$$\text{s.t.} \quad \|\xi\|^2 \leq \rho$$

where the matrix A is the signal model, namely

$$\begin{pmatrix} \hat{x}^i \\ \hat{u}^i \end{pmatrix} = \begin{pmatrix} \mathbb{1} & I \\ \mathbb{1} & 0 \end{pmatrix} \begin{pmatrix} \hat{d} \\ \hat{\xi} \end{pmatrix},$$

and the constraint can be represented as

$$\left\| \begin{pmatrix} 0 & I \end{pmatrix} \begin{pmatrix} \hat{d} \\ \hat{\xi} \end{pmatrix} \right\|^2 = \left\| B \begin{pmatrix} \hat{d} \\ \hat{\xi} \end{pmatrix} \right\|^2 \leq \rho\,.$$

However, the problem is difficult to solve in a closed form [11]. Furthermore, previous formulation has the following disadvantages: (1) requires heavy numerical algorithms for determine the optimal solution (typically SVD decomposition), (2) the value of ρ is not known in advance. We thus consider the following regularized problem

$$\min_{\hat{d},\hat{\xi}} \quad \left\| \begin{pmatrix} x^i \\ u^i \end{pmatrix} - A \begin{pmatrix} \hat{d} \\ \hat{\xi} \end{pmatrix} \right\|^2 + \nu \left\| B \begin{pmatrix} \hat{d} \\ \hat{\xi} \end{pmatrix} \right\|^2 \qquad \text{(IV.10)}$$

where $\nu > 0$ is a parameter whose choice is typically rather difficult. We propose here to use the Generalized Cross-Validation method [12]. This consists in choosing

$$\nu = \arg \min \phi(\nu)$$

where

$$\phi(\nu) = \frac{\|(A^T A + \nu B^T B)^{-1}(x^i, u^i)^T\|}{\text{tr}\,(A^T A + \nu B^T B)^{-1}}\,.$$

Taking the derivative of the argument of (IV.10), after some algebraic efforts, the solution of the optimization problem is

$$(\hat{d},\hat{\xi})^T = (x^i, u^i)^T A \left(A^T A + \nu B^T B\right)^{-1}\,.$$

The inverse of the matrix in the previous equation can be computed in closed form using the following result:

[1]We disregard in the following the time dependence both because all the variable are considered at a certain time instance t and to keep the notation lighter.

Proposition 4.2: If $\nu > 0$ then

$$\left(A^T A + \nu B^T B\right)^{-1} =$$
$$\frac{1}{M_i(1+2\nu)} \begin{pmatrix} 1+\nu & -\mathbb{1}^T \\ -\mathbb{1} & \dfrac{M_i(1+2\nu)I + \mathbb{1}\mathbb{1}^T}{1+\nu} \end{pmatrix} \quad \text{(IV.11)}$$

Proof: By Schur's complement we obtain

$$\left(A^T A + \lambda a^T a\right)^{-1} =$$
$$\begin{pmatrix} \left(2M_i - \dfrac{M_i}{1+\lambda}\right)^{-1} & \mathbb{1}^T(\mathbb{1}\mathbb{1}^T - 2M_i(1+\lambda)I)^{-1} \\ (\mathbb{1}\mathbb{1}^T - 2M_i(1+\lambda)I)^{-1}\mathbb{1} & \left((1+\lambda)I - \dfrac{\mathbb{1}\mathbb{1}^T}{2M_i}\right)^{-1} \end{pmatrix}$$

From [13] it follows that

$$\left((1+\lambda)I - \frac{\mathbb{1}\mathbb{1}^T}{2M_i}\right)^{-1} = \frac{I}{1+\nu} + \frac{\mathbb{1}\,\mathbb{1}^T}{M_i(1+2\nu)(1+\nu)},$$

It is easy from here to show that the resulting matrix is (IV.11). ∎

Since we are interested to know $\epsilon_i(t) = x^i - d\,\mathbb{1}$, we observe that an estimate of this difference is $\hat{\xi}$. This is given by

$$\hat{\xi} = \frac{x^i}{1+\nu} - \frac{\nu\,\mathbb{1}^T x^i + (1+\nu)\,\mathbb{1}^T u^i}{M_i(1+2\nu)(1+\nu)}\,\mathbb{1} \quad \text{(IV.12)}$$

The covariance estimator we have developed in this section is depicted on the right of the block diagram shown in Figure 1. Thus, using $\hat{\xi}$ as an estimate of the estimation error we compute the error covariance from samples.

As it appears clearly from Figure 1, the total estimator is a cascade of two sub-systems. One, on the left, which is adaptive, and is used to produce a good estimate of the $d(t)$ and the other that provides a good estimate of the error covariance.

V. IMPLEMENTATION

The implementation of the estimation algorithm is shown as Algorithm 1. First, each sensor initializes the local mean estimation error $\hat{m}_\epsilon = 0$ (see line 2) and its local covariance matrix estimate with the noise covariance, i.e. $\hat{\Gamma}_i(0) = \sigma^2 I$ (see line 2 in the algorithm), where we remark that we are using the "hat" since these are sample estimates of the real mean and covariance of the error. The optimal weights are computed using equations (IV.6) and (IV.7) (lines 8 and 9). In line 11 we compute the optimal estimate. Lines from 12 to 15 implement the covariance update based on the available data by solving the constrained least-square considered in the end of subsection IV-B. Sample mean and covariance of the error are updated in lines 14 and 15. Notice that we used a recursive way of computing (IV.8) and (IV.9). In the algorithm, the inversion of the covariance matrix should be computed. This is not a difficult operation in resource constrained sensor networks, since each node has generally a rather limited number of neighbors, and thus the size of the matrix $\hat{\Gamma}_i$ is small.

Algorithm 1 Estimation algorithm for node i
1. $t := 0$
2. $\hat{m}_\epsilon := 0$
3. $\hat{\Gamma}_i(0) := \sigma^2 I$
4. $x_i(0) := u_i(0)$
5. **while** forever **do**
6. $\quad M_i := |\mathcal{N}_i|$
7. $\quad t := t+1$
8. $\quad \kappa_i(t) := \dfrac{(\hat{\Gamma}_i(t-1) + \beta I)^{-1}\mathbb{1}}{\sigma^{-2}M_i + \mathbb{1}^T\hat{\Gamma}_i(t-1) + \beta I)^{-1}\mathbb{1}}$
9. $\quad \eta_i(t) := \dfrac{\mathbb{1}}{M_i + \sigma^2\,\mathbb{1}^T(\hat{\Gamma}_i(t-1)+\beta I)^{-1}\mathbb{1}}$
10. $\quad$ Let $j \in \mathcal{N}_i$
11. $\quad x_i(t) := \sum_j \kappa_{i_j}(t)x_j(t-1) + \eta_{i_j}(t)u_j(t)$
12. $\quad \hat{\xi} := \dfrac{x^i}{1+\nu} - \dfrac{\nu\,\mathbb{1}^T x^i + (1+\nu)\,\mathbb{1}^T u^i}{M_i(1+2\nu)(1+\nu)}\mathbb{1}$
13. $\quad \hat{\epsilon}_i := \hat{\xi}$
14. $\quad \hat{m}_{\epsilon_i}(t) := \dfrac{t-1}{t}\hat{m}_{\epsilon_i}(t-1) + \dfrac{1}{t}\hat{\epsilon}_i(t)$
15. $\quad \hat{\Gamma}_i(t) := \dfrac{t-1}{t}\hat{\Gamma}_i(t-1) + \dfrac{1}{t}(\hat{\epsilon}_i(t)-\hat{m}_{\epsilon_i}(t))(\hat{\epsilon}_i(t)-\hat{m}_{\epsilon_i}(t))^T$
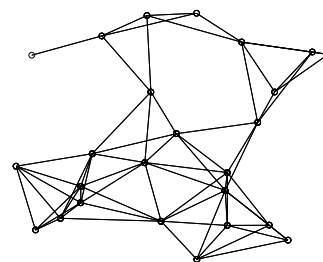16. **end while**

Fig. 2. The topology of the network with $N = 25$ nodes used in the simulations.

Note that the algorithm is implemented under the assumption that each node is able to compute and communicate data within the sampling instance.

VI. NUMERICAL RESULTS

Numerical simulations have been carried out in order to validate performance of different solutions. In particular we have compared the estimator proposed in this paper with an estimator that takes as estimate an average of the received information from the neighbors, which we call here average estimator and the estimator previously proposed in [8]. We have simulated a network with $N = 25$ nodes randomly deployed on a squared area of size $N/2$ and such that two nodes were able to communicate if and only if their relative distance was less or equal to $1.6\sqrt{N}$. The generated network is shown in Figure 2. The average number of neighbors for each node is about 6.2. The signal to track was

$$d(t) = 3\sin\left(\frac{2\pi t}{1000}\right) - 2.5\cos\left(\frac{2\pi t}{1500}\right)\cdot\sin\left(\frac{2\pi t}{1000}\right)^{2/3}.$$

The measurement noise was with variance $\sigma^2 = 1.2$.

The realizations for all 25 nodes are shown in Figure 3. The first plot shows the raw measurements, the second the output of the average estimator, the third the output of the decentralized estimator proposed in [8], and in the last plot the output of the new proposed estimator is reported. The performance improvement is clearly visible in Figure 4, where the same realizations are shown between $t = 1000$ and $t = 2000$. Numerical comparisons give:

Estimator	Std. dev. MSE	β	ρ
Average	0.276	-	-
Previous ([8])	0.178	0.0	-
Present paper	0.139	0.03	0.7

Thus the proposed decentralized estimator yields an improvement of about 50% with respect to the average estimator, and of about 21% with respect to the estimator we proposed in [8]. The choice of the filter parameters β and ρ is in general dependent on the maximum signal variation. Simulations shows the following rules of thumb:

  i) Choose β so that the maximum singular value of $K(t)$ is bounded away from 1. In particular some preliminary tests shows that $\beta = 0.01$ to 0.05 bounds the value of $\bar{\gamma}(K(t))$ in the range 0.65 to 0.8 rather independently on the size of the graph and its connectivity for a given Δ.

 ii) The computation of ν might be demanding on resource limited sensor nodes. However, observe that the matrix $(A^T A + \nu B^T B)^{-1}$ can be computed in closed form, so one can create a lookup table of $\phi(\nu)$ for different values of ν. Notice that the matrix is small, $(M_i + 1) \times (M_i + 1)$ and that it can be stored efficiently because of is particular structure. Given a x^i and u^i one can then search for the minimum ν in an efficient way.
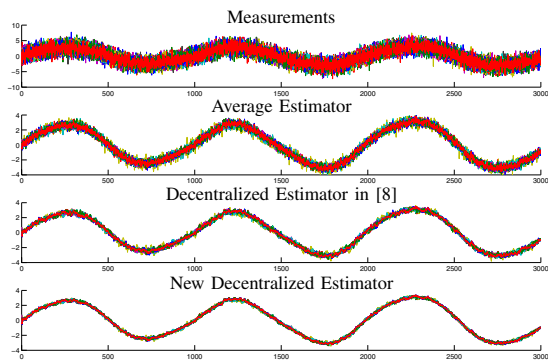

Fig. 3. Realization of all the estimates of the $N = 25$ nodes of the network, versus time. The first plot show the measurements. The second shows the average estimator, the second the distributed estimator proposed in [8] and the last shows the new proposed estimator. The noise variance is of 1.2.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an improved version of the decentralized cooperative estimation algorithm proposed
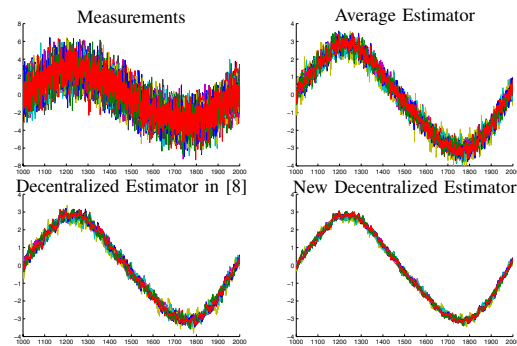

Fig. 4. Zoom of the realizations in Figure 3 in the interval $t \in [1000, 2000]$.

in [8], for the estimation of time-varying signal using a wireless sensor network. A mathematical framework is proposed to design a cascade structured filter. The filter is supposed to run locally in each node. The proposed estimator depends on two parameters. Some rule-of-thumb and numerical procedure are suggested in order to choose the parameter values.

Future work includes the refinement of some of the bounds we have used in the design in order to provide a more precise guidance in the choice of the filter parameters. Stability analysis of the filter and robustness to packet-losses are currently under investigation.

REFERENCES

[1] J. J. Xiao, A. Riberio, Z.-Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wirless sensor netowrks," *IEEE Signal Processing Magazine*, 2006.
[2] Z.-Q. Luo and J. Liu, "Distributed signal processing in sensor netorks," *IEEE Signal Processing Magazine*, vol. 14, 2006.
[3] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks," *IEEE Signal Processing Magazine*, 2005.
[4] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *In Proc. of IEEE IPSN*, 2005.
[5] L. Xiao, S. Boyd, and S. J. Kim, "Distributed average consensus with least-mean-square deviation," *Submitted to Journal of Parallel and Distributed Computing*, 2006.
[6] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *System Control Letter*, 2004.
[7] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," *In Proc. of IEEE CDC*, 2005.
[8] A. Speranzon, C. Fischione, and K. E. Johansson, "Distributed and collaborative estimation over wireless sensor networks," *Proceedings of IEEE Conference on Decision and Control*, 2006.
[9] L. Ljung, *System Identification: Theory for the User (2nd Edition)*. Prentice Hall PTR, 1998.
[10] L. Ljung and B. Walhberg, "Asymptotic properties of the least-squares method for estimating transfer functions and disturbance spectra," *Advances in Applied Probability*, vol. 24, no. 2, 1992.
[11] G. H. Golub and U. Von Matt, "Quadratically constrained least squares and quadratic problems," *Numerische Mathematik*, vol. 59, no. 6, 1991.
[12] ——, "Tikhonov regularization for large scale problems," Stanford SCCM, Stanford, Tech. Rep. 97-03, 1997.
[13] K. S. Miller, "On the inverse of the sum of matrices," *Mathematics Magazine*, vol. 54, no. 2, 1981.