On Distributed Estimation for Sensor Networks

Alberto Speranzon, Carlo Fischione, and Karl Henrik Johansson

***Abstract*—Distributed estimators for sensor networks are discussed. The considered problem is on how to track a noisy time-varying signal jointly with a network of sensor nodes. We present a recent scheme in which each node computes its estimate as a weighted sum of its own and its neighbors' measurements and estimates. The weights are adaptively updated to minimize the variance of the estimation error. Theoretical and practical properties of the algorithm are illustrated. The results provide a tool to trade-off communication constraints, computing efforts and estimation quality.**

I. Introduction

A wireless sensor network (WSN) is a network of autonomous devices that can sense their environment, make computations and communicate over radio with neighboring devices. WSNs have a growing domain of application in areas such as environmental monitoring, industrial automation, intelligent buildings, search and surveillance, and public transportation [1]–[3]. Today they are mostly used for monitoring and diagnosis, but their potential capability goes beyond that since they can provide real-time information for closed-loop control systems [4], [5]. The characteristics of WSNs motivate the development of new classes of distributed estimation and control algorithms, which explore these systems' limited power, computing and communication capabilities. It is important that the algorithms have tuning parameters that can be adjusted according to the demands set by the applications. In this paper, we investigate such a distributed estimation algorithm for tracking an unknown time-varying physical variable.

The main contribution of this paper is a novel distributed minimum variance estimator. A noisy time-varying signal is jointly tracked by a WSN, in which each node computes an estimate as a weighted sum of its own and its neighbors' measurements and estimates. The filter weights are time varying and updated locally. The filter has a cascade structure with an inner loop producing the state estimate and an outer loop producing an estimate of the error covariance. The state estimate is obtained as the solution of an optimization problem with quadratic cost function and quadratic constraints. We show that the problem has a distributed implementation with conditions that can be checked locally. It is argued that the estimator is practically stable if the signal to track is slowly varying, so the estimate of each node converges to a neighborhood of the signal to track. The estimate in each node has consequently a small variance and a small bias. A bound on the estimation error variance, which is linear in the measurement noise variance and decays with the number of neighboring nodes, is presented. The algorithm is thus characterized by a trade-off between the amount of communication and the resulting estimation quality. Compared to similar distributed algorithms in the literature, the one presented in this paper gives better estimates, but to the cost of an increased computational complexity. This is illustrated in the implementation discussion and the computer simulations in the latter part of the paper. An extended version of the current paper with proofs and further discussion has been submitted for journal publication [6]. Early versions of the results see [7], [8].

Distributed signal processing is a very active research area due to the recent developments in wireless networking and computer and sensor technologies. The estimator presented in this paper has two particular characteristics: it does not rely on a model of the signal to track, and its filter coefficients are time varying. It is related to recent contributions on low-pass filtering by diffusion mechanisms, e.g., [7]–[14]. Many of these papers focus on diffusion mechanisms to have each node of the network obtaining the average of the initial samples of the network nodes. Major progress has been made in understanding how the convergence behavior of these consensus or state-agreement problems. It is not straightforward to carry over this work to the problem of tracking a time-varying signal. An attempt is made in [12], where a scheme for sensor fusion based on a consensus filter is proposed. Each node computes a local weighted least-squares estimate and the authors show it converges to the maximum-likelihood solution for the overall network. An extension of this approach is presented in [15], where the authors study a distributed average computation of a time-varying signal, when the signal is affected by a zero-mean noise. A convex optimization problem is posed to derive the edge weights, which each node uses to minimize the least mean square deviation of the estimates. The same linear filter is also considered in [16], where the weights are computed off-line to speed up the computation of the averages. Further characterization of consensus filters for distributed sensor fusion is given in [14].

Another approach to distributed estimation is to use nonlinear filters based on self-synchronization and coupling functions, e.g., [17]–[20]. In this case, the estimate of each node is provided by the state of a nonlinear dynamical system. This

The work by A. Speranzon was partially supported by the European Commission through the Marie Curie Transfer of Knowledge project BRIDGET (MKTD-CD 2005 029961). The work by C. Fischione and K. H. Johansson was done in the framework of the HYCON NoE, contract number FP6-IST-511368, and RUNES IP, contract number FP6-IST-004536. It was partially funded also by the Swedish Foundation for Strategic Research and by the Swedish Research Council.

A. Speranzon is with Unilever R&D Port Sunlight, Quarry Road East Bebington, Wirral, CH63 3JW, United Kingdom. E-mail: alberto.speranzon@unilever.com. C. Fischione and K. H. Johansson are with School of Electrical Engineering, Royal Institute of Technology, Osquldas väg 10, 100-44 Stockholm, Sweden. E-mails: {carlofi,kallej}@ee.kth.se.

system is coupled to some of the other nodes by a static coupling function. Some conditions on the coupling function that lead to state synchronization asymptotically is investigated in [20].

Distributed filtering using model-based approaches are studied in various wireless network contexts, e.g., [21]–[25]. One possible approach is using distributed Kalman filters. More recently there are attempts to mix the diffusion mechanism, discussed previously, with distributed Kalman filtering, e.g., [13], [26]. A plausible approach is to communicate the estimates of the local Kalman filters, and then average these values using a diffusion strategy.

Let us briefly summarize the originality of our approach compared to the literature. First, note that our estimator tracks a time-varying signal, while [10]–[12] are limited to averaging initial samples. Our approach does not require a model of the system that generates the signal to track, in contrast to model-based approaches, e.g., [13], [25]. We do not impose a pre-assigned coupling law among the nodes as in [20]. Compared to [12]–[14], we do not rely on the Laplacian matrix associated to the communication graph, but consider a more general model of the filter structure. Moreover, our filter parameters are computed through distributed algorithms, whereas for example [15] and [16] rely on centralized algorithms for designing the filters. Note that in the early versions of our contribution [7], [8], we extended the algorithms in [12]–[14] by designing the filter weights such that the variance of the estimation errors is minimized. In the current paper, we improve the filter design considerably and the performance limit of the filter is characterized.

The outline of the paper is as follows. Section II presents the distributed estimation problem considered throughout the paper. A centralized minimum variance optimization problem is given and its solution is characterized. The distributed estimator design is discussed in Section III. A distributed minimum variance optimization problem is given and its relation to the centralized problem is indicated. By restricting the set of admissible filter weights, it is possible to obtain a completely distributed solution, where convergence is guaranteed. A bound on the estimation error variance is computed. The latter part of Section III discusses estimation of the error covariance. Section IV presents the detail of the implementation of the estimation algorithm. Numerical results illustrating the performance of the proposed estimator and comparing it to some related proposals are also given. Finally, Section V concludes the paper.

Notation:: We denote the non-negative integers $\mathbb{N}_0 = \{0, 1, 2, \dots\}$. With $|\cdot|$ we denote either the absolute value or the cardinality, depending on the context. With $\|\cdot\|$ we denote the ℓ^2-norm of a vector and the spectral norm of a matrix. Given a matrix $A \in \mathbb{R}^{n\times n}$, we denote with $\lambda_r(A)$, $1 \le r \le n$, its r-th eigenvalue, with $\lambda_{\min}(A) = \lambda_1(A)$ and $\lambda_{\max}(A) = \lambda_n(A)$ being the minimum and maximum eigenvalue, respectively, where the order is taken with respect to the real part. We refer to its largest singular value as $\gamma_{\max}(A)$. The trace of A is denoted $\operatorname{tr} A$. With I and $\mathbb{1}$ we denote the identity matrix and the vector $(1, \dots, 1)^T$, respectively. Given a stochastic variable x we denote by $\mathbb{E}\, x$ its expected value. In order to keep light the notation, we disregard the time dependence when it is clear from the context.

II. Preliminaries

In this section we state the problem and we derive centralized conditions under which the estimation error converges. We then pose a centralized optimization problem that yields weights for minimum variance estimation. The centralized case is instructive for the design of the distributed estimator.

A. Problem Formulation

Consider $N > 1$ sensor nodes placed in random and static positions on the space. We assume that each node measures a common scalar signal $d(t)$ corrupted by additive noise:

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

with $t \in \mathbb{N}_0$ and where $v_i(t)$ is zero-mean white noise. Let us collect measurements and noise variables in vectors, $u(t) = (u_1(t), \dots, u_N(t))^T$ and $v(t) = (v_1(t), \dots, v_N(t))^T$, so that we can rewrite the previous equation as

$$u(t) = d(t)\,\mathbb{1} + v(t)\,, \qquad t \in \mathbb{N}_0\,.$$

The covariance matrix of $v(t)$ is supposed to be diagonal $\Sigma = \sigma^2 I$, so $v_i(t)$ and $v_j(t)$, for $i \neq j$, are uncorrelated. The additive noise, in each node, can be averaged out only if nodes communicate measurements or estimates. Note that the communication rate of the measurements and estimates should be just fast enough to track the variations of $d(t)$. Indeed, increasing the sampling rate, in general, is not beneficial because measurements might then be affected by auto-correlated noise.

It is convenient to model the communication network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ 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 $i \in \mathcal{V}$ plus the node itself as

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

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

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

We assume that neighboring estimates and measures are always successfully received, i.e., there are no packet losses.[1]

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

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

Note that the matrices $K(t)$ and $H(t)$ can be interpreted as the adjacency matrices of two graphs with time-varying weights. These graphs are compatible with the underlying communication network represented $\mathcal{G}$. We denote this as $K(t) \simeq \mathcal{G}$ and $H(t) \simeq \mathcal{G}$.

Given a WSN modelled as a connected graph $\mathcal{G}$, we have 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 locally by node i.

B. Convergence of the Estimation Error in the Centralized Scenario

Here we derive conditions on $K(t)$ and $H(t)$ that guarantee the estimation error to converge. Define the estimation error $e(t) = x(t) - d(t)\,\mathbb{1}$. Introduce $\delta(t) = d(t) - d(t-1)$, so that the error dynamics can be described as

$$\begin{aligned} e(t) = K(t)e(t-1) + d(t)(K(t) + H(t) - I)\,\mathbb{1} \\ - \delta(t)K(t)\,\mathbb{1} + H(t)v(t)\,. \end{aligned} \tag{II.3}$$

Taking the expected value with respect to the stochastic variable $v(t)$, we obtain

$$\begin{aligned} \mathbb{E}\,e(t) = K(t)\,\mathbb{E}\,e(t-1) + d(t)(K(t) + H(t) - I)\,\mathbb{1} \\ - \delta(t)K(t)\,\mathbb{1}\,. \end{aligned} \tag{II.4}$$

We have the following result.

Proposition 2.1: Consider the system Equation (II.3). Assume that

$$(K(t) + H(t))\,\mathbb{1} = \mathbb{1}\,, \tag{II.5}$$

and that there exists $0 \leq \gamma_0 < 1$ such that

$$\gamma_{\max}(K(t)) \leq \gamma_0 \tag{II.6}$$

for all $t \in \mathbb{N}_0$.

(i) If $H(t)\,\mathbb{1} = \mathbb{1}$, for all $t \in \mathbb{N}_0$, then

$$\lim_{t\to+\infty} \mathbb{E}\,e(t) = 0\,.$$

(ii) If $|\delta(t)| < \Delta$, for all $t \in \mathbb{N}_0$, then

$$\lim_{t\to+\infty} \|\,\mathbb{E}\,e(t)\| \leq \frac{\sqrt{N}\Delta\gamma_0}{1-\gamma_0}\,. \tag{II.7}$$

[1]This assumption is motivated by the fact that we assume the network is static, that appropriate channel and source coding are applied, and there is an Automatic Repeat Request (ARQ) protocol. These are natural assumptions in many WSN applications. Note that we implicitly assume that the sampling time between measures is long relative to the coherence time of the wireless channel coefficients, so there is enough time to detect and retransmit erroneous packets until they are successfully received. More details are given in Section III.

Proposition 2.1(i) provides the condition $H(t)\,\mathbb{1} = \mathbb{1}$ under which the estimate is unbiased. It is possible to show that in this case the variance is minimized if $K(t) = 0$ and

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

Note that nodes do not use any memory and that the error variance at each node is proportional to its neighborhood size. However, if $d(t)$ is slowly varying, then, under the assumptions of Proposition 2.1(ii), it is possible to guarantee that $\|\,\mathbb{E}\,e(t)\|$ tends to a neighborhood of the origin, but the estimate might be biased. Note also that $\|\,\mathbb{E}\,e(t)\|$ has the meaning of a cumulative bias, in the sense that it is a function of the sum of the N biases of individual nodes.

The size of the cumulative bias can be kept small with respect to the signal to track by defining a proper value of γ_0. In particular, Equation (II.7) can be related to the Signal-to-Noise Ratio (SNR) of the average of the estimate in an intuitive way as follows. Let P_d denote the average power of d and let P_b denote the desired power of the biases of the average of the estimates. Then, we define the desired SNR as $\text{SNR} = P_d/P_b$. Since there are N nodes, we consider the average SNR of each node as $\Upsilon = \text{SNR}/N$. Let us assume that we want the estimator to guarantee that the right-hand side of Equation (II.7) is equal to this desired $\sqrt{\text{SNR}}$. This is equivalent to that

$$\gamma_0 = \frac{\sqrt{\Upsilon}}{\sqrt{\Upsilon} + \Delta}\,.$$

The right-hand side is useful in the tuning of the estimator, so we denote it as $f(\Delta, \Upsilon)$. By choosing an appropriate Υ, we have a guaranteed convergence property of the estimator given by the corresponding $f(\Delta, \Upsilon)$. This function is particularly useful, since in next sections it will allow us to relate the size of the bias of estimates with the variations of the signal to track, and the stability of the estimates.

C. Centralized Variance Minimization

We show in this subsection how we can determine the matrices $K(t)$ and $H(t)$ so that the bias is kept small and the variance minimized. The error covariance matrix is given by

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

Using the error update Equation (II.3), we have that the covariance is updated according to

$$P(t) = K(t)P(t-1)K(t)^T + \sigma^2 H(t)H(t)^T\,, \tag{II.8}$$

where we use the fact that $x(t-1)$ and $u(t)$ are independent stochastic variables. We want to find $K(t)$ and $H(t)$ so that, given the covariance matrix $P(t-1)$, the covariance $P(t)$ is minimized. We consider the trace of $P(t)$ as a measure of the size of $P(t)$. It represents a cumulative error variance,

namely, the sum of the error variance at each node. We have the following optimization problem

$$\begin{aligned}\mathcal{P}_1: \quad \min_{K(t),H(t)} \quad & \mathrm{tr}\,(K(t)P(t-1)K(t)^T) \\ & + \sigma^2 \mathrm{tr}\,(H(t)H(t)^T) \\ \text{s.t.} \quad & (K(t)+H(t))\,\mathbb{1} = \mathbb{1}\,, \\ & \gamma_{\max}(K(t)) \le f(\Delta,\Upsilon)\,, \\ & K(t) \simeq \mathcal{G}\,, \quad H(t) \simeq \mathcal{G}\,.\end{aligned} \tag{II.9}$$

Notice that the objective function is quadratic in $K(t)$ and $H(t)$ for a given $P(t-1)$. The first constraint is the linear matrix equality (II.5). The second constraint, which ensures that the expected value of the estimation error converges to a neighborhood of zero, can be written as a linear matrix inequality using Schur complement [27]. The last two constraints, impose the structure of the matrices $K(t)$ and $H(t)$ to be compatible with the graph $\mathcal{G}$.

The cost function of problem $\mathcal{P}_1$ may suggest that it is possible to distribute the optimization by letting each node minimize its own error variance. This approach is impossible, however, because the nodes are coupled through the global constraints: $(K(t)+H(t))\,\mathbb{1} = \mathbb{1}$ and $\gamma(K(t)) \le \gamma_0$.

Although the optimization problem $\mathcal{P}_1$ can conceptually be solved using standard numerical optimization tools, it clearly requires a powerful central node collecting data, computing new weights, and dispatching them back to the nodes. There could also be large delays (due to multi-hop routing of data from nodes to the central unit), and large power consumptions, beside the typical disadvantage that centralized solutions are not fault tolerant.

In the following sections, we propose a fully decentralized solution, where each node computes its weights minimizing the variance of its estimate.

III. Distributed Estimator Design

In this section we describe how each node computes adaptive weights to minimize its estimation error variance. Starting from the centralized problem $\mathcal{P}_1$, we first show that we can transform the global constraints into distributed ones. The constraint $(K(t)+H(t))\,\mathbb{1} = \mathbb{1}$ is easily handled. It turns out that the constraint $\gamma_{\max}(K(t)) \le f(\Delta,\Upsilon)$ can be translated into a set of constraints of the type $\sum_{j\in\mathcal{N}_i} k_{ij}^2 \le \psi_i$, where ψ_i is a constant that can be computed locally by the nodes. Using these new constraints, we pose a optimization problem for finding optimal filter weights that minimize the error variance in each node. A complication is that the weights depend on the error covariance matrix, which is not available at each node. We end this section by discussing a way of estimating it.

A. Distributed Variance Minimization

Let $M_i = |\mathcal{N}_i|$, which denotes the number of neighbors of node i, including the node itself. 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:

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

Similarly, we introduce vectors $\kappa_i^T(t), \eta_i^T(t) \in \mathbb{R}^{M_i}$ corresponding to the non-zero elements of row i of the matrices $K(t)$ and $H(t)$, respectively, and ordered according to node indices. The expected value of the estimation error at node i can be written as

$$\mathbb{E}\, e_i(t) = \kappa_i^T(t)\,\mathbb{E}\,\epsilon_i(t-1) - \kappa_i^T(t)\delta(t)\,\mathbb{1}\,, \tag{III.1}$$

where we used the fact that $d(t) - d(t-1) = \delta(t)$ and that $(K(t)+H(t))\,\mathbb{1} = \mathbb{1}$. Note that the latter inequality is equivalent to that $(\kappa_i(t)+\eta_i(t))^T\,\mathbb{1} = 1$. We will assume that $e_i(0) = u_i(0)$. It follows that

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

where $\Gamma_i(t) = \mathbb{E}\,(\epsilon_i(t) - \mathbb{E}\,\epsilon_i(t))(\epsilon_i(t) - \mathbb{E}\,\epsilon_i(t))^T$. To minimize the variance of the estimation error in each node, we need to determine $\kappa_i(t)$ and $\eta_i(t)$ so that the previous expression is minimized at each time instance. We have the following optimization problem:

$$\mathcal{P}_2: \quad \min_{\kappa_i(t),\eta_i(t)} \quad \kappa_i^T(t)\Gamma_i(t-1)\kappa_i(t) + \sigma^2\eta_i^T(t)\eta_i(t) \tag{III.2}$$

$$\text{s.t.} \quad (\kappa_i(t)+\eta_i(t))^T\,\mathbb{1} = 1\,, \tag{III.3}$$

$$\gamma_{\max}(K(t)) \le f(\Delta,\Upsilon)\,. \tag{III.4}$$

Note that the inequality constraint (III.4) is still global, as $\gamma_{\max}(K(t))$ depends on all $\kappa_i(t)$, $i = 1,\dots,N$. We show next that it can be replaced by the local constraint

$$\|\kappa_i(t)\| \le \psi_i\,, \quad t \in \mathbb{N}_0\,, \tag{III.5}$$

where $\psi_i > 0$ is a constant that can be computed locally.

For $i = 1,\dots,N$, let us define the set $\Theta_i = \{j \ne i : \mathcal{N}_j \cap \mathcal{N}_i \ne \emptyset\}$, which is the collection of nodes located at two hops distance from node i plus neighbor nodes of i. We have the following result.

Proposition 3.1: Suppose there exist $\psi_i > 0$, $i = 1,\dots,N$, such that

$$\psi_i + \sqrt{\psi_i} \sum_{j\in\Theta_i} \sqrt{\alpha_{i,j}^{(i)}\alpha_{i,j}^{(j)}\psi_j} \le f^2(\Delta,\Upsilon)\,, \tag{III.6}$$

where $\alpha_{i,j}^{(i)}, \alpha_{i,j}^{(j)} \in (0,1)$ are such that

$$\sum_{c\in\mathcal{N}_j\cap\mathcal{N}_i} k_{ic}^2 \le \alpha_{i,j}^{(i)} \sum_{r=1}^{M_i} \kappa_{ii_r}^2 \qquad \sum_{c\in\mathcal{N}_j\cap\mathcal{N}_i} k_{jc}^2 \le \alpha_{i,j}^{(j)} \sum_{r=1}^{M_j} \kappa_{ji_r}^2.$$

If $\|\kappa_i(t)\|^2 \le \psi_i$, $i = 1,\dots,N$, then $\gamma_{\max}(K(t)) \le f(\Delta,\Upsilon)$.

Proposition 3.1 provides a simple local condition on the filter coefficients such that $\gamma_{\max}(K) \le f(\Delta,\Upsilon)$. We can expect that Proposition 3.1 is in general conservative, because no a-priori knowledge of the network topology is used, the proof relies on the Geršgorin theorem and the Cauchy-Schwartz inequality. There are many other ways to bound the eigenvalues of a matrix by its elements than the one used in the proof above, e.g., [28, pages 378–389]. However, we do not know of any other bounds requiring only local information,

useful for distributed implementation. Note also that Perron-Frobenius theory cannot be directly applied to bound the eigenvalues, because we make no assumption on the sign of the elements of $K(t)$.

The parameters $\alpha_{i,j}^{(i)}$ and $\alpha_{i,j}^{(j)}$ in Proposition 3.1 can all be set to one. It gives, however, conservative bounds on the maximum eigenvalue of KK^T. In Section IV, we will show how to chose these parameters to avoid too conservative bounds.

B. Optimal Weights for Variance Minimization

Using previous results, we can rewrite problem $\mathcal{P}_2$ as:

$$\mathcal{P}_3: \quad \min_{\kappa_i(t),\eta_i(t)} \quad \kappa_i(t)^T\Gamma_i(t-1)\kappa_i(t)+\sigma^2\eta_i(t)^T\eta_i(t) \tag{III.7}$$

$$\text{s.t.} \quad (\kappa_i(t)+\eta_i(t))^T\mathbb{1} = 1$$
$$\|\kappa_i\|^2 \le \psi_i\,, \tag{III.8}$$

The optimization problem is convex, because the cost function is convex ($\Gamma(t-1)$ is positive definite, since it represents the covariance matrix of Gaussian random variable) and the two constraints are convex. The problem admits a strict interior point solution, corresponding to $\kappa_i(t)=0$ and $\eta_i(t)\,\mathbb{1}=1$. Thus, Slater's condition is satisfied so strong duality holds [29, pag. 226]. The problem, however, does not have a closed form solution, so we need to rely on numerical algorithms to derive the optimal $\kappa_i(t)$ and $\eta_i(t)$. The following proposition provides a rather specific characterization of the solution.

Proposition 3.2: For a given positive definite matrix $\Gamma_i(t-1)$, the solution to problem $\mathcal{P}_2$ is given by

$$\kappa_i(t) = \frac{\sigma^2(\Gamma_i(t-1)+\xi_i I)^{-1}\mathbb{1}}{\sigma^2\,\mathbb{1}^T(\Gamma_i(t-1)+\xi_i I)^{-1}\mathbb{1}+M_i} \tag{III.9}$$

$$\eta_i(t) = \frac{\mathbb{1}}{\sigma^2\,\mathbb{1}^T(\Gamma_i(t-1)+\xi_i I)^{-1}\mathbb{1}+M_i}\,, \tag{III.10}$$

with $\xi_i \in \left[0, \max(0, \sigma^2/\sqrt{M_i\psi_i} - \lambda_{\min}(\Gamma_i(t-1)))\right]$.
Proposition 3.2 gives an interval within which the optimal ξ_i can be found. The first constraint in problem $\mathcal{P}_2$ resembles that of the water-filling problem for power allocation in wireless networks [29]. Analogously to that problem, simple search algorithms can be considered to numerically solve for ξ_i, for example, a bisection algorithm. Note that each node i needs to know the covariance matrix $\Gamma_i(t-1)$ to compute the weights.

C. Bounds on the Error Variance

The optimal weights from Proposition 3.2 gives the following estimation error variance.

Proposition 3.3: Let $\kappa_i(t)$ and $\eta_i(t)$ be an optimal solution given by (III.9) and (III.10). Then

$$\mathbb{E}\,(e_i(0) - \mathbb{E}\,e_i(0))^2 = \sigma^2$$
$$\mathbb{E}\,(e_i(t) - \mathbb{E}\,e_i(t))^2 \le \frac{\sigma^2}{M_i}\,, \quad t \in \mathbb{N}_0 \setminus \{0\}\,.$$

A consequence of Proposition 3.3 is that the estimation error in each node is always upper bounded by the variance of the estimator that computes the averages of the M_i measurements $u_i(t)$. The bound is obviously rather conservative, since we do not use any knowledge about the covariance matrix $\Gamma_i(t)$. Proposition 3.2 helps us to improve the bound in Proposition 3.3 as follows.

Corollary 3.4: The optimal value of $\kappa_i(t)$ and $\eta_i(t)$ are such that the error variance at node i satisfies

$$\mathbb{E}\,(e_i(t) - \mathbb{E}\,e_i(t))^2 \le \frac{\sigma^2}{M_i + \left(\sum_{j\in\mathcal{N}_i} M_j^{-1} + (M_i\psi_i)^{-1/2}\right)^{-1}}\,.$$

The choice of the constants ψ_i, $i=1,\dots,N$, in the local constraint of problem $\mathcal{P}_3$ is critical for the performance of the distributed estimator. A method for distributed computation of suitable values of ψ_i is given in [6].

D. Estimation of Error Covariance

Estimating the error covariance matrix is in general hard for the problem considered in this paper, because the estimator is a time-varying system and the stochastic process x, and thus e, is not stationary. However, if we consider the signals in the quasi-stationary sense, estimation based on samples guarantees to give good results. We have the following definition.

Definition 3.5 ([30, pag. 34]): A signal $s(t): \mathbb{R} \to \mathbb{R}$ is said to be quasi-stationary if there exists a positive constant C and a function $R_s: \mathbb{R} \to \mathbb{R}$, such that s fulfills the following conditions

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

It is easy to see that the time-varying linear system (II.2) is uniformly bounded-input bounded-output stable [31, pag. 509]. If a quasi-stationary signal is the input of such system, then its output is also quasi-stationary [32]. In our case, the measurement signal $u(t)$ is (component-wise) stationary and ergodic and thus also quasi-stationary. This implies that also $x(t)$ is quasi-stationary, since it is the output of a uniformly exponentially stable time-varying linear system. Thus, we estimate the error covariance using the sample covariance. Specifically, we have that the mean $\mathbb{E}\,\epsilon_i = m_{\epsilon_i}(t)$ and covariance $\Gamma_i(t)$ can be estimated from samples as

$$\hat{m}_{\epsilon_i}(t) = \frac{1}{t}\sum_{\tau=0}^{t}\hat{\epsilon}_i(\tau) \tag{III.11}$$

$$\hat{\Gamma}_i(\tau) = \frac{1}{\tau}\sum_{\tau=0}^{t}(\hat{\epsilon}_i(\tau)-\hat{m}_{\epsilon_i}(\tau))(\hat{\epsilon}_i(\tau)-\hat{m}_{\epsilon_i}(\tau))^T\,, \tag{III.12}$$

where $\hat{\epsilon}_i(t)$ is the an estimate of the error. Thus the problem reduces to design an estimator of $\epsilon_i(t)$. Node i has estimates $x_{i_j}(t)$ and measurements $u_{i_j}(t)$, $i_j \in \mathcal{N}_i$, available. Let $x^{(i)}(t)$ and $u^{(i)}(t)$ denote the collection of all these variables. We can model this data set as

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

where $\beta(t) \in \mathbb{R}^{M_i}$ models the bias of the estimates and $w(t)$ is zero-mean Gaussian noise modelling the variance of the estimator. Summarizing: node i has available $2M_i$ data values in which half of the data are corrupted by a small biased term $\beta(t)$ and a low variance noise $w(t)$ and the other half is corrupted by zero-mean Gaussian noise $v(t)$ with high variance. It is clear that using only $u^{(i)}(t)$ to generate an estimate $\hat{d}(t)$ of $d(t)$, which could then be used to estimate $\hat{\epsilon}_i(t) = x^{(i)}(t) - \hat{d}(t)\,\mathbb{1}$, would have the advantage of being unbiased. However, its covariance is rather large since M_i is typically small. Thus, using only measurements to estimate $d(t)$ yield to an over-estimate of the error, which results in poor performance. On the other hand, using only $x^{(i)}(t)$ we obtain an under-estimate of the error. This makes the weights $\eta_i(t)$ rapidly vanish and the signal measurements are discarded, thus tracking becomes impossible. From these arguments, in order to use both $x^i(t)$ and $u^i(t)$ we pose a linear least square problem as follows:

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

with $A \in \mathbb{R}^{2M_i \times M_i+1}$ and $B \in \mathbb{R}^{M_i \times M_i+1}$

$$A = \begin{pmatrix} \mathbb{1} & I \\ \mathbb{1} & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & I \end{pmatrix},$$

and ρ being the maxim value of the squared norm of the bias. However, the previous problem is difficult to solve in a closed form (it typically requires heavy numerical algorithms to find the solution, as SVD decomposition [33]). Notice also that, in general, the value of ρ is not known in advance, being it a maximum value of the cumulative bias. We thus consider the following regularized problem

$$\min_{\hat{d},\hat{\beta}} \quad \left\| \begin{pmatrix} x^i \\ u^i \end{pmatrix} - A \begin{pmatrix} \hat{d} \\ \hat{\beta} \end{pmatrix} \right\|^2 + \nu \left\| B \begin{pmatrix} \hat{d} \\ \hat{\beta} \end{pmatrix} \right\|^2 \qquad \text{(III.13)}$$

where $\nu > 0$ is a parameter whose choice is typically rather difficult. Notice that a stochastic least square problem cannot be used since the cross covariance between the data $x^{(i)}(t)$ and $u^{(i)}(t)$ is not known and it seems difficult to estimate.

The solution of (III.13) is

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

The inverse of the matrix in the previous equation can be computed in closed form [6].

Since we are interested in estimating $\epsilon_i(t) = x(t) - d(t)\,\mathbb{1}$ we observe that such an estimate is given by $\hat{\beta}$. From the solution of (III.13), we have

$$\hat{\beta} = \frac{x^i}{1+\nu} - \frac{\nu\,\mathbb{1}^T x^i + (1+\nu)\,\mathbb{1}^T u^i}{M_i(1+2\nu)(1+\nu)}\,\mathbb{1} \qquad \text{(III.14)}$$

For the choice of the parameter ν we propose to use the Generalized Cross-Validation (GCV) method [34]. This consists in choosing ν as

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

Typically the GCV methods is computationally expensive since the trace of the matrix $(A^T A + \nu B^T B)^{-1}$ is difficult to compute, but in our case we have a closed form representation of the matrix, and thus the computation is not difficult. However, it might be computationally difficult to carry out the minimization. Observing that

$$\begin{aligned} \nu &= \arg\min \frac{\|(A^T A + \nu B^T B)^{-1} A^T (x^i, u^i)^T\|}{\operatorname{tr}(A^T A + \nu B^T B)^{-1}} \\ &\leq \arg\min \frac{\|(A^T A + \nu B^T B)^{-1} A^T\|}{\operatorname{tr}(A^T A + \nu B^T B)^{-1}} \|(x^i, u^i)^T\| , \end{aligned} \qquad \text{(III.15)}$$

a sub-optimal value of ν can be computed solving the right hand side of (III.15). Note that the first term in the right hand side of (III.15) is a function of ν that can be computed off-line and stored in a look-up table at the node. Then, for different data, the problem becomes that of searching in the table.

Using (III.14) with the parameter ν computed from (III.15) we can then estimate the error mean and covariance matrix applying (III.11) and (III.12), respectively.

IV. Implementation and Numerical Results

This section presents the estimator structure and the algorithmic implementation followed by some numerical results.

A. Estimator Structure and Implementation

Figure 1 summarizes the structure of the estimator implemented in each node. The estimator has a cascade structure with two sub-systems: the one to the left is an adaptive filter that produces the estimate of d; the one to the right computes an estimate of the error covariance matrix Γ_i. In the following, we discuss in some detail a pseudo-code implementation of the blocks in the figure.

The estimator is presented as Algorithm 1. Initially, the distributed computation of the threshold is performed (lines 1–8): node i updates its threshold ψ_i until a given precision ϖ is reached. In the computations of ψ_i, we chose $\alpha_{i,j}^{(i)} = |\mathcal{N}_j \cap \mathcal{N}_i|/(M_i - 1)$ and $\alpha_{i,j}^{(j)} = |\mathcal{N}_j \cap \mathcal{N}_i|/(M_j - 1)$. This works well in practice because k_{ii_r}, $i_r = 1, \ldots, M_i$, are of similar magnitude. Indeed, the stability of the average of the estimation error established in Section II-B, and the bounds on the error variance in Section III-C, ensure that estimates among nodes have similar performance. Numerical results show that that the while-loop (lines 4–8) converges after about 10–20 iterations.

The estimators for the local mean estimation error and the local covariance matrix are then initialized (lines 9–10). The main loop of the estimator is lines 13–24. Lines 14–19 are related to the left subsystem of Figure 1. The optimal weights are computed using Equations (III.9) and (III.10) (lines 17–18). Notice that the optimal Lagrangian multiplier ξ_i is computed using the function `bisection` which takes as argument the interval $[0, \max(0, \sigma^2/\sqrt{M_i \psi_i} - \lambda_{\min}(\Gamma_i(t-1)))]$ where the optimal value lays. Notice that, if the nodes have limited computational power, so that the minimum eigenvalue of the matrix $\Gamma_i(t-1)$ cannot be exactly computed, an upper-bound based on Geršgorin can be used instead. The estimate of

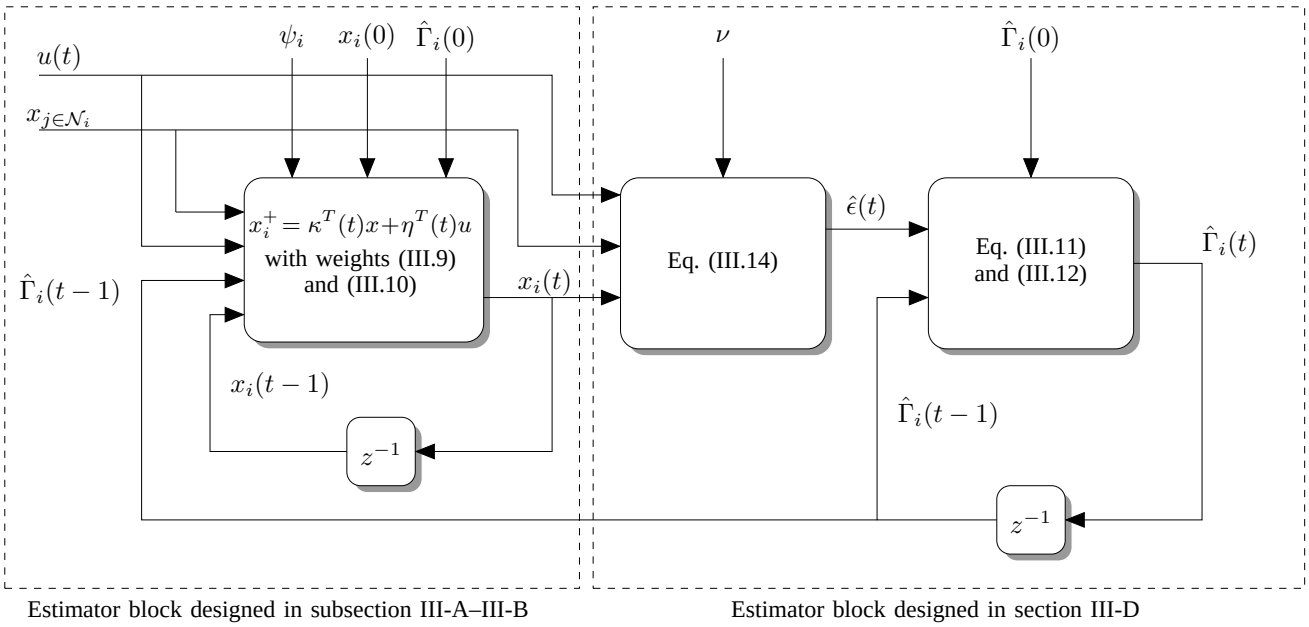

Fig. 1. Block diagram of the proposed estimator. It consists of two subsystems in a cascade coupling. The subsystem to the left is an adaptive filter that produces the estimate of $d(t)$ with small variance and bias. The subsystem to the right estimates the error covariance matrix.

Algorithm 1 Estimation algorithm for node i

1. $t := 0$
2. $\psi_i(t-1) = 0$
3. $\psi_i(t) = 1/M_i$
4. **while** $|\psi_i(t) - \psi_i(t-1)| \geq \varpi = 10^{-10}$ **do**
5. $\quad \psi_i(t+1) = T_i(\psi(t))$
6. $\quad$ collect thresholds from nodes in Θ_i
7. $\quad t := t + 1$
8. **end while**
9. $t := 0$
10. $\hat{m}_{\epsilon_i}(t) := 0$
11. $\hat{\Gamma}_i(t) := \sigma^2 I$
12. $x_i(t) := u_i(t)$
13. **while** forever **do**
14. $\quad M_i := |\mathcal{N}_i|$
15. $\quad t := t + 1$
16. $\quad \xi_i$ = bisection $\left([0, \max(0, \sigma^2/\sqrt{M_i\psi_i} - \lambda_{\min}(\Gamma_i(t-1)))]\right)$
17. $\quad \kappa_i(t) := \dfrac{\sigma^2(\hat{\Gamma}_i(t-1) + \xi_i I)^{-1}\,\mathbb{1}}{M_i + \sigma^2\,\mathbb{1}^T(\hat{\Gamma}_i(t-1) + \xi_i I)^{-1}\,\mathbb{1}}$
18. $\quad \eta_i(t) := \dfrac{\mathbb{1}}{M_i + \sigma^2\,\mathbb{1}^T(\hat{\Gamma}_i(t-1) + \xi_i I)^{-1}\,\mathbb{1}}$
19. $\quad x_i(t) := \sum_{j\in\mathcal{N}_i} \kappa_{i_j}(t)x_j(t-1) + \sum_{j\in\mathcal{N}_i} \eta_{i_j}(t)u_j(t)$
20. $\quad \hat{\beta} := \dfrac{x^i}{1+\nu} - \dfrac{\nu\,\mathbb{1}^T x^i + (1+\nu)\,\mathbb{1}^T u^i}{M_i(1+2\nu)(1+\nu)}\mathbb{1}$
21. $\quad \hat{\epsilon}_i := \hat{\beta}$
22. $\quad \hat{m}_{\epsilon_i}(t) := \dfrac{t-1}{t}\,\hat{m}_{\epsilon_i}(t-1) + \dfrac{1}{t}\,\hat{\epsilon}_i(t)$
23. $\quad \hat{\Gamma}_i(t) := \dfrac{t-1}{t}\,\hat{\Gamma}_i(t-1) + \dfrac{1}{t}\,(\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t))(\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t))^T$
24. **end while**

$d(t)$ is computed in line 19. Lines 20–23 are related to the right subsystem of Figure 1. These lines implement the error covariance estimation by solving the constrained least-squares minimization problem described in subsection III-D. Sample mean and covariance of the estimation error are updated in lines 22–23. These formulas correspond to recursive implementation of (III.11) and (III.12).

Let us comment on the inversions of the estimated error covariance matrix $\hat{\Gamma}_i$ in lines 17–18. In general, the dimension of $\hat{\Gamma}_i$ is not a problem because we consider cases when the number of neighbors is small. Precautions have still to be taken, because even though the error covariance matrix Γ_i is always positive definite, its estimate $\hat{\Gamma}_i$ may not be positive definite before sufficient statistics are collected. In our implementation, we use heuristics to ensure that $\hat{\Gamma}_i$ is positive definite.

B. Numerical Results

Numerical simulations have been carried out in order to validate performance of the proposed distributed estimator. We compare the our estimator with some similar estimators related to the literature. We consider the following five estimators:

- E_1: $K = H = (I - L)/2$ where L is the Laplacian matrix associated to the graph $\mathcal{G}$.
- E_2: $K = 0$ and $H = [h_{ij}]$ with $h_{ij} = 1/M_i$ if node i and j are connected, and $h_{ij} = 0$ otherwise. Thus, the updated estimate is the average of the measurements.
- E_3: $K = [k_{ij}]$, where $k_{ii} = 1/2M_i$, $k_{ij} = 1/M_i$ if node i and j are connected, $k_{ij} = 0$ otherwise, whereas $H = [h_{ij}]$ with $H_{ii} = 1/2M_i$, and $h_{ij} = 0$ elsewhere. This is the average of the old estimates and node's single measurement.

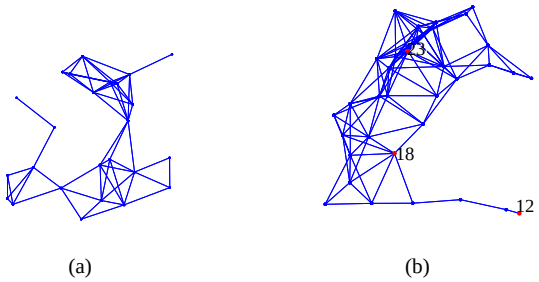

Fig. 2. Topology of the networks with $N = 25$ nodes (on the left) and $N = 35$ (on the right) used in the simulations. For the network with $N = 35$, three nodes are highlighted, corresponding to the identifier 12, 18, and 23. They have the following number of neighbors: $|\mathcal{N}_{12}| = 2$, $|\mathcal{N}_{18}| = 8$, and $|\mathcal{N}_{23}| = 15$. The node with maximum degree in all the network is node 23.

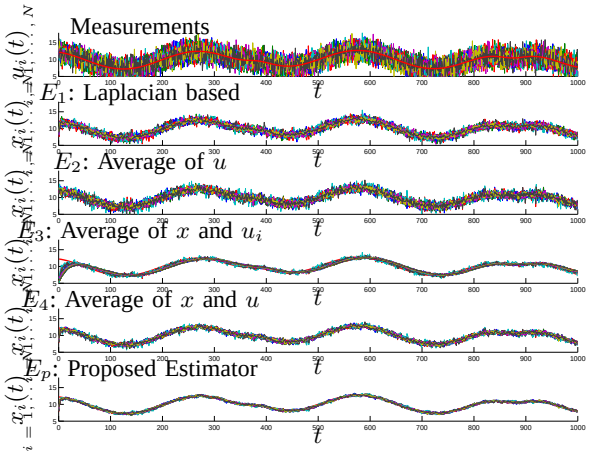

Fig. 3. Plots showing $N = 35$ realizations of the measurements and estimates at each node for each estimator.

E_4: $K = H$ with $k_{ij} = 1/2M_i$ if node i and j are connected, and $i = j$. The updated estimate is the average of the old estimates and all local measurements.

E_p: The estimator proposed in this paper.

The estimators $E_1, \ldots, E_4$ are based on various heuristics. They are related to proposals in the literature, e.g., E_1 uses filter coefficients given by the Laplacian matrix, cf., [12]–[14]. It is important to note, however, that in general the weights based on Laplacian do not ensure the minimization of the variance of the estimation error. Notice that we did not consider the centralized solution. Although this would be interesting, it is computationally difficult to solve problem $\mathcal{P}_1$ at each time step for each node, even for small networks.

We have benchmarked the estimators with various test signals d. Here we limit the discussion to a specific case. We suppose that we know a bound Δ on the variation of d. We set Δ to be 10% larger than its actual value. We have chosen the desired average SNR to $\Upsilon = 10$, see Section II. We consider the two networks $\mathcal{G}_{25}$ and $\mathcal{G}_{35}$ with $N = 25$ and $N = 35$ nodes, respectively, shown in Figure 2. These networks are obtained by distributing the nodes randomly over a squared

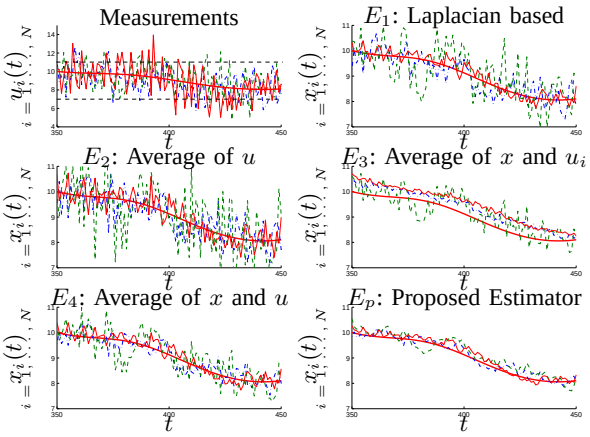

Fig. 4. Zoom of some of the curves in Figure **??**. In particular, we plot the measurements and estimates of the nodes 12, 18 and 23 having the minimum degree, degree equal to the average degree of the network, and maximum, respectively (see Figure 2). In thick solid curve is shown the signal $d(t)$. The dashed curves show the measurement and estimate at node 12, in dash-dotted those at node 18 and the solid curves show those at node 23. The horizontal lines in the the top-left figure are the interval within which the estimates variate. We chose to have different scales to make more clear the estimation process.

area of size $N/3$. The graph is then obtained by letting two nodes communicate if their relative distance is less than $\sqrt{N}$.

We discuss in detail the distributed estimator over the network network $\mathcal{G}_{35}$. Measurements and estimates for all nodes are shown in Figure 3. Clearly, the measurements are quite noisy, and in particular $\sigma^2 = 1.5$. All estimators, $E_1, \ldots, E_4$ and E_p, are able to track the signal, but the quality of the estimates are varying quite a bit. It is evident that E_1 and E_2 give the worst estimates, while E_p performs best. The relative performance between $E_1, \ldots, E_4$ is rather obvious given how their estimate is constructed, e.g., E_2 simply take the average of the measurements while E_4 averages over both measurements and estimates. By choosing the weights appropriately, we see that the proposed estimator E_p gives substantially lower estimation variance. Figure 4 shows a zoom of Figure 3 for the time interval $[350, 450]$. The figure compares the measurements and estimates of the three nodes highlighted in Figure 2. These nodes represent the node with minimum connectivity (dashed curve), average connectivity (dash-dotted curve) and maximum connectivity (solid curve). The thick line correspond to d. Note that the node with low connectivity is not following d very well. We also see that the estimate produced by E_3 has a quite substantial bias. In general, we have observed through extensive simulations that E_3 work well for low-frequency signals to track, whereas E_4 works better for signal with higher frequency. Numerical studies of various networks confirm the type of behaviors we see in Figures 3 and 4.

V. Conclusions

In this paper, we have presented a fully distributed minimum variance estimator for wireless sensor networks. The purpose

of such estimator is accurate tracking of a time varying signal using noisy measurements. A mathematical framework is proposed to design a filter, which runs locally in each node. It only requires a cooperation among neighboring nodes. In order to obtain a minimum variance estimator, we started from a centralized optimization problem, and then we converted it into a decentralized problem transforming global constraints into distributed ones. The filter structure is composed by a cascade of two blocks: the first block computes the estimator coefficients at each time instance, and the second block estimates the error covariance matrix needed, by the first block, at next step. The estimator coefficients are designed such that the local behavior of a node ensures the overall estimation process to be stable. We showed that the distributed estimator is stable, with mean and variance of the estimation error bounded. Numerical results proved that our filter outperforms existing solutions proposed in literature, as well as other heuristic solutions. Future work includes stability analysis of the filter with respect to packet losses, and experimental validation in our laboratory setting.

ACKNOWLEDGMENT

The authors would like to thank Håkan Hjalmarsson, Björn Johansson, Mikael Johansson and Bo Wahlberg for fruitful discussions.

REFERENCES

[1] D. Estrin, R. Govindan, J. Eidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, 1999.

[2] H. Gharavi and P. R. Kumar, Eds., *Proceedings of IEEE: Special Issue on Sensor Networks and Applications*, vol. 91, 2003.

[3] Z.-Q. Luo, M. Gatspar, J. Liu, and A. Swami, Eds., *IEEE Signal Processing Magazine:Special Issue on Distributed Signal Processing in Sensor Networks*. IEEE, 2006.

[4] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. Sastry, "Distributed control applications within sensor networks," *IEEE Proceedings, Special Issue on Distributed Sensor Networks*, 2004.

[5] K.-E. Årzén, A. Bicchi, G. Dini, S. Hailes, K. H. Johansson, J. Lygeros, and A. Tzes, "A component-based approach to the design of networked control systems," *European Journal of Control*, vol. 13, no. 2–3, 2007.

[6] A. Speranzon, C. Fischione, and K. H. Johansson, "A distributed minimum variance estimator for wireless sensor networks," 2007, submitted for journal publication.

[7] ——, "Distributed and collaborative estimation over wireless sensor networks," in *IEEE Conference on Decision and Control*, 2006.

[8] ——, "A distributed estimation algorithm for tracking over wireless sensor networks," in *IEEE International Conference on Communication*, 2007, to appear.

[9] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[10] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, 2004.

[11] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the state agreement problem," *Automatica*, 2007, to appear.

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

[13] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Approximate distributed Kalman filtering in sensor networks with quantifiable performance," in *IEEE Conference on Decision and Control*, 2005.

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

[15] 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.

[16] S. B. L. Xiao, "Fast linear iterations for distributed averaging," *System Control Letter*, 2004.

[17] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization : A Universal Concept in Nonlinear Sciences*. Cambridge University Press, 2003.

[18] Y.-W. Hong, L. Cheow, and A. Scaglione, "A simple method to reach detection consensus in massively distributed sensor networks," in *IEEE International Symposium on Information Theory*, 2004.

[19] S. Barbarossa and F. Celano, "Self-organizing sensor networks designed as a population of mutually coupled oscillators," in *IEEE Signal Processing Advances in Wireless Communications*, 2005.

[20] S. Barbarossa and G. Scutari, "Decentralized maximum likelihood estimation for sensor networks composed of nonlinearly coupled dynamical system," *IEEE Transactions on Signal Processing*, 2007, to appear.

[21] F. Zhao, J. Liu, L. Guibas, and J. Reich, "Collaborative signal and information processing: an information-directed approach," *Proceedings of the IEEE*, vol. 91, no. 8, 2003.

[22] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *Eurasip Journal on Applied Signal Processin*, no. 4, pp. 378–391, 2003.

[23] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Transaction on Parallel and Distributed System, Vol. 15, No. 11*, 2004.

[24] G. Di Stefano, F. Graziosi, and F. Santucci, "Distributed positioning algorithm for ad-hoc networks," *International Workshop on Ultra Wideband Systems (IWUWBS)*, 2003.

[25] A. Giridhar and P. Kumar, "Distributed clock synchronization over wireless networks: Algorithm and analysis," in *45th IEEE Conference on Decision and Control*, 2006.

[26] P. Alriksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging," in *In Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, 2006.

[27] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, ser. Studies in Applied Mathematics. Philadelphia: SIAM, 1994.

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

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

[30] L. Ljung, *System Identification: Theory for the User (2nd Edition)*. Prentice Hall PTR, 1998.

[31] W. J. Rugh, *Linear Sysmtem Theory*, 2nd ed. Prentice Hall, 1996.

[32] 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.

[33] G. H. Golub and U. Von Matt, "Quadratically constrained least squares and quadratic problems," *Numerische Mathematik*, vol. 59, no. 6, 1991.

[34] ——, "Tikhonov regularization for large scale problems," Stanford SCCM, Stanford, Tech. Rep. 97-03, 1997.