In-Network-Processing: Distributed Consensus-Based Linear Estimation

Henning Paul, *Member, IEEE,* Jörg Fliege, and Armin Dekorsy, *Member, IEEE*

***Abstract*—In a cooperative broadcast scenario, a group of nodes in a network aims to reconstruct a common message. In this paper, we present a new algorithm for distributed consensus-based estimation in such scenarios. Possible applications comprise mobile communication systems and sensor networks. Starting with a least squares estimation problem, the algorithm is developed using techniques from optimization theory. The required communication effort for parallel implementation in a resource-constrained network is estimated and compared to existing approaches. We show that the proposed algorithm requires fewer iterations and a reduced communication overhead per iteration while keeping the estimation accuracy. A modification of the algorithm based on an approximation is presented, which reduces the communication effort even further. All results are corroborated by computer simulations considering different system parameters.**

***Index Terms*—Wireless sensor networks, distributed estimation, distributed consensus, in-network processing.**

I. Introduction

COOPERATIVE communication in wireless networks has gathered major interest in the research community during the past years. A common scenario is the cooperative broadcast case, in which a group of nodes, connected via inter-node links, aims to recover a common message broadcast by a detached station. Possible applications are, e.g., CoMP (Cooperative MultiPoint) or data aquisition in Wireless Sensor Networks (WSNs) for use, e.g., in logistics, medicine or environmental sciences. The amount of information about the message that is locally available at the sensing nodes is in many cases not sufficient, requiring cooperation among the nodes for successful reconstruction. While it is possible to perform centralized reconstruction in a dedicated Data Fusion Center [1], this would be a single point of failure, and a distributed algorithm promises higher robustness against node and link failures. By performing some amount of processing at nodes and employing inter-node transmission, we aim for obtaining the centralized solution through in-network processing. Using consensus-based reconstruction within the network, we ensure that eventually, the centralized solution is available at all nodes.

In this paper, we will present a new algorithm for consensus-based distributed linear estimation. We introduce the algorithm for Least Squares (LS) optimization (e.g. [2]), but the application to other criteria (e.g., [3], [4]) is possible. To solve the LS problem within the network, we follow a mathematical framework using the Augmented Lagrangian approach. But compared to previous work such as [5], [6], we implement the consensus constraint within the network in a novel fashion, resulting in a more general and flexible scheme. We will show that this approach reduces the communication overhead and required number of operations compared to existing algorithms whilst keeping the estimation quality.

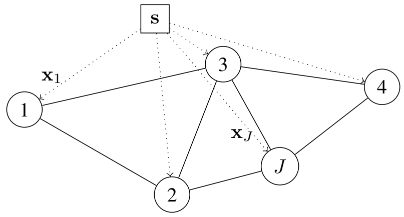

Fig. 1. A network of J nodes receiving different observations $\mathbf{x}_j$ of the same quantity $\mathbf{s}$.

II. Problem Formulation

Fig. 1 illustrates the cooperative broadcast scenario. J nodes are connected with inter-node links, forming a sensor network. In the following, we require the graph describing the network to be connected, i.e., every node is able to reach every other node, albeit using several hops. For the derivation of the algorithm, we consider the inter-node links to be ideal, i.e. information exchange on these links is error-free. At every node j in the set $\mathcal{J}$ of all nodes, knowledge of the message $\mathbf{s}$ is desired, while only a disturbed observation $\mathbf{x}_j$ of it is available. We can model the relationship between these variables as linear, using a disturbance matrix $\mathbf{H}_j$ and an additional noise term $\mathbf{n}_j$, resulting in the linear equation

$$\mathbf{x}_j = \mathbf{H}_j\mathbf{s} + \mathbf{n}_j \tag{1}$$

with $\mathbf{s} \in \mathbb{R}^{N\times 1}$, $\mathbf{x}_j \in \mathbb{R}^{M\times 1}$, $\mathbf{H}_j \in \mathbb{R}^{M\times N}$ and $\mathbf{n}_j \in \mathbb{R}^{M\times 1}$. M and N therefore denote the dimensions of the observation and quantity vectors, respectively. For the sake of simplicity, only real valued systems are considered, but extension to complex valued variables is straightforward.

To reconstruct $\mathbf{s}$ in (1), one approach is to perform individual Least Squares (LS) estimation per node, resulting in local estimates $\hat{\mathbf{s}}_j$. In general, these estimates will differ over the nodes j and therefore are not a consensus solution: The presence of noise $\mathbf{n}_j$ or a rank deficiency of $\mathbf{H}_j$ leads to a poor estimate. Furthermore, if $M < N$, (1) is underdetermined and multiple solutions exist. If we want to exploit the entire information on $\mathbf{s}$ available in the network, finding a unique solution for all nodes, we have to solve the centralized problem

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}'} ||\mathbf{x}^{\mathrm{s}} - \mathbf{H}^{\mathrm{s}}\mathbf{s}'||^2 \tag{2}$$

using the stacked observation vector $\mathbf{x}^{\mathrm{s}} = \left[\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_J^{\mathrm{T}}\right]^{\mathrm{T}}$ and

Manuscript received August 9, 2012. The associate editor coordinating the review of this letter and approving it for publication was G. Lazarou.

H. Paul and A. Dekorsy are with the Department of Communication Engineering, University of Bremen, Bremen, Germany (e-mail: {paul, dekorsy}@ant.uni-bremen.de).

J. Fliege is with the Centre of Operational Research, Management Sciences and Information Systems, University of Southampton, Southampton, United Kingdom (e-mail: J.Fliege@soton.ac.uk).

Digital Object Identifier 10.1109/LCOMM.2012.121788

disturbance matrix $\mathbf{H}^{\mathrm{s}} = \left[\mathbf{H}_1^{\mathrm{T}}, \ldots, \mathbf{H}_J^{\mathrm{T}}\right]^{\mathrm{T}}$. One approach to solve (2) is centralized processing in a Data Fusion Center. Obviously, the sensors' observations and disturbance matrices need to be forwarded through the network to a central processing node, which requires routing protocols and communication effort. Furthermore, an outage of the fusion center will cause the whole network to fail, making it a single point of failure.

Therefore, our goal is to solve the centralized problem (2) in a distributed fashion within the network. In literature, there exists a number of algorithms for consensus-based distributed estimation (e.g., [7], [8], [9]). Our proposed algorithm exhibits the advantage that every communication of nodes with its neighbors only takes place in a broadcast fashion, leading to a simple, yet performant algorithm. Its derivation is detailed in the following.

III. Approach

Due to its structure, the centralized LS criterion as introduced in (2) can be decomposed in parallel local estimation problems coupled by a consensus constraint:

$$\begin{aligned} \{\hat{\mathbf{s}}_j | j \in \mathcal{J}\} &= \arg \min_{\{\mathbf{s}_j | j \in \mathcal{J}\}} \sum_{j=1}^{J} ||\mathbf{x}_j - \mathbf{H}_j \mathbf{s}_j||^2 \\ \text{s.t.} \quad \mathbf{s}_j &= \mathbf{s}_i \quad \forall \quad j \in \mathcal{J}, \quad i \in \mathcal{N}_j. \end{aligned}$$

The consensus constraint $\mathbf{s}_j = \mathbf{s}_i \quad \forall \quad j \in \mathcal{J}, \quad i \in \mathcal{N}_j$ (as introduced, e.g., in [5]) enforces agreement on the estimate $\mathbf{s}_j$ at any node j with all nodes i in its (graph theoretic) neighborhood $\mathcal{N}_j$. In combination with above requirement of a connected graph, this leads to overall consensus in the whole network.

Obviously, this problem can not be solved in a distributed fashion, since all local estimates $\mathbf{s}_j$ and $\mathbf{s}_i$ are directly coupled. To facilitate parallel processing, we introduce auxiliary variables. In [5], which uses a similar approach and therefore will be used as a reference for the performance evaluation later, two variables per edge of the network are employed, triplicating the consensus contraint into $\mathbf{s}_j = \mathbf{z}_{ji}$, $\mathbf{s}_i = -\mathbf{z}'_{ji}$ and $\mathbf{z}_{ji} = -\mathbf{z}'_{ji}$. In contrast to this, we propose the use of only one auxiliary variable $\mathbf{z}_j$ per node. This allows for a more elegant derivation and, as will be shown below, results in a lower required number of iterations and reduced communication overhead. The constraint equations resulting from this approach read

$$\mathbf{s}_j = \mathbf{z}_i, \quad \mathbf{s}_j = \mathbf{z}_j \quad \forall \quad j \in \mathcal{J}, \quad i \in \mathcal{N}_j. \tag{3}$$

In the Augmented Lagrangian method [10], [11] employed in the following for the solution of this optimization problem, every constraint will be regarded explicitely and equally, with dedicated Lagrange multipliers $\boldsymbol{\lambda}_{j,i}$ and $\mathbf{v}_j$ and penalty parameters $\mu_{j,i}$, α_j respectively, resulting in the Lagrange cost function

$$\begin{aligned} \mathcal{L}(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{\lambda}) &= \frac{1}{2} \sum_{j=1}^{J} ||\mathbf{x}_j - \mathbf{H}_j \mathbf{s}_j||^2 \qquad (4) \\ &- \sum_{j=1}^{J} \left[\mathbf{v}_j^{\mathrm{T}} (\mathbf{s}_j - \mathbf{z}_j) + \sum_{i \in \mathcal{N}_j} \boldsymbol{\lambda}_{j,i}^{\mathrm{T}} (\mathbf{s}_j - \mathbf{z}_i) \right] \\ &+ \sum_{j=1}^{J} \left[\frac{1}{2\alpha_j} ||\mathbf{s}_j - \mathbf{z}_j||^2 + \sum_{i \in \mathcal{N}_j} \frac{1}{2\mu_{j,i}} ||\mathbf{s}_j - \mathbf{z}_i||^2 \right]. \end{aligned}$$

The expressions inside the square brackets can be combined if the definitions

$$\boldsymbol{\lambda}_{j,j} := \mathbf{v}_j, \quad \mu_{j,j} := \alpha_j \tag{5}$$

are introduced and the summation over i is extended to include j. Applying these simplifications, the cost function can be decomposed into

$$\mathcal{L}(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{\lambda}) = \sum_{j=1}^{J} F_j(\mathbf{s}_j, \boldsymbol{z}, \boldsymbol{\lambda}) \tag{6}$$

with

$$\begin{aligned} F_j(\mathbf{s}_j, \boldsymbol{z}, \boldsymbol{\lambda}) &= \frac{1}{2} ||\mathbf{x}_j - \mathbf{H}_j \mathbf{s}_j||^2 - \sum_{i \in \mathcal{N}_j \cup j} \boldsymbol{\lambda}_{j,i}^{\mathrm{T}} (\mathbf{s}_j - \mathbf{z}_i) \\ &+ \sum_{i \in \mathcal{N}_j \cup j} \frac{1}{2\mu_{j,i}} ||\mathbf{s}_j - \mathbf{z}_i||^2, \end{aligned} \tag{7}$$

which can be minimized w.r.t. $\mathbf{s}_j$ at every node individually. Calculating its gradient with respect to $\mathbf{s}_j$ and setting it to zero, the condition

$$\begin{aligned} \mathbf{s}_j &= \left(\mathbf{H}_j^{\mathrm{T}} \mathbf{H}_j + \sum_{i \in \mathcal{N}_j \cup j} \frac{1}{\mu_{j,i}} \mathbf{I} \right)^{-1} \qquad (8) \\ &\cdot \left[\mathbf{H}_j^{\mathrm{T}} \mathbf{x}_j + \sum_{i \in \mathcal{N}_j \cup j} \left(\boldsymbol{\lambda}_{j,i} + \frac{1}{\mu_{j,i}} \mathbf{z}_i \right) \right] \end{aligned}$$

on the variables $\mathbf{s}_j$ is obtained. The right hand side of above equation depends, apart from $\mathbf{H}_j$ and $\mathbf{x}_j$ which are constant to the optimization, on $\mathbf{z}_i$ and the Lagrange multipliers $\boldsymbol{\lambda}_{j,i}$, thus also for these variables corresponding expressions have to be found.

Regarding $\mathbf{z}_j$, following considerations can be made: Noting that each edge in the network is covered exactly twice by the double summations in (4), we can rearrange the indices j and i and obtain the representation

$$\mathcal{L}(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{\lambda}) = \sum_{j=1}^{J} F'_j(\boldsymbol{s}, \mathbf{z}_j, \boldsymbol{\lambda}), \tag{9}$$

where each of the functions

$$\begin{aligned} F'_j(\boldsymbol{s}, \mathbf{z}_j, \boldsymbol{\lambda}) &= \frac{1}{2} ||\mathbf{x}_j - \mathbf{H}_j \mathbf{s}_j||^2 - \sum_{i \in \mathcal{N}_j \cup j} \boldsymbol{\lambda}_{i,j}^{\mathrm{T}} (\mathbf{s}_i - \mathbf{z}_j) \\ &+ \sum_{i \in \mathcal{N}_j \cup j} \frac{1}{2\mu_{i,j}} ||\mathbf{s}_i - \mathbf{z}_j||^2. \end{aligned} \tag{10}$$

depends on $\mathbf{z}_j$ only. Setting its derivative w.r.t. $\mathbf{z}_j$ to zero, the condition

$$\mathbf{z}_j = \frac{1}{\sum_{i \in \mathcal{N}_j \cup j} \frac{1}{\mu_{i,j}}} \sum_{i \in \mathcal{N}_j \cup j} \left[-\boldsymbol{\lambda}_{i,j} + \frac{1}{\mu_{i,j}} \mathbf{s}_i \right] \tag{11}$$

is obtained. Eqs. (8) and (11) suggest an iterative update of the variables $\mathbf{s}_j$ and $\mathbf{z}_j$. In that case also the variables $\boldsymbol{\lambda}_{i,j}$ need to be updated iteratively. Their update equation for the Augmented Lagrangian method is given, e.g., by [10] and reads

$$\boldsymbol{\lambda}_{i,j}^{(k+1)} = \boldsymbol{\lambda}_{i,j}^{(k)} - \frac{1}{\mu_{i,j}}(\mathbf{s}_i^{(k+1)} - \mathbf{z}_j^{(k+1)}), \tag{12}$$

with the superscript (k) denoting the value of the corresponding variable in the kth iteration.

The complete update equations of the distributed iterative algorithm therefore are

$$\mathbf{s}_j^{(k+1)} = \left(\mathbf{H}_j^{\mathrm{T}}\mathbf{H}_j + \sum_{i\in\mathcal{N}_j\cup j}\frac{1}{\mu_{j,i}}\mathbf{I}\right)^{-1} \cdot \left[\mathbf{H}_j^{\mathrm{T}}\mathbf{x}_j + \sum_{i\in\mathcal{N}_j\cup j}\left(\boldsymbol{\lambda}_{j,i}^{(k)} + \frac{1}{\mu_{j,i}}\mathbf{z}_i^{(k)}\right)\right] \tag{13}$$

$$\mathbf{z}_j^{(k+1)} = \frac{1}{\sum\limits_{i\in\mathcal{N}_j\cup j}\frac{1}{\mu_{i,j}}}\sum_{i\in\mathcal{N}_j\cup j}\left[-\boldsymbol{\lambda}_{i,j}^{(k)} + \frac{1}{\mu_{i,j}}\mathbf{s}_i^{(k+1)}\right], \tag{14}$$

$$\boldsymbol{\lambda}_{i,j}^{(k+1)} = \boldsymbol{\lambda}_{i,j}^{(k)} - \frac{1}{\mu_{i,j}}(\mathbf{s}_i^{(k+1)} - \mathbf{z}_j^{(k+1)}). \tag{15}$$

For $\mathbf{s}_j^{(0)} = \mathbf{z}_j^{(0)} = \boldsymbol{\lambda}_{i,j}^{(0)} = \mathbf{0}$, we can show similar to [5], that $\mathbf{s}_j^{(k)}$ only depends on the observations $\mathbf{x}_\ell$ in a linear fashion. This simplifies bit error analysis, which is not performed here due to space limitations.

A. Communication overhead

The communication effort between nodes is often a bottleneck of the overall system. Thus, the communication overhead of the proposed algorithm has to be determined and compared to existing algorithms. The required communication effort between the nodes can be determined by inspection of the update equations (13)–(15). Channel knowledge $\mathbf{H}_j$ and observation $\mathbf{x}_j$ do not change over the iterations k, so for calculation of $\mathbf{s}_j^{(k+1)}$, all current auxiliary variables $\mathbf{z}_i^{(k)}$ of the nodes' neighbors (and the node itself) and the Lagrange variables $\boldsymbol{\lambda}_{j,i}^{(k)}$ are required. The update of $\mathbf{z}_j^{(k+1)}$ similarly requires knowledge on $\boldsymbol{\lambda}_{i,j}^{(k)}$ and the neighbors' (and its own) previously calculated estimates $\mathbf{s}_i^{(k+1)}$. Obviously, at least $\mathbf{s}_j^{(k+1)}$ and $\mathbf{z}_j^{(k+1)}$ need to be broadcast to all the neighboring nodes. Inspecting (12), we find that all Lagrange multipliers can be updated locally at the nodes where their knowledge is required, since after $\mathbf{s}_i^{(k+1)}$ and $\mathbf{z}_j^{(k+1)}$ have been broadcast, everything required for the update is available at the nodes. Regarding the penalty parameter $\mu_{i,j}$, we are aware that a proper update rule for it can improve the estimation accuracy [10], [11], but for the sake of simplicity, in the following all $\mu_{i,j}$ are chosen to be identical and constant.

The algorithm presented in [5] requires the distribution of the estimates $\mathbf{s}_j$ to all its neighbors, e.g., as a broadcast, and more costly, the transmission of the individual Lagrange parameters to each of its neighbors as unicast transmissions. The algorithm proposed here only requires two broadcasts of $\mathbf{s}_j$ and $\mathbf{z}_j$ and thus reduces the communication overhead.

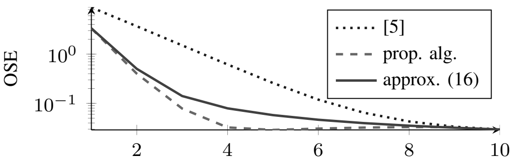

Fig. 2. Overall squared error over iterations k for $J = 6$ nodes and one random signal, channel and noise realization.

To further reduce the communication effort, we propose the following approximation: For an update of $\mathbf{s}_j$, the node j requires knowledge on $\mathbf{z}_j$ and all $\mathbf{z}_i \,\forall\, i \in \mathcal{N}_j$, which have to be broadcast from the corresponding nodes i. An approximation of $\mathbf{z}_i$ at node j can be obtained if most terms of the summation (14) are neglected and only variables $\mathbf{s}_i$ available from the previous broadcast are employed:

$$\tilde{\mathbf{z}}_{i,j}^{(k+1)} = \frac{1}{\frac{1}{\mu_{i,i}} + \frac{1}{\mu_{j,i}}}\left[-\boldsymbol{\lambda}_{i,i}^{(k)} - \boldsymbol{\lambda}_{j,i}^{(k)} + \frac{1}{\mu_{i,i}}\mathbf{s}_i^{(k+1)} + \frac{1}{\mu_{j,i}}\mathbf{s}_j^{(k+1)}\right]. \tag{16}$$

This approximation avoids the exchange of $\mathbf{z}_j$. Note, that $\tilde{\mathbf{z}}_{i,j}$ has to be used for the updates of both $\mathbf{s}_j$ and $\boldsymbol{\lambda}_{i,j}$, because otherwise an exchange of Lagrange variables between neighboring nodes as in [5] would be required, thwarting the desired reduction of the communication effort. Furthermore note, that $\mathbf{z}_j$ does not need to be approximated, since it only requires variables $\mathbf{s}_i$ available from the previous broadcast.

IV. Simulation results

In the following, the proposed algorithm and its variant using the approximation (16) will be compared to the algorithm in [5] by means of Monte Carlo computer simulations. As figures of merit, the number of iterations n_{it} required until the stopping criterion is met and the resulting overall squared error

$$\text{OSE} = \sum_{j=1}^{J} ||\mathbf{s} - \mathbf{s}_j^{(n_{\text{it}})}||^2 \tag{17}$$

shall be employed. For the stopping criterion, the norm of the gradient of (4) was evaluated. The updating loop was terminated[1] as soon as it fell below 0.1. Since the performance of the algorithm depends on the properties of $\mathbf{s}_j$, $\mathbf{H}_j$ and $\mathbf{n}_j$, those variables were generated randomly. Mean value and standard deviation of resulting n_{it} and OSE are approximated based on 1000 realizations of the random variables.

Of course, the convergence speed and the the approximation's impact on it also depend on the network topology. To investigate its effect, different, randomly generated networks are compared. These are generated by placing J nodes randomly in a unit square. Two nodes are assumed to share a link if their distance lies below 0.5.

A. Convergence behavior for random network

In Fig. 2, the behavior of the OSE over iteration index k is depicted for one instance of a system with $J = 6$ nodes, $M = 5$ and $N = 2$. As elements for $\mathbf{s}$ and $\mathbf{H}_j$, real valued, zero mean gaussian distributed values of unit variance were chosen. The elements of the noise vectors $\mathbf{n}_j$ were generated analogously, but with variance 0.2. The variables $\mu_{i,j}$ have been fixed to a value of 1. It can be seen that the proposed algorithm converges significantly faster than the algorithm in [5]. The approximation (16) deteriorates the

[1] The authors are well aware that this central criterion cannot be implemented in a distributed network, but the design of a practical stopping criterion is beyond the scope of this paper and subject of future work.

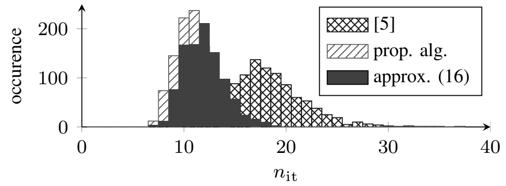

Fig. 3. Statistics of n_{it} for 1000 different realizations of $\mathbf{s}$, $\mathbf{H}_j$ and $\mathbf{n}_j$.

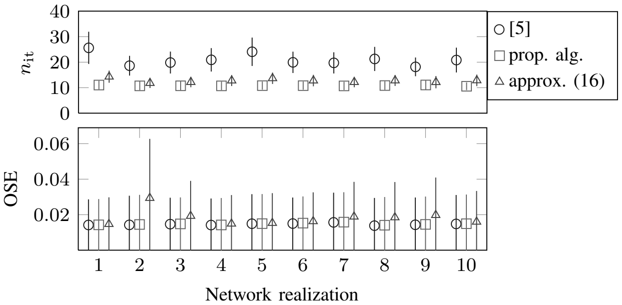

Fig. 4. Mean ± standard deviation of n_{it} (top) and resulting OSE (bottom) for 10 different random network realizations with 1000 random signal, channel, noise realizations each.

convergence only moderately. To ensure a fair comparison, the stepsize parameter in [5] has been chosen such that the update equation (12) and its counterpart in [5] are identical. This effect can be explained by the fact that within one iteration, 2 exchanges of information-bearing variables take place, so that not only information of the neighbors, but also of the neighbors' neighbors is incorporated in the current estimate $\mathbf{s}_j$. The approximation omits certain terms of the sum in (14) and thus discards information, which slows down the convergence process.

In order to corroborate the superiority of the proposed algorithm and the legitimation of the approximation, for 1000 different realizations of $\mathbf{s}$, $\mathbf{H}_j$ and $\mathbf{n}_j$, the number of iterations n_{it} until the stopping criterion is met has been determined. The resulting histogram is shown in Fig. 3. In average, the proposed algorithm requires $\overline{n}_{\mathrm{it}} = 11$ iterations, each with $J = 6$ broadcasts of $\mathbf{z}_j$ and $\mathbf{s}_j$, consisting of $N = 2$ elements each, resulting in the broadcast of 264 numbers in total. The algorithm in [5] requires in average $\overline{n}_{\mathrm{it}} = 18$ iterations, each with broadcast of all $\mathbf{s}_j$ and unicast of 2 Lagrange variables per edge of the network, consisting of $N = 2$ elements each. With 10 edges present in the network under investigation, we end up with a broadcast of 216 numbers and unicast of 720 numbers. The approximation halves the communication effort per iteration compared to the exact algorithm, but requires $\overline{n}_{\mathrm{it}} = 12$ iterations in average, resulting in 144 numbers to be broadcast.

The algorithms' dependency on the network topology has been investigated by repeating above simulations for 10 more random networks. The resulting mean ± standard deviation of n_{it} is depicted in Fig. 4 (top). Since the required number of iterations does not necessarily give an indication on the estimation accuracy, the resulting mean OSE ± standard deviation is shown below. It can be seen that the proposed algorithm always requires fewer iterations than [5] at identical estimation accuracy. The approximation in most cases increases the number of required iterations only slightly compared to the exact solution, but in general leads to an increased residual error.

V. Conclusion and Future Work

We have developed an algorithm for consensus-based distributed LS estimation which exhibits a reduced communication effort, while keeping the estimation accuracy. For verification, we performed computer simulations which additionally indicate that the required number of iterations is reduced compared to the state of the art. Apart from this algorithm which solved the LS problem exactly, additionally a heuristic approximation is proposed which is able to reduce the communication overhead even further. The simulative analysis of this approximation suggests only a small loss of estimation accuracy and increase of required number of iterations compared to the exact solution.

The algorithm developed in this paper can be applied to underdetermined local problem without any modification. Preliminary investigations have shown that also in such systems, a reduction of communication overhead and required number of iterations compared to existing algorithms can be observed. In future work, the applicability of the approximation introduced herein to such scenarios will be examined. The analysis of error-prone inter-node links and their effect on the algorithms' performance is another important task to be approached.

References

[1] Z. Chair and P. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-22, no. 1, pp. 98–101, Jan. 1986.

[2] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.

[3] I. Schizas, G. Mateos, and G. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365–2382, June 2009.

[4] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2320–2330, May 2011.

[5] H. Zhu, A. Cano, and G. Giannakis, "Distributed consensus-based demodulation: algorithms and error analysis," *IEEE Trans. Wireless Commun.*, vol. 9, no. 6, pp. 2044–2054, June 2010.

[6] H. Zhu, G. Giannakis, and A. Cano, "Distributed in-network channel decoding," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3970–3983, Oct. 2009.

[7] V. Saligrama, M. Alanyali, and O. Savas, "Distributed detection in sensor networks with packet losses and finite capacity links," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4118–4132, Nov. 2006.

[8] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[9] F. Cattivelli and A. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[10] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd edition. Springer Science+Business Media, 2006.

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