Distributed Relaxation on Augmented Lagrangian for Consensus based Estimation in Sensor Networks

Guang Xu, Henning Paul, Armin Dekorsy
Department of Communications Engineering, University of Bremen, 28359 Bremen, Germany
Email: {xu,paul,dekorsy}@ant.uni-bremen.de

***Abstract*—This paper presents new distributed estimation algorithms adopting the well-known relaxation methods Jacobi and Gauss-Seidel (GS) to solve consensus based estimation problems in sensor networks. In the algorithms, all sensor nodes work cooperatively with one-hop data exchange to estimate the signals emitted from some information sources using iterative processing. Consequently, a lot of communication overhead will be produced by a parallel processing of the Jacobi-based algorithm. In order to reduce the overhead, the GS-based algorithm with sequential update is used to decrease the overhead per update step. Additionally, the convergence of both distributed algorithms can be accelerated by the successive over relaxation (SOR) method leading to another distributed algorithm, and the corresponding overhead in the distributed processing will be further reduced.**

I. INTRODUCTION

In wireless sensor networks, a group of sensor nodes aims to detect the signals emitted from some information sources. A common estimate on the sources messages is required throughout the whole network. To this end, either centralized or distributed estimation can be used [1]. For the centralized approach, all data is forwarded by local nodes to a central node that performs the estimation, while for the distributed case, each node performs the estimation locally with some information exchanged between neighboring nodes. Some algorithms based on primal and dual decomposition [2], e.g., the distributed consensus estimation (DiCE) algorithm [3] were applied for such distributed estimation. Another algorithm called Fast-DiCE [4] was proposed to accelerate the convergence of the DiCE algorithm. Here, our new approaches, which are based on the relaxation method [5] for distributed processing, can be used to solve the same consensus estimation problem as well. The performance of these algorithms in terms of convergence and communication overhead will be evaluated.

II. SYSTEM DESCRIPTION

Fig. 1 shows a sensor network composed of a set of nodes $\mathcal{J} = \{1, ..., J\}$ equipped with N_R receive antennas each. These nodes are forming a random graph with total E edges. It is assumed that each node $j \in \mathcal{J}$ can only exchange information with its neighboring nodes $\mathcal{N}_j \subseteq \mathcal{J}$. A local observation $\mathbf{y}_j \in \mathbb{C}^{N_\mathrm{R}\times 1}$ of the total N_I input signals from K sources $\boldsymbol{x} \in \mathbb{C}^{N_\mathrm{I}\times 1}$ is made by each node j. Such local observations can be modeled as $\mathbf{y}_j = \mathbf{H}_j\boldsymbol{x} + \mathbf{n}_j$ with channel matrix $\mathbf{H}_j \in \mathbb{C}^{N_\mathrm{R}\times N_\mathrm{I}}$ and additive Gaussian noise $\mathbf{n}_j \in \mathbb{C}^{N_\mathrm{R}\times 1}$. To recover the source signals $\boldsymbol{x}$ among all nodes, information available in the whole network can be aggregated at a central node to perform, e.g., a centralized zero forcing (ZF) solution $\mathbf{x}_\mathrm{ZF} = (\mathbf{H}_\mathrm{cen}^\mathrm{H}\mathbf{H}_\mathrm{cen})^{-1}\mathbf{H}_\mathrm{cen}^\mathrm{H}\mathbf{y}$ based on the least square (LS) criterion with a stacked vector $\mathbf{y} = [\mathbf{y}_1^\mathrm{T}, \dots, \mathbf{y}_J^\mathrm{T}]^\mathrm{T} \in \mathbb{C}^{JN_\mathrm{R}\times 1}$ and a stacked channel matrix $\mathbf{H}_\mathrm{cen} = [\mathbf{H}_1^\mathrm{T}, \dots, \mathbf{H}_J^\mathrm{T}]^\mathrm{T} \in \mathbb{C}^{JN_\mathrm{R}\times N_\mathrm{I}}$. For distributed estimation, each node can also use the LS criterion to solve for the estimate $\mathbf{x}_j, j \in \mathcal{J}$ in a distributed fashion employing consensus constraints:

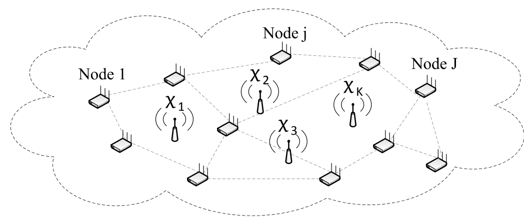

Fig. 1. A sensor network with J nodes receiving messages from K sources

$$\mathbf{x}_j = \arg\min_{\mathbf{x}'_j} \frac{1}{2}\sum_{j=1}^{J}\left\|\mathbf{y}_j - \mathbf{H}_j\mathbf{x}'_j\right\|^2, \text{s.t. } \mathbf{x}_j = \mathbf{x}_i, \forall i \in \mathcal{N}_j. \quad (1)$$

Such a constrained convex problem can be solved e.g., by the DiCE algorithm using a decoupling approach to minimize the augmented Lagrangian (AL) [2] on (1) among nodes in parallel. For our new approach, we reformulate the distributed problem (1) using the stacked estimates vector $\mathbf{x} = [\mathbf{x}_1^\mathrm{T}, \cdots, \mathbf{x}_J^\mathrm{T}]^\mathrm{T} \in \mathbb{C}^{JN_\mathrm{I}\times 1}$, the stacked block diagonal channel matrix $\mathbf{H} = \mathrm{blkdiag}(\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_J) \in \mathbb{C}^{JN_\mathrm{R}\times JN_\mathrm{I}}$, and a matrix $\mathbf{A}$ containing block elements $\mathbf{A}_{ij} \in \{\mathbf{I}, -\mathbf{I}, \mathbf{0}\}$ to represent the constraints in matrix form:

$$\mathbf{x} = \arg\min_{\mathbf{x}'} \frac{1}{2}\left\|\mathbf{y} - \mathbf{H}\mathbf{x}'\right\|^2, \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{0}. \quad (2)$$

The corresponding Lagrangian cost function is formulated as:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{H}\mathbf{x}\right\|^2 - \boldsymbol{\lambda}^\mathrm{T}\mathbf{A}\mathbf{x} + \frac{1}{2\mu}\|\mathbf{A}\mathbf{x}\|^2 \quad (3)$$

with stacked multipliers vector $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^\mathrm{T}, \dots, \boldsymbol{\lambda}_J^\mathrm{T}]^\mathrm{T} \in \mathbb{C}^{2EN_\mathrm{I}\times 1}$ and penalty parameter μ. The stacked estimate $\mathbf{x}$ can be solved by setting $\partial\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})/\partial\mathbf{x} = 0$ and obtaining:

$$(\mathbf{H}^\mathrm{H}\mathbf{H} + \frac{1}{\mu}\mathbf{A}^\mathrm{T}\mathbf{A})\mathbf{x} = \mathbf{H}^\mathrm{H}\mathbf{y} + \mathbf{A}^\mathrm{T}\boldsymbol{\lambda} \quad (4)$$

III. DISTRIBUTED RELAXATION BASED ALGORITHMS

Based on the linear equation (4), the estimate $\mathbf{x}$ can be solved in a distributed fashion using the Jacobi method [5]. As a consequence, all components in $\mathbf{x}$, i.e., local estimates

$\mathbf{x}_j, j \in \mathcal{J}$ can be updated in parallel. The update of local estimate $\mathbf{x}_j$ of the Jacobi-based consensus estimation (Jacobi-CE) algorithm is given by:

$$\mathbf{x}_j^{k+1} = \left(\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j + \frac{2|\mathcal{N}_j|}{\mu}\mathbf{I}_{N_{\mathrm{I}}}\right)^{-1}\left(\mathbf{H}_j^{\mathrm{H}}\mathbf{y}_j + \sum_{i\in\mathcal{N}_j}\boldsymbol{\lambda}_{ji}^k - \sum_{i\in\mathcal{N}_j}\boldsymbol{\lambda}_{ij}^k + \sum_{i\in\mathcal{N}_j}\frac{2\mathbf{x}_i^k}{\mu}\right). \quad (5)$$

For the update of $\mathbf{x}_j^{k+1}$ at iteration $k+1$, each node j only requires the estimates $\mathbf{x}_i^k$ and multipliers $\boldsymbol{\lambda}_{ij}^k$ of the previous iteration k from its neighboring nodes i. Once the local estimates $\mathbf{x}_j^{k+1}$ and multipliers $\boldsymbol{\lambda}_{ji}^{k+1}$ are updated, every node exchanges the newest information with its neighboring nodes and starts the update in the next iteration. However, due to the information exchange between nodes, large communication effort over node-node links is required due to the parallel processing per update step. In order to reduce the overhead, the GS method [5] is applied with a sequential update per iteration leading to the GS based consensus estimation (GS-CE) algorithm. The update of estimate $\mathbf{x}_j$ is given by:

$$\mathbf{x}_j^{k+1} = \left(\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j + \frac{2|\mathcal{N}_j|}{\mu}\mathbf{I}_{N_{\mathrm{I}}}\right)^{-1}\left(\mathbf{H}_j^{\mathrm{H}}\mathbf{y}_j + \sum_{i\in\mathcal{N}_j}\boldsymbol{\lambda}_{ji}^k + \sum_{i\in\mathcal{N}_j^-}\left(\frac{2\mathbf{x}_i^{k+1}}{\mu} - \boldsymbol{\lambda}_{ij}^{k+1}\right) + \sum_{i\in\mathcal{N}_j^+}\left(\frac{2\mathbf{x}_i^{k}}{\mu} - \boldsymbol{\lambda}_{ij}^{k}\right)\right), \quad (6)$$

where $\mathcal{N}_j^-$ denotes the set of neighboring nodes of node j with indices $i < j$ and $\mathcal{N}_j^+$ vice versa. The local estimate $\mathbf{x}_j^{k+1}$ in iteration $k+1$ at node j is updated with the newest estimates $\mathbf{x}_i^{k+1}$ and multipliers $\boldsymbol{\lambda}_{ij}^{k+1}$ of the neighboring nodes $i \in \mathcal{N}_j^-$ as well as $\mathbf{x}_i^k$ and $\boldsymbol{\lambda}_{ij}^k, i \in \mathcal{N}_j^+$ from previous iteration k. When both $\mathbf{x}_j^{k+1}$ and $\boldsymbol{\lambda}_{ji}^{k+1}$ are updated, they are sent to the neighboring nodes $i \in \mathcal{N}_j$, then node $j+1$ starts to update. Here, the estimates $\mathbf{x}_j$ are updated sequentially from node to node per iteration. Therefore, only a small quantity of overhead is produced per update step by the sequential update in GS-CE compared to the parallel update in Jacobi-CE.

Moreover, as [5] indicates that the convergence of both Jacobi and GS methods can be accelerated by a successive over-relaxation method, we apply this approach to the above algorithms leading to the Jacobi-based SOR on consensus estimation (JSOR-CE) and GS-based SOR on consensus estimation (GSSOR-CE) algorithms, respectively. In both algorithms, the local estimates $\mathbf{x}_j^{k+1}$ are further regularized with a relaxation parameter $\omega > 1$ in order to speed up the update. But due to space limitation, more details will be given in the final paper.

IV. PERFORMANCE EVALUATION

In the following, all algorithms are evaluated by means of the Monte Carlo method for a randomly generated network of $J = 10$ nodes and $K = 4$ sources. Both nodes and sources are assumed to be equipped with a single antenna, i.e, $N_{\mathrm{R}} = N_{\mathrm{T}} = 1$. For the evaluation, the mean squared error (MSE) on the estimates is averaged over all nodes. In Fig. 2a, it can be observed that all algorithms converge to the central solution, but due to the sequential update, the GS-based algorithms show slow convergence. Compared to the algorithms from the state of the art, only JSOR-CE shows a faster convergence than the DiCE algorithm. However, a significant overhead can be saved by GSSOR-CE and GS-CE as shown in Fig. 2b, since per step only one node needs to transmit the information. For the JSOR-CE algorithm, much overhead is saved compared to Fast-DiCE where more information is exchanged between nodes per step [4], while the Jacobi-CE algorithm shows a similar performance to DiCE.

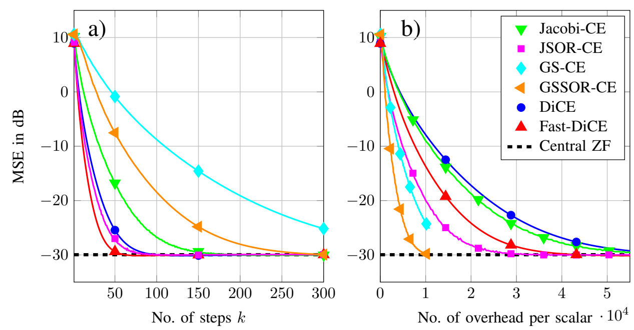

Fig. 2. MSE performance averaged over 1000 trials vs. a) No. of steps and b) total no. of overhead. $J = 10, K = 4, N_{\mathrm{R}} = N_{\mathrm{T}} = 1$, random topology, SNR = 10dB

V. CONCLUSION AND OUTLOOK

In this extended abstract, we presented Jacobi and Gauss-Seidel based distributed algorithms for consensus estimation in sensor networks. The Jacobi-CE algorithm adopts parallel processing among nodes resulting in a faster convergence, but it produces higher overhead compared to the GS-CE algorithm where the nodes update sequentially leading to relative low overhead per step. Both algorithms are accelerated by the SOR method leading to a further reduction of overhead. In the final paper, more details about the presented algorithms will be given, and we will further investigate a hybrid algorithm that takes advantage of both parallel and sequential processing for the distributed estimation. All algorithms will also be evaluated in a scenario with erroneous inter-node links. Furthermore, a full bibliography will be given in the final paper.

REFERENCES

[1] A. Swami, Q. Zhao, Y. Hong, and L. Tong, *Wireless Sensor Networks: Signal Processing and Communications*. Wiley, Nov. 2007.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[3] H. Paul, J. Fliege, and A. Dekorsy, "In-Network-Processing: Distributed Consensus-Based Linear Estimation," *IEEE Communications Letters*, vol. 17, no. 1, pp. 59–62, Jan. 2013.

[4] G. Xu, H. Paul, D. Wübben, and A. Dekorsy, "Fast Distributed Consensus-based Estimation for Cooperative Wireless Sensor Networks," in *18th International ITG Workshop on Smart Antennas (WSA 2014)*, Erlangen, Germany, Mar. 2014.

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