MODEL-DISTRIBUTED SOLUTION OF REGULARIZED LEAST-SQUARES PROBLEM OVER SENSOR NETWORKS

Reza Arablouei[1], Kutluyıl Doğançay[2], Stefan Werner[3], and Yih-Fang Huang[4]

[12]School of Engineering, University of South Australia, Mawson Lakes, SA, Australia

[3]Dept. of Signal Processing and Acoustics, School of Electrical Engineering, Aalto University, Finland

[4]Department of Electrical Engineering, University of Notre Dame, IN, USA

ABSTRACT

We develop a fully-distributed iterative algorithm for finding a model-distributed least-squares solution of systems of linear equations over sensor networks. Here, model-distributed means the solution vector is distributed across the network rather than being replicated at each node. For this purpose, we devise a dual regularized least-squares problem via a suitable decomposition of the normal equations associated with the original problem. The resultant dual problem can be solved in a fully-decentralized and iterative manner by means of the diffusion-based Pareto optimization strategy. We verify the usefulness of the proposed algorithm via both theoretical analysis and numerical examples.

***Index Terms*—**Model distribution; distributed solvers; iterative solvers; diffusion adaptation; wireless sensor networks; least squares.

1. INTRODUCTION

Computational algorithms for solving systems of linear equations constitute a significant part of numerical linear algebra and play a key role for many signal-processing applications. As a standard approach, the method of least squares (LS) is used to yield an approximate solution of over-determined systems of linear equations in which there are more equations than unknowns [1].

Consider the following over-determined system of linear equations:

$$\mathbf{Ax} \approx \mathbf{b} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{K\times L}$ is the system matrix, $\mathbf{x} \in \mathbb{R}^{L\times 1}$ is the solution vector, $\mathbf{b} \in \mathbb{R}^{K\times 1}$ is the observation vector, $L \in \mathbb{N}$ is the system order, and $K \in \mathbb{N} \geq L$ is the number of equations. The LS problem associated with solving (1) for $\mathbf{x}$ is posed as the following minimization problem:

$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|^2 \tag{2}$$

In many applications, solutions to systems of linear equations or their corresponding LS problems need be obtained over a sensor network where each node houses a part of the data, $\mathbf{A}$ and $\mathbf{b}$ [2]-[4]. Nodes of a sensor network are normally capable of acting autonomously but often pursue a common goal through collaboration. Critical resources of the sensor nodes, specifically their energy supply and computational capacity, are typically constrained. Therefore, it is imperative to limit the internode communications to only within the immediate neighborhood as well as minimize the computational complexity when performing any task over a sensor network.

Several techniques have been proposed to solve the LS problem (2) over a network of agents (nodes) in a fully distributed manner, i.e., without the use of any fusion center, clustering, or multi-hop communication. Among them are the adaptive distributed estimation algorithms based on consensus, e.g., [5]-[8], or diffusion, e.g., [9]-[21], together with the push-sum-based algorithm of [2]. In these works, $\mathbf{A}$ is assumed to be distributed over the network nodes in a row-wise fashion. The kth entry of $\mathbf{b}$ is also assumed to be observed in the node that has the kth row of $\mathbf{A}$. Thus, each node calculates an estimate replica of the solution vector $\mathbf{x}$ and cooperation of the nodes helps improve the overall estimation performance.

In this paper, we consider the case where the matrix $\mathbf{A}$ is distributed among $N \in \mathbb{N}$ nodes of a connected sensor network in a *column-wise* fashion and $\mathbf{b}$ is accessible by at least one node. Accordingly, the matrix $\mathbf{A}$ is divided by columns into N blocks and each node $i \in \{1,2,\dots,N\}$ has access to a block of $L_i \in \mathbb{N}$ columns, denoted by $\mathbf{A}_i \in \mathbb{R}^{K\times L_i}$. Consistent with the partitioning of $\mathbf{A}$, the solution vector $\mathbf{x}$ is also divided into N subvectors so that each node i is responsible for its subvector, denoted by $\mathbf{x}_i \in \mathbb{R}^{L_i\times 1}$, that corresponds to $\mathbf{A}_i$. Hence, we have

$$\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2|\dots|\mathbf{A}_N],$$
$$\mathbf{x} = [\mathbf{x}_1^\top|\mathbf{x}_2^\top|\dots|\mathbf{x}_N^\top]^\top,$$
$$\sum_{i=1}^{N} L_i = L.$$

This scenario is referred to here as the *model-distributed* LS problem. To the best of our knowledge, such a problem has previously only been considered in [22]. However, the algorithm proposed in [22], which is based on the idea of multi-splitting of $\mathbf{A}$ [23], requires flooding of certain data at each iteration, which necessitates broadcast (global) communications. Therefore, this algorithm is not suitable for sensor network applications, even so it can be effectively implemented over a parallel computing platform. Moreover, although related in spirit to our work, [34] deals with the problem of model-distributed dictionary learning over sensor networks.

Our approach to the solution of the considered model-distributed LS problem is through the definition of a distributed convex optimization problem, which is also of LS type and can be viewed as the dual of the original problem. We solve this dual problem using the gradient-descent diffuse strategy for distributed Pareto optimization developed in [24], [25] and further studied in [3], [4], [26]-[28]. The proposed algorithm is iterative, fully distributed, has a per-iteration per-node computational complexity of $\mathcal{O}(L_i K)$, and requires the nodes to share a vector of size K with their immediate neighbors at each iteration. Interestingly, nodes need not share their partial knowledge of $\mathbf{A}$ or $\mathbf{x}$ with the other nodes but exchange only their estimates of a common vector that is the solution of the defined dual problem. This common vector cannot be used to extrapolate any information about the primary data of other nodes. Thus, the proposed algorithm in fact respects the possible data privacy of the nodes.

We analyze the convergence performance of the proposed algorithm and find a region for its step-size that guarantees its stability. We also show that the algorithm converges to the exact solution in all the nodes when the step-size tends to zero. Numerical examples corroborate the effectiveness of the proposed algorithm.

2. ALGORITHM DESCRIPTION

Solving the system of linear equations (1) for $\mathbf{x}$ can be cast as a regularized LS problem expressed by

$$\min_{\mathbf{x}}\{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2+\eta\|\mathbf{x}\|^2\} \tag{3}$$

where $\|\cdot\|$ stands for the Euclidean norm and $\eta \in \mathbb{R}_{>0}$ is the regularization parameter. The main reason for regularization is to prevent the problem from being ill-posed due to possible rank-deficiency of $\mathbf{A}$.

The normal equations associated with (3) are written as

$$(\mathbf{A}^\top\mathbf{A}+\eta\mathbf{I}_L)\mathbf{x}=\mathbf{A}^\top\mathbf{b}$$

where $\mathbf{I}_L$ is the $L\times L$ identity matrix. Consequently, $\mathbf{x}$ is given by

$$\begin{aligned}\mathbf{x}&=(\mathbf{A}^\top\mathbf{A}+\eta\mathbf{I}_L)^{-1}\mathbf{A}^\top\mathbf{b}\\&=\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top+\eta\mathbf{I}_K)^{-1}\mathbf{b}\\&=\mathbf{A}^\top\mathbf{f}^o\end{aligned}$$

where

$$\mathbf{f}^o=(\mathbf{A}\mathbf{A}^\top+\eta\mathbf{I}_K)^{-1}\mathbf{b}\in\mathbb{R}^{K\times 1}.$$

If $\mathbf{f}^o$ is known globally throughout the network, each node i can calculate its associated part of $\mathbf{x}$ via

$$\mathbf{x}_i=\mathbf{A}_i^\top\mathbf{f}^o.$$

In order to compute $\mathbf{f}^o$ in all nodes using only in-network processing and local communications, we employ the diffusion strategy for distributed Pareto optimization developed in [24], [25]. To this end, we define the following quadratic global cost function whose unique minimizer is $\mathbf{f}^o$:

$$\mathcal{J}(\mathbf{f})=\frac{1}{2}\mathbf{f}^\top(\mathbf{A}\mathbf{A}^\top+\eta\mathbf{I}_K)\mathbf{f}-\mathbf{f}^\top\mathbf{b}. \tag{4}$$

Since, we have

$$\mathbf{A}\mathbf{A}^\top=\sum_{i=1}^{N}\mathbf{A}_i\mathbf{A}_i^\top,$$

the function $\mathcal{J}(\mathbf{f})$ can be written as the sum of node-specific individual cost functions:

$$\mathcal{J}(\mathbf{f})=\sum_{i=1}^{N}J_i(\mathbf{f})$$

where

$$J_i(\mathbf{f})=\frac{1}{2}\mathbf{f}^\top\left(\mathbf{A}_i\mathbf{A}_i^\top+\frac{\eta}{N}\mathbf{I}\right)\mathbf{f}-\frac{\delta_i}{B}\mathbf{f}^\top\mathbf{b},$$

$$\delta_i=\begin{cases}1 & \text{if } \mathbf{b} \text{ is available at node } i\\ 0 & \text{if } \mathbf{b} \text{ is not available at node } i,\end{cases}$$

and $B\in\mathbb{N}$ is the number of nodes that have access to $\mathbf{b}$. Using the adapt-then-combine diffusion strategy, we can iteratively minimize (4) over the network in a fully-distributed manner and obtain an estimate of $\mathbf{f}^o$ at each node i and iteration n, denoted by $\mathbf{f}_{i,n}$. The relevant recursive iterations take the following form:

$$\begin{aligned}\mathbf{g}_{i,n-1}&=\mathbf{f}_{i,n-1}-\mu\,\nabla_{\mathbf{f}}J_i(\mathbf{f}_{i,n-1})\\&=\left[\mathbf{I}_K-\mu\left(\mathbf{A}_i\mathbf{A}_i^\top+\frac{\eta}{N}\mathbf{I}_K\right)\right]\mathbf{f}_{i,n-1}+\frac{\mu\delta_i}{B}\mathbf{b}\end{aligned} \tag{5}$$

$$\mathbf{f}_{i,n}=\sum_{j\in\mathcal{N}_i}c_{i,j}\mathbf{g}_{j,n} \tag{6}$$

where $\mathbf{g}_{i,n}$ is the intermediate estimate at node i and iteration n and $\mu\in\mathbb{R}_{>0}$ is the step-size parameter. The set $\mathcal{N}_i$ denotes the closed neighborhood of node i, i.e., it comprises all nodes that are connected to node i within one hop including the node i itself. The combination weights $\{c_{i,j}\}$ are positive real numbers that satisfy

$$\sum_{j=1}^{N}c_{i,j}=\sum_{i=1}^{N}c_{i,j}=1$$

$$c_{i,j}=0\ \text{ if } j\notin\mathcal{N}_i,$$

which implies that the combination matrix $\mathbf{C}$ composed of $\{c_{i,j}\}$ is doubly-stochastic. Obviously, nodes share their intermediate estimates only with their neighbors at each iteration.

3. CONVERGENCE ANALYSIS

The cost function of node i, $J_i(\mathbf{f})$, is twice continuously differentiable and has an invariant positive-definite Hessian matrix:

$$\begin{aligned}\mathbf{H}_i &= \nabla_{\mathbf{f}}^2 J_i(\mathbf{f}) \\ &= \left(\mathbf{A}_i\mathbf{A}_i^\top + \frac{\eta}{N}\mathbf{I}_K\right).\end{aligned} \tag{7}$$

Therefore, we have [29]

$$\nabla_{\mathbf{f}} J_i(\mathbf{f}_{i,n-1}) = \nabla_{\mathbf{f}} J_i(\mathbf{f}^o) - \mathbf{H}_i(\mathbf{f}^o - \mathbf{f}_{i,n-1}). \tag{8}$$

Subtracting (5)-(6) from $\mathbf{f}^o$ and using (7)-(8) give

$$\begin{aligned}\check{\mathbf{g}}_{i,n-1} = &\left[\mathbf{I}_K - \mu\left(\mathbf{A}_i\mathbf{A}_i^\top + \frac{\eta}{N}\mathbf{I}_K\right)\right]\check{\mathbf{f}}_{i,n-1} \\ &+ \mu\left(\mathbf{A}_i\mathbf{A}_i^\top + \frac{\eta}{N}\mathbf{I}_K\right)\mathbf{f}^o - \mu\frac{\delta_i}{B}\mathbf{b}\end{aligned} \tag{9}$$

$$\check{\mathbf{f}}_{i,n} = \sum_{j\in\mathcal{N}_i} c_{i,j}\check{\mathbf{g}}_{j,n} \tag{10}$$

where

$$\check{\mathbf{g}}_{i,n} = \mathbf{f}^o - \mathbf{g}_{i,n}$$
$$\check{\mathbf{f}}_{i,n} = \mathbf{f}^o - \mathbf{f}_{i,n}.$$

If we further define

$$\check{\mathbf{g}}_n = \begin{bmatrix}\check{\mathbf{g}}_{1,n}\\ \vdots \\ \check{\mathbf{g}}_{N,n}\end{bmatrix} \text{ and } \check{\mathbf{f}}_n = \begin{bmatrix}\check{\mathbf{f}}_{1,n}\\ \vdots \\ \check{\mathbf{f}}_{N,n}\end{bmatrix}$$

and collect (9) and (10) for all i, we arrive at

$$\check{\mathbf{f}}_n = \acute{\mathbf{C}}(\mathbf{I}_{KN} - \mu\mathbf{D})\check{\mathbf{f}}_{n-1} + \mu\acute{\mathbf{C}}\mathbf{q} \tag{11}$$

where

$$\mathbf{D} = \operatorname{blockdiag}\left\{\mathbf{A}_1\mathbf{A}_1^\top + \frac{\eta}{N}\mathbf{I}_K, \dots, \mathbf{A}_N\mathbf{A}_N^\top + \frac{\eta}{N}\mathbf{I}_K\right\},$$

$$\mathbf{q} = \begin{bmatrix}\left(\mathbf{A}_1\mathbf{A}_1^\top + \frac{\eta}{N}\mathbf{I}_K\right)\mathbf{f}^o - \frac{\delta_1}{B}\mathbf{b} \\ \vdots \\ \left(\mathbf{A}_N\mathbf{A}_N^\top + \frac{\eta}{N}\mathbf{I}_K\right)\mathbf{f}^o - \frac{\delta_N}{B}\mathbf{b}\end{bmatrix},$$

$$\acute{\mathbf{C}} = \mathbf{C}\otimes\mathbf{I}_K,$$

and $\otimes$ is Kronecker product.

As $\mathbf{C}$ is doubly-stochastic, $\acute{\mathbf{C}}$ is also doubly-stochastic and can be associated with a non-expansive mapping. Therefore, the recursion (11) is stable and convergent if $\mathbf{I}_{KN} - \mu\mathbf{D}$ is stable, i.e.,

$$r\{\mathbf{I}_{KN} - \mu\mathbf{D}\} < 1$$

or

$$0 < \mu < \frac{2}{r\{\mathbf{D}\}} \tag{12}$$

where $r\{\cdot\}$ denotes the spectral radius. Considering the definition of $\mathbf{D}$, its spectral radius is calculated as

$$\begin{aligned}r\{\mathbf{D}\} &= \max_i r\{\mathbf{A}_i\mathbf{A}_i^\top\} + \frac{\eta}{N} \\ &= \max_i r\{\mathbf{A}_i^\top\mathbf{A}_i\} + \frac{\eta}{N}.\end{aligned}$$

If μ is properly chosen to satisfy (12), (11) will converge to

$$\begin{aligned}\check{\mathbf{f}}_\infty &= \lim_{n\to\infty}\check{\mathbf{f}}_n \\ &= \mu\left(\mathbf{I}_{KN} - \acute{\mathbf{C}} + \mu\acute{\mathbf{C}}\mathbf{D}\right)^{-1}\acute{\mathbf{C}}\mathbf{q}.\end{aligned}$$

Being doubly-stochastic, $\mathbf{C}$ has a unique eigenvalue at one and both its right and left eigenvectors corresponding to this eigenvalue are $1/\sqrt{N}\mathbf{1}_N$ where $\mathbf{1}_N$ is the $N\times 1$ all-ones vector [30]. Therefore, $\mathbf{I}_{KN} - \acute{\mathbf{C}}$ has K eigenvalues at zero with the corresponding (left and right) eigenvectors that are the columns of $1/\sqrt{N}\mathbf{1}_N\otimes\mathbf{I}_K$. Thus, according to Proposition 2 of [31], which can be seen as a substitute of the l'Hôpital's rule for matrices, we have

$$\lim_{\mu\to 0}\mu\left(\mathbf{I}_{KN} - \acute{\mathbf{C}} + \mu\acute{\mathbf{C}}\mathbf{D}\right)^{-1} = \mathbf{S}\left(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top\otimes\mathbf{I}_K\right)$$

where

$$\mathbf{S} = \left[\left(\mathbf{I}_{KN} - \acute{\mathbf{C}}\right)^\top\left(\mathbf{I}_{KN} - \acute{\mathbf{C}}\right) + \mathbf{D}\acute{\mathbf{C}}^\top\acute{\mathbf{C}}\mathbf{D}\right]^{-1}\mathbf{D}\acute{\mathbf{C}}^\top.$$

Consequently, we have

$$\begin{aligned}\lim_{\mu\to 0}\check{\mathbf{f}}_\infty &= \mathbf{S}\left(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top\otimes\mathbf{I}_K\right)\acute{\mathbf{C}}\mathbf{q} \\ &= \frac{1}{N}\mathbf{S}\begin{bmatrix}\mathbf{I}_K & \cdots & \mathbf{I}_K \\ \vdots & \ddots & \vdots \\ \mathbf{I}_K & \cdots & \mathbf{I}_K\end{bmatrix}\begin{bmatrix}\left(\mathbf{A}_1\mathbf{A}_1^\top + \frac{\eta}{N}\mathbf{I}_K\right)\mathbf{f}^o - \frac{\delta_1}{B}\mathbf{b} \\ \vdots \\ \left(\mathbf{A}_N\mathbf{A}_N^\top + \frac{\eta}{N}\mathbf{I}_K\right)\mathbf{f}^o - \frac{\delta_N}{B}\mathbf{b}\end{bmatrix} \\ &= \frac{1}{N}\mathbf{S}\begin{bmatrix}\left(\sum_{i=1}^N \mathbf{A}_i\mathbf{A}_i^\top + \eta\mathbf{I}_K\right)\mathbf{f}^o - \mathbf{b} \\ \vdots \\ \left(\sum_{i=1}^N \mathbf{A}_i\mathbf{A}_i^\top + \eta\mathbf{I}_K\right)\mathbf{f}^o - \mathbf{b}\end{bmatrix} \\ &= \mathbf{0}_{KN}\end{aligned}$$

where $\mathbf{0}_{KN}$ is the $KN\times 1$ zero vector. This means that $\mathbf{f}_{i,n}$ for all i converge to $\mathbf{f}^o$ when the step-size tends to zero.

4. NUMERICAL EXAMPLES

We consider an application of multichannel system identification [32] over a sensor network with $N = 10$ nodes and a topology as shown in Fig. 1. Each channel has a length of $L_i = 2$ and is identified at one node; hence, we have $L = 20$. We place $K = 50$ regressor vectors in $\mathbf{A}$ as its rows. These vectors are zero-mean multivariate Gaussian with an arbitrary covariance matrix. Each node has access to a $K\times L_i = 50\times 2$ column block of $\mathbf{A}$. The entries of $\mathbf{b}$, denoted by

$b_k \in \mathbb{R}$, are related to the rows of $\mathbf{A}$, denoted by $\mathbf{a}_k \in \mathbb{R}^{1\times L}$, via

$$b_k = \sum_{i=1}^{N} \mathbf{a}_{k,i}\mathbf{h}_i + v_k$$

where $\mathbf{a}_{k,i} \in \mathbb{R}^{1\times L_i}$ is the part of $\mathbf{a}_k$ that is available in node i, $\mathbf{h}_i \in \mathbb{R}^{L_i\times 1}$ is the system parameter vector of the ith channel, and $v_k \in \mathbb{R}$ is zero-mean Gaussian noise with a variance that is set to yield an average signal-to-noise ratio of 10 dB. We use the proposed algorithm to calculate $\hat{\mathbf{h}}_{i,n} = \mathbf{A}_i^\top \mathbf{f}_{i,n}$ as an LS estimate of $\mathbf{h}_i$ at each node i. We use the Metropolis weights [3], [33] for $\mathbf{C}$ such that

$$c_{i,j} = 1/\max\left(|\mathcal{N}_i|, |\mathcal{N}_j|\right)$$

where $|\mathcal{N}_i|$ denotes the cardinality of $\mathcal{N}_i$, i.e., the degree of node i. We also set the regularization parameter δ to 10^{-3}.

In Figs. 2 and 3, we plot the misalignment, defined as

$$\frac{\sum_{i=1}^{N}\left\|\hat{\mathbf{h}}_{i,n} - \mathbf{h}_i\right\|^2}{\sum_{i=1}^{N}\|\mathbf{h}_i\|^2},$$

and the relative misalignment, defined as

$$\left\|\left[\hat{\mathbf{h}}_{1,n}^\top, \dots, \hat{\mathbf{h}}_{N,n}^\top\right]^\top - \hat{\mathbf{h}}\right\|^2 / \|\hat{\mathbf{h}}\|^2,$$

against n, i.e., the number of iterations exercised by the algorithm as (5)-(6), respectively. Here, $\hat{\mathbf{h}}$ is the estimate achieved by the centralized solution and is given by

$$\hat{\mathbf{h}} = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \eta\mathbf{I}_K)^{-1}\mathbf{b}.$$

The curves in Figs. 2 and 3 are averaged over 100 independent trials and are given for different values of the step-size as well as for both cases of having $\mathbf{b}$ available in only one node ($B = 1$) and in all the nodes ($B = N$). It is seen that the step-size governs a trade-off between convergence speed and accuracy of the algorithm. Moreover, the curves associated with the same step-sizes almost overlay confirming that the value of B does not have any significant effect on the performance in the considered scenario.

4. CONCLUSION

We developed a fully-distributed algorithm for solving the regularized linear least-squares problem over a sensor network in a model-distributed way. To this end, we formulated a dual distributed convex optimization problem and utilized the diffusion strategy for its iterative solution. We evaluated the performance of the proposed algorithm through theoretical analysis and simulation examples.

The proposed algorithm is applicable to batch processing. However, extensions to handle streaming data and perform online learning as well as efforts to create resilience against node or link failure are underway.

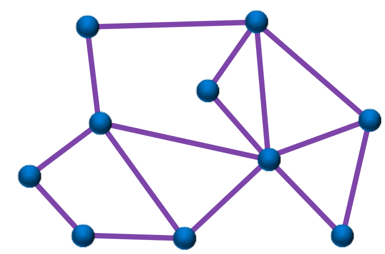

Fig. 1. Topology of the considered sensor network.

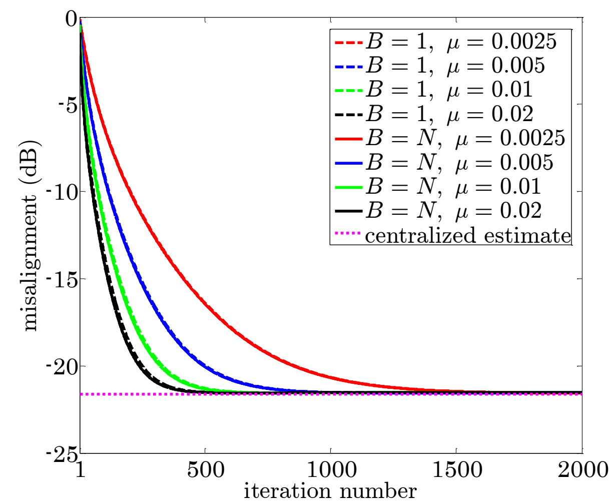

Fig. 2. Misalignment for different values of the step-size when one or all nodes have access to $\mathbf{b}$.

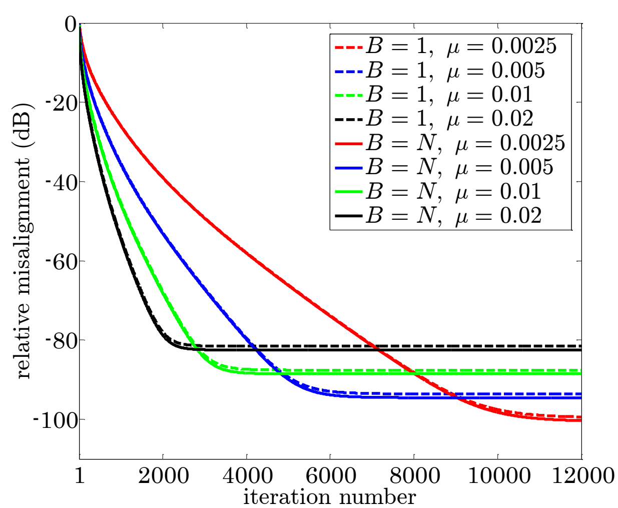

Fig. 3. Relative misalignment for different values of the step-size when one or all nodes have access to $\mathbf{b}$.

5. REFERENCES

[1] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, Philadelphia, PA: SIAM, 2000.

[2] K. E. Prikopa, H. Straková, and W. N. Gansterer, "Analysis and comparison of truly distributed solvers for linear least squares problems on wireless sensor networks," in *Proc. European Conf. Parallel Computing*, Porto, Portugal, Aug. 2014, pp. 403-414.

[3] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311-801, Jul. 2014.

[4] A. H. Sayed, "Adaptive networks," *Proc. IEEE* , vol. 102, no. 4, pp. 460-497, Apr. 2014.

[5] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic, "Decentralized parameter estimation by consensus based stochastic approximation," *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 531–543, Mar. 2011.

[6] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4583–4588, Nov. 2009.

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

[8] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[9] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, vol. 3, R. Chellapa and S. Theodoridis, Eds., Academic Press, 2013, pp. 323-454.

[10] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, vol. 30, pp. 155-171, May 2013.

[11] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[12] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, pp. 1035–1048, Mar. 2010.

[13] R. Arablouei, S. Werner, Y.-F. Huang, and K. Doğançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 472-484, Jan. 2014.

[14] M. O. Sayin and S. S. Kozat, "Compressive diffusion strategies over distributed networks for reduced communication load," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5308–5323, Oct. 15, 2014.

[15] R. Arablouei, K. Doğançay, S. Werner, and Y.-F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3510-3522, Jul. 2014.

[16] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865-1877, May 2008.

[17] R. Arablouei, S. Werner, and K. Doğançay, "Partial-diffusion recursive least-squares estimation over adaptive networks," in *Proc. IEEE Int. Workshop Computational Advances Multi-Sensor Adaptive Process.*, Saint Martin, Dec. 2013, pp. 89-92.

[18] R. Arablouei, K. Doğançay, and S. Werner, "Reduced-complexity distributed least-squares estimation over adaptive networks," in *Proc. IEEE Int. Workshop Signal Process. Advances Wireless Commun.*, Darmstadt, Germany, June 2013, pp. 155-159.

[19] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, pp. 3122–3136, Jul. 2008.

[20] R. Arablouei, S. Werner, K. Doğançay, and Y.-F. Huang, "Analysis of a reduced-communication diffusion LMS algorithm," arXiv:1408.5845.

[21] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based LMS for node-specific adaptive parameter estimation," arXiv:1408.3354.

[22] R. A. Renaut, "A parallel multisplitting solution of the least squares problem," *Numerical Linear Algebra Applicat.*, vol. 5, no. 1, pp. 11-31, 1998.

[23] D. P. O'Leary and R. E. White, "Multi-splitting of matrices and parallel solution of linear systems," *SIAM J. Algebraic Discrete Methods*, vol. 6, no. 4, pp. 630–640, Oct. 1985.

[24] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE J. Select. Topics Signal Process.*, vol. 7, no. 2, pp. 205-220, Apr. 2013.

[25] J. Chen and A. H. Sayed, "Distributed Pareto-optimal solutions via diffusion adaptation," in *Proc. IEEE Workshop Statistical Signal Process.*, Ann Arbor, MI, Aug. 2012, pp. 648-651.

[26] J. Chen and A. H. Sayed, "On the limiting behavior of distributed optimization strategies," in *Proc. Allerton Conf.*, Monticello, IL, Oct. 2012, pp. 1535-1542.

[27] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—Part I: transient analysis," arXiv:1312.7581.

[28] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—Part II: performance analysis," arXiv:1312.7580.

[29] C. T. Kelley, *Iterative Methods for Optimization*, Philadelphia, PA: SIAM, 1999.

[30] R. A. Brualdi, *Combinatorial Matrix Classes*, Cambridge Univ. Press, 2006.

[31] W. Kratz, "A substitute of l'Hospital's rule for matrices," *Proc. Amer. Math. Soc.*, vol. 99, no. 3, pp. 395-402, Mar. 1987.

[32] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*, Springer, 2006.

[33] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[34] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," arXiv:1402.1515.