# DIFFUSION-BASED DISTRIBUTED ADAPTIVE ESTIMATION UTILIZING GRADIENT-DESCENT TOTAL LEAST-SQUARES

*Reza Arablouei*[*], *Stefan Werner*[†], *and Kutluyıl Doğançay*[‡]

[*‡]Institute for Telecommunications Research, University of South Australia, Mawson Lakes SA, Australia
[†]Dept. of Signal Processing and Acoustics, School of Electrical Engineering, Aalto University, Finland
[‡]School of Engineering, University of South Australia, Mawson Lakes SA, Australia

## ABSTRACT

We develop a gradient-descent distributed adaptive estimation strategy that compensates for error in both input and output data. To this end, we utilize the concepts of total least-squares estimation and gradient-descent optimization in conjunction with a recently-proposed framework for diffusion adaptation over networks. The proposed strategy does not require any prior knowledge about the noise variances and has a computational complexity comparable to the diffusion least mean square (DLMS) strategy. Simulation results demonstrate that the proposed strategy provides significantly improved estimation performance compared with the DLMS and bias-compensated DLMS (BC-DLMS) strategies when both the input and output signals are noisy.

***Index Terms*****—**adaptive networks, diffusion adaptation, distributed adaptive filtering, gradient-descent optimization, total least-squares.

## 1. INTRODUCTION

Several types of self-organized systems can be modeled using adaptive networks that perform decentralized information processing and optimization. A group of spatially-dispersed nodes with processing and learning capabilities typically forms an adaptive network. In such networks, the nodes are interconnected within a static or dynamic topology and cooperate with each other via local information exchanges to perform real-time distributed estimation. The nodes can adapt to varying statistical and topographical conditions of the data and the network thanks to the constant dissemination of information across the network [1]-[3].

A considerable body of literature on distributed adaptive estimation over networks has accumulated in recent years (see, e.g., [3]-[25] and the references therein). Of particular interest are the diffusion adaptation strategies that are especially well-suited for applications where all the nodes in the network have a common objective [4]-[12]. Diffusion strategies are powerful techniques that allow adaptive cooperation and learning over networks by relying merely on local interactions and in-network processing. They are also superior to the incremental [13]-[16] and consensus strategies [17]-[25] in various aspects [1], [26].

Prominent among the diffusion strategies is the diffusion least mean square (DLMS) strategy that offers a simple but effective way to implement distributed adaptive filtering over networks [11]. However, the DLMS strategy assumes that the input data at all nodes are observed accurately and only the filter outputs of the nodes are corrupted by noise. This assumption is often unrealistic since several types of error, e.g., sampling, quantization, modeling, and instrument errors, can contribute to the inaccuracy of the observed input data. Therefore, in practice, the DLMS strategy may have a poor estimation performance because of failing to account for the error in the input data.

Bias compensation is one possible approach to enhance the performance of the DLMS strategy at the presence of input data noise [5], [7]. However, this approach requires the exact knowledge of noise variances at the input of all the nodes. Such information is usually hard to obtain beforehand and needs to be estimated during the adaptation process. This in turn imposes processing overhead and inaccuracy.

Total least-squares (TLS) is a fitting method that improves the accuracy of the least-squares estimation techniques when both the input and output data are subject to observational error. TLS minimizes the perturbation in the input and output data that is required to fit the input data to the output observations [27]-[29]. Two distributed TLS algorithms for optimization over networks, which use eigendecomposition of the augmented data covariance matrix and the inverse power iterations (IPI), have been proposed in [17] and [18], respectively. These algorithms are consensus-based and consequently subject to the limitations of this category. In addition, they have computational complexities of $O(L^3)$ and $O(L^2)$, respectively, where $L$ is the system order.

This work was supported in part by a Commonwealth Scientific and Industrial Research Organisation (CSIRO) scholarship, the Academy of Finland, and the Centre of Excellence in Smart Radios and Wireless Research (SMARAD) at the Aalto University.

In this paper, we propose a diffusion-based gradient-descent TLS strategy for distributed adaptive estimation. For this purpose, we devise a network TLS cost function that is the sum of individual node cost functions. Then, employing the general framework for diffusion strategies over adaptive networks developed in [4], we minimize the network cost function by means of stochastic gradient-descent optimization in a distributed fashion over the network. Unlike the bias-compensated DLMS (BC-DLMS) strategy, the proposed diffusion strategy does not require any knowledge of the noise variances. Moreover, similar to DLMS and BC-DLMS, the computational complexity of the new strategy is $O(L)$. Simulation results show that it outperforms both DLMS and BC-DLMS.

## 2. PROPOSED STRATEGY

Let us consider a connected network with $M$ nodes, each having a linear system described by

$$\tilde{\mathbf{x}}_{k,n}^T\mathbf{h} = \tilde{y}_{k,n},\ k = 1,2,\ldots,M$$

where $\mathbf{h} \in \mathbb{R}^{L\times 1}$ is the vector of network-wide unknown and sought-after parameters, $\tilde{\mathbf{x}}_{k,n} \in \mathbb{R}^{L\times 1}$ and $\tilde{y}_{k,n} \in \mathbb{R}$ are the input vector and the output signal of the node number $k \in \{1,2,\ldots,M\}$ at time index $n \in \mathbb{N}$, respectively, and superscript $T$ denotes matrix/vector transposition. Suppose that both the input and output are observed in noise as $\mathbf{x}_{k,n} = \tilde{\mathbf{x}}_{k,n} + \boldsymbol{\upsilon}_{k,n}$ and $y_{k,n} = \tilde{y}_{k,n} + \nu_{k,n}$ where $\boldsymbol{\upsilon}_{k,n} \in \mathbb{R}^{L\times 1} \sim \mathcal{N}(\mathbf{0}_L, \varsigma_k^2\mathbf{I}_L)$ and $\nu_{k,n} \in \mathbb{R} \sim \mathcal{N}(0,\sigma_k^2)$ are the corresponding noises with $\mathbf{0}_L$ being the $L\times 1$ all-zero vector and $\mathbf{I}_L$ the $L\times L$ identity matrix. The noises are independent of each other and the input data.

If all the data is collected and processed in a fusion center, a centralized adaptive filtering process can iteratively estimate $\mathbf{h}$ exploiting the data available across the network up to the current time. The tap weights of this adaptive filter, denoted by $\mathbf{w}_n \in \mathbb{R}^{L\times 1}$, are taken as the estimate at iteration $n$. We wish to compute $\mathbf{w}_n$ such that it fits the input data to the output data by incurring minimum perturbation. This amounts to determining the minimum input data perturbation matrixes $\boldsymbol{\Delta}_n \in \mathbb{R}^{M\times L}$, the minimum output data perturbation vectors $\boldsymbol{\delta}_n \in \mathbb{R}^{M\times 1}$, and the filter weights $\mathbf{w}_n$ that satisfy

$$(\mathbf{X}_n^T + \boldsymbol{\Delta}_n)\mathbf{w}_n = \mathbf{y}_n + \boldsymbol{\delta}_n \tag{1}$$

where

$$\mathbf{X}_n = [\mathbf{x}_{1,n}, \mathbf{x}_{2,n}, \ldots, \mathbf{x}_{M,n}],$$

$$\mathbf{y}_n = [y_{1,n}, y_{2,n}, \ldots, y_{M,n}]^T.$$

We may cast this fitting problem as

$$[\boldsymbol{\Delta}_n, \boldsymbol{\delta}_n] = \arg\min_{\boldsymbol{\Delta},\boldsymbol{\delta}} \|[\boldsymbol{\Delta}, \boldsymbol{\delta}]\|_F \quad \text{subject to } \mathbf{y}_n + \boldsymbol{\delta} \in \text{Range}\{\mathbf{X}_n^T + \boldsymbol{\Delta}\} \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Once $\boldsymbol{\Delta}_n$ and $\boldsymbol{\delta}_n$ are found from (2), any $\mathbf{w}_n$ satisfying (1) is the desired solution. Using the singular value decomposition (SVD) of the augmented data matrix $[\mathbf{X}_n^T, \mathbf{y}_n]$, the total least-squares (TLS) solution is given by

$$\mathbf{w}_n = -\frac{\mathbf{v}_{n,1:L}}{v_{n,L+1}} \tag{3}$$

where $\mathbf{v}_n$ is the right singular vector corresponding to the smallest singular value of $[\mathbf{X}_n^T, \mathbf{y}_n]$ and $\mathbf{v}_{n,1:L}$ and $v_{n,L+1}$ denote the $L$ uppermost elements and the $(L+1)$th element of $\mathbf{v}_n$, respectively [29].

The solution of (3) is optimal. However, obtaining it comes at the expense of performing the SVD of a $M \times (L+1)$ matrix at each iteration. In the light of the analysis of [29], a computationally more efficient alternative approach would be to minimize the following cost function over $\mathbf{w} \in \mathbb{R}^{L\times 1}$:

$$J_{\text{net}}(\mathbf{w}) = E\left[\frac{\left\|[\mathbf{X}_n^T, \mathbf{y}_n]\begin{bmatrix}\mathbf{w}\\-1\end{bmatrix}\right\|^2}{\left\|\begin{bmatrix}\mathbf{w}\\-1\end{bmatrix}\right\|^2}\right] \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean norm and $E[\cdot]$ is the expectation operator. Observe that $J_{\text{net}}(\mathbf{w})$ is in fact the Rayleigh quotient of

$$\boldsymbol{\Psi} = E\left\{\begin{bmatrix}\mathbf{X}_n\\\mathbf{y}_n^T\end{bmatrix}[\mathbf{X}_n^T, \mathbf{y}_n]\right\}$$

with argument $[\mathbf{w}^T, -1]^T$ and reaches its minimum value $\lambda_{\min}$ (the smallest eigenvalue of $\boldsymbol{\Psi}$) when $[\mathbf{w}^T, -1]^T$ is the eigenvector corresponding to $\lambda_{\min}$ [30].

We may rewrite (4) as

$$\begin{aligned} J_{\text{net}}(\mathbf{w}) &= \frac{E[\|\mathbf{X}_n^T\mathbf{w} - \mathbf{y}_n\|^2]}{\|\mathbf{w}\|^2 + 1} \\ &= \sum_{k=1}^{M} \frac{E\left[\left(\mathbf{x}_{k,n}^T\mathbf{w} - y_{k,n}\right)^2\right]}{\|\mathbf{w}\|^2 + 1} \\ &= \sum_{k=1}^{M} J_k(\mathbf{w}) \end{aligned}$$

where $J_k(\mathbf{w})$ can be considered as the individual cost function associated with node $k$.

It is known that the critical points of the Rayleigh quotient cost function, $J_{\text{net}}(\mathbf{w})$, are the eigenvectors of $\boldsymbol{\Psi}$ and the critical values of $J_{\text{net}}(\mathbf{w})$ are the eigenvalues of $\boldsymbol{\Psi}$. Moreover, $\lambda_{\min}$ is the only stable critical value (local and global minimum) of $J_{\text{net}}(\mathbf{w})$. As a result, $\mathbf{h}$ is the unique minimizer of $J_{\text{net}}(\mathbf{w})$ and the global minimum of $J_{\text{net}}(\mathbf{w})$ can be reached using the gradient-descent method from any initial point given the choice of an appropriate step-size [31]-[33].

The gradient of the individual cost function for node $k$ is calculated as

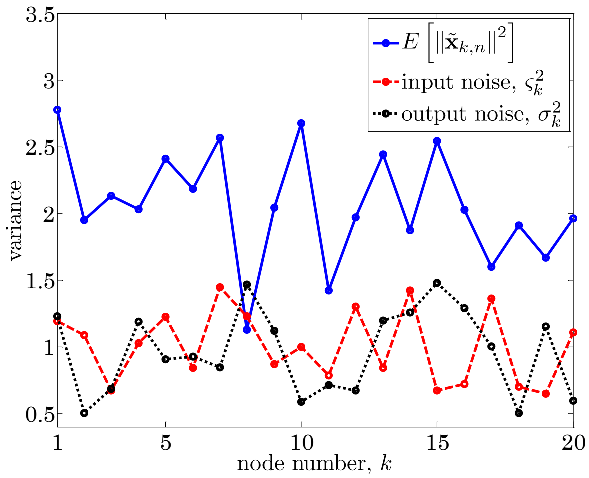


Fig. 1. The expected squared norm of the input data vectors together with the variances of the input and output noises at each node.

$$\begin{aligned}\mathbf{g}_k(\mathbf{w}) &= \frac{\partial J_k(\mathbf{w})}{\partial \mathbf{w}^T}\\ &= \frac{-2E\left[\mathbf{x}_{k,n}\left(y_{k,n}-\mathbf{x}_{k,n}^T\mathbf{w}\right)\right](\|\mathbf{w}\|^2+1)-2\mathbf{w}E\left[\left(y_{k,n}-\mathbf{x}_{k,n}^T\mathbf{w}\right)^2\right]}{(\|\mathbf{w}\|^2+1)^2}\\ &\approx \frac{-2\mathbf{x}_{k,n}\left(y_{k,n}-\mathbf{x}_{k,n}^T\mathbf{w}\right)(\|\mathbf{w}\|^2+1)-2\mathbf{w}\left(y_{k,n}-\mathbf{x}_{k,n}^T\mathbf{w}\right)^2}{(\|\mathbf{w}\|^2+1)^2}.\end{aligned}$$

Subsequently, a centralized gradient-descent total least-squares (GDTLS) estimate of the network parameters can be iteratively achieved as

$$\begin{aligned}\mathbf{w}_n &= \mathbf{w}_{n-1} - \tfrac{\mu}{2M}\textstyle\sum_{k=1}^{M}\mathbf{g}_k(\mathbf{w}_{n-1})\\ &= \mathbf{w}_{n-1} + \tfrac{\mu}{M}\textstyle\sum_{k=1}^{M}\epsilon_{k,n}\left(\mathbf{x}_{k,n}+\epsilon_{k,n}\mathbf{w}_{n-1}\right)\end{aligned}$$

where

$$\epsilon_{k,n} = \frac{y_{k,n}-\mathbf{x}_{k,n}^T\mathbf{w}_{n-1}}{\|\mathbf{w}_{n-1}\|^2+1}$$

and $\mu > 0$ is a step-size parameter.

As for the in-network processing contrasted with the centralized processing, the nodes cooperate with each other to perform a distributed estimation over the network and lay off the fusion center. In this way, each node makes its own estimation of the unknown parameters, namely $\mathbf{w}_{k,n}$, and shares its data with its neighbours. Using the general framework for diffusion-based distributed adaptive optimization, introduced in [4], an adapt-then-combine (ATC) diffusion strategy utilizing the gradient-descent TLS approach can be formulated as

$$\begin{aligned}\mathbf{z}_{k,n} &= \mathbf{w}_{k,n-1} - \tfrac{\mu_k}{2}\textstyle\sum_{l\in\mathcal{N}_k} c_{l,k}\mathbf{g}_l(\mathbf{w}_{k,n-1})\\ &= \mathbf{w}_{k,n-1} + \mu_k\textstyle\sum_{l\in\mathcal{N}_k} c_{l,k}\epsilon_{l,k,n}\left(\mathbf{x}_{l,n}+\epsilon_{l,k,n}\mathbf{w}_{k,n-1}\right)\\ \mathbf{w}_{k,n} &= \textstyle\sum_{l\in\mathcal{N}_k} a_{l,k}\mathbf{z}_{l,n}\end{aligned}$$

where

$$\epsilon_{l,k,n} = \frac{y_{l,n}-\mathbf{x}_{l,n}^T\mathbf{w}_{k,n-1}}{\left\|\mathbf{w}_{k,n-1}\right\|^2+1},$$

$\mu_k > 0$ are the step-size parameters, and $c_{l,k} > 0$ and $a_{l,k} > 0$ are the combination parameters that satisfy

$$\begin{aligned}&c_{l,k} = a_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k,\ k = 1,2,\ldots,M,\\ &\textstyle\sum_{l=1}^{M} c_{l,k} = 1,\ k = 1,2,\ldots,M,\\ &\textstyle\sum_{k=1}^{M} c_{l,k} = 1,\ l = 1,2,\ldots,M,\end{aligned}$$

and

$$\textstyle\sum_{l=1}^{M} a_{l,k} = 1,\ k = 1,2,\ldots,M.$$

The set $\mathcal{N}_k$ contains $k$ and indexes of the nodes that are directly connected to node $k$ and can exchange information with it.

The above algorithm has two update equations for adaptation and combination of the estimates. At each iteration, all the nodes update their estimates using the data available within their neighborhoods and store them in the intermediate variables, $\mathbf{z}_{k,n}$. Then, they combine the intermediate estimates available within their neighborhoods to obtain the new estimates, $\mathbf{w}_{k,n}$. Obviously, each node shares its input-output data, i.e., $\mathbf{x}_{k,n}$ and $y_{k,n}$, as well as its intermediate estimate, $\mathbf{z}_{k,n}$, with its neighbors. By swapping the order of the adaptation and combination steps, a combine-then-adapt (CTA) version can also be expressed as

$$\begin{aligned}\mathbf{z}_{k,n-1} &= \textstyle\sum_{l\in\mathcal{N}_k} a_{l,k}\mathbf{w}_{l,n-1}\\ \mathbf{w}_{k,n} &= \mathbf{z}_{k,n-1} + \mu_k\textstyle\sum_{l\in\mathcal{N}_k} c_{l,k}\epsilon_{l,k,n}\left(\mathbf{x}_{l,n}+\epsilon_{l,k,n}\mathbf{z}_{k,n-1}\right).\end{aligned}$$

We call the proposed strategy *diffusion gradient-descent total least squares* (D-GDTLS).

## 3. SIMULATIONS

We consider a distributed errors-in-variables system identification problem where the unknown system is an arbitrary finite-impulse-response filter of order $L = 2$ and unit energy. The identification task is carried out by a connected network consisting of $M = 20$ nodes where each node is linked to three other nodes in average. The input data vectors of the nodes are zero-mean Gaussian and independent in time and space. The misalignment, as the estimation performance metric, is defined as $E[\|\mathbf{h}-\mathbf{w}_n\|^2]$ and evaluated by ensemble-averaging over $10^4$ independent runs. The steady-state misalignment is obtained by averaging over 200 steady-state values. The expected squared norm of the input data vectors as well as the variance of the input and output noises at each node are shown in Fig. 1. We use metropolis weights for the adaptation phase and simple averaging weights for the combination phase [11]. We also initialize the estimates of all the algorithms to all-zero vectors and tune to step-sizes

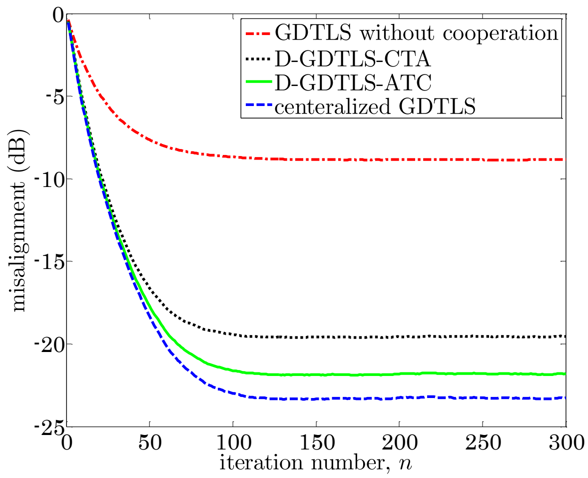


Fig. 2. Learning curves of different implementations of the proposed strategy averaged over all the nodes.

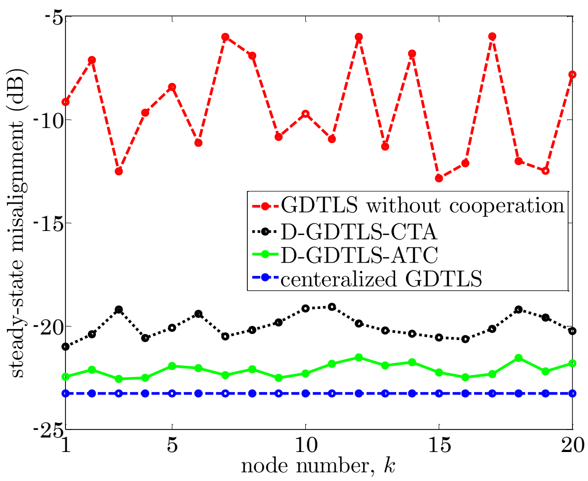


Fig. 3. Steady-state misalignments of different implementations of the proposed strategy at each node.

such that all the algorithms have almost the same initial convergence rate.

In Figs. 2 and 3, we compare the performance of the ATC and CTA implementations of the proposed strategy with the centralized and non-cooperative implementations of the GDTLS algorithm. In the non-cooperative case, each node updates its estimate using the GDTLS algorithm without exchanging any information with its neighbors. In Figs. 4 and 5, we compare the performance of the ATC implementations of the D-GDTLS, DLMS, and BC-DLMS strategies. In Figs. 2 and 4, we plot the misalignments averaged over all the nodes against the iteration number and in Figs. 3 and 5, we plot the node-specific steady-state misalignments versus the node number.

The simulation results show that the performance of the proposed strategy is superior to those of its contenders, the DLMS and BC-DLMS strategies. Moreover, its ATC implementation outperforms the CTA implementation, which is consistent with the previous predictions and observations [1], [4], [11].

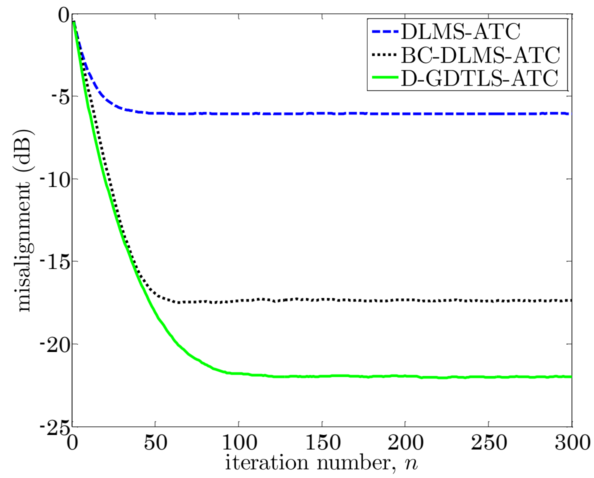


Fig. 4. Learning curves of different strategies averaged over all the nodes.

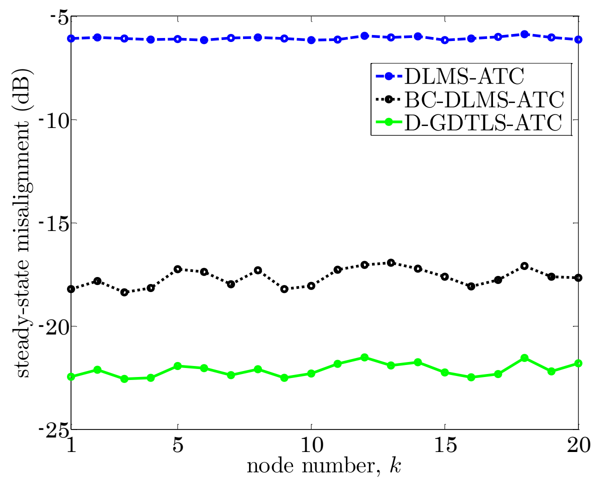


Fig. 5. Steady-state misalignments of different strategies at each node.

## 4. CONCLUSION

We have developed a new diffusion-based distributed adaptive filtering and estimation strategy by defining a total least-squares network cost function that is the sum of individual node-specific cost functions and minimizing it over the network and in a cooperative manner via the gradient-descent method. The proposed strategy has the same order of complexity as the diffusion least mean square (DLMS) and bias-compensated DLMS (BC-DLMS) strategies and, as attested to by the simulation results, outperforms these strategies when both the input and output data are observed in noise. Unlike the BC-DLMS strategy, the new strategy does not require any prior knowledge of the noise variances.

## REFERENCES

[1] A. Sayed, "Diffusion adaptation over networks," to appear in *E-Reference Signal Processing*, R. Chellapa and S.

Theodoridis, Eds., Elsevier, 2013, available at http://arxiv.org/abs/1205.4220.

[2] S.-Y. Tu and A. H. Sayed, “Mobile adaptive networks,” *IEEE J. Sel. Topics. Signal Process.*, vol. 5, no. 4, pp. 649–664, Aug. 2011.

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

[4] J. Chen and A. H. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[5] R. Abdolee, B. Champagne, and A. H. Sayed, “A diffusion LMS strategy for parameter estimation in noisy regressor applications,” in *Proc. European Signal Process. Conf.*, Bucharest, Romania, Aug. 2012, pp. 749-753.

[6] X. Zhao, S.-Y. Tu, and A. H. Sayed, “Diffusion adaptation over networks under imperfect information exchange and non-stationary data,” *IEEE Trans. on Signal Process.*, vol. 60, no. 7, pp. 3460-3475, Jul. 2012.

[7] A. Bertrand, M. Moonen, and A. H. Sayed, “Diffusion bias-compensated RLS estimation over adaptive networks,” *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5212–5224, Nov. 2011.

[8] 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.

[9] N. Takahashi, I. Yamada, and A. H. Sayed, “Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis,” *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, Sep. 2010.

[10] F. S. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

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

[12] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[13] C. G. Lopes and A. H. Sayed, “Incremental adaptive strategies over distributed networks,” *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[14] M. G. Rabbat and R. D. Nowak, “Quantized incremental algorithms for distributed optimization,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[15] A. Nedic and D. P. Bertsekas, “Incremental subgradient methods for nondifferentiable optimization,” *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.

[16] D. P. Bertsekas, “A new class of incremental gradient methods for least squares problems,” *SIAM J. Optim.*, vol. 7, no. 4, pp. 913–926, 1997.

[17] 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.

[18] A. Bertrand and M. Moonen, “Low-complexity distributed total least squares estimation in ad hoc sensor networks,” *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4321–4333, Aug. 2012.

[19] 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.

[20] 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, 2009.

[21] 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.

[22] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Performance analysis of the consensus-based distributed LMS algorithm,” *EURASIP J. Advances Signal Process.*, vol. 2009, article ID 981030, 19 pages, 2009.

[23] 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, 2009.

[24] I.D. Schizas, A. Ribeiro, and G. B. Giannakis, “Consensus in ad hoc WSNs with noisy links—part I: distributed estimation of deterministic signals,” *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, 2008.

[25] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, “Consensus in ad hoc WSNs with noisy links—part II: distributed estimation and smoothing of random signals,” *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1650–1666, 2008.

[26] S.-Y. Tu and A. H. Sayed, “Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks,” *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[27] I. Markovsky and S. Van Huffel, “Overview of total least-squares methods,” *Signal Processing*, vol. 87, pp. 2283-2302, 2007.

[28] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, Philadelphia, PA: SIAM, 1991.

[29] G. H. Golub and C. F. Van Loan, “An analysis of the total least squares problem,” *SIAM J. Numer. Anal.*, vol. 17, no. 6, pp. 883–893, Dec. 1980.

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

[31] G. Cirrincione and M. Cirrincione, *Neural-Based Orthogonal Data Fitting: The EXIN Neural Networks*, Hoboken, NJ: Wiley, 2010.

[32] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2008.

[33] B. E. Dunne and G. A. Williamson, “Analysis of gradient algorithms for TLS-based adaptive IIR filters,” *IEEE Trans. Signal Process.*, vol. 52, pp. 3345–3356, Dec. 2004.