Traffic Influence in Distributed Networks Using Cooperation and Set-Membership Filtering

Marco Fernandes S. Xaud,[1] Diego A. Zanon,[2] Ana Luiza Dallora[3]
[1]Automation and Control Engineering Department
[2]Electronics and Computer Engineering Department
[3]Information and Computing Engineering Department
Poli - Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
marco.fernandes, diegozanon, ana_dallora@poli.ufrj.br

Marcello L. R. de Campos
Electrical Engineering Program
COPPE - Federal University of Rio de Janeiro
P.O. Box 68504, Rio de Janeiro, Brazil
mcampos@ieee.org

***Abstract*— This paper investigates the influence of data loss in the performance of cooperative adaptive filters in distributed networks. The algorithms analyzed are those with and without data selection based on innovation. Our simulation results indicate that set-membership adaptation algorithms, which perform some form of innovation check prior to transmission of data to the neighbors, have better performance than their counterparts which flood the network with data at every iteration. Therefore the space-time data selection of set-membership adaptive filters reduces computational complexity and energy consumption, and also improves convergence performance in case of data loss during transmission.**

Keywords*— *sensor networks, adaptive filters, cooperation, set-membership adaptive filtering

I. Introduction

The use of sensors is very common in tracking, monitoring and control applications. The disposition of a number of these devices together with the connections that enable them to diffuse data and parameters define a sensor network. The *ad hoc* mesh wireless technology introduces new perspectives to sensor network applications since they can auto-configure themselves in situations without pre-existent communication infrastructure, enabling more data portability in a larger range of places demanding lower resources. Another benefit is the independence of a central device for routing and diffusing the information gathered, so the nodes can communicate with each other directly, enabling the estimation of parameters to be done in a decentralized way.

In a centralized parameter estimation scenario, composed of a great number of sensors, data processing can be very slow (which is not suitable, for sensors are usually powered by batteries), or even prohibitive, as the convergence of a considerable volume of data on a single device might demand high performance hardware and high cost. Moreover, the geographical location of the central node with respect to the remote sensors is to be taken into consideration, for long distance communications imply high energy consumption and also high cost. An additional disadvantage of this approach is the considerable raise of data traffic in the network, that could bring congestion problems leading to information loss. A possible solution for this particular problem is to employ mechanisms of selective cooperation. A related work covered concepts of selective algorithms [1]. However, the advantages of these algorithms, when compared with the ones without data selection in a scenario inclined to data loss, were overlooked.

This paper proposes an analysis of the speed of convergence of different approaches taken by cooperative algorithms in the particular situation mentioned. For this purpose, a simple traffic model was implemented and simulated along with the algorithms.

This work was partially supported by CNPq under contracts 306445/2007-7, 503706/2007-8, and 478282/2007-9.

II. Distributed Adaptive Filtering with Selective Cooperation

An adaptive filter changes its parameters according to an optimization algorithm in order to approximate some desired behavior by means of minimizing a established criterion. In centralized networks, parameter estimation may require high processing capacity in the central unit node, not necessarily close to the sensors, which may demand long-distance, expensive, and low-efficiency communication. On the other hand, estimation performed in decentralized ways can generate high traffic of data between nodes, especially if there is cooperation among them.

A. General Distributed Adaptive Filter Operation

In a spatially distributed network [2]– [7], we assume that each node has a sensor with the ability to measure a data pair $\{\mathbf{x}_m(k), d_m(k)\}$, where m denotes the node index, and $\mathbf{x}_m(k) \in \mathbb{R}^N$ and $d_m(k) \in \mathbb{R}$ are input and desired output signals of some unknown system, respectively.

If each node executes its own adaptation algorithm without exchanging information with any neighbor, we can say that the nodes do not "cooperate" among themselves. On the other hand, if cooperation exists, nodes can use information from their neighbors, like data pairs and local estimates.

In that context, we define the neighborhood $\mathcal{N}_m$ of node m as the set of nodes directly connected to it, i.e., they are one *hop* away in the network, including itself (see Fig.1). In a cooperation algorithm, there are basically four main steps in each iteration [1]: *transmission* (data pair spreading), *estimation* (production of a local estimate *update*), *diffusion* (exchange of local estimates), and *consensus* (production of a final estimate for the iteration). After data pair spreading, the second step for our reference node m is to use the data spread by its neighbors in some function f in order to generate a *local estimate*. The node can also use its previous local information about the parameters, that is, function f performs a space-time update

$$\phi_m(k) = f[\mathbf{w}_m(k-1),\ \mathbf{x}_l(k),\ d_l(k);\ l \in \mathcal{N}_m] \quad (1)$$

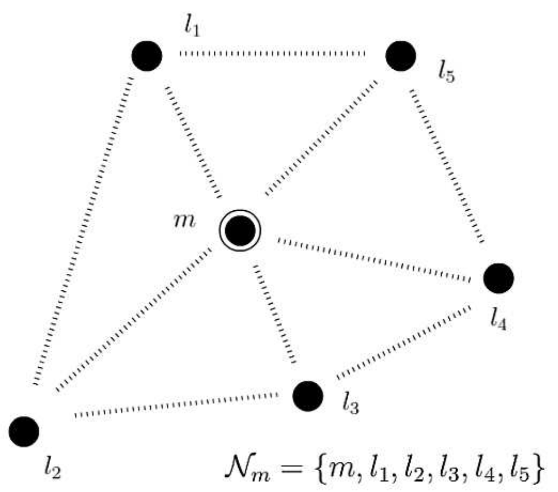

Fig. 1. Ad-Hoc network and neighborhood $\mathcal{N}_m$ of m.

where $\phi_m(k)$ is the computed local estimate, $\mathbf{w}_m(k-1)$ is the consensus calculated in the previous iteration, and $l \in \mathcal{N}_m$ refers to each node in the neighborhood of m. After the diffusion of all estimates, the following step is the consensus, i.e., the computation of vector $\mathbf{w}_m(k)$ based on a defined criterion about how to use all the exchanged local estimates. Thus there is a function g, evaluated at node m, which uses all the exchanged information with the neighbors, i.e.,

$$\mathbf{w}_m(k) = g[\phi_l(k);\; l \in \mathcal{N}_m] \tag{2}$$

where $\mathbf{w}_m(k)$ is the consensus estimate at iteration k.

B. Set-Membership Concept

It is usual in cooperative adaptive filtering that all the nodes spread their information. Before estimation takes place, there is *feedforward traffic* during data pair transmission, whereas after diffusion takes place, nodes exchange their estimates with their neighbors, which we call here *feedback traffic*. The concept of Set-Membership (SM) applied to time as well to spatial updating yields substantial savings in computation per node and in traffic among nodes [1].

SM adaptation algorithms use an upper bound γ for the output error to access *innovation* brought by new data. Innovation is an important concept, which SM algorithms use to indicate if updates are needed. The strategy has been applied to adaptive filters for selective updating in time, whereby data brought at every iteration is checked for innovation. In [1], this strategy was extended for selective updating in space, whereby data to be sent to neighbors is checked for innovation. If there is not enough innovation in the data, they do not need to be shared with neighbors. Therefore the evaluation of a local estimate and the transmission of data pairs or estimates are not performed if deemed unnecessary. As a consequence, there is a drastic reduction of traffic in the network. In this work, we use different implementations of NLMS (Normalized Least-Mean Squares) algorithms with Set-Membership and evaluate their behavior when network traffic can cause package loss.

C. Diffusion SM-NLMS

This is the most simple variation of the SM-NLMS algorithm used in this work. The algorithm performs spatial and time innovation at node m upon reception of data pairs and local estimates from the neighbors.

Let us define the constraint set $\mathscr{H}_l(k)$ as

$$\mathscr{H}_l(k) = \{\phi \in \mathbb{R}^N : |d_l(k) - \phi^{\mathrm{T}}\mathbf{x}_l(k)| \leq \gamma\}. \tag{3}$$

In the NLMS algorithm, no data selection is used, and all the nodes in the neighborhood of m have to share data pairs, which are sequentially used to perform update in the current estimate $\phi_m(k)$. In the SM-NLMS algorithm, data selection means that a data pair $\{d_l(k),\ \mathbf{x}_l(k)\}$ is considered to bring innovation, therefore used in the local estimate, if the available local estimate $\phi_m(k)$ does not belong to the constraint set associated with $\{d_l(k),\ \mathbf{x}_l(k)\}$. The differential strategy of the Diffusion SM-NLMS algorithm when compared to the NLMS algorithm is to verify this condition for every data pair received from neighbors. Every node in the neighborhood of m has to share its data pair with it, but node m attempts to discard data from those for which $\phi_m(k) \in \mathscr{H}_l(k), l \in \mathcal{N}_m(k)$. With this data selection, the algorithm checks, at every time, innovation provided by all data pairs available. The result of this method is the reduction of computational cost at each node, since the amount of information in estimation is reduced. We can imagine a situation where none of the collected data pairs implies innovation, hence no update is performed and diffusion is not needed. In that specific case, there is also a reduction in feedback traffic.

D. SM-NLMS (NFF) - Non-Feedforward Traffic

This is an alternative to the previous algorithm, whereby nodes exchange only their local estimates $\phi_l(k)$, but not data pairs [1]. Because of this restriction, there is a total reduction of feedforward traffic at the expense of less spatial diversity information used.

E. SM-NLMS (SIC) - Transmission with Spatial Innovation Check

The two previous algorithms propose different ways of communicating data pairs. The first one suggests innovation selection upon reception of data pairs, whereas the second one provides feedforward traffic reduction by not transmitting data pairs. An alternative idea is to implement a spatial innovation check before data pair transmission, i.e., each node $l \in \mathcal{N}_m$ decides whether or not its data pair will benefit its neighbors. A *spatial innovation set* $\mathcal{N}_m'(k)$ is defined as [1]

$$\mathcal{N}_m'(k) = \{l \in \mathcal{N}_m : \phi_l(k-1) \notin \mathscr{H}_l(k)\} \tag{4}$$

where $\mathcal{N}_m'(k)$ is the restricted neighborhood of node m, that is, $\mathcal{N}_m'(k)$ contains only the nodes that have broadcast data pairs. Naturally, update is performed using only data pairs from nodes within this set. The estimation is computed as follows [1]:

```
At each node $m$:
    $\phi_m(k) = \mathbf{w}_m(k-1)$
    For each $l \in \mathcal{N}_m'(k)$:
        $e_l(k) = d_l(k) - \phi_m^{\mathrm{T}}(k)\mathbf{x}_l(k)$
        $\phi_m(k) \leftarrow \phi_m(k) + \frac{\alpha_l(k)e_l(k)}{\|\mathbf{x}_l(k)\|^2}\mathbf{x}_l(k).$
```

where $\alpha_l(k)$ is called the data-dependent step size, i.e.,

$$\alpha_l(k) = \begin{cases} 1-\gamma/|e_l(k)| & \text{if } |e_l(k)| > \gamma \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Note that the loop involving each node l is restricted to the spatial innovation set $\mathcal{N}_m'(k)$, which can change in subsequent iterations. The parameter $\alpha_l(k)$ allows node m to check if the received data pair from each neighbor actually brings innovation if compared to the current estimate $\phi_m(k)$. Thus, if $\alpha_l(k)$ is equal to zero, it means no innovation. For the particular case where all parameters $\alpha_l(k)$ from every l are equal to zero, no data pair is used, hence no update is performed and the diffusion of $\phi_m(k)$ is unnecessary.

F. SM-NLMS (SIC-RFB) - Transmission with Spatial Innovation Check and Reduced Feedback Traffic

As mentioned before, data pairs can yield updates or not, according to the values of $\alpha_l(k)$. The reference node does not diffuse its estimate only if all $\alpha_l(k), l \in \mathcal{N}_m'(k)$ are zero. However, the strategy of SM-NLMS SIC-RFB is to introduce a *local innovation check*. A local estimate will not be performed if the local data pair does not bring innovation, regardless of possible innovation brought by the neighbors. If no innovation is brought by $\{d_m(k), \mathbf{x}_m(k)\}$, then this node does not even compute an update in $\phi_m(k)$, so that both feedback traffic and computational cost are reduced. The algorithm is [1]:

```
$e_m'(k) = d_m(k) - \mathbf{w}_m^{\mathrm{T}}(k-1)\mathbf{x}_m(k)$
If $|e_m'(k)| > \gamma$
    $\phi_m(k) = \mathbf{w}_m(k-1)$
    For each $l \in \mathcal{N}_m'(k)$
        $e_l(k) = d_l(k) - \phi_m^{\mathrm{T}}(k)\mathbf{x}_l(k)$
        $\phi_m(k) \leftarrow \phi_m(k) + \frac{\alpha_l(k)e_l(k)}{\|\mathbf{x}_l(k)\|^2}\mathbf{x}_l(k).$
```

where $\alpha_l(k)$ is calculated according to Equation (5).

G. Diffusion and Consensus

After the local estimate evaluation, the next step for all the nodes at the present iteration is the diffusion of their local updates $\phi_l(k)$. In other words, in the diffusion phase, every estimate which offers update with respect to the past has to be communicated to all the neighbors. After diffusion, a consensus is necessary.

There are many different methods for consensus. In [1] an algorithm of sphere combinations is proposed, in which all the estimates are considered to be centers of spheres with a previously calculated radius. In a pair-wise sequential update, the new consensus update is the center of the sphere that tightly outer bounds the intersection of two of these spheres. In contrast, a simple method is the weighted average, in which the consensus is calculated as

$$\mathbf{w}_m(k) = \sum_{l \in \mathcal{N}_m} a_l(k)\phi_l(k) \tag{6}$$

where $a_l(k)$ is the weight factor for each local estimate $\phi_l(k)$. If all of them are equally important, we can use an arithmetic average, where all the $a_l(k)$ have the same value. The focus of this work is to analyze the influence of package loss due to traffic in the performance of the algorithms, not to compare different consensus strategies. We chose to use the strategy given in Equation (6) in all simulations.

III. Traffic Modelling and Erlang B Distribution

When using the decentralized parameter estimation approach instead of the centralized one, the volume of data flowing in the network raises a discussion about the relevance of traffic issues in the convergence process of the previously referred algorithms. Data loss is likely in situations where the amount of information to be exchanged exceeds the capacity of the channels or devices involved. As adaptive filters are very sensitive to their input signals, package loss may have a deleterious influence in performance. In order to simulate this scenario, network traffic was modeled according to the *Erlang B* model (or *Erlang's Loss Model*) which was chosen due to its simplicity and lack of queuing support. In this particular model, blocked connections, which might have been consequence of congestion, cause data to be discarded and information to be lost. There is also the assumption that data arrive randomly [5], are independent from each other forming a Poisson Process, and are treated in the incoming order. The equation of Erlang B provides us with the probability P of blocked connections, given a number N of available channels among nodes, and the traffic T in Erlang units:

$$P = \frac{\frac{T^N}{N!}}{\sum_{i=0}^{N} \frac{T^i}{i!}} \tag{7}$$

where $T = C_a C_h$, C_a is the data arrival rate, and C_h is the average connection holding time.

IV. Simulations

In this section, we display the results of experiments where the algorithms are tested when there is inter-node traffic, and consequently data loss. Their behavior is compared with the ideal situation when there is no traffic.

The network topology is the same for all experiments, with $M = 10$ nodes, all of them connected. Therefore the neighborhood sets are all equal and include all nodes. The unknown system to be identified has 10 coefficients, randomly generated. The input signal and additive noise at each node are white Gaussian signals with zero mean and variance calculated such that the SNR was 30 dB. For the MSE calculation, we averaged 1000 independent runs.

In order to highlight the effects of traffic, we assumed a very high blocking probability for the full traffic situation, i.e., when the algorithm generates either feedforward or feedback traffic in every iteration. We used the traffic model described in Section III with each node being served by 9 independent connections and receiving a total traffic of 80 Erlangs. This gives a blocking probability for a full traffic situation approximately equal to 89%.

Figure 2 shows the simulation results for the NLMS algorithm with $\mu = 0.2$. We can clearly see that data loss causes the algorithm to slow down considerably. When traffic is taken into account, the algorithm behaves as if nodes did not cooperate and had to act upon the local information only.

Traffic generated by set-membership algorithms is expected to decrease as the algorithm approaches convergence. Therefore in order to simulate traffic and package loss for these algorithms, we divided the learning curves into five regions and calculated how many data pair transmissions and how many parameter diffusions were done, in average, per region. This gave us a relative measure of generated traffic in comparison with an algorithm that always shares data pairs and always diffuses parameter estimates (total

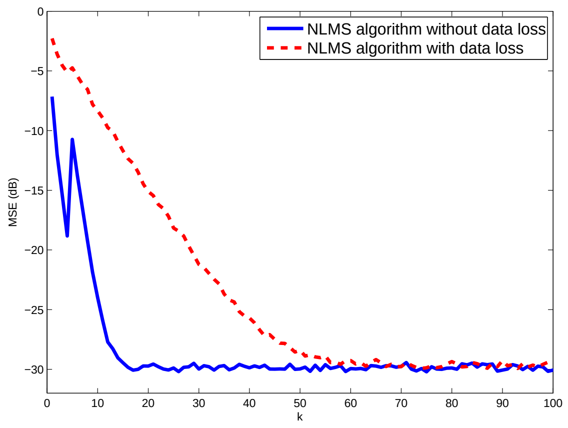

Fig. 2. Effect of high traffic and high probability of data loss in the behavior of the NLMS algorithm.

TABLE I
GENERATED TRAFFIC DUE TO DATA PAIR TRANSMISSION AND PARAMETER DIFFUSION (PART I)

	Diffusion SM-NLMS Algorithm		SM-NLMS-NFF Algorithm	
Region	Feedforward	Feedback	Feedforward	Feedback
1 (k=1 to 10)	100%	89.1%	0%	65%
2 (k=11 to 20)	100%	41.5%	0%	33.7%
3 (k=21 to 30)	100%	24.3%	0%	12.8%
4 (k=31 to 40)	100%	22.0%	0%	6.7%
5 (k=41 to 150)	100%	21.8%	0%	3.6%

traffic situation). For example, the Diffusion SM-NLMS algorithm described in Section II-C generates 100% of feedforward traffic, but a decreasing amount of feedback traffic, because as it approaches convergence, parameter updates, and consequently diffusion, become less frequent. On the other hand, the SM-NLMS-NFF described in Section II-D generates zero feedforward traffic. Tables I and II show the generated traffic due to data pair transmission and parameter diffusion (feedforward and feedback traffic, respectively) for the five regions considered for the four SM algorithms discussed in Section II.

The algorithms with selective cooperation generate less traffic and are expected to face a smaller blocking probability, and consequently less data loss. Based on the average traffic generated for each region, we calculated the corresponding blocking probability faced by the Diffusion SM-NLMS, the SM-NLMS-NFF, the SM-NLMS-SIC, and the SM-NLMS-SIC-RFB algorithms. For all of them we used $\gamma = \sqrt{5\sigma^2}$, where σ^2 is the variance of the additive noise.

Figures 3, 4, 5, and 6 show the comparisons for the Diffusion SM-NLMS, the SM-NLMS-NFF, the SM-NLMS-SIC, and the SM-NLMS-SIC-RFB algorithms with and without data loss due to traffic. Notice that the blocking probabilities faced by each of these algorithms are different, for they depend on the generated

TABLE II
GENERATED TRAFFIC DUE TO DATA PAIR TRANSMISSION AND PARAMETER DIFFUSION (PART II)

	SM-NLMS-SIC Algorithm		SM-NLMS-SIC-RFB Algorithm	
Region	Feedforward	Feedback	Feedforward	Feedback
1 (k=1 to 10)	30.3%	78.7%	40.7%	40.7%
2 (k=11 to 20)	10.1%	30.2%	19.7%	19.7%
3 (k=21 to 30)	6.8%	16.7%	13.8%	13.8%
4 (k=31 to 40)	5.5%	12.2%	11.0%	11.0%
5 (k=41 to 150)	3.6%	6.0%	6.9%	6.9%

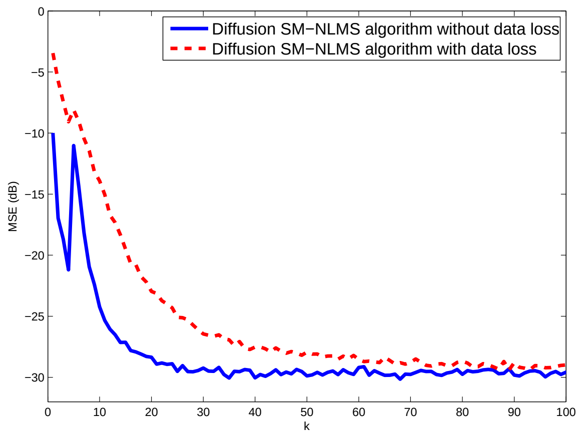

Fig. 3. Effect of high traffic and high probability of data loss in the behavior of the Diffusion SM-NLMS algorithm.

traffic. Notice also that SM algorithms suffer less than the NLMS algorithm from data loss, in addition to present faster convergence and less complexity.

V. CONCLUSIONS

Distributed parameter estimation with selective cooperation is known to offer reduced computational complexity and reduced energy consumption per node. In set-membership algorithms, this often prevents unnecessary transmission of data, either local measurements or local estimates. As a consequence, such algorithms also cause less traffic in the network, less congestion and collisions, and ultimately less innovative data to be lost. This paper investigated the impact of traffic on the performance of adaptive filters in distributed parameter estimation applications. Using a simple traffic model to take into account blocking probability, and consequent data loss, we verified that data loss may cause severe degradation of performance in cooperative adaptive filters if no data selection is used. On the other hand, in such scenarios the advantages of set-membership for data selection in space and in time may be even more pronounced.

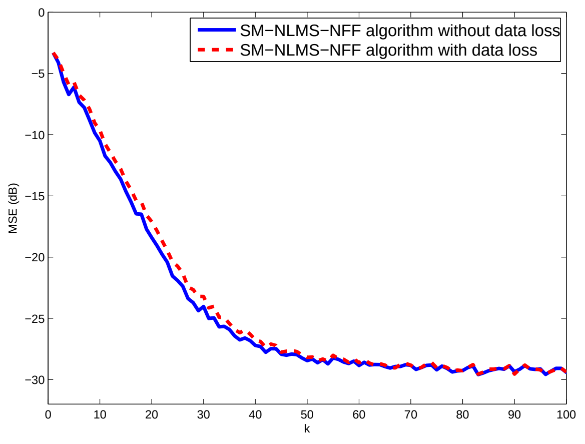

Fig. 4. Effect of high traffic and high probability of data loss in the behavior of the SM-NLMS-NFF algorithm.

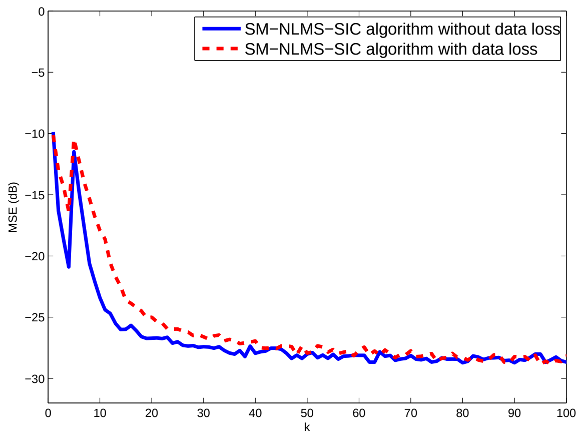

Fig. 5. Effect of high traffic and high probability of data loss in the behavior of the SM-NLMS-SIC algorithm.

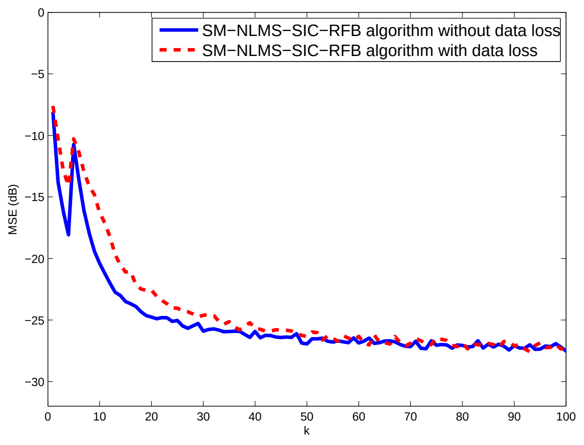

Fig. 6. Effect of high traffic and high probability of data loss in the behavior of the SM-NLMS-SIC-RFB algorithm.

REFERENCES

[1] S. Werner, Y.-F. Huang, M. L. R. de Campos, and V. Koivunen, "Distributed parameter estimation with selective cooperation," in *International Conference on Acoustics, Speech, and Signal Processing*. Taipei: IEEE, April 2009.

[2] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Third International Symposium on Information Processing in Sensor Networks*. IEEE, April 2004, pp. 20–27.

[3] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, September 2004.

[4] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Fourth International Symposium on Information Processing in Sensor Networks*. IEEE, 2005, pp. 63–70.

[5] ——, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proceedings of the 5th international conference on Information processing in sensor networks*, ACM. Nashville, USA: ACM, 2006, pp. 168–176.

[6] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3. IEEE, 2007, pp. III–917–III–920.

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