OPTIMALITY OF EXPECTATION PROPAGATION BASED DISTRIBUTED ESTIMATION FOR WIRELESS SENSOR NETWORK INITIALIZATION

John MacLaren Walsh, S. Ramanan[*]

Drexel University
Dept. of Elec. & Comp. Eng.
Philadelphia, PA

Phillip A. Regalia[†]

Catholic University of America
Dept. of Elec. Eng & Comp. Sci.
Washington, DC

ABSTRACT

We establish that expectation propagation (EP), under some mild requirements and when properly organized, provides sensors with optimal Bayes estimators during the initialization phase of a large randomly deployed wireless sensor network, regardless of the cost function chosen. We are considering the initialization phase to be the period during which the sensors do not yet know their locations and channel/interference strengths, and thus must use random sleep schedules until they have estimated them. During this initialization phase, any other scheme for distributed Bayesian estimation utilizing communication among the same nodes must have equal or worse performance to EP. We discuss the sub-optimality of some other proposed schemes for distributed estimation in sensor networks: consensus propagation and distributed adaptive filtering, arguing that these techniques may presently be seen as seeking suboptimal performance among particular cost functions and with a goal of reduced computation and complexity relative to EP.

1. MODEL FOR INITIALIZATION PHASE OF A WIRELESS SENSOR NETWORK

Once the nodes in a wireless sensor network have information regarding their relative positions and the interference/channel strengths in the communications channels between them, in order to make efficient use of energy they should opportunistically communicate with each other and utilize non-random sleep strategies. However, many models for wireless sensor network deployment, such as dropping them from a plane flying over the field of interest, involve effectively placing them randomly. Since the sensor nodes are to be cheap, need to last a long time on their battery power supplies, and will have limited computational ability, it is likely that they will not have GPS based positioning abilities, and furthermore, it is even more likely that, immediately after deployment, they will not know the communications channel qualities between them. Thus, during the initialization phase immediately after deployment, the sensor network will need to estimate these quantities. During this time, the sensor network will still need to employ duty cycling to decrease power consumption, and thus it is likely that a random sleep strategy, in which a randomly selected subset of nodes is awake at any given time, will be used.

Continuing with more specificity, assume that the sensor network is composed of N nodes $n_1, \ldots, n_N$. We model this random sleep strategy as in [1] by slotting time k corresponding to different instants in the sleep cycle, and by defining by a collection of sets $\{\mathcal{A}(k) | k \in \{1, \ldots, K\}\}$, where $\mathcal{A}(k)$ includes the indices of those sensor nodes. We assume that the random sleep strategy is selected according to a pseudo-random number generator, and repeats after K time instants, so that $\mathcal{A}(k) = \mathcal{A}(k + K)$. We further assume regular sleep strategies for clarity, in which each node is awake exactly c time instants out of the K time instants in a sleep cycle, and that $|\mathcal{A}(k)| = d$ so that a constant number of nodes are awake for each time instant, although there is no great difficulty in generalizing our results to irregular sleep strategies as in [2]. Note that these restrictions imply that $K = \frac{c}{d} N$. We can associate with the sleep cycle a bipartite graph, as shown in Figure 1, in which left nodes correspond to different sensors, and right nodes correspond to different time instants k, and an edge connects a left node n_i with a right node k if $i \in \mathcal{A}(k)$. When we say that the sleep strategies are selected randomly, we mean that the pseudo-random number generator is chosen so that the so-constructed bipartite graph is drawn uniformly from the set of such (fixed left and right degree) bipartite graphs.

Since the initialization phase has been defined as the time at which the channel qualities are not yet known at the sensor nodes, let us assume that each node in the wireless sensor network would like to know the channel gain $h_{i,j}$ between nodes n_i and n_j. We consider a flat fading model here for simplicity, but there is nothing difficult in extending the ideas to the frequency selective case. The fact that the sensor nodes were randomly placed by sampling their positions

[*] J. M. Walsh and S. Ramanan were supported by the National Science Foundation under grant CCF-0728496.

[†] P. A. Regalia was supported by the National Science Foundation under grant CCF 0728521.

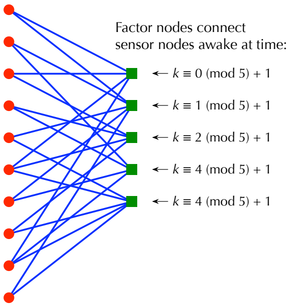
Factor nodes connect
sensor nodes awake at time:
← $k \equiv 0 \pmod 5 + 1$
← $k \equiv 1 \pmod 5 + 1$
← $k \equiv 2 \pmod 5 + 1$
← $k \equiv 4 \pmod 5 + 1$
← $k \equiv 4 \pmod 5 + 1$

Fig. 1. A bipartite graph representing a sleep strategy. Left nodes represent different sensors, right nodes represent different time instants.

$\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ i.i.d. from a distribution $\mathsf{p}_{\mathbf{x}}(\mathbf{x})$, yields a prior distribution on the channel gains which incorporates dependence among them. In particular, channel fades are frequently modelled according to path loss models, which in their highest level of generality give probability distributions for the channel gain $h_{i,j}$ between nodes n_i and n_j based on the distance between them $\|\mathbf{x}_i - \mathbf{x}_j\|_2$. This gives a prior distribution for the channel gains, which we collect into the matrix $\mathbf{H} := [h_{i,j} | i, j \in \{1, \dots, N\}]$ as

$$\mathsf{p}_{\mathbf{H}} := \int \prod_{i=1}^{N} \mathsf{p}_{\mathbf{x}_i} \prod_{j>i} \mathsf{p}_{h_{i,j}|\|\mathbf{x}_i - \mathbf{x}_j\|_2} \mathsf{d}\mathbf{x}_1 \cdots \mathsf{d}\mathbf{x}_N$$

We presently outline a scheme for sounding the channels to collect observations among the awake nodes during the sleep cycle, and a way to estimate the channels collaboratively by collecting data among different sleep cycles, while still keeping the amount of information transmitted (measured in number of real numbers) among the awake nodes per sleep cycle constant.

2. CHANNEL TRAINING MODEL

Classically when one is estimating the channel gain along a point to point wireless link, one transmits a pre-arranged "training" sequence $q_1, \dots, q_M$, e.g. drawn i.i.d. from an appropriate distribution, known both to the transmitter and receiver. The received sequence at the receiver is then modelled as

$$r_m := h q_m + w_m$$

with w_m modelled as independent i.i.d. noise sequence, usually modelled as Gaussian distributed with zero mean and variance σ^2. The channel coefficient is then inferred from using the observations r_m and any prior information regarding the channel coefficient h.

In order to generalize this technique to the training of channels for the very large wireless sensor network, we note that only those nodes that are awake may communicate directly with each other. We organize the collection of observations as follows: at time k, those wireless sensor nodes which are awake $\mathcal{A}(k)$, successively take turns transmitting a training sequence. Only one node transmits at a time, and the other awake nodes all record their observations, so that the all of the observations made during sleep cycle instant k are

$$r_{k,i,j,m} := h_{i,j} q_{i,m} + w_{k,i,j,m}, \; \forall i, j \in \mathcal{A}(k)$$

for $m \in \{1, \dots, M\}$ where the noise process $w_{k,i,j,m}$ is i.i.d. zero mean variance σ^2 Gaussian, and the prior agreed training sequences were generated by drawing $q_{i,m}$ i.i.d. from a given distribution, usually chosen to be zero mean unit variance Gaussian.

Let us focus next on the joint probability distribution for the observations and the channel coefficients that this model generates. Collect all of the observations made during sleep cycle time instant k into the vector $\mathbf{r}_k$:

$$\mathbf{r}_k := [r_{k,i,j,m} | i, j \in \mathcal{A}(k), \; m \in \{1, \dots, M\}]$$

Note that the observations made at different time instants are independent of one another given the channel coefficients $\mathbf{H}$, allowing the joint distribution for the observations and the channel coefficients to be written as

$$\mathsf{p}_{\mathbf{r},\mathbf{H}} := \mathsf{p}_{\mathbf{H}} \prod_{k=1}^{K} \mathsf{p}_{\mathbf{r}_k|\mathbf{H}}$$

Now, at the end of the distributed channel estimation, each sensor will have a possibly different estimate of the channel coefficients. To include the fact that each sensor would like to have a copy of the channel coefficient estimate, copy the coefficient matrix $\mathbf{H}$ N times to get $\mathbf{H}_1, \dots, \mathbf{H}_N$, with $\mathbf{H}_i$ being the copy of the channel coefficient matrix for node n_i. Then we can write the joint distribution as

$$\mathsf{p}_{\mathbf{r},\mathbf{H},\mathbf{H}_1,\dots,\mathbf{H}_N} = \prod_{k=1}^{K} \mathsf{p}_{\mathbf{r}_k|\mathbf{H}} \prod_{i=1}^{N} \delta(\mathbf{H} - \mathbf{H}_i) \left(\mathsf{p}_{\mathbf{H}}(\mathbf{H}_i)\right)^{\frac{1}{N}}$$

with δ being the point mass distribution at zero. Note that the a posteriori marginal density for any of the $\mathbf{H}_i$ yields the a posteriori density for $\mathbf{H}$ as desired.

Because the nodes have limited computational abilities, we assume that a node will need on the order of the amount of an entire sleep cycle (K time instants) in order to react to (either re-route or produce the result of a calculation dependent on) any messages it receives from other nodes. This requirement can be attributed to the fact that the messages must be decoded (they will have to be encoded with error correction codes since they must be passed over noisy channels), interpreted, and reacted to, while it also proves analytically convenient for the upcoming results. If this entire sleep cycle reaction time constraint is obeyed, then after ℓ complete sleep cycles, a sensor node n_i has, through the random sleep strategy, only the opportunity to communicate (even indirectly) with those nodes that are no more than 2ℓ edges away from

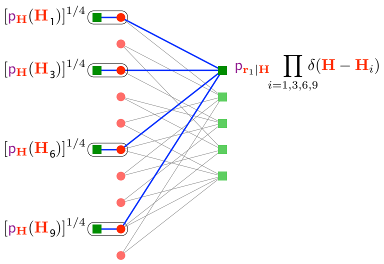
$[\mathrm{p}_{\mathbf{H}}(\mathbf{H}_1)]^{1/4}$
$[\mathrm{p}_{\mathbf{H}}(\mathbf{H}_3)]^{1/4}$
$[\mathrm{p}_{\mathbf{H}}(\mathbf{H}_6)]^{1/4}$
$[\mathrm{p}_{\mathbf{H}}(\mathbf{H}_9)]^{1/4}$
$\mathrm{p}_{\mathbf{r}_1|\mathbf{H}} \prod_{i=1,3,6,9} \delta(\mathbf{H}-\mathbf{H}_i)$

Fig. 2. An example ($\ell = 1$) factor graph that EP operates on during wireless sensor network initialization.

it in the bipartite graph representing the sleep cycle. Thus, after ℓ sleep cycles, the estimates for $\mathbf{H}_i$ can only be consistent with the nodes that are no more than 2ℓ edges away from it in the bipartite graph of the random sleep strategy, and additionally can only be influenced by the observations in this subgraph. This motivates considering a different joint distribution which takes into account the information that can be gathered and the consistency that can be enforced after only ℓ iterations with respect to sensor node n_i

$$\mathrm{p}^{(\ell)}_{\mathbf{r}(\mathcal{T}(i,\ell)),\mathbf{H},\mathbf{H}(\mathcal{P}(i,\ell))} = \prod_{k\in\mathcal{T}(i,\ell)} \mathrm{p}_{\mathbf{r}_k|\mathbf{H}} \prod_{j\in\mathcal{P}(i,\ell)} \delta(\mathbf{H}-\mathbf{H}_j)\left(\mathrm{p}_{\mathbf{H}}(\mathbf{H}_j)\right)^{\frac{1}{g(\ell)}} \quad (1)$$

where $g(\ell)$ is the number of sensor nodes no more than 2ℓ edges away from a given sensor in the sensor network in the bipartite graph representing the sleep strategy, and $\mathcal{T}(i,\ell)$ and $\mathcal{P}(i,\ell)$ are the number of sleep cycle time instants k' and node indices j no more than 2ℓ edges away form sensor node n_i in the sleep cycle bipartite graph, respectively. Here, we have used the additional notation $\mathbf{H}(\mathcal{P}(i,\ell)) := \{\mathbf{H}_j \,|j \in \mathcal{P}(i,\ell)\}$ and $\mathbf{r}(\mathcal{T}(i,\ell)) := \{\mathbf{r}_k \,|k \in \mathcal{T}(i,\ell)\}$. The joint density (1) provides the exact a marginal a posteriori distribution for $\mathbf{H}_i$ given those observations that can be collected and consistency constraints that can be enforced after ℓ iterations. For purposes that will become clearer in the next section, we associate the factorization (1) with a factor graph, as in Figure 2.

3. EP BASED DISTRIBUTED CHANNEL ESTIMATION

We presently show that EP can be used for distributed estimation during the initialization phase of a wireless sensor network, by merely considering it as a message passing algorithm on the factor graph shown in Figure 2. The algorithm operates as follows. During each time instant k in the sleep cycle, once all of the nodes have transmitted their training sequence, they take turns broadcasting their observations *made during only this awake period*, collectively denoted by $\mathbf{r}_k$, (using a sufficiently low rate code so as to mitigate the fact that channel gains are not yet known) to each other, along with the parameters for their exponential family (right going) messages calculated in terms of previous saved left going messages as

$$\boldsymbol{\rho}^{(p-1)}_{i\to k} := \sum_{k'\in\mathcal{N}(i)\setminus\{k\}} \boldsymbol{\lambda}^{(p-1)}_{k'\to i}$$

where $\mathcal{N}(i) := \{k|i \in \mathcal{A}(k)\}$. They then calculate their new left going messages, with the new message $\boldsymbol{\lambda}^{(p)}_{k\to i}$ solving the equation

$$\frac{\int \mathbf{v}(\mathbf{H}) \exp\left((\boldsymbol{\lambda}^{(p)}_{k\to i} + \boldsymbol{\rho}^{(p-1)}_{i\to k}) \cdot \mathbf{v}(\mathbf{H})\right) \mathrm{d}\mathbf{H}}{\int \exp\left((\boldsymbol{\lambda}^{(p)}_{k\to i} + \boldsymbol{\rho}^{(p-1)}_{i\to k}) \cdot \mathbf{v}(\mathbf{H})\right) \mathrm{d}\mathbf{H}} = \frac{\int \mathbf{v}(\mathbf{H}) \mathrm{p}_{\mathbf{r}_k|\mathbf{H}} \exp\left(\mathbf{v}(\mathbf{H}) \cdot \sum_{i'\in\mathcal{A}(k')} \boldsymbol{\rho}^{(p-1)}_{i'\to k'}\right) \mathrm{d}\mathbf{H}}{\int \mathrm{p}_{\mathbf{r}_k|\mathbf{H}} \exp\left(\mathbf{v}(\mathbf{H}) \cdot \sum_{i'\in\mathcal{A}(k')} \boldsymbol{\rho}^{(p-1)}_{i'\to k'}\right) \mathrm{d}\mathbf{H}} \quad (2)$$

which they save and go to sleep. Here p is the number of sleep cycles which have occurred, and after the entire pth sleep cycle has finished the messages from sleep cycle $p-1$ can be erased, and p may be incremented. Note that the observations are only made and transmitted during the first sweep of the sleep cycle, after which the communication only involves left and right going messages. We presently show how to select the exponential family that the messages being passed should lie in.

4. MESSAGE FAMILY SELECTION

We assume that the conditional distributions for the observations $\mathbf{r}_k$ made during the sleep period k given

$$\mathbf{h}_k := \{h_{i,j}|i, j \in \mathcal{A}(k)\}$$

are exponential families, so that

$$\mathrm{p}_{\mathbf{r}_k|\mathbf{h}_k}(\mathbf{r}_k|\mathbf{h}_k) = \exp\left(\mathbf{s}(\mathbf{h}_k)\cdot\mathbf{t}(\mathbf{r}_k) - \psi_{\mathbf{t}}(\mathbf{s}(\mathbf{h}_k))\right)$$

Furthermore, we assume that the prior distribution for the channel is an exponential family distribution

$$\mathrm{p}_{\mathbf{H}}(\mathbf{H}) = \exp\left(\mathbf{u}(\mathbf{H})\cdot\boldsymbol{\gamma} - \psi_{\mathbf{u}}(\boldsymbol{\gamma})\right)$$

Then, we will select the message exponential family to be used in EP to be

$$\mathbf{v}(\mathbf{H}) := \mathsf{basis}\left(\left[\mathbf{u}(\mathbf{H})^T, \mathbf{s}(\mathbf{h}_1)^T, \cdots, \mathbf{s}(\mathbf{h}_N)^T\right]^T\right)$$

where T denotes transposition, and the basis operation removes any elements of the vector argument which can be written as linear combinations (with coefficients drawn from the real numbers) of other elements in the vector argument, and does nothing if all of the elements of the vector argument are linearly independent (when coefficients are drawn from the real numbers). Note that this message exponential family selection, *after* each copy of the channel matrix $\mathbf{H}_i$ has been re-interpreted as coming from it, makes EP equivalent to belief propagation (BP). Furthermore, with this message exponential family selection, the factor node computation (2) amounts to a summation of the natural parameters of the conditional distribution $\mathsf{p}_{\mathbf{r}_k|\mathbf{H}}$ with the incoming messages. However, the difficulty of message family selection has lead some authors to dismiss BP as a possibility, proposing instead to pass message histograms for the densities involved [3]. It is this fact that motivated starting with an EP formulation in the present context, from which a proper message family selection has been made clear.

5. OPTIMALITY OF EP BASED DISTRIBUTED BAYESIAN ESTIMATION

With probability $\rightarrow 1$ in the limit as the number of nodes in the network $N \rightarrow \infty$ (with the number of sleep cycle instants scaling accordingly), the subgraph of the factor graph containing all nodes no more than 2ℓ edges away from a variable node (corresponding to a particular sensor) $\mathbf{H}_i$ becomes a tree [4, Appendix A]. Note that this idea forms the basis of density evolution performance analysis of BP [4] and EP [2]. This implies that EP with the given message exponential family density, which is equal to BP, provides the exact a posteriori density for $\mathbf{H}_i$ given those observations no more than 2ℓ edges away from node n_i, which are the only nodes which node n_i has had any communication (both direct or indirect) with after ℓ iterations. Since the a posteriori density is necessary in Bayesian estimation if we want the freedom to select any cost function (which is minimized to find the Bayesian estimate), and EP calculates the a posteriori density exactly, any method which shares the same message passing schedule must have equal or worse performance compared to EP: it can do no better than calculate the a posteriori distribution given the observations of those nodes with which it has had contact, either direct or indirect.

6. COMPARISON WITH DISTRIBUTED ADAPTIVE FILTERING TECHNIQUES

Given the growing literature on consensus based methods for estimation in sensor networks, it is important to orient the optimality result just presented with respect to other techniques in the literature. Wireless sensor network distributed MAP detection and estimation using BP was explored in [5], but that work neither recognized the potential for BP/EP to be used in contexts where different sensors wish to infer different parameters as in [1], nor considered its potential for use with Bayesian estimation with arbitrary cost functions, nor exploited the potential of the sleep strategies in terms of model structure. Alternatively, diffusion LMS [6, 7, 8] and its close RLS cousin [9] might also be considered for the distributed channel estimation application, although some modifications are first required. The standard formulation in those algorithms involves a data model as

$$\mathbf{r}_{k,i,m} = h q_{k,i,m} + w_{k,i,m}$$

in which the channel coefficient h is the *same* as seen from each sensor n_i, which is not the case here since the random sensor deployment has no reason to give a channel gain which appears constant throughout the sensor network area. The same limitation would apply to consensus propagation algorithms [10] to the extent that they are formulated for the estimation of network-based averages. In the special case of network channel estimation (in which all of the nodes wish to know all of the channel gains in the entire network and the models are typically conditionally linear), this limitation can, in principle, be overcome by considering a multichannel formulation of LMS / RLS algorithms (e.g., [11], [12]) which can likewise be given a distributed flavor [13], [14]. To adapt these algorithms to the distributed channel estimation problem, all of the channel coefficients $\mathbf{H}$ are collected into a vector $\mathbf{h}$, which all of the nodes aim to estimate. The statistics of the training signals for the distributed LMS/RLS are then selected so that the channel coefficients not appearing in the observations at a node n_i at a particular time instant (due to the sleep strategy) are multiplied by training signal elements which are identically zero.

On the other hand, the process of adapting (the multichannel versions) of these algorithms to the distributed channel estimation problem could lead to excitation problems for the adaptive algorithms due to this necessary modification of regressor statistics. Furthermore, due to the inherently linear model employed, these distributed adaptive filtering algorithms when applied to the problem of channel estimation with a known training sequence, by design aim at iteratively searching for the linear minimum mean squared error (LMMSE) estimate of the observations $\mathbf{r}_k$ (of given order) given the training signals q, with the coefficients of the linear combination yielding the estimates of the channel coefficients $\mathbf{H}$. This process does not make explicit use of the prior distribution on the channel coefficients $\mathbf{H}$, in fact, the distributed RLS/LMS treats them as deterministic. Because the MMSE estimate (the conditional expectation given the observations) always has lower or equal MSE to the LMMSE estimator, there is a built in decrease in performance: even were diffusion LMS to reach its optimum its performance would be at best equal to that of EP as we have formulated it for initialization in large sensor networks employing random sleep strategies *and for the particular squared error cost function*. Furthermore, owing to its stochastic gradient descent nature, the algorithm also suffers from misadjustment due to weight noise from the small but finite step size, further decreasing its performance from that obtained by EP, even for its particular cost function, and even when allowed the same or more frequent message passing schedule as EP (which must be carried out along the edges of the sleep strategy graph because an asleep node can not communicate). Additionally, diffusion LMS, when local consistency is enforced at each descent step, which corresponds to each additional observation $\mathbf{r}_{k,i,m}$ over the index m, would also appear to require more inter-sensor node communication than the (inherently block formulated over m) EP.

However, as was the case with LMS, the main benefit of diffusion LMS relative to the exact a posteriori density calculation is the significant reduction in computational complexity and the ability of the algorithm to perform well despite model mismatch, or unknown signal models. Future work, then, must make a careful comparison of the relative computational complexity and communications requirements (per message passed) of the two methods under the sensor network sleep strategies. Such a comparison must aim to see just how much the increase in performance and versatility of EP costs so that end users can decide which of the two algorithms fit their application.

7. CONCLUSIONS

We have provided theoretical results proving the ability of EP to provide distributed channel estimation during the initialization phase of a wireless sensor network. Because it provides exact a posteriori distributions for the channels given the observations in this scenario, EP can be adapted to provide any Bayes estimate from any cost function, providing the best performance possible with any such cost function over any scheme with internode communication obeying the same schedule. We compared our results with alternative techniques, such as distributed adaptive filtering and consensus propagation, showing that the alternative methods under the channel estimation model presented in this paper must have at best equal performance to EP. Now that superior performance and versatility of EP under these conditions has been established, the paper concluded with a suggestion for future work comparing the amount of communication required by EP (with its block formulation presented here) with the distributed adaptive filtering methods, as well as the computational complexity of the methods for particular models.

8. REFERENCES

[1] J. M. Walsh and P. A. Regalia, "Expectation propagation for distributed estimation in sensor networks," in *8th IEEE International Workshop on Signal Processing Advances for Wireless Communications (SPAWC)*, Helsinki, Finland, June 2007.

[2] J. M. Walsh and P. A. Regalia, "Belief propagation distributed estimation in sensor networks: An optimized energy accuracy tradeoff," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008, To appear.

[3] A. T. Ihler, III J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric Belief Propagation for Sensor Network Self-Calibration," *IEEE J. Select. Areas Commun.*, vol. 23, april 2005.

[4] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[5] V. Saligrama, M. Alanyali, and O. Savas, "Disributed Detection in Sensor Networks with Packet Losses and Finite Capacity Links," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4118–4132, Nov. 2006.

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

[7] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Honolulu, HI, Apr. 2007, vol. 3, pp. 917–920.

[8] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Consensus-based distributed least-mean-square algorithm using wireless sensor networks," in *Proc. 45th Allerton Conference*, Monticello, IL, 2007.

[9] I. D. Schizas, G. Mateos and G. B. Giannakis, "Distributed Recursive Least-Squares Using Wireless Ad Hoc Sensor Networks," in *Proc. of 41st Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2007.

[10] C. C. Moallemi and B. Van Roy, "Consensus propagation," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

[11] M. Viberg, B. Ottersten and T. Kailath, "Detection and estimation in sensor arrays using weighted subspace fitting," *IEEE Trans. Signal Processing*, vol. 39, no. 11, pp. 2436–2449, Nov. 1991.

[12] G. Glentis and N. Kalouptsidis, "Fast adaptive algorithms for multichannel filtering and system identification," *IEEE Trans. Signal Processing*, vol. 40, no. 10, pp. 2433–2458, Oct. 1992.

[13] H. Lev-Ari, "Modular architectures for adaptive multichannel lattice algorithms," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 35, no. 4, pp. 543–552, Apr. 1987.

[14] H. V. Jagadish, S. K. Rao and T. Kailath, "Array architectures for iterative algorithms," *Proceedings of the IEEE*, vol. 25, no. 9, pp. 1304–1321, Sept. 1987.