DISTRIBUTED DECISION IN SENSOR NETWORKS BASED ON LOCAL COUPLING THROUGH PULSE POSITION MODULATED SIGNALS

Loreto Pescosolido and Sergio Barbarossa

INFO-COM Dept., University of Rome "La Sapienza,"via Eudossiana 18, 00184 Rome, Italy

e-mail:{loreto,sergio}@infocom.uniroma1.it

ABSTRACT

In this work we propose a physical layer design, based on Pulse Position Modulated (PPM) signals, for a decentralized wireless sensor network implementing an iterative consensus algorithm. The proposed scheme does not require any MAC protocol to avoid or resolve collisions, and is also suitable for a half-duplex implementation. The considered network model assumes only local coupling among the nodes, thus allowing for low transmit power even in large scale networks. Furthermore, we show how to remove the effect of propagation delays, multipath, and non perfect synchronization among the nodes, without requiring any channel parameter estimate. As an example of application, we consider a simple parameter estimation problem, which is instrumental to discuss the fundamental trade-offs arising in the system parameters settings, when both observation noise and coupling noise are considered in the performance analysis.

Index Terms— Decentralized sensor networks, consensus, PPM

1. INTRODUCTION

A considerable amount of research efforts have been recently devoted to the design and analysis of consensus algorithms [2]. The fields of application of such system range from cooperative dynamic control of moving agents [1] to decentralized sensor networks [3, 4] (see also the references therein). Despite the considerable number of works in this area, not as much attention has been devoted to find out appropriate radio interfaces to implement the proper interaction among the nodes. One exception is represented by [6], where a consensus protocol is proposed based on the principle of data driven consensus and employing Type Based Multiple Access. The great relevance of this aspect is clear if one considers the possibly very large number of nodes of the network and the inherently iterative nature of the consensus algorithms. These two aspects, in fact, pose serious problems to the design of a communication strategy based on conventional approaches involving the use of MAC and scheduling protocols. In the case of a wireless network, in fact, issues like collision avoidance, with possibly repeated transmission in each iteration, or iteration number tracking for each received packet, would involve a very complicated system design, and a considerable amount of consumed energy.

The scope of this work is to propose a physical layer strategy that allows the nodes to implement the consensus algorithm without the need of avoiding or resolving the collision of packets orriving on the same node at the same time. Our approach requires very simple operations at the receiver to extract from the received signal just the amount of information that is useful to compute the update term that, in each iteration, a node adds to its state in order to implement the equations governing the convergence to a consensus value. A distinctive feature of our approach is that it does not require the receiver to discriminate the signals coming from different transmitters, and it does not require any channel estimate. We will show that the proposed method is robust to propagation delays, multipath, and non perfect synchronization among the nodes, provided that the signalling interval, i.e. the time occurring between two successive iterations, is large enough.

We consider a conventional discrete-time consensus algorithm, see e.g. [7], in which the states of the nodes converge to a common value, and show how to directly implement it at the physical layer through the use of PPM signals. In particular, given a set of N nodes, the equation for the dynamical system in each node is given by [2]:

$$x_i[n+1] = x_i[n] + K\left(\sum_{j=1}^{N} a_{ij}\left(x_j[n] - x_i[n]\right)\right), \qquad (1)$$

where $x_i[n]$ is the state variable of the i-th node at the n-th iteration, $\forall i \in \{1, \ldots, N\}$; $a_{ij} \geq 0$ are the coupling coefficients between node i and its neighbors, that we assume to be real and positive, and K is a coupling gain constant. In this work, we assume that the coupling coefficients are constant, or, in other words, that the network topology is fixed. In Section 2 we describe our system model, i.e. the considered discrete-time algorithm and its convergence properties. In Section 3 we describe the proposed physical layer design. As an application example, in Section 4, we will consider, for simplicity, the estimation of a common parameter of interest, and describe the fundamental tradeoff in the system parameter settings, when the joint effect of observation noise and coupling noise are considered in the performance analysis. Clearly, the same implementation could be used, in principle, to perform different tasks, as long as their mathematical formulation can be cast in terms of consensus. In Section 5, finally, we draw our conclusions, also pointing out several possible extensions of the proposed design.

2. SYSTEM MODEL

We consider a network of N nodes that is able to reach a globally optimal decision by means of a distributed consensus algorithm. Each node implements the discrete-time dynamical system (1), whose state variable evolution depends on the initial conditions and, in each iteration, on a term that carries information about the state variables of the node's neighbors. Defining the state increment

$$\delta_i[n+1] \triangleq K \sum_{j \in \mathcal{N}_i} a_{ij}\left(x_j[n] - x_i[n]\right), \qquad (2)$$

and incorporating the presence of coupling noise, the considered update equations and initial conditions can be written as

$$\begin{cases} x_i[n+1] = x_i[n] + \delta_i[n+1] + K w_i[n] \\ x_i[0] = s_i \qquad i = 1, \ldots, N \end{cases}, \qquad (3)$$

This work has been partially funded by the European Project IST-WINSOC under grant no. 0033914.

where $w_i[n]$ represents an additive Gaussian noise term with zero mean and variance σ_w^2, and s_i, $i = 1, \ldots, N$, are the initializing parameters. To emphasize the local nature of the interactions among nodes, in (2) we have introduced, for each node i, the set of its neighbors $\mathcal{N}_i$, i.e. the set of nodes for which $a_{ij} \neq 0$. Clearly the meaning of the sum in (2) is the same as that of the sum in (1).

We assume that the noise realizations referring to different nodes and to different iterations are statistically independent.

The convergence of the system depends on the coupling gain K and on the network topology, which is usually represented by a graph with vertexes corresponding to the nodes. If the coefficient a_{ij} is greater than zero, there is a link between the nodes i and j, in the direction from j to i, with weight a_{ij}.

The graph associated to the network can be either *undirected*, i.e. with $a_{ij} = a_{ji}\ \forall (i,j) \in \{1, \ldots, N\}^2$, or *directed*, in which case the coefficients can be asymmetric. Since the focus of this paper is on the physical layer design we assume, w.l.o.g., that the graph is undirected, but the same design could be used as well with consensus algorithms running on networks with asymmetric graphs.

We briefly resume now the convergence properties of the above algorithm. For details the interested reader is encouraged to check, e.g., [2,3,7]. Under the fixed topology assumption, it can be showed that, if the graph is connected, in the absence os noise, the system converges[1] to the following asymptotic value of the states variables[2]

$$\lim_{n \longrightarrow \infty} x_i[n] = \frac{1}{N} \sum_{i=1}^{N} x_i[0] \triangleq x^*,\ \forall i \in \{1, \ldots, N\}. \quad (4)$$

The convergence rate depends on the coupling strength, which is determined by the products Ka_{ij}. In our set up, the coefficients a_{ij} represent a physical parameter that depends on the nodes' transmit power and the path losses. For a given set of coefficients a_{ij}, which guarantees the network connectivity, the multiplying constant K determines the convergence speed.

It is important to remark that the local nature of the interaction allows the use of *short range transmissions*, with respect to the network size, provided that the nodes density, and the transmit power, are sufficiently high to insure the network connectivity.

3. IMPULSE RADIO IMPLEMENTATION

We assume that time is divided into slots of T_s seconds, denoted *signalling intervals*. In each time slot, a single iteration of the above algorithm is carried out. In practice, in the n-th iteration, each node transmits a signal representing its state variable, and receive the signals transmitted from its neighbors. Building on its received signal, each node computes the state increment (2) to add to its state variable.

The implementation we propose is based on Pulse Position Modulated signals (PPM). Each node modulates a baseband pulse $p(t)$ of duration τ_p seconds, much shorter than T_s. We assume $p(t)$ has unit area, i.e. $\int_0^{\tau_p} p(t) = 1$.

Each node transmits a pulse $A_i p(t)$, with A_i chosen in order to enforce an average pulse power $\mathcal{P}_i = (1/\tau_p) \int_0^{\tau_p} A_i^2 p^2(t)$. The average *radiated* power of a node during the evolution of the algorithm is given by the average pulse power multiplied by the duty cycle τ_p/T_s: $\mathcal{P}_{i,av} = \mathcal{P}_i(\tau_p/T_s)$.

[1]Convergence is guaranteed provided that the coupling gain is chosen in a suitable interval, namely $K \in (0, 1/2\lambda_{L\max})$, where $\lambda_{L\max}$ is the largest eigenvalue od the Laplacian matrix defined in Section 3.1.

[2]Asymptotic convergence is valid only assuming the absence of coupling noise. In the presence of coupling noise, convergence has to be defined in a statistical sense, see Section 3.

Let us assume that the state variables x_i, $i \in \{1, \ldots, N\}$, can take values comprised in a range $[-\Delta/2, \Delta/2]$. In the n-th iteration, each node maps its state variable $x_i[n]$ *to the position* $t_i[n]$ of the baseband pulse $p(t)$ in the interval $[-T_m/2, T_m/2]$, that we call *mapping* interval, using the linear mapping:

$$t_i[n] = x_i[n](T_m/\Delta). \quad (5)$$

As explained in Subsection 3.1, to recover the effect of propagation delays, multipath, and nonperfect synchronization, the signalling interval duration must be chosen greater than the mapping interval one, i.e. $T_s > T_m$.Focusing on the n-th iteration, we place the origin of the time axis at the instant corresponding to the center of the mapping interval. With this choice, the signalling interval corresponds to $[-T_m/2\ ,\ T_s - T_m/2]$, and the signal transmitted from the i-th node during the n-th signalling interval is:

$$p_i(t) \triangleq A_i p(t - t_i[n]). \quad (6)$$

3.1. Signal processing at the receiver

We assume a multipath propagation channel in each link, so that the signal transmitted from node j is received from node i in L_{ij} replicas with respective delays $\tau_{ij}^1, \ldots \tau_{ij}^{L_{ij}}$, and path losses $h_{ij}^1, \ldots h_{ij}^{L_{ij}}$. We chose the signalling interval T_s so that it contains a guard interval equal to the maximum delay of the network, i.e.

$$T_s > T_m + \tau_{\max}, \quad (7)$$

where $\tau_{\max}$ is defined as

$$\tau_{\max} \triangleq \max \tau_{ij}^l, \forall (i,j) \in \{1, \ldots, N\}^2, \forall l \in \{1, \ldots, L_{ij}\}. \quad (8)$$

In each signalling interval, each sensor receives a signal of the form

$$r_i(t) = \sum_{j \in \mathcal{N}_i} \sum_{l=1}^{L_{ij}} h_{ij}^l A_j p\left(t - t_j[n] - \tau_{ij}^l\right) + \tilde{v}_i(t), \quad (9)$$

where $\tilde{v}_i(t)$ is the receiver noise at the output of the matched filter, assumed to be Gaussian with zero mean and variance $\tilde{\sigma}^2$.

In writing (9), we have implicitly assumed that the nodes are synchronized at the starting instant of each iteration. Later, we will see that nonperfect synchronization can be recovered by simply adding a guard interval *before* the start of the *mapping* interval.

We now describe the procedure that the receiver implements to extract from the received signal the rightmost term in the right hand side of (1), which we call the *state increment* to update the current state value. We show next that, thanks to the PPM mechanism, recovering the state increment simply amounts for a sensor to perform a double integration.

To compute its state increment in the n-th iteration, the i-th receiver performs the following operations:

1) shift of the time axis origin from the center of the mapping interval to *the emission time of its own pulse in the same iteration*. In this way, the received signal can be rewritten as

$$r_i(t) = \sum_{j \in \mathcal{N}_i} \sum_{l=1}^{L_{ij}} h_{ij}^l A_j p\left(t - t_j[n] - \tau_{ij}^l + t_i[n]\right) + \tilde{v}_i(t), \quad (10)$$

and the signalling interval corresponds now to $[-T_m/2 - t_i[n]\ ,\ T_s - T_m/2 - t_i[n]]$.

2) integration of the received signal, producing a signal $z_i(t)$, with $t \in [0, T_s]$, according to the following rule:

$$z_i(t) \triangleq \int_{\tilde{T}_s - t}^{\tilde{T}_s} r_i(t')\, dt', \tag{11}$$

where $\tilde{T}_s \triangleq T_s - T_m/2 - t_i[n]$, is the final instant of the signalling period in the new time axis. It can be showed that

$$z_i(t) = \sum_{j\in\mathcal{N}_i} \sum_{l=1}^{L_{ij}} h_{ij}^l A_j u\left(t - \left(\tilde{T}_s - (t_j[n] - t_i[n])\right) + \tau_{ij}^l\right) + \tilde{v}_i(t), \quad \forall t \in [0, T_s], \tag{12}$$

where we have defined the integrated pulse $u(t) \triangleq \int_0^t p(t')\, dt'$, and the integrated noise term $v_i(t) \triangleq \int_{\tilde{T}_s - t}^{\tilde{T}_s} \tilde{v}_i(t' + t_i[n])\, dt'$. Notice that, in the absence of noise, $z_i(T_s) = \sum_{j\in\mathcal{N}_i} \sum_{l=1}^{L_{ij}} h_{ij}^l A_j$.

3) integration of the processed signal $z_i(t)$ a second time, from 0 to T_s, and subtraction[3] of $z_i(T_s)(T_m/2 + t_i[n])$, obtaining:

$$\begin{aligned} y_i[n] &= \int_0^{T_s} z_i(t')\, dt' - z_i(T_s)\left(\frac{T_m}{2} + t_i[n]\right) \\ &= \sum_{j\in\mathcal{N}_i} \sum_{l=1}^{L_{ij}} h_{ij}^l A_j \left(t_j[n] - t_i[n] + \tau_{ij}^l\right) + v_i[n], \end{aligned} \tag{13}$$

where we have used the property that the pulses $p(t)$ have unit area. The noise term, here, is $v_i[n] \triangleq \int_0^{T_s} v_i(t')\, dt'$.

4) multiplication of $y_i[n]$ for $K\Delta/T_m$, obtaining

$$\begin{aligned} \tilde{\delta}_i[n+1] =& K\frac{\Delta}{T_m} y_i[n] \\ =& K \sum_{j\in\mathcal{N}_i} a_{ij}(x_j[n] - x_i[n]) + \\ & K\frac{\Delta}{T_m} \sum_{j\in\mathcal{N}_i} \sum_{l=1}^{L_{ij}} a_{ij}^l \tau_{ij}^l + K w_i[n], \end{aligned} \tag{14}$$

where we have used (5), and introduced the coefficients $a_{ij}^l \triangleq h_{ij}^l A_j$ and $a_{ij} \triangleq \frac{\Delta}{T_m} \sum_{l=1}^{L_{ij}} a_{ij}^l$, and the noise term $w_i[n] \triangleq v_i[n]\, \Delta/T_m$.

Defining $\Psi_i \triangleq \sum_{j\in\mathcal{N}_i} \sum_{l=1}^{L_{ij}} a_{ij}^l \tau_{ij}^l$ as the *"delays-gains product sum"* at node i, and using (2), we can rewrite (14) as

$$\tilde{\delta}_i[n+1] = \delta_i[n+1] + K w_i[n] + \Psi_i K\Delta/T_m. \tag{15}$$

We can see that this term contains the correct expression of the state increment (2), plus a noise term and bias term. Interestingly, Ψ_i does not depend on the state variables, but only on the gains and the delays, which are assumed constant during the evolution of the dynamical systems. This observation suggests a simple way for each node to estimate, *prior to the algorithm run*, its own Ψ_i, to be able to subtract it from (15), thus obtaining an unbiased estimate of the correct increment. In fact, looking at (13) one can see that if all the nodes transmit a pulse centered at the origin, and perform the double integration, each of them obtains exactly an estimate $\hat{\Psi}_i$ of its own bias term. The estimate can be made more accurate by simply reiterating the transmission more times, and averaging over the observations. It can be showed that the effect of an inaccurate estimate is to introduce a common drift in the evolution of the states variables.

Remark 1: The proposed scheme can be equally well implemented with half-duplex transceivers. In the absence of delays, this is obvious, since if a pulse is not seen by the receiver because its arrival time coincides with the emission time of the same node, its contribution would disappear from the integral term. But this is exactly what is desired, as this means that the state variables of transmitter and receiver were equal, and hence their difference would in any case give a null contribution to the state increment. The situation is different in the presence of propagation delays, but it can be shown that the estimation and subtraction of Ψ_i compensates for the signals shadowing due to the possible use of *half-duplex* transceivers.

Remark 2: In the presence of synchronization offsets among the nodes, their effects can be recovered simply inserting two guard intervals in each slot, of duration $\varphi_{\max} \triangleq \max_{(i,j)\in\{1,\ldots,N\}^2}(|\varphi_{ij}|)$, with φ_{ij} being the offset between nodes i and j. In this way, it is insured that the pulse emitted by a node in the n-th iteration, will be received by its neighbor nodes in the signalling interval referred to the same iteration. The resulting delays are given now by $\tilde{\tau}_{ij}^l = \tau_{ij}^l + \varphi_{ij}$ and could be negative, hence the guard interval at the beginning of the slot. Nevertheless, their presence is compensated by the same mechanism described above.

As far as the noise term is concerned, it can be showed that the variance of the discrete-time noise term $w_i[n]$ at the output of the double integrator, is related to the variance $\tilde{\sigma}^2$ of the noise $\tilde{v}_i(t)$ at the output of the receiver's matched filter through

$$\sigma_w^2 = \tilde{\sigma}^2 (\Delta/T_m)^2 T_s^3/3.$$

The presence of coupling noise implies that the convergence of the system has to be intended in a statistical sense. To be precise, it can be showed that the linear system converges only in the mean. It can be proved, see [7], that, due to the presence of coupling noise, the state of each node undergoes a random walk, i.e. the evolution of $x_i[n]$ is affected by an additive zero mean Gaussian noise term $\upsilon_i[n]$. If $\boldsymbol{L}$ is the *Laplacian* matrix associated to the network graph[4], and $\lambda_1, \ldots, \lambda_N$ are the eigenvalues of the matrix $(\boldsymbol{I} - K\boldsymbol{L})$, in non-decreasing order, it can be shown that, $\forall i = 1, \ldots, N-1$, $\lambda_i \in (-1, 1)$; $\lambda_N = 1$; and.the variance of $\upsilon_i[n]$ is given by

$$\sigma_{\upsilon_i}^2[n] = \frac{K^2\sigma_w^2}{N}\left(n + \sum_{i=1}^{N-1} \frac{1-\lambda_i^{2n}}{1-\lambda_i^2}\right), \tag{16}$$

i.e. it diverges to infinity as n increases.

Resorting to methods coming from the theory of statistical approximation [5], it can be showed that the divergence of the noise can be eliminated by using a time varying gain that decreases with time, e.g. $K[n] = K_0/n$.

4. NUMERICAL RESULTS

We illustrate the system performance with the following simple example. Assume that the network task is to estimate a common scalar parameter ϑ building on the set of observations gathered at the nodes $s_i = \vartheta + e_i$, $\forall i \in \{1, \ldots, N\}$, where e_i is the observation error, that we assume Gaussian with zero mean and variance σ_e^2. The ideal globally optimal estimate, which is given by $\hat{\vartheta}_{opt} \triangleq \frac{1}{N}\sum_{i=1}^N s_i$, can be computed through the distributed

[3]The effect of noise on the value of $z_i(T_s)$ is typically negligible with respect to the effect of the noise resulting from the second integration given by (13).

[4]The Laplacian is defined as the matrix with entries $l_{ij} = -a_{ij}$, if $i \neq j$, $l_{ij} = \sum_{j\in N_i} a_{ij}$ if $i = j$.

algorithm by setting $x_i[0] = s_i$. The *decentralized* estimate after n iterations, that we denote as $\hat{\vartheta}_d(n)$, is affected by a zero mean error, with variance is given by the sum of the variance of the error inherent in the *ideal* global estimate, which is σ_e^2/N, plus a term coming from the coupling noise, whose entity depends on K and on the nodes transmit power through (16).

We have simulated a network of $N = 20$ nodes deployed on a square area of 100 m^2, with multipath channels composed of $L_{ij} = 4$ paths between each couple of nodes, and with symmetric path losses modeled as $h_{ij}^1 = \sqrt{1/\left(1 + 1/d_{ij}^2\right)}$ for the first path (d_{ij} being the distance between nodes i and j), and decreasing for the remaining paths. The maximum delay was $\tau_{\max} = 0.5T_m$. The nodes compute the decentralized estimate of the parameter ϑ that we have assumed comprised in $[-30, 30]$. The local observation error variance was $\sigma_e^2 = 100$, and we have dimensioned the states range $[-\triangle/2, \triangle/2]$ choosing $\triangle = 50$.

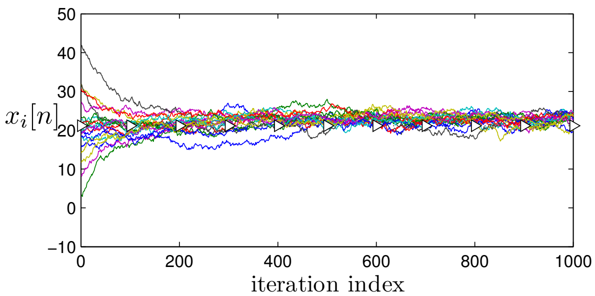

Fig. 1. States evolution of a single run

In Fig. 1, we have plotted a single realization of the algorithm run obtained through the proposed physical layer implementation. We have imposed an equal transmit power $\mathcal{P}_i$ for all the nodes. In this case the ratio between the *transmit* power[5] and the receivers noise variance was $\mathcal{P}_i/\tilde{\sigma}^2 = 25\ dB$, the coupling gain was $K = 0.15$. The plots show that the proposed design is able to effectively implement the algorithm, as the states (solid colored lines) converge, in the mean, towards the average of their initial values (white triangles).

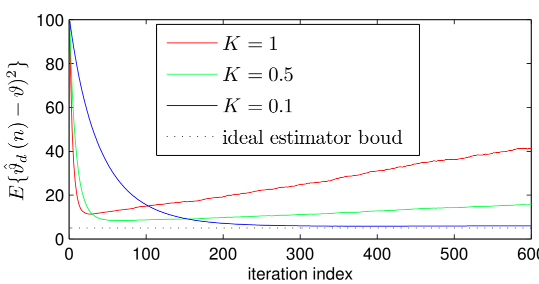

Fig. 2. Estimate variance

In Fig. 2, we show the corresponding decentralized estimation performance, in terms of the estimation error variance $\mathbb{E}\{(\hat{\vartheta}_d(n) - \vartheta)^2\}$ (solid lines), averaged over 1000 trials, for different values of K, with $\mathcal{P}_i$. The fact that for large n, the variance increases, is due to the random walk effect mentioned in Section 3.1.

[5]Notice that the values of the required SNR at the receivers is indeed very low, if one considers the assumed path loss model and a ratio between the *transmit* power and the *receiver* noise variance of 25 dB.

Since the effect of the coupling noise depends on these parameters, for a given constraint on the estimate reliability, there is a trade-off between the number of iterations necessary to let the algorithm converge (which decreases as K increases), and the transmit power. In fact, to improve the estimate reliability one has to either decrease K or increase $\mathcal{P}_i$. We can see that with $\mathcal{P}_i/\sigma^2 = 25\ dB$, the ideal lower bound (given in this case by $\sigma_e^2/N = 5$, dotted line) can be practically achieved with $K = 0.1$.

5. CONCLUSION

In this paper we have proposed a possible physical layer implementation, based on PPM signals, of a discrete-time distributed consensus algorithm suitable for wireless sensor networks. We have shown that the proposed method is robust to propagation delays, multipath, and synchronization offsets among the nodes, provided that the maximum delay between a transmitted signal and its replicas at the receiver, added to the time interval to which the state variables are mapped, does not exceed the signalling interval. The possibility to overcome these impairments is given by the estimation of a parameter that quantifies, at each node, the sum of the products of the delays by the coupling coefficients between each node and its neighbors. The implementation of the estimators does not require, at the nodes, operations different from those required by the algorithm in an idealized scenario with no delays. The most relevant advantage of this physical layer design is that it does not require the receiver to distinguish between the signals coming from different nodes, or to estimate the channel parameters, but still it is capable to implement the consensus algorithm. This fact is important because it drastically reduces the complexity of each node, through the simplification of the MAC layer, and the related reduction of energy consumption, which in a wireless sensor network is a critical issue. In a future work we will show how to extend the proposed design to systems with higher resilience to the effect of coupling noise, i.e. with an output noise variance that does not diverge to infinity.

6. REFERENCES

[1] W. Ren, R. W. Beard, and E. M. Atkins, "A Survey of Consensus Problems in Multi-agent Coordination," in *Proc. of the 2005 American Control Conference (ACC '05)*, Portland, OR, USA, June 8–10, 2005.

[2] 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.

[3] L. Xiao, S. Boyd, and S. Lall, "A Scheme for Distributed Sensor Fusion Based on Average Consensus," in *Proc. 4th IEEE International Conference on Information Processing in Sensor Networks (IPSN'05)*, Los Angeles, CA, USA, Apr. 25–27, 2005.

[4] S. Barbarossa and G. Scutari, "Bio Inspired Sensor Networks Design," *IEEE Signal Processing Mag.*, vol. 24, no. 3, pp. 26–35, Mar. 2007.

[5] G. N. Saridis, "Stochastic Approximation Methods for Identification and Control - a Survey," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 798–809, Dec. 1974.

[6] A. Scaglione, "On the Wireless Communication Architecture for Consensus Problems," in *Information Theory and its Applications (ITA 2007)*, San Diego, CA, USA, Jan. 2007.

[7] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed Average Consensus with Least-Mean-Square Deviation," *Journal of Parallel and Distributed Computing, 67(1):33-46, 2007*, vol. 67, no. 1, pp. 33–46, Jan. 2007.