

Consensus-based distributed sensor calibration and least-square parameter identification in WSNs

Saverio Bolognani, Simone Del Favero, Luca Schenato, Damiano Varagnolo

Department of Information Engineering, University of Padova,
Via Gradenigo 9/B, 35131, Padova. Fax: +39 049 827 7614.

WWW: <http://www.dei.unipd.it/>.

E-mail: {saverio.bolognani, simone.delfavero, schenato,
damiano.varagnolo}@dei.unipd.it

Abstract

In this paper we study the problem of estimating the channel parameters for a generic wireless sensor network (WSN) in a completely distributed manner, using consensus algorithms. Specifically, we first propose a distributed strategy to minimize the effects of unknown constant offsets in the reading of the Radio Strength Signal Indicator (RSSI) due to uncalibrated sensors. Then we show how the computation of the optimal wireless channels parameters, which are the solution of a global least-square optimization problem, can be obtained with a consensus-based algorithm. The proposed algorithms are general algorithms for sensor calibration and distributed least-square parameter identification, and do not require any knowledge on the global topology of the network nor the total number of nodes. Finally, we apply these algorithms to experimental data collected from an indoor wireless sensor network.

Keywords: distributed computing, sensor calibration, least-square estimation, parameter identification, consensus, wireless sensor networks.

1 Introduction

¹Wireless sensor networks (WSNs), i.e. networks of smart devices that can sense, compute and exchange information with their neighbors, are becoming very popular because of their promise to revolutionize many engineering areas involving monitoring and control [22]. The strength of WSNs resides in their flexibility and scalability, since the same hardware and software can be rapidly reconfigured and adapted to manage rather different applications, from ambient monitoring to people tracking, from industrial control to energy management in buildings. However, many challenges ranging from HW design, to real-time middleware prototyping, from data routing protocols to distributed signal processing still remain to be solved before WSNs can become really ubiquitous and successful.

In this paper we address some of the modeling and algorithmic issues which are at the base of localization and target tracking applications using WSNs [16, 18]. In fact, the wireless radio in each node of the WSN can be used not only to communicate but also to measure the radio signal strength associated with the received packet. Since the signal strength is a function of the locations of the transmitter and the receiver, it can be used to estimate their relative position. There are two main approaches to target tracking: map-based and range-based. In the map-based approach the position of the moving target is obtained by finding the most likely location which matches the recorded signal strength based on previously learned maps [18, 27]. This strategy can be a good solution but it requires extensive work to learn the maps. Differently, the range-based algorithms first try to estimate relative distance based on simple models of the wireless channel and then they estimate the position by triangulation, similarly to the GPS system where the static nodes of the WSN play the role of the satellites in the GPS [16]. This approach requires a higher node density than the map-based one, but it does not require an extensive learning phase. We focus on this last approach.

¹This work was supported by the CaRiPaRo foundation under the WiseWai founded project and by EU STREP project named "FeedNetBack".

Most of the previous work on range-based tracking proposed in the literature focuses on triangulation algorithms where the wireless channel model parameters are assumed to be known or are identified off-line by collecting all data in some centralized location [21]. Unfortunately, these parameters are strongly dependent on the environment [14], in particular indoor, therefore it is desirable to identify them in-situ, possibly using distributed algorithms suitable for the limited computational resources of the WSN nodes. Moreover, the radio signal strength provided by the radio chips of the sensor nodes are not very precise mainly due to uncalibrated offsets in the receiving nodes. As a consequence, the estimated distance can be constantly biased in some nodes, thus degrading tracking performance. Therefore, it is necessary to devise some strategies to compensate these offsets [30].

The main contribution of this work is to propose the use of consensus algorithms for automatically calibrating the sensors without the use of a reference node, and for least-square-estimating the optimal channel parameters with a distributed algorithm. Consensus algorithms are a very popular class of distributed algorithms which have been successfully applied to coordinated robotics [17], time synchronization [26, 24], and distributed estimation [28]. These algorithms share some similarities with distributed asynchronous iterative algorithms for solving, or at least approximately solving in least-square sense, linear equations $Ax = b$, where $x \in \mathbb{R}^n$ is the vector of unknown parameters, and the matrix $A \in \mathbb{R}^{m \times n}$ and the vector $b \in \mathbb{R}^m$ are known [2, 29, 13]. These latter algorithms are particularly suitable for solving sparse equations where the number of equations is equal to the number of unknowns parameters, i.e. when A is square and has many zero-entries. Although, in principle these algorithms can be applied also to a non-square matrix A , since the least square solution of $Ax = b$ is equivalent to $A^T Ax = A^T b$, it might happen that $A^T A \in \mathbb{R}^{n \times n}$, which is now square, is dense even if A is sparse, thus losing many of their advantages. Moreover, these algorithms naturally lead to parallel implementation since each element of vector x can be computed by a distinct agent. However, in many applications, such as in sensor networks, the number of physical agents, i.e. the sensor nodes, coincides with the size m of the vector b rather than the size n of the vector x , thus making the distributed implementation not feasible. Differently, consensus algorithms can be effective also in these contexts as we will show in the proposed least-square parameter identification. Nonetheless, a more thorough comparison between consensus algorithms and asynchronous iterative algorithms is ought, however this is beyond the scope of this work and we refer the interested reader to the aforementioned references and to [23] for more details. Although the two proposed consensus algorithms used for implementing distributed sensor calibration and distributed least-square parameter identification, are applied to localization and tracking for WSNs, they are very general. In fact, we will show how any problem concerned with the compensation of measurement offsets affecting a network of sensors, where the offsets cannot be directly measured but only pairwise differences of offsets are available, can be solved with the proposed approach. Moreover, we will also show how any least-square parameter identification problem based on distributed measurements does not need to collect all data in single location to compute the optimal (centralized) solution, but can be solved in a distributed fashion via a consensus algorithm. Another important contribution of this work is to mathematically model the wireless channel and the communications protocols of typical WSN based on experimental data, which is an aspect that it is often overlooked, leading to models which are unrealistic. For example, due to packet loss or time synchronization, it is rather problematic in consensus algorithms running over WSNs to enforce convergence to the mean of initial condition, i.e. to enforce average consensus. Therefore, in our work we will pose particular care in exploring the tradeoffs between perfect average consensus and randomized consensus.

The paper is organized as follows. In Section 2, we introduce some preliminary mathematical terminology and classical results of consensus algorithms which will be necessary to prove convergence of the proposed consensus algorithms. In the same section we also provide a general mathematical model for the typical communication schemes and the wireless channel model in WSNs. In Section 3 we describe the experimental testbed used to collect data and, based on these data, we find the numerical values for the model parameters given in the previous Section. In Section 5 we propose a consensus-based strategy for calibrating sensors with unknown measurement offset readings and we applied it to experimental data. In Section 6 we introduce a consensus-based least square algorithm for identifying the wireless channel parameters under different communication strategies and we highlight trade-offs between performance, speed of convergence and computation complexity, based on experimental data. Finally, in Section 7 we summarize the results and propose future research directions.

2 WSNs Modeling

2.1 Connectivity and Communications Models

We model a WSN as a set of N nodes numbered w.l.o.g. from 1 to N , i.e. $\mathcal{N} = \{1, \dots, N\}$. Since nodes communicate using a wireless channel, the transmission is not reliable, i.e. there is a non-zero packet loss probability. We model this communication unreliability with the *connectivity matrix* $C \in \mathbb{R}^{N \times N}$, where $[C]_{ij} = c_{ij} \in [0, 1]$ is the probability that node j can successfully transmit a message to node i . Since the wireless channel is approximately symmetric, we further assume that $C = C^T$ and $c_{ii} = 1, \forall i$. We define the *c-connectivity graph* $\mathcal{G}_c = (\mathcal{N}, \mathcal{E}_c)$ associated to the connectivity matrix C as the graph s.t. (i, j) belongs to the set of the edges \mathcal{E}_c if and only if $c_{ij} \geq c$. This graph is undirected (i.e. $\forall i, j \in \mathcal{N}, (i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$) since the matrix C is symmetric. We also denote with $\mathcal{V}(i) = \{j \mid (i, j) \in \mathcal{E}, i \neq j\}$ the set of *neighbors* of node i and with the *degree* $d(i) = |\mathcal{V}(i)|$ its cardinality.

The matrix C can be experimentally estimated by letting each node broadcast M packets at random instants (with retransmission intervals sufficiently large in order to avoid collisions), making each node i record the number m_{ij} of messages received by each node j and setting $\hat{c}_{ij} = \frac{m_{ij}}{M}$. Subsequently, each node communicates its \hat{c}_{ij} to its neighbors and sets $c_{ij} = \frac{\hat{c}_{ij} + \hat{c}_{ji}}{2}$ since \hat{c}_{ij} and \hat{c}_{ji} are different being empirical means.

In terms of communication there are three common strategies adopted in WSNs: the *broadcast* communication where one node i transmits a message to all its neighbors $\mathcal{V}(i)$, the *asymmetric gossip* where a node i transmits a message to a specific node $j \in \mathcal{V}(i)$, and the *symmetric gossip* where a node i transmits a message to a specific node $j \in \mathcal{V}(i)$ and then waits to receive a reply message from the same node. Moreover, associated to these two communications strategies, there are two possible modalities: *sequential* and *randomized*.

In the sequential broadcast each node in the network transmits sequentially according to a deterministic sequence, and the time interval τ between two transmissions is constant. Similarly, in the sequential gossip each edge in the network is sequentially selected for communication and the intercommunication interval τ is constant. In the randomized broadcast one node i turns on with a uniform random probability $\frac{1}{N}$ and the intercommunication interval is an exponential random variable with mean τ . Similarly, in the randomized gossip one node i turns on with a uniform random probability $\frac{1}{N}$ and selects one edge at random among all its neighbors with uniform probability $\frac{1}{d(i)}$.

2.2 Wireless Channel Model

Here we model the behavior of the wireless channel between two nodes in terms of received power P_{rx} (in *dBm*). We start by considering the most general model and then we highlight which are the most relevant elements based on experimental data. The Radio Signal Strength Indicator (RSSI) measured by a generic node i after having successfully received a packet sent by the generic node j can be modeled in the most general form as:

$$P_{rx}^{ij} = f\left(P_{tx}^j, \mathbf{x}_i, \mathbf{x}_j, i, j, t\right)$$

where P_{tx}^j (in *dBm*) is the nominal transmitted power, i and j are the IDs representing the receiver and the transmitter nodes, respectively, $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^3$ are their spatial positions and t is the time when the communication occurs. The previous equation can be decomposed into simpler elements which takes into account different effects. Combining the models of each element, described in [15] and [14], and adding parts due to offsets in the RSSI measurements and in the power transmission, we obtain the following model:

$$P_{rx}^{ij} = P_{tx}^j + r_j + f_{pl}(\|\mathbf{x}_i - \mathbf{x}_j\|) + f_{sf}(\mathbf{x}_i, \mathbf{x}_j) + f_a(\mathbf{x}_i, \mathbf{x}_j) + v_{ff}(t) + o_i \quad (1)$$

where:

- P_{tx}^j is the nominal transmitted power and r_j is the *transmission offset* between the nominal and the effectively transmitted power. This factor is due to fabrication mismatches and it is assumed to be constant in time;
- $f_{pl}(\cdot)$ represents the *Path Loss* effect, modeled as (see [14]):

$$f_{pl}(d) = \beta - 10\gamma \log_{10}(d) \quad (2)$$

where $d = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the euclidean distance between the nodes i and j , β represents the radio receiver gain at a nominal distance of $d = 1m$, and γ is the loss factor (in an ideal outdoor setting $\gamma \approx 2$). The parameters β and γ are in general unknown since they depend on the specific environment where the WSN is placed;

- $f_{sf}(\cdot)$ takes into account the *Shadow Fading* and other slow fading components. It is assumed (see [15]) to be symmetric (i.e. $f_{sf}(\mathbf{x}_i, \mathbf{x}_j) = f_{sf}(\mathbf{x}_j, \mathbf{x}_i)$) and Gaussian with a spatial correlation dependent on the difference between the distances of the various points. More precisely, $\mathbb{E}_{\mathbf{x}}[f_{sf}(\mathbf{x}_i, \mathbf{x}_j)] = 0$ and $\mathbb{E}_{\mathbf{x}}[(f_{sf}(\mathbf{x}_i, \mathbf{x}_j))^2] = \sigma_{sf}^2$ are the spatial mean and variance where the expectation is performed w.r.t. to the random node positions. Moreover, let $\mathbf{x}_i, \mathbf{x}_j^a$ and $\mathbf{x}_i, \mathbf{x}_j^b$ two different configurations s.t. $\delta = \|\mathbf{x}_j^a - \mathbf{x}_j^b\|$, then the spatial correlation is:

$$\mathbb{E}_{\mathbf{x}} [(f_{sf}(\mathbf{x}_i, \mathbf{x}_j^a)) (f_{sf}(\mathbf{x}_i, \mathbf{x}_j^b))] = \sigma_{sf}^2 \rho_D^{\delta/D}$$

where ρ_D is a parameter and D is the typical correlation distance. Note that the expected value of f_{sf} is assumed to be zero;

- $f_a(\cdot)$ represents the *channel asymmetry* factor. It is due to non symmetric reflections, and we model it as a Gaussian r.v. with zero-mean and covariance $\mathbb{E}_{\mathbf{x}}[f_a^2(\mathbf{x}_i, \mathbf{x}_j)] = \sigma_a^2$;
- $v_{ff}(\cdot)$ represents the *fast fading* component that can be modeled (see [14]) as a white temporal noise with zero-mean and covariance $\mathbb{E}_t[v_{ff}^2(t)] = \sigma_{ff}^2$;
- $o_i(\cdot)$ represents the *offset that affects the measured received strength* of the receiving node due to fabrication mismatches in the radio chip. For example, in the case of the nodes used in our experimental testbed the RSSI sensor has a tolerance of ± 6 dB (see [5]).

Eqn. (1) is a general model for the wireless channel, in which parameters depend on the physical environment where the WSN is placed and on the sensors under consideration. It is important to remark that these parameters are not known in advance but they need to be estimated on-site. This is the objective of the rest of the paper. In the next section we describe the experimental testbed used to collect experimental data, we determine some of the numerical values for the terms in Eqn. (1) and we show that some of them are negligible. Then, in Section 5 we estimate the offsets o_i , and in Section 6 we identify β and γ .

3 Experimental Testbed and Model Validation

The experimental data used in the simulations consist in a series of measurements relative to packet transmissions and receptions performed by a net of 25 Tmote-Sky [20] nodes equipped with the Chipcon CC2420 RF Transceiver [5] and powered by alkaline batteries. These nodes were randomly placed inside a single conference room of $15m \times 10m$ at about $50cm$ from the ground. The relative position of the nodes is shown in Figure 2.



Figure 1: Picture of the experimental testbed room: Aula Magna “A. Lepschy”, Dept. of Information Engineering, University of Padova.

Each node implemented the randomized broadcast communication using the same transmission power P_{tx} and intercommunication interval $\tau = 15s$ so that the event of a packet collision is negligible. Each node sent a fixed number of packets $M = 500$, each one including the sender node ID, and also stored a table with the total number of messages received from their neighbors and the corresponding RSSI measures P_{rx}^{ij} .

These tables were then collected for off-line data processing. In particular, from these data we constructed the connectivity matrix C . Given the short distance among nodes, each node received at least one packet from any other node, however the empirical packet reception probability was different. In fact, the c -connectivity graph \mathcal{G}_c obtained for $c = 0.75$ (i.e. removing poor links with showed an empirical packet loss probability greater than 25%) is not the complete graph, even if it is still connected, as shown in Figure 2.

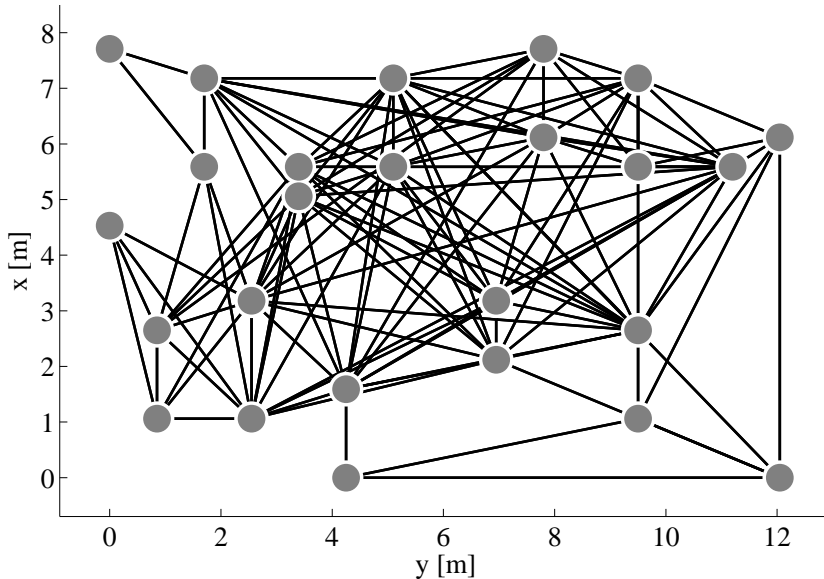


Figure 2: Network topology and node displacement of experimental testbed. Only edges with empirical packet loss smaller than 25% are displayed.

In the following we explain how it is possible to estimate the various parameters of the wireless channel model (1) using the various $P_{rx}^{ij}(t)$ collected from the nodes.

The transmission power offsets r_j of Eqn. (1) can be directly measured substituting the antenna of the nodes with a probe connected to a power measurer. Measurements made on the set of the nodes used for the experimental data showed that these offsets are negligible (see [32]), so in the following we will ignore them, i.e. we set $r_i = 0, \forall i$.

Then for every link $(i, j) \in \mathcal{E}$ in the connectivity graph, we compute the empirical mean of the received power $\bar{P}_{rx}^{ij} = \frac{1}{M_{ij}} \sum_t P_{rx}^{ij}(t)$, and the empirical variance $(\hat{\sigma}_{ff}^{ij})^2 = \frac{1}{M_{ij}} \sum_t (P_{rx}^{ij}(t) - \bar{P}_{rx}^{ij})^2$, where M_{ij} is the total number of messages received. The empirical variance around each link is due to fast fading, therefore, the estimate for the fast fading variance is:

$$\sigma_{ff}^2 = \frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} (\sigma_{ff}^{ij})^2.$$

The measurements \bar{P}_{rx}^{ij} include the effects of path loss, shadow-fading, channel asymmetry and reception offsets. We can try to identify first the contribution of the channel asymmetry and reception offset by noting that the path loss and the shadow fading are symmetric, i.e. $\Delta \bar{P}_{rx}^{ij} = \bar{P}_{rx}^{ij} - \bar{P}_{rx}^{ji} = f_a^{ij} - f_a^{ji} + o_i - o_j$, where for ease of notation $f_a^{ij} = f_a(\mathbf{x}_i, \mathbf{x}_j)$. We can also remove the effects of the offsets by noting that $\Delta \bar{P}_{rx}^{ijk} = \Delta \bar{P}_{rx}^{ij} + \Delta \bar{P}_{rx}^{jk} + \Delta \bar{P}_{rx}^{ki} = f_a^{ij} - f_a^{ji} + f_a^{jk} - f_a^{kj} + f_a^{ki} - f_a^{ik}$. We experimentally observed that $\Delta \bar{P}_{rx}^{ijk}$ has approximately zero-mean over the set of all the independent feasible cycles (i, j, k) , set that we denote with \mathcal{C} . Since the nodes are sufficiently far from each other and we have experimentally observed that the shadow fading correlation distance $D \approx 10cm$, all f_a^{ij} can be considered uncorrelated, therefore we can compute the covariance of the channel asymmetry as:

$$\sigma_a^2 = \frac{1}{6|\mathcal{C}|} \sum_{(i,j,k) \in \mathcal{C}} (\Delta \bar{P}_{rx}^{ijk})^2.$$

If we also assume independence between channel asymmetry components f_a^{ij} and the offsets o_i , we can estimate the offset variance σ_o^2 from the following formula:

$$2\sigma_o^2 + 2\sigma_a^2 = \frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} (\Delta \bar{P}_{rx}^{ij})^2.$$

Finally, we can estimate the parameters $\theta = [\beta \ \gamma]^T$ of the path loss channel. As it will be shown in the next section, it is possible to calibrate sensors by adding a compensating offset \hat{o}_i such that $o_i + \hat{o}_i = \alpha$ for all nodes. Averaging all sensor readings received from the same node removes the effect of fast-fading, therefore the calibrated average received power $\hat{P}_{rx}^{ij} = \bar{P}_{rx}^{ij} + \hat{o}_i$ is given by:

$$\hat{P}_{rx}^{ij} = P_{tx} + \beta - 10\gamma \log(d_{ij}) + f_{sf}^{ij} + f_a^{ij} + \alpha$$

where $f_{sf}^{ij} = f_{sf}(\mathbf{x}_i, \mathbf{x}_j)$. Since β needs to be estimated and α is constant, we can assume w.l.o.g. that $\alpha = 0$, since its contribution will be included in the estimated β . Shadow fading f_{sf}^{ij} and channel asymmetry f_a^{ij} are unknown but they can be assumed to be independent zero-mean disturbances, therefore it is possible to find the best mean square estimate of the unknown parameter as $\hat{\theta}_{LS} = (A^T A)^{-1} A^T b$, where $A = [a_1 \dots a_M]^T$, $b = [b_1 \dots b_M]$, and $M = |\mathcal{E}|$. The generic elements of matrix A and vector b are $a_m = [1 \ -10\log(d_{ij})]^T$ and $b_m = (\hat{P}_{rx}^{ij} - P_{tx})$, where $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ and \hat{P}_{rx}^{ij} are known. Figure 3 shows the identified path-loss model and all collected pairs $(\hat{P}_{rx}^{ij}, d_{ij})$. The residues obtained from the path-loss model correspond to the variance due to the shadow fading and channel asymmetry, i.e.:

$$\sigma_a^2 + \sigma_{sf}^2 = \frac{1}{|\mathcal{E}|} \|A\hat{\theta}_{LS} - b\|^2.$$

Table 1 summarizes the estimated parameters of the model (1) based on the experimental data collected. Note that the terms due to the asymmetry in the channel, f_a^{ij} , can be safely neglected when compared to the slow fading terms, f_{sf}^{ij} , i.e. the wireless channel is indeed symmetric.

β [dBm]	γ [dBm]	σ_{sf} [dBm]	σ_a [dBm]	σ_{ff} [dBm]	σ_o [dBm]	r_i [dBm]
-45.7	1.76	3.78	0.16	1.31	1.01	≈ 0

Table 1: Results of the estimation of the channels parameters of the model (1) via the centralized estimation strategies presented in Section 3.

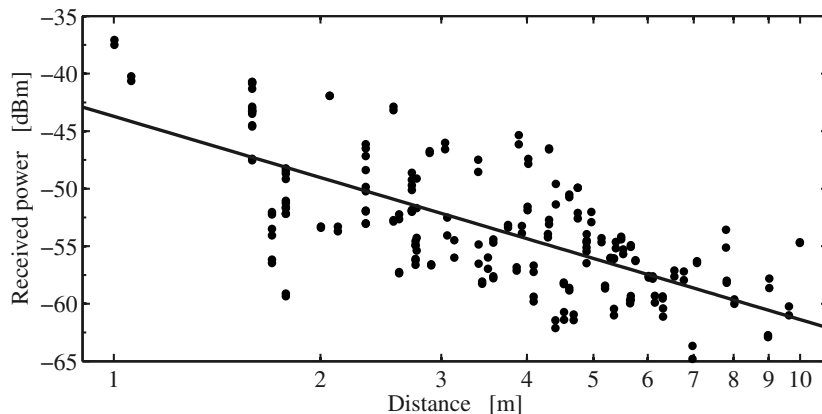


Figure 3: Estimated path-loss model for the wireless channel of the experimental environment using the standard centralized mean square estimate. The continuous line represents the path-loss function, while the dots are the measures experimentally collected.

4 Distributed Algorithms on WSNs

In this section we recall some useful results about consensus algorithms which we will use in the following sections to develop a distributed algorithm suitable for wireless sensor networks. In particular they will be used to estimate the sensor offsets o_i (Section 5) and to estimate the channel parameters β and γ (Section 6).

Let us consider the following linear update equations:

$$x(t+1) = Q(t)x(t) \quad (3)$$

where $x \in \mathbb{R}^N$ and $Q \in \mathbb{R}^{N \times N}$ is a *stochastic matrix*, i.e. $[Q]_{ij} = q_{ij} \geq 0$ and $\sum_{j=1}^N q_{ij} = 1, \forall i$, i.e. each row sums to unity. A stochastic matrix Q is said *doubly stochastic* if also $\sum_{i=1}^N q_{ij} = 1$, i.e. each column sums to unity. Clearly if a stochastic matrix is symmetric then it is also doubly stochastic. We also define the graph associated with the matrix Q to be $\mathcal{G}_Q = (\mathcal{N}, \mathcal{E}_Q)$, where the nodes $\mathcal{N} = \{1, 2, \dots, N\}$ and $\mathcal{E}_Q = \{(i, j) \mid q_{ij} > 0\}$. We also say that a matrix Q is *compatible* with the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, denoted with $Q \sim \mathcal{G}$, if its associated graph $\mathcal{G}_Q = (\mathcal{N}, \mathcal{E}_Q)$ is such that $\mathcal{E}_Q \subseteq \mathcal{E}$. We denote with \mathbb{G}_{sa} the set of graphs which include all self-arcs, i.e. $\mathcal{G} \in \mathbb{G}_{sa}$ iff $(i, i) \in \mathcal{E}, \forall i \in \mathcal{N}$.

A graph is *rooted in k* if there exists a tree embedded in the graph and rooted at node $k \in \mathcal{N}$ which spans all nodes, and *strongly rooted in k* if node k is directly connected to all other nodes, i.e. $(j, k) \in \mathcal{E}, \forall j \in \mathcal{N}$. A graph is *strongly connected* if there is a path from any node to any other node in the graph. Clearly a strongly connected graph implies that it is also a rooted graph for any node. A graph is *complete* if $(i, j) \in \mathcal{E}, \forall i, j \in \mathcal{N}$. The *concatenation* of two graphs $\mathcal{G}_1 = (\mathcal{N}, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{N}, \mathcal{E}_2)$ is a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}) = \mathcal{G}_2 \circ \mathcal{G}_1$ such that $(i, j) \in \mathcal{E}$ if there exists $k \in \mathcal{N}$ such that $(k, j) \in \mathcal{E}_1, (i, k) \in \mathcal{E}_2$. Similarly, the *union* of two graphs is a graph $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$ for which $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$. Clearly $\mathcal{G}_1 \cup \mathcal{G}_2 = \mathcal{G}_2 \cup \mathcal{G}_1$ and $\mathcal{G}_2 \circ \mathcal{G}_1 \neq \mathcal{G}_1 \circ \mathcal{G}_2$.

Let us consider the following definitions:

Definition 1. *Let us consider Eqn. (3). We say that $Q(t)$ solves the **consensus problem** if $\lim_{t \rightarrow \infty} x^i(t) = \alpha, \forall i = 1, \dots, N$, where $x^i(t)$ is the i -th component of the vector $x(t)$. We say that $Q(t)$ solves the **average consensus problem** if in addition to the previous condition we have $\alpha = \frac{1}{N} \sum_{i=1}^N x^i(0)$. If $Q(t)$ is a random variable, then we say that Q solves the **probabilistic (average) consensus problem** if the limit above exists almost surely.*

This definitions includes a wide class of consensus strategies: strategies with a time-invariant matrix $Q(t) = Q$, deterministic time-varying strategies $Q(t)$, and randomized strategies where $Q(t)$ is drawn from some distributions on a set of stochastic matrices \mathcal{Q} . The next theorems describes some sufficient conditions which guarantee deterministic and probabilistic (average) consensus.

Theorem 1. *Let us consider the sequence of constant matrices $Q(t) = Q$. If the graph $\mathcal{G}_Q \in \mathbb{G}_{sa}$ and is rooted, then Q solves the consensus problem. If in addition Q is doubly stochastic, then \mathcal{G}_Q is strongly connected and Q solves the average consensus problem. Moreover, the convergence rate in both cases is exponential and it is given by second largest eigenvalue in absolute value of the matrix Q .*

Theorem 2. *Let us consider a deterministic sequence of stochastic matrices $\{Q(t)\}_{t=0}^{+\infty}$ and the corresponding associated graphs $\mathcal{G}(t) = \mathcal{G}_{Q(t)}$. Suppose $\mathcal{G}(t) \in \mathbb{G}_{sa}, \forall t$. Suppose also there exists a finite positive integer number T such that the graphs $\mathcal{G}'(\cdot)$ obtained from the composition of the graphs $\mathcal{G}(\cdot)$ in the following way: $\mathcal{G}'(\tau) = \mathcal{G}(\tau \cdot T) \circ \mathcal{G}(\tau \cdot T + 1) \circ \dots \circ \mathcal{G}(\tau \cdot T + T - 1)$ with $\tau = 0, 1, \dots$ are all rooted. Then the sequence $Q(t)$ solves the consensus problem. If the matrices $Q(t)$ are all doubly stochastic, then they solve the average consensus problem.*

Theorem 3. *Let us consider a random i.i.d. sequence of stochastic matrices $\{Q(t)\}_{t=0}^{+\infty}$ drawn according to some distribution from the set \mathcal{Q} , and the stochastic matrix $\bar{Q} = \mathbb{E}[Q(t)]$. If the graphs $\mathcal{G}(t) = \mathcal{G}_{Q(t)} \in \mathbb{G}_{sa}, \forall t$ and if $\mathcal{G}_{\bar{Q}}$ is rooted, then the sequence $Q(t)$ solves the probabilistic consensus problem. If in addition $Q(t)$ are all doubly stochastic, then they solve the probabilistic average consensus problem.*

The first theorem is concerned with constant consensus matrix and shows how convergence conditions can be reframed as a graph problem which is easy to verify [7]. There are also algorithms based on convex optimizations that, given a symmetric graph, find a doubly stochastic matrix Q compatible with the graph which maximize the rate of convergence [31]. The second theorem focuses on deterministic time-varying consensus algorithms and shows that it is not necessary for the communication graph to be connected at any iteration but over a fixed time window [4, 19]. The last theorem addresses the consensus problem in

a probabilistic context that arises from randomized communication strategies [3, 11] or networks subject to random external disturbances, such as link or node failure [12].

We now provide some details of the consensus algorithm in the framework of WSN and wireless communication. In particular, the update matrix Q takes the following forms for the different wireless communication strategies that have been presented in Section 2.1.

In the broadcast strategy the consensus matrix Q_i^B (the superscript “ B ” stands for broadcast) when a node i transmits is given by (note that $i \notin \mathcal{V}(i)$):

$$[Q_i^B]_{mn} = \begin{cases} 1 & \text{if } m = n \notin \mathcal{V}(i) \\ 1 - w & \text{if } m = n \in \mathcal{V}(i) \\ w & \text{if } m \in \mathcal{V}(i), n = i \\ 0 & \text{otherwise} \end{cases}$$

where $w \in (0, 1)$ is a tuning parameter and often $w = \frac{1}{2}$. In the symmetric gossip, when the edge (i, j) is selected, the consensus matrix Q_{ij}^G (the superscript “ G ” stands for gossip) is given by:

$$[Q_{ij}^G]_{mn} = \begin{cases} 1 & \text{if } m = n \neq j \text{ and } m = n \neq i \\ 1 - w & \text{if } m = n = j \text{ or } m = n = i \\ w & \text{if } (m, n) = (i, j) \text{ or } (m, n) = (j, i) \\ 0 & \text{otherwise} \end{cases}$$

The consensus matrices defined above are based on the assumption that there is no link failure during the communication. In the following we assume that the communication graph \mathcal{G} coincides with the c -connectivity graph \mathcal{G}_c defined in Section 2.1. When a link (i, j) fails in broadcast communication, the matrix Q_i^B needs to be modified with $[Q_i^B]_{jj} = 1, [Q_i^B]_{ji} = 0$. Instead, when it happens in symmetric gossip, there is no communication at all and then no update is performed, i.e. $Q_{ij}^G = I$ is the identity matrix. Note that in broadcast we have $Q_{ij}^B = I$ if all links fail.

Based on the randomized communication modeling with link failure probability, it is possible to show that the expected consensus matrix $\bar{Q}^B = \mathbb{E}[Q^B(t)]$ generated for the broadcast strategy is given by:

$$[\bar{Q}^B]_{mn} = \begin{cases} 1 - \frac{c \cdot w \cdot d(n)}{N} & \text{if } m = n \\ \frac{c \cdot w}{N} & \text{if } m \in \mathcal{V}(n) \\ 0 & \text{otherwise} \end{cases}$$

Note that $\bar{Q}^B = (\bar{Q}^B)^T$ is symmetric and hence doubly stochastic, although the matrices Q_i^B are never symmetric. Moreover $\mathcal{G}_{\bar{Q}^B} = \mathcal{G}_c$, i.e. the graph associated with the expected consensus matrix \bar{Q}^B coincides with the underlying communication graph \mathcal{G}_c . Therefore, if \mathcal{G}_c is strongly connected, then this implies that the randomized broadcast guarantees probabilistic consensus although it does not guarantee average consensus for all possible realizations of $Q^B(t)$. Even if the gossip matrices are not doubly stochastic, the *expected* consensus matrix \bar{Q}^B is doubly stochastic, therefore the elements converge to the average of the initial conditions in mean sense. One might also wonder if \bar{Q}^B provides some information about convergence rate for the randomized strategy. In [11] there is an extensive analysis of rates of convergence and mean square analysis for the dispersion of final consensus value w.r.t. the average of initial conditions. The main message being that the second largest eigenvalue of \bar{Q}^B provides only an optimistic rate of convergence, and that the dispersion of the final consensus value from the average of the initial conditions decreases as the number of nodes increases. As we will see in Section 6.1, the parameter w can be tuned to obtain a final consensus value closer to the average of the initial conditions, at the price of slower convergence rate. This trade-off has been observed and studied also in [10].

Similarly, the expected consensus matrix \bar{Q}^G for the symmetric gossip is given by:

$$[\bar{Q}^G]_{mn} = \begin{cases} 1 - \frac{\sum_{i \in \mathcal{V}(n)} \frac{2c \cdot w}{N(d(n) + d(i))}}{N(d(m) + d(n))} & \text{if } m = n \\ \frac{2c \cdot w}{N(d(m) + d(n))} & \text{if } (m, n) \in \mathcal{E}_c, m \neq n \\ 0 & \text{otherwise} \end{cases}$$

Obviously $\bar{Q}^G = (\bar{Q}^G)^T$ since all the gossip matrices Q_{ij}^G from which the distribution is drawn are symmetric by construction. Similarly to the broadcast, we have $\mathcal{G}_{\bar{Q}^G} = \mathcal{G}_c$. Therefore, if \mathcal{G}_c is strongly connected, then the randomized symmetric gossip guarantees probabilistic average consensus. Compared to the randomized broadcast, the randomized symmetric gossip guarantees average consensus for all realizations, but it is more expensive from a communication point of view. Indeed, at least two packets

with reception acknowledge need to be exchanged at every step of the consensus iteration, while for the broadcast only one packet is transmitted and no reception acknowledge is needed. Moreover, the rate of converge is much slower, as can be guessed by the observation that the off-diagonal elements of the matrix \bar{Q}^G are smaller than their counterparts in \bar{Q}^B , i.e. there is slower information propagation. We will discuss these differences in more details in Section 6.1.

5 Consensus-Based Sensors Calibration

Experimental evidence indicates that sensor offsets o_i in the nodes are not negligible and can be substantially large for some node (up to 6dB). The effect of this offset is to bias the estimate of the distance between two nodes, which is particularly harmful in tracking application. In fact, if one node has a high offset o_i , then its estimated distance from a moving node is smaller than the true distance. Since unknown location of a moving target is obtained, similarly to the GPS, by triangulating its position from three or more static nodes whose position is known, the estimated position will be closer to the node with high offset o_i than it should be. This is particularly clear in Figure 4, which reports a tracking experiment where the moving node to be tracked is following a straight line (the basketball court centerline) between two rows of nodes of a WSN. However, its estimated trajectory is not straight but it is bent to the left (left panel). When the two central nodes on one side are swapped with the other side, the estimated trajectory is now bent to the right, thus clearly showing a problem due to uncalibrated offsets. Here we present a fully distributed and simple strategy which aims at estimating and removing the offsets o_i from each node, and we show its benefits on experimental data.

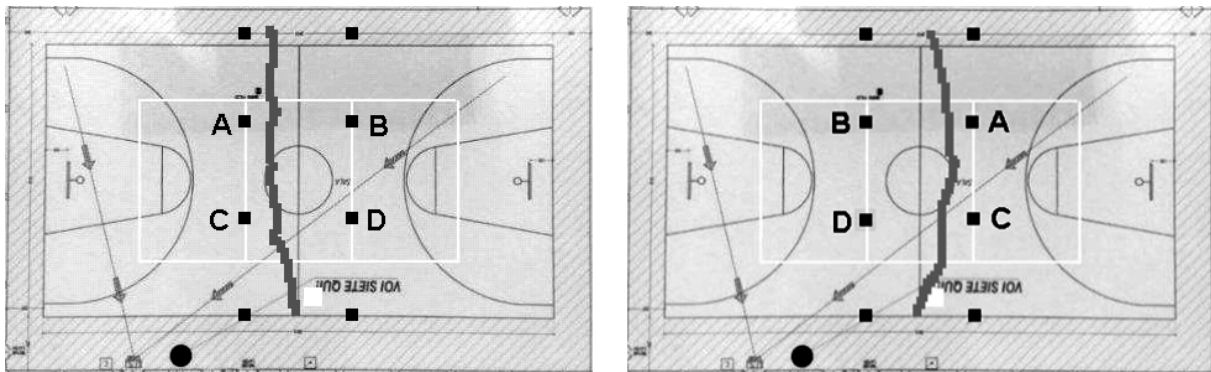


Figure 4: Experiment inside a basketball court showing the effects of reception offsets in WSN tracking when nodes are swapped. True trajectory in both panels is the court centerline. Courtesy of ST Microelectronics [25].

5.1 Offset calibration algorithm

Ideally, we would like to add to the reading of received power a compensation offset \hat{o}_i such that $o_i + \hat{o}_i = 0$, and then use the compensated received power $\bar{P}_{rx}^{ij} = P_{rx}^{ij} + \hat{o}_i$ to estimate the relative distance. However, we do not have the possibility to directly measure o_i of each node, nonetheless we would like to at least partially compensate it. More precisely, we would like to have

$$o_i + \hat{o}_i = \alpha, \quad \alpha \approx 0$$

for all nodes. If $\alpha \neq 0$ such strategy does not compensate the offset, but at least all nodes either underestimate or overestimate the relative distance similarly, therefore after a GPS-like triangulation stage these errors should partially cancel out. We now show how this strategy can be casted as a consensus problem. Let us consider a static WSN where the nodes are at fixed positions and transmit at the same power P_{tx} . Let y_{ij} be the average received signal strength by node i from node j :

$$\bar{P}_{rx}^{ij} = \frac{1}{T} \sum_{t=1}^T P_{rx}^{ij}(P_{tx}, \mathbf{x}_i, \mathbf{x}_j, i, j, t) = f_{ij} + o_i + f_{ij}^a + r_j + \frac{1}{T} \sum_{t=1}^T v_{ff}(t) \approx f_{ij} + o_i \quad (4)$$

where P_{rx}^{ij} is modeled as in Eqn. (1), $f_{ij} = P_{tx} + f_{pl}(\|\mathbf{x}_j - \mathbf{x}_i\|) + f_{sf}(\mathbf{x}_j, \mathbf{x}_i)$, and $f_{ij}^a = f_a(\mathbf{x}_i, \mathbf{x}_j)$. The approximation is based on parameters in Table 1 which imply that $|f_{ij}^a + r_j + \frac{1}{T} \sum_{t=1}^T v_{ff}(t)| \ll |o_i|$ for T

sufficiently large, $v_{ff}(t)$ being white noise. Note that f_{ij} is symmetric, i.e. $f_{ij} = f_{ji}$. The next theorem shows how the problem of compensating the offset o_i can be casted as a consensus problem:

Theorem 4. *Let us consider the c -connectivity graph $\mathcal{G}_c = (\mathcal{N}, \mathcal{E}_c)$ of a WSN, and let $Q(t) \sim \mathcal{G}_c$ a sequence of stochastic matrices that solves the (probabilistic) consensus problem. Assume that $y_{ij} = f_{ij} + o_i$ where $f_{ij} = f_{ji}$. Consider the following algorithm:*

$$\hat{o}_i(0) = 0, \quad i \in \mathcal{N} = \{1, \dots, N\} \quad (5)$$

$$\hat{o}_i(t+1) = \hat{o}_i(t) + \sum_{j \in \mathcal{V}(i)} q_{ij}(t)(y_{ji} - y_{ij} + \hat{o}_j(t) - \hat{o}_i(t)) \quad (6)$$

where $q_{ij}(t) = [Q(t)]_{ij}$. Then $\lim_{t \rightarrow \infty} o_i + \hat{o}_i(t) = \alpha$ where $\alpha \in [\min_i(o_i), \max_i(o_i)]$. If in addition $Q(t)$ is doubly stochastic $\forall t$, then $\alpha = \frac{1}{N} \sum_{i \in \mathcal{N}} o_i$.

Proof. Let us define the new variables $x_i(t) = o_i + \hat{o}_i(t)$. From this definition it follows that $x_i(0) = o_i + \hat{o}_i(0) = o_i$. Moreover, Eqn. (6) can be rewritten as follows:

$$\begin{aligned} \hat{o}_i(t+1) + o_i &= \hat{o}_i(t) + o_i + \sum_{j \in \mathcal{V}(i)} q_{ij}(t)(f_{ji} + o_j - f_{ij} - o_i + \hat{o}_j(t) - \hat{o}_i(t)) \\ x_i(t+1) &= x_i(t) + \sum_{j \in \mathcal{V}(i)} q_{ij}(t)(x_j(t) - x_i(t)) \\ &= \left(1 - \sum_{j \in \mathcal{V}(i)} q_{ij}(t)\right)x_i(t) + \sum_{j \in \mathcal{V}(i)} q_{ij}(t)x_j(t) \\ &= q_{ii}(t)x_i(t) + \sum_{j \in \mathcal{V}(i)} q_{ij}(t)x_j(t) \end{aligned}$$

The last equation can be written in compact form as $x(t+1) = Q(t)x(t)$. Since $Q(t)$ solves the (probabilistic) consensus problem, then $\lim_{t \rightarrow \infty} x_i(t) = \alpha$. The claim that $\alpha \in [\min_i(o_i), \max_i(o_i)]$ follows from the property that if Q is a stochastic matrix, then $\max(Qx) \leq \max_i(x)$ and $\min(Qx) \geq \min(x)$ [9]. \square The previous theorem indicates how we can compensate the offsets o_i without knowing their values. Also, it is not necessary to know the exact value of f_{ij} since it is symmetric. In practice the assumption $\bar{P}_{rx}^{ij} = y_{ij} = f_{ij} + o_i$ is not exact, therefore, as we will show shortly, this leads to an oscillating steady state behavior in the consensus algorithm.

We might wonder whether there is an appropriate choice of $Q(t)$ to have $\alpha \approx 0$, which is the ideal solution. We can argue that the offsets o_i of the radio chips are on average null, have some dispersion due to imperfect fabrication and are independent, i.e. $\mathbb{E}[o_i] = \mu_o = 0$, $\mathbb{E}[o_i^2] = \sigma_o^2$, and $\mathbb{E}[o_i o_j] = \mathbb{E}[o_i]\mathbb{E}[o_j] = 0$. It is well known that the best estimate of the mean μ_o given a set of offsets is $\mathbb{E}[\mu_o | o_1, \dots, o_N] = \frac{1}{N} \sum_{i \in \mathcal{N}} o_i = \alpha^*$ which has the property that $\mathbb{E}[\alpha^*] = \mu_o = 0$ and $\mathbb{E}[(\alpha^*)^2] = \frac{\sigma_o^2}{N}$, i.e. the average consensus is the strategy for which α is closer to zero in mean square sense and its error becomes smaller and smaller as the number of nodes N increases.

Although the best choice for the compensation of offsets o_i is to choose doubly stochastic $Q(t)$'s, this can be difficult to be enforced in a WSN, since it requires synchronization among the nodes and compensation for packet loss. However it is not necessary to enforce the average consensus since the nonzero offset α reached after the calibration phase is completely absorbed during the identification of the path loss model parameter β .

5.2 Simulations based on experimental data

The proposed algorithm for distributed offset calibration has been tested off-line on the same set of data collected during the experimental setup described in Section 3. Here we considered the c -connectivity graph \mathcal{G}_c with $c = 0.1$, i.e. we considered all links which received at least 10% of the packets. Differently from the graph with $c = 0.75$ shown in Figure 2, the resulting graph with $c = 0.1$ reported in Figure 5 is complete, i.e. all edges exist. The set of all the edges has been divided into two subsets: the first subset of edges (60% of the total edges, in black in Figure 5) has been used for the estimation of the node offsets. Therefore the proposed distributed sensor calibration algorithm has been executed on the data collected on these edges. In particular, the calibration algorithm was set with $y_{ij} = \bar{P}_{rx}^{ij}$ corresponding to these edges. The second subset (40% of the total edges, in grey in Figure 5) has been used in a second stage for

validation purposes: we evaluated the asymmetric difference $(\bar{P}^{ij} + \hat{o}_i) - (\bar{P}^{ji} + \hat{o}_j)$ on the data collected on this subset. This approach allows us to both evaluate the effect of the offset removal in a rigorous way, and to validate at the same time the model we proposed.

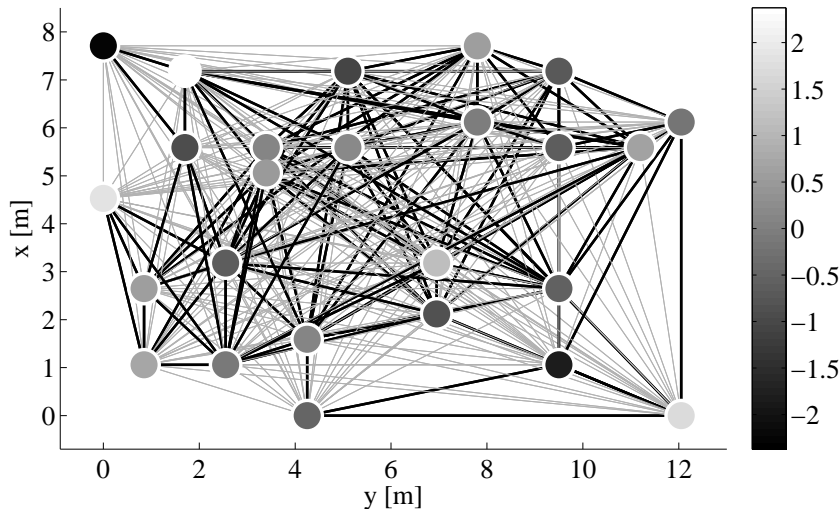


Figure 5: Network topology and node displacement for c -connectivity graph for $c = 0.1$. Nodes' grey intensity represents the estimated offset \hat{o}_i after calibration. Black and grey edges represent the edges used for training and validation data sets, respectively.

We simulated the randomized broadcast consensus described in Section 4 on the graph \mathcal{G}_c using the experimental data and including i.i.d. packet loss failure set by the connectivity matrix C . Figure 6 shows the behavior of the consensus algorithm for a specific realization with two different values of the weight parameter in matrices $Q(t)$. The steady state compensation offsets $\hat{o}_i(\infty)$ are displayed in Figure 5 where the gray intensity of the nodes is proportional to $\hat{o}_i(\infty)$. Since the true node offsets o_i are unknown it is not possible to plot the behavior of $x_i(t) = o_i + \hat{o}_i(t)$ which are the variables that should converge to a common value, however the fact that all \hat{o}_i converge to a steady state is an indication of correct functioning. It is also interesting to note the effect of unmodeled measurement noise arising from having neglected channel asymmetry and fast fading. In fact for larger w , i.e. for larger weight on the off-diagonal terms in the consensus matrix, the oscillation at steady state is not negligible, i.e. a large w tends to amplify noise. On the other hand, a small w leads to slower rate of convergence, thus indicating a tradeoff between convergence rate and noise sensitivity. Note also that the magnitude of steady state values of \hat{o}_i is consistent with the a-priori dispersion indicated by the standard deviation σ_o reported in Table 1.

In order to evaluate the effectiveness of the offset calibration, we tested the channel asymmetry after calibration on a validation set different from the set used for computing the offsets \hat{o}_i . The results of this second stage has been plotted in Figure 7. The white bars represent the distribution of $|\bar{P}^{ij} - \bar{P}^{ji}|$ on the validation edges, before the distributed sensor calibration algorithm is executed. The black bars, instead, show the distribution of $|(\bar{P}^{ij} + \hat{o}_i) - (\bar{P}^{ji} + \hat{o}_j)|$ after the algorithm has run. The offset reduction clearly appears. After the calibration, 56% of the validation edges have an asymmetric difference smaller than $0.5dBm$ (it was 24% before calibration), while 88% of them have an absolute error smaller than $1dBm$ (it was 50% before calibration). After the offset removal algorithm, almost all the measurements (99.4% of them) are affected by an asymmetric error smaller than $2dBm$.

The importance of offset removal in the received power measurements is evident when these measurements are used for wireless-based localization. In fact, relative distance is estimated by inverting the path-loss function based on the calibrated measured power \hat{P}_{rx}^{ij} , i.e. $\hat{d}_{ij} = 10^{\frac{P_{tx}^j - \hat{P}_{rx}^{ij} + \beta}{10\gamma}} = 10^{\frac{P_{tx}^j - P_{rx}^{ij} + \hat{o}_i + \beta}{10\gamma}}$. If the calibration offset \hat{o}_i is not included in the previous formula, there can be measurements errors up to $6dBm$ due to uncalibrated offsets, as Figure 7 suggests. In fact, a systematic calibration error of $6dBm$ corresponds to an uncertainty range from $0.9m$ to $4.4m$ when estimating the relative position of a node at $2m$, and to a practically useless estimation when the node is farther. An error of $1dBm$, on the other hand, corresponds to an error in the distance of only $28cm$ for a $2m$ long link, and to a $1.4m$ error when the node is at $10m$ distance.

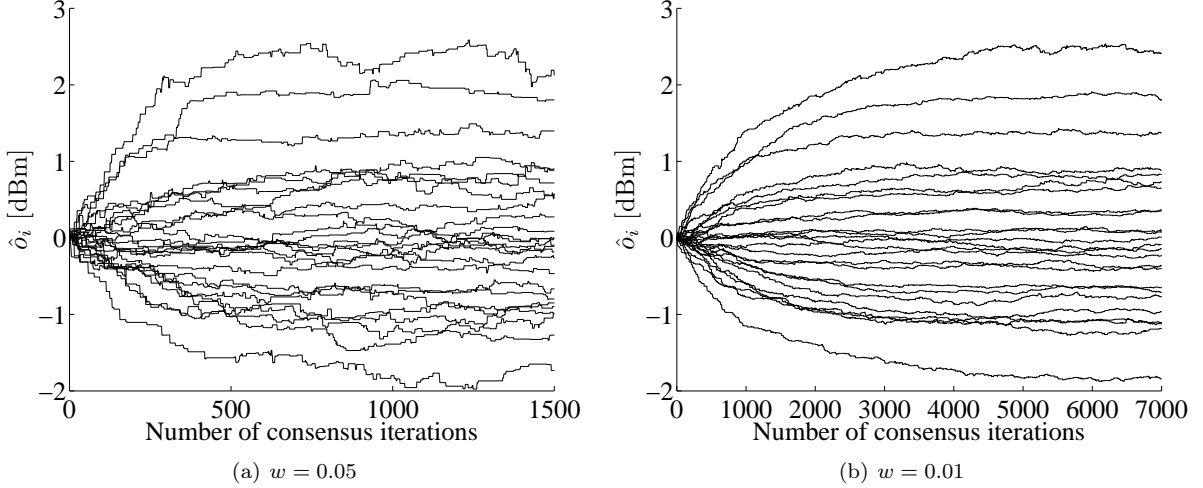


Figure 6: Offset estimation \hat{o}_i for each node of the considered WSN using randomized broadcast consensus for different values of the consensus weight w .

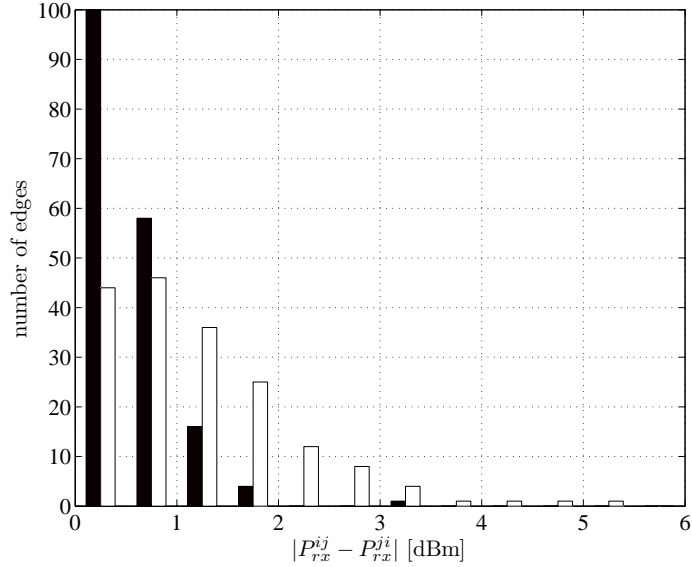


Figure 7: Asymmetric error distribution before ($|\bar{P}_{rx}^{ij} - \bar{P}_{rx}^{ji}|$, white) and after ($|\bar{P}_{rx}^{ij} + \hat{o}_i - \bar{P}_{rx}^{ji} - \hat{o}_j|$, black) distributed sensor calibration.

6 Consensus-Based Least Square Parameters Identification

A Least Square Parameter Identification (LSPI) problem arises when we have a data set $\mathcal{D} = \{(a_m, b_m), m = 1, \dots, M\}$ where $a_m \in \mathbb{R}^\ell$ and $b_m \in \mathbb{R}$. The data set is generated according to the model $a_m^T \theta = b_m + v_m$, where $\theta \in \mathbb{R}^\ell$ is a parameter vector to be estimated and $v_m \in \mathbb{R}$ is an unknown error. Let us define the matrix $A \in \mathbb{R}^{M \times \ell}$, $A = [a_1 \dots a_M]^T$ and the vectors $b, v \in \mathbb{R}^M$, $b = [b_1 \dots b_M]^T$, $v = [v_1 \dots v_M]^T$. The least square identification of the parameter θ is defined as follows:

$$\hat{\theta}_{LS} = \arg \min_{\theta} \|v\| = \arg \min_{\theta} \|A\theta - b\| = (A^T A)^{-1} A^T b$$

where we implicitly assumed that the matrix $A^T A$ is non singular. We now present a theorem showing how the centralized least square parameter identification can be performed over graphs.

Theorem 5. Let $\mathcal{G}_c = (\mathcal{N}, \mathcal{E}_c)$ be the c -connectivity graph associated to a communication network with N nodes, i.e. $N = |\mathcal{N}|$, and let $\mathcal{D}(i) = \{(a_m, b_m)\}$ the partition of the whole data set \mathcal{D} available to i -th node, satisfying $\mathcal{D}(i) \cap \mathcal{D}(j) = \emptyset, i \neq j$, $\cup_{i \in \mathcal{N}} \mathcal{D}(i) = \mathcal{D}$, $|\mathcal{D}(i)| = M_i$ and $|\mathcal{D}| = M = \sum_{i \in \mathcal{N}} M_i$.

Suppose that $Q(t)$ are stochastic matrices consistent with the graph, i.e. $Q(t) \sim \mathcal{G}_c, \forall t$, which solve the (probabilistic) average consensus problem. Let $x_i^A \in \mathbb{R}^{\ell \times \ell}$ and $x_i^b \in \mathbb{R}^\ell$ for $i = 1, \dots, N$ and consider the following algorithm:

$$x_i^A(0) = \sum_{m \in \mathcal{D}(i)} a_m a_m^T, \quad \forall i \in \mathcal{N} \quad (7)$$

$$x_i^b(0) = \sum_{m \in \mathcal{D}(i)} a_m b_m \quad (8)$$

$$x_i^k(t+1) = q_{ii}(t)x_i^k(t) + \sum_{j \in \mathcal{V}(i)} q_{ij}(t)x_j^k(t), \quad k = A, b \quad (9)$$

$$\eta_i(t) = (x_i^A(t))^{-1} x_i^b(t) \quad (10)$$

where $q_{ij}(t) = [Q(t)]_{ij}$. Then we have:

$$\lim_{t \rightarrow \infty} \eta_i(t) = \hat{\theta}_{LS}, \quad \forall i \in \mathcal{N}$$

Proof. Let us define the matrix $S = A^T A = \sum_{i=1}^M a_i a_i^T$ and the vector $d = A^T b = \sum_{i=1}^M a_i b_i$, therefore $\hat{\theta}_{LS} = S^{-1}d$. By construction we have that $\lim_{t \rightarrow \infty} x_i^A(t) = \frac{1}{N} \sum_{i=1}^N x_i^A(0) = \frac{1}{N} \sum_{i=1}^M a_i a_i^T = \frac{1}{N} S$ and similarly $\lim_{t \rightarrow \infty} x_i^b(t) = \frac{1}{N} \sum_{i=1}^N x_i^b(0) = \frac{1}{N} \sum_{i=1}^M a_i b_i = \frac{1}{N} d$. By continuity $\lim_{t \rightarrow \infty} \eta_i(t) = (\frac{1}{N} S)^{-1} \frac{1}{N} d = S^{-1}d = \hat{\theta}_{LS}$. Note that the sums are defined from 1 to M due to the structure of Equations (7) and (8). \square

This theorem shows that LSPI can be computed as the solution of a distributed algorithm which does not require the knowledge of the total number of nodes N or the total number of data M available. Moreover, the data can be arbitrarily partitioned among nodes. Since the matrix $S = A^T A$ is symmetric it is not necessary to compute all its ℓ^2 entries, therefore the x_i^A can be reduced to a vector of size $(\ell^2 + \ell)/2$. Nonetheless the complexity in terms of communication, i.e. the dimension of the vector of parameters to be averaged, is $O(\ell^2)$ which can be impractical if the dimension ℓ of the unknown parameter θ is large. We will show in the next section that there are strategies that trade-off accuracy in the identification of $\hat{\theta}_{LS}$ for a decrease in communication complexity to $O(\ell)$.

The problem addressed in the previous theorem belong to a more general class of problems that can be solved with consensus algorithms. In fact any optimization problem can be written as:

$$\xi = f \left(\frac{1}{N} \sum_{i \in \mathcal{N}} g_i(z_i) \right)$$

for some appropriate choice of functions f and g_i , where z_i represents the data available to node i . Some examples of the previous class of problem includes generalized means [1], χ -consensus [6] and distributed Kalman filter [28].

6.1 Simulations based on experimental data

In this section we apply the results presented in the previous section and identify in a distributed manner the unknown path-loss channel parameters (β, γ) given in equation (2) using different communication strategies. As mentioned above these two parameters are used in localization and target tracking algorithms in order to estimate relative distances between the moving node and the nodes of the static WSN. Therefore, it is critical to be able to identify the path-loss parameters in a distributed way, in a manner that is robust to node failure, with minimal exchange of data and low computational power, and without a central unit. It has to be noted that an accurate *a-priori* model for power loss in different indoor environments is almost unavailable (for example γ can vary from 1 to 6 according to the room sizes, the amount of furniture and people and the number of walls that the signal has to cross in average). Furthermore, the same environment can present a hourly or daily variation of these parameters due to the periodic presence of people populating the indoor spaces [8]. Fortunately, distributed algorithms with these features can be used to periodically or adaptively identify the channel parameters in a changing environment.

Based on these considerations, the focus of this section is to compare the performances of three different communication strategies which have different characteristics in terms of rate of convergence, communication complexity and parameter identification accuracy. The first and the second strategies are based on the implementation of the distributed least square identification described in Theorem 5 using the randomized broadcast and the randomized symmetric gossip, respectively. The third strategy performs the randomized symmetric gossip consensus on local estimates $\hat{\theta}_i$ of the channel parameters vector θ , rather than on the local least-square sufficient statistics (x^A, x^b) relative to $(A^T A, A^T b)$ of Theorem 5. Each strategy has its own advantages. In fact, the randomized symmetric gossip guarantees average consensus, therefore it is guaranteed to provide the best identification accuracy since it satisfies the hypotheses of Theorem 5. Randomized broadcast does not guarantee average consensus, and consequently the best performance, however it is very easy to implement since it needs no coordination between nodes. Moreover it is faster than the symmetric gossip since, on average, there are $d(i)$ updates per iteration compared with only 2 updates for the symmetric gossip. Finally, the strategy based on the average consensus of the local least-square estimates does not guarantee optimal performance nor best speed of convergence, however the number of parameters to be exchanged among nodes is equal to the size ℓ of the parameter vector θ , while for the first two strategies it is proportional to ℓ^2 .

We now describe in detail how the simulations are obtained. We considered the c -connectivity graph $\mathcal{G}_c = (\mathcal{N}, \mathcal{E}_c)$ for $c = 0.75$ shown in Figure 2. The data set $\mathcal{D}(i)$ available to each node i is given by $\mathcal{D}(i) = \{(\bar{P}_{rx}^{ij}, d_{ij}) \mid j \in \mathcal{V}(i)\}$, i.e. all the averaged received power measurements from each neighbor coupled with the corresponding relative distance (note that the distances are assumed to be known by the nodes). The data set of all measurements is indicated with $\mathcal{D} = \cup_{i \in \mathcal{N}} \mathcal{D}(i)$. We also assume that the offset calibration procedure of Section 5 has been performed in order to obtain the compensating offsets \hat{o}_i , and that the effect of fast-fading can be neglected since the measurements have been averaged over a large number of packets. Therefore, as shown at the end of Section 3, the channel parameters $\theta = [\beta \ \gamma]^T$ can be identified using a least square minimization by setting $a_m = [1 \ -10 \log(d_{ij})]$, $b_m = \bar{P}_{rx}^{ij} - P_{tx} + \hat{o}_i$, where $m = 1, \dots, M$ indicates a generic data element, and $M = |\mathcal{D}| = |\mathcal{E}_c|$. Using the same terminology of Theorem 5 we indicate with $\hat{\theta}_{LS}$ the centralized least-square estimate using the complete data set \mathcal{D} . We also indicate with $\hat{\theta}_{LS}^i$ the least-square estimate performed by the i -th node using only its data set $\mathcal{D}(i)$, which is the best estimate a node can have without communicating with the others. The performance (in terms of identification accuracy) is based on the residues of the estimate $\hat{\theta}$ given by:

$$J(\hat{\theta}) = \left\| A\hat{\theta} - b \right\|.$$

Note that A and b are constructed using the whole data set, and therefore $J(\hat{\theta})$ represents the global residual. Since $\hat{\theta}_{LS} = \arg \min_{\theta} J(\hat{\theta})$, it is obvious that $J(\hat{\theta}_{LS}) \leq J(\hat{\theta}_{LS}^i), \forall i$ from which it follows $J(\hat{\theta}_{LS}) \leq \frac{1}{N} \sum_{i \in \mathcal{N}} J(\hat{\theta}_{LS}^i)$. Being $\eta_i(0) = \hat{\theta}_{LS}^i$, if all $Q(t)$'s are doubly stochastic then from Theorem 5 it follows that $\lim_{t \rightarrow \infty} J(\eta_i(t)) = J(\hat{\theta}_{LS}), \forall i$, and so $\lim_{t \rightarrow \infty} \frac{1}{N} \sum_{i \in \mathcal{N}} J(\eta_i(t)) = J(\hat{\theta}_{LS})$.

In the first simulation, we tested the randomized broadcast least-square strategy using the connectivity matrix C defined in Section 2.1 for the link failure probabilities. Figure 8 shows the identified channel parameters of all nodes $\eta_i(t) = [\hat{\beta}_i(t) \ \hat{\gamma}_i(t)]^T$ as a function of the number of iterations for a typical realization of the system (thought as the stochastic process of information exchange). It can be seen that the identified parameters of all nodes converge to a common value, however, since broadcast does not guarantee average consensus, identified parameters do not necessarily coincide with the optimal estimate $\hat{\theta}_{LS}$. It is also interesting to note that most of the nodes have already good estimates of the parameters without communicating with the others, since most of them have lots of links and there are only two parameters to estimate. However, there are some nodes which have poor initial estimates, especially the ones on the perimeter of the graph which have few links. Nonetheless, thanks to the consensus algorithm, they rapidly converge to a good value.

In the second set of simulations, shown in Figure 9, we compared the rate of convergence and the steady state identification error for the three different strategies described above. More precisely, we compared the average estimation residual $\bar{J}(k) = \frac{1}{N} \sum_{i \in \mathcal{N}} J(\hat{\theta}_i(k))$ of all nodes as a function of iteration error. To reduce the randomness due to the choice of a particular realization of $\{P(t)\}_{t \in \mathbb{N}}$ we actually depicted $\mathbb{E}[\bar{J}(k)]$, approximately computed as the average of 50 independent extractions of the sequence $\{P(t)\}_{t \in \mathbb{N}}$. In Table 2 it is reported also the steady state dispersion of $\bar{J}(k)$ around its mean value, obtained by recording the maximum and the minimum value of $\bar{J}(k)$ over the 50 extractions. In the bottom line the residual of the centralized optimal estimate is also reported for comparison.

Initially we tested the randomized broadcast least square algorithm for two different weights w . As already mentioned, larger w leads to faster convergence rates, however it also leads, in mean, to a larger

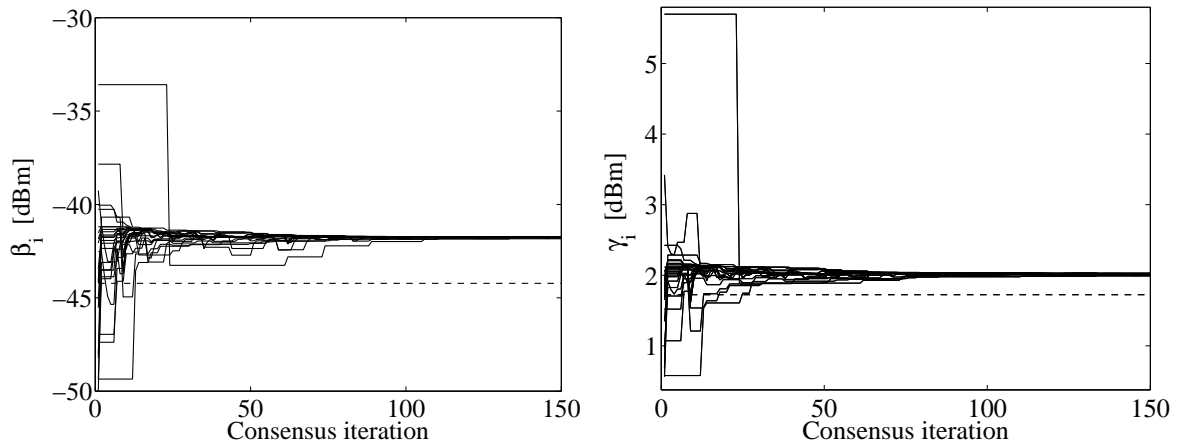


Figure 8: Convergence of parameter estimates β_i, γ_i using randomized broadcast least-square consensus and consensus weight $w = 0.5$. The dashed lines are the centralized least squares estimates $\hat{\beta}_{LS}, \hat{\gamma}_{LS}$.

Consensus Algorithm	$\mathbb{E}[J(\infty)]$	$\max J(\infty)$	$\min J(\infty)$
Broadcast $w = 0.5$	3.9816	4.1477	3.9320
Broadcast $w = 0.25$	3.9615	4.0919	3.9318
Symmetric Gossip	3.9307	3.9307	3.9307
Ave. of local estim.	3.9635	3.9635	3.9635

	$J_{Cent.L.S.}$
Centralized LS	3.9307

Table 2: Comparison of the mean estimation residual.

steady state identification error (see [10]). We also have that the steady state value is strongly realization dependent, as it can be noticed from the large dispersion interval. This is due to the fact the first communications tend to bias the final value toward the initial condition of that node. Differently, if w is reduced, then this bias is smoothed out and $\mathbb{E}[\bar{J}(k)]$ end up closer to exact average consensus. Also the dispersion of the single realization with respect to $\mathbb{E}[\bar{J}(k)]$ reduces. Moreover it has been proved in [11] that the distance of $\mathbb{E}[\bar{J}(k)]$ from the average consensus decreases by increasing the number of nodes in the network, thus suggesting fast convergence rate with negligible performance degradation as compared, for example, to random symmetric gossip.

The same Figure 9 also shows the performance of the randomized symmetric gossip least square algorithm. As expected, the rate of convergence is slower, but the final value converges to the minimum identification error given by the centralized least-square estimate $J(\hat{\theta}_{LS})$. We remark that all the single realizations tend to the exact optimal value, as shown by the fact that there is no dispersion around the mean value (Table 2), not only that $\mathbb{E}[\bar{J}(k)]$ tends to optimal value.

Finally, we tested also a randomized gossip algorithm that directly averages the local least-square estimates. As shown in Figure 9, this strategy has the same rate of convergence of the randomized symmetric gossip (which computes the exact centralized least-square solution), but a slightly worse performance. However, in terms of communication complexity this algorithm only requires the exchange of 2 parameters while the exact distributed least square one requires in this example the exchange of 4 parameters. It has to be noticed, though, that if the initial estimates were less reliable (for instance because the graph topology were much less connected) then the distributed least square would behave far better than the simple solution of an average of the local least squares estimations.

7 Conclusions

In this work we proposed consensus-based algorithms for wireless sensor networks and we successfully applied them to experimental data collected from a real WSN. In particular we applied these algorithms

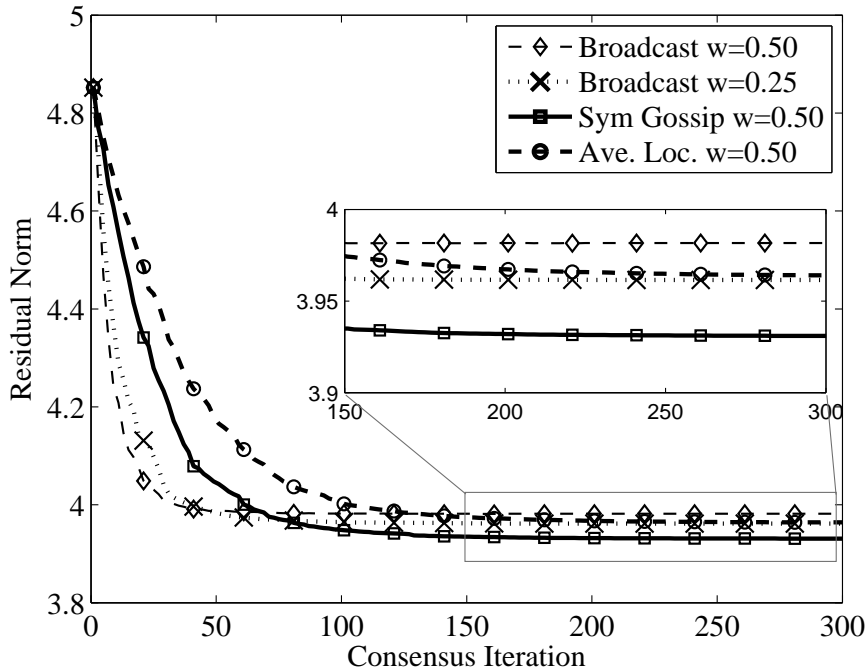


Figure 9: Comparison of the mean estimation residual $\mathbb{E}[\bar{J}(k)]$ for different randomized consensus algorithms. $\mathbb{E}[\bar{J}(k)]$ is approximately computed as the average of 50 independent extractions of the sequence $\{P(t)\}_{t \in \mathbb{N}}$.

to remove unknown offsets from the sensor measurements and to identify the parameters of the wireless channel for localization and tracking purposes.

However, these algorithms are rather general and can be applied in other fields and research areas. Indeed, we showed how it is possible to cast a wide class of problems into the consensus framework, such as problems in which the agents have to actually agree on a common estimate of few parameters (like in the least-square fitting), and problems in which every agent has to estimate its own parameter (like in the offset-removal algorithm). This duality deserves further investigation, possibly leading to a tighter relation between consensus algorithms [23] and asynchronous parallel iterative methods for the solution of system of linear equations [2].

The field of application of the proposed solution are definitely wider than the few applications presented in this work. For example, the offset removal algorithm could also be used to detect malfunctioning sensors by observing the magnitude of the compensation offset $\hat{\delta}_i$, while the least square square parameter identification algorithm can be used to identify any model parameter which is linear in the data.

Many issues remain to be explored, in particular in terms of correctly modeling real WSNs. For example we shown that although the optimal solution to some problems depends on the average of the initial conditions, there are algorithms which do not guarantee convergence to the average, nonetheless providing good performances. Therefore, there is a definite need to better understand the trade-offs between performance, rate of convergence, communication complexity and noise sensitivity for different consensus strategies on real WSNs. Another important research avenue is the formulation of possibly nonlinear or non-standard problems into standard consensus problems, which is by no mean trivial.

Acknowledgements: the authors would like to acknowledge the reviewers for their precious advices, and Giovanni Zanca, Francesco Zorzi and the rest of the SIGNET group of the University of Padova for the discussions about the wireless channel modeling and the data support.

References

- [1] D. Bauso, L. Giarré, and R. Pesenti. Nonlinear protocols for optimal distributed consensus in networks of dynamic agents. *Systems and Control Letters*, 55:918–928, 2006. 13
- [2] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989. 2, 16
- [3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory/ACM Transactions on Networking*, 52(6):2508–2530, June 2006. 8
- [4] M. Cao, A. S. Morse, and B. D. O. Anderson. Reaching a consensus in a dynamically changing environment: A graphical approach. *SIAM Journal on Control and Optimization*, 47:575–600, February 2008. [Online] Available at <http://www.eng.yale.edu/controls/pending/flockp1.pdf>. 7
- [5] Chipcom AS. *SmartRF CC2420 Datasheet, rev. 1*, November 2003. 4
- [6] J. Cortés. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44:726–737, 2008. 13
- [7] O.L.V. Costa, M.D. Fragoso, and R.P. Marques. *Discrete-Time Markov Jump Linear Systems (Probability and its Applications)*. Springer, 2004. 7
- [8] M. Dominguez-Duran, D. Claros, C. Urdiales, and F. Coslado. Dynamic calibration and zero configuration positioning system for wsn. *Electrotechnical Conference, 2008. MELECON 2008. The 14th IEEE Mediterranean*, pages 145–150, May 2008. 13
- [9] J. L. Doob. *Stochastic Processes*. John Wiley & Sons, Inc., New York, 1953. 10
- [10] F. Fagnani and S. Zampieri. Randomized consensus algorithms over large scale networks. *Information Theory and Applications Workshop*, pages 150–159, February 2007. 8, 15
- [11] F. Fagnani and S. Zampieri. Randomized consensus algorithms over large scale networks. *Selected Areas in Communications, IEEE Journal on*, 26(4):634–649, May 2008. 8, 15
- [12] F. Fagnani and S. Zampieri. Average consensus with packet drop communication. *SIAM J. on Control and Opt.*, to appear. 8
- [13] Andreas Frommer and Daniel B. Szyld. On asynchronous iterations. *J. of Computation and Applied Mathematics*, 123:201–216, 2000. 2
- [14] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005. 2, 3, 4
- [15] M. Gudmundson. Correlation model for shadow fading in mobile radio systems. *Electronics Letters*, 27(23):2145–2146, November 1991. 3, 4
- [16] Lingxuan Hu and David Evans. Localization for mobile sensor networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 45–57, New York, NY, USA, 2004. ACM. 1
- [17] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003. 2
- [18] Konrad Lorincz and Matt Welsh. Motetrack: a robust, decentralized approach to rf-based location tracking. *Personal and Ubiquitous Computing*, 11(6):489–503, 2006. 1
- [19] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, Feb. 2005. 7
- [20] Moteiv Corporation. *Tmote Sky Datasheet, rev. 1.0.4*, November 2006. 4
- [21] N. Patwari, III Hero, A.O., M. Perkins, N.S. Correal, and R.J. O’Dea. Relative location estimation in wireless sensor networks. *Signal Processing, IEEE Transactions on*, 51(8):2137–2148, Aug. 2003. 2

- [22] Special Issue Proceedings of the IEEE, editor. *Sensor networks and applications*, volume 91, August 2003. 1
- [23] R. Olfati Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in multi-agent networked systems. *Proceedings of IEEE*, 95(1):215–233, January 2007. 2, 16
- [24] L. Schenato and G. Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 2289–2294, New Orleans, U.S.A., December 2007. 2
- [25] Ilaria Solida. *Localization services for IEEE802.15.4/Zigbee devices. Mobile Node Tracking (in Italian)*. Master’s thesis, Department of Information Engineering, University of Padova, 2007. 9
- [26] R. Solis, V.S. Borkar, and P.R. Kumar. A new distributed time synchronization protocol for multihop wireless networks. *Decision and Control, 2006 45th IEEE Conference on*, pages 2734–2739, December 2006. 2
- [27] Umberto Spagnolini and A.V. Bosisio. Indoor localization by attenuation maps: model-based interpolation for random medium. In *9th ICEAA International Conference on Electromagnetics in Advanced Application*, pages 1–4, Turin, Italy, September 2005. 1
- [28] D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Distributed sensor fusion using dynamic consensus. In *Proceedings of the 16th IFAC World Congress*, July 2005. 2, 13
- [29] John C. Strikwerda. A probabilistic analysis of asynchronous iteration. *J. of Linear Algebra and its Applications*, 349:125–154, 2002. 2
- [30] K. Whitehouse and D. Culler. Calibration as parameter estimation in sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless sensor networks and applications*, pages 59–67, 2002. 2
- [31] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, September 2004. 7
- [32] G. Zanca and F. Zorzi. Measurements on CC2420 radio chipset. Technical report, Department of Information Engineering, University of Padova, Italy, 2008. 5