Distributed sensor calibration and least–square parameter identification in WSNs using consensus algorithms

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

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 WSN.**

Index Terms— **distributed computing, sensor calibration, least-square estimation, parameter identification, consensus, wireless sensor networks**

I. INTRODUCTION

Wireless 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 [1]. Their strength resides in 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 aspects of one popular application for WSNs, namely localization and target tracking, which has been widely studied in the last few years. 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 location 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 [2], [3]. 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 satellites correspond to the static nodes of the WSN [4]. This approach requires a higher nodes density than the map-based one, but it does not require extensive learning phase. We focus on this last approach.

This work has been supported by the CaRiPaRo foundation under the Wise-Wai funded project.

All the authors are with the Dept. of Information Eng., Univ. of Padova, Via Gradenigo 9/B, 35131, Padova. Fax: +39 049 827 7614. WWW: `http://www.dei.unipd.it/`. E-mails: {saverio.bolognani, simone.delfavero, schenato, damiano.varagnolo}@dei.unipd.it.

Most of the previous work on range-based tracking proposed in the literature assume that the wireless channel model parameters are known or are identified off-line by collecting all data in some centralized location [5]. Unfortunately, these parameters are strongly dependent on the environment [6], [7], in particular indoor, therefore it is desirable to identify them in-situ, possibly using distributed algorithms suitable for the WSN node computational resources. Moreover, the radio signal strength indicator 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 [8].

The main contribution of this work is to propose the use of consensus algorithms for automatic sensor calibration, and for least–square–estimation of the optimal channel parameters. Consensus algorithms are a very popular class of distributed algorithms which has been successfully applied to coordinated robotics [9], time synchronization [10], [11], and distributed estimation [12]. Although the algorithms we propose are applied to localization and tracking for WSNs, they are very general since they can be applied in any context where there is a need to calibrate sensors and to solve a global least square identification problem. Another very 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 unrealistic models. For example, due to packet loss or time synchronization, it is rather problematic in WSNs to enforce convergence to the mean of initial condition, i.e. to enforce average consensus. Therefore, in our work we posed particular care in exploring the tradeoffs between perfect average consensus and randomized consensus.

The paper is organized as follows. In Section II we provide a general mathematical model for the typical communication schemes and the wireless channel model in WSNs. In Section III we describe the experimental testbed used to collect data and, based on these data, we find the appropriate parameters for the mathematical models given in Section II. In Section IV we propose a consensus-based strategy for calibrating sensors with unknown measurement offset readings. In Section V we show how the least square parameter identification problem can be reframed as an average consensus problem. In Section VI we apply the proposed 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. Finally, in Section VII we summarize the results and propose future research directions.

II. WSNs MODELING

A. Connectivity and Communications Models

We model a WSN as a set $\mathcal{N} = \{1, \ldots, N\}$ of N nodes. 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. $(i,j) \in \mathcal{E}_c$ if and only if $c_{ij} \geq c$. This graph is undirected since the matrix C is symmetric. We also denote with $\mathcal{V}(i) = \{j \,|\, (i,j) \in \mathcal{E}, i \neq j\}$ the set of *neighbors* of node i and with the *degree* $d(i) = |\mathcal{V}(i)|$ its cardinality.

The matrix C can be easily experimentally evaluated by letting each node broadcast M packets at random instants (with retransmissions times sufficiently big in order to avoid or reduce collisions).

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 allowed 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 $\frac{1}{d(i)}$.

The WSN model we just presented is an effective mathematical framework for the consensus algorithms, on which our work is widely based on. For a detailed description of consensus algorithms and their analysis we refer to [13]. Here we will just recall that a consensus algorithm is characterized by a sequence of stochastic matrices $\{P(t)\}_{t\in\mathbb{N}}$, which describe the update of the internal state of the whole network when communication takes place. Depending on the communication strategy the sequence can be deterministic or stochastic, with $P(t)$ being a random matrix drawn for a set of possible matrices. The randomness in the matrix is tied to the randomness in the choice of the communicating nodes.

In the broadcast strategy the consensus matrix P_i^B ("B" stands for broadcast) when a node i transmits is given by:

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

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 P_{ij}^G (the superscript "G" stands for gossip) is given by:

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

The consensus matrices defined above are based on the assumption that there is no link failure during the communication. In a c-connectivity graph, there is also an additional probability that the transmission is not successful. In this case, the probability of failure of transmission from node i to a node $j \in \mathcal{V}(i)$ is equal to $1-c$. When link failure happens in broadcast communication, the matrix P_i^B needs to be modified with $[P_i^B]_{jj} = 1, [P_i^B]_{ji} = 0$. Instead, when it happens in symmetric gossip, there is no communication at all, and then no update is performed, i.e. $P_{ij}^G = I$.

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

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

Note that $\overline{P}^B = (\overline{P}^B)^T$ is symmetric and hence doubly stochastic, although the matrices P_i^B are never symmetric. Moreover $\mathcal{G}_{\overline{P}^B} = \mathcal{G}_c$, i.e. the graph associated with the expected consensus matrix $\overline{P}^B$ coincides with the underlying communication graph $\mathcal{G}_c$. Consider then a random consensus algorithm $P(t)$, $P(t)_{ii} > 0$ almost surely, and call $\bar{P} = \mathbb{E}[P(t)]$. A well established fact [13] is that if $\mathcal{G}_{\overline{P}}$ is strongly connected then the algorithm achieves consensus w.p. 1. If moreover the matrices $P(t)$ are all doubly stochastic then the algorithm achieves average consensus w.p. 1. 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 $P^B(t)$. Even if the gossip matrices are not doubly stochastic, the *expected* consensus matrix $\overline{P}^B$ is doubly stochastic, therefore the elements converge to the average of the initial conditions in mean sense. One might also wonder if $\overline{P}^B$ provides some information about convergence rate for the randomized strategy. In [13] 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 $\overline{P}^B$ 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 VI, 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.

Similarly, the expected consensus matrix $\overline{P}^G$ for the symmetric gossip is given by:

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

Obviously $\overline{P}^G = (\overline{P}^G)^T$ since all the gossip matrices P_{ij}^G from which the distribution is drawn are symmetric by construction. Similarly to the broadcast, we have $\mathcal{G}_{\overline{P}^G} = \mathcal{G}_c$. 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 is needed (with no acknowledge). Furthermore, with the symmetric gossip just two nodes receive informations while with the broadcast strategy all the neighbor nodes of the broadcaster do. The rate of converge is then much slower, as can also be guessed noting that the off-diagonal elements of the matrix $\overline{P}^G$ are smaller than their counterparts in $\overline{P}^B$, i.e. there is slower information propagation. We will discuss these differences in more detail in Section VI.

B. Wireless Channel Model

We model the behavior of the wireless channel between two nodes in terms of received power P_{rx} (in dBm) as follows:

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

where P_{tx}^j (in dBm) is the transmitted power, i and j are the receiver and the transmitter nodes, $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^3$ are their spatial positions and t is the time when the communication occurs.

r_j is the *transmission offset* between the nominal and the effectively transmitted power (due to fabrication mismatches; it is assumed to be constant in time). $f_{pl}(\cdot)$ represents the generic *Path Loss* effect, and it is modeled to be (see [6]):

$$f_{pl}(\|\mathbf{x}_i - \mathbf{x}_j\|) = \beta - 10\gamma \log_{10}(\|\mathbf{x}_i - \mathbf{x}_j\|) \tag{2}$$

Fig. 1. Picture of the experimental testbed room: Aula Magna "A. Lepschy", Dept. of Information Engineering, University of Padova.

where β represents the radio receiver gain at a nominal distance of $d = 1m$, and γ is the loss factor. $f_{sf}(\cdot)$ models the *Shadow Fading* and other slow fading components. It is assumed (see [14]) 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. $f_a(\cdot)$ represents the *channel asymmetry* factor. It is due to non symmetric reflections, and we model it to be 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 [6]) 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 *measured received strength offset* 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 measurer has a tolerance of $\pm 6dB$ (see [15]).

The parameters of Eqn. (1) depend on the physical environment where the WSN is placed and on the sensors under consideration, therefore they are not known in advance but they need to be estimated on-site. In the next section we describe the experimental testbed used to collect experimental data from which we will estimate the wireless channel parameters.

III. Experimental Testbed

The experimental data used in the simulations consist in a series of measurements relative to packet sendings and receptions performed by a net of 25 Tmote-Sky [16] nodes equipped with the Chipcon CC2420 RF Transceiver [15]. These nodes were randomly placed inside a conference room of $15m \times 10m$ at about $50cm$ from the ground. The relative position of the nodes is shown in Figure 2.

Each node implemented the randomized broadcast communication using the same transmission power P_{tx} and intercommunication interval $\tau = 15s$ (corresponding to the expected time between 2 transmissions of the same node, sufficiently high in order to consider as rare a packet collision). 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 $c = 0.70$ (i.e. removing links with an empirical packet loss probability greater than 30%) is not the complete graph, even if it is still strongly connected, as shown in Figure 2.

In the following we explain how it is possible to estimate or measure 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 shown that these offsets are negligible (see [17]), so in the following we will ignore this kind of offsets, 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 only, thus, the estimate for the fast fading variance is:

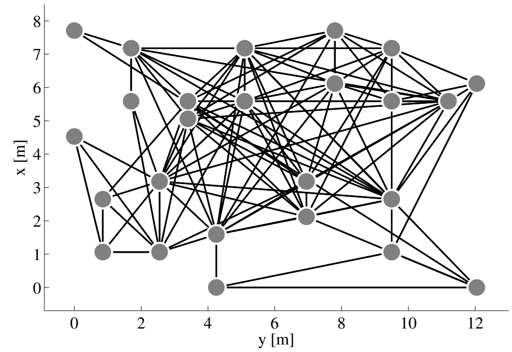


Fig. 2. Network topology and node displacement of experimental testbed. Only edges with empirical packet loss smaller than 25% are displayed.

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

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 assume also 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 by 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 $A = [a_1 \ldots a_M]^T$, $b = [b_1 \ldots b_M]$, 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.:

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

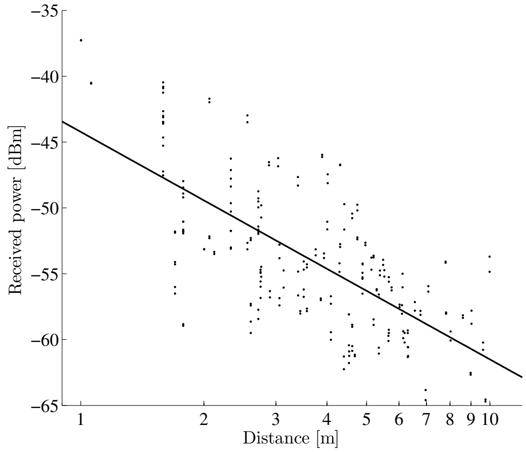


Fig. 3. Estimated path-loss model for the wireless channel, using the standard centralized least square estimate. The line represents the path-loss function, while the dots are the collected measures.

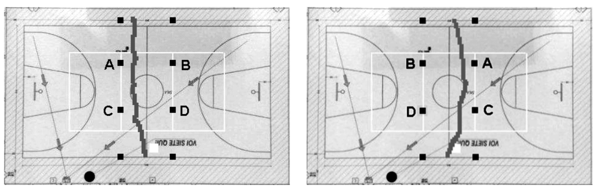


Fig. 4. Experiment inside a basketball court showing the effects of reception offsets in WSN tracking when nodes are swapped. True trajectory in both panels is the court cenerline. Courtesy of ST Microelectronics [18].

Table I summarizes the estimated parameters of the model (1) based on the experimental data collected.

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

TABLE I

ESTIMATED CHANNELS PARAMETERS FOR THE MODEL (1)

IV. DISTRIBUTED SENSORS CALIBRATION

As shown in the previous section, 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. Since unknown location of a moving target is obtained 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.

A. *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 $\widehat{P}_{rx}^{ij} = P_{rx}^{ij} + \hat{o}_i$ 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 this is not critical, as the nonzero offset reached after the calibration phase is completely absorbed during the identification of the path loss model parameter β. 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:

$$\begin{aligned} \bar{P}_{rx}^{ij} &= \frac{1}{T}\sum_{t=1}^{T} P_{rx}^{ij}\left(P_{tx}, \mathbf{x}_i, \mathbf{x}_j, i, j, t\right) \\ &= f_{ij} + o_i + f_{ij}^a + r_j + \frac{1}{T}\sum_{t=1}^{T} v_{ff}(t) \approx f_{ij} + o_i \quad (3) \end{aligned}$$

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 I 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, being $v_{ff}(t)$ 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 1: Let us consider the c-connectivity graph $\mathcal{G}_c = (\mathcal{N}, \mathcal{E}_c)$ of a WSN, and let $P(t) \sim \mathcal{G}_c$ 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:

$$\begin{aligned} \hat{o}_i(0) &= 0, \quad i \in \mathcal{N} = \{1, \ldots, N\} \quad (4) \\ \hat{o}_i(t+1) &= \hat{o}_i(t) + \sum_{j \in \mathcal{V}(i)} p_{ij}(t)\big(y_{ji} - y_{ij} + \hat{o}_j(t) - \hat{o}_i(t)\big) \quad (5) \end{aligned}$$

where $p_{ij}(t) = [P(t)]_{ij}$. Then $\lim_{t\to\infty} o_i + \hat{o}_i(t) = \alpha$ where $\alpha \in [\min_i(o_i), \max_i(o_i)]$. If in addition $P(t)$ are doubly stochastic, 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. (5) can be rewritten as follows:

$$\begin{aligned} \hat{o}_i(t+1) + o_i &= \hat{o}_i(t) + o_i + \textstyle\sum_{j\in\mathcal{V}(i)} p_{ij}(t) \cdot \\ &\quad \cdot \big(f_{ji} + o_j - f_{ij} - o_i + \hat{o}_j(t) - \hat{o}_i(t)\big) \\ x_i(t+1) &= x_i(t) + \sum_{j\in\mathcal{V}(i)} p_{ij}(t)\big(x_j(t) - x_i(t)\big) \\ &= \Big(1 - \sum_{j\in\mathcal{V}(i)} p_{ij}(t)\Big)x_i(t) + \sum_{j\in\mathcal{V}(i)} p_{ij}(t)x_j(t) \\ &= p_{ii}(t)x_i(t) + \sum_{j\in\mathcal{V}(i)} p_{ij}(t)x_j(t) \end{aligned}$$

The last equation can be written in compact form as $x(t+1) = P(t)x(t)$. Since $P(t)$ solves the (probabilistic) consensus problem, then $\lim_{t\to\infty} x_i(t) = \alpha$. The claim that $\alpha \in [\min_i(o_i), \max_i(o_i)]$ follows from the property that if P is a stochastic matrix, then $\max(Px) \leq \max_i(x)$ and $\min(Px) \geq \min(x)$ [19]. ■

The previous theorem indicates how we can compensate the unknown offsets o_i. 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, resulting in an oscillating steady state behavior in the consensus algorithm.

We might wonder whether there is an appropriate choice of $P(t)$ 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

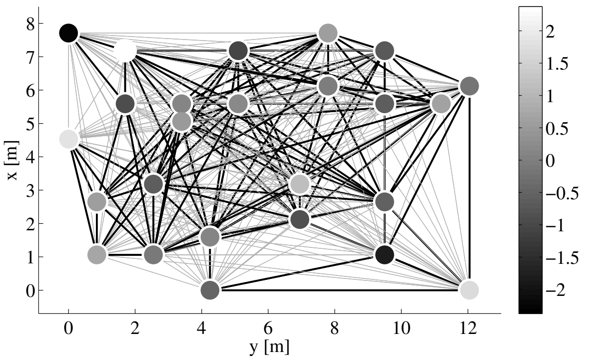
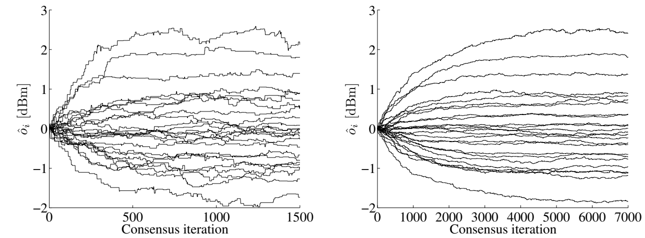


Fig. 5. Network topology and node displacement for c-connectivity graph for $c = 0.1$. Nodes' grey intensity represents the estimated offset $\hat{o}_i$ after calibration. Black and grey edges represent the edges used for training and validation data sets, respectively.



(a) $w = 0.05$ (b) $w = 0.01$

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

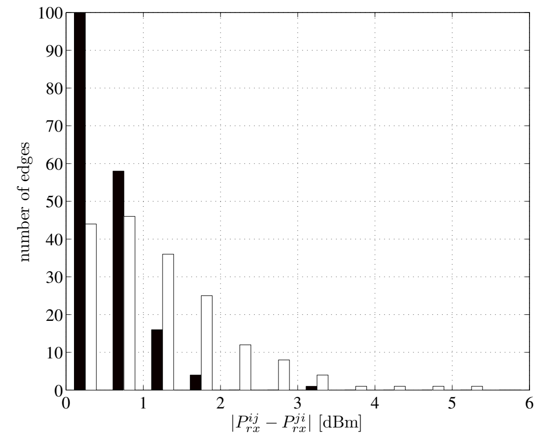


Fig. 7. Asymmetric error distribution before ($|\bar{P}_{rx}^{ij} - \bar{P}_{rx}^{ji}|$, white) and after ($|\bar{P}_{rx}^{ij} + \hat{o}_i - \bar{P}_{rx}^{ji} - \hat{o}_j|$, black) the distributed sensor calibration algorithm.

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 $\mathbb{E}[\mu_o \,|\, o_1, \ldots, o_N] = \frac{1}{N}\sum_{i\in\mathcal{N}} o_i = \alpha^*$ 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 the error becomes smaller and smaller as the number of nodes N increases.

Although the best choice for the compensating the offsets o_i is to choose $P(t)$ which are doubly stochastic, this can be difficult to enforce in a WSN since it requires synchronization among the nodes and compensation for packet loss.

B. Simulations 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 III. 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, 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, 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, and to validate at the same time the model we proposed.

We simulated the randomized broadcast consensus 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 $P(t)$. The steady state compensation offsets $\hat{o}_i(\infty)$ have been reported in Figure 5 by coloring of the nodes with a gray intensity 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 I.

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 has 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 uncertainty of only $28cm$ for a $2m$-long link, and of only $1.4m$ when the node is at $10m$.

There is another practical advantage of running the distributed sensor calibration algorithm before using the network of anchor nodes for localization. If a reception offset is estimated for each node of the network, then there is no need of a bidirectional communication between the anchor nodes and the node that has to be localized. Indeed, there is no need of data trasmission at all between the mobile node and the anchor nodes: it is enough that the mobile node regularly broadcasts toward the rest of the network, without establishing any kind of data connection, an then the network itself estimates its location.

V. Distributed Least Square Estimation with Consensus

In this section we show how to cast the Least Square Parameter Identification (LSPI) problem as an average consensus problem. We begin introducing some basic results on consensus algorithms.

Theorem 2: 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 $z_i \in \mathbb{R}^p, i = 1, \ldots, N$ the data available to i-th node. Suppose that $\xi \in \mathbb{R}^r$ can be defined as follows:

$$\xi = f\Big(\frac{1}{N}\sum_{i\in\mathcal{N}} g_i(z_i)\Big) \tag{6}$$

where $g_i : \mathbb{R}^p \to \mathbb{R}^q$ are generic functions, and $f : \mathbb{R}^q \to \mathbb{R}^r$ is continuous. Suppose that $P(t)$ are stochastic matrices consistent with the graph, i.e. $P(t) \sim \mathcal{G}_c, \forall t$, which solve the (probabilistic) average consensus problem. Let:

$$\begin{aligned} x_i(0) &= g_i(z_i), \quad \forall i \in \mathcal{N} && (7)\\ x_i(t+1) &= p_{ii}(t)x_i(t) + \sum_{j\in\mathcal{V}(i)} p_{ij}(t)x_j(t) && (8)\\ \eta_i(t) &= f(x_i(t)) && (9) \end{aligned}$$

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

$$\lim_{t\to\infty} \eta_i(t) = \xi, \quad \forall i \in \mathcal{N}.$$

Proof: The proof is constructive. If $P(t)$ solves the average consensus problem, then $\lim_{t\to\infty}\prod_{\tau=0}^{t-1}P(\tau) = \frac{1}{N}\mathbf{1}\mathbf{1}^T$, where $\mathbf{1} = [1\ 1\ \ldots\ 1]^T \in \mathbb{R}^N$. If we define the matrix $X = [x_1\, x_2\, \ldots\, x_N]^T \in \mathbb{R}^{N\times q}$, then Eqn. (8) can be written in compact form as $X(t+1) = P(t)X(t)$, from which it follows that $\lim_{t\to\infty} X(t) = \frac{1}{N}\mathbf{1}\mathbf{1}^T X(0)$, i.e. $\lim_{t\to\infty} x_i(t) = \frac{1}{N}\sum_{i\in\mathcal{N}} x_i(0) = \frac{1}{N}\sum_{i\in\mathcal{N}} g(z_i)$. Using the continuity of the function f, we can claim that $\lim_{t\to\infty}\eta_i(t) = \xi, \quad \forall i \in \mathcal{N}$. ■

This theorem simply states that if the variable we need to compute is a function of the mean of some transformation of the data, then it can be computed using distributed average consensus algorithms. An analogous statement of this theorem has been proposed in the context of continuous-time consensus problems by Bauso et al. [20] and later generalized by Cortes [21], known as χ-consensus. Here we proposed a discrete time counterpart which is less demanding in terms of conditions on the functions f and g. In general, the key point is to find the appropriate functions g and f which solve the original problem, being the theorem a straightforward application of consensus theory. Many problems cannot be quite written as in Eqn. (6), but in the more general form $\xi = f\left(\sum_{i\in\mathcal{N}} g(z_i)\right)$. The previous theorem can still be used if N is known, with the simple change $\eta_i = f(Nx_i)$ while the rest of the algorithm is left unchanged. One might wonder if it is possible to compute N distributively, if its value is not known in advance. The following theorem provides a partial answer to this question.

Theorem 3: Let $\mathcal{G}_c = (\mathcal{N}, \mathcal{E}_c)$ the c-connectivity graph associated to a communication network with N nodes, i.e. $N = |\mathcal{N}|$, and assume that there is a special node $k \in \mathcal{N}$, referred as **graph leader**. Without loss of generality, we set $k = 1$. Suppose that $P(t)$ are stochastic matrices consistent with the graph, i.e. $P(t) \sim \mathcal{G}_c, \forall t$, which solve the (probabilistic) average consensus problem. Let:

$$\begin{aligned} x_1(0) &= 1,\\ x_i(0) &= 0, \ \forall i = 2, \ldots, N\\ x_i(t+1) &= p_{ii}(t)x_i(t) + \sum_{j\in\mathcal{V}(i)} p_{ij}(t)x_j(t), \ \forall i \in \mathcal{N}\\ \eta_i(t) &= \frac{1}{x_i(t)} \end{aligned}$$

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

$$\lim_{t\to\infty} \eta_i(t) = N, \quad \forall i \in \mathcal{N}.$$

Proof: The proof is analogous to Theorem 2. By construction it is easy to see that $\lim_{t\to\infty} x_i = \frac{1}{N}\sum_{i=1}^N x_i(0) = \frac{1}{N}$. ■

This theorem shows that it is possible to compute the number of nodes using an average consensus algorithm as long as a leader is elected. As a consequence, the algorithm is not truly distributed, as not all the nodes perform the same actions, being the initialization different. Nonetheless, the problem of leader election is a very well studied problem and efficient algorithms exist [22], therefore a fully distributed algorithm would consist in two stages: a leader election stage and an average consensus stage.

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

$$\hat\theta_{LS} = \arg\min_\theta \|v\| = \arg\min_\theta \|A\theta - b\| = A^\dagger b$$

where $A^\dagger$ represents the pseudo-inverse of A. We now present a theorem showing how the centralized least square parameter identification can be performed over graphs.

Theorem 4: 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 $P(t)$ are stochastic matrices consistent with the graph, i.e. $P(t) \sim \mathcal{G}_c, \forall t$, 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 $i = 1, \ldots, N$ and consider the following algorithm:

$$\begin{aligned} x_i^A(0) &= \sum_{m\in\mathcal{D}(i)} a_m a_m^T, \quad \forall i \in \mathcal{N} && (10)\\ x_i^b(0) &= \sum_{m\in\mathcal{D}(i)} a_m b_m && (11)\\ x_i^k(t+1) &= p_{ii}(t)x_i^k(t) + \sum_{j\in\mathcal{V}(i)} p_{ij}(t)x_j^k(t), \ k = A, b && (12)\\ \eta_i(t) &= \left(x_i^A(t)\right)^\dagger x_i^b(t) && (13) \end{aligned}$$

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

$$\lim_{t\to\infty} \eta_i(t) = \hat\theta_{LS}, \quad \forall i \in \mathcal{N}.$$

Proof: Let us define the matrix $S = A^TA = \sum_{i=1}^M a_i a_i^T$ and the vector $q = A^Tb = \sum_{i=1}^M a_i b_i$, therefore $\hat\theta_{LS} = S^\dagger q$. By construction we have that $\lim_{t\to\infty} x_i^A(t) = \frac{1}{N}\sum_{i=1}^N x_i^A(0) = \frac{1}{N}\sum_{i=1}^M a_i a_i^T = \frac{1}{N}S$ and similarly $\lim_{t\to\infty} x_i^b(t) = \frac{1}{N}\sum_{i=1}^N x_i^b(0) = \frac{1}{N}\sum_{i=1}^M a_i b_i = \frac{1}{N}q$. By continuity $\lim_{t\to\infty}\eta_i(t) = (\frac{1}{N}S)^\dagger\frac{1}{N}q = S^\dagger q = \hat\theta_{LS}$. ■

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 $S = A^TA$ 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.

VI. SIMULATIONS

In this section we apply the results presented in the previous section to distributively identify the unknown path-loss channel parameters (β, γ) using consensus algorithms and testing different communication strategies. As mentioned above, these two parameters are used in localization algorithms in order to estimate relative distances between nodes. Therefore, it is critical to be able to identify the path-loss parameters in a distributed way, robustly 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 [7]. 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 LSPI described in Section V using the randomized broadcast and the randomized symmetric gossip, respectively. The third strategy performs the randomized symmetric gossip consensus on local estimates of the channel parameters vector θ rather than on the least-square sufficient statistics x^A, x^b of Theorem 4. Each strategy has its own advantages. In fact, the randomized symmetric gossip guarantees average consensus, therefore, when applied to x^A and x^b, it is guaranteed to provide the best identification accuracy since it satisfies the hypotheses of Theorem 4. Randomized broadcast does not guarantee average consensus, and consequently the best performance, however it is very easy to be implemented 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 2 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 $\mathcal{G}_c = (\mathcal{N}, \mathcal{E}_c)$ shown in Figure 2. The data set $\mathcal{D}(i)$ available to each node i is given by $\mathcal{D}(i) = \{(\bar{P}_{rx}^{ij}, d_{ij}) \,|\, j \in \mathcal{V}(i)\}$, i.e. all the averaged received power measurements from each neighbor coupled with the corresponding relative distances (note that the distances are assumed to be known). 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 IV 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 III, 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 $m = 1, \ldots, M$ indicates a generic data element, and $M = |\mathcal{D}| = |\mathcal{E}_c|$. Using the same terminology of Theorem 4 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 assessment (in terms of identification accuracy) is based on the residues of the estimate $\hat{\theta}$ given by:

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

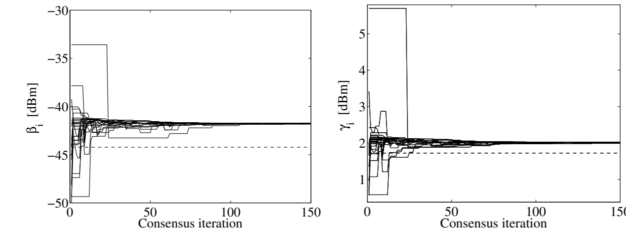


Fig. 8. Convergence of parameter estimates β_i, γ_i using randomized broadcast least-square consensus and consensus weight $w = 0.5$. The dashed lines are the centralized least squares estimates $\hat{\beta}_{LS}, \hat{\alpha}_{LS}$.

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 the $P(t)$'s are doubly stochastic then from Theorem 4 it follows that $\lim_{t\to\infty} J(\eta_i(t)) = J(\hat{\theta}_{LS}), \forall i$, and so $\lim_{t\to\infty} \frac{1}{N}\sum_{i\in\mathcal{N}} J(\eta_i(t)) = J(\hat{\theta}_{LS})$.

In the first simulation, we tested the randomized broadcast least-square strategy using the connectivity matrix C defined in Section II-A 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 and 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 II 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 steady state identification error. We also have that the steady state value is strongly realization dependent, as it can be noticed from to 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 [13] that the distance of $\mathbb{E}[\bar{J}(k)]$ form 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

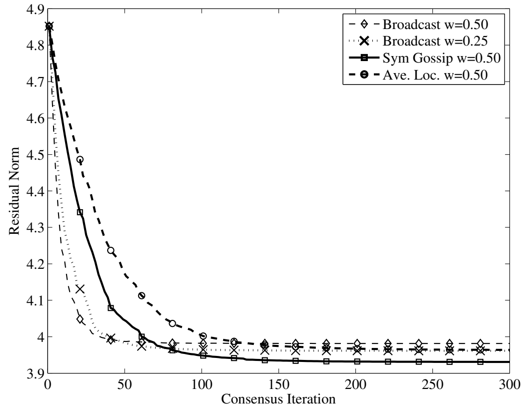


Fig. 9. Comparison of the mean estimation residual $\mathbb{E}[\bar{J}(k)]$ for different randomized consensus algorithms.$\mathbb{E}[\bar{J}(k)]$ is approximately computed as the average of 50 independent extractions of the sequence $\{P(t)\}_{t\in\mathbb{N}}$.

Consensus Algorithm	$\mathbb{E}[\bar{J}(\infty)]$	$\max \bar{J}(\infty)$	$\min \bar{J}(\infty)$
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 II

COMPARISON OF THE MEAN ESTIMATION RESIDUAL.

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 II), not only that $\mathbb{E}[\bar{J}(k)]$ tends to optimal value. Finally, we tested also the randomized gossip algorithm based on direct average of local least-square estimate. More precisely, this strategy is based on the general algorithm of Theorem 2 where $x_i(0) = \theta^i_{LS}$ and $\eta_i(t) = x_i(t)$, therefore $\lim_{t\to\infty} \eta_i(t) = \frac{1}{N}\sum_{n\in\mathcal{N}} \hat{\theta}^i_{LS} \neq \hat{\theta}_{LS}$. 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 that the simple solution of an average of the local least squares estimations.

VII. 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 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. For example, the offset removal algorithm could also be used to detect malfunctioning sensors by observing the magnitude of the compensation offset $\hat{o}_i$, 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.

VIII. ACKNOWLEDGMENTS

The authors would like to thank Giovanni Zanca, Francesco Zorzi and all the SIGNET group of the University of Padova for the discussions about wireless channel modeling and the data support.

REFERENCES

[1] S. I. Proceedings of the IEEE, Ed., *Sensor networks and applications*, vol. 91, no. 8, August 2003.

[2] K. Lorincz and M. Welsh, "Motetrack: a robust, decentralized approach to rf-based location tracking," *Personal and Ubiquitous Computing*, vol. 11, no. 6, pp. 489–503, 2006.

[3] U. Spagnolini and A. Bosisio, "Indoor localization by attenuation maps: model-based interpolation for random medium," in *ICEAA Intl. Conf. on Electromagnetics in Adv. Application*, Sept. 2005.

[4] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Tenth Annual International Conference on Mobile Computing and Networking*. MobiCom, 2004.

[5] N. Patwari, A. O. H. III, M. Perkins, N. Correal, and R. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE transaction on signal processing*, 2002.

[6] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[7] M. Dominguez-Duran, D. Claros, C. Urdiales, F. Coslado, and F. Sandoval, "Dynamic calibration and zero configuration positioning system for WSN," in *IEEE MELECON 08*, May 2008.

[8] K. Whitehouse and D. Culler, "Calibration as parameter estimation in sensor networks," in *Proceedings of the 1st ACM Intl. Workshop on Wireless sensor networks and applications*, 2002, pp. 59–67.

[9] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.

[10] R. Solis, V. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, December 2006.

[11] 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*, New Orleans, U.S.A., December 2007, pp. 2289–2294.

[12] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed sensor fusion using dynamic consensus," in *Proccedings of the 16th IFAC World Congress*, July 2005.

[13] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *submitted to IEEE Trans. on Selected areas in communication*, 2007.

[14] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics Letters*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.

[15] *SmartRF CC2420 Datasheet, rev. 1*, Chipcom AS, Nov. 2003. [Online]. Available: http://www.chipcon.com/

[16] *Tmote Sky Datasheet, rev. 1.0.4*, Moteiv Corporation, Nov. 2006. [Online]. Available: http://www.sentilla.com/pdf/eol/tmote-sky-datasheet.pdf

[17] G. Zanca and F. Zorzi, "Measurements on CC2420 radio chipset," Department of Information Engineering, University of Padova, Italy, Tech. Rep., 2008.

[18] I. 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.

[19] J. L. Doob, *Stochastic Processes*. New York: John Wiley & Sons, Inc., 1953.

[20] D. Bauso, L. Giarré, and R. Pesenti, "Nonlinear protocols for optimal distributed consensus in networks of dynamic agents," *Systems and Control Letters*, 2006.

[21] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, pp. 726–737, 2008.

[22] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann, 1997.