Information diffusion algorithms over WSNs for non-asymptotic confidence region evaluation

Alex Calisti, Davide Dardari, Gianni Pasolini, Michel Kieffer, Francesca Bassi

Information Diffusion Algorithms over WSNs for Non-Asymptotic Confidence Region Evaluation

Alex Calisti, Davide Dardari, Gianni Pasolini
DEI-University of Bologna
V.le Risorgimento, 2, Bologna, Italy

Michel Kieffer, Francesca Bassi
Laboratoire des Signaux et Systemes, CNRS CentraleSupelec
Univ Paris-Sud, F- 91192 Gif-sur-Yvette, France

***Abstract*—Getting confidence regions for parameter estimates obtained from data collected by a wireless sensor network (WSN) is very important to assess the performance of the estimator. The sign perturbed sums (SPS) approach has been proposed recently to defined exact confidence regions in a centralized setting even if only few measurements are available. SPS may be distributed to get confidence regions at each node of a WSN. This paper investigates a data dissemination strategy called Tagged and Aggregated Sums (TAS), exploiting the particularities of SPS, to efficiently provide each node with the information necessary to evaluate locally the confidence region. TAS and flooding (FL) algorithms have been investigated through simulations and then implemented on commercial sensor nodes. The impact of collision avoidance mechanisms at the medium access control (MAC) layer is also experimentally assessed. Performance comparisons show that TAS outperforms FL in structured networks.**[1]

I. Introduction

A WSN consists of energy-limited sensing devices deployed to monitor the surrounding world and estimate one or more physical parameters (parameter vector). In a centralized setup, a central unit collects all the information and performs the estimation task, whereas in a distributed setup sensor nodes exchange information and accomplish the estimation locally.

Whatever the adopted processing strategy, either centralized or distributed, in many applications a simple point estimate of the parameter vector of interest is not sufficient if not associated with a confidence region to assess the estimation uncertainty. Classically, the estimation accuracy is obtained from Cramér-Rao-like bounds [1]. Confidence regions can also be derived as a by-product of distributed Kalman filtering [2]. However, strong assumptions on measurement noise (typically Gaussian) are necessary and a good characterization of confidence regions is only possible for a large number of measurements.

The SPS [3], [4] method, instead, defines exact confidence regions under mild conditions on the distribution of the measurement noise even with a low number of measurements. Provided that the regression model is linear and that the measurement noise samples are independently and symmetrically distributed, the SPS method allows the derivation, from a *finite* set of measurements, of confidence regions with *prescribed confidence levels* around the least squares (LS) estimate of the parameter vector. Initially proposed for centralized estimation, SPS has been shown in [5] to be amenable to distributed estimation in a WSN.

[1]This research has been supported by the European H2020 project XCycle (Grant no. 635975) and by the RFO (Univ. of Bologna) awarded by G. Pasolini.

A. Main Contributions

This paper addresses the distributed evaluation of confidence regions as defined by the SPS approach, for implementation in WSNs. In [5] we showed that confidence regions, as defined by SPS, may be evaluated in a distributed way: the nodes share their local information with each other and the confidence region computation is performed locally. The information diffusion strategy, in addition to network topology, determines the amount of data exchanged, which needs to be restrained. On this regard, a novel information diffusion strategy, named TAS, is presented in this paper.

It exploits the peculiarities of the SPS approach to distill and aggregate the information to be transmitted by each node, with the objective to reduce the total amount of data exchanged within the WSN. Its performance, in terms of generated traffic load, is compared with that of the classical FL algorithm that, conversely, does not perform any aggregation.

The performance of both algorithms has been firstly investigated through simulations and then measured on a real WSN. Simulations and experimental results show that TAS outperforms FL in structured networks. Analytical investigations have been carried out as well, which are reported in an extended version of this paper [6].

II. Problem Formulation

Consider some spatial field described by the following parametric model

$$y\left(\mathbf{x}, \boldsymbol{\theta}\right) = \boldsymbol{\varphi}^T\left(\mathbf{x}\right)\boldsymbol{\theta}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ is some vector of experimental conditions (time, location...) under which the field is observed, $\boldsymbol{\varphi}\left(\mathbf{x}\right)$ is the regressor function, and $\boldsymbol{\theta}$ is the vector of unknown parameters, belonging to the parameter space $\Theta \subset \mathbb{R}^{n_\mathrm{p}}$. Measurements are taken by a network of n sensor nodes, spread at random locations $\mathbf{x}_i \in \mathbb{R}^{n_x}$, $i = 1, \ldots, n$. Node i collects the scalar measurement y_i according to the local measurement model

$$y_i = y\left(\mathbf{x}_i, \boldsymbol{\theta}^*\right) + w_i = \boldsymbol{\varphi}_i^T\boldsymbol{\theta}^* + w_i, \tag{2}$$

where $\boldsymbol{\varphi}_i = \boldsymbol{\varphi}(\mathbf{x}_i)$ is the $n_\mathrm{p} \times 1$ deterministic regressor vector at $\mathbf{x}_i$; $\boldsymbol{\theta}^*$ is the true value of the $n_\mathrm{p} \times 1$ parameter vector; w_i represents the measurement noise at Node i. As in [4], the random variables with realizations w_i, $i = 1 \ldots, n$ are assumed to be statistically independent and to follow a symmetrical distribution. We consider the worst case in which the value of $\boldsymbol{\varphi}_i$ is assumed known only by Node i. Moreover, the regressors are such that $\det \mathbf{Q}_n \neq 0$, where

$$\mathbf{Q}_n = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T. \tag{3}$$

The purpose of the network is to let each node capable of computing locally the confidence region of the estimate of $\boldsymbol{\theta}^*$ with the lowest impact on network traffic.

The centralized SPS approach [3], [4] assumes all measurements and regressors to be known at the central processing unit. It defines an exact confidence region around the least squares estimate $\widehat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}^*$, obtained as the solution of the normal equations $\sum_{k=1}^{n} \boldsymbol{\varphi}_k \left(y_k - \boldsymbol{\varphi}_k^T \boldsymbol{\theta}\right) = \mathbf{0}$. For that purpose, as in [4], consider the *unperturbed sum* as the following function over Θ

$$\mathbf{s}_0(\boldsymbol{\theta}) = \mathbf{Q}_n^{-1/2} \sum_{k=1}^{n} \boldsymbol{\varphi}_k \left(y_k - \boldsymbol{\varphi}_k^T \boldsymbol{\theta}\right) \tag{4}$$

and the $m-1$ *sign-perturbed sums*, defined $\forall j = 1, \ldots, m-1$ as the following functions over Θ

$$\mathbf{s}_j(\boldsymbol{\theta}) = \mathbf{Q}_n^{-1/2} \sum_{k=1}^{n} a_{j,k} \boldsymbol{\varphi}_k \left(y_k - \boldsymbol{\varphi}_k^T \boldsymbol{\theta}\right), \tag{5}$$

where $a_{j,i} \in \{\pm 1\}$ are realizations of independent random signs.[2] For each $\boldsymbol{\theta} \in \Theta$, one considers the elements of the set

$$\mathcal{Z}(\boldsymbol{\theta}) = \left\{ z_j(\boldsymbol{\theta}) = ||\mathbf{s}_j(\boldsymbol{\theta})||_2^2 \right\}_{j=0,1,\ldots,m-1}, \tag{6}$$

and lists them in increasing order, giving rise to a permutation $\pi_{\boldsymbol{\theta}}(\cdot) : \{0, \ldots, m-1\} \to \{0, \ldots, m-1\}$. One defines the set

$$\boldsymbol{\Sigma}_q = \left\{ \boldsymbol{\theta} \in \Theta \mid \pi_{\boldsymbol{\theta}}(0) \leq m - 1 - q \right\} \tag{7}$$

which contains all $\boldsymbol{\theta} \in \Theta$ for which the rank of $z_0(\boldsymbol{\theta})$ in the ordering is among the $m - q$ smallest.

In [3], [4], it was proven that

$$\mathrm{Prob}(\boldsymbol{\theta}^* \in \boldsymbol{\Sigma}_q) = 1 - \frac{q}{m}. \tag{8}$$

As a consequence, by properly setting the parameters m and q, it is possible to define $\boldsymbol{\Sigma}_q$ as a non-asymptotic confidence region with *exact* confidence level $1 - q/m$.

In the following, the distributed computation of $\boldsymbol{\Sigma}_q$ is addressed considering different information diffusion strategies.

[2] A random sign is a symmetric ± 1 value random variable taking both values with the same probability.

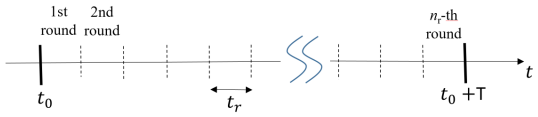

Figure 1: Time scheduling management

III. Information Diffusion Algorithms

In this section, concurrent procedures for information diffusion applicable to any network topology are considered. The purpose is that each node collects the largest amount of information with the lowest amount of data exchanged in the network so that it is able to compute locally the confidence region of the LS estimate for any $\boldsymbol{\theta}^*$.

In the following we propose a novel information diffusion strategy, named TAS, aimed at efficiently disseminating the information required by the SPS approach for the distributed computation of confidence regions.

The main idea of TAS, which will be detailed in Section III-B, is to let each node aggregate data before transmitting them to the other nodes, exploiting the fact the SPS approach does not require the knowledge of each single measurement and regressor but, rather, the knowledge of their aggregation in the form of the sums reported in (4) and (5).

In order to assess the effectiveness of such strategy, the FL algorithm has been considered as a benchmark, as it does not perform any data aggregation. Although it is true other information diffusion strategies exist [7], [8], they are tailored to specific network topologies, either flat or hierarchical (tree, clusters, ...) or require some side information. TAS is, instead, agnostic, as it is meant as a general purpose solution. For this reason, the FL algorithm, which is agnostic as well, is its natural term of comparison.

Obviously, given a fixed network topology, it is always possible to design an ad-hoc information diffusion algorithm that minimizes the amount of exchanged data (even TAS could be customized to specific topologies). However, we are interested in designing procedures that are not tailored to any specific network configuration.

Before entering into the details of TAS and FL, few words on the time axis management are required, which hold for both algorithms: We assume that the WSN is operated for a time interval of T seconds, denoted measurement period, which is divided into n_r rounds of equal duration $t_\mathrm{r} = \frac{T}{n_\mathrm{r}}$ (see Fig. 1). TAS and FL operations take place according to such timing organization, as detailed hereafter.

A. Flooding algorithm

Data may be disseminated via FL algorithm, used as a benchmark in this paper. At round $r = 1$, Node i broadcasts its own privy pair $(\boldsymbol{\varphi}_i, y_i)$, along with a tag containing its identifier, and receives data from its neighbors. The privy data consists of $d_\mathrm{FL} = n_\mathrm{p} + 1$ real values, corresponding to its measurement and n_p regressors, whereas the transmission cost d_TAG for the tag vector, which indicates the indexes of the nodes whose measurements are present in the packet, depends

Round	From Node	Data	Tag vector						
0	1	$\boldsymbol{\delta}_1$	1	0	0	0	0	0	0
1	2	$\boldsymbol{\delta}_2$	0	1	0	0	0	0	0
	3	$\boldsymbol{\delta}_3$	0	0	1	0	0	0	0
2	2	$\boldsymbol{\delta}_{4,7}$	C	0	0	1	0	0	1
	3	$\boldsymbol{\delta}_{4,6}$	C	0	0	1	0	1	0
3	2	$\boldsymbol{\delta}_{5,6}$	0	0	C	0	1	1	0
	3	$\boldsymbol{\delta}_5$	0	C	0	0	1	0	0

Table I: Table $\mathbf{R}^{(1)}$ for Node $k=1$ using TAS in the network of Figure 2; C indicates elements that have been removed from the tag vector and partial sums during the distillation phase

on the way it is represented, *e.g.*, as a list of integers, in which case it is of variable length with r, or a constant-size vector of binary flags. The latter is considered in this work.

The dimension of data broadcast by Node i at successive rounds is an integer multiple of d_{FL}, possibly zero.

On successive rounds, Node i has possibly received new data from its neighbors and broadcasts previously received pairs that it did not transmit before, along with the respective identifiers. Retransmission of already sent data is avoided keeping trace of previously transmitted information in a local runtime table. Provided that no transmission error occurs, at the end of the information diffusion phase, each node has the whole set $\{(\boldsymbol{\varphi}_i, y_i)\}_{\forall i}$ needed to evaluate (7).

Ideally, transmission rounds are repeated until all nodes collect all the information, e.g., by checking the tag vector is full of ones. Upon completion, each node is able to compute (4) and (5), for any $\boldsymbol{\theta}$, and to locally derive the confidence region using the full set of data. In practice, transmission rounds may stop when the measurement period T expires or when all nodes do not detect any transmitted information from their neighbors over a given time interval. In the latter cases, the local confidence region characterization may be performed on a reduced, possibly different across nodes, set of data.

B. Tagged and aggregated sums (TAS) algorithm

The TAS algorithm is based on the following consideration. Expanding (4) and (5) one gets,

$$\mathbf{s}_0(\boldsymbol{\theta})=\mathbf{Q}_n^{-1/2}\left(\sum_{k=1}^{n}\boldsymbol{\varphi}_k y_k-\left(\sum_{k=1}^{n}\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T\right)\boldsymbol{\theta}\right) \tag{9}$$

$$\mathbf{s}_j(\boldsymbol{\theta})=\mathbf{Q}_n^{-1/2}\left(\sum_{k=1}^{n}a_{j,k}\boldsymbol{\varphi}_k y_k-\left(\sum_{k=1}^{n}a_{j,k}\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T\right)\boldsymbol{\theta}\right). \tag{10}$$

The evaluation of (9) and (10) for any value of $\boldsymbol{\theta}\in\Theta$ does not necessarily require the knowledge of each term in the sums but rather of

$$\boldsymbol{\delta}_{1\ldots n}=\Bigg\{\underbrace{\sum_{k=1}^{n}\boldsymbol{\varphi}_k y_k}_{n_{\text{p}}\text{ real values}},\quad\underbrace{\sum_{k=1}^{n}\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T}_{n_{\text{p}}^2\text{ real values}},\quad\underbrace{\left\{\sum_{k=1}^{n}a_{j,k}\boldsymbol{\varphi}_k y_k\right\}_{j=1}^{m-1}}_{n_{\text{p}}(m-1)\text{ real values}},\quad\underbrace{\left\{\sum_{k=1}^{n}a_{j,k}\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T\right\}_{j=1}^{m-1}}_{n_{\text{p}}^2(m-1)\text{ real values}}\Bigg\} \tag{11}$$

The main idea of the TAS algorithm is to propagate data structures similar to (11), composed of *partial sums* not necessarily ranging from $k=1$ to n, but covering a subset of $\{1,\ldots,n\}$. At each transmission round, Node k generates and transmits partial sums built from data previously received from neighbors and stored in a local table $\mathbf{R}^{(k)}$. The main challenge of the TAS algorithm is to determine a way to organize the content of the transmitted partial sums so that each node is able, after the termination of the transmission phase, to build the complete sums (11), or to compute partial sums with the maximum number of elements using the received partial sums. The main advantage of TAS with respect to FL is that the transmitted data sets are of constant size, independently of the transmission round. The size of the data sets is

$$d_{\text{TAS}}=m\left(n_{\text{p}}+n_{\text{p}}\frac{n_{\text{p}}+1}{2}\right) \tag{12}$$

real values, *independently* of the number of elements in the partial sums. The evaluation of d_{TAS} takes into account the fact that $\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T$ is symmetric[3]. As in FL, the tag vector has to be transmitted along with the data set at each transmission round. For TAS we adopted the same representation of the tag vector used by FL. This means that its transmission cost is the same for both algorithms.

The TAS algorithm consists of six phases, namely, i) initialization, ii) reception, iii) distillation, iv) aggregation, v) transmission, and vi) wrap-up. They are introduced hereafter.

i) *Initialization phase*. As in the FL algorithm, the transmitted packet is formed by a data set and by a tag vector. During the initialization phase, Node k, $\forall k\in\{1,\ldots,n\}$ creates the packet $\left(\mathbf{t}_1^{(k)},\mathbf{d}_1^{(k)}\right)$ to be sent in round $r=1$. The tag vector $\mathbf{t}_1^{(k)}$ flags only Node k. The data set $\mathbf{d}_1^{(k)}$ contains the local quantities related to Node k

$$\mathbf{d}_1^{(k)}=\left\{\boldsymbol{\varphi}_k y_k,\left\{\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T\right\},\left\{a_{j,k}\boldsymbol{\varphi}_k y_k\right\}_{\forall j},\left\{a_{j,k}\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T\right\}_{\forall j}\right\}. \tag{13}$$

After initialization, the reception, distillation, aggregation, and transmission phases are sequentially repeated until a termination condition is met.

ii) *Reception phase*. At each round r, Node k collects the messages containing the partial sums transmitted by its neighbors (according to the given logical topology), whose set is denoted $\mathcal{N}(k)$.

iii) *Distillation phase*. At the end of the reception phase of round r, Node k compares the incoming tag vectors $\mathbf{t}_r^{(j)}$, $j\in\mathcal{N}(k)$ to the previously received tag vectors, to detect whether the packets received at round r contain new information. If possible, partial sums received during round r are *cleared* of partial sums previously received. The cleared partial sums are then stored in $\mathbf{R}^{(k)}$. This phase reduces the number of contributors to each partial sum, so that the different

[3]Since $\sum_{k=1}^{n}\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T$ is symmetric, instead of transmitting all its ${\boldsymbol{n}_{\text{p}}}^2$ elements, it is sufficient to transmit $\boldsymbol{n}_{\text{p}}$ values for the diagonal plus $\sum_{\boldsymbol{d=1}}^{\boldsymbol{n_{\text{p}}-1}}\boldsymbol{d}=\frac{\boldsymbol{n_{\text{p}}(n_{\text{p}}-1)}}{\boldsymbol{2}}$ values for the upper (or lower) part, that gives $n_{\text{p}}\frac{n_{\text{p}}+1}{2}$. The same holds for the $(m-1)$ terms $\left\{\sum_{k=1}^{n}a_{j,k}\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T\right\}_{j=1}^{m-1}$.

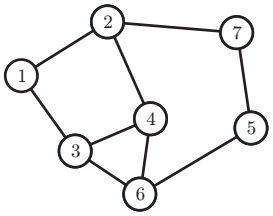

Figure 2: Toy network example

partial sums can be more easily recombined in the following *aggregation phase*, with each contributor counted no more than once.

Example 1 (Distillation phase): Consider the network of Figure 2 and the evolution of $\mathbf{R}^{(1)}$ given in Table I. For $r=0$, Node 1 only holds its own data and forms partial sums from these data stored in

$$\boldsymbol{\delta}_1=\left\{\boldsymbol{\varphi}_1 y_1, \left\{\boldsymbol{\varphi}_1\boldsymbol{\varphi}_1^T\right\}, \left\{a_{j,1}\boldsymbol{\varphi}_1 y_1\right\}_{\forall j}, \left\{a_{j,1}\boldsymbol{\varphi}_1\boldsymbol{\varphi}_1^T\right\}_{\forall j}\right\}.$$

During round $r=1$, Node 1 broadcasts these partial sums and receives partial sums formed with the privy data from Node 2 and partial sums formed with the privy data from Node 3. During round $r=2$, Node 1 receives a packet containing partial sums combining data from Nodes 1, 4, and 7, forwarded by Node 2, as well as a packet containing partial sums combining data from Nodes 1, 4, and 6, forwarded by Node 3. The content of these two packets is stored in $\mathbf{R}^{(1)}$, after having removed the contribution related to Node 1 from each previously received partial sum (this is indicated by a C in the tag vector in Table I). Node 1 thus gets

$$\boldsymbol{\delta}_{4,6} = \left\{ \sum_{k\in\{4,6\}} \boldsymbol{\varphi}_k y_k, \sum_{k\in\{4,6\}} \boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T, \left\{\sum_{k\in\{4,6\}} a_{j,k}\boldsymbol{\varphi}_k y_k\right\}_{\forall j}, \left\{\sum_{k\in\{4,6\}} a_{j,k}\boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T\right\}_{\forall j} \right\} \quad (14)$$

and $\boldsymbol{\delta}_{4,7}$. At the end of round $r=3$, Node 1 receives a packet with partial sums combining data from Nodes 3, 5, and 6, forwarded by Node 2, as well as a packet with partial sums combining data from Nodes 2 and 5, forwarded by Node 3.

iv) *Aggregation phase*. To create the packet to be broadcast at round r, Node k aggregates the partial sums available in $\mathbf{R}^{(k)}$ at round $r-1$ and which were *not* previously aggregated. This is done by summing the available partial sums to produce $\mathbf{d}_r^{(k)}$ and merging the related tag vectors to produce $\mathbf{t}_r^{(k)}$. In order to avoid duplication of terms in the sums, rows i and j of $\mathbf{R}^{(k)}$ can be merged in $\left(\mathbf{d}_r^{(k)}, \mathbf{t}_r^{(k)}\right)$ iff the intersection of i-th and j-th row tag vectors is empty. If this condition is not met, only the row with smallest index is aggregated in a transmitted packet.

Example 2 (Aggregation phase): Consider the evolution of $\mathbf{R}^{(2)}$ for Node 2 given in Table II. At the end of round $r=1$, Node 2 holds partial sums related to the data from Nodes 1, 2, 4, and 7, stored in $\boldsymbol{\delta}_1$, $\boldsymbol{\delta}_2$, $\boldsymbol{\delta}_4$, and $\boldsymbol{\delta}_7$. A packet containing $\boldsymbol{\delta}_2$ has already been transmitted in round $r=1$. The other

Round	From Node	Data	Tag vector						
0	2	$\boldsymbol{\delta}_2$	0	1	0	0	0	0	0
1	1	$\boldsymbol{\delta}_1$	1	0	0	0	0	0	0
	4	$\boldsymbol{\delta}_4$	0	0	0	1	0	0	0
	7	$\boldsymbol{\delta}_7$	0	0	0	0	0	0	1
2	1	$\boldsymbol{\delta}_3$	0	C	1	0	0	0	0
	4	$\boldsymbol{\delta}_6$	0	C	C	0	0	1	0
	7	$\boldsymbol{\delta}_5$	0	C	0	0	1	0	0

Table II: Table $\mathbf{R}^{(2)}$ for Node $k=2$ using TAS in the network of Figure 2; C indicates elements that have been removed from the tag vector and partial sums during the distillation phase

tag vectors do not intersect, as a consequence, the aggregated sums will involve $\boldsymbol{\delta}_1$, $\boldsymbol{\delta}_4$, and $\boldsymbol{\delta}_7$. The packet containing the aggregated sums is received by Node 1 and cleared of $\boldsymbol{\delta}_1$.

v) *Transmission phase*. The message obtained at the end of the aggregation phase is broadcast to all neighbor nodes. After the last transmission phase, the objective for Node k is the computation of the local confidence region, using the data collected so far and aggregated in the final partial sum $\boldsymbol{\delta}_F^{(k)}$, evaluated in the wrap-up phase. The information diffusion process stops for Node k when it has collected all the information from other nodes or, more realistically, when the measurement period T has expired.

vi) *Wrap-up phase*. The wrap-up phase can be performed by a Node whenever it needs to compute the confidence region during or at the end of the information diffusion process. Ideally, Node k evaluates a linearly weighted sum $\boldsymbol{\delta}_F^{(k)} = \sum_l \widehat{b}_l^{(k)}\boldsymbol{\delta}_l^{(k)}$, where $\boldsymbol{\delta}_l^{(k)}$ contains the partial sums at the l-th row of $\mathbf{R}^{(k)}$ and $\widehat{\mathbf{b}}^{(k)}$ is the solution of the following constrained optimization problem

$$\widehat{\mathbf{b}}^{(k)} = \arg\max_{\mathbf{b}} \ \mathbf{b}^T\mathbf{T}^{(k)}\mathbf{1}, \quad (15)$$

with the constraints

$$c_i^{(k)} = \sum_l b_l \ t_{l,i}^{(k)} \in \{0,1\}, \ \ i=1,2,\ldots,n. \quad (16)$$

$$\det \sum_l b_l \left(\sum_{k\in\mathbf{t}_l^{(k)}} \boldsymbol{\varphi}_k\boldsymbol{\varphi}_k^T \right) \neq 0. \quad (17)$$

Here, the binary vector $\widehat{\mathbf{b}}^{(k)}$ selects the rows of $\mathbf{R}^{(k)}$ to be combined in the partial sums. In particular, the positions of the elements of $\widehat{\mathbf{b}}^{(k)}$ equal to 1 select the corresponding rows of $\mathbf{R}^{(k)}$. Moreover, $t_{l,i}^{(k)}$ are the elements of the tag matrix $\mathbf{T}^{(k)}$ of $\mathbf{R}^{(k)}$, with l and i denoting the row and column indexes, respectively. The constraints (16) are related to the presence indicator of the quantities associated to Nodes $i=1,\ldots,n$. Imposing $c_i^{(k)} \in \{0,1\}$ in (16) ensures that all measurements contribute similarly to the final sign perturbed sums, with some measurements possibly not contributing at all. In the latter case, one obtains a confidence region associated to the LS estimate of $\boldsymbol{\theta}^*$ involving only the corresponding subset of sensor measurements. The constraint (17) is introduced to allow the computation of an approximation of $\mathbf{Q}_n^{-1/2}$ relying on possibly less than n terms.

Once a satisfying solution has been found, Node k can locally compute an exact confidence region based on $\boldsymbol{\delta}_F^{(k)}$, from which the following quantities are evaluated

$$\widetilde{\mathbf{s}}_0^{(k)}(\boldsymbol{\theta}) = \widetilde{\mathbf{Q}}^{-1/2} \sum_{i=1}^{n} c_i^{(k)} \boldsymbol{\varphi}_i \left(y_i - \boldsymbol{\varphi}_i^T \boldsymbol{\theta}\right) \tag{18}$$

$$\widetilde{\mathbf{s}}_j^{(k)}(\boldsymbol{\theta}) = \widetilde{\mathbf{Q}}^{-1/2} \sum_{i=1}^{n} c_i^{(k)} a_{j,i} \boldsymbol{\varphi}_i \left(y_i - \boldsymbol{\varphi}_i^T \boldsymbol{\theta}\right) \; \forall j = 1, \ldots, m-1, \tag{19}$$

with

$$\widetilde{\mathbf{Q}} = \frac{1}{\sum_{i=1}^{n} c_i^{(k)}} \sum_{i=1}^{n} c_i^{(k)} \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T. \tag{20}$$

If several satisfying solutions for (15-16) have been found, the one maximizing (17) should be selected to get the smallest confidence region, as in D-optimal experiment design [9].

Remark 1: The TAS algorithm is inspired from network coding. The main difference is that Node k does not need to recover the privy data of all nodes, but the decoding of their partial sums suffices.

Remark 2: The efficiency of TAS with respect to FL comes from the fact that the size d_{TAS} of the data sets exchanged does not increase as the number of rounds increases, as it happens in FL.

IV. Simulation Results

Simulations results presented in this section have been obtained considering sensor nodes randomly deployed over a square of side of one measurement unit, which transmit information over lossless links (*i.e.*, no transmission errors and no packet collisions). Confidence regions have been evaluated with the interval analysis techniques described in [10] and the Intlab library [11] for interval computations. Data are generated considering the model (1), with randomly generated parameters and regressors using realizations of independent zero-mean unit variance Gaussian variables.

Due to lack of space only random trees are considered in this paper.[4] In particular, we build a spanning tree on top of a random unstructured network, setting the inter-node communication range $d_{\text{comm}} = \sqrt{\frac{\log_2 n}{2n}}$. According to [12], this range guarantees almost sure connectivity of a network of n nodes, deployed on a unit area.

For the SPS approach, we chose $q = 1$ and $m = 10$ to be able to characterize 90% confidence regions according to (8).

For each n (see the horizontal axis in Figure 3), 100 random tree realizations are instantiated. TAS and FL are compared in terms of the required number of data to be transmitted in each network realization. The success rate of TAS is the percentage of network realizations that proved favorable to TAS, *i.e.*, for which less measurements need to be exchanged to get all information required by the SPS approach reaching all nodes of the network.

[4] A complete investigation, considering different network topologies as well as further analytical and simulation results, is presented in [6].

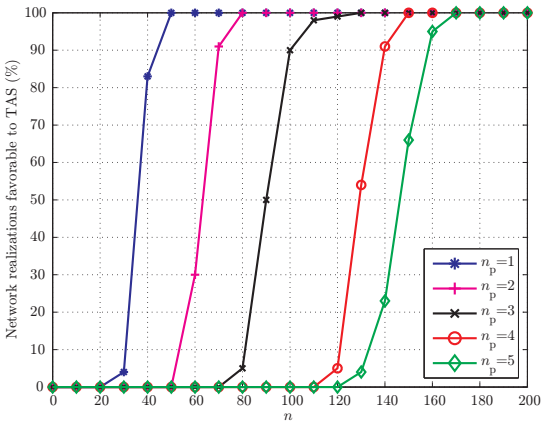

Figure 3: Percentage of network realizations favorable to TAS, in terms of required data exchanges, compared to FL, as a function of n and n_{p}.

Parameter	Symbol	value
Number of nodes	n	52
Maximum number of children of each node	$n_{\max}$	5
Measurement period	T	2 s
Number of rounds	n_{r}	$\{2, 3, \ldots, 30\}$
Number of neighbours	n_{n}	$\{2, 4, 8, 17, 33\}$
Number of parameters to be estimated	n_{p}	1
Number of sign perturbed sums	m	10

Table III: Parameters of the experimental setup

Figure 3 shows this success rate as a function of n, for several values of n_{p}. We can observe that there always exists a threshold value of n, depending on n_{p}, above which TAS outperforms FL, *i.e.*, the percentage closes to 100%.

Supported by such results, we also implemented both TAS and FL on a real network in order to compare their performance in a real scenario, in the presence of interference and possible MAC issues. This experimental test-bed is presented in the following section, along with its outcomes.

V. Experimental results

This section describes the practical implementation of both TAS and FL on commercial sensor nodes deployed in a real scenario. These experiments aim at:

- revealing critical issues that could arise when operating TAS on real networks;
- investigating possible interactions between TAS and the networking protocol stack, with particular reference to the MAC strategy adopted by commercial devices;
- providing a truthful performance comparison between TAS and FL.

The experimental setup and the outcomes of the measurement campaign are described in the following. For the reader's convenience, the most significant symbols and their meaning are recalled in Table III.

A. Experimental setup

1) Sensor nodes: Both TAS and FL have been implemented on EMB-Z2530PA sensor nodes [13] equipped with

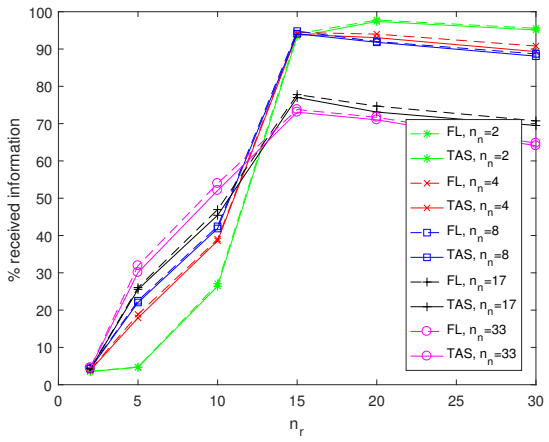

Figure 4: Average percentage of information received by a node. The legend entries and the curves in the right-hand part of the figure are in the same order

temperature sensors. These devices are compliant with the IEEE802.15.4 standard, thus adopting the carrier sense multiple access with collision avoidance (CSMA/CA) protocol at the MAC layer. Different setups have been adopted in terms of transmission power, in order to control the average number of neighbors n_{n}, ranging from 2 to 33.

2) Network topology: The random tree topology has been considered in this paper, thus a tree structure is randomly established by the nodes themselves at each run.

3) Network setup and data management: The tree construction starts from the root (level 0), which randomly selects the number n_{c} of children with uniform (discrete) distribution in $[1, 2, ..., n_{\mathrm{max}}]$. Provided that a sufficient number of nodes is available within the coverage range of the root, n_{c} of them are randomly selected as its children. Otherwise, all (thus less than n_{c}) available nodes are joined to level 1. The same procedure is repeated by each node of level 1 and then iterated at all levels, until all nodes join the tree.

Once the network has been established, the information diffusion algorithm, either FL or TAS, is started, beginning from the leaves up to the root and then in the opposite direction. In our experimental setup, the information transmitted by a node to its father is not overheard by its children, to reflect the typical behavior of energy efficient routing protocols that put in the sleep mode nodes not involved in the current transmission phase.

For FL, each payload contains the amount of data transmitted, measurements and regressors, and a unique tag vector that identifies the contributing nodes. For TAS, payloads contain partial aggregated sums and a tag vector indicating the contributing nodes. In both cases the tag consists of a vector of d_{TAG} bits, with 1s at the positions corresponding to the indexes of the contributing nodes.

4) Time scheduling management: Nodes are allowed to transmit only during the round pertaining to the tree level they belong to. Data (measurements and corresponding regressors for FL or aggregated sums for TAS) are then exchanged beginning from the leaves up to the root and then in the opposite direction.

To emulate the time jitter in nodes operations caused by local clocks drift in a distributed network as well as to avoid all nodes access the channel simultaneously, thus congesting the MAC, each node defers the measurement phase, and therefore also the beginning of the information diffusion algorithm, by locally choosing a random delay $\Delta_i \in [0, t_{\mathrm{r}}]$, with $i = 1, 2, ..., n$.

All nodes stop data dissemination once n_{r} rounds have been completed. The amount of data transmitted/received by each node are then collected to allow the analysis of the behavior of the TAS and FL algorithms.

Remark 3: A trade-off is expected between n_{r} and t_{r}. In fact, for a given degree of connectivity, n_{r} should be large enough to allow the diffusion of data over the whole network, hopping from one neighbor to the other. A the same time, t_{r} should be large enough to reduce as much as possible collisions with neighbors' transmissions within the same round. Since the product $n_{\mathrm{r}} t_{\mathrm{r}}$ is equal to the measurement period T, the values of n_{r} and t_{r} should be jointly and properly chosen for the given (application dependent) T, in order to maximize the amount of information received by nodes.

This aspect, that arises from the interplay between the information diffusion strategy and the MAC protocol, will be investigated in the following section.

B. Measured Performance

A network of $n = 52$ nodes performing simple temperature measurements has been considered, thus $n_{\mathrm{p}} = 1$. The temperature θ^* is assumed constant in the area where the nodes are deployed.

The data to be transmitted by the FL algorithm are collections of pairs $(\boldsymbol{\varphi}_i, y_i)$, consisting in this case of $d_{\mathrm{FL}} = 2$ real values (which may be quantized) in the first round and that might grow in the subsequent ones.

Also in the experimental test-bed we chose $q = 1$ and $m = 10$ to be able to characterize 90% confidence regions according to (8). Therefore, the amount of data transmitted at each round by TAS is $d_{\mathrm{TAS}} = 20$ real values (which may also be quantized) and remains constant.

In such conditions ($n = 52$, $n_{\mathrm{p}} = 1$, $m = 10$) TAS should outperform FL, as suggested by simulation results in Fig. 3.

The measurement period is taken as $T = 2$ s. n_{r} ranges from 2 to 30 and, therefore, t_{r} varies from 1 s down to 67 ms. Finally, the tag size d_{TAG} is 7 Bytes, and the maximum number of children of each node of the tree is $n_{\mathrm{max}} = 5$.

For each chosen setup (transmit power, n_{r}), we performed the measurement campaign over 100 network realizations and we derived the average (over the 100 resulting networks) amount of information received by each node and the average amount of data transmitted in the whole network.

Figure 4 shows the average proportion ρ of data reaching a given node in a tree topology for various n_{r} and n_{n}.

The value of n_{r} that maximizes the average amount of received data depends on n_{n}. For low values of n_{r}, the performance is limited by the constraint on the maximum

Neighbors	FL	TAS
2	2047	1330
4	2022	1331
8	1978	1322
17	1400	1042
33	1256	972

Table IV: Average amount of transmitted data (scalars) within the whole network in the case $n_{\mathrm{r}} = 15$.

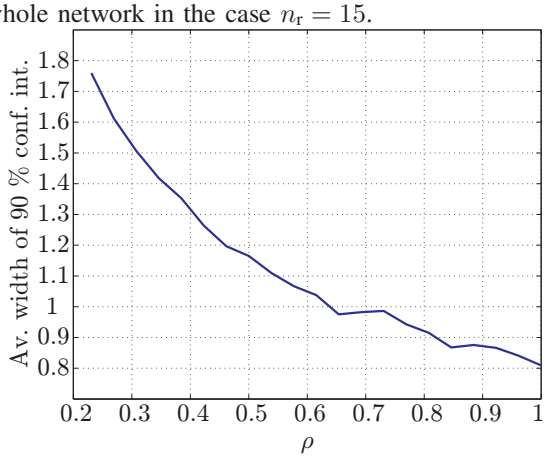

Figure 5: Average width of the 90 % confidence region as a function of the proportion ρ of measurements collected by a node with FL or TAS.

number of allowed hops (that is coincident with n_{r}), that might not be sufficient for a particular data to reach all nodes in the network. On the contrary, for large values of n_{r} the performance is limited by the MAC, as a small t_{r} increases the collision probability.

From the same figure one can also see that better performances are obtained when the network is characterized by a low degree of connectivity n_{n}, provided that a sufficiently high number of rounds can be allocated within the measurement period. In fact, large n_{n}, *i.e.*, high power levels, generate more interference among nodes that leads to critical conditions at the MAC level. This suggests that a proper power control strategy able to keep n_{n} at minimum values for connectivity is beneficial both for network performance as well as to save energy.

Figure 4 shows that FL and TAS perform similarly in all conditions, hence they are equivalent from the viewpoint of the amount of received information. They differ, instead, in terms of amount of transmitted information, as evident in Table IV, which reports the average amount (over 100 network realizations) of transmitted data (scalars) within the whole network in the case $n_{\mathrm{r}} = 15$.

As can be observed, TAS outperforms FL, managing to reduce the amount of transmitted information and being an effective solution to mitigate the network burden for distributed confidence region evaluations.

To evaluate the influence of the proportion of measurements received by each node on the quality of the confidence region that can be derived, a temperature measurement has been performed by each of the n nodes of the network. For different target proportions $\rho \in [0, 1]$ of measurements reaching some node of the network, 100 random selections of a subset of measurements have been considered and a 90 % confidence region evaluation has been performed. Figure 5 describes the evolution of the average width of the 90 % confidence region as a function of the proportion of measurements collected by a given node. From Fig. 4 and 5, one may deduce the width of the confidence interval that may be obtained with FL or TAS, when not all measurements have reached some node. One can for example see that even if only 80 % of the measurements have reached a node, the width of the confidence region is only 10 % larger than that obtained from all measurements.

VI. Conclusions

In this paper we have proposed and investigated a novel information diffusion strategy, namely TAS, especially designed for the distributed evaluation of non-asymptotic confidence regions in WSNs with the SPS approach. The TAS algorithm has been compared with the classical FL in terms of average transmitted information in a real scenario using experimental data. Both algorithms have been implemented on off-the-shelf sensor nodes organized according to a tree topology. The impact of the MAC strategy has been highlighted and discussed. The contribution has shown that, on tree topologies, the proposed TAS algorithm outperforms the FL algorithm.

References

[1] D. Jourdan, D. Dardari, and M. Win, "Position error bound for UWB localization in dense cluttered environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, pp. 613–628, April 2008.

[2] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *Proc. of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009, pp. 7036–7042.

[3] B. C. Csáji, M. C. Campi, and E. Weyer, "Non-asymptotic confidence regions for the least-squares estimate," in *Proc. IFAC SYSID*, Brussels, Belgium, 2012, pp. 227–232.

[4] ——, "Signed-Perturbed Sums: A new system identification approach for constructing exact non-asymptotic confidence regions in linear regression models," *IEEE Trans. on Signal Processing*, vol. 63, no. 1, pp. 169–181, January 2015.

[5] V. Zambianchi, M. Kieffer, F. Bassi, G. Pasolini, and D. Dardari, "Distributed SPS algorithms for non-asymptotic confidence region evaluation," in *Proc. of the 23rd European Conference on Networking and Communication, EUCNC 2014*, Bologna, Italy, June 2014.

[6] V. Zambianchi, F. Bassi, A. Calisti, D. Dardari, M. Kieffer, and G. Pasolini, "Distributed non-asymptotic confidence region computation over sensor networks," *submitted to IEEE Transactions on Signal and Information Processing over Networks*, 2016.

[7] A. R. Khan, N. Rakesh, A. Bansal, and D. K. Chaudhary, "Comparative study of WSN protocols (LEACH, PEGASIS and TEEN)," in *2015 Third International Conference on Image Information Processing (ICIIP)*, Dec 2015, pp. 422–427.

[8] S. Das and J. M. F. Moura, "Distributed kalman filtering with dynamic observations consensus," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4458–4473, Sept 2015.

[9] E. Walter and L. Pronzato, *Identification of Parametric Models from Experimental Data*. London: Springer-Verlag, 1997.

[10] M. Kieffer and E. Walter, "Guaranteed characterization of exact non-asymptotic confidence regions as defined by LSCR and SPS," *Automatica*, vol. 50, no. 2, pp. 507–512, February 2014.

[11] S. Rump, "INTLAB - INTerval LABoratory," in *Developments in Reliable Computing*, T. Csendes, Ed. Dordrecht: Kluwer Academic Publishers, 1999, pp. 77–104. [Online]. Available: http://www.ti3.tu-harburg.de/rump/

[12] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar 2000.

[13] EMBIT. (2016) EMB-Z2530PA. [Online]. Available: http://www.embit.eu/products/wireless-modules/emb-z2530pa/