Scalable Data-Coupled Clustering for Large Scale WSN

Mihaela I. Chidean, Eduardo Morgado, Eduardo del Arco, Julio Ramiro-Bargueño, and Antonio J. Caamaño

***Abstract*—Self-organizing algorithms (SOAs) for wireless sensor networks (WSNs) usually seek to increase the lifetime, to minimize unnecessary transmissions or to maximize the transport capacity. The goal left out in the design of this type of algorithms is the capability of the WSN to ensure an accurate reconstruction of the sensed field while maintaining the self-organization. In this work, we formulate a general framework where the data from the resulting clusters ensures the well-posedness of the signal processing problem in the cluster. We develop the second-order data-coupled clustering (SODCC) algorithm and the distributed compressive-projections principal component analysis (D-CPPCA) algorithm, that use second-order statistics. The condition to form a cluster is that D-CPCCA does not fail to resolve the Principal Components in any given cluster. We show that SODCC is scalable and has similar or better message complexity than other well-known SOAs. We validate these results with extensive computer simulations using an actual LS-WSN. We also show that the performance of SODCC is, comparative to other state-of-the-art SOAs, better at any compression rate and needs no prior adjustment of any parameter. Finally, we show that SODCC compares well to other energy efficient clustering algorithms in terms of energy consumption while excelling in data reconstruction Average SNR.**

***Index Terms*—WSN, self-organization, data-coupled clustering, compressed projections, principal component analysis.**

I. Introduction

WIRELESS sensor networks (WSNs) aim at the correct recovery of the data measured by the sensors and gathered by a Data Fusion Center (DFC). A simple measure-and-transmit policy is not suitable for a Large Scale WSN (LS-WSN, high amount of resource limited nodes and large datasets); it causes the network collapse [1], due to interference or blocking. Therefore, to maintain the network operation, complex or even ingenious techniques are needed. In this work, we consider two complementary strategies: 1) network partitioning and 2) in-network processing, and propose a general framework for LS-WSNs that combines them using the characteristics of the measured data field. Our goal is to exploit the data correlations (i.e., spatio-temporal) to partition the network in such way that is data-coupled, leading to a well-posed problem for the in-network algorithm.

The partition of the network into clusters provides scalability to the LS-WSNs [2], [3]. By clustering a WSN different goals are sought: load balance, fault tolerance or network connectivity. The measured data—the reason to exist of LS-WSNs—and their recovery are of little importance to popular clustering algorithms [3]. One of the alternatives are the data-driven clustering algorithms, where the decision criteria are based on characteristics of the measured data. For example, Le *et al.* [4] use the dissimilarity between the actual and the average sensed data as decision criteria. Wang *et al.* [5] consider an entropy-based divergence measure criterion, aiming to increase the global compression gain for distributed source coding in multimedia WSNs. Hung *et al.* [6] and Liu *et al.* [7] compute the Manhattan distance between the measurements as similarity metric for approximate data gathering in WSNs.

The self-organization of the network is a desirable feature for the WSN partitioning algorithm as no centralized or external entity is required. The early propositions of Self-Organized Algorithms (SOA) focused on maximizing the network lifetime considering a limited energy budget [8], [9], minimizing the amount of control message interchange [10] and adapting to entrance or exit of sensor nodes into the WSN [11]. However, to the best of our knowledge, no attempt to use the data correlations to drive a self-organizing partitioning algorithm has been made.

Regarding in-network processing, the distributed source coding [12] with rate-distorsion control in the data compression is one of the most researched distributed processing techniques for WSNs. Although it has been proposed for some years ago [13], its practical applications have been scarce. For example, it has been applied to distributed video coding [14], but its main hindrance is the huge amount of inter-node control message generated. A different in-network strategy is used in [15] where a distributed Principal Component Analysis (PCA) is applied to data from clusters partitioned according to the *k*-means criterion, where no data characteristics are attended.

In the present work we aim to propose an algorithm that increases the operation efficiency (accuracy of reconstructed measurements and life expectancy) of WSNs by using both *data-coupled* clustering and in-network processing techniques. The main difference between the data-coupled and data-driven clustering algorithms is the use of the measured data. While

Manuscript received October 3, 2014; revised February 26, 2015; accepted April 13, 2015. Date of publication April 20, 2015; date of current version September 7, 2015. This work has been partially supported by the Research Projects S2013/MAE-2835 from the Autonomous Community of Madrid and TEC2013-48439-C4-1-R from the Spanish Ministry of Economy and Competitiveness. The work of M. I. Chidean is supported by the FPU Research Grant AP2012-2981 from the Spanish Ministry of Education, Culture and Sports. The associate editor coordinating the review of this paper and approving it for publication was S. Cui.

The authors are with the Department of Signal Theory and Communications, Rey Juan Carlos University of Madrid, Fuenlabrada 28943, Spain (e-mail: mihaela.chidean@urjc.es; eduardo.morgado@urjc.es; eduardo.delarco@urjc.es; julio.ramiro@urjc.es; antonio.caamano@urjc.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2015.2424693

data-coupled algorithms partition the network by complying with the requirements of the in-network processing algorithm to recover the measured field with the highest possible fidelity, data-driven algorithms do not abide to this constraint. The additional condition has the effect of redefining the capacity limits by reducing the amount of transmitted packets at large distances. With the present proposal the network is able to operate more efficiently and the network lifetime is increased.

The main contributions of this work are:

- the proposal of a general framework for the self-organized data-coupled algorithms.
- the Distributive Compressive Projection PCA (D-CPPCA) algorithm.
- the Second-Order Data-Coupled Clustering (SODCC) algorithm.

The usage of both proposed algorithms in a LS-WSN allows for optimal network partitioning in terms of reconstruction quality of the sensed field and scalability.

Following, in Section II we detail the general framework and the formulations of both D-CPPCA and SODCC algorithms. In Section III we analyze the cluster size distribution and the scalability of SODCC, and propose solutions to the drawbacks of the present formulation. In Section IV we evaluate performance of SODCC with D-CPPCA in an actual LS-WSN setting and compare it with state-of-the-art clustering algorithms (both SOA and non-SOA) in terms of reconstructed data quality and power consumption. Finally, in Section V we conclude the present work, detailing the comparative advantages and shortfalls of the proposed algorithms.

II. Self-Organized Data-Coupled Algorithms

In this section we detail the main contributions of this work. First, we propose a general framework that combines both clustering and in-network processing algorithms and seeks a clustering configuration that helps the further processing. Next, we describe the in-network processing algorithm D-CPPCA used in this work, that is based on second-order statistics of the measured data. Following we explain the Fast Subspace Decomposition (FSD) algorithm [16] that estimates the dimension of the signal subspace of a dataset by using second-order statistics. The FSD is used by the SODCC algorithm, proposed in the last subsection, an algorithm that configures the WSN properly for D-CPPCA, following the general framework.

A. General Framework

Let us consider a LS-WSN with N sensor nodes, configured in N_c non-overlapping clusters, each with size N_i, where $i = 1, \ldots, N_c$.

Let be an in-network processing algorithm implemented in a distributed fashion in each of the N_c clusters that form the WSN. For example, this algorithm can exploit the redundancy of the sensed data and encode it, to transmit K_i encoded messages instead of the N_i data messages ($K_i < N_i$) to the DFC. To obtain a good reconstruction in the DFC, the algorithm needs a well-posed problem in each cluster (i.e., non-zero determinant of the experimental covariance matrix), regardless of the cluster size. Then, this processing algorithm may use some statistic properties of the sensed data to be optimized, such as the variance or kurtosis. Such data processing algorithm must allow for independent operation of partitions of the problem. Further, it must exhibit an asymptotic behaviour (as the number of partitions diminish) tending to that of the centralized problem [17].

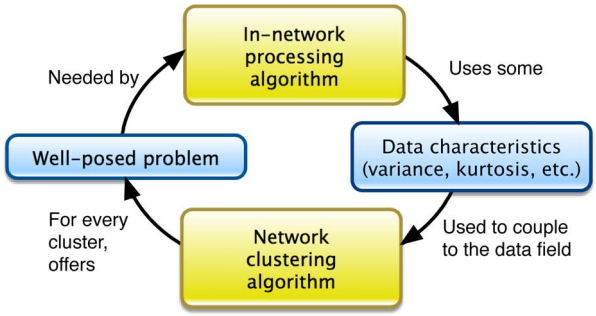

Fig. 1. General framework for self-organized data-coupled algorithms.

On the other hand, data coupled clustering algorithms also use some characteristic of the sensed data as part of the decision criterion and guide for the network partitioning. As both in-network processing and clustering algorithms use the same data, the obvious choice, in terms of computational burden, is to use the same characteristics. Therefore, the clustering algorithm couples the network configuration to the sensed data by forming clusters in which the in-network processing algorithm has a well-posed problem to solve.

Fig. 1 summarizes this general framework and shows the link between the two strategies considered in this work. This framework generates a family of in-network processing and clustering algorithms combinations, each determined by the statistic employed.

In this work, we select the Compressive-Projections PCA (CPPCA) [18] as the in-network processing algorithm, described in detail in the following subsection. This algorithm is PCA based and therefore uses second-order statistics as data characteristics for the encoding/decoding. CPPCA is also prone to be separated into independent working pieces.

As the in-network processing is based on the variance, the clustering algorithm that we present in this work also uses second-order statistics to decide the cluster formation. In short, the dimension of the signal subspace of the dataset gathered by each cluster is computed iteratively using the FSD [16]. The value obtained is used in the decision criterion, as we will see in detail in Section II-D.

With this general framework, a traditional and centralized data gathering, as the one plotted in Fig. 2(a), is not only converted to distributed data gathering (see Fig. 2(b)), but to a *data-coupled* setting, as in Fig. 2(c). The main advantages of a data-coupled setting include the well-posedness of the encoding problem in each cluster, the adjustment of the cluster configuration to the needs of the sensed data and the possibility

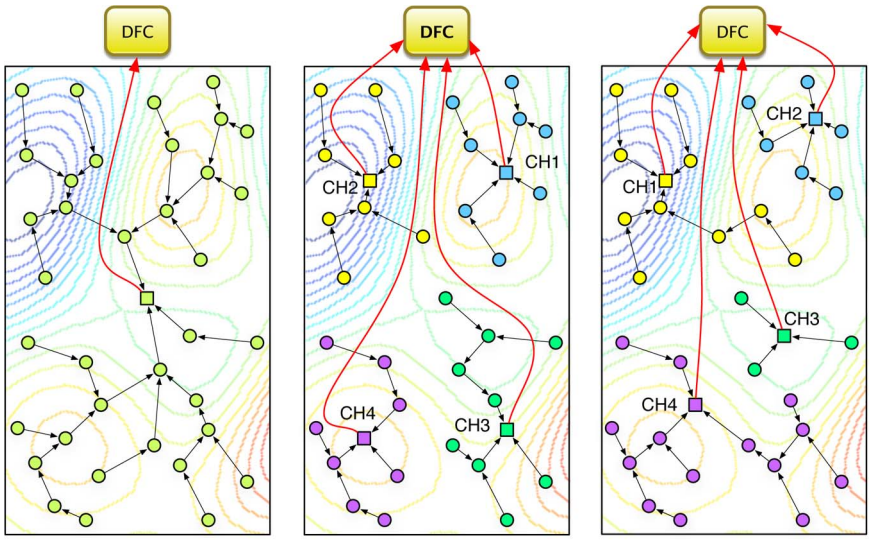

Fig. 2. Example of system architecture for (a) centralized, (b) distributed and (c) data-coupled settings. Circles and squares indicate sensors and Cluster Head (CH) locations, respectively, and arrows show the transmission direction.

to adjust the compression rate (namely K_i/N_i) and the quality of the reconstruction to the needs of each network area.

B. Distributed Compressive-Projection PCA (D-CPPCA)

As previously mentioned, in this work we selected a PCA based in-network algorithm, namely CPPCA, that was first proposed by Fowler in [18] for satellite hyperspectral imagery compression. This algorithm shifts the computational burden of PCA from the resource-limited satellite (encoder) to the Earth base-station (decoder). The light-encoder/heavy-decoder system architecture of CPPCA seems to be perfect for a LS-WSN scenario.

CPPCA is a distributable algorithm, as it can be implemented independently in each cluster. In this setting, the Cluster Head (CH) acts as the encoder for that N_i nodes that form the i-th cluster in the network. As mentioned in the previous subsection, the use of the Distributed CPPCA procedure to provide the clusters with data compression capabilities is but one of the many possible algorithms. We have selected this algorithm for the sake of simplicity as random projections are simple vector-matrix multiplications that can be easily carried out in sensor nodes.

To provide basic definitions and to relate the LS-WSNs terminology with the CPPCA one, we now summarize the method. For a complete explanation refer to [18].

Consider that each of the N_i nodes that form a cluster have M measurements available. The $N_i \times M$ data matrix $\mathbf{X}$ is to be assembled as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$, where each column is a zero-mean vector $\mathbf{x}_m \in \mathbb{R}^{N_i}$, $m = 1, \ldots, M$. Let $\mathbf{\Sigma}$ be the covariance matrix of $\mathbf{X}$, estimated as $\mathbf{\Sigma} = \mathbf{X}\mathbf{X}^\top/M$, where $(\cdot)^\top$ indicates the transpose operation.

The main advantage of CPPCA is that the data encoding is but an orthonormal projection to a lower-dimension random subspace. Consider the subspace $\mathcal{P}$ of dimension K_i and the orthonormal Compressed Projection matrix $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_{K_i}]$ of size $N_i \times K_i$, which provides an orthonormal projection onto $\mathcal{P}$. Therefore, $\mathbf{y}_m = \mathbf{P}\mathbf{P}^\top\mathbf{x}_m = \mathbf{P}\tilde{\mathbf{y}}_m$ is the projection of $\mathbf{x}_m$ onto $\mathcal{P}$, where $\tilde{\mathbf{y}}_m$ represents the encoded vector. The covariance matrix of $\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_M]$ is

$$\tilde{\mathbf{\Sigma}} = \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top/M = \mathbf{P}^\top\mathbf{X}\mathbf{X}^\top\mathbf{P}/M = \mathbf{P}^\top\mathbf{\Sigma}\mathbf{P} \tag{1}$$

Regarding the implementation of the CPPCA algorithm, the encoder splits its dataset $\mathbf{X}$ into J partitions $\mathbf{X}^{(j)}$, $j = 1, \ldots, J$, corresponding each to a projection matrix $\mathbf{P}^{(j)}$ of a different K-dimensional subspace $\mathcal{P}^{(j)}$. Then it computes and transmits the projected vectors $\tilde{\mathbf{Y}}^{(j)} = \mathbf{P}^{(j)}\mathbf{X}^{(j)}$ to the decoder. In the original CPPCA procedure, the projection operators $\mathbf{P}^{(j)}$ are known *a priori* by the decoder.

The covariance matrix $\tilde{\mathbf{\Sigma}}^{(j)}$ of the received projected data $\tilde{\mathbf{Y}}^{(j)}$ is estimated, using the Rayleigh-Ritz (RR) procedure [19]. The POCS (Projections Onto Convex Sets) [18] optimization is performed using a set of the Ritz vectors obtained from $\tilde{\mathbf{\Sigma}}^{(j)}$, to resolve the first L_i eigenvectors and assemble them into a $N_i \times L_i$ matrix $\mathbf{\Psi}$. The matrix $\mathbf{\Psi}$ is an approximation of the L_i-component PCA transform, with $L_i \leqslant K_i$. Once $\mathbf{\Psi}$ is obtained, the PCA coefficients are recovered solving $\tilde{\mathbf{Y}}^{(j)} = (\mathbf{P}^{(j)})^\top\mathbf{\Psi}\check{\mathbf{X}}^{(j)}$.

CPPCA is suitable for LS-WSNs, since it loads the sensors with a low computational burden and the number of transmissions towards the DFC decreases proportional to K_i (along with the resulting accesses to the wireless channel). It has been previously shown that CPPCA applied to a flat WSN (with no clusters) has similar performance to PCA in a LS-WSN [20].

For a WSN partitioned in N_c clusters, we define the D-CPPCA in-network processing algorithm, where in each CH uses CPPCA to encode the data measured by the entire cluster and the DFC acts as unique decoder for the network.

C. Dimension of the Signal Subspace of a Cluster

To follow the general framework in Section II-A, the data-coupled clustering algorithm proposed in this work uses the same order statistics as the in-network algorithm D-CPPCA. Thus, we firstly explain the method used to compute the dimension of the signal subspace, value that is later used in the decision criterion of the clustering algorithm.

To compute the dimension of the signal subspace robustly and iteratively as the cluster evolves, we employ the FSD algorithm [16]. The FSD algorithm estimates the first $\hat{d}$ RR eigenvalues and eigenvectors (spanning the signal subspace) up to the $\hat{d} = d$ iteration, where d is the actual dimension. FSD is based on the Lanczos method and has $O(N^2 d)$ computational complexity, much lower than that of the traditional eigendecomposition, that has order of $O(N^3)$.

For a $N \times M$ matrix, the statistic $\varphi_{\hat{d}}$ is defined as

$$\varphi_{\hat{d}} = M(N-\hat{d}) \log \left[\frac{\sqrt{\frac{1}{N-\hat{d}} \left(\|\tilde{\mathbf{\Sigma}}\|^2 - \sum_{n=1}^{N} \theta_n^2 \right)}}{\frac{1}{N-\hat{d}} \left(\mathsf{Tr}\tilde{\mathbf{\Sigma}} - \sum_{n=1}^{N} \theta_n \right)} \right] \tag{2}$$

where $\|\cdot\|$ is the Frobenius norm and θ_n are the RR eigenvalues. In each iteration, the $\varphi_{\hat{d}}$ statistic is computed and, for $\hat{d} \geqslant d+1$, $\varphi_{\hat{d}}$ approaches a χ^2 distribution with $(1/2)(N-\hat{d})(N-\hat{d}+1) - 1$ degrees of freedom.

Finally, it is proven that, for M samples, the following equation is valid [16]

$$\varphi_{\hat{d}} \leqslant \gamma_{\hat{d}}\, c(M) \tag{3}$$

where $\gamma_{\hat{d}}$ is a threshold computed *a priori* as the end tail of the χ^2 distribution. Also, function $c(M)$ must comply with the following:

$$\lim_{M\to\infty} \frac{c(M)}{M} = 0 \quad \text{and} \quad \lim_{M\to\infty} \frac{c(M)}{\log\log M} = \infty \tag{4}$$

In practice, its asymptotic behavior must be "slower" than linear but "faster" than $\log\log$; functions such as $c(M) = \log(M)$ or $c(M) = \sqrt{\log(M)}$ can be used [16].

Finally, using matrix perturbation analysis, it is possible to set a lower bound on the amount of samples to recover the largest eigenvalue [21]. Let be the noise power as σ^2, the modulus of the first Principal Component of $\tilde{\mathbf{\Sigma}}$ as $\|\mathbf{v}\|^2$ and, the signal to noise ratio needed to detect the largest eigenvalue as $\mathsf{SNR}_v = \|\mathbf{v}\|^2/\sigma^2$. The stochastic self-adjoint matrix $\tilde{\mathbf{\Sigma}}$ experiences a *phase transition* for M samples obtained from N sensors s.t. $M/N \geqslant \mathsf{SNR}_v^{-2}$ (Eq. (2.19) in [21]). During phase transition the $\hat{d}$ largest eigenvalues "collapse" from noise to signal subspace eigenvalues. This means that the minimum amount of samples M needed to detect the largest eigenvalue of signal to noise ratio SNR_v is

$$M \geqslant \frac{N}{\mathsf{SNR}_v^2} \tag{5}$$

D. Second-Order Data-Coupled Clustering Algorithm (SODCC)

In this section, we present a self-organized data-coupled clustering algorithm; the algorithm and the sensed field are coupled by means of a second order statistic. The goal of the clustering algorithm is that nodes with higher spatio-temporal correlation between them belong to the same cluster. The algorithm's decision criterion is based on the dimension of the signal subspace of the dataset gathered by groups of nodes. To ensure a well-posed correlation matrix in each cluster, the necessary and sufficient condition is that the signal subspace dimension needs to be lower that the number of nodes.

Considering that the in-network processing algorithm is D-CPPCA, for the data-coupled clustering algorithm we hypothesize that: *for a cluster with N_i nodes and $M_i = N_i/\mathsf{SNR}_v^2$ measurements per node, and for $\hat{d} \leqslant N_i$ s.t. $\varphi_{\hat{d}} \leqslant \gamma_{\hat{d}}\, c(M)$, the dimension of the signal subspace for that cluster, the CP-PCA algorithm solves a well-posed problem* (and the dataset can be compressed). We propose a two stage algorithm that: 1) initializes random CHs in the LS-WSN and 2) grows the clusters until enough data is gathered to ensure convergence.

1) First Stage: Consider a LS-WSN with N role-free nodes (nodes that are neither CH nor sensor nodes). In the first stage (see Algorithm 1), role-free nodes decide randomly to turn into CH, with an *a priori* probability P—independently and in a fully distributed fashion. The new CHs request their first neighbours (nodes that interact with a one hop communication) to form part of their clusters; only role-free nodes answer. Each CH assigns the role of sensor nodes to its selected neighbours and finishes the cluster initialization. Role-free nodes repeat Algorithm 1 once a timeout T_1 is reached. Once a larger timeout $T_2 \gg T_1$ is reached, all remaining role-free nodes turn into CH and the first stage is finished.

Algorithm 1 First stage: CLUSTER INITIALIZATION

1: Random decision to turn into CH
2: **if** CH **then**
3: REQUEST to first neighbours (answer from role-free nodes)
4: UNION message to selected neighbours
5: **end if**

Two key design decision must be taken: i) criteria to select neighbours (e.g., received signal strength; transmission delay; first-come, first-served) and, ii) maximum cluster size for the first stage (N^{1st}). The maximum cluster size is needed as no data structure is considered in this stage and the only aim is the initial cluster seeding. However, if any CH receives less answers from neighbours than the maximum allowed, that cluster is established with less nodes. Therefore, at the end of the first stage are expected clusters with sizes $1, 2, 3, \ldots, N^{1st}$.

A possible alternative implementation of this first stage can be done by means of the Rapid or Persistent algorithms [10] with a budget of $B = N^{1st}$ nodes.

Finally, all parameters used in the first stage of the algorithm are independent of the data and do not interfere with the final

network configuration, as they are only used to obtain a random initial cluster seeding.

2) Second Stage: The key for the "growing" phase of SODCC (see Algorithm 2) is: as the cluster gathers data (at least M_i measurements per node), it computes the signal subspace dimension $\hat{d}$. If FSD is unable to determine a dimension smaller than cluster size, an aggregation of neighbouring clusters is mandated by the CH; otherwise, the cluster stops growing. The minimum M_i samples needed for CPPCA problem to be well-posed can be tuned as a function of the minimum required SNR_v.

Algorithm 2 Second stage: CLUSTER GROWING

```
1: M_i = N_i × SNR_v^{-2}
2: if CH then
3:    WAIT for M_i measurements per node
4:    d̂ ← FSD dimension estimation of cluster data
5:    if d̂ ⩾ N_i then
6:       FUSION with selected CH; decision of new CH
7:       if new CH then
8:          Gather all data from the other CH; update N_i
9:       end if
10:   end if
11: else
12:   Send data to CH
13: end if
```

Clusters with high spatio-temporal correlation estimate a dimension lower than the cluster size ($\hat{d} < N_i$), i.e., the dimension of the noise subspace is at least 1 and signal and noise subspaces are separable. In this case, the cluster size fulfills the clustering algorithm convergence criterion. On the other side, clusters with low spatio-temporal correlation do not meet the convergence condition in Eq. (3), i.e., signal and the noise subspaces are non separable. In this second case, the cluster must grow in size (fusion with another cluster) to achieve the convergence criterion; fusion is mandatory for the selected cluster, even if it has already met the convergence condition.

The aggregation criterion impacts on the final cluster configuration. SODCC aims to increase the spatio-temporal correlation of the gathered data in each cluster. This objective is achieved when the aggregation criterion seeks to, e.g., minimize the distance between CHs, minimize the cluster area, maximize the quality of the wireless channel between the CHs. SODCC is flexible enough such that it allows additional constraints to be included into the aggregation criterion (e.g., energy efficiency).

Node synchronization is not a requirement for SODCC. Every node starts the algorithm just after it is switched on. If new nodes are incorporated to the LS-WSN, they start the procedure anew.

III. PERFORMANCE ANALYSIS OF THE SODCC ALGORITHM

In this section we perform a mathematical analysis of the expected performance of SODCC. The main performance metrics in LS-WSNs, such as transport capacity, end-to-end delay or jitter, are directly related to the number of clusters and the cluster size distribution. First, we propose an analytical expression for the cluster size distribution for the stationary state, i.e. at the end of the second stage of SODCC, using the dynamic scaling *ansatz*. To prove the scalability of SODCC, we compute the average cluster size and analyze it in a LS-WSN scenario, using the newly proposed analytical expression. Next, we compute the Variance to Mean Ratio (VMR) to further analyze the operation of SODCC and the variability of cluster sizes obtained in order to obtain the new capacity limits of the network. Finally, we explore the drawbacks of the present formulation, in terms of stationarity of the data and multi-sensor capabilities, and propose solutions.

A. Statistical Distribution of Cluster Sizes

To obtain an analytical expression for the statistical distribution of the cluster sizes, an apparently trivial observation must be made: What is the relation between cluster sizes that result from SODCC? Due to the random initial cluster distribution (first stage) and due to the coupling of the resulting clusters to the data-field (second stage), cluster size power-laws must emerge i.e., there are exponentially less large clusters than small clusters. And this fact holds at the end of the first stage of SODCC, as well as throughout the second stage.

The *ansatz* of dynamic scaling for aggregation of clusters [22] allows the formulation of the cluster size distribution and its long-term evolution. This distribution can be then described by a dynamic scaling function of the form

$$n(t, N_i) \sim t^{-w} N_i^{-\tau} f\left(\frac{N_i}{t^z}\right) \tag{6}$$

where dependent variables t and N_i are time and cluster size, respectively. The three terms present in the equation describe: 1) the aggregation process through a power-law, with $w > 0$, 2) the stationary distribution through a different power-law, with $0 < \tau < 2$, and 3) the characteristic cluster size, with $z > 1$. The scaling function $f(x)$ has power-law behavior for small x, i.e., $x^{2-\tau}$ for $x \ll 1$ and $f(x) \ll 1$ for $x \gg 1$.

According to Eq. (1) in [22], in a close-to-stationary state where the cluster size N_i is small with respect to time ($N_i/t^z \ll 1$),

$$n(t, N_i) \sim t^{-w} N_i^{-\tau} \tag{7}$$

with $w = (2 - \tau)z$. This condition establishes the relation between the parameters of Eq. (6) and it is fulfilled with constant number of nodes in the network (N). Similar scaling relations have been used in the application of percolation theory to evaluate the capacity of wireless ad-hoc networks [23].

Eq. (7) provides a power-law dependence on the cluster size from the smallest cluster size possible (which is 2) to infinity. As infinitely large WSNs are impractical, we assert that the cluster size distribution for the stationary state and a WSN with N sensor nodes has to fulfill the following condition:

$$\sum_{N_i=1}^{N} n(t, N_i) = 1 \tag{8}$$

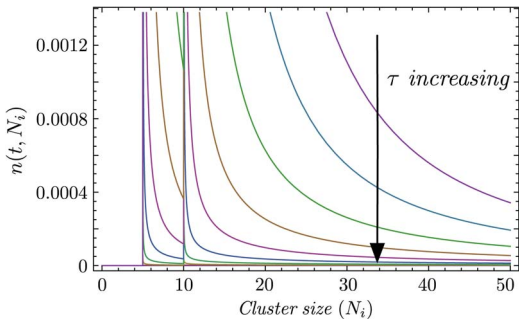

Fig. 3. Cluster size distribution according to Eq. (9) with $t = 50$ and $N = 50$; $\hat{d}_{\mathsf{min}} = 5$, $\hat{d}_{\mathsf{max}} = 10$, $0.1 \leqslant \tau \leqslant 2$ and $z = 1.40$.

The time dynamics of the network partition, governed by the second stage of SODCC, are quantized at the characteristic times $t^{(j)} = 4 \times j$ ($j = 1, \ldots, N_i^{max}$, being $N_i^{max} = \max(N_i)$. At these moments the phase-transition condition for matrix $\tilde{\mathbf{\Sigma}}$ is fulfilled and the FSD statistic is computed, and a fraction of the clusters with size N_i transition to clusters with size, at least $N_i + 2$.

Let $\tilde{\mathbf{\Sigma}}_{N\times N}^{(j)}$ be the covariance matrix of the entire WSN, evaluated at the characteristic time $t^{(j)}$. Consider that for each matrix available, the signal subspace dimension $\hat{d}^{(j)}$ is estimated. The probability mass function of the resulting $\hat{d}^{(j)}$ has a compact support in $j \in \mathbb{N}$ and models the spatio-temporal correlations of the gathered data. The minimum ($\hat{d}_{\mathsf{min}}$) and maximum values ($\hat{d}_{\mathsf{max}}$) mark the "origin" of separate cluster size distributions that decay with a power-law with the same rate.

Finally, we formulate the cluster size distribution in the stationary state as:

$$n(t, N_i) = \kappa \times t^{-w} \left(g(N_i, \hat{d}_{\mathsf{min}}) + g(N_i, \hat{d}_{\mathsf{max}}) \right) \tag{9}$$

where

$$g(x, x_0) = \theta(x - x_0)(x - x_0)^{-\tau} \tag{10}$$

$\theta(x_0)$ is the Heaviside step-function and

$$\kappa = \frac{\tau - 1}{(N - \hat{d}_{\mathsf{min}})^{1-\tau} + (N - \hat{d}_{\mathsf{max}})^{1-\tau} - 2} \tag{11}$$

is the normalization constant s.t. Eq. (8) is satisfied. To show the shape of the cluster size distribution, we represent it in Fig. 3 with $\hat{d}_{\mathsf{min}} = 5$, $\hat{d}_{\mathsf{max}} = 10$, $N = 50$, and $0.1 \leqslant \tau \leq 2$.

B. Scalability of SODCC

To assess the scalability of SODCC, we further analyze Eq. (9) in a LS-WSN scenario. First, we compute the average value of the cluster size ($\langle N_i \rangle$) to check whether if SODCC is scalable, and second, we compute the VMR to analyze the behavior of the cluster sizes in this scenario.

Therefore, the average value of the cluster size is

$$\langle N_i \rangle = \int_0^N n(t, N_i)\, N_i\, dN_i \tag{12}$$

$$= \frac{1}{\tau - 1} \left[\hat{d}_{\mathsf{min}} + \hat{d}_{\mathsf{max}} - 2 + \frac{1}{\tau - 2} \frac{\hat{d}_{\mathsf{min}} - N(\tau - 1)}{\left(N - \hat{d}_{\mathsf{min}}\right)^{\tau-1}} + \frac{1}{\tau - 2} \frac{\hat{d}_{\mathsf{max}} - N(\tau - 1)}{\left(N - \hat{d}_{\mathsf{max}}\right)^{\tau-1}} \right] \tag{13}$$

Eq. (13) shows that both the minimum and maximum signal subspace dimensions contribute on $\langle N_i \rangle$, apart of the intuitive contributions of N and τ. In this expression, as we approach to the large-scale ($N \gg \hat{d}_{\mathsf{max}}$) and the stationary ($t \to \infty$) states, the correlations in the data grow at a smaller pace than the average cluster size does ($N \gg (\hat{d}_{\mathsf{max}} - \hat{d}_{\mathsf{min}})$), so mean cluster size tends to

$$\langle N_i \rangle \sim \frac{2}{\tau - 1} \left[\hat{d}_{\mathsf{max}} - 1 + \frac{1}{\tau - 2} \frac{\hat{d}_{\mathsf{max}} - N(\tau - 1)}{(N - \hat{d}_{\mathsf{max}})^{\tau-1}} \right] \tag{14}$$

And in the largest scale state, when the WSN is so large that the data correlations begin to be irrelevant, the mean cluster size is simplified to

$$\langle N_i \rangle \sim \frac{2}{\tau - 1} + \frac{2N^{2-\tau}}{2 - \tau} \tag{15}$$

For a clustering algorithm to be scalable, $\langle N_i \rangle$ must grow at similar rate as N. From Eq. (15), we can clearly see that this is the case when $\tau \to 1$. This parameter depends on the initial localization and connectivity of the nodes and it merits further investigation of the conditions under which SODCC seems to exhibit *super-scalability* i.e., $\langle N_i \rangle$ grows at higher rate than N. In a super-scalability case, the amount of clusters grows at a smaller rate than the network size and novel results on the transport capacity of the wireless networks [1] should arise.

Finally, measuring the VMR of $n(t, N_i)$ in a LS-WSN scenario,

$$\mathsf{VMR} = \frac{\int_0^N n(t, N_i)\, N_i^2\, dN_i}{\int_0^N n(t, N_i)\, N_i\, dN_i} \tag{16}$$

$$\sim \left(\frac{2 - \tau}{3 - \tau} \right) \left(\frac{N^3 - N^\tau}{N^2 - N^\tau} \right) \tag{17}$$

we can see that it tends to infinity as $N \to \infty$. This expression shows that the variability in the cluster sizes have linear dependence with the number of nodes N in the WSN. So, SODCC will partition the LS-WSN in clusters with diverse sizes, rather than in clusters of a typical size of $\langle N_i \rangle$.

Finally, it is obvious from the formulation above that the aggregation exponent τ is solely determined by the second

stage of SODCC. This fact puts SODCC in the same class as cluster aggregation by diffusion [22]. Due to the universality of critical phenomena [24], SODCC has a critical point ($\tau = \tau_c$). The Self Organized Critical behaviour should emerge with a nonlinear response function. Such function applied at node level allows for nonlinear transduction of the sensed physical field and also for a nonlinear feature extraction algorithm at cluster level. The optimization of operational parameters in this setting gives way to optimal information transmission [25] and computational capabilities at both cluster and network level [26].

C. Message Complexity

In this section we analyze the number of messages that are needed to stablish for SODCC a single cluster of size N_i in a network of size N, i.e., the message complexity of the clustering algorithm. It is customary in network partitioning algorithms to evaluate a worst-case scenario for each specific setting. For SODCC, the worst-case is realized when the outcome of the first stage is a number of clusters equal to N. Then, cluster fusion takes place, in each characteristic time, among clusters of equal size. The final cluster size distribution should then result in a delta function (e.g., when the $\hat{d}_{\mathsf{min}} = \hat{d}_{\mathsf{max}}$ and N is large). Then, it can be shown that the amount of messages to form a single cluster is, at most, $4 \times (N_i^2 + 1)$ and, therefore, its message complexity is $O(N_i^2)$ (See Appendix).

As a comparison, we can see that other SOAs exhibit similar or worse message complexity. In the case of the Persistent algorithm [10], the number of messages to form a single cluster in the worst-case scenario is $2 \times N_i^2$ and its message complexity is $O(N_i^2)$. Other algorithms exhibit even worse message complexity, such as the Expanding Ring algorithm which has a message complexity of $O(N)$ [27].

D. Limitations and Solutions

In the present formulation, the SODCC algorithm has two main drawbacks: 1) lack of adaptability to non-stationary data statistics, and 2) lack of support for multiple sensors and data types. With respect to the first, it can be sidestepped with a straightforward adaptation of the SODCC algorithm, i.e., by restarting the first stage of SODCC from time to time. There are several ways to define the reset time. For example, by using the largest meaningful time dynamics of the measured data (t_{dyn}), the dataset can be split in a sequence batches (e.g., for temperature it could be a seasonal time-period). Then, at the beginning of each data batch, the SODCC algorithm restarts the first stage and partitions the network according to the new data characteristics. Another option is to use the convergence time of the SODCC algorithm t_{conv} as reset time, i.e., the time until the variation of the cluster density is lower than a given threshold ε. Based on this definition and using Eq. (9), the convergence time can be calculated as

$$\left| \frac{1}{n(t, N_i)} \frac{\partial n(t, N_i)}{\partial t} \right|_{t=t_{\mathsf{conv}}} \leqslant \varepsilon \tag{18}$$

and, therefore, the convergence time of the SODCC algorithm, t_{conv}, can be expressed as

$$t_{\mathsf{conv}} = \frac{w}{\varepsilon} \tag{19}$$

Although both t_{dyn} and t_{conv} encode the time dynamics of the data, the former is to be equal or lower that the latter to allow for the stabilization of the clustering configuration and to prevent constant reconfiguration of the network.

However, a more efficient strategy in terms of algorithm complexity would be to allow for a converged cluster configuration to evolve in time while keeping track of the non-stationary characteristics of the data. Thus, we would remove the need to restart SODCC after each data batch. In practical terms, this solution should allow clusters to grow, diminish or dissolve according to the time-dependent data statistics. This is out of the scope of the present work and will be addressed in the future.

With respect to the second of the drawbacks, the multiple sensor capabilities, it has to be taken into account that S different measured physical variables will most assuredly exhibit different spatio-temporal correlations. Therefore, the application of the FSD algorithm to the different physical variables will result in different $\hat{d}^{\,i}_{\mathsf{min}}$ and $\hat{d}^{\,i}_{\mathsf{max}}$ for all the different $i = 1, \ldots, S$ sensors. In this situation, an *embedded layers* approach is a possible strategy that can be considered, where the minimum $\hat{d}^{\,i}_{\mathsf{min}}$ determines the unique cluster configuration obtained. This configuration resolves the smallest spatio-temporal correlations present in any of the measurements at the cost of a small scalability loss, as Eq. (13) depends (at first order) linearly on $\hat{d}_{\mathsf{min}}$. Moreover, the measurements of the data with less variation will be affected by an increase in the measurement noise, due to the lower than needed cluster sizes.

An alternative strategy is the *independent layers* approach. In this case, the SODCC is applied independently to each of the different physical variable measured and it results in concurring cluster configurations that share initialization but differ in almost every other aspect. This approach benefits from scalability in all the measured physical variables and is suitable for WSNs where the sensor nodes measure different subsets of all the physical variables. On the other hand, the total computational burden is increased $n-$th fold as the problem is solved by parallelization with no apparent gains. The question that arises in this situation is how concurring cluster configurations could operate simultaneously in a resource-limited environment such as a WSN. Furthermore, the optimal operation of SODCC in each layer cannot be presumed of achieving optimality in the aggregate. Therefore, optimizing the use of spatio-temporal cross-correlations among the sensed variables could result in a minimum overall reconstruction error at controlled scalability.

However, the most faithful approach to the present problem with a WSN with N nodes equipped with S different sensors seems to rely on a tensor representation. Thus, the original data matrix $\mathbf{X}$ is substituted by the data tensor $\hat{X} \in \mathbb{R}^{M \times S \times N}$

$$\hat{X} = \sum_{i=1}^{k} \pi_i x_i \otimes s_i \otimes w_i + \mathsf{noise} \tag{20}$$

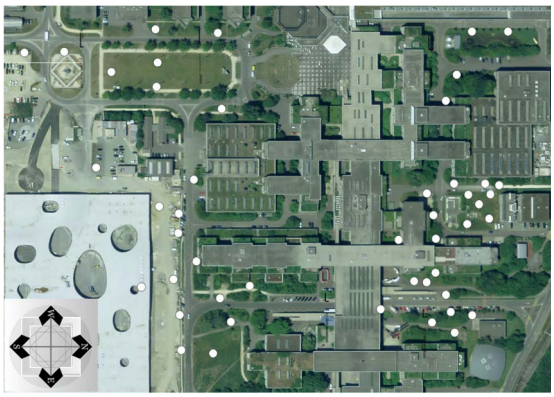

Fig. 4. Localization in a scaled map of the $N = 47$ sensor nodes used in the computer simulations. Reproduced by permission of swisstopo (BA13063).

where $x_i \in \mathbb{R}^M$ represents the data set from the sensors, $s_i \in \mathbb{R}^S$ represents the set of the different type of sensors (temperature, humidity, light...), $w_i \in \mathbb{R}^N$ is the set of wireless nodes and $\pi_i \in \mathbb{R}^k$ are factor weights. The direct (tensor equivalent) application of random projections procedure to the third order $\hat{X}$ tensor would lead to a dimensionality reduction of the tensor but not to an order reduction [28]. Apart from the fact that the FSD procedure has not been formulated for tensors, the inherent problem with this formulation is that most tensor problems are extremely complex from a computational point of view [29]. That could be a problem for CHs in clusters with a great number of nodes.

Random projections can also be used in a CPPCA-like procedure to reduce not the dimensionality but the order of the $\hat{X}$ tensor by projecting it into a set of matrices [30]. Those matrices can be treated as matrix $\mathbf{X}$ to recover their signal subspace dimension on a way similar to the embedded layers approach. Finally, simultaneous diagonalization of such matrices (now with the FSD algorithm) can be carried out in the DFC to produce estimates of the factors of the original tensor. These and other solutions will be addressed in future works.

IV. Computer Simulations and Results

In this section we analyze the performance of the D-CPPCA and SODCC algorithms through computer simulations. The dataset used is obtained from an actual LS-WSN data, in particular temperature data gathered by the LUCE deployment from the Sensorscope project [31]. From all the devices available in the aforementioned deployment, we selected the subset of $N = 47$ exterior sensor nodes plotted in Fig. 4 using white dots. The dataset used contains $M = 10^4$ samples per sensor (measured every minute for approximately one week) preprocessed: 1) missing data (4.08% of the total) and outliers (0.14% of the total) are replaced by the previous accurate value; 2) trend and seasonal components are extracted; and 3) the dynamic range is normalized to the domain $[-1,1]$.

We have previously used this dataset to test Centralized CP-PCA with flat network configuration as an in-network processing technique for LS-WSNs [20].

A. Space-Time Correlations of the Dataset

To evaluate the space-time correlations in the dataset, we estimate the dimension of the dataset using FSD over $\Sigma_{N\times N}$ with an increasing number of samples per sensor node (see Fig. 5). Matrix $\Sigma_{N\times N}$ is the covariance matrix considering the data from all the sensors forming the WSN. The dimension of the signal subspace has a monomodal statistical distribution centered in $\hat{d} = 6$, with $\hat{d}_{\mathsf{min}} = 2$ and $\hat{d}_{\mathsf{max}} = 8$. Therefore, partitions of the network are expected to exhibit similar dimensions of the signal subspace, but always limited by the cluster size. That is, clusters with N_i nodes are expected to exhibit signal subspace dimensions larger than 2 and smaller than $N_i - 1$.

B. Performance of SODCC

We carried out 10^5 independent simulations of the SODCC algorithm. For the fist stage we set the maximum cluster size $N^{1st} = 3$ (minimum cluster size to detect a signal subspace of dimension $\hat{d}_{\mathsf{min}} = 2$) and the probability to become CH in the initialization $P = 0.35$ (consistent with N^{1st}). The criteria used by the CHs in the present work is such that it minimizes the time delay response. Initial tests revealed that Algorithm 1 requires multiple iterations to assign roles to all the nodes in the WSN, so we set T_2 such that the algorithm iterates up to three times. For the second stage: 1) the lower bound of the signal to noise ratio of the largest eigenvalue is set to $\mathsf{SNR}_v = 0.5$, so $M_i \geqslant 4 \times N_i$; 2) the aggregation goal is to minimize the time delay response between CHs; and 3) the CH that started the fusion turns into a sensor node of the resulting cluster.

1) Temporal Evolution of the Clustering: First of all, we analyse the evolution of the cluster sizes along the second stage (first stage is only initialization): Fig. 6 displays the results with normalized histograms. Subfigures from left to right and top to bottom are for specific snapshots: when clusters with sizes $s = 1, \ldots, 8$ decide whether fusions are needed (blue proportion) or not (red proportion). The key fact that Fig. 6 reveals is: clusters formed by 9 (or more) nodes are stable—they don't need further fusions. We conclude that: 1) the temperature data indeed exhibit spatio-temporal correlations, and 2) the data-driven clustering algorithm captures the data correlations and properly stops.

The time evolution depicted in Fig. 6 is clearly not a stationary state. Condition $N_i/t^z \ll 1$ is not met and therefore, Eq. (9) is not applicable.

2) Stationary Cluster Size Distribution: We just mentioned the spatio-temporal correlation of temperature data; but is it small or large-scale correlation? And how does the clustering algorithm adapts to this dataset? Fig. 7 summarizes the final configurations obtained as a normalized histogram of the cluster sizes. The gray shaded area is the function defined in Eq. (9), the theoretical cluster size distribution for the stationary state. In this figure, $N = 47$, $\hat{d}_{\mathsf{min}} = 2$ and $\hat{d}_{\mathsf{max}} = 8$ are obtained from the used temperature dataset and $\tau = 1.13$ is the only adjustable parameter (according to the minimum square error criteria). In Fig. 7 we also show the spatial distribution of clusters with $N_i = 3$ and $N_i = 9$ nodes. We can see that medium-sized clusters coalesce around areas with large-scale

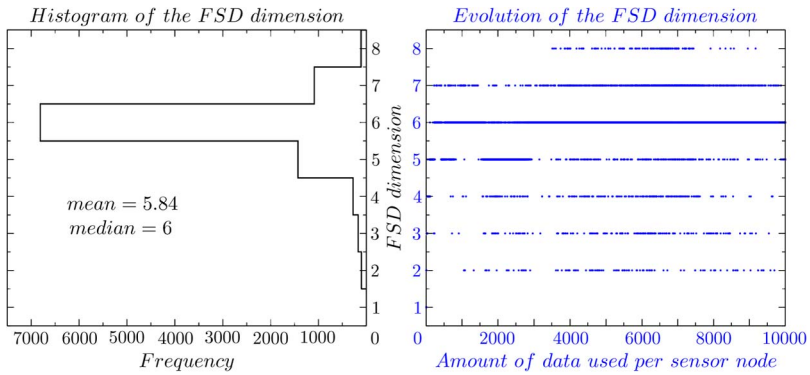

Fig. 5. Histogram of the signal subspace dimension of the dataset estimated with FSD (left). Time evolution of the signal subspace dimension estimated with FSD with an increasing number of samples per sensor node (right).

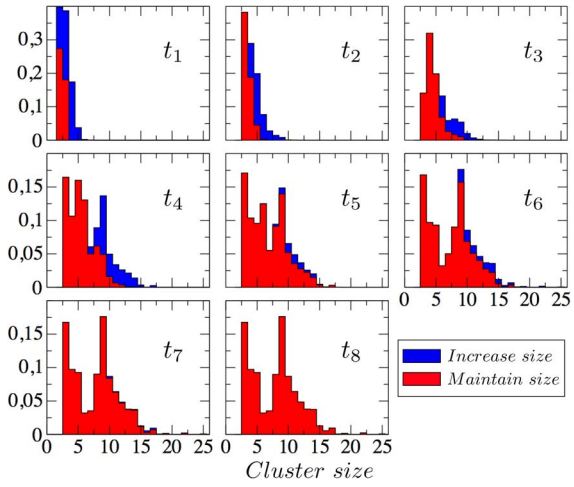

Fig. 6. Data-driven clustering algorithm during the second stage: normalized histograms of cluster sizes. From left to right and top to bottom, results about decisions taken by clusters with increasing sizes (from $s = 1$ to $s = 8$) at different times $t_s = 4 \cdot s$. Highlighted in blue is the proportion of clusters that grow in size.

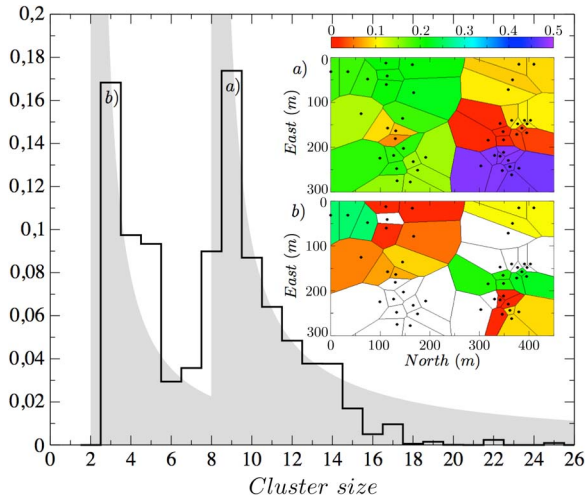

Fig. 7. Normalized histogram of cluster sizes of the final configuration (black solid line) and the theoretical cluster size distribution with $\tau = 1.13$ (gray shaded area). Spatial distributions of nodes and their respective probability to form clusters with $N_i = 9$ (a) and $N_i = 3$ (b) nodes (white represents zero probability).

correlations, while small-sized clusters form around areas with small-scale correlations. We conclude that: 1) the temperature data have both small and large-scale correlations, 2) the data-coupled clustering algorithm is able to adapt to both small and large-scale correlations, and 3) this adaptation leeds to a power law behavior for the cluster size distribution.

Finally, we plot the probability of all borders between clusters to appear in Fig. 8; zero probability borders are not plotted and reveal nodes that always belong to the same cluster. Comparing Fig. 8 with Fig. 4 the differences between the distribution of borders are self-explanatory: 1) buildings and large distances act as natural borders, 2) a high density of nodes does not always imply high correlation in the dataset, and 3) the SODCC algorithm adapts to the environment.

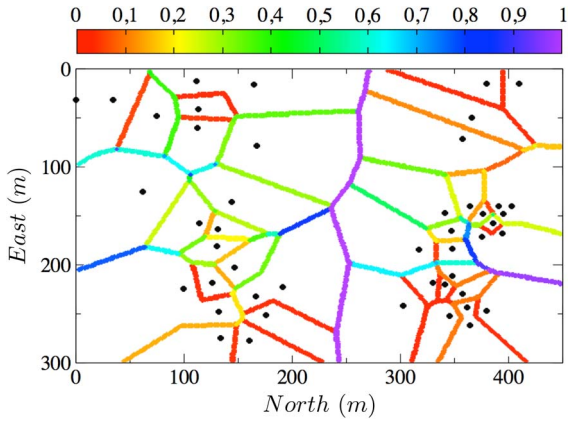

Fig. 8. Distribution of borders (separation between clusters). Absence of border indicates zero probability.

C. Field-Reconstruction Performance

In this section we evaluate the tradeoff between the compression ratio and the goodness of the data reconstruction for the SODCC and the D-CPPCA algorithms, within the framework presented in this work. For it, we apply the D-CPPCA coding/decoding procedure to all 10^5 configurations obtained by the data-coupled clustering algorithm. For each clustering configuration, K_i is varied between 2 and $N_i^{max}/2$, being

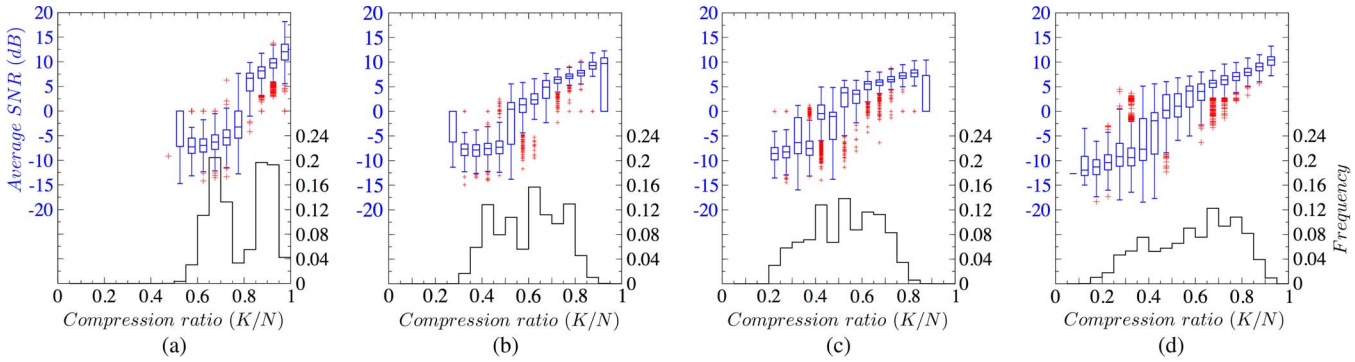

Fig. 9. Normalized histogram of the compression ratio K/N, plotted using the right side Y axis (continuous line). Box plot of the average SNR vs. K/N, plotted using the left side Y axis. We represent the performance of the Persistent algorithm for budgets of 3 (a), 6 (b) and 9 (c) sensor nodes per cluster. In Figure (d) we represent the performance of the SODCC algorithm. (a) Persistent $N_i = 3$ nodes; (b) persistent $N_i = 6$ nodes; (c) persistent $N_i = 9$ nodes; (d) SODCC.

$N_i^{max} = \max(N_i)$, resulting in more than 57.000 independent simulations. For the D-CPPCA encoding, we set parameter $J = 20$ since it exhibits a good tradeoff between accuracy and computational complexity [18]. For the CPPCA decoding, parameter L_i is chosen to maximize the average SNR between the reconstruction and the original data. In each simulation, both K_i and L_i do not vary throughout the different clusters. In the smallest clusters, however, no data compression is realized and therefore $K_i = L_i = N_i$. The total compression ratio (K/N) for the entire WSN in each simulation is evaluated as the mean value of the individual compression ratios (K_i/N_i) of each cluster, i.e.

$$\frac{K}{N} = \frac{1}{N_c} \sum_{i=1}^{N_c} \min\left(1, \frac{K_i}{N_i}\right) \tag{21}$$

To compare the performance of SODCC to other state-of-the-art SOAs, we also apply the D-CPPCA coding/decoding procedure to clustering configurations obtained with the Persistent algorithm [10], using the same parameters as described above. The Persistent algorithm, one of the most representative SOA, is decoupled from the measured field as the clusters self-organization does not depend on the data. In short, Persistent organizes the nodes in clusters of size as close as possible to N_i, being N_i a preset parameter indicating the node budget for each cluster. In this comparison, we consider three different budget parameters, namely $N_i \in \{3, 6, 9\}$, to sweep cluster configurations of interest to the present data set. These cluster configurations are prone to "resonate" with the spatio-temporal correlations of our dataset (see Section IV-A) and the budgets are closely related to the statistical behavior of the SODCC clustering algorithm.

1) Quantitative Evaluation of the Performance: From Eq. (21) we observe that selecting a compression ratio for a given cluster (fixing K_i) does not automatically fix the total compression ratio of the network to $K/N = K_i/\langle N_i \rangle$. In the bottom parts of each of the graphs of Fig. 9 we represent the normalized histograms of the resulting K/N for the three different Persistent settings and SODCC. Whereas in the former, the cluster size distribution has a single peak around value N_i, in the latter the cluster size distribution is shown in Fig. 7. The shapes of these histograms depend on the values of K_i used in the computer simulations. For example, consider a network configuration that has $N_c = 9$ clusters with the following cluster sizes: 3, 3, 3, 4, 4, 5, 6, 9 and 10. With this configuration, we varied K_i from 2 to 5 (as $N_i^{max} = 10$ in this particular case). The only case where all clusters obtain $K_i/N_i < 1$ is the one where $K_i = 2$, and in all the other cases there are several clusters for which $K_i/N_i = 1$ (as $K_i \geq N_i$).

At the top of Fig. 9 we represent the boxplot of the obtained average SNR for both the Persistent and SODCC algorithms. The quality metric for the reconstruction of the measured field that has been defined elsewhere [18]. Each compression ratio K/N corresponds to a distribution of average SNRs, obtained from multiple simulations and summarized by a boxplot representation. We see that a higher range for K/N is obtained with the network configurations outcomed by SODCC. This fact shows the ability of the SODCC algorithm to adapt to the data and also enables more K/N—average SNR possibilities to be achieved.

Analyzing the general behavior of the boxplots, we observe similarities between the results for Persistent and SODCC algorithms. Each of the four boxplots show two distinct behaviors separated by a *crossover point*: 1) a very-low SNR regime, where the compression rate is too high for the clustering configuration, not enough information is transmitted and the DFC is not able to decode the signal components; and 2) a linear regime, where the compression rate is appropriate for the clustering configuration, enough information is transmitted, and the DFC recovers all the signal components. Clearly, the linear regime is desirable in each of the four cases analyzed.

The differences between Persistent and SODCC are noticeable on the value of K/N for which the crossover point occurs. That is, whereas the crossover point for the Persistent with budget $N_i = 3$ is around $K/N \approx 0.8$, for $N_i = 6$ is $K/N \approx 0.5$ and for $N_i = 9$, it is further displaced to $K/N \approx 0.4$. The crossover point for SODCC is not as sharp as for Persistent and is located around $K/N \approx 0.3$. Lower values of K/N at the crossover point are desirable, as larger linear regime are obtained.

Moreover, analyzing the range of K/N and the values of average SNR obtained in each of the four cases, we observe that higher values of K/N lead to better resolution of the small-scale spatio-temporal correlations.

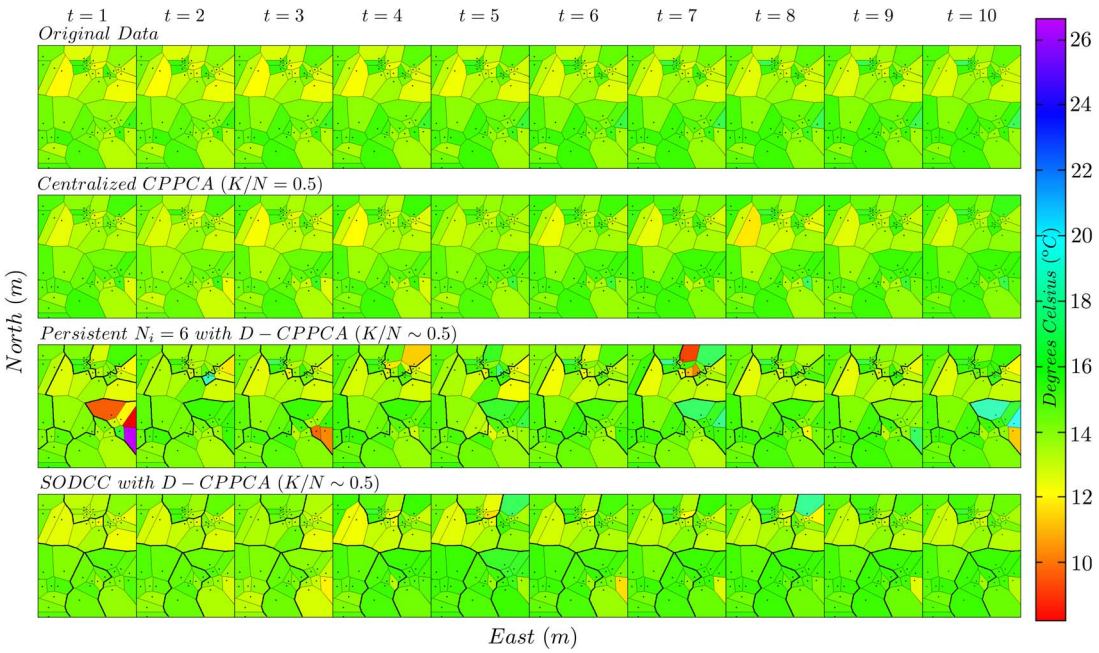

Fig. 10. Comparison of 10 consecutive snapshots of the original data of the measured temperature field with the Centralized CPPCA ($K/N = 0.5$), Persistent with $N_i = 6$ nodes and D-CPPCA ($K/N \sim 0.5$) and SODCC with D-CPPCA ($K/N \sim 0.5$) configurations. In the third and fourth row, the heavy lines indicate the borders of the resulting clusters for each algorithm. The colorbar range has been selected to stress the differences in performances among the algorithms. However, the typical differences are less than 2 °C.

2) Qualitative Evaluation of the Performance: The data gathered by all sensor nodes every minute can be seen as a snapshot of the temperature field. Similar to the evaluation of image processing algorithms, we now proceed to realize a qualitative evaluation of the performance of SODCC used with D-CPPCA.

In Fig. 10, we represent 10 consecutive snapshots of the temperature field, each identified by the parameter $t = 1, \ldots,$ 10 *min*. We show configurations with total compression ratio K/N of 50%, a setting with noticeable data compression gain. Rows in the figure represent:

- *Original data*—the actual dataset used in the computer simulations. This set of snapshots is used as a baseline for the performance evaluation.
- *Centralized CPPCA* ($K/N = 0.5$)—the data recovered in the DFC applying the Centralized CPPCA in-network processing technique with $K/N = 0.5$.
- *Persistent* $N_i = 6$ *with D-CPPCA* ($K/N \sim 0.5$)—the data recovered in the DFC using the Persistent clustering algorithm [10] with a budget of $N_i = 6$ nodes and using D-CPPCA as in-network processing algorithm, obtaining $K/N \sim 0.5$. The heavy lines indicate the borders of the clusters.
- *SODCC with D-CPPCA* ($K/N \sim 0.5$)—the data recovered by the DFC considering a representative output of the SODCC algorithm and using D-CPPCA as in-network processing algorithm, obtaining $K/N \sim 0.5$.

In this figure, we observe differences between the original data and the data recovered by the DFC. The best performance belongs to the Centralized setting, which is not a suitable configuration for data gathering in a LS-WSN from the energetic and computational points of view. The next best performance belongs to SODCC with D-CPPCA setting, as it is able to capture a higher amount of details and with very few errors.

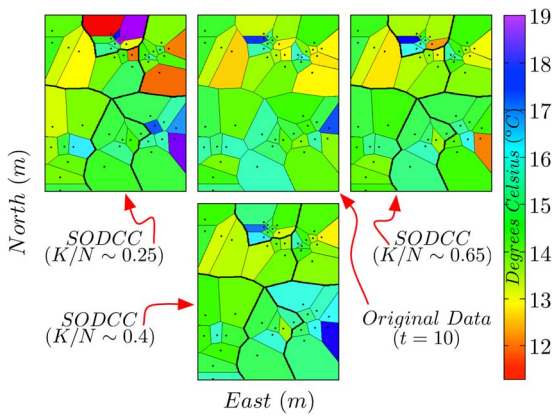

Fig. 11. Comparison of one snapshot of original data of the temperature field with the reconstruction obtained from three different configurations of SODCC with D-CPPCA. The heavy lines indicate the borders of the clusters. Increasing K/N are represented counter clockwise.

With this qualitative evaluation, we see that although the average SNR obtained has a low value (~ 5.2 *dB*), the snapshots of the reconstructed temperature field show the actual temperature with details well within the dynamic range of the image.

In Fig. 11 we evaluate the performance of SODCC with D-CPPCA considering different compression ratios (K/N). We use the same snapshot ($t = 10$ min) and the same clustering configuration as in Fig. 10. In this figure, we corroborate

TABLE I
COMPARISON BETWEEN THE LEACH CLUSTERING ALGORITHM AND THE SODCC WITH D-CPPCA ALGORITHMS WITH $K/N \sim \{0.3, 0.5\}$ IN TERMS OF AVERAGE ENERGY CONSUMPTION AND AVERAGE SNR OBTAINED

Configuration	Transmission	Reception	Processing	$Average\ SNR\ (dB)$		
				Min	Mean	Max
$LEACH\ with\ p = 0.35$	423.89%	0.52%	0%	0.01	0.41	0.81
$SODCC\ with\ D-CPPCA\ (K/N \sim 0.3)$	77.63%	99.60%	59.89%	-16.43	-8.43	-2.06
$SODCC\ with\ D-CPPCA\ (K/N \sim 0.5)$	100%	100%	100%	-5.87	1.24	6.82

that the reconstruction error decreases as the total compression ratio increases. Incremental improvements from localized clusters are cumulative for increasing values of K/N. For example, we see that the differences with the original data in the top center cluster in the $K/N \sim 0.25$ configuration are corrected in the $K/N \sim 0.4$ configuration. These corrections are further improved with the $K/N \sim 0.65$ configuration, while corrections to the lower right cluster are incorporated into the reconstruction.

In conclusion, the data-coupled strategy of SODCC actually provides benefits against a traditional SOA. We showed that both the range of possible compression ratios K/N and the linear regime for the average SNR obtained with the SODCC algorithm are larger, compared to the obtained with the Persistent algorithm. Moreover, the SODCC clustering algorithm does not need prior information about the LS-WSN and the characteristics of the data, unlike Persistent that needs an appropriate N_i value. Furthermore, we can qualitatively see that the selection of the target value of K/N involves a tradeoff between the quality of the reconstruction and reduction in the amount of packets transmitted through the WSN.

D. Energy Consumption

In this section we want to evaluate the energy consumption behaviour of our proposal. For this purpose, we select the baseline energy efficient network partitioning algorithm LEACH [8]. To explicitly evaluate the gains of the in-network processing *vs.* raw data transmission tradeoff, we split our computation into: 1) transmission consumption, 2) reception consumption, and 3) processing consumption. In other words, we estimate the consumption of the transmission power amplifiers (the major power sink in a sensor node), of the electronic components of the physical layer that receive the data (equalizers, filters, amplifiers, etc...) and of the processor while executing the different sequence of instructions of the different algorithms. These values are computed, respectively, using the following expressions:

$$tx = c_t \times \#\text{bits transmitted} \times d^{\eta} \tag{22}$$

$$rx = c_r \times \#\text{bits received} \tag{23}$$

$$proc = c_p \times \#\text{instructions executed} \tag{24}$$

where c_t, c_r and c_p are constants unique to each device that relate the calculated values to the actual consumption. This comparison is technology and platform independent, as we do not specify any particular device ($c_t = c_r = c_p = 1$). Regarding the transmission consumption, we take into account the link distance and an attenuation exponent (e.g., $\eta = 3$) to ensure constant quality of the link.

We estimated the energy consumption for the following configurations:

- *LEACH with* $p = 0.35$—the clustering is performed using the LEACH algorithm with the probability of a node turning into CH of $p = 0.35$ in each round. This value is identical to the one used in the first stage of SODCC in this work. Regarding the data gathering, the only measured data transmitted to the DFC by each cluster is the one measured by the CH. As the CH is changed every round, the DFC receives snapshots of the actual temperature field, sampled in different locations over the time. By considering this configuration, we aim at reducing the energy consumption to a minimum, as LEACH is an energy efficient clustering algorithm and as this data gathering technique (only the CH reports its measurements) is one of the simplest. Finally, as this is an algorithm with very low computational burden, we consider that the processing consumption is negligible.
- *SODCC with D-CPPCA* ($K/N \sim 0.3$)—the clustering is performed using the SODCC algorithm and the in-network processing is performed using the D-CPPCA algorithm, with compression ratio of $K/N \sim 0.3$. We selected this case as the final compression ratio is similar to the compression ratio obtained by the previous LEACH configuration.
- *SODCC with D-CPPCA* ($K/N \sim 0.5$)—similar to the previous configuration, except for the compression ratio, being $K/N \sim 0.5$ in this case.

We use this last configuration as the baseline for the energy consumption comparison.

The energy consumption results are shown in Table I. The obtained results for the two SODCC and D-CPPCA configurations confirms the logic of "more compression, less energy consumption," as less packets are transmitted between CHs and DFC and less processing is performed. Regarding the consumption of the LEACH algorithm, the value for the reception consumption is to be expected, as only control messages involved in the clustering are transmitted between sensor nodes and CH. On the other side, the transmission consumption is strikingly high, being four times higher than the baseline configuration. This fact has a simple explanation and it is one of the main drawbacks of LEACH: the communication between all nodes has to be single-hop. Therefore, all nodes must be able to communicate with the most distant node, thus spending a lot of energy in the process. The balance between in-network processing and raw

data transmission is clearly resolved in favour of the SODCC with D-CPPCA.

Moreover, in Table I we also show the average SNR obtained in each of the used configurations, to also analyze the tradeoff between energy consumption and data quality. These results show that it can be very beneficial to invest the available energy to locally process the measured data, from both network lifetime and data quality points of view.

V. Conclusions and Future Work

In this work we proposed a general framework for LS-WSN that uses a combination of in-network processing and clustering algorithms to obtain a self-organized and data-coupled network configuration. The criterion to form the clusters ensures the well-posedness of the in-cluster signal processing.

To demonstrate the general framework we developed a clustering algorithm which is based in a second-order statistic, SODCC, and a distributed, compressing algorithm which maximizes variance in the resolved components, D-CPPCA. The condition to form a cluster is that D-CPCCA does not fail to resolve the Principal Components in any given cluster.

We performed the theoretical analysis of SODCC using the *ansatz* of dynamic scaling of the cluster sizes. We were able to obtain an analytical expression for the distribution of the cluster sizes, as function of the network size and the signal subspace dimension. We also performed a comparative analysis of the message complexity of SODCC. We showed that SODCC is scalable and has similar or better message complexity than other well-known self-organizing algorithms. We also listed the main drawbacks of the present SODCC formulation and proposed several solutions to sidestep them.

We validated the theoretical findings via extensive computer simulations using an actual LS-WSN, the LUCE dataset and node localizations. These simulations also revealed the ability of SODCC to adapt the cluster spatial distribution and size to the measured data, in contrast to the performance of state-of-the-art SOA. Furthermore, we showed that SODCC, with D-CPPCA, allows for fine tuning of the compression rate. This tuning achieves a tradeoff between the reconstructed data quality and the amount of transmitted messages. The performance of SODCC in average SNR of the reconstruction is, comparative to other SOA algorithms, better at any compression rate. Furthermore, it needs no prior adjustment of any parameter. Finally, we showed that investment of the available energy in each node to locally process the measured data can be favorable from both network lifetime and data quality points of view.

In future works, we will undertake the optimization of SODCC. The theoretical analysis showed that it is possible to modify the aggregation clustering rate in the second stage of SODCC to achieve optimal scaling behaviour (superscalability). Moreover, SODCC is but one of the possible formulations for a data-coupled SOA. Different in-network processing algorithms that allow the data-coupling based on different statistics (e.g., higher order statistics or entropy based measures) may overcome identified shortfalls of SODCC, including 1) network re-configuration to adapt to non-stationary sensed-field, or 2) unbalanced computational burden between the CH and the sensor nodes. These shortcomings will be addressed in further developments.

Appendix

Let be $N_i = 2^K$, with $K \in \mathbb{N}$, the size of the analyzed cluster, formed in the worst-case scenario for SODCC.

To form a cluster with size N_i, the CH needs $4 \times N_i^2$ data values to compute the FSD dimension ($4 \times N_i$ values per node). Therefore, there are

$$4 \times (N_i - 1)^2 \tag{25}$$

data messages transmitted between the sensor nodes and the CH (the CH does not to transmit its own data to itself).

In addition, in each characteristic time $t^{(k)} = 4 \times k$, being $k = 1, \ldots, K$, through the network are transmitted

$$4 \times k \times \frac{N_i}{2^k} \tag{26}$$

data messages between the CH that needs the fusion to the CH that accepts the fusion.

Therefore, the total amount of data messages transmitted through the network to form a cluster of size N_i is

$$T_m = 4 \times (N_i - 1)^2 + \sum_{k=i}^{K} 4 \times k \times \frac{N_i}{2^k} \tag{27}$$

$$= 4 \times (N_i - 1)^2 + 4 \times N_i \times \sum_{k=i}^{K} \frac{k}{2^k} \tag{28}$$

$$< 4 \times (N_i - 1)^2 + 4 \times N_i \times 2 \tag{29}$$

$$= 4 + 4 \times N_i^2 \tag{30}$$

References

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

[2] H. El Gamal, "On the scaling laws of dense wireless sensor networks: The data gathering channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 1229–1234, Mar. 2005.

[3] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, no. 14/15, pp. 2826–2841, Oct. 2007.

[4] T. D. Le, N. D. Pham, and H. Choo, "Towards a distributed clustering scheme based on spatial correlation in WSNs," in *Proc. IWCMC*, 2008, pp. 529–534.

[5] P. Wang, R. Dai, and I. F. Akyildiz, "Collaborative data compression using clustered source coding for wireless multimedia sensor networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[6] C.-C. Hung, W.-C. Peng, and W.-C. Lee, "Energy-aware set-covering approaches for approximate data collection in wireless sensor networks," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 1993–2007, Nov. 2012.

[7] Z. Liu, W. Xing, B. Zeng, Y. Wang, and D. Lu, "Distributed spatial correlation-based clustering for approximate data collection in WSNs," in *Proc. IEEE 27th Int. Conf. AINA*, 2013, pp. 56–63.

[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Hawaii Int. Conf. Syst. Sci.*, 2000, pp. 1–10.

[9] S. A. Sert, H. Bagci, and A. Yazici, "MOFCA: Multi-objective fuzzy clustering algorithm for wireless sensor networks," *Appl. Soft Comput.*, vol. 30, pp. 151–165, May 2015.

[10] R. Krishnan and D. Starobinski, "Efficient clustering algorithms for self-organizing wireless sensor networks," *Ad Hoc Netw.*, vol. 4, no. 1, pp. 36–59, Jan. 2006.

[11] S. Park, K. Shin, A. Abraham, and S. Han, "Optimized self organized sensor networks," *Sensors*, vol. 7, no. 5, pp. 730–742, May 2007.
[12] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.
[13] X. Zixiang, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sep. 2004.
[14] R. Puri, A. Majumdar, P. Ishwar, and K. Ramchandran, "Distributed video coding in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 94–106, Jul. 2006.
[15] Y. Liang, M.-F. Balcan, and V. Kanchanapally, "Distributed PCA and k-means clustering," in *Proc. Big Learn. Workshop NIPS*, 2013, pp. 1–8.
[16] G. Xu and T. Kailath, "Fast subspace decomposition," *IEEE Trans. Signal Process.*, vol. 42, no. 3, pp. 539–551, Mar. 1994.
[17] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.
[18] J. E. Fowler, "Compressive-projection principal component analysis," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2230–2242, Oct. 2009.
[19] B. N. Parlett, *The Symmetric Eigenvalue Problem*. Philadelphia, PA, USA: SIAM, 1980, vol. 7.
[20] M. I. Chidean, E. Morgado, J. Ramiro-Bargueño, and A. J. Caamaño, "Self-organized distributed compressive projection in large scale wireless sensor networks," in *Proc. IEEE 24th PIMRC*, 2013, pp. 2000–2004.
[21] B. Nadler, "Finite sample approximation results for principal component analysis: A matrix perturbation approach," *Ann. Statist.*, vol. 36, no. 6, pp. 2791–2817, 2008.
[22] T. Vicsek and F. Family, "Dynamic scaling for aggregation of clusters," *Phys. Rev. Lett.*, vol. 52, no. 19, pp. 1669–1672, 1984.
[23] M. Franceschetti, O. Dousse, D. N. C. Tse, and P. Thiran, "Closing the gap in the capacity of wireless networks via percolation theory," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 1009–1018, Mar. 2007.
[24] P. C. Hohenberg and B. I. Halperin, "Theory of dynamic critical phenomena," *Rev. Mod. Phys.*, vol. 49, no. 3, pp. 435–479, Sep. 1977.
[25] J. M. Beggs and D. Plenz, "Neuronal avalanches in neocortical circuits," *J. Neurosci.*, vol. 23, no. 35, pp. 11 167–11 177, Dec. 2003.
[26] R. Legenstein and W. Maass, "Edge of chaos and prediction of computational performance for neural circuit models," *Neural Netw.*, vol. 20, no. 3, pp. 323–334, Apr. 2007.
[27] C. Ramamoorthy, A. Bhide, and J. Srivastava, "Reliable clustering techniques for large, mobile packet radio networks," in *Proc. IEEE INFOCOM*, 1987, pp. 218–226.
[28] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, May 2011.
[29] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-Hard," *J. ACM*, vol. 60, no. 6, pp. 45:1–45:39, Nov. 2013.
[30] V. Kuleshov, A. Chaganty, and P. Liang, "Tensor factorization via matrix factorization," *J. Mach. Learn. Res.: W&CP*, vol. 38, pp. 507–516, 2015.
[31] F. Ingelrest *et al.*, "SensorScope: Application-specific sensor network for environmental monitoring," *ACM Trans. Sens. Netw.*, vol. 6, no. 2, pp. 1–32, 2010.

Mihaela I. Chidean received the B.Sc. degree in telecommunication engineering and computer systems engineering from Rey Juan Carlos University (URJC), Fuenlabrada, Spain, in 2011 and the M.Sc. degree in multimedia and communications from the Carlos III University of Madrid, Madrid, Spain, in 2013. She is currently working toward the Ph.D. degree with the Department of Signal Theory and Communications, URJC. Her research interests include body sensor networks with medical applications, physiological signal processing, dynamic routing, distributed signal processing, and wireless sensor networks for energy efficiency.

Eduardo Morgado received the degree in telecommunication engineering from the Carlos III University of Madrid, Madrid, Spain, in 2004 and the Ph.D. degree from the Rey Juan Carlos University, Fuenlabrada, Spain, in 2009. Currently, he is an Associate Professor with the Department of Signal Theory and Communications, Rey Juan Carlos University. His research interests include signal processing for wireless communications with applications to ad hoc and sensor networks.

Eduardo del Arco received the B.Eng. degree in electric and electronic engineering from Glyndwr University, Wrexham, U.K., in 2008; the M.Eng. degree in telecommunication engineering from the University of Alcala, Madrid, Spain, in 2009; and the M.Sc. degree in telecommunication engineering from Rey Juan Carlos University, Fuenlabrada, Spain, in 2013. He is currently working toward the Ph.D. degree with the Department of Signal Theory and Communications, Rey Juan Carlos University. His research interests include wireless sensor networks, vehicular communications, and submodular optimization.

Julio Ramiro-Bargueño received the bachelor of science degree in physics, the Advanced Studies Diploma, and the Ph.D. degree in physics from Universidad Autonoma de Madrid, Madrid, Spain, in 1991, 1993, and 1997, respectively. Since 2004, he has been an Associate Professor with the Department of Signal Theory and Communications, Rey Juan Carlos University, Fuenlabrada, Spain. His research interests include wireless communications, wireless sensor networks, and new communications for vehicular applications.

Antonio J. Caamaño received the joint B.Sc. and M.Sc. degrees in theoretical physics from Universidad Autonoma de Madrid, Madrid, Spain, in 1995 and the Ph.D. degree in telecommunications engineering from the Carlos III University of Madrid, Madrid, Spain, in 2003. Since 2003, he has been an Associate Professor with the Department of Signal Theory and Communications, Rey Juan Carlos University, Fuenlabrada, Spain. His main research interests include the fields of MANET optimization, bioengineering, and statistical signal processing.