Evaluating the Gap Between Compressive Sensing and Distributed Source Coding in WSN

Michele Rossi, Mohsen Hooshmand, Davide Zordan, Michele Zorzi

***Abstract*—Despite the large body of theoretical research available on compression algorithms for wireless sensor networks (WSNs), only recently have researchers started to design and analyze practical distributed compression techniques. Also, approaches belonging to different fields such as signal processing (e.g., discrete Fourier transforms and compressive sensing) or information theory (e.g., distributed source coding) and networking are seldom evaluated against one another. In the present contribution, we consider practical *lossy* compression schemes that rely on different techniques, such as the exploitation of the temporal and spatial dynamics of the signal as well as recent algorithms based on Compressive Sensing (CS). These techniques are adapted so as to be efficiently applied, within the same data collection framework, to a distributed WSN. Hence, we carry out a comparative performance analysis of these schemes, assessing their performance in terms of reconstruction error *vs* energy requirements. From this, several interesting observations are derived, which allow the identification of the best performing algorithm(s) as a function of the spatio-temporal characteristics of the signal. For CS, we assess the impact of the node selection scheme (scheduling) and gauge its performance gap with respect to an idealized CS scheme where the signal covariance matrix is perfectly known at the reconstruction point. We finally identify areas that need improvement, especially for the enhancement of CS-based compression.**

I. Introduction

Lossy compression is a key functionality for a Wireless Sensor Network (WSN). Its effectiveness is based on the premise that compressed data implies the transmission of smaller packets. This, in turn, alleviates collisions at the channel access and promotes energy savings at the transmitter and the receiver [1].

However, the research on fully distributed, practical and lightweight compression schemes is far from being over. In fact, many aspects remain open, such as a solid understanding of the suitability of the different techniques depending on the type of signal as well as their comparative performance evaluation. The objective of the present paper is to provide a step forward in the understanding of this matter, identifying the most promising approaches as a function of relevant signal statistics and pointing out future research directions.

The effectiveness of distributed compression depends upon implementation related facts, such as the energy required for compression versus that required for transmission. In [2], the authors studied this tradeoff when the sensor nodes *independently compress* their readings, by only exploiting the temporal correlation (TC) in the sampled signal. The best algorithms that were identified in [2] are Discrete Cosine Transform (DCT)-based and Lightweight Temporal Compression (LTC) [3], with the former providing the best compression performance and the latter being the most energy efficient in terms of local computations at the sensors.

Alternatively, one can exploit the spatial correlation in the measured data and, based on that, apply Distributed Source Coding (DSC). Practical DSC methods were first proposed in [4] (termed DISCUS and based on *syndrome encoding*), where the authors suggested the use of convolutional codes as a means to spatially compress the data. In short, this is achieved by splitting the input data sequence $\mathbf{x}$ according to so called *bins* and, for each reading, the bin identifier (`binID`) is sent in place of the actual data point. Compression is achieved as the `binID` needs fewer bits than the original data. If the bins are properly designed, i.e., they have certain distance properties, reconstruction at the receiver is possible through the acquisition of some side information, i.e., an uncompressed (and spatially correlated) data sequence $\mathbf{y}$ that is sent from another sensor. The original idea of [4] consists of computing the bins using a channel code, and in particular using the associated syndrome sequence $\mathbf{s}$. At the receiver, the decoder performs maximum likelihood decoding of the original data sequence $\mathbf{x}$ based on $\mathbf{s}$ and on the side information $\mathbf{y}$. The DISCUS scheme has been subsequently improved, going from the simple convolutional codes of [4] to more powerful Turbo or Low Density Parity Check (LDPC) codes, see [5], which are able to approach theoretical compression ratios.

In addition, recent advancements in the signal processing field have fostered the development of the new theory of Compressive Sensing (CS) [6], [7]. Let N be the number of sensors and $\mathbf{x} \in \mathbb{R}^N$ be the vector representing the input signal (one reading per sensor). Also, assume that $\mathbf{x}$ has a sparse representation in some space, referred to here as $\mathbf{u}$ (i.e., there exists a linear transformation that sparsifies $\mathbf{x}$). The CS theory provides means to solve for $\mathbf{u}$, with high probability, ill-posed linear systems of the type $\mathbf{y} = \mathbf{\Psi}\mathbf{u}$, where $\mathbf{y} \in \mathbb{R}^M$ and $M \ll N$. Once $\mathbf{u}$ is known, the complete signal $\mathbf{x}$ is retrieved applying the inverse of the sparsification transform. This concept can be used to concoct distributed and lightweight signal compression schemes, as proposed in [8]. Hence, CS-based compression is based on the idea that only a small number ($M \ll N$) of sensor nodes send their (uncompressed) readings, namely, the signal $\mathbf{y}$. At the WSN data collector (referred to as the *sink*), the original data $\mathbf{x}$ is retrieved with high probability using the sampled (incomplete) data $\mathbf{y}$ and a suitable transformation basis.

Contribution: in the present paper we consider lossy compression schemes that rely on different techniques, such

All authors are with the Department of Information Engineering (DEI) at the University of Padova, via Gradenigo 6/b, 35131, Padova, Italy. Email: {`rossi,mohsen,zordanda,zorzi`}`@dei.unipd.it`. The work in this paper has been supported in part by the European Commission through the FP7 SWAP project, grant agreement no. 251557.

as the exploitation of the temporal (DCT, LTC) and spatial (DSC) dynamics of the signal as well as recent CS schemes, that utilize the signal correlation in space and time. Notably, despite the large amount of research available on theoretical aspects, only recently (DISCUS has been the first practical approach to DSC) have researchers started to look at practical distributed compression techniques. Most importantly, approaches belonging to different fields such as signal processing (CS, DCT) or information theory (DSC) and networking (LTC) are seldom evaluated against one another. The aim of the present paper is to *shed some light on the comparative performance of these algorithms*.

To this end, the aforementioned schemes are adapted and integrated with practical aggregation and data gathering strategies under realistic WSN settings. Thus, they are compared in terms of their compression and energy consumption performance by varying relevant signal statistics and network parameters. The lessons that we have learned from this study are multifold. First, the best performing scheme depends on the signal statistics. CS-based compression is preferable across the entire range of temporal correlations when the spatial correlation is moderate. As the spatial correlation becomes high, DSC schemes perform best and when the temporal correlation is high, TC-based algorithms (DCT, LTC) perform almost as well as CS, providing a reasonable alternative to the latter. Notably, CS-based schemes show a non-negligible gap between their theoretical performance (a lower bound computed assuming perfect statistical knowledge of the signal at the sink) and the actual one. We believe this gap can be reduced and we identify how that can be done as part of a future research work.

Organization of the paper: in Section II, we start by describing the system model; this includes details on the network topology, the data gathering (routing) structure, a clustering procedure that we use for DSC, the energy consumption and the signal models. In Section III we discuss the compression algorithms that we compare in the paper. In Section IV we present a performance evaluation of the selected compression algorithms and in Section V we summarize our main findings and highlight future research directions.

II. System Model

Signal model: next, we briefly outline the adopted signal model. We consider the method of [9], that has been validated against real signals and allows the generation of time-varying spatial fields with tunable correlation characteristics.

We target applications that sample the sensor field at regular intervals. Thus, time $t = 0, 1, 2, \dots$ is slotted and the slot duration is fixed and equal to Δt. We denote the spatial and temporal domain by $\mathcal{D} = [-x_D, x_D] \times [-y_D, y_D]$ and by $\mathcal{T} = i\Delta t$, $i = 0, 1, 2, \dots$, respectively. A point in space is indicated with $\mathbf{p} = (x, y) \in \mathcal{D}$. With $x(\mathbf{p}, t) : \mathcal{D} \times \mathcal{T} \to \mathbb{R}$ we indicate the generated signal, which is stationary in space and time, with mean $\mu_x = 0$, variance $\sigma_x^2 = 1$ and tunable correlation structure.[1]

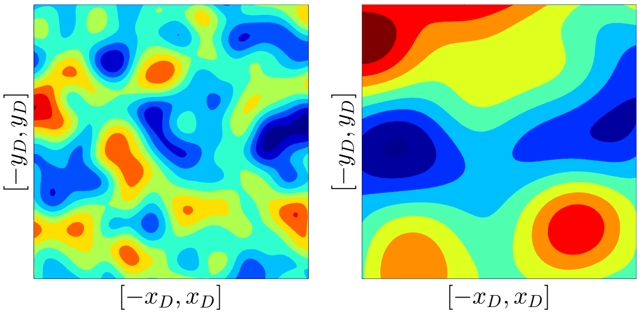

Fig. 1. Signal example for $\gamma = 0.1$ (left) and $\gamma = 1$ (right).

As commonly assumed in the modeling of environmental spatio-temporal signals [9], [10], the correlation function is assumed to be separable in the space and time components:

$$\rho(\mathbf{p}_1, t_1, \mathbf{p}_2, t_2) = \rho_S(\mathbf{p}_1, \mathbf{p}_2)\rho_T(t_1, t_2) \ , \tag{1}$$

where $\mathbf{p}_1$ and $\mathbf{p}_2$ are two points in $\mathcal{D}$, t_1 and $t_2 \geq t_1$ are two time instants, $\rho_S(\cdot)$ and $\rho_T(\cdot)$ respectively denote the spatial and temporal correlation functions. From the signal stationarity, (1) can be equivalently rewritten as a function of $d = \|\mathbf{p}_1 - \mathbf{p}_2\|_2$ and $\Delta t = t_2 - t_1$.

Any suitable correlation function [11] can be used in the signal generation procedure. In this paper, for $\rho_S(d)$ we use a Gaussian function:

$$\rho_S(d) = \exp\left\{-\frac{d^2}{\gamma\alpha^2}\right\} \ , \tag{2}$$

where α is a scaling factor that depends on the field size and γ is a free parameter used to control the spatial correlation. In Fig. 1 we show a single time instant for two synthetic signals generated with this model, obtained with $\gamma \in \{0.1, 1\}$.

For the temporal correlation, we use $\rho_T(\Delta t) = \rho \in [0, 1]$, referred to as the temporal correlation coefficient (this amounts to assuming an exponential correlation function).

Topology and data gathering: we consider WSNs comprising N nodes, deployed uniformly at random in the area $\mathcal{D}$, with $x_D = y_D = 50$ m. For the transmission range R of the nodes we adopt a unit disk model so that sensors can only communicate with those nodes placed at a distance less than or equal to R. We use $R = 2x_D\sqrt{5}/\sqrt{N}$ to guarantee that the structure is connected with high probability under any deployment. The sink node is placed at the center of the WSN area. For the routing, each sensor considers as its next hop the node within its range that provides the largest advancement toward the sink (geographical forwarding), in Fig. 2 we show an example topology for $N = 50$ nodes.

Clustering: for the DSC technique we have used the weighted clustering algorithm (WCA) of [12]. This makes it possible to group the sensors into a predetermined number N_c of clusters obtained so as to evenly divide the WSN area. Within each cluster, a clusterhead (CH) is elected, so that the CH will minimize the average distance with respect

[1] Note that we can obtain any other values for the mean and variance by shifting and scaling the random field without affecting the correlation characteristics.

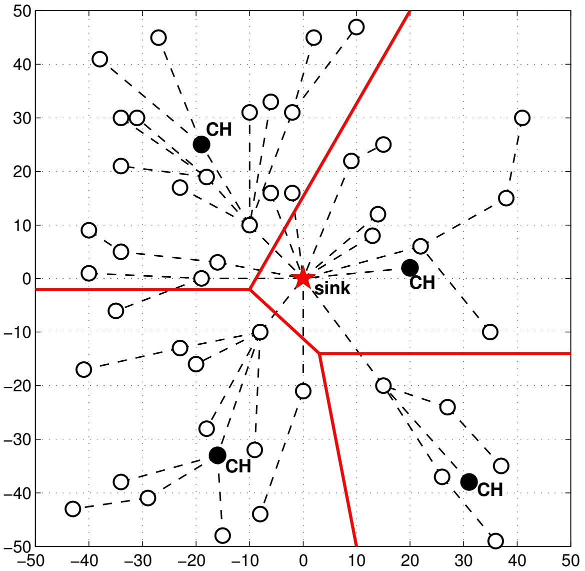

Fig. 2. Example of WSN topology with $N = 50$ sensor nodes. The data gathering tree is indicated by dashed lines, the clusterheads are represented by black-filled circles, different clusters are delimited by solid lines and the node in the center is the sink.

to the non-CH nodes within its cluster. See Fig. 2 for an example with $N_c = 4$.

Energy model: for every compression technique, we have counted the number of additions, multiplications, divisions and comparisons executed at the nodes. Thus, according to the considered sensor hardware, we have translated these figures into the related number of clock cycles and we have subsequently turned the latter into the corresponding energy expenditure. In addition, we have considered the energy consumption associated with the transmission and reception of each packet. In this paper, for these figures we have considered a MSP 430 micro-processor and a CC2420 radio [2] from Texas Instruments.

III. Compression Techniques

A. Temporal Correlation-based Compression

For this class of algorithms we consider LTC and DCT, as these are the best performing algorithms from [2]. With them, each sensor node independently compresses a time series, exploiting the temporal correlation of the signal. Both algorithms work on time series of T subsequent (scalar) readings $\{x_1, x_2, \dots, x_T\}$.

LTC: LTC is a low complexity algorithm that uses a linear model to approximate a time series, according to a preset error tolerance $\varepsilon \geq 0$. The algorithm works by approximating multiple readings through a single segment, so that the segment will be within the given error tolerance for all points, see [2]. This algorithm has a linear complexity in the number of readings in the time series. Also, a segment is described by two coefficients, so compression is achieved when the number of readings that are covered through it (that depends on ε) is larger than 2. It follows that LTC is effective when the signal exhibits a high temporal correlation, whereas in the case of uncorrelated readings its performance is worse than sending the time series uncompressed (as confirmed by the results of Section IV).

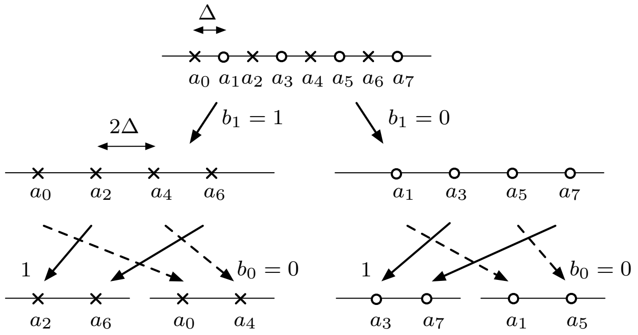

Fig. 3. Ungerboeck tree-based binning procedure [13]. In this example, the alphabet $\mathcal{A}$ contains 8 symbols $\{a_0, \dots, a_7\}$, where these are classified into bins according to (b_1, b_0) ($b_i \in \{0, 1\}$). b_1 splits the 8 symbols into two subsets (or "bins", containing crosses and circles). b_0 allows splitting each subset into two further sets (bins), obtaining the four bins at the bottom of the figure. Note that symbols in the same bin have maximum distance.

DCT: as observed in [2], DCT has a good energy compaction property, which means that the energy of the signal tends to be distributed within the first few DCT coefficients. Based on this, we implemented DCT by taking the time series $\{x_1, x_2, \dots, x_T\}$, computing its DCT and retaining the first T' coefficients, where $T' \leq T$ determines the level of compression and, at the same time, the representation accuracy. In our implementation, also for DCT we considered a maximum error tolerance ε (see also [2]). This implies that the compressor has to evaluate the representation accuracy of the compressed sequence, adding additional coefficients until the tolerance ε is met.

Compression strategy: the compression is orchestrated according to cycles of T time slots. Each sensor independently collects a time series $\{x_1, x_2, \dots\}$ and, when T readings are available, these are transformed according to the compression strategy in use (either DCT or LTC). The resulting coefficients are stored in the payload of a single(multiple) packet(s) that is(are)[2] sent to the sink and the process is repeated.

B. Spatial Correlation-based Compression

As per the DSC scheme, as suggested in [4] we use the Ungerboeck tree-based binning scheme [13], an example of which is provided in Fig. 3. We consider the transmission of an analog reading, which is first quantized (uniform quantization is assumed in Fig. 3, with quantization step Δ) into a certain number of levels N_ℓ ($N_\ell = 8$ in the figure). We refer to n as the number of bits required to represent the quantized symbol. After this, bins are obtained by grouping

[2] Multiple packets may be needed to carry the compressed signal, depending on T, the compression process and the payload size.

the quantized symbol as shown in Fig. 3. In this figure, we use four bins $(B_0, \ldots, B_3)$, which means that $k = 2$ bits (b_1, b_0) are sufficient to represent the binID. Note that following this approach the symbols within each bin have maximum distance, whilst assuring an even distribution of the symbols in the bins. Also, the distance property is maintained for all bins. This procedure can be applied for any n and k.

Compression strategy: given the above splitting procedure, the DSC scheme works as follows. As said above, the WSN is divided into a number N_c of clusters. At each time slot t, each non-CH node in the cluster compresses its reading $x_t \in \mathbb{R}$ by first quantizing it into x_t^q according to Δ and N_ℓ.[3] Subsequently, the quantized symbol x_t^q is mapped into the corresponding bin according to the Ungerboeck procedure, using $k < n$ bits. At time t, the resulting binID, indicated by s_t (k bits), is stored at the sensor node in place of the quantized symbol x_t^q. The CH simply quantizes its current reading y_t and stores in its internal memory its quantized representation y_t^q (n bits). Every T time slots the memory of both non-CH and CH nodes is flushed and the data therein is sent to the sink. In detail, CH nodes store the last T quantized readings in a single packet (payload of nT bits), which is sent to the sink via multi-hop routing. Non-CH nodes store the last T compressed readings (the payload is now kT bits) into a single packet and send it to the sink via multi-hop routing.

For the procedure to be executed at the sink, let us consider a generic cluster. The decompressor first looks at the packet received from the corresponding CH, which contains T uncompressed readings y_t^q (side information). Consider now a non-CH node i in this cluster. For each binID s_t contained in the packet from this node, with $t \in \{1, \ldots, T\}$, we refer to B_{s_t} as the corresponding bin and the estimated (quantized) symbol for node i is obtained as:

$$\hat{x}_t^q = \operatorname{argmin}_{x^q \in B_{s_t}} |x^q - y_t^q| \,, \tag{3}$$

where we look for the element in the bin B_{s_t} that has minimum Euclidean distance with respect to the side information y_t that is received from the CH.

From the bin construction procedure, it descends that $\hat{x}_t^q$ will be equal to the original reading x_t^q if the difference between x_t^q and the side information y_t^q is strictly smaller than $2^{k-2}\Delta$. Note that this depends on the (spatial) correlation properties of the signal. Also, we consider the quantization error as negligible; note that this is a good assumption in the considered scenario.[4] Also, it is possible to improve the performance of the above procedure through the use of coding for the generation of binID sequences (LDPC being a popular example). In this paper, we consider the memoryless procedure outlined above, noting that it is often preferred for WSNs due to its lightweight character, see, e.g., [14]. Extension to more complex coding approaches, along with the evaluation of their additional complexity, are left as future work.

C. CS-based Compression

As stated in the introduction, CS is a mathematical framework that makes it possible to solve ill-posed linear systems when the unknown vector $\mathbf{x}$ is sparse. In our problem, $\mathbf{x} = [x(1), x(2), \ldots, x(N)]^T \in \mathbb{R}^N$, where the i-th element $x(i)$ represents the reading generated by sensor node i. Including the temporal dimension, we indicate the complete WSN signal at time t by $\mathbf{x}_t$ (with the term “complete,” we mean containing the readings from all the sensor nodes).

At any time t, CS can be used to approximate $\mathbf{x}_t$ by measuring a subset of the readings, i.e., collecting a smaller vector $\mathbf{y}_t$ of size $M \ll N$ [8].

Here, we assume that the signal $\mathbf{x}_t$ is sparse and zero mean,[5] which means that there exists an invertible $N \times N$ transformation matrix $\mathbf{\Phi}_t$ such that

$$\mathbf{x}_t = \mathbf{\Phi}_t \mathbf{u}_t \,, \tag{4}$$

where $\mathbf{u}_t \in \mathbb{R}^N$ is M-sparse.[6] Hence, assuming that $\mathbf{\Phi}_t$ is known, $\mathbf{x}_t$ can be retrieved from its sparse representation using (4).

Compression strategy: at each time slot t the sink collects the readings from $M \ll N$ sensors (see Section III-E for the sensor selection strategies), storing them into vector $\mathbf{y}_t$.[7] Thus, we can write $\mathbf{y}_t = \mathbf{R}_t \mathbf{x}_t$, where $\mathbf{R}_t$ is a sampling matrix of size $M \times N$. In detail, $\mathbf{R}_t$ is an all zero matrix, except for a single one in each row and at most a single one in each column (i.e., $\mathbf{y}_t$ is a sampled version of $\mathbf{x}_t$). Using the previous definitions, we have:

$$\mathbf{y}_t = \mathbf{R}_t \mathbf{x}_t = \mathbf{R}_t \mathbf{\Phi}_t \mathbf{u}_t \overset{\text{def}}{=} \mathbf{\Psi}_t \mathbf{u}_t \,. \tag{5}$$

(5) can be solved for $\mathbf{u}_t$, once the sparsification matrix $\mathbf{\Phi}_t$ (see Section III-D) and the sampling matrix $\mathbf{R}_t$ are known. Once $\mathbf{u}_t$ is known, (4) is used to retrieve the complete signal $\mathbf{x}_t$. Note that the signal that we obtain through this procedure, referred to as $\hat{\mathbf{x}}_t$, is an interpolated version of $\mathbf{x}_t$.

D. Sparsification Basis $\mathbf{\Phi}_t$

As demonstrated in [15], although standard transforms such as the Fast Fourier Transform (FFT) or the DCT perform satisfactorily as sparsification bases for video sequences, they are rather ineffective for typical WSN signals. A solution to this entails the use of Principal Component Analysis (PCA) [16]. PCA is a statistical processing technique that uses the sample covariance matrix $\mathbf{\Sigma}$ to convert a correlated signal into a number of principal components that is usually smaller than the number of variables in the original

[3] Quantization is performed according to a minimum distance criterion, as in [4].

[4] We consider $n = 16$ bits per reading, with a signal range of ≈ 2 units. This leads to $\Delta = 3 \cdot 10^{-5}$, which is negligible compared to the reconstruction error, see Section IV.

[5] This assumption is not restrictive as the approach explained here can be applied to the signal after subtracting its (estimated) mean $\overline{\mathbf{x}}_t$ [8].

[6] A vector is defined M-sparse when it only has M significant elements, whereas the amplitude of the remaining $N - M$ elements is negligible with respect to the average energy per component, $(1/N)\sqrt{< \mathbf{u}_t, \mathbf{u}_t >}$.

[7] Note that M can be varied across subsequent time slots.

signal $\mathbf{x}$. That is, PCA finds a smaller space where $\mathbf{x}$ can be projected by minimizing the loss of information that occurs from the original (size N) to the projected (size M) space. The projection basis is obtained by the M eigenvectors of the sample covariance matrix that better represent the signal (i.e., that capture most of its energy). Note that the projection basis corresponds to the sparsification basis $\mathbf{\Phi}_t$ that we are looking for.

In this paper, matrix $\mathbf{\Phi}_t$ is obtained as follows:

1) Use $\mathbf{y}_t$ and the current $\mathbf{\Psi}_t$ matrix to obtain $\mathbf{y}_t \to \hat{\mathbf{x}}_t$ (through (5), solved via ℓ_1-minimization).
2) Store $\hat{\mathbf{x}}_t$ into a buffer, containing the T most recent (estimated) vectors $\mathcal{X} = \{\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_{t-1}, \dots, \hat{\mathbf{x}}_{t-T+1}\}$.
3) Estimate $\mathbf{\Sigma}_t$ (and $\overline{\mathbf{x}}_t$) from the vectors in $\mathcal{X}$.
4) Compute the new $\mathbf{\Psi}_{t+1}$ from $\mathbf{\Sigma}_t$ using PCA.

More computationally efficient algorithms can be found in [17]. These are not considered for the results in this paper.

E. Sampling Strategy $\mathbf{R}_t$

In this section, we briefly discuss the sampling strategies that we have considered for matrix $\mathbf{R}_t$.

Random node selection (RNS) [8]: at each time slot $t = 0, 1, 2, \dots$, on average, a number M of nodes transmit their readings to the sink. A simple and distributed technique to achieve this is to define a transmission probability p_{tx} so that $M = Np_{\text{tx}}$. Hence, at time t, each node independently decides whether it will be transmitting to the sink according to p_{tx}; in that case, its current reading is stored in the payload of a single packet, which is sent to the sink through multi-hop routing. If user $i \in \{1, \dots, N\}$ transmits at time t, $\mathbf{R}_t$ will have a (single) one in the i-th column.

Deterministic node selection (DNS): this approach works in data collection cycles of T time slots each, as we now explain. A predetermined and deterministic sampling strategy is computed by the sink and disseminated to all WSN nodes. Exactly M nodes sample the signal at each time slot and we define $T = \kappa\lceil N/M \rceil$, with κ integer. For the discussion that follows, consider the first collection cycle, i.e., time slots $\{1, 2, \dots, T\}$. The subset of sampling nodes is computed according to the following criteria: 1) the M selected nodes shall evenly cover the area $\mathcal{D}$, 2) all nodes sample the signal at least once within $\lceil N/M \rceil$ time slots. Actually, they sample exactly once if N/M is integer and up to two times otherwise. In the latter case, $\lfloor N/M \rfloor M$ nodes sample the signal once in the first $\lfloor N/M \rfloor$ slots, whereas in the last time slot, $(N \mod M)$ nodes sample the signal for the first time and $M - (N \mod M)$ nodes resample it to assure that there are M sampling nodes.

This is achieved through a suitable *monitoring schedule*, which is obtained using the theory in [18]. At time t, $\mathbf{R}_t$ will have a one in column $i \in \{1, \dots, N\}$ if the monitoring schedule includes node i for transmission in this time slot. To improve the efficiency in the transmission of the sampled readings we use the following *data aggregation* approach. From the above discussion, each node will be sampling the signal $n_s(T) \in \{\kappa, \kappa+1, \dots, 2\kappa\}$ times (depending on whether N/M is integer) in the T time slots (in the slots indicated by the *monitoring schedule*). In the last slot T, the sensor will aggregate the $n_s(T)$ samples storing them in the payload of a single (or multiple) packet(s), which is(are) sent to the sink for processing. The sink collects the incoming packets and, given its knowledge of the sampling matrix R_t, associates back the readings from the nodes to the corresponding time slots. This permits to obtain y_t at the sink for all time slots $\{1, 2, \dots, T\}$. This process is repeated for the following cycles. As we show in Section IV, this aggregation strategy allows a substantial improvement with respect to RNS.

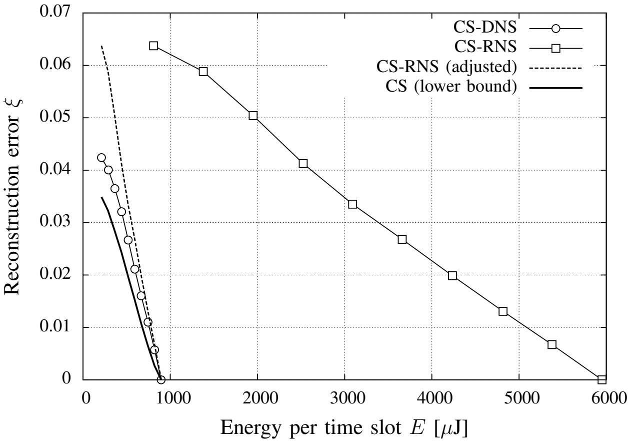

Fig. 4. Error ξ vs energy consumption E for $\rho = 0$ and $\gamma = 10^{-3}$ (uncorrelated signal in both time and space).

IV. Results

In this section, we compare TC-based algorithms (LTC, DCT) against SC-based (DSC) and CS-based techniques. We consider a network of $N = 50$ nodes with the signal model of Section II with varying spatial (γ) and temporal (ρ) correlation parameters for a collection cycle length of $T = 50$ time slots.[8] For the performance metrics, we compute the average reconstruction error at the sink $\xi \triangleq \mathrm{E}_t\left[\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_1\right]$, where $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$ respectively represent the original signal and the one reconstructed at the sink, and the total average energy consumption per time slot, E, expressed in Joule (including the transmission and processing tasks performed by the WSN nodes). Each sensor reading takes $n = 16$ bits of memory and for DSC we have used $k = 4$ bits to represent the bin identifiers. In the following, we show tradeoff curves comparing ξ (y-axis) against E (x-axis), where the performance of DSC and CS schemes is obtained by varying an independent parameter as follows: DSC) the number of clusters N_c is varied between 5 and N in steps of 5; CS-RNS) p_{tx} goes from 0.1 to 1 in steps of 0.1; CS-DNS) M goes from 5 to N in steps of 5. A free parameter is not needed for DCT and LTC as our implementation of these two schemes is self-tuning, i.e., it takes the error ξ

[8] Although not presented here due to space constraints, results similar to those that we discuss in this section have been obtained for higher values of T.

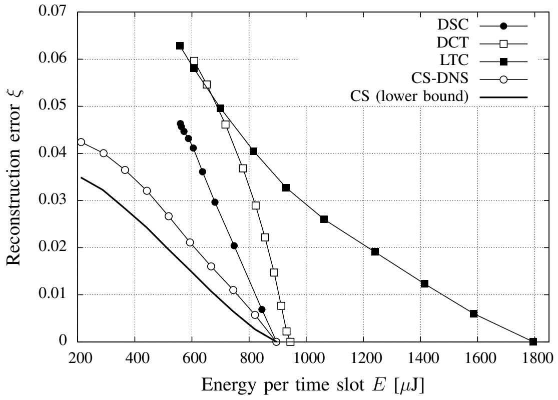

Fig. 5. Error ξ vs energy consumption E for $\rho = 0$ and $\gamma = 10^{-3}$ (uncorrelated signal in both time and space).

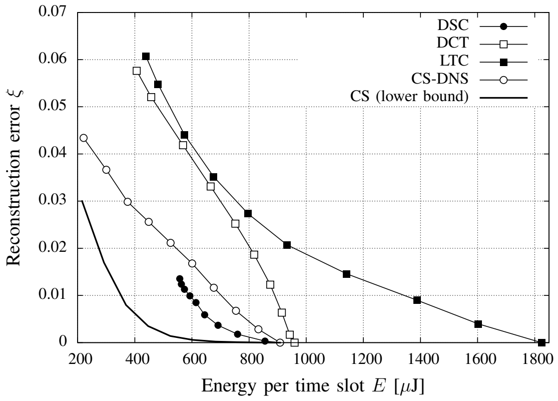

Fig. 6. Error ξ vs energy consumption E for $\rho = 0.5$ and $\gamma = 5$ (highly correlated signal in both time and space).

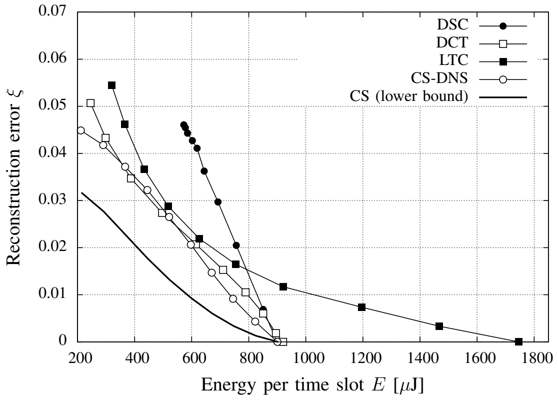

Fig. 7. Error ξ vs energy consumption E for $\rho = 0.8$ with $\gamma = 10^{-3}$ (temporally correlated but spatially uncorrelated signal).

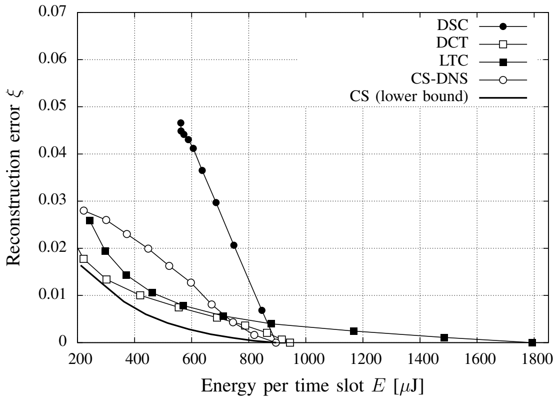

Fig. 8. Error ξ vs energy consumption E for $\rho = 0.98$ and $\gamma = 10^{-3}$ (temporally correlated but spatially uncorrelated signal).

as the input parameter. All results were obtained averaging over a sufficiently large number of simulations so that the confidence intervals (not shown in the graphs for readability) are within 5% of the average values plotted.

We start our discussion with Fig. 4, where we show the performance of CS for $\rho = 0$ and $\gamma = 10^{-3}$, i.e., an uncorrelated signal in space and time. As stated in Section III-C, practical CS algorithms compute the sparsification basis from the PCA, which is obtained from incomplete observations of the signal. Here, we obtain a lower bound on the error recovery performance of CS-DNS by calculating the sample covariance matrix from the *complete* signal $\mathbf{x}_t$, assuming that all the past samples (excluding the current one at time t) are available at the sink with no errors and at no additional cost with respect to the acquisition of the incomplete signal set. This idealized algorithm is referred to in the following plots as "lower bound" and has the same energy consumption as CS-DNS. From Fig. 4, we observe a few interesting facts. First, CS-DNS substantially outperforms CS-RNS (originally proposed in [8]) and the reason for this resides in 1) the aggregation strategy adopted by CS-DNS and 2) the type of sampling (i.e., deterministic versus probabilistic). To understand the impact of each of these, we considered an additional curve, obtained by adjusting the energy consumption performance of CS-RNS so that its equals that of CS-DNS. A direct comparison between this (adjusted) curve and CS-DNS reveals the improvement, in terms of reconstruction error ξ, that is brought about by DNS. In this case, the aggregation process does not affect ξ. As expected, this improvement is especially high when p_{tx} is small, i.e., when the signal is randomly sampled by a small number of sensors. The reason behind this is that a properly designed deterministic schedule provides a more uniform sampling (in both time and space) of the signal that, in turn, translates into a better estimation of its statistical properties at the sink (i.e., of its covariance matrix). Further, substantial energy savings are possible through data aggregation, as this alleviates the negative impact due to the transmission of large packet headers (assumed to be 13 bytes for the results in this paper). CS-RNS is consistently outperformed by CS-DNS for all values of ρ and ξ. For this reason, it will no longer be considered in the rest of the paper.

Next, we focus on the comparative analysis of CS, DSC, DCT and LTC, which is shown in Figs. 5–8. From these figures, a few key observations can be made. **a)** When the signal is uncorrelated in space and time (Fig. 5), CS-DNS is the scheme of choice and performs close to its theoretical bound. Temporal correlation schemes should be avoided, as their energy consumption in this case is higher than that incurred in sending all the data uncompressed (given by the abscissa value where CS and DSC reach $\xi = 0$). Especially, LTC is rather inefficient as two coefficients (i.e., one segment) are sent for each data point for small values of ρ. **b)** As the signal correlation increases (see Fig. 6, with $\rho = 0.5$ and $\xi = 5$), DSC better uses the increased spatial correlation, outperforming CS as can be observed for $\xi \geq 1$. The performance of DCT and LTC is only marginally improved. **c)** For a temporal correlation as high as $\rho = 0.8$ (see Fig. 7), the performance of DCT and LTC reaches that of CS. **d)** For signals that are highly correlated in time but spatially uncorrelated, the situation is reversed, see Fig. 8. In this case, DCT and LTC both outperform CS, whereas DSC performs worst and its use is not recommended.

As a general remark, the gap between CS-DNS and its lower bound increases for increasing signal correlation. This means that the quality of the sample covariance matrix is highly impacted in this case. This, in turn, affects the accuracy of the related PCA transform, providing less accurate approximations of the signal. These results indicate that there is still some room for improvement for CS, whose performance can be ameliorated so as to approach that of DSC and temporal-compression schemes. To achieve this goal, we need to concoct improved covariance estimators for incomplete signal sequences. To this end, a sensible approach could be the use of spatial correlation estimators to filter the noise affecting the sample covariance. Future work has to be carried out in this direction.

All in all, however, if the correlation statistics are unknown, CS is deemed a valid compression approach as it often outperforms competing algorithms and, in the worst cases, it performs in between temporal and spatial correlation-based compression.

V. Lesson Learned and Future Work

In this paper, we have performed a comparative performance evaluation of selected lossy compression schemes for WSN. This analysis can be seen as a preliminary study for a more in-depth research work, as it serves to pinpoint the pitfalls of the considered schemes. In fact, we found that no single algorithm performs best in all settings, and compressive sensing in general provides good performance that, in the worst case, lies in between that of temporal and spatial compression schemes.

Most importantly, our analysis reveals that there are interesting avenues for future research. One of these consists in the improvement of CS schemes, and specifically in the design of improved estimators for the sparsification basis (obtained through PCA), which is strictly related to the covariance structure of the signal. Also, CS schemes can be implemented as well by exploiting a clustered WSN, i.e., by restricting the scope of the matrix inversion within each of the clusters. Finally, more powerful DSC algorithms can be considered in our framework, accounting for LDPC codes. These, on the one hand, are expected to better approach the theoretical bounds in terms of reconstruction error but, on the other hand, also imply a higher computational burden at the sensor nodes. The computational requirements of these more complex DSC approaches have to be evaluated.

References

[1] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, “Practical data compression in wireless sensor networks: A survey,” *Elsevier Journal of Network and Computer Applications*, vol. 35, no. 1, Jan. 2012.

[2] D. Zordan, B. Martinez, I. Villajosana, and M. Rossi, “On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking,” *ACM Transactions on Sensor Networks*, vol. 11, no. 1, Aug. 2014.

[3] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin, “Lightweight temporal compression of microclimate datasets,” in *IEEE International Conference on Local Computer Networks (LCN)*, Tampa, FL, US, Nov. 2004.

[4] S. S. Pradhan and K. Ramchandran, “Distributed Source Coding Using Syndromes (DISCUS): Design and Construction,” *IEEE Transactions on Information Theory*, vol. 49, no. 3, Mar. 2003.

[5] Z. Xiong, A. D. Liveris, and S. Cheng, “Distributed Source Coding for Sensor Networks,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, Sep. 2004.

[6] E. Candès and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Trans. on Information Theory*, vol. 52, no. 12, Dec. 2006.

[7] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhauser (Series: Applied and Numerical Harmonic Analysis), 2013.

[8] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, “Sensing, Compression and Recovery for WSNs: Sparse Signal Modeling and Monitoring Framework,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, Oct. 2012.

[9] D. Zordan, G. Quer, M. Zorzi, and M. Rossi, “Modeling and Generation of Space-Time Correlated Signals for Sensor Network Fields,” in *IEEE GLOBECOM*, Houston, Texas, US, Dec. 2011.

[10] M. Umer, L. Kulik, and E. Tanin, “Spatial interpolation in wireless sensor networks: localized algorithms for variogram modeling and Kriging,” *Geoinformatica*, vol. 14, no. 1, Jan. 2010.

[11] P. Abrahamsen, “A review of Gaussian random fields and correlation functions,” Norwegian Computing Center, Oslo, Tech. Rep., 1997.

[12] M. Chatterjee, S. K. Das, and D. Turgut, “An On-Demand Weighted Clustering Algorithm (WCA) for Ad Hoc Networks,” in *IEEE GLOBECOM*, San Francisco, CA, US, Nov. 2000.

[13] G. Ungerboeck, “Channel Coding with Multilevel/Phase Signals,” *IEEE Transactions on Information Theory*, vol. IT-28, no. 1, Jan. 1982.

[14] J. E. Barceló-Lladó, A. M. Pérez, and G. Seco-Granados, “Enhanced Correlation Estimators for Distributed Source Coding in Large Wireless Sensor Networks,” *IEEE Sensors*, vol. 2, no. 9, Sep. 2012.

[15] G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, “On the interplay between routing and signal representation for compressive sensing in wireless sensor networks,” in *IEEE Workshop on Information Theory and Applications (ITA)*, San Diego, CA, US, Feb. 2009.

[16] C. R. Rao, “The Use and Interpretation of Principal Component Analysis in Applied Research,” *Sankhya: The Indian Journal of Statistics*, vol. 26, no. 4, Dec. 1964.

[17] P. M. Hall, D. Marshall, and R. R. Martin, “Incremental Eigenanalysis for Classification,” in *British Machine Vision Confernece*, Southampton, UK, Sep. 1998.

[18] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, “Power Efficient Monitoring Management in Sensor Networks,” in *IEEE WCNC*, Atlanta, GA, US, Mar. 2004.