DASS: Distributed Adaptive Sparse Sensing

Zichong Chen, Juri Ranieri, Runwei Zhang, and Martin Vetterli, *Fellow, IEEE*

***Abstract*—Wireless sensor networks are often designed to perform two tasks: sensing a physical field and transmitting the data to end-users. A crucial design aspect of a WSN is the minimization of the overall energy consumption. Previous researchers aim at optimizing the energy spent for the communication, while mostly ignoring the energy cost of sensing. Recently, it has been shown that considering the sensing energy cost can be beneficial for further improving the overall energy efficiency. More precisely, sparse sensing techniques were proposed to reduce the amount of collected samples and recover the missing data using data statistics. While the majority of these techniques use fixed or random sampling patterns, we propose adaptively learning the signal model from the measurements and using the model to schedule when and where to sample the physical field. The proposed method requires minimal on-board computation, no inter-node communications, and achieves appealing reconstruction performance. With experiments on real-world datasets, we demonstrate significant improvements over both traditional sensing schemes and the state-of-the-art sparse sensing schemes, particularly when the measured data is characterized by a strong intra-sensor (temporal) or inter-sensors (spatial) correlation.**

***Index Terms*—Wireless sensor networks, sparse sensing, adaptive sampling scheduling, compressive sensing, energy efficiency.**

I. Introduction

In a wireless sensor network (WSN), sensor nodes are deployed to take periodical measurements of a certain physical field at different locations. Consider a continuous-time spatio-temporal field that we would like to monitor with the WSN; denote it as $x(\boldsymbol{p},t)$, where $\boldsymbol{p}$ and t represent the spatial position and the time, respectively. We also define a vector $\boldsymbol{x} \in \mathbb{R}^N$ containing a discretization of such a field with a sufficiently high resolution for our purposes. The target of the WSN is to recover $\boldsymbol{x}$ with the maximum precision.

One of the primary goals in designing a WSN is its reduction of the energy consumption, to extend its lifetime without replacing or recharging the batteries of sensor nodes. The energy consumption of a sensor node mainly comes from three activities: sensing, data-processing and communication. Traditionally, the costs for processing and communication are assumed to dominate the overall energy consumption, while the cost for sensing is considered negligible. Therefore, a traditional WSN collects as much data as possible, that is subsequently compressed and transmitted with the lowest possible rate. In other words, it collects a vector of samples $\boldsymbol{y}_0$ that is equal to the discretized physical field $\boldsymbol{x}$ with some additive noise,

$$\boldsymbol{y}_0 = \boldsymbol{I}\boldsymbol{x} + \boldsymbol{\omega}, \tag{1}$$

where $\boldsymbol{I}$ is the identity matrix of size N and $\boldsymbol{\omega}$ represents the noise; see Fig. 1(a) for an example.

Ignoring the energy cost for sensing is sub-optimal, if sensing consumes a comparable amount of energy to communication and data processing. In fact, new sampling paradigms optimizing the overall energy consumption have been proposed and show that further reductions of the energy consumption are possible. The basic idea involves a reduction of the number of collected samples and a reconstruction of the missing data using algorithms exploiting the structure available in the measured data. The reduction of the collected samples is done by designing a sampling operator $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ with $M \ll N$, that it is used instead of the identity matrix as,

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{\omega}. \tag{2}$$

Note that $\boldsymbol{y}$ is significantly shorter than $\boldsymbol{x}$ and the reconstruction algorithm must estimate a significant amount of information from a limited amount of data. Therefore, regularization and constraints are added to the problem so that a stable solution can be obtained. Moreover, the reconstruction algorithm must be jointly designed with the sampling matrix $\boldsymbol{\Phi}$ to obtain a precise estimate of $\boldsymbol{x}$.

Pioneering work on sparse sampling considered compressive sensing (CS) as a reconstruction scheme. CS attempts to recover $\boldsymbol{x}$ by solving a convex optimization problem, under the assumption that $\boldsymbol{x}$ is sparse in a known dictionary $\boldsymbol{\Pi}$. However, the solution is only approximate and it is exact if $\boldsymbol{\Pi}$ and $\boldsymbol{\Phi}$ satisfy certain requirements that are generally hard to check [4]. Initially, [9], [16], [23] proposed the use of a sampling matrix $\boldsymbol{\Phi}$ composed of random i.i.d. Gaussian entries. Note from Fig. 1(b) that such $\boldsymbol{\Phi}$ has very few zero elements. Therefore, the number of sensing operations is not actually reduced because we need to know all the values of $\boldsymbol{x}$ to compute $\boldsymbol{y}$. Moreover, if we adopt a distributed algorithm, a dense $\boldsymbol{\Phi}$ requires the sensor nodes to transmit their local samples to the other nodes, causing an excessive energy consumption for communications.

To overcome such limitations, [17], [26] proposed to use a sparse matrix $\boldsymbol{\Phi}$ which contains very few non-zero elements. More precisely, $\boldsymbol{\Phi}$ has generally only one non-zero element per row and the locations of such elements determine the spatio-temporal sampling pattern, see Fig. 1(c). However, the sampling patterns in these schemes are either fixed or randomly generated and thus not well adapted to the measured signal.

Manuscript received November 6, 2013; revised April 8, 2014 and September 5, 2014; accepted December 18, 2014. Date of publication January 7, 2015; date of current version May 7, 2015. The results of this research are reproducible: The datasets and Matlab codes used to generate figures can be found in our reproducible repository at http://rr.epfl.ch/. This research was supported by Swiss National Centre of Competence in Research and ERC Advanced Investigators Grant of European Union. The associate editor coordinating the review of this paper and approving it for publication was Y. Chen.

The authors are with the LCAV, I&C, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne 1015, Switzerland (e-mail: chenzc04@gmail.com; juri.ranieri@epfl.ch; runwei.zhang@epfl.ch; martin.vetterli@epfl.ch).

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.2014.2388232

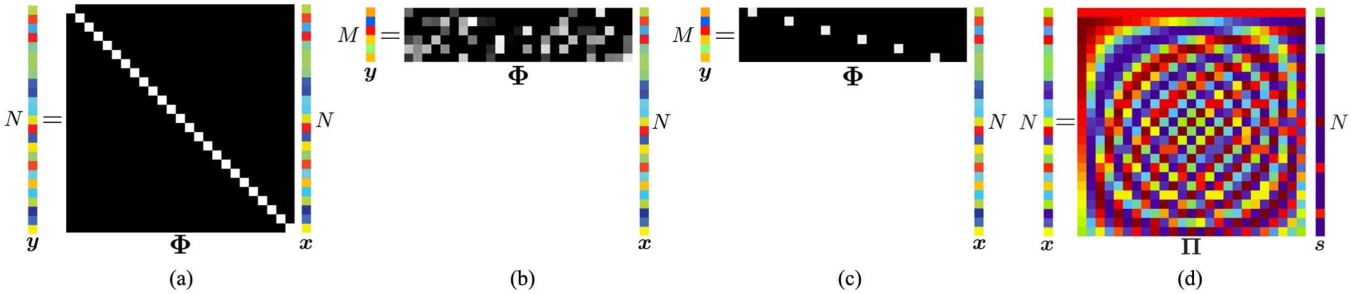


Fig. 1. Comparison of various sensing schemes proposed in the literature (the noise term $\boldsymbol{\omega}$ is omitted for simplicity). We consider a discretized version of the sampled physical field that is contained into a vector $\boldsymbol{x}$. In (a) we depict the traditional approach where we measure the physical field in each spatio-temporal location, thus having an identity operator $\boldsymbol{I}$. In (b), we reduce the number of samples by taking random projections of the measurements. Note that we need to measure all the elements of $\boldsymbol{x}$ and we are just reducing the number of stored samples. On the other hand, in (c) we are reducing the number of measured samples using a sparse sampling matrix $\boldsymbol{\Phi}$. Note that the methods in (b) and (c) require a set of conditions regarding $\boldsymbol{x}$ and $\boldsymbol{\Phi}$ to be satisfied [5]. Among these conditions, we note that $\boldsymbol{x}$ must be sparse under a certain known dictionary $\boldsymbol{\Pi}$, see (d). (a) Traditional sensing; (b) CS—Dense matrix; (c) CS—Sparse matrix; (d) sparsity dictionary.

Moreover, it is generally hard to guarantee the recovery of a faithful representation of $\boldsymbol{x}$, because the sparsity of dictionary $\boldsymbol{\Pi}$ usually changes over time and it may not satisfy the theoretical requirements of CS [5].

Since the statistics of $\boldsymbol{x}$ are often unknown and varying over time, it may be advantageous to consider the decomposition

$$\boldsymbol{x} = \boldsymbol{\Psi}^t \boldsymbol{\alpha}, \tag{3}$$

where $\boldsymbol{\Psi}^t$ is the time-varying model and $\boldsymbol{\alpha} \in \mathbb{R}^K$ is a low dimensional representation of $\boldsymbol{x}$ with $K \ll N$. While the ignorance and the non-stationarity of the model $\boldsymbol{\Psi}^t$ forces us to learn it from the samples collected in the past, it may give us the advantage of optimizing the sampling pattern $\boldsymbol{\Phi}^t$ according to $\boldsymbol{\Psi}^t$. The non-stationarity of $\boldsymbol{\Phi}^t$ is a feature diversifying our approach from the CS algorithms, where the sensing patterns are usually fixed as shown in Fig. 1.

This new problem statement raises new challenges. While the model $\boldsymbol{\Psi}^t$ can be learnt from the incomplete measurements $\boldsymbol{y}$ using an online version of the principal component analysis (PCA), selecting the sampling pattern $\boldsymbol{\Phi}^t$ for minimizing the construction error is a combinatorial problem. In this paper, we propose a generalized version of FrameSense, an algorithm that generates a near-optimal sensor placement for inverse problems [19]. More precisely, instead of optimizing the sensor placement, we optimize the spatio-temporal sampling pattern of the WSN. The obtained sampling pattern is generally irregular, time-varying and optimized to gather the maximum amount of information. In particular, it simultaneously exploits the intra-node (temporal) and inter-node (spatial) correlation potentially present in the data. See Fig. 2 for a graphical illustration of the low-dimensional model and of the irregular sampling patterns.

Our method derives from and extends the sparse sensing framework proposed by Quer *et al.* [17]: the signal is first approximated by a linear model $\boldsymbol{\Psi}^t$, and the sampling scheduling is defined in space and time by a sampling matrix $\boldsymbol{\Phi}^t$. Our major contribution is that we improve the way in which the spatio-temporal correlation is exploited, such that the sampling pattern is dynamically adapted to the low dimensional model of the signal.

It is worth mentioning that the proposed method imposes no on-sensor computation nor inter-node communication. Each sensor node simply collects measurements according to a designated sampling pattern and transmits the data to a common server. The server receives all the data from one or multiple sensor nodes and performs signal reconstruction. This is actually in accordance to the setup of distributed source coding [22], where no inter-node communication is used. Hence, the proposed algorithm provides an alternative solution to the distributed coding problem: the communication rate is reduced and the reconstruction error is bounded without using any inter-node communication.

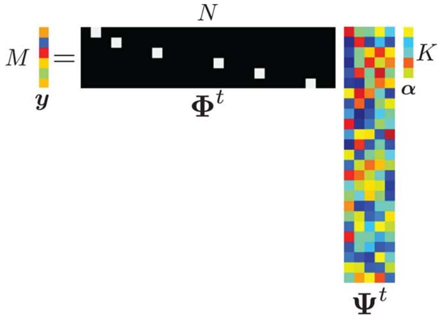


Fig. 2. Graphical representation of the mathematical model of the proposed sensing scheme. The signal is modeled by an unknown time-varying linear K-dimensional model $\boldsymbol{\Psi}^t$ that is learn from the collected measurements. The sampling pattern $\boldsymbol{\Phi}^t$ is optimized at run-time according to the signal model and measures only M values out of the N available ones.

The proposed algorithm is tested on different sets of real-word data, outperforming both the traditional sensing schemes and the state-of-the-art sparse sensing schemes, in terms of reconstruction quality of $\boldsymbol{x}$ given a fixed amount of measurements. Given the aforementioned characteristics, we call the proposed method "*Distributed Adaptive Sparse Sensing*," or *DASS*.

II. Problem Formulation

In this section, we first state the sampling scheduling problem for a WSN having just one sensor. At the end of the section, we generalize the problem statement to a WSN with multiple nodes. We consider a block-based sensing strategy, meaning that the WSN samples the field for a certain time T and at the end we reconstruct the vector $\boldsymbol{x}$ from the collected samples. Note that the block length is known and defined *a priori*. Table I is the summary of notations that are used in this paper.

TABLE I
SUMMARY OF NOTATION

N	desired number of samples in a block	M	number of measurements in a block, equals $\lfloor N\gamma \rfloor$
Δ_T	temporal resolution of original signal	f	sampling frequency of original signal, equals $1/\Delta_T$
f_s	average sampling frequency of the sensor	γ	subsampling rate f_s/f
$\widetilde{\boldsymbol{x}}$	reconstructed signal $\in \mathbb{R}^N$	$\boldsymbol{x}$	original signal $\in \mathbb{R}^N$
$\boldsymbol{y}$	measured signal $\in \mathbb{R}^M$	$\boldsymbol{\omega}$	measurement noise
$\boldsymbol{\tau}^t$	sampling pattern of the t-th block	$\boldsymbol{\Phi}^t$	sampling matrix of the t-th block $\in \mathbb{R}^{M\times N}$
$\overline{\boldsymbol{x}}$	mean of the signal $\in \mathbb{R}^N$	$\boldsymbol{\Psi}^t$	signal model of the t-th block $\in \mathbb{R}^{N\times K}$
$\boldsymbol{\alpha}$	low dimensional representation of $\boldsymbol{x} \in \mathbb{R}^K$	$\widetilde{\boldsymbol{\Psi}}^t$	rows of $\boldsymbol{\Psi}^t$ selected by $\boldsymbol{\tau}^t$ $\in \mathbb{R}^{M\times K}$

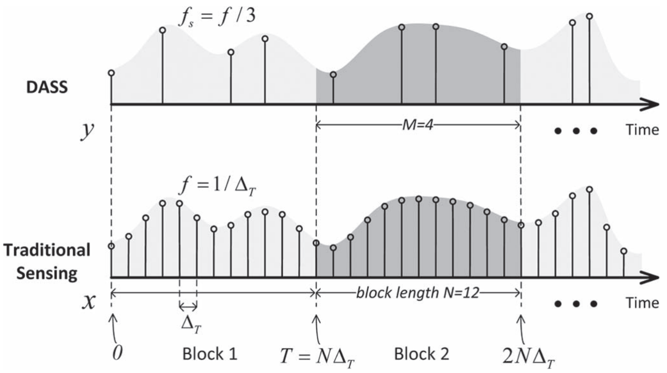


Fig. 3. Upper plot: optimized temporal sampling pattern of DASS. Lower plot: traditional sensing scheme, where samples are collected regularly in time. The subsampling factor is $\gamma = 1/3$, since we collect 4 samples instead of 12 in each block.

For each temporal block, the discrete physical field $\boldsymbol{x}$ is composed of N samples of $x(\boldsymbol{p},t)$,

$$\boldsymbol{x} = [x(\boldsymbol{p},0), x(\boldsymbol{p},\Delta_T), \ldots, x(\boldsymbol{p},(N-1)\Delta_T)]^\top, \tag{4}$$

where $\boldsymbol{p}$ indicates the sensor node location and Δ_T is the sampling period. Note that Δ_T determines the desired temporal resolution and its inverse is the sampling frequency, $f = 1/\Delta_T$. The temporal duration of a block is $T = N\Delta_T$, that is also the maximum delay this sensing scheme occurs—the larger T, the longer the delay. See Fig. 3 for a graphical representation of the physical field and its discrete version $\boldsymbol{x}$.

We denote the reconstructed physical field obtained from the WSN samples as $\widetilde{\boldsymbol{x}}$. In a sparse sampling scenario, we aim at reconstructing $\widetilde{\boldsymbol{x}}$ from just a subset of elements of $\boldsymbol{x}$. More precisely, we measure M elements out of N, where $M < N$. The set of indices $\boldsymbol{\tau}^t = \{\tau_i^t\}_{i=1}^M$ denotes the indices of these M samples and it is chosen adaptively according to the previous measurements. The sampling pattern $\boldsymbol{\tau}^t$ uniquely determines the sampling matrix $\boldsymbol{\Phi}^t \in \mathbb{R}^{M\times N}$ that previously appears in (2):

$$\boldsymbol{\Phi}_{i,j}^t = \begin{cases} 1 & \text{if } j = \tau_i^t \\ 0 & \text{otherwise.} \end{cases}$$

The sensing matrix $\boldsymbol{\Phi}^t$ has exactly one non-zero element per row, and usually a maximum of one non-zero element per column. Here either $\boldsymbol{\Phi}_{i,j}^t$ or τ_i^t can be interpreted as a temporal selector deciding when the node should take a sample—the index j indicates the time index within a block. It is important to underline that $\boldsymbol{\Phi}^t$ and $\boldsymbol{\tau}^t$ are time-varying and potentially changing at every block to adapt to the signal model $\boldsymbol{\Psi}^t$. Fig. 3 shows an example of sampling patterns where $\boldsymbol{\tau}^t$ changes for each block.

We define $f_s = \frac{M}{N} \cdot f = \gamma f$ to be the average sampling frequency of the sensor node.[1] The subsampling rate $\gamma = f_s/f < 1$ is an important figure of merit for a sparse sampling algorithm—the lower the γ, the lower the energy consumed for sensing.

The measured signal $\boldsymbol{y} \in \mathbb{R}^M$ is defined as

$$\boldsymbol{y} = \boldsymbol{\Phi}^t \boldsymbol{x} + \boldsymbol{\omega}, \tag{5}$$

where $\boldsymbol{\omega}$ represents the measurement noise, that is modeled as an additive white Gaussian noise (AWGN), since the thermal effects [14] or/and quantization [25] are often the dominating terms.[2] Throughout the paper, we mainly discuss the simpler case of i.i.d. noises; however, we will shortly discuss the extension to the generic case of noise having a correlation matrix $\boldsymbol{\Sigma}_{\boldsymbol{\omega}}$. We define the signal-to-noise ratio (SNR) of the measurement as following, which will be used in the evaluations:

$$\text{SNR(dB)} = 10\log_{10}\left(\frac{\|\boldsymbol{x}\|_2^2}{\|\boldsymbol{\omega}\|_2^2}\right). \tag{6}$$

The target of DASS is to optimize the sampling pattern $\boldsymbol{\Phi}^t$ at the t-th block according to $\boldsymbol{\Psi}^t$ such that we collect the minimum number of samples M while still being able to recover precisely the original signal. Since we modeled the noise as a AWGN, we assess the quality of the recovered signal by using root-mean-square error (RMSE):

$$\epsilon = \frac{1}{\sqrt{N}}\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|_2.$$

Multiple-Node Scenario: While the above problem statement focuses on a single-sensor scenario for simplicity of notation, it is simple to generalize the statement to a WSN with more than one sensor node. More precisely, we assume that the nodes are synchronized,[3] so that we can concatenate all the measured blocks at different locations $\boldsymbol{p}_i$ in a unique signal block $\boldsymbol{x}$. Fig. 4 shows an example. $\boldsymbol{x}_A, \boldsymbol{x}_B, \boldsymbol{x}_C$ are signal blocks from three different locations, $\boldsymbol{\Phi}_A^t, \boldsymbol{\Phi}_B^t, \boldsymbol{\Phi}_C^t$ are the respective sampling matrices for each location, and $\boldsymbol{y}_A, \boldsymbol{y}_B, \boldsymbol{y}_C$ are the respective measurements. We can write

$$\begin{bmatrix} \boldsymbol{y}_A \\ \boldsymbol{y}_B \\ \boldsymbol{y}_C \end{bmatrix} = \boldsymbol{\Phi}^t \begin{bmatrix} \boldsymbol{x}_A \\ \boldsymbol{x}_B \\ \boldsymbol{x}_C \end{bmatrix} + \boldsymbol{\omega}, \text{ where } \boldsymbol{\Phi}^t = \begin{bmatrix} \boldsymbol{\Phi}_A^t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_B^t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Phi}_C^t \end{bmatrix}. \tag{7}$$

Here different sensors can take different number of samples and $\boldsymbol{\Phi}_A^t, \boldsymbol{\Phi}_B^t, \boldsymbol{\Phi}_C^t$ can have different sizes. Thus, $\boldsymbol{\Phi}^t$ can be interpreted as a general spatio-temporal selector to choose *when*

[1] Note that we denote f_s as an average sampling frequency given the irregular and time-varying sampling pattern.

[2] Other noise model may be of interest for specific sensors; for example the noise term of a Geiger counter is usually modeled as a Poisson process.

[3] Note that the proposed method does not require a precise synchronization. In fact, eventual variations of the model due to the lack of synchronization are handled by the proposed method thanks to the adaptive learning of the model.

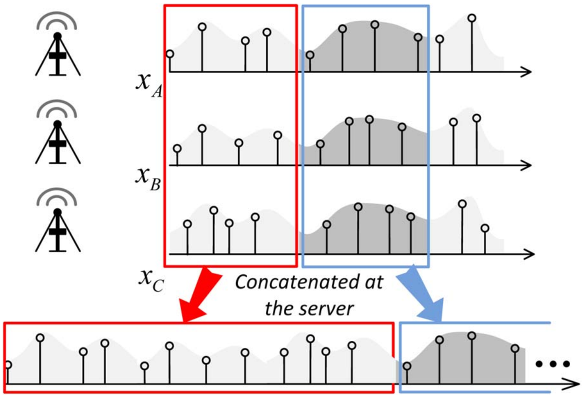


Fig. 4. Signals of multiple distributed sensor nodes can be concatenated into a single signal stream at the server for recovery.

and where to sample such that we collect the maximum amount of information. Moreover, it is worth mentioning that $\mathbf{\Phi}^t$ is optimized for each block to adapt to the time-varying model of the physical field.

III. Building Blocks

The proposed method is graphically represented in Fig. 5 and is based on the three building blocks described in this section:

1) The desired signal $\tilde{\boldsymbol{x}}$ is reconstructed using the collected measurements $\boldsymbol{y}$, the signal model $\mathbf{\Psi}^t$ and the estimated mean $\bar{\boldsymbol{x}}$ (Section III-A).
2) We use the measurements $\boldsymbol{y}$ to update the approximation model $\mathbf{\Psi}^t$, $\bar{\boldsymbol{x}}$ (Section III-B).
3) The sampling pattern for the next temporal block $\boldsymbol{\tau}^{t+1}$ is optimized according to $\mathbf{\Psi}^t$ and is transmitted back to the sensor node(s) (Section III-C).

The overhead of DASS on the sensor node is minimal in practice. First, the sampling pattern $\boldsymbol{\tau}^t$ has a sparse structure and hence it can be encoded efficiently with a few bytes per block. Therefore, the extra communication cost for receiving $\boldsymbol{\tau}^t$ is minimal. Second, all the algorithmic complexity of DASS is at the server side, while the sensor nodes only need to sample and transmit the signal according to the sampling pattern received from the server. Therefore, the CPU and memory requirements of the sensor node are minimal.

In what follows, we analyze each block explaining the challenges and the proposed solution.

A. Signal Approximation and Reconstruction

Due to the nature of most physical fields, a signal block is partially predictable by analyzing past data. In many cases, this predictability can be expressed by assuming that the signal belongs to a K-dimensional linear subspace $\mathbf{\Psi}^t \in \mathbb{R}^{N\times K}$. Such a subspace approximates $\boldsymbol{x}$ as

$$\widehat{\boldsymbol{x}} = \mathbf{\Psi}^t\boldsymbol{\alpha} + \bar{\boldsymbol{x}}, \tag{8}$$

where $\widehat{\boldsymbol{x}}$ is the approximated field, $\boldsymbol{\alpha} \in \mathbb{R}^K$ is the vector of the projection coefficients and $\bar{\boldsymbol{x}}$ is the mean of $\boldsymbol{x}$.

If the modeling subspace $\mathbf{\Psi}^t$ is well designed and K is sufficiently large compared to the complexity of $\boldsymbol{x}$, the signal realization $\boldsymbol{x}$ can be accurately expressed with just $K \ll N$ coefficients contained in $\boldsymbol{\alpha}$. To find such a subspace, we analyze all the past signal realizations and estimate at the t-th block the K-dimensional subspace $\mathbf{\Psi}^t$ that minimizes the expected approximation error

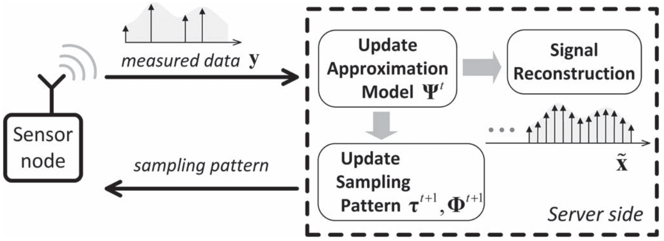

measured data $\mathbf{y}$
Update Approximation Model $\mathbf{\Psi}^t$
Signal Reconstruction
Sensor node
sampling pattern
Update Sampling Pattern $\boldsymbol{\tau}^{t+1}, \mathbf{\Phi}^{t+1}$
$\tilde{\mathbf{x}}$
Server side

Fig. 5. Representation of the operations of DASS in a WSN. The sensor node sends the measured data to the processing server and receives the sampling pattern for the next temporal block. The server uses the data to update the signal model $\mathbf{\Psi}^t$, reconstructs the discrete physical field $\tilde{\boldsymbol{x}}$ and optimizes the sampling pattern $\boldsymbol{\tau}^{t+1}$ for the sensor nodes. Note that $\boldsymbol{\tau}^{t+1}$ uniquely determines $\mathbf{\Phi}^{t+1}$.

$$\epsilon_a = \frac{1}{\sqrt{N}}\mathbb{E}\left(\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_2\right). \tag{9}$$

This is a dimensionality reduction problem that can be solved by the well known technique of *principal component analysis (PCA)*.[4] It has an analytic solution but it requires the covariance matrix $\boldsymbol{C_x}$.

Unfortunately, in our scenario it is hard to estimate $\boldsymbol{C_x}$ since we have access only to M out of N elements of $\boldsymbol{x}$. However, if the M sampled elements are varying at each temporal block t, we may collect enough information to have a sufficiently precise estimate of $\boldsymbol{C_x}$. We present a set of methods to estimate $\boldsymbol{C_x}$ in Section III-B.

Note that the approximation through $\mathbf{\Psi}^t$ exploits the spatial and temporal correlation among the elements of $\boldsymbol{x}$. The higher the correlation available in $\boldsymbol{x}$, the lower the dimensionality of the subspace $\mathbf{\Psi}^t$, the number of parameters K and the necessary measurements M. Hence, one of the key aspects is the choice of the signal block length T. In fact, it should be chosen such that the delay of the WSN respects the design specification while maximizing the correlation among the blocks. For example, a sensor measuring the outdoor light intensity naturally shows diurnal patterns. Therefore, if we choose a block length of one hour, the correlation within the signal block is usually weak. On the other hand, if we choose a block length of one day, the correlation is stronger.

Once the approximation model $\mathbf{\Psi}^t$ is estimated, recovering the signal $\tilde{\boldsymbol{x}}$ amounts to estimating $\boldsymbol{\alpha}$ from the measurements $\boldsymbol{y}$ when considering the approximated signal model,

$$\boldsymbol{y} \approx \mathbf{\Phi}^t\widehat{\boldsymbol{x}} + \boldsymbol{\omega} = \mathbf{\Phi}^t\left(\mathbf{\Psi}^t\boldsymbol{\alpha} + \bar{\boldsymbol{x}}\right) + \boldsymbol{\omega}. \tag{10}$$

If $\boldsymbol{\omega}$ is an i.i.d. Gaussian random noise, we can recover $\boldsymbol{\alpha}$ by solving an Ordinary Least Square (OLS) problem [17]:

$$\tilde{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - \mathbf{\Phi}^t\bar{\boldsymbol{x}} - \mathbf{\Phi}^t\mathbf{\Psi}^t\boldsymbol{\alpha}\|_2^2, \tag{11}$$

[4] The CS-based sparse sensing methods in [17] also used PCA for subspace learning.

which has the following analytic solution,

$$\widetilde{\boldsymbol{\alpha}} = (\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)^{\dagger}(\boldsymbol{y} - \boldsymbol{\Phi}^t\bar{\boldsymbol{x}}). \tag{12}$$

Here $(\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)^{\dagger}$ is the Moore-Penrose pseudoinverse of $\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t$ that is defined for a generic matrix $\boldsymbol{A}$ as $\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*$. We can slightly vary the reconstruction technique to account for more complicated noise models. For example, if the noise is distributed according a known covariance matrix $\Sigma_{\boldsymbol{\omega}}$, we can estimate $\boldsymbol{\alpha}$ as

$$\widetilde{\boldsymbol{\alpha}} = \left((\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)^*\Sigma_{\boldsymbol{\omega}}^{-1}\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t\right)^{-1}(\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)^*\Sigma_{\boldsymbol{\omega}}^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}^t\bar{\boldsymbol{x}}). \tag{13}$$

We can generalize such an estimator to other additive noise models, see [15]. For the remainder of the paper we keep considering the i.i.d. Gaussian noise case. Nonetheless, we show in Appendix A that the proposed framework is compatible to correlated noises by whitening the measured data before the processing.

Once we define the estimator for $\boldsymbol{\alpha}$, the reconstruction algorithm is straightforward and is described in Algorithm 1. The following proposition states the necessary conditions to find a unique solution and provides an upper bound for the reconstruction error, that is going to be fundamental when optimizing the sampling pattern. Such a result is an adaption of two classic results of linear algebra [11].

Proposition 1: Consider a sensor network measuring a physical field as in (10) where the M measurements are corrupted by an i.i.d. Gaussian noise with variance σ^2. If $M \geq K$, $\boldsymbol{\Psi}^t \in \mathbb{R}^{N\times K}$ is formed by orthonormal columns and $\operatorname{rank}(\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t) = K$, then $\widetilde{\boldsymbol{x}}$ can be uniquely determined using Algorithm 1. The reconstruction error is bounded by

$$\epsilon^2 = \frac{1}{N}\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|_2^2 \leq \frac{1}{\lambda_K}\epsilon_a^2 + \sigma^2\sum_{i=1}^{K}\frac{1}{\lambda_i}, \tag{14}$$

where ϵ_a is the approximation error due to the signal model $\boldsymbol{\Psi}^t$ and λ_i $(1 \leq i \leq K)$ is the i-th largest eigenvalue of the matrix $\boldsymbol{\Psi}^{t*}\boldsymbol{\Phi}^{t*}\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t$.

Algorithm 1 Signal reconstruction

```
Require: $\boldsymbol{\Psi}^t$, $\bar{\boldsymbol{x}}$, $\boldsymbol{\tau}^t$ and $\boldsymbol{\Phi}^t$
Ensure: $\widetilde{\boldsymbol{x}}$
  1: Measure the signal $\boldsymbol{y}$ according to $\boldsymbol{\tau}^t$.
  2: $\widetilde{\boldsymbol{x}} = \boldsymbol{\Psi}^t(\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)^{\dagger}(\boldsymbol{y} - \boldsymbol{\Phi}^t\bar{\boldsymbol{x}}) + \bar{\boldsymbol{x}}$.
```

Proof: Since the Gaussian noise is independent from the approximation error, we can treat them independently. Moreover, it is sufficient to compute the error on the estimation of $\boldsymbol{\alpha}$ given the assumed orthonormality of the columns of $\boldsymbol{\Psi}^t$.

For reconstruction error due to the approximation error ϵ_a, we look at the worst case scenario with the following optimization problem,

$$\begin{aligned} \max \quad & \left\|(\boldsymbol{\Psi}^t\boldsymbol{\Psi}^t)^{\dagger}(\boldsymbol{x} - \widehat{\boldsymbol{x}})\right\|_2^2 \\ \text{subject to} \quad & \frac{1}{N}\|(\boldsymbol{x} - \widehat{\boldsymbol{x}})\|_2^2 = \epsilon_a, \end{aligned}$$

whose solution is proportional to the smallest eigenvalue of $(\boldsymbol{\Psi}^t\boldsymbol{\Psi}^t)^{\dagger}$. More precisely, it is possible to show that the approximation noise is upper bounded by $\frac{1}{\lambda_K}\epsilon_a^2$, where ϵ_a is the norm of the approximation error.

For the reconstruction error due to the white noise, we use a known result of frame theory, see [10]. We merge the two results to conclude the proof. ■

The upper-bound of the total error ϵ is a function of both the approximation error ϵ_a and the measurement noise $\boldsymbol{\omega}$. The former term depends on the number of parameters K: when $K = N$, we have $\epsilon_a = 0$ and it grows when we decrease K. However, the rate at which the error increases depends on the spectrum of $\boldsymbol{C_x}$. In fact, if $\boldsymbol{x}$ has elements that are highly correlated, a small K could be sufficient to model $\boldsymbol{x}$ with a small approximation error. The latter term can be controlled directly by optimizing the sampling pattern. More precisely, we cannot reduce σ but we can reduce the amplification due to the spectrum λ_k through an optimization of the sampling matrix $\boldsymbol{\Phi}^t$.

Note that the part involving ϵ_a only depends on the smallest eigenvalue because we are not guaranteed that the approximation error *spreads* over all the eigenvectors of $\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t$. In fact, the worst case scenario is represented by the approximation error being in the same direction of the eigenvector with the smallest eigenvalue and ϵ_a is consequently maximally amplified.

Compared to the methods based on CS, our approach based on a low-dimensional model and OLS has the following advantages: i) the solution is easy to compute and it requires a single matrix inversion, ii) it enables an analysis of the reconstruction error and a consequent optimization of the sampling pattern $\boldsymbol{\tau}^t$ such that the upper-bound of ϵ is minimized.

B. Learning From Incomplete Data Over Time

In Section III-A, we have highlighted some challenges regarding the estimation of the covariance matrix $\boldsymbol{C_x}$—a fundamental step to determine the approximation model $\boldsymbol{\Psi}^t$. Most of the challenges derive from the lack of a sufficiently large set of realizations of $\boldsymbol{x}$, that are needed to estimate $\boldsymbol{C_x}$. First, there is virtually no past data for a newly installed WSN. Second, $\boldsymbol{C_x}$ is likely to vary over time. Third, a high ratio of data points $(1-\gamma)$ are not available for the estimation since we collect sparse measurements. Therefore, we need an on-line algorithm that estimates and adaptively updates the covariance matrix $\boldsymbol{C_x}$ from incomplete data.

The main difficulty is the lack of complete realizations of $\boldsymbol{x}$. Two strategies are generally considered to overcome such a problem. The first one proposes to estimate from $\boldsymbol{y}$ an interpolation $\boldsymbol{x}_{\text{interp}}$ using classic methods such as linear, polynomial or spline interpolation. The second strategy skips the estimation of $\boldsymbol{C_x}$ and attempts to perform directly the principal component analysis on the data having missing entries, see [18].

From our quantitative results, the second class of algorithms is less powerful for our purposes. Therefore, we focus our attention on the interpolation methods. More precisely, we analyze two different methods that implement an adaptive learning and updating of the approximation model $\boldsymbol{\Psi}^t$ from the interpolated signal $\boldsymbol{x}_{\text{intep}}$.

The first, whose pseudocode is given in Algorithm 2, uses a FIFO buffer to store the most recent L blocks. Whenever a new block is added into the buffer, the oldest block in the buffer is excluded. As the approximation model is estimated according to the signal realizations in the buffer, this scheme is able to capture the variation of signal statistics over time.

Algorithm 2 Updating $\boldsymbol{\Psi}^t$, $\bar{\boldsymbol{x}}$ using a buffer

```
Require: $\boldsymbol{y}$, $L$
Ensure: $\boldsymbol{\Psi}^t$, $\bar{\boldsymbol{x}}$
  interpolate $\boldsymbol{y} \rightarrow \boldsymbol{x}_{\text{intep}}$.
  insert $\boldsymbol{x}_{\text{intep}}$ into a buffer storing the most recent $L$ blocks.
  estimate $\mathbf{C}_{\boldsymbol{x}}$ and $\bar{\boldsymbol{x}}$ from the buffer.
  $\boldsymbol{\Psi}^t$ is formed by the eigenvectors corresponding to the $K$
    largest eigenvalues of the matrix $\mathbf{C}_{\boldsymbol{x}}$.
```

The second, see Algorithm 3, adaptively updates the approximation model via a technique called incremental PCA [12]. It does not keep signal realizations in memory, instead, it stores the largest K eigenvalues of $\boldsymbol{C}_{\boldsymbol{x}}$, $\boldsymbol{\lambda} = \{\lambda_i\}$, for $i = 1, \ldots, K$. This method requires significantly less memory (K versus $N \times L$), and shows better performance when compared to Algorithm 2. Note that in both algorithms, the choice of L depends on the variability of the signal statistics for each specific application. In practice, we can cross-validate this parameter to find a suitable value (e.g., $L = 30$). We discuss and compare the performance of these two algorithms in the experimental results.

Algorithm 3 Updating $\boldsymbol{\Psi}^t$, $\bar{\boldsymbol{x}}$ using incremental PCA

```
Require: $\boldsymbol{y}$, $L$, $\boldsymbol{\Psi}^{t-1}$, $\boldsymbol{\lambda}^{t-1}$, $\bar{\boldsymbol{x}}^{t-1}$
Ensure: $\boldsymbol{\Psi}^t$, $\boldsymbol{\lambda}^t$, $\bar{\boldsymbol{x}}^t$
  interpolate $\boldsymbol{y} \rightarrow \boldsymbol{x}_{\text{intep}}$.
  $\mathbf{a} = \boldsymbol{\Psi}^{t-1*}(\boldsymbol{x}_{\text{intep}} - \bar{\boldsymbol{x}}^{t-1})$.
  $\mathbf{b} = (\boldsymbol{\Psi}^{t-1}\mathbf{a} + \bar{\boldsymbol{x}}^{t-1})) - \boldsymbol{x}_{\text{intep}}$, and then normalize $\mathbf{b}$.
  $c = \mathbf{b}^*(\boldsymbol{x}_{\text{intep}} - \bar{\boldsymbol{x}}^{t-1})$.
  $\mathbf{D} = \frac{1}{L+1}\begin{bmatrix} \text{diag}(\boldsymbol{\lambda}^{t-1}) & \mathbf{0} \\ \mathbf{0}^* & 0 \end{bmatrix} + \frac{L}{(L+1)^2}\begin{bmatrix} \mathbf{aa}^* & c\mathbf{a} \\ c\mathbf{a}^* & c^2 \end{bmatrix}$.
  Solve the eigenproblem: $\mathbf{D} = \mathbf{R} \cdot \text{diag}(\boldsymbol{\lambda}') \cdot \mathbf{R}^{-1}$, $\boldsymbol{\lambda}'$ is
    sorted in decreasing order.
  $\boldsymbol{\Psi}' = [\boldsymbol{\Psi}^{t-1}\ \mathbf{b}] \cdot \mathbf{R}$.
  update $\boldsymbol{\Psi}^t$ as the first $K$ columns of $\boldsymbol{\Psi}'$.
  update $\boldsymbol{\lambda}^t$ as the first $K$ values of $\boldsymbol{\lambda}'$.
  update $\bar{\boldsymbol{x}}^t$ as $(L\bar{\boldsymbol{x}}^{t-1} + \boldsymbol{x}_{\text{intep}})/(L+1)$.
```

C. Sampling Scheduling Algorithm

According to Proposition 1, minimizing the overall error ϵ is equivalent to finding the optimal sampling pattern $\boldsymbol{\tau}$ that minimizes (14). We fix the values of K and M in the optimization process, and hence the approximation error ϵ_a is fixed. In this paper, we assume that the model $\boldsymbol{\Psi}^t$ is sufficiently precise and the dimensions K is large enough so that the term due to the white noise σ is dominant. Note that if the approximation error decays exponentially fast with K, there exists always a small K such that $\epsilon_a \ll \sigma$. We will show in the experimental part that meteorological data exhibits such an exponential decay of the approximation error.

To optimize the scheduling pattern, we would like to find the sampling pattern that minimizes the following cost function,

$$\Theta(\widetilde{\boldsymbol{\Psi}}^t) = \sum_{k=1}^{K} \frac{1}{\lambda_k}, \tag{15}$$

where λ_k are the eigenvalues of $(\widetilde{\boldsymbol{\Psi}}^t)^*\widetilde{\boldsymbol{\Psi}}^t$, and $\widetilde{\boldsymbol{\Psi}}^t = \boldsymbol{\Phi}^t\boldsymbol{\Psi}^t$. Note that this optimization is equivalent to finding the M rows of $\boldsymbol{\Psi}^t$ that forms the submatrix $\widetilde{\boldsymbol{\Psi}}^t$ with the smallest $\Theta(\widetilde{\boldsymbol{\Psi}}^t)$. However, it has been already shown that such optimization is NP-hard [7], [8] and has a complexity $O(\binom{N}{M})$, which is prohibitively high in practice.

Therefore, we investigate approximate solutions to the scheduling problem that can be implemented efficiently. These approximate solutions are usually hard to find because the cost function $\Theta(\widetilde{\boldsymbol{\Psi}}^t)$ has many local minima that are arbitrarily far away from the global minimum. Therefore, proxies of $\Theta(\widetilde{\boldsymbol{\Psi}})$ are usually chosen as a cost function for the approximated algorithm with a twofold aim: (i) inducing an indirect minimization of $\Theta(\widetilde{\boldsymbol{\Psi}}^t)$ and (ii) being efficiently optimized by standard techniques, as convex optimization or greedy algorithms.

In this paper, we extend our recent work [19] about near-optimal sensor placement for linear inverse problems to solve the sampling scheduling problem. In [19], we considered a generic linear inverse problem defined as,

$$\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\alpha}, \tag{16}$$

where $\boldsymbol{x}$ contains all the possible sensors locations, $\boldsymbol{\Psi}$ is a known linear model and $\boldsymbol{\alpha}$ a set of parameters that we would like to estimate. Then, we assumed that we can measure only a subset of elements of $\boldsymbol{x}$, hence the sensor-placement problem, and proposed an algorithm based on the greedy minimization of a proxy: the frame potential [6]. Such a proxy is defined as

$$\text{FP}(\boldsymbol{\Psi}^t, \mathcal{S}) = \sum_{i,j \in \mathcal{S}} \left| \langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle \right|^2, \tag{17}$$

where $\boldsymbol{\psi}_i$ is the i-th row of $\boldsymbol{\Psi}^t$ and $\mathcal{S}$ contains the set of candidate locations for sensing. Under some mild solutions, we proved that such an algorithm is near-optimal w.r.t. the RMSE of the solution. While this proof is quite technical and available in [19], there is an intuitive explanation of why the frame potential is a good proxy. Shortly speaking, the frame potential favors the rows of $\boldsymbol{\Psi}^t$ that are closer to be orthogonal to each other. Therefore, the algorithm selects the sensor locations containing a large amount of information regarding the measured physical field.

In this work, we note that the sensor placement problem and the sampling scheduling problem are extremely similar: we have a linear inverse problem and we would like to estimate a set of parameters $\boldsymbol{x}$ using the least number of measurements $\boldsymbol{y}$

without compromising the RMSE of the estimation. Nonetheless, there are a set of differences characterizing the latter:

- the model $\boldsymbol{\Psi}^t$ that we use to model the data measured by the sensor network is generally not known *a priori* and is time-variant,
- the sensor placement is optimized according to the spatial correlation, while here we consider the joint spatio-temporal correlation,
- the sensor placement is determined at design-time, while the sampling schedule is time-varying and optimized at run-time.

The sampling scheduling algorithm proposed in this paper is based on an equivalent greedy "worst-out" procedure: as input we have the signal model $\boldsymbol{\Psi}^t$ and we initially consider the identity matrix of size N as the sampling matrix $\boldsymbol{\Phi}^{t+1}$. At each iteration, we remove the row of $\boldsymbol{\Phi}^{t+1}$ that maximizes the cost function (17). After $N-M+1$ iterations we are left with an optimized $\boldsymbol{\Phi}^{t+1}$ that has only M elements different from zero and has near-optimal performance when reconstructing $\boldsymbol{x}$ from the measurements $\boldsymbol{y}$. Note that if $\boldsymbol{\Psi}^t$ satisfies the conditions given in [19], the obtained sampling matrix $\boldsymbol{\Phi}^{t+1}$ recovers $\boldsymbol{x}$ from the measurements $\boldsymbol{y}$ with a near-optimal RMSE.

In most of the scenarios, the sampling schedule optimized according to the proposed greedy minimization of the frame potential has state-of-the-art performance. However, there exists scenarios where a uniform sampling schedule could be better [26], such as when the temporal measurements are characterized by a low-pass spectrum. Therefore, at the end of the greedy optimization we compare the RMSE obtained by the greedy and the uniform schedule and opt for the one with smallest reconstruction error. Note that the reconstruction error cannot be computed exactly given the uncertainty on the approximation error and we use the expression provided by Proposition 1, that bounds the RMSE for any given sampling matrix $\boldsymbol{\Phi}^t$.

A detailed description of the overall algorithm is given in Algorithm 4. Note that for the very first block of data during system startup, the uniform sampling schedule is used for initialization.

Algorithm 4 Greedy sampling scheduling

```
Require: $\boldsymbol{\Psi}^t$, $M$
Ensure: $\boldsymbol{\tau}^{t+1}$ for the next temporal block
Initialize the set of removed sampling indices: $\mathcal{L} = \emptyset$.
Initialize the set of selected sampling indices: $\mathcal{S} = \{1, \dots, N\}$.
Find the first two rows to eliminate, $\mathcal{L} = \arg\max_{i,j \in \mathcal{S}} |\langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle|^2$.
Update $\mathcal{S} = \mathcal{S} \setminus \mathcal{L}$.
repeat
    Find the optimal row, $i^* = \arg\max_{i \in \mathcal{S}} \mathrm{FP}(\boldsymbol{\Psi}^t, \mathcal{S} \setminus i)$.
    Update the set of removed indices, $\mathcal{L} = \mathcal{L} \cup i^*$.
    Update the set of selected indices, $\mathcal{S} = \mathcal{S} \setminus i^*$.
until $|\mathcal{S}| = M$
$\boldsymbol{\tau}^{t+1} = \arg\min_{\boldsymbol{\tau}} \left\{ \frac{\varepsilon_a^2}{\lambda_K} + \sigma^2 \Theta(\widetilde{\boldsymbol{\Psi}}^t), \boldsymbol{\tau} \text{ is uniform pattern or } \mathcal{S} \right\}$.
```

TABLE II
SUMMARY OF METHODS USED IN EXPERIMENTS

Abbreviation	Reconstruction Algorithm	Sampling Scheduling
CS [17, 26]	(18)	uniform
CSN [4, 17]	(19)	uniform
OLS-random [17]	Alg. 1	random
OLS-uniform [17]	Alg. 1	uniform
DASS	Alg. 1	Alg. 4

IV. COMPARISONS WITH BASELINE METHODS

In this section, we briefly summarize the state-of-the-art methods for the sparse sensing problem. They will serve as the baseline for comparisons in Section V.

The first category of methods [17], [26] is based on compressive sensing (CS). With the notations introduced in Section II, $\boldsymbol{x}$ is the unknown signal, $\boldsymbol{y}$ contains the incomplete measurements, and $\boldsymbol{\Phi}$ is a sparse sampling matrix with only M elements different from zero. We assume $\boldsymbol{x}$ to be sparse w.r.t. a dictionary $\boldsymbol{\Pi}$. More precisely, we have $\boldsymbol{x} = \boldsymbol{\Pi}\boldsymbol{s}$ and $\boldsymbol{s}$ has just a few coefficients different from zero, that is $\|\boldsymbol{s}\|_0 \ll N$ (see [3] for more details). By approximating the ℓ_0 norm with the ℓ_1 norm [4], the reconstruction method for the noiseless case is:

$$\min_{\boldsymbol{s} \in \mathbb{R}^N} \|\boldsymbol{s}\|_1, \text{ s.t. } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Pi}\boldsymbol{s}, \tag{18}$$

while the one for the noisy case is

$$\min_{\boldsymbol{s} \in \mathbb{R}^N} \|\boldsymbol{s}\|_1, \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Pi}\boldsymbol{s}\|_2 \leq \xi, \tag{19}$$

where ξ measures the energy of the noise. Problem (18) and (19) are both convex and can be solved [4] in polynomial time using various solvers, in general iterative or based on convex optimization. In both methods, we use uniform sampling as the sampling scheduler—$\tau_j^t = \lfloor jN/M \rfloor$.

The second category of baseline methods [17] are based on learning the K-dimensional time-varying model $\boldsymbol{\Psi}^t$ and a reconstruction via OLS as in Algorithm 1. We use two sampling schedulers, namely, a uniform sampling, and a random sampling where τ_j^t is randomly selected with a uniform distribution.

Table II lists all the methods (including DASS) that are evaluated in the experiments. To have a fair comparison, $\boldsymbol{\Pi}$ in CS-based methods and $\boldsymbol{\Psi}^t$ in OLS-based methods are both learnt[5] by the incremental PCA described in Algorithm 3.

V. EVALUATIONS OF DASS AND SPARSE SENSING METHODS

In this section we evaluate the performance of DASS and compare it with the state-of-the-art sparse sensing methods. Besides the experiments on the single-node case, we also verify DASS in the multi-node case where nearby sensor nodes measure spatially correlated signals. We use two real-world meteorological datasets as the ground truth, namely *Payerne* and *Valais*:

[5] The experimental results show that $K = M$ is the best choice for CS-based methods, while $K < M$ is a parameter which needs to be optimized for OLS-based methods, see Section V-A.

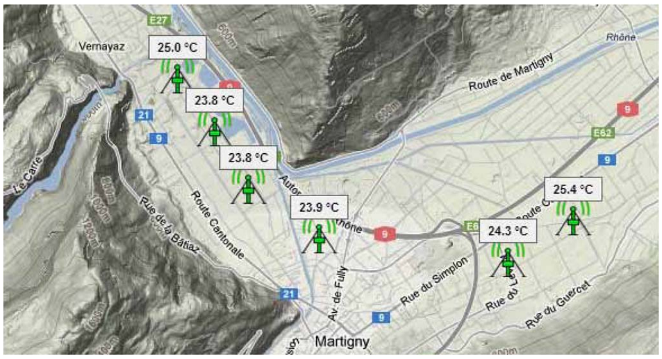


Fig. 6. Locations of the sensor nodes that collected the data-set *Valais*.

TABLE III
SUMMARY OF EXPERIMENTAL DATASETS

Dataset name	Physical quantity	Number of nodes	Number of days
Payerne	temperature, solar radiation	1	1500
Valais	temperature	20	125

- *Payerne* is provided by MeteoSwiss [1]. This dataset contains 1500 days of continuous measurements for two physical quantities (temperature and solar radiation),[6] which are suitable for studying long-term performance of DASS. As MeteoSwiss only deployed a few observation stations across the whole nation, we use *Payerne* for evaluating the single-node case.
- *Valais* is provided by a microclimate monitoring service provider [13]. A total of 20 stations are deployed in a mountain valley. Fig. 6 shows six of them, covering an area of around 18 km^2. The deployments were started in March 2012 and collected 125 days of continuous temperature measurements. We use *Valais* for evaluating the multi-node case.

The two datasets are summarized in Table III. For both datasets, there are 144 uniformly sampled data points for each day. We choose the day as the length of each block, that is, $N = 144$.

One of the targets of this section is to evaluate DASS and compare it with other algorithms when the sensing device induces measurement noise. Since we do not know the groundtruth of the physical field, we assume that *Payerne* and *Valais* represent the real value of the field $\boldsymbol{x}$. Then, we add white Gaussian noise to simulate the effect of noisy measurements. We evaluate the algorithms for different SNR of the measurement (as defined in (6)).

Note that the main merit figure considered in this section is the final reconstruction error under a fixed subsampling rate γ. Since all sparse sensing schemes directly transmit the sensing samples without further data compression, two schemes with the same γ have the same amount of energy consumed for sensing and communication,[7] regardless of which sensing platform is used.

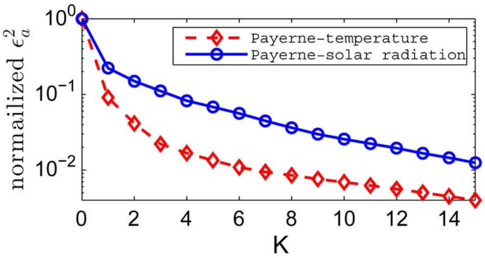


Fig. 7. Normalized approximation error for the two considered datasets as a function of the model parameter K. Note how the error monotonically decreases with K given the optimality of PCA. Moreover, we highlight how the approximation error shows an exponential decay with K; thus confirming our assumption described in Section III-C.

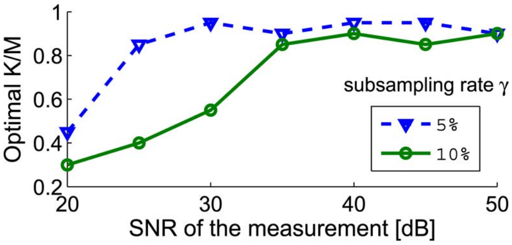


Fig. 8. Optimal ratio K/M of DASS for a fixed subsampling rate w.r.t. the SNR of the measurement (*Payerne*-temperature dataset). First, we note that K/M must be smaller than 1 according to Proposition 1. Second, we note that for an increasing quality of the measurements we can collect just $M \approx K$ samples, meaning that the reconstruction algorithm is less influenced by the noise and we need less samples. As a conclusive note, we would expect the plots to be monotonically increasing. However, this is not the case due to the random nature of the noise model and to the near-optimality of scheduling algorithm.

A. Components of DASS

In this section, we evaluate the key components of DASS, including the optimal choice of K, the cost function $\Theta(\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)$ in the sampling scheduling algorithm, and the performance of adaptive learning algorithms.

Optimal Choice of Dimension K: First, the larger the K the smaller the approximation error for any dataset, the only difference being the decay rate of such an error. Such aspect for the two considered dataset is depicted in Fig. 7, where the data has been normalized for $K = 0$. Note that for both datasets we have an exponential decay of the approximation error as a function of K. Therefore, there exists a small K for which the approximation error is negligible w.r.t. the Gaussian noise corrupting the measurements, as we have previously assumed.

As stated in Proposition 1, the overall reconstruction error ϵ is a function of both the approximation error ϵ_a(9) and the cost function $\Theta(\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)$(15). Generally, ϵ_a decreases with K and $\Theta(\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)$ increases with K, hence there is an optimal choice of K for minimizing the overall error. The optimal K depends on the data statistics, the subsampling rate, and the SNR of the measurement. By cross-validation, Fig. 8 shows the optimal ratio K/M for *Payerne*-temperature. We can see that DASS generally opts for a larger K when the SNR of measurement increases. This is intuitive since with better measurements we can afford a more complicated model with a weaker regularization.

Sampling Scheduling: The greedy algorithm proposed in Section III-C (Algorithm 4) finds an approximate solution of the sampling scheduling problem. By Proposition 1, $\Theta(\boldsymbol{\Phi}^t\boldsymbol{\Psi}^t)$

[6] We denote by *Payerne*-temperature the dataset of temperature measurements. The notation is similar for solar radiation.

[7] The processing costs of the considered sparse sensing methods are negligible.

TABLE IV
AVERAGE $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ ACHIEVED BY DIFFERENT SAMPLING SCHEDULING METHODS ($\gamma = 10\%$, SNR of the measurement = 30 dB)

Method / *Payerne*	uniform	random	Alg. 4
Temperature	0.56	4.9×10^{15}	0.54
Solar radiation	4.5×10^{5}	1.8×10^{15}	0.97

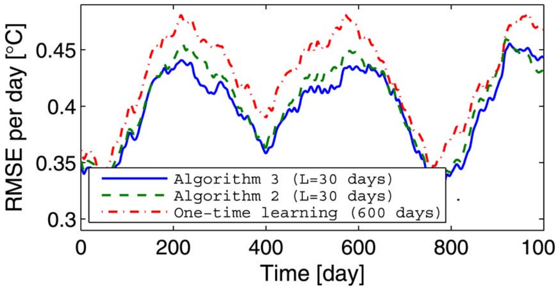


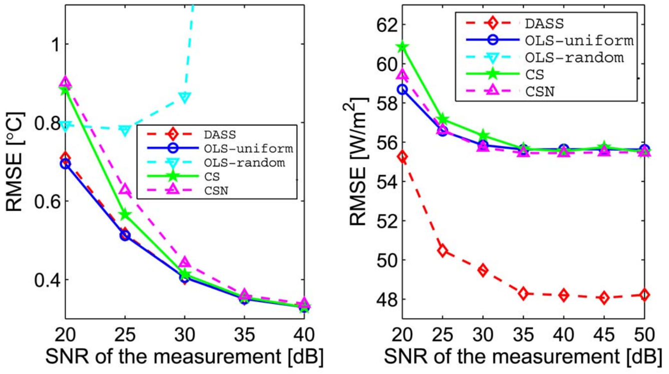
Fig. 9. Learning curves of DASS (*Payerne*-temperature, $\gamma = 10\%$, SNR of the measurement = 30 dB): comparisons of two online learning algorithms and a one-time learning algorithm with long backlog of past data. Note that Algorithm 3 achieves always the lowest error.

determines the reconstruction error. Table IV shows the value of $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ achieved by different sampling scheduling methods for different datasets. Note that a higher value indicates worse stability w.r.t. noise. We can see that the greedy algorithm achieves the best result for the two datasets. In particular, it is substantially better than uniform sampling for solar radiation data. For temperature data, since $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ of the uniform sampling strategy is already near the lower bound,[8] the greedy algorithm provides little improvement. In the next section, we demonstrate how these improvements translate into better reconstruction performance for DASS.

Learning Over Time: DASS is designed to learn the signal statistics from past data. In practical scenarios, a long backlog of data is usually unavailable and thus DASS should be designed to learn the model from scratch. We proposed Algorithm 2 and Algorithm 3 for this task. Fig. 9 shows the learning curves of these two algorithms over three years of data. As a benchmark, we considered an offline method that learns the model from 600 days of past data and is represented by the red-dotted curve. The offline method derives the transform matrix once for all from the complete signal. However, using this matrix may still provide worse results as the signal is non-stationary.

Note how Algorithm 2 and Algorithm 3 capture the signal statistics precisely. In particular, it is interesting to note that even if they use less data—the last 30 days—they are generally better than the offline method that considers 600 days of data. It is clear that the non-stationary signal model $\mathbf{\Psi}^t$ is captured only by the adaptive on-line algorithms. Moreover, Algorithm 3 with incremental PCA performs better than the buffer-based Algorithm 2.

In the following experiments, we will only consider Algorithm 3 due to its better performance and lower memory requirements.



(a) (b)

Fig. 10. Reconstruction error (RMSE) w.r.t. SNR of the measurement, of DASS, OLS-uniform, OLS-random, CS and CSN, respectively ($\gamma = 10\%$). The SNR is assumed to be accurately estimated. (a) *Payerne*-temperature. (b) *Payerne*-solar radiation. DASS is either on par with the best method, see (a), or significantly better, see (b). Note that in (b) OLS-random is not visible in the plot because it is significantly worse than the other methods.

B. DASS Versus Baseline Methods

Here, we compare DASS with the baseline methods introduced in Table II, namely, CS, CSN, OLS-random, and OLS-uniform.

Known Noise Level: For DASS, we need to choose the optimal K according to the cross-validation studied in Fig. 8. A similar parameter tuning is necessary for CSN, where ξ in Problem (19) represents the noise level. Therefore, whenever we consider the case of noisy measurements, an estimate of the SNR of the measurement is necessary to avoid degradations of the reconstruction quality.

In the first experiment, we assume that the estimation of the SNR is exact. Fig. 10 shows the comparison results of DASS, OLS-uniform, OLS-random, CS and CSN, for both temperature and solar radiation data. First, note that OLS-uniform generally performs better than the two CS-based schemes, especially in low SNR regime. In high SNR regime (>35 dB), OLS-uniform, CS and CSN tend to perform the same. Second, the bad performance of OLS-random indicates that random sampling is not a valid sampling strategy for neither temperature nor solar radiation signals. Third, while DASS and OLS-uniform perform almost equivalently for temperature data, we can note that DASS is substantially better for solar radiation data. This fact is in accordance with the analysis of $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ given in Table IV: if $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ is large due to uniform sampling (see solar radiation data), then the sampling scheduling algorithm of DASS (Algorithm 4) significantly improves the effectiveness of sensing while preserving the average sampling rate.

Error in Noise Estimation: In practice, the estimation of the noise level might be not exact. Here, we study the performance deviation of the considered algorithms when there is an error in such estimates. More precisely, we fix all the parameters and we vary the estimation error of the SNR and then measure the performance of the algorithms in terms of RMSE.

Fig. 11 shows the reconstruction error with respect to the estimation error of SNR, whereas the true SNR is 30 dB. We can see that DASS performs the best, and generally DASS and OLS-uniform are both stable w.r.t. errors in the SNR estimation.

[8] The lower bound of $\Theta(\mathbf{\Phi}^t\mathbf{\Psi}^t)$ is $\gamma = M/N$ if and only if $\mathbf{\Phi}^t\mathbf{\Psi}^t$ is a basis.

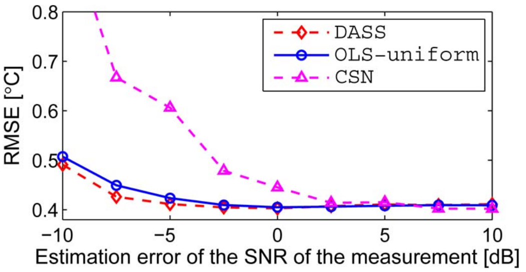


Fig. 11. Reconstruction error (RMSE) w.r.t. estimation error of the SNR of the measurement, of OLS-uniform, DASS and CSN, respectively (*Payerne-temperature*, $\gamma = 10\%$). The true SNR is 30 dB. Note that the proposed method is more robust to errors in the estimation of the noise power, when compared to other methods.

However, the performance of CSN degrades severely when the SNR is underestimated. The reason behind this large gap is that DASS and OLS-uniform both solve a least square problem (11), which could automatically reveal the unknown noise variance after optimization. On the contrary, CSN requires a known noise variance in the objective function, and hence it can be affected severely if the SNR is not correctly estimated *a priori*.

According to results given in Figs. 10 and 11, DASS is both more *accurate* and *robust* when compared to the state-of-the-art sparse sensing methods.

C. DASS on Multiple Sensor Nodes

As discussed in Section II, the concept of DASS can be extended to multiple sensor nodes by concatenating the collected samples in a single vector $\boldsymbol{y}$ and using the same strategy as for the single-node case.

Merging the data of all the spatial nodes possibly augments the correlation; DASS may exploits such correlation to reduce the sampling rate. In fact, if all the measurements collected by the sensors are linearly independent then DASS generates the same sampling scheduling that would have been optimized for each sensor individually. However, if there exists some correlation between the different sensor nodes, then DASS jointly optimizes the sensor scheduling so that the total average sampling rate is reduced.

We denote by *Joint DASS* the scheme that jointly reconstructs the signals of the WSN (Fig. 4), and *Independent DASS* the scheme that independently reconstructs the signals of each node. Note that in both schemes, sensor nodes are operating in a purely distributed manner; the difference is that *Joint DASS* aggregates the sensed data of all nodes and jointly processes them.

Fig. 12 shows the ratio between the subsampling rates of *Joint DASS* and *Independent DASS*, using the data-set *Valais*. We only show up to six nodes because the benefit stabilized at 30% with more than 4 nodes in the experiments. We can see that as the number of sensor nodes increases, the required sampling rate of *Joint DASS* gradually decreases. In particular, with 4 nodes we can reduce the number of samples by 70% with *Joint DASS*. Therefore, exploiting the spatial correlation further enhances the energy reduction of DASS. On the other hand, the benefit flatten out when we consider 5 or more sensor nodes. The intuition behind this phenomenon is that as the number of nodes increases, there are more nodes far apart from each other and hence the spatial correlations reduce accordingly.

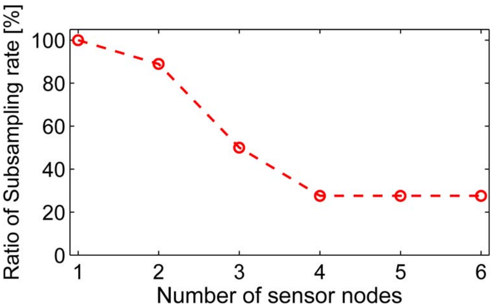
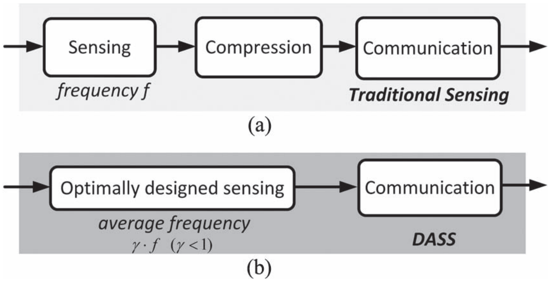


Fig. 12. Ratio of sampling rate between *Joint DASS* and *Independent DASS*, such that both schemes have the same reconstruction error (*Valais*, SNR of the measurement = 20 dB). Note that the joint scheme always reduces the number of samples required, this is due to the spatial correlation available in the sampled data.



(a)

(b)

Fig. 13. Two approaches to sensing in a WSN node. (a) Traditional scheme: collect periodical samples at a frequency f, compress and transmit the compressed data. (b) DASS: collect samples with an optimized temporal pattern at an average frequency $\gamma \cdot f$ and transmit the raw data.

VI. Energy Saving Over Traditional Data Collection Schemes

In Section V, we have shown that DASS achieves better sensing precision w.r.t the state-of-the-art *sparse sensing schemes*. In this section, we study the *overall energy saving* of DASS w.r.t. the *traditional data collection schemes* [20], [27]. The energy saving is particularly significant on platforms where the energy consumed for sensing is more pronounced. This is intuitive since DASS can substantially reduce the number of sensing samples. Nevertheless, our analysis shows that this saving is also noticeable on platforms with small sensing cost, e.g., a *Tmote-sky* node [24].

The traditional data collection schemes typically sample the physical field at a high frequency f as in (1) and then compress the samples to reduce the communication rate, see Fig. 13(a). In contrast, DASS collects measurements using an optimized sampling pattern and a reduced average sensing frequency $\gamma \cdot f$, where $\gamma < 1$. Then, each sensor node transmits the raw data points without any compression, see Fig. 13(b). In both traditional schemes and DASS, we aim at precisely reconstructing the signal $\boldsymbol{x}$. Furthermore, we restrict the discussion to the single node scenario. Our scheme can exploit both temporal and spatial correlations among the data gathered. However, the traditional schemes can only achieve energy savings by exploiting the temporal correlation on a single sensor node, because joint compression from multiple sensor nodes requires higher energy cost for the inter-node communications.

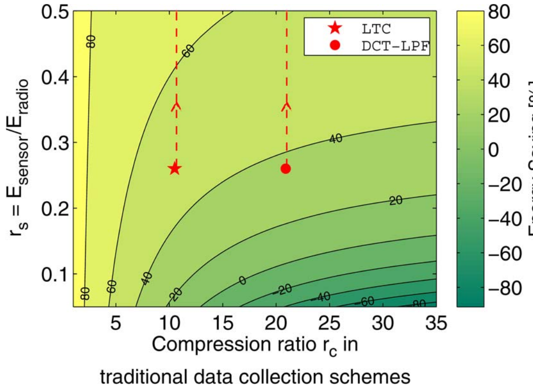


Fig. 14. Relative energy saving of DASS ($\gamma = 10\%$) w.r.t. traditional data collection schemes. The saving depends on the sensing platform (value of $\mathbf{r}_s$) and the compression ratio $\mathbf{r}_c$ in traditional sensing. The "star" and "circle" markers represent the energy saving on *Tmote-sky*, when DASS achieves the same reconstruction error as traditional sensing using LTC and DCT-LPF compression methods [27] (on dataset *Payerne*-temperature). The dashed lines indicate further savings when $\mathbf{r}$ increases, that is for sensors with higher energy costs.

It is clear that DASS reduces the energy consumption for the sensing operations over the traditional scheme. However, DASS may not necessarily consume less communication energy, since the compression ratio $\mathbf{r}_c$[9] used in traditional sensing is generally better than $1/\gamma$. In fact, existing data compression schemes can achieve a compression ratio $\mathbf{r}_c$ of $1.5 \sim 5$ for lossless coding [20], and $5 \sim 50$ for lossy coding [27], while a typical value of γ used in DASS is 0.1. Hence, there is a tradeoff between the energy saved on sensing and communications.

Such a tradeoff between the different energy consumption depends on platform-specific parameters. In particular, we denote the energy consumption for collecting and transmitting one sample as E_{sensor} and E_{radio}, respectively. An interesting figure is the ratio between the two energy values, that we denote as $\mathbf{r}_s = E_{sensor}/E_{radio}$. Intuitively, the larger $\mathbf{r}_s$, the larger the energy savings obtained by DASS. For the traditional data collection schemes, we assume that the compression step has a negligible energy cost. For DASS we use a subsampling rate of $\gamma = 0.1$, which means that 10% of the original signal is sampled and transmitted.

Under these assumptions, we can quantitatively analyze the relative energy savings of DASS w.r.t. the traditional sensing as a 2-D function of the platform parameter $\mathbf{r}_s$ and the compression ratio $\mathbf{r}_c$ achieved by the compression stage of the traditional scheme. Such function representing the energy saving is plotted in Fig. 14. We see that there is a line, indicated by the zero value, that defines where DASS is more energy-efficient than the traditional schemes. Above the line, a WSN consumes less energy if it uses DASS and vice versa. Note that DASS is only less efficient in the scenarios where the compression ratio $\mathbf{r}_c$ is very high and the platform parameter $\mathbf{r}_s$ is very low.

We also looked at the energy savings for a plausible real world scenario. More precisely, we consider *Tmote-sky*, a low-power sensing platform widely used in WSNs [24]; it has a photodiode sensor that measures the light intensity of the surroundings and communicates with others through short-range radio. We measured the two energy consumptions E_{sensor} and E_{radio} of *Tmote-sky* in a set of experiments, and an example of the results is given in Fig. 15. In particular, the experiments indicate that $\mathbf{r}_s = 0.26$. To evaluate the energy consumption of a traditional scheme, we need to choose a specific compression algorithm and measure the achieved $\mathbf{r}_c$. Zordan *et al.* [27] have recently compared various lossy compression algorithms and showed that DCT-LPF [27] achieves the best performance in terms of compression ratio. However, it is also a complex algorithm and may have a significant energy consumption on a resource-limited platform such as *Tmote-sky*. Therefore, we also consider a lightweight algorithm, LTC [21], that achieves the lowest energy consumption on WSN nodes if the energy cost for compression is considered.

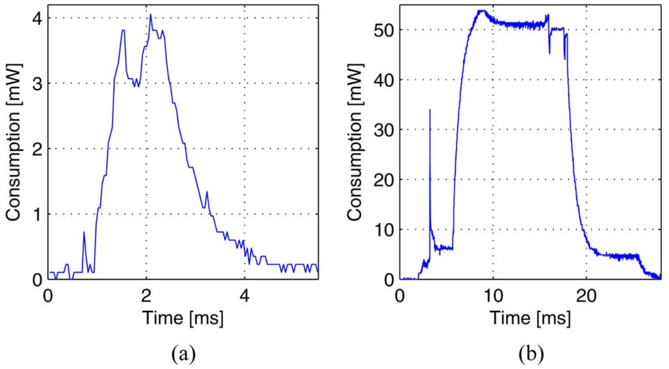


Fig. 15. Energy consumptions of a *Tmote-sky* sensor: (a) while the node measures one sample of light intensity (two-bytes), $E_{sensor} = 7.5 \times 10^{-6}$ J; (b) while the node transmits a packet with 24 bytes of payload, $24 \cdot E_{radio} = 6.9 \times 10^{-4}$ J.

Here, we ignore the energy cost of compression and we compare both algorithms with DASS. Note that, if we consider computational energy cost, the benefit of DASS will be even larger since it requires minimal on-board computation. We implement and evaluate the two algorithms on the dataset *Payerne*-temperature, and record the corresponding compression ratio $\mathbf{r}_c$ when their reconstruction errors are the same as those achieved by DASS.

The "star" and "circle" markers in Fig. 14 show the energy savings of DASS over a *Tmote-sky* that compresses the data with LTC and DCT-LPF, respectively. The energy savings for the two cases are equal to 50% and 35%. It is worth mentioning that the compression ratios achieved in Fig. 14 ("star" and "circle" markers) are specific of the considered meteorological datasets. There might be extreme cases where traditional compression schemes achieve a very high compression ratio (e.g., $r_c = 100$), and the respective saving falls below zero. However, we observe in Fig. 14 that the energy savings can still be obtained in such cases, if $\mathbf{r}_s$ increases due to a higher energy cost for sensing, as denoted by the dashed lines. This scenario could be realistic for many WSNs, in particular those using sensor belonging to the following two classes:

- Sensors with high energy consumption: for example an air pollution sensors consume $30 \sim 50$ mW instead of the 3 mW of a *Tmote-sky*'s light sensor.

[9] $\mathbf{r}_c$ equals uncompressed size/compressed size.

- Sensors with long sampling time: for example the anemometer, a sensor that measures wind's direction and strength, requires 1 $\sim$ 3 seconds of continuous measurement per sample instead of the 4 ms of the *Tmote-sky*'s light sensor.

VII. Conclusion

In this paper, we proposed DASS, a novel approach for sparse sampling that optimizes sparse sampling patterns for precisely recovering spatio-temporal physical fields. DASS is based on three main blocks. First, it adaptively learns the signal statistics from past data. Second, it dynamically adjusts the sampling pattern according to the time-varying signal statistics. Third, it recovers the signal from the limited amount of collected samples and according to the learnt signal statistics.

We demonstrated the effectiveness of DASS through extensive experiments using two real-world meteorological datasets. The results show significant improvements over the state-of-the-art methods. These improvements are more pronounced in the presence of significant spatial and/or temporal correlation in the sampled data by WSN.

We evaluated DASS on static WSNs; however, DASS is flexible and can be applied to other sensing scenarios such as mobile WSNs. For instance, sensors are installed on top of buses for collecting various environmental data along their trajectories [2]. The collected samples show strong correlation due to the fixed route periodically taken by the buses. In future work, we will analyze the advantages of an optimized sensing schedule in such cases, where the constraint is not the energy consumption but the relatively slow speed of sampling of certain pollution sensors.

Appendix A
Extension of DASS to Correlated Noise

In this appendix, we show how DASS deals with a sensor network with correlated sensing noises.

We assume the noise $\boldsymbol{\omega}$ to have zero mean and a correlation matrix $\Sigma_{\boldsymbol{\omega}}$ and recall that our measurements are defined as

$$\boldsymbol{x}+\boldsymbol{\omega},$$

where $\boldsymbol{x}$ is the actual physical field that we are measuring.

We define $\boldsymbol{\Omega} = (\Sigma_{\boldsymbol{\omega}})^{-0.5}$ and we whiten the measurements according as

$$\boldsymbol{\Omega}(\boldsymbol{x}+\boldsymbol{\omega}) = \boldsymbol{x}'+\boldsymbol{\omega}',$$

where the noise $\boldsymbol{\omega}'$ is now i.i.d. Therefore, we note that DASS can be easily applied to sensing scenarios where the noise is not i.i.d. by whitening the measured data before the processing.

References

[1] MeteoSwiss: The Federal Office of Meteorology and Climatology of Switzerland. [Online]. Available: http://www.meteoswiss.admin.ch

[2] K. Aberer *et al.*, "Opensense: Open community driven sensing of environment," in *Proc. IWGS*, 2010, pp. 39–42.

[3] E. Candès, "Compressive sampling," in *Proc. Int. Congr. Math., Invited Lectures*, 2006, pp. 1433–1452.

[4] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.

[5] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[6] P. G. Casazza, M. Fickus, J. Kovačević, M. Leon, and J. Tremain, "A physical interpretation of tight frames," in *Harmonic Analysis and Applications*. New York, NY, USA: Springer-Verlag, 2006, pp. 51–76.

[7] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *Proc. ACM STOC*, Jul. 2009, pp. 45–54.

[8] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approx.*, vol. 13, no. 1, pp. 57–98, 1997.

[9] M. F. Duarte, M. B. Wakin, D. Baron, and R. G. Baraniuk, "Universal distributed sensing via random projections," in *Proc. IPSN*, Nashville, TN, USA, Apr. 2006, pp. 177–185.

[10] M. Fickus, D. G. Mixon, and M. J. Poteet, "Frame completions for optimally robust reconstruction," in *Proc. SPIE*, 2011, vol. 8138, pp. 81380Q-1–81380Q-8.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.

[12] P. M. Hall, D. Marshall, and R. R. Martin, "Incremental eigenanalysis for classification," in *Proc. BMVC*, 1998, pp. 286–295.

[13] F. Ingelrest *et al.*, "Sensorscope: Applicationspecific sensor network for environmental monitoring," *ACM Trans. Sensor Netw.*, vol. 6, no. 2, Feb. 2010, Art. ID. 17.

[14] J. Johnson, "Thermal agitation of electricity in conductors," *Phys. Rev.*, vol. 32, no. 1, p. 97, Jul. 1928.

[15] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, Mar. 1993.

[16] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proc. Mobicom*, 2009, pp. 145–156.

[17] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, compression, and recovery for wsns: Sparse signal modeling and monitoring framework," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3447–3461, Oct. 2012.

[18] T. Raiko, A. Ilin, and J. Karhunen, "Principal component analysis for sparse high-dimensional data," in *Proc. ICONIP*, 2008, pp. 566–575.

[19] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1135–1146, Mar. 2014.

[20] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proc. SenSys*, 2006, pp. 265–278.

[21] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin, "Lightweight temporal compression of microclimate datasets," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, 2004, pp. 516–524.

[22] K. Viswanatha, S. Ramaswamy, A. Saxena, and K. Rose, "Error-resilient and complexity-constrained distributed coding for large scale sensor networks," in *Proc. IPSN*, 2012, pp. 293–304.

[23] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," in *Proc. IPSN*, 2007, pp. 331–339.

[24] G. Werner-Allen *et al.*, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Comput.*, vol. 10, no. 2, pp. 18–25, Mar./Apr. 2006.

[25] B. Widrow and I. Kollár, *Quantization Noise*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

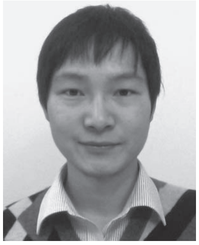
[26] X. Wu and M. Liu, "In-situ soil moisture sensing: Measurement scheduling and estimation using compressive sensing," in *Proc. IPSN*, 2012, pp. 1–12.

[27] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, "On the performance of lossy compression schemes for energy constrained sensor networking," *ACM Trans. Sensor Netw.*, vol. 11, no. 1, Nov. 2014, Art. ID. 15.

Zichong Chen received the B.Sc. degree (with honor) from Tsinghua University in June 2008, and the Ph.D. degree from Swiss Federal Institute of Technology (EPFL) in January 2013. He spent one year as a Post-doctoral Researcher at EPFL in 2013, and has been working at the Media Lab of Huawei research center in Hangzhou since 2014.

His research interests include computer vision, image/video processing, mobile sensing and statistical learning.

Juri Ranieri received both the B.S. and M.S. degrees in electronic engineering in 2007 and 2009, respectively, from Universitá di Bologna, Italy, and the Ph.D. degree from École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2014. From July to December 2009, he joined the Audiovisual Communications Laboratory (LCAV) at EPFL as a Visiting Student. From January 2010 to August 2010, he was with IBM Zurich to investigate the lithographic process as a signal processing problem. From April 2013 to July 2013, he was an intern at Lyric Labs of Analog Devices, Cambridge, MA, USA. His main research interests are inverse problems of physical fields and the spectral factorization of autocorrelation functions.



Runwei Zhang was born in Jiangsu, China, in 1988. He received his bachelor's degree from the Department of Automation, Tsinghua University, China, in 2010. During the summer of 2009, he was a Research Intern at Viterbi School of Engineering, University of Southern California. He is currently pursuing the Ph.D. degree with the Audiovisual Communications Lab, École Polytechnique Fédérale de Lausanne. His current research interest is mainly on network optimization under energy constraints.

Martin Vetterli (S'86–M'86–SM'90–F'95) received the Dipl. El.-Ing. degree from ETH Zurich (ETHZ), Switzerland, in 1981, the M.S. degree from Stanford University in 1982, and the Doctoratés Sciences degree from EPF Lausanne (EPFL), Switzerland, in 1986.

He was an Associate Professor in electrical engineering with Columbia University, New York, NY, USA, and a Full Professor in electrical engineering and computer sciences with the University of California, Berkeley, CA, USA, before joining the Communication Systems Division of EPFL. He held several positions at EPFL, including Chair of Communication Systems and Founding Director of the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS). From 2004 to 2011, he was Vice President of EPFL and since March 2011, he is the Dean of the School of Computer and Communications Sciences. He has been leading the Swiss National Science Foundation since January 1, 2013. He works in the areas of electrical engineering, computer sciences and applied mathematics. His work covers wavelet theory and applications, image and video compression, self-organized communications systems and sensor networks, as well as fast algorithms, and has led to about 150 journals papers. He is the co-author of three textbooks, with J. Kovacevic, *Wavelets and Subband Coding* (Prentice-Hall, 1995), with P. Prandoni, *Signal Processing for Communications*, (CRC Press, 2008) and with J. Kovacevic and V. Goyal, *Foundations of Signal Processing* (Cambridge University Press, 2012).

His work has won him numerous prizes, like Best Paper Awards from EURASIP in 1984 and the IEEE Signal Processing Society in 1991, 1996, and 2006, the Swiss National Latsis Prize in 1996, the SPIE Presidential Award in 1999, the IEEE Signal Processing Technical Achievement Award in 2001 and the IEEE Signal Processing Society Award in 2010. He is a Fellow of IEEE, ACM, and EURASIP, was a member of the Swiss Council on Science and Technology (2000–2004), and is an ISI highly cited researcher in engineering.