WSN-Control: Signal Reconstruction through Compressive Sensing in Wireless Sensor Networks

Giorgio Quer, Davide Zordan, Riccardo Masiero, Michele Zorzi, Michele Rossi
Department of Information Engineering, University of Padova, Italy
Email: {quergior, zordanda, masieror, zorzi, rossi}@dei.unipd.it

***Abstract*—The main contribution of this paper is the implementation and experimental evaluation of a signal reconstruction framework for Wireless Sensor Networks (WSNs). We design WSN-Control, an architecture to control a WSN from an external server connected to the Internet. Within such architecture, we implement a compression and recovery technique that combines Principal Component Analysis (PCA) and Compressive Sensing (CS) to reconstruct signals with many components from a sensor field through the collection of a relatively small number of samples, i.e., through incomplete representations of the actual signal. Overall, our experimental results show that a careful use of CS recovery is effective and can lead to a fully automated system for data gathering and reconstruction of real world and non-stationary signals in WSNs. In detail, WSN-Control effectively recovers signals showing some temporal and/or spatial correlation, from a relatively small number of samples, even below 20%, keeping the relative reconstruction error smaller than $5 \cdot 10^{-3}$. Signals with more irregular and quickly varying statistics are also recovered, even though the reconstruction error becomes highly dependent on the number of collected samples. CS minimization is obtained through the recently proposed NESTA optimization algorithm. Our implementation of CS recovery is available in [1].**

Keywords – Sensor Networks, Compressive Sensing, Data Gathering, Signal Reconstruction, Nesterov Minimization, Nesta.

I. Introduction

In this paper we present WSN-Control, a framework for the reconstruction of signals from Wireless Sensor Networks (WSNs) with a large number of nodes. Our application scenario includes data monitoring from a sensor field where the signal of interest shows spatial as well as temporal correlation. The relatively large number of deployed sensors allows the reconstruction of the physical phenomenon of interest with high accuracy. Nevertheless, the correlation structure of the signal makes it possible to acquire sufficiently accurate representations (in space and time) of the signal without collecting samples from every network node at each data collection round. In fact, our goal is the implementation and characterization of a data acquisition system that allows the signal reconstruction with high accuracy through the collection of data from a small number of sensors. This system shall be self-adaptable to changes in the signal statistics and tunable in terms of target reconstruction error. For the signal reconstruction we advocate the use of Compressive Sensing (CS) [2], [3], a technique that has been proved to be effective for the compression of video and images [4] and, recently, also for the reconstruction of signals in WSNs [5].

In the last few years the problem of collecting data from WSNs while performing joint compression and coding has been studied quite intensively. For example, [6] used classical source coding (see e.g., [7]) to design routing algorithms with re-encoding of information at the relay nodes, and [8] investigated the relation between the routing scheme and the location of aggregation/compression points in terms of the correlation structure of the data. We observe that these approaches are in general characterized by a significant cost in the transmission of control packets to implement the proposed joint data gathering and compression schemes. In fact, the transmissions from correlated sources must be coordinated in time (for example suppressing the transmission of redundant data) and this implies the synchronization of node activities at channel access and routing levels.

CS is based upon the fundamental fact that a signal having a sparse representation in one basis[1] can be recovered from a small number of projections into a second basis that is *incoherent* [2] with respect to the first one. In fact, for an M-sparse signal composed of N samples, approximately L ($L \simeq \xi M$, where $\xi \simeq 3$) of its projections into the second basis suffice for its reconstruction with high probability, where the value of ξ depends on how incoherent the two bases are. The implication of this is that instead of sampling the signal N times, ξM measurements (with $M \ll N$) may suffice for reconstruction. In a WSN scenario, a critical point for the successful application of CS to WSNs consists of finding suitable *incoherent* bases for the projections, without requiring any pre-processing by the data gathering nodes [9].

Another critical point at the core of the CS recovery of the original sparse signal is a constraint convex optimization problem, the minimization of the ℓ_1-norm of a vector in the solution space. In order to be applied into real WSNs, this algorithm should be both accurate and lightweight. Most of the algorithms proposed in the literature are not able to guarantee both accuracy and low complexity, e.g., second order methods such as interior points [10] are accurate, but have

The work presented in this paper has been performed within the MOSAICS project, "MOnitoring Sensor and Actuator networks through Integrated Compressive Sensing and data gathering", a research project funded by the University of Padova under grant no. CPDA094077.

[1] That is, many of the coefficients used to represent the signal in this basis are zero or negligible with respect to the other coefficients.

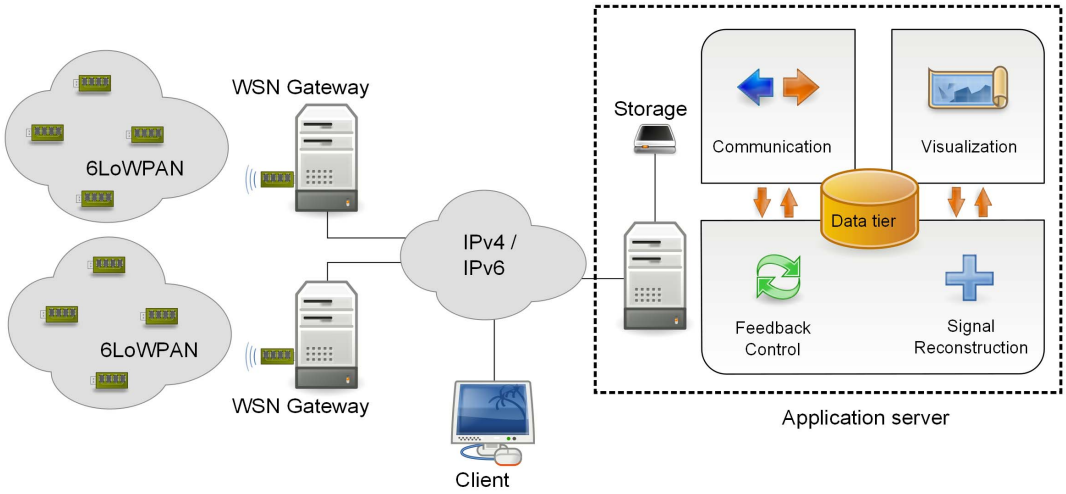


Fig. 1. WSN-Control architecture.

high computational complexity, while first order methods [11] are usually faster, but in order to be also accurate they need a large number of iterations. In WSN-Control we have integrated a recently proposed reconstruction algorithm to solve the convex optimization problem of CS, i.e., NESTA [12], that applies the Nesterov [13] minimization method to CS. NESTA relates the parameters of the optimization algorithm directly to the measurement noise and the desired accuracy, making the application of the algorithm very flexible since it does not require a long setup. The algorithm has been implemented within our framework for a direct application to WSNs.

One of the first papers proposing the use of CS in a distributed networking protocol is [14], which targets the energy efficient estimation of data in a WSN and proposes a technique that requires temporal synchronization of transmission activities and only works for single-hop networks. In [15] the authors consider a WSN where some nodes may fail; the goal is to identify these nodes through a distributed algorithm that exploits random projections, that are linear combinations of the states of the nodes. We shall observe that random projections are implemented through a preliminary pre-distribution phase where data is routed through a gossiping algorithm. While the approach gives good results in terms of reconstruction error its pre-distribution phase can be expensive in terms of consumed energy at the nodes. [16] exploits the data correlation both temporally and spatially. Signal measurements are projected into the incoherent basis at each source sensor, taking into account their temporal correlation. The spatial correlation is instead exploited at the base station. However, real WSN signals are not taken into account in the evaluation of the performance of the algorithm. [5] proposes a similar solution testing it with signals from actual WSN deployments and [17] analyzes this CS recovery technique from a Bayesian point of view, showing that under certain assumptions it is equivalent to maximum a posteriori (MAP) recovery.

The main contributions of the present paper are:

- the design and implementation of the WSN-Control architecture to access and control all the operations in a WSN from a server external to the WSN and connected to the Internet;
- the integration of the compression and recovery technique of [5] into the Signal Reconstruction and Feedback Control framework, that is directly applicable to a monitoring WSN, sensing any kind of spatially and temporally correlated signal;
- the integration of the NESTA CS-recovery technique into the Signal Reconstruction block for online reconstruction of the signal of interest at the server, together with the needed network algorithms, e.g., routing or feedback control, into a Web-based system for WSNs;
- the performance analysis of WSN-Control using signals from actual WSN deployments.

The paper is structured as follows. Section II presents the application scenario and the WSN-Control software architecture and Section III reviews the chosen CS-recovery technique. In Section IV we integrate the CS-recovery technique into the WSN-Control framework through the Signal Reconstruction and Feedback Control module to manage the compression and recovery mechanism. Results for real signals are given in Section V, and Section VI concludes the paper.

II. WSN-Control Architecture

A diagram of the WSN-Control architecture is given in Fig. 1. The WSN (possibly composed of separate sensor islands) is accessed through a number of WSN gateways. Sensor nodes adopt a protocol stack based on 6LoWPAN and run a suitable routing protocol to send the gathered data to the gateways. For a more detailed description of the protocols running in the WSN the reader is referred to [18].

The core of the WSN-Control system is the Application Server (see Fig. 1). This server is a Web application composed of the following blocks: 1) Visualization, 2) Communication and 3) Signal Reconstruction and Feedback Control.

1) *Visualization:* this block creates a 3D representation of the gathered data, and is also responsible for the user interface and for the related Applet and Java Server Page (JSP) technology [19].
2) *Communication:* this block is responsible for the reception of data from the WSN and for the transmission of data gathering requests to the sensor nodes. In addition, along with these requests, it also broadcasts *feedback* messages that set the transmission behavior of all the sensor nodes for the next data collection round.
3) *Signal Reconstruction and Feedback Control:* this block, at each data collection round, reconstructs the entire WSN signal from the received measurements. Feedback messages are generated and sent to the sensor nodes based on the time sample according to the technique in [5] so as to adapt the transmission behavior for the following data collection rounds. In particular, our aim is to minimize the number of nodes that send their measurements at each data collection round, while keeping the reconstruction error below a certain threshold.

The Web application is connected to a database that maintains the WSN measurements collected at previous data collection rounds. These are instrumental to the estimation of the signal statistics that, in turn, are used to obtain the incoherent transformation basis needed by CS, see Section III.

III. Applying CS: A Mathematical Overview

In this section we briefly overview the mathematical tools that are used in the compression and recovery methods implemented in the WSN-Control framework. We start describing Compressive Sensing (CS), then we detail how to obtain the transformation basis required by CS through Principal Component Analysis (PCA) [20], and finally we recall the key concepts of the Nesterov algorithm to efficiently solve the convex optimization problem in Compressive Sensing [12].

A. Compressive Sensing

CS is the mathematical tool exploited to perform distributed compression of an N-dimensional signal and centralized recovery of the signal at the server. We represent the signal as a column vector $\mathbf{x}^{(k)} \in \mathbb{R}^N$, where each element of the vector corresponds to the value measured by one of the N sensors, collected according to a fixed sampling rate at discrete times $k = 1, 2, \ldots, K$. The signal should be recovered at the server from an ideally small number of random projections of $\mathbf{x}^{(k)}$, namely $\mathbf{y}^{(k)} \in \mathbb{R}^L$ with $L \leq N$, according to the equation:

$$\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{x}^{(k)} . \tag{1}$$

In our framework $\mathbf{\Phi}^{(k)}$ captures the way in which the sensor data is gathered at the sink, i.e., $\mathbf{\Phi}^{(k)}$ is an $L \times N$ sampling matrix with a one in each row and at most a single one in each column, so vector $\mathbf{y}^{(k)}$ becomes a subsampled version of $\mathbf{x}^{(k)}$. In order to recover the original sensed signal $\mathbf{x}^{(k)}$ from $\mathbf{y}^{(k)}$, we assume for the moment that there exists an invertible $N \times N$ sparsifying matrix $\mathbf{\Psi}$ such that

$$\mathbf{x}^{(k)} = \mathbf{\Psi}\mathbf{s}^{(k)} , \tag{2}$$

where $\mathbf{s}^{(k)} \in \mathbb{R}^N$ and $\mathbf{s}^{(k)}$ is M-sparse with $M \leq L$, i.e. it has only M significant components, while the other $N - M$ are negligible with respect to the average energy per component, defined as:

$$E_s^{(k)} = \frac{\sqrt{\langle \mathbf{s}^{(k)}, \mathbf{s}^{(k)} \rangle}}{N} , \tag{3}$$

where for any two column vectors $\mathbf{a}$ and $\mathbf{b}$ of the same length, we define $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T\mathbf{b}$. At the receiver, it is equivalent to calculate a good approximation of either of the two vectors $\mathbf{x}^{(k)}$ or $\mathbf{s}^{(k)}$, as due to (2) there is a one-to-one mapping between them. Using (1) and (2) we can write

$$\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{x}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{\Psi}\mathbf{s}^{(k)} = \mathbf{\Theta}^{(k)}\mathbf{s}^{(k)} , \tag{4}$$

that is in general an ill-posed and ill-conditioned system with $\mathbf{\Theta}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{\Psi}$ of dimensions $L \times N$, since the number of variables N is larger than the number of equations L and a small variation in the input signal can cause a large variation in the output. However, since we design the matrix $\mathbf{\Psi}$ such that $\mathbf{s}^{(k)}$ is a sparse vector, as explained in Section III-B, we can invert the system and find the optimal solution with high probability solving a convex optimization problem [4], as described in Section III-C.

B. Learning the sparsifying matrix

In standard CS [2], the sparsifying basis $\mathbf{\Psi}$ is assumed to be given and fixed with time, but this is not the case for a realistic WSN scenario, where the signal of interest, $\mathbf{x}^{(k)}$, is unknown and its statistical characteristics can vary with time. To face this problem we advocate the use of Principal Component Analysis (PCA) [20], which is based on the Karhunen-Loève expansion, that is a method to represent the best M-terms approximation of a given N-dimensional signal, with $M < N$, exploiting the knowledge of the correlation structure of the signal. Since we do not have perfect knowledge of the correlation structure of the signal in a WSN monitoring application, we can have a good approximation through PCA, which is based on estimating the covariance matrix of the signal of interest $\mathbf{x}^{(k)}$. We assume to collect measurements of the signal at discrete times $k = 1, \ldots, K$, and from these measurements we can approximate the mean vector $\overline{\mathbf{x}}$ and the covariance matrix $\widehat{\mathbf{\Sigma}}$ as:

$$\overline{\mathbf{x}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}^{(k)} , \quad \widehat{\mathbf{\Sigma}} = \frac{1}{K}\sum_{k=1}^{K}(\mathbf{x}^{(k)} - \overline{\mathbf{x}})(\mathbf{x}^{(k)} - \overline{\mathbf{x}})^T . \tag{5}$$

Given the above equations, we consider the orthonormal matrix $\mathbf{U}$ whose columns are the eigenvectors of the covariance matrix $\widehat{\mathbf{\Sigma}}$, placed in decreasing order with respect to the corresponding eigenvalues. If we define the vector $\mathbf{s}^{(k)}$ as:

$$\mathbf{s}^{(k)} \stackrel{def}{=} \mathbf{U}^T(\mathbf{x}^{(k)} - \overline{\mathbf{x}}) , \tag{6}$$

by construction we have that the entries of vector $\mathbf{s}^{(k)}$ are in decreasing order, i.e., $s_1^{(k)} \geq s_2^{(k)} \geq \cdots \geq s_N^{(k)}$. Assuming that the instances $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}$ of the process $\mathbf{x}$ are correlated, as is often the case in WSN monitoring applications, there exists an $M \leq N$ such that all the component $s_i^{(k)}$ with

0. Initialize $\mathbf{x}_0$.

For $t \geq 0$,

1. Compute $\nabla f(\mathbf{x}_t)$.
2. Compute $\mathbf{r}_{t+1}$:

$$\mathbf{r}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{Q}_p} \left\{ p(\mathbf{x}, \mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle \right\}.$$

3. Compute $\mathbf{z}_{t+1}$:

$$\mathbf{z}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{Q}_p} \left\{ p(\mathbf{x}, \mathbf{x}_0) + \sum_{i=0}^{t} \alpha_i \langle \nabla f(\mathbf{x}_i), \mathbf{x} - \mathbf{x}_i \rangle \right\}.$$

4. Update $\mathbf{x}_{t+1}$:

$$\mathbf{x}_{t+1} = \tau_t \mathbf{z}_{t+1} + (1 - \tau_t)\mathbf{r}_{t+1} \, .$$

5. Stop if given criterion is satisfied.

TABLE I
THE NESTEROV MINIMIZATION ALGORITHM FOR SMOOTH FUNCTIONS.

$i = M+1, \ldots, N$ are negligible with respect to the average energy defined in (3), where the actual value of M depends on the spatial correlation of the signal. According to (6) we can write

$$\mathbf{x}^{(k)} = \overline{\mathbf{x}} + \mathbf{\Psi}\mathbf{s}^{(k)} \, , \tag{7}$$

where we have defined the sparsifying matrix $\mathbf{\Psi} = \mathbf{U}$. The N-dimensional vector $\mathbf{s}^{(k)}$ obtained through PCA turns out to be M-sparse, so it can be efficiently recovered with CS, solving a convex optimization problem as detailed in the next section.

C. Convex optimizer

At time k, in order to reconstruct the original signal $\mathbf{x}^{(k)}$ at the receiver we must invert the ill-posed system defined by Eq. (4), where $\mathbf{\Psi}$ is obtained as detailed in the previous section. For simplicity of the notation, we hereby assume that $\overline{\mathbf{x}} = 0$, as this only counts as an additional term. Moreover, under the assumption that $\mathbf{s}^{(k)}$ has a certain degree of sparsity and under specific assumptions on the matrix $\mathbf{\Theta}^{(k)}$ (that are verified in our case, see, e.g., [21]), inverting (4) has been proven [2] to be equivalent to solving the convex minimization problem

$$\begin{aligned} \hat{\mathbf{s}}^{(k)} &= \underset{\mathbf{s}^{(k)}}{\operatorname{argmin}} \|\mathbf{s}^{(k)}\|_{\ell_1} \\ \text{s.t.} \quad \mathbf{y}^{(k)} &= \mathbf{\Theta}^{(k)}\mathbf{s}^{(k)} \, , \end{aligned} \tag{8}$$

where $\|\cdot\|_{\ell_1}$ is the ℓ_1-norm of a vector, i.e., for a given vector $\mathbf{a}$ of N elements, $\|\mathbf{a}\|_{\ell_1} = \sum_{i=1}^{N} |a_i|$. In WSN-Control, as suggested in [12], this optimization problem is solved through NESTA, which is an application to CS of the Nesterov minimization algorithm extended to non-smooth functions. As a first step, in the following we review the Nesterov minimization method [22]. Subsequently, we discuss the extension of this method to non-smooth functions and finally we explain how it is applied to CS.

Nesterov minimization: this method solves convex optimization problems of the type

$$\min_{\mathbf{x}\in\mathcal{Q}_p} f(\mathbf{x}) \, , \tag{9}$$

where the convex function to minimize, $f(\mathbf{x}) : \mathcal{Q}_p \to \mathbb{R}$, is defined in the convex set $\mathcal{Q}_p \subseteq \mathbb{R}^N$, e.g., of the form

$$\mathcal{Q}_p = \{\mathbf{x} : \mathbf{b} = \mathbf{Q}\mathbf{x}\} \, , \tag{10}$$

where $\mathbf{Q}$ is an $M \times N$ matrix, with $M \leq N$, and $\mathbf{b} \in \mathbb{R}^M$ is a given constant vector. Moreover, the function $f(\mathbf{x})$ must be smooth, i.e., it must be differentiable and its gradient must be Lipschitz:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_{\ell_2} \leq C\|\mathbf{x} - \mathbf{y}\|_{\ell_2} \, , \tag{11}$$

where $C > 0$ is a constant [22]. The algorithm proposed by Nesterov to solve (9) is listed in Table I and discussed in the following: NESTEROV MINIMIZATION ALGORITHM

0. Initialize $\mathbf{x}_0$: a possible initialization method for $\mathbf{x}_0$ is $\mathbf{x}_0 = \mathbf{Q}^T\mathbf{b}$. Set $t = 0$.
1. Computation of the gradient of $f(\mathbf{x}_t)$.
2. Computation of $\mathbf{r}_{t+1}$: $\mathbf{r}_{t+1}$ is a first sequence of vectors that converges towards the minimum of $f(\mathbf{x})$. The first term $p(\mathbf{x}, \mathbf{x}_t)$ is a proximity function (also referred to as *penalty function*) weighing more those points that are farther away from the current solution $\mathbf{x}_t$. We have

$$p(\mathbf{x}, \mathbf{x}_t) = \frac{C}{2}\|\mathbf{x} - \mathbf{x}_t\|_{\ell_2}^2 \, . \tag{12}$$

   The second term corresponds to a gradient descent minimization with step $|\mathbf{x} - \mathbf{x}_t|$. Note that the step size is controlled by the first term, which penalizes large deviations from $\mathbf{x}_t$.
3. Computation of $\mathbf{z}_{t+1}$: $\mathbf{z}_{t+1}$ is a second sequence of vectors that also converges to the minimum of $f(\mathbf{x})$. The first term is equal to (12) but with $\mathbf{x}_0$ in place of $\mathbf{x}_t$. The second term corresponds to a gradient descent minimization accounting for all previous partial solutions $\mathbf{x}_i$, $i \leq t$.
4. The solution is updated as a weighted average of $\mathbf{r}_t$ and $\mathbf{z}_t$, using a suitable combination coefficient τ_t.
5. A possible stopping criterion, originally proposed in [12], is the following. Let $\overline{f}(\cdot)$ be the average of $f(\cdot)$ during the last ten iterations

$$\overline{f}(\mathbf{x}_t) = \frac{1}{\min\{10, t\}} \sum_{i=1}^{\min\{10,t\}} f(\mathbf{x}_{t-i}) \, . \tag{13}$$

   The algorithm is terminated when

$$\Delta f = \frac{|f(\mathbf{x}_t) - \overline{f}(\mathbf{x}_t)|}{\overline{f}(\mathbf{x}_t)} < \delta \, . \tag{14}$$

The coefficients α_t, τ_t must be chosen to guarantee convergence, see [13]. The impact of the constant δ is studied in Section V.

Application of Nesterov minimization to CS: reference [13] extended the Nesterov algorithms to *non-smooth*

functions, showing that this extension is possible when these functions can be re-written as a maximization problem. Subsequently, with the NESTA algorithm [12], the theory of [13] has been applied to CS. In detail, (8) is re-written as

$$\min_{\mathbf{s}^{(k)} \in \mathcal{Q}'_p} \|\mathbf{s}^{(k)}\|_{\ell_1} \ , \tag{15}$$

where $\mathcal{Q}'_p$, at time k, is the convex set defined as

$$\mathcal{Q}'_p = \left\{ \mathbf{s}^{(k)} : \|\mathbf{y}^{(k)} - \mathbf{\Theta}^{(k)}\mathbf{s}^{(k)}\|_{\ell_2} \leq \epsilon \right\} , \tag{16}$$

where $\mathbf{s}^{(k)} \in \mathbb{R}^N$ is a sparse vector with only M significant elements with $M \ll N$, $\epsilon \geq 0$ is a small number and $\mathbf{\Theta}^{(k)}$ is an $L \times N$ and real matrix having linearly independent rows ($M \leq L \leq N$). In [12], $\|\mathbf{s}^{(k)}\|_{\ell_1}$ is re-written as a maximization problem, i.e.,

$$\|\mathbf{s}^{(k)}\|_{\ell_1} = \max_{\mathbf{u} \in Q_d} \langle \mathbf{u}, \mathbf{s}^{(k)} \rangle \ , \tag{17}$$

where $Q_d \subseteq \mathbb{R}^N$ is the ℓ_∞ ball defined as

$$Q_d = \{ \mathbf{u} : \|\mathbf{u}\|_\infty \leq 1 \} \ . \tag{18}$$

Hence, $\|\mathbf{s}^{(k)}\|_{\ell_1}$ is approximated by the smooth function

$$\|\mathbf{s}^{(k)}\|_{\ell_1} \simeq f_\mu(\mathbf{s}^{(k)}) = \max_{\mathbf{u} \in Q_d} \left\{ \langle \mathbf{u}, \mathbf{s}^{(k)} \rangle - \frac{\mu}{2} \|\mathbf{u}\|^2_{\ell_2} \right\} . \tag{19}$$

It can be shown that $\nabla f_\mu(\mathbf{s}^{(k)})$ is Lipschitz with constant $C = 1/\mu$ and thus the Nesterov optimization algorithm can be applied to such function. In conclusion, the NESTA method of [12] amounts to solving

$$\min_{\mathbf{s}^{(k)} \in \mathcal{Q}'_p} \max_{\mathbf{u} \in Q_d} \left\{ \langle \mathbf{u}, \mathbf{s}^{(k)} \rangle - \frac{\mu}{2} \|\mathbf{u}\|^2_{\ell_2} \right\} . \tag{20}$$

Note that (20) can now be solved using the algorithm in Table I, where the inner maximization problem (19) can be solved in linear time through the sequential evaluation of the elements of $\mathbf{u}$. In fact, defining $\hat{\mathbf{u}}$ as

$$\hat{\mathbf{u}} = \operatorname*{argmax}_{\mathbf{u} \in Q_d} \left\{ \langle \mathbf{u}, \mathbf{s}^{(k)} \rangle - \frac{\mu}{2} \|\mathbf{u}\|^2_{\ell_2} \right\} \tag{21}$$

we have

$$\hat{u}_i = \begin{cases} s_i^{(k)}/\mu & |s_i^{(k)}| \leq \mu \\ +1 & |s_i^{(k)}| > \mu \text{ and } s_i^{(k)} > 0 \\ -1 & |s_i^{(k)}| > \mu \text{ and } s_i^{(k)} < 0 \end{cases} , \ i = 1, \ldots, N \ , \tag{22}$$

IV. Integrating CS into WSN-Control

The algorithms described in the previous section are implemented in the block *Signal Reconstruction and Feedback Control* of Fig. 1. This block is responsible for sending to the network nodes the instructions to perform data gathering, at each time sample k, and to reconstruct the original signal $\mathbf{x}^{(k)}$ from its compressed version $\mathbf{y}^{(k)}$. The block is divided into three components: 1) a database responsible for storing the data received from the WSN, 2) the Feedback and Control component that handles incoming data, passes them to the reconstruction block and sends data gathering requests to the WSN nodes and 3) the Signal Reconstruction component, which reconstructs the original signal using PCA and CS. In this section, we describe the implementation of each component of this block. Note that our architecture was designed in a flexible and modular manner and, in turn, can support different reconstruction and data gathering techniques.

Database: The database receives at discrete times the measurements from the WSN. Each measure is described by four fields:

- ID: the identifier of the sensor that transmitted the data;
- type: the type of data transmitted, e.g., in our testbed they can represent temperature, luminosity in two different ranges (320–730 or 320–1100 nm), battery voltage or humidity;
- value: the corresponding value measured by the sensor specified in the ID field;
- time stamp: expressed in discrete time units $k = 1, 2, \ldots$.

Feedback Control: This component is responsible for the synchronization of the entire reconstruction process. It implements a data gathering technique [5] that alternates between a *training phase* and a *monitoring phase*. During the former all sensors transmit to the gateway their data, which is used to estimate the signal statistics in terms of mean and covariance, see (5). This information is subsequently used to reconstruct the WSN signal in the latter phase, where only a small fraction of nodes transmit.

Specifically, the training phase lasts K_1 time samples. Here, the Feedback Control sets the probability of transmission for the sensor nodes to $p_{\text{tx}} = 1$ and sends a feedback message with this p_{tx} to all of them. In this way, all sensors transmit their data and the server during the training phase receives K_1 complete vectors with N measurements[2]. During the monitoring phase, which lasts K_2 time samples, the Feedback Control sets $p_{\text{tx}} = L/N$, with $L < N$, and sends this probability to the WSN nodes. Thus, for K_2 time samples each sensor transmits its data with this probability (i.e., on average only L of the N sensors transmit at each time sample). Training and monitoring phases are interleaved as follows:

$$\ldots, \underbrace{\mathbf{y}^{(k)}, \ldots, \mathbf{y}^{(k+K_1-1)}}_{\substack{\text{training phase} \\ p_{\text{tx}} = 1 \\ \dim(\mathbf{y}^{(j)}) = N}}, \underbrace{\mathbf{y}^{(k+K_1)}, \ldots, \mathbf{y}^{(k+K_1+K_2-1)}}_{\substack{\text{monitoring phase} \\ p_{\text{tx}} = L/N \\ E\left[\dim(\mathbf{y}^{(j)})\right] = L}}, \ldots$$

where $\dim(\mathbf{y}^{(j)})$ is the number of the components of $\mathbf{y}^{(j)}$ and $E[\dim(\mathbf{y}^{(j)})]$ is the expected value of $\dim(\mathbf{y}^{(j)})$.

Signal Reconstruction: The Signal Reconstruction block is responsible for recovering a good approximation of the original signal from the data stored in the database. As a first step, at each time k it lists the corresponding values stored in the database and creates the vector $\mathbf{y}^{(k)}$. Thus, it calculates the

[2]In case there is an error during the transmissions, so that not all N measurements are received, the server simply considers as training set to perform (5) the last complete training set received.

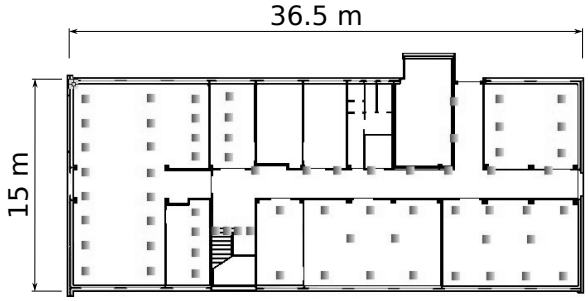


Fig. 2. Layout of the WSN testbed at the University of Padova.

matrix $\mathbf{\Phi}^{(k)}$ based on the corresponding IDs, according to (1). If k is a time sample in the training phase, it just calculates $\hat{\mathbf{x}}^{(k)}$ inverting (1), since in this case $\mathbf{\Phi}^{(k)}$ is an $N \times N$ invertible matrix, and records this data into the database. Instead, if k corresponds to a sample in the monitoring phase, the Signal Reconstruction block uses $\mathbf{y}^{(k)}$, $\mathbf{\Phi}^{(k)}$ and the samples of the last recorded training set $\hat{\mathbf{x}}^{(j)}, \ldots, \hat{\mathbf{x}}^{(j+K_1-1)}$ to infer the reconstructed value of $\hat{\mathbf{x}}^{(k)}$. In order to do so, this block implements the PCA and CS techniques described in Section III. The statistics necessary to build the sparsifying matrix $\mathbf{\Psi}$ are derived from the samples of the recorded training set through (5), and the convex optimization problem in (8) is solved using the algorithms described in Section III-C.

V. Experimental Results

The proposed signal reconstruction framework has been implemented in the WSN testbed of Fig. 2, which has been deployed on the ground floor of the Department of Information Engineering, University of Padova, Italy, including $N = 68$ TmoteSky wireless sensor nodes that communicate with IEEE 802.15.4 radio transceivers. These sensors can measure five different signals: temperature, humidity, luminosity in two different ranges (320–730 and 320–1100 nm), and the voltage of their battery. For the performance analysis we have considered temperature and humidity, which have high spatial and temporal correlation, and luminosity, that instead is an unpredictable signal with high variability in space and time. In the following we test the performance of the NESTA algorithm integrated in the Signal Reconstruction and Feedback Control module and of the whole module implemented in the testbed, for a measurement campaign of about 10^3 samples taken at a sampling period of $3 \cdot 10^2$ s.

The NESTA algorithm has been rewritten in C++ and integrated into the signal reconstruction module of WSN-Control. The implementation of the algorithm is available from [1], while we refer the reader to [12] for the original version of NESTA. As a first step, we selected the parameters of the optimization δ, which determines when the algorithm stops, see (14), ϵ, which is a constant used to relax the equality in (10), see (16) and μ, which is related to the accuracy of the approximation of the ℓ_1-norm, see (19). In our results we have fixed the value of μ, keeping into account that smaller values imply a better approximation and bigger values imply a faster convergence. We have chosen $\mu = 10^{-2}$, since by

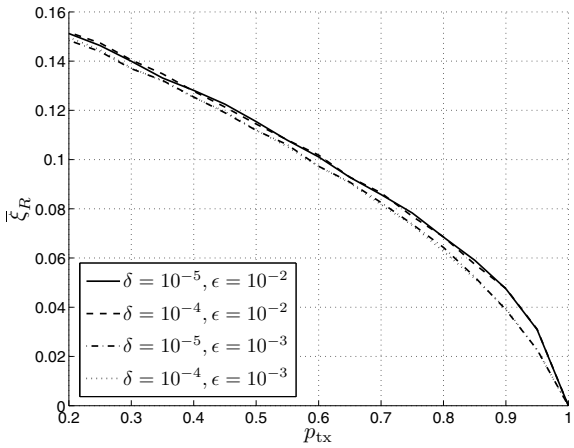


Fig. 3. Performance of the NESTA algorithm for a luminosity signal, varying δ and ϵ.

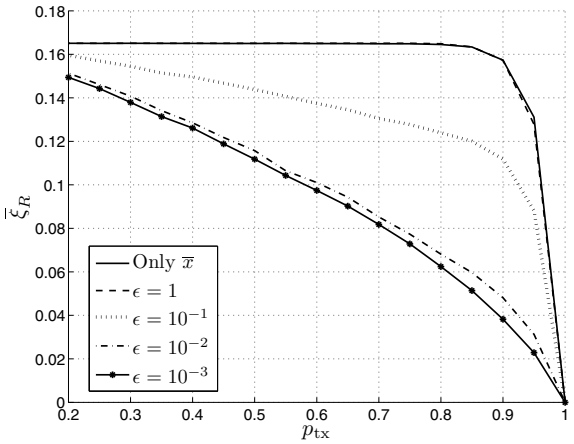


Fig. 4. Performance of the NESTA algorithm for a luminosity signal, with $\delta = 10^{-5}$ and varying ϵ.

simulations we have noticed that this value only marginally impacts the result of the minimization while guaranteeing a fast convergence. In Figs. 3 and 4 we represent the performance of the algorithm varying δ and ϵ for the luminosity signal. The x-axis represents p_{tx} during the monitoring phase, see Section IV, while the y-axis represents the average relative reconstruction error in the whole simulation ($k = 1, \ldots, K$), defined as

$$\overline{\xi}_R = \frac{1}{K} \sum_{k=1}^{K} \xi_R^{(k)} \,, \tag{23}$$

where $\xi_R^{(k)}$ is the relative reconstruction error at time k, i.e.,

$$\xi_R^{(k)} = \frac{\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}^{(k)}\|_2}{\|\mathbf{x}^{(k)}\|_2} \,. \tag{24}$$

In Fig. 3 we notice that varying the value of δ in the range $[10^{-4}, 10^{-5}]$ does not affect the recovery performance; for our results we picked $\delta = 10^{-5}$. In Fig. 4 instead we vary $\epsilon \in [10^{-3}, 1]$. For $\epsilon \leq 10^{-2}$, the reconstruction error is quite insensitive to ϵ, while it sharply increases for larger

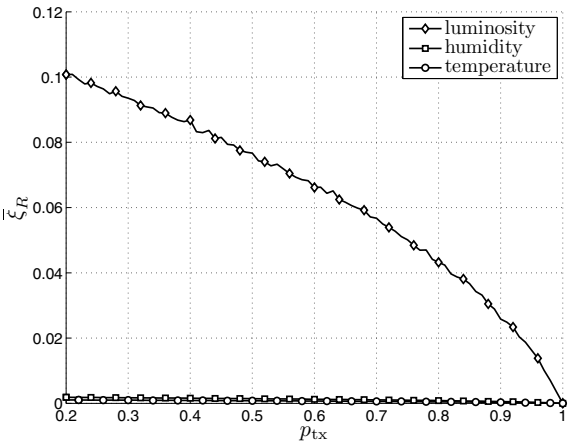


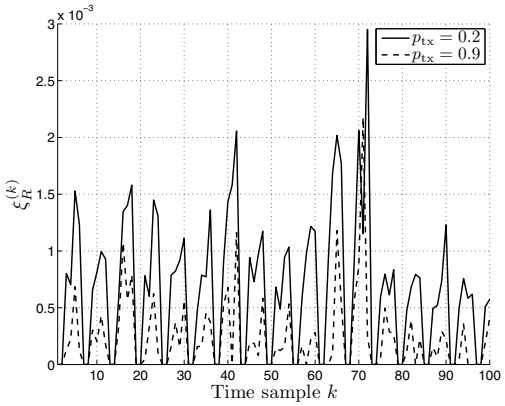
Fig. 5. Average reconstruction error for the Signal Reconstruction and Feedback Control module, signals from University of Padova testbed.

values of ϵ. We decided to adopt $\epsilon = 10^{-2}$ as it provided reasonable performance across all our experiments. It is worth noting that for $\epsilon = 1$ we reach the upper bound of the reconstruction error, represented by the error in the case where we have no optimization, but we consider as reconstructed signal $\hat{\mathbf{x}}^{(k)} = \overline{\mathbf{x}}$, i.e., the average calculated over the previous training set, see (5). This is due to the fact that for a large value of ϵ the constraints are relaxed, so the set of possible solutions $\mathcal{Q}'_p$ includes also the null solution $\hat{\mathbf{s}}^{(k)} = 0$, that is the sparsest one in Eq. (8). In this case the algorithms (7) gives as outcome $\hat{\mathbf{x}}^{(k)} = \overline{\mathbf{x}}$.

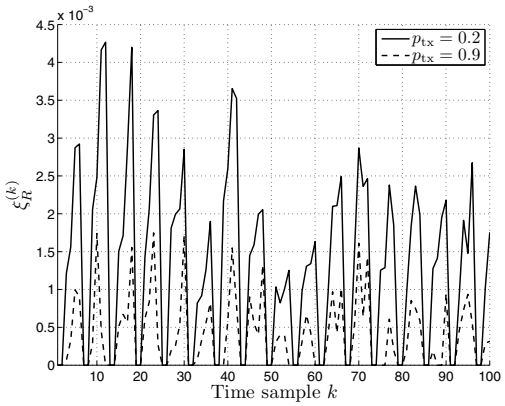
In Fig. 5 we show the performance of the Signal Reconstruction and Feedback Control module as a function of p_{tx} for signals sensed in the WSN testbed at the University of Padova. In this set of experiments we set the length of the training phase to $K_1 = 2$ and the length of the monitoring phase to $K_2 = 4$. We note that for highly correlated signals like temperature and humidity, the reconstruction error is sufficiently small, i.e., below $\overline{\xi}_R < 0.01$, for relatively small values of $p_{\text{tx}} \approx 0.2$. Instead for more unpredictable signals like luminosity the error increases sharply with decreasing p_{tx}; in this case an error below $\overline{\xi}_R < 0.05$ is achievable with $p_{\text{tx}} > 0.75$.

In Fig. 6 we analyze the behavior of the reconstruction error $\xi_R^{(k)}$ of the three selected signals as a function of time k for two values of $p_{\text{tx}} = 0.2$ and 0.9. First of all, we observe an oscillatory behavior of $\xi_R^{(k)}$, that is due to the alternation of training and monitoring phases. Specifically, during training phases $\xi_R^{(k)} = 0$ as the signal is entirely measured, whereas during monitoring phases $\xi_R^{(k)}$ presents higher values. This is because the statistics calculated during any given training phase in (5) tends to become outdated during the subsequent monitoring phase and, as a consequence, the reconstruction performance deteriorates. As expected, we notice that a large p_{tx}, e.g., $p_{\text{tx}} = 0.9$, gives a significantly smaller reconstruction error than, e.g., $p_{\text{tx}} = 0.2$.

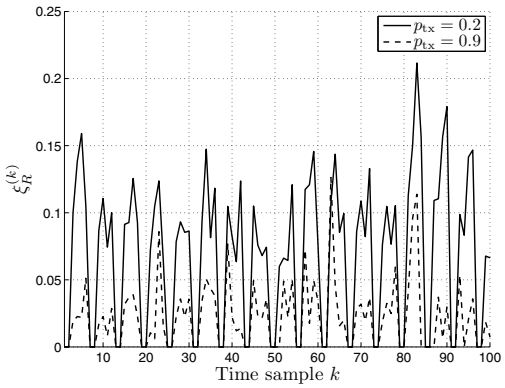
Finally, in order to validate our approach, we have simulated



(a) Temperature signal.



(b) Humidity signal.



(c) Luminosity signal.

Fig. 6. Reconstruction error as a function of time k.

our framework testing it with other real signals from the EPFL WSN deployment LUCE, see [23], that is composed of $N = 81$ sensor nodes. The parameters of the recovery method are the same that we used for the previous figures. In Fig. 7 we represent the average reconstruction error $\overline{\xi}_R$ for humidity and for surface and ambient temperature. The signals have been gathered in a measurement campaign of about 10^3 samples taken at a sampling period of $9 \cdot 10^2$ s. The first signal is highly correlated and as such its error is bounded under $\overline{\xi}_R < 0.01$. On the contrary, the remaining two signals have a more unpredictable behavior and show similar performance to

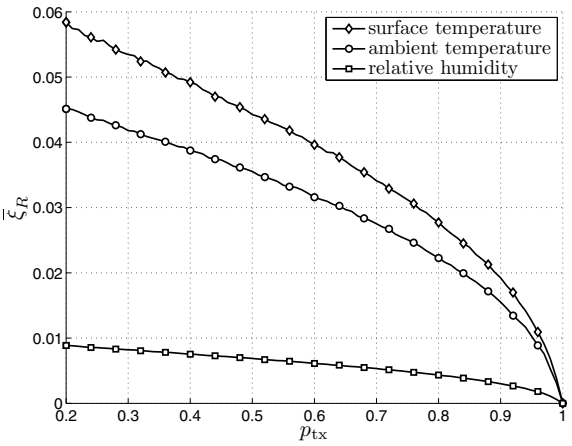


Fig. 7. Average reconstruction error for the Signal Reconstruction and Feedback Control module, signals from EPFL LUCE WSN deployment.

the luminosity signal of Fig. 5. We remark that, although the experimental data used for Figs. 5 and 7 have been collected from different testbeds, the corresponding plots maintain the same shape, meaning that our recovery algorithm is stable and tunable as a function of the signal statistics.

VI. Conclusions

This paper presents the Signal Reconstruction and Feedback Control module, which uses Compressive Sensing for the reconstruction of signals in WSNs, addressing its integration into the WSN-Control architecture. CS is efficiently implemented according to the NESTA optimization algorithm. Experimental results show that the signal gathering and reconstruction technique adopted is effective in WSNs, leading to savings in terms of number of transmitting nodes (and thus in terms of energy cost for the WSN). These savings depend on the signal statistics and, for correlated signals, can be quantified in a substantial reduction of the communication cost with respect to the case where all WSN measurements are collected. For instance, with $p_{\mathrm{tx}} = 0.2$, $K_1 = 2$ and $K_2 = 4$ we can obtain savings of about $1-(K_1+K_2 p_{\mathrm{tx}})/(K_1+K_2) \simeq 53\%$ in terms of number of transmissions. These techniques have been tested in an environmental monitoring WSN, but they can be used in a wide range of applications that aim to compress and recover a distributed signal with a certain level of spatial and temporal correlation.

Our future work will be devoted to the adaptive and autonomous selection of p_{tx} and to an improved estimation procedure for the relevant signal statistics of (5) that will work on incomplete signal representations. This will allow the removal of the training phase, which will lead to a further reduction of the communication cost.

References

[1] D. Zordan, G. Quer, R. Masiero, and M. Rossi, "C++ Implementation of the NESTA Algorithm for Signal Recovery through Compressive Sensing," May 2010. [Online]. Available: http://www.dei.unipd.it/~rossi/software.html

[2] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[3] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 4036–4048, Apr. 2006.

[4] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[5] R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "Data Acquisition through joint Compressive Sensing and Principal Component Analysis," in *IEEE Globecom 2009*, Honolulu, Hawaii, USA, Nov.-Dec. 2009.

[6] A. Scaglione and S. D. Servetto, "On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks," in *ACM MOBICOM*, Atlanta, GA, USA, Sep. 2002.

[7] D. Slepian and J. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Trans. on Information Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[8] S. Pattem, B. Krishnamachari, and R. Govindan, "The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks," in *Int. Conf. on Information Processing in Sensor Networks (IPSN)*, Berkeley, CA, USA, Apr. 2004.

[9] 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 *Information Theory and Applications Workshop (ITA 2009)*, San Diego, CA, US, Feb. 2009.

[10] E. Candès, "ℓ_1 magic," Tech. Rep., 2007.

[11] A. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, no. 1, pp. 586–597, 2007.

[12] S. Bercker, J. Bobin, and E. J. Candés, "NESTA: a fast and accurate first order method for sparse recovery." *Submitted for publication*. [Online]. Available: http://www-stat.stanford.edu/~candes/papers/NESTA.pdf

[13] Y. E. Nesterov, "Smooth minimization of non–smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, may 2005.

[14] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Joint source-channel communication for distributed estimation in sensor networks," *IEEE Trans. on Information Theory*, vol. 53, no. 10, pp. 3629–3653, Oct. 2007.

[15] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressive Sensing for Networked Data: a Different Approach to Decentralized Compression," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92–101, Mar. 2008.

[16] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed Compressed Sensing of Jointly Sparse Signals," in *Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, Oct. 2005.

[17] R. Masiero, G. Quer, M. Rossi, and M. Zorzi, "A Bayesian Analysis of Compressive Sensing Data Recovery in Wireless Sensor Networks," in *The International Workshop on Scalable Ad Hoc and Sensor Networks, SASN'09*, Saint Petersburg, Russia, Oct. 2009.

[18] A. P. Castellani, N. Bui, P. Casari, M. Rossi, Z. Shelby, and M. Zorzi, "Architecture and Protocols for the Internet of Things: A Case Study," in *International Workshop on the Web of Things (WoT)*, Mannheim, Germany, Mar. 2010.

[19] D. Heffelfinger, *Java EE 5 Development with NetBeans 6*. Packt Publishing, 2008.

[20] C. R. Rao, "The Use and Interpretation of Principal Component Analysis in Applied Research," *Sankhya: The Indian Journal of Statistics*, vol. 26, pp. 329–358, 1964.

[21] E. Candès and M. Wakin, "An Introduction to Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 2, no. 25, pp. 21–30, Mar. 2008.

[22] Y. E. Nesterov, "A method for unconstrained convex minimization problem with rate of convergence $\mathcal{O}(1/k^2)$," *Doklady AN USSR (Translated as Soviet Math. Docl.)*, vol. 269, no. 3, pp. 543–547, 1983.

[23] "EPFL LUCE SensorScope WSN," Last time accessed: September 2009. [Online]. Available: http://sensorscope.epfl.ch/