CONTRIBUTIONS TO ENERGY CONSERVATION IN WIRELESS SENSOR NETWORKS

Felipe da Rocha Henriques

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientadores: Eduardo Antônio Barros da Silva
Lisandro Lovisolo

Rio de Janeiro
Dezembro de 2015

CONTRIBUTIONS TO ENERGY CONSERVATION IN WIRELESS SENSOR NETWORKS

Felipe da Rocha Henriques

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Lisandro Lovisolo, D.Sc

Prof. Marcello Luiz Rodrigues de Campos, Ph.D.

Prof. Richard Demo Souza, D.Sc.

Prof. Alexandre Sztajnberg, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2015

Henriques, Felipe da Rocha

Contributions to Energy Conservation in Wireless Sensor Networks/Felipe da Rocha Henriques. – Rio de Janeiro: UFRJ/COPPE, 2015.

XVI, 92 p.: il.; 29, 7cm.

Orientadores: Eduardo Antônio Barros da Silva
Lisandro Lovisolo

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2015.

Bibliography: p. 84 – 92.

1. Compressive Sensing. 2. Reconstruction. 3. Wireless Sensor Networks. I. Silva, Eduardo Antônio Barros da *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Dedico esta obra aos meus pais:
Maria Lúcia da Rocha Henriques
e Jorge Luiz de Carvalho
Henriques

Agradecimentos

Gostaria de agradecer em primeiro lugar à Deus, que me iluminou, me deu sabedoria e conduziu meus passos até onde eu jamais imaginei que poderia chegar. Agradeço à minha família que sempre me deu apoio e compreensão nos momentos de dificuldade. Agradeço aos meus amigos, todos que ajudaram nesta importante etapa da minha vida, os mais próximos e os mais distantes. A distância física não é problema quando aqueles que amamos estão dentro do nosso coração.

Agradeço muito aos meus orientadores, professores Eduardo Antônio Barros da Silva e Lisandro Lovisolo. Os senhores são muito mais do que orientadores, sempre acreditaram em mim, me ajudaram muito não só na minha vida acadêmica, são exemplos de professores. Em especial agradeço ao professor (e amigo) Lisandro que me acompanha e orienta desde o mestrado. É uma honra ser o seu primeiro aluno de doutorado. O seu exemplo de dedicação à educação foi o principal fator que me fez decidir por essa carreira belíssima, e trabalhar com você sempre me motiva a querer continuar nela.

Gostaria de agradecer ao professor Marcelo Gonçalves Rubinstein (UERJ) por toda ajuda desde o mestrado. Uma parte importante deste trabalho tem suas contribuições, que sempre foram fundamentais para minha carreira acadêmica.

Por fim, gostaria de agradecer imensamente aos professores Marcello Luiz Rodrigues de Campos, Ph.D; Richard Demo Souza, D.Sc; e Alexandre Sztajnberg, D.Sc., por terem gentilmente aceitado participar da minha banca.

Combati o bom combate,
terminei a minha carreira,
guardei a fé.
2 Tim 4,7

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

CONTRIBUIÇÕES PARA A CONSERVAÇÃO DE ENERGIA EM REDES DE SENSORES SEM FIO

Felipe da Rocha Henriques

Dezembro/2015

Orientadores: Eduardo Antônio Barros da Silva
Lisandro Lovisolo

Programa: Engenharia Elétrica

Uma Rede de Sensores Sem Fio (RSSF) é um tipo de rede ad hoc na qual os seus nós sensores são capazes de coletar dados do ambiente. Cada nó é alimentado por uma bateria, com um tempo de vida limitado. Portanto, um dos desafios para as RSSFs é a conservação de energia dos nós sensores. Neste trabalho, uma RSSF é utilizada para monitorar dados ambientais. Inicialmente, propõe-se DECA (*Distributed Energy Conservation Algorithm*). O algoritmo prediz o valor da medida e o período de inatividade dos nós sensores, colocando-os em um modo de economia de energia. Os sinais são reconstruídos no nó sorvedouro, a partir das amostras recebidas. DECA tem por objetivo garantir que o erro de reconstrução do processo monitorado seja menor do que uma fração do valor atual. Além disso, a técnica chamada *Compressive Sensing* (CS) é explorada com o objetivo de economizar energia dos nós sensores, através da compressão de dados. Assumindo que o sinal a ser monitorado é esparso, ou seja, possui poucos coeficientes não-nulos em alguma base, medidas lineares são tomadas com funções incoerentes com essa base. O sinal monitorado é recuperado a partir de um número de medidas da ordem da esparsidade do sinal. Isto é, o sinal é representado em um espaço de dimensão menor, a uma taxa menor que a *Taxa de Nyquist*. As medidas a serem transmitidas são quantizadas e o comportamento taxa-distorção do sinal reconstruído é avaliado sob diferentes níveis de quantização. Três métodos de reconstrução distintos são analisados. O impacto da perda de pacotes na reconstrução do sinal monitorado também é discutido. Uma maneira eficiente de realizar aproximação sucessiva das medidas é investigada e um esquema de transmissão incremental é proposto.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

CONTRIBUTIONS TO ENERGY CONSERVATION IN WIRELESS SENSOR NETWORKS

Felipe da Rocha Henriques

December/2015

Advisors: Eduardo Antônio Barros da Silva
Lisandro Lovisolo

Department: Electrical Engineering

A Wireless Sensor Network (WSN) is an ad hoc network whose nodes can collect environmental data. Each node is supplied by a battery, with a limited lifetime. Thus, one of the challenges for WSNs is the energy conservation of sensor nodes. In this work, a WSN is employed to monitor environmental data. First of all, we propose DECA (Distributed Energy Conservation Algorithm). The algorithm aims at predicting both the value of the measurement and the inactivity period of sensor nodes in order to put sensor nodes into an energy saving mode. The signals are reconstructed at the sink node, from the samples received from sensor nodes. DECA aims at guaranteeing that the reconstruction error of the monitored process is lower than a given fraction of its actual value. Moreover, the technique called Compressive Sensing (CS) is explored, aiming at saving sensor nodes energy by means of data compression. Assuming that the signal of interest is sparse, i.e., that the signal has few nonzero coefficients in some basis, linear measurements are taken with functions that are incoherent with the sparsifying basis, and the monitored signal is recovered from a number of measurements of the order of its sparsity. Thus, the signal can be represented in a smaller dimension space, at a rate that is lower than the Nyquist rate. The transmitted measurements are quantized with a linear scalar quantizers and the rate-distortion behavior of the reconstructed signal is evaluated for different quantization levels. Three distinct reconstruction methods are analyzed. The impact of packet loss at the reconstruction of the monitored signal is also discussed. It is also investigated how to efficiently perform successive approximation of the CS measurements, and an incremental transmission scheme for that purpose is proposed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent advances in micro-electronics and wireless communications made it possible to develop and deploy low cost, low energy consumption and tiny sensors. These have been used to build Wireless Sensor Networks (WSNs) [1]. A WSN is an ad hoc network and can be applied in several domains [2–6]: i) in medical applications, to remotely monitor patients and their biometric data; ii) for military purposes, to monitor forces; iii) in industrial automation; iv) to sense variables in a region of interest; among others.

Sensor nodes have four basic units: a sensing unit, to sense environmental data like temperature, pressure or humidity; a processing unit; a communication unit, for transmission and reception tasks; and an energy unit, often comprising just a battery. These nodes have an autonomy, operating as long as their batteries have energy [1]. Thus, research on methods leading to energy saving in WSNs have turned into an important issue, in order to increase sensor nodes autonomy.

A survey of energy saving methods for WSNs is presented in [7], including a taxonomy for them. This work deals with energy conservation in WSNs and proposes two distinct schemes that fit with two of the approaches presented by [7] (Figure 1.1). The first one is a novel Distributed Energy Conservation Algorithm (DECA) and uses an asynchronous sleep/wakeup protocol, in which nodes enter in an inactivity state in order to save energy; the other one explores data compression by means of Compressive Sensing (CS) technique, to reduce the amount of data in transmissions.

1.1 Energy Saving in WSNs

According to [1], communication (i.e., transmission and reception) is the task that requires more energy in a WSN. This means that it may be advantageous for a node to process data, in order to compress it or to decide whether to transmit it or not and thus save energy while sleeping [8]. In [9], spatial and temporal correlations between measured samples are used to decrease the amount of transmissions, saving

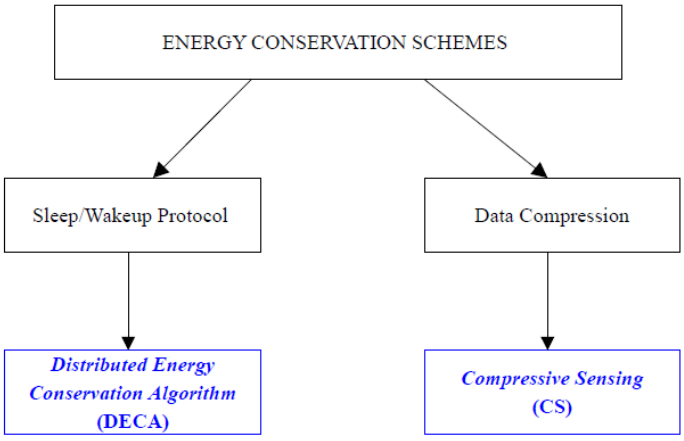

Figure 1.1: Energy conservation schemes proposed in this work.

sensor node energy. The sink predicts the field that is being sensed by the sensor nodes using information that it receives from them. Using the received data, the sink estimates for how long each node may sleep and then sends different messages for the distinct nodes composing the WSN conveying their sleeping periods. The sink is a node that receives all traffic from WSN, and is usually more robust than sensor nodes. This strategy is completely centralized as the sink decides for how long the nodes can sleep, and a given node may not be put to sleep at all, in the case that packets are lost, thus severely impacting energy saving.

Similarly, in [10], the temporal pattern of samples measured by sensor nodes is used to reduce the amount of transmissions to the sink node. The monitored process is compared against its expected behavior. If the measurements match then the node does not transmit its value in order to reduce the amount of transmissions and thus increasing network lifetime. If the measurements do not match the expected behavior then the node transmits the values. Since this scheme does not involve node sleeping, a confinement of its gain in network lifetime is obtained.

In [11], a sleep/wakeup scheme is proposed, in which a network coordinator periodically transmits a beacon frame with a sleeping command. Therefore, this scheme is centralized and nodes enter the sleeping state synchronously upon the reception of this command. To improve energy saving, a power control mechanism in the MAC sub-layer is also proposed, based on the distance between neighboring nodes.

Another sensor-sleeping scheme to save node energy is proposed in [12]. It includes a routing protocol to optimize the number of active paths in the network. In addition, a duty-cycled MAC protocol based on a Markov model [13] is used to

determine the sleeping timers for the proposed routing protocol.

In [4], a "derivative-based-prediction" is employed in each sensor node to manage the need for transmissions. It simply allows each node to verify if the measured quantity did not change enough, in which case it is not transmitted. This approach resembles the one previously presented in [14]. The energy conservation algorithm in [14] aims at reducing the amount of transmissions by managing the need for them. A sample is transmitted only if the percentage variation between it and the last transmitted sample is greater than a given threshold and nodes sleep between transmissions.

The proposals in [4, 9, 10] consider a uniform sampling interval for the sampled measurements. This is similar to the proposals for "signal compression" based on piecewise linear smoothing [15]. However, once a node is put into sleep mode, the sampling is not uniform anymore, this aspect emerged in [14] and is fully considered by the algorithm presented in this work, and previously presented in [16].

Signal reconstruction for WSNs based signal capture using CS is proposed in [17]. The correlations among signals detected by different sensors are explored to further reduce the amount of samples required for reconstruction. Environmental data gathered by a WSN located in the Intel Berkeley Research Lab [18] are considered. In [19], a scheme is proposed in which samples of the signal acquired by sensor nodes are reordered, in order to obtain a more compressible signal. Thus, less transmissions are required, what reflects in sensor node energy savings.

Since the transmission infrastructure used by a WSN is digital, the sensor measurements must be quantized before transmission. However, the analyses performed in [17] and [19] disregarded this aspect, assuming that measurements are densely quantized. Some insight on the quantization of compressive sensed measurements is presented in [20]. The 1-bit Compressive Sensing framework is proposed, which preserves only the sign information of each random measurement. In this scheme, the signal is recovered within a scale factor. This 1–bit CS framework is applied to data gathering in WSNs in [21]. The average distortion introduced by quantization on CS measurements is studied in [22] by comparing two reconstruction algorithms, the standard Basis Pursuit [23] and the Subspace Pursuit [24] CS recovery schemes. In [25], an empirical analysis of the rate–distortion performance of CS is presented, in the context of image compression.

The transmission over a noisy channel and the evaluation of the reconstruction error in the presence of packet erasures, while saving sensor nodes energy, are considered in [26]. Two approaches for CS are studied: a cluster–based and a consensus–based scheme. A basis pursuit denoising (BPDN) algorithm is used for reconstruction by the sink node.

In order to save sensor nodes energy in a surveillance application using a Visual

Sensor Network (VSN), a routing framework called PRoFIT is proposed in [27]. In this VSN application scenario, sensing nodes divide images into bit–planes [28]. There are two layers that are created for captured images. The first one contains the most significant bit–planes (transmitted with higher priority) and the second layer has the least significant bit–planes (transmitted with lower priority). Firstly, from the first layer, the image is reconstructed at the sink with a certain degree of detail, so that some immediate action can be taken based on the image content. If a more detailed reconstruction is required, the sink uses the second layer with remaining bit–planes. This strategy can be understood as a successive approximation scheme [29] [30], in which each bit–plane is incrementally transmitted, in order to provide a better reconstruction.

1.2 Objectives and Contributions

In this work we propose two distinct schemes that aim at saving sensor nodes energy, in a WSN monitoring application:

- Distributed Energy Conservation Algorithm (DECA) [31]; and
- Compressive Sensing (CS) [32].

Specific objectives and contributions of both proposed schemes are briefly discussed hereafter.

1.2.1 Objectives and Contributions: DECA for WSNs

The first scheme proposed in this work is a Distributed Energy Conservation Algorithm (DECA) for Wireless Sensor Networks in monitoring applications. Roughly, it intends to predict a future measured value and this predicted behavior is used to define the inactivity period IP_i of a given node S_i. Each sensor node sleeps during its inactivity period, and then the node wakes up, measures the desired quantity and transmits it, reiterating the process. Energy saving and the resulting increase in network lifetime are achieved by both reducing the amount of transmissions and also by putting nodes to sleep during their inactivity periods. In this sense the proposed algorithm works by adapting sensor nodes duty-cycle [33], but taking into account data to define the sleeping/inactivity period. Algorithms that work in the MAC sub-layer aiming at energy saving by powering-off sensor transceivers abound [34–38]. Differently, DECA tries to follow the process computing the nodes sleeping/inactivity periods from the process evolution with time. This falls in the methods that the authors of [39] call as "data-aware techniques" as the methods, techniques and algorithms in [16, 39–44].

The decision whether or not to sleep is performed by each node individually, i.e., locally. That is, the nodes do not depend on the reception of sleeping commands or data from the sink. Therefore, when making this decision, two aspects need to be considered. The first is that for computing IP_i, each node S_i must consider how the sink reconstructs the process from the measurements the sink receives. As a consequence, in order to guarantee a trusty reconstruction of the process, we impose the constraint of keeping the reconstruction error within an acceptable distortion criterion at the sink and not solely at the sensor. Therefore, since each node makes its own decision, the proposed strategy is decentralized and distributed.

An important issue to be considered when computing the inactivity/sleeping period of a sensor is: how does putting a given node to sleep impact the network connectivity? Therefore, the second aspect driving DECA design refers to multi-hop networks, where nodes can act both as sources (when they measure samples of the monitored process) and as routers (when, besides measuring, they also forward packets from neighbors). That is, the inactivity period of a sensor node must be at least considered by a sensor that is routing its messages. Inversely, when deciding its inactivity period, each node must account for the impact of this decision on network connectivity, that is, on other nodes. This is considered in DECA; nodes forward their own inactivity periods together with the measurements. Moreover, a sensor node considers the inactivity periods of its first-hop neighbors (that uses this sensor node as a router) to compute its own inactivity period.

Therefore, although running in the application layer and the decision of how long to sleep being taken autonomously by each sensor, network topology and routes awareness is inherent to DECA since the inactivity periods of sensors that are routed by a node are considered by the node when deciding for how long to sleep. In addition, a fringe benefit of forwarding inactivity periods is that sensor nodes are not required to be synchronized, since the algorithm does not demand an absolute time base, employing just relative times (time intervals).

We should highlight that DECA differs from existing proposals in the literature as: i) DECA is applied in the application layer; ii) DECA does not impose an uniform sampling interval for the measurements time-series; iii) by design, DECA imposes the reconstruction error of the sensed variable to be within an acceptable distortion criterion; iv) DECA is decentralized and distributed, since each and every node decides by itself for how long to sleep, although considering information about the nodes it forwards packets for; and v) DECA uses a deterministic approach for computing the node sleeping/inactivity period.

Since nodes are put to sleep during their estimated inactivity periods, in order to evaluate the impact of this in the network connectivity, we propose a metric that we call success ratio, that measures the amount of data packets that sensor nodes

can forward from their neighbors.

We run simulations with real environmental data, and results show a significant increase in the network lifetime by using DECA, as compared to WSNs without any energy management scheme. We have also verified that network connectivity is not impaired by the algorithm, what, indeed, makes it worth to use DECA on any monitoring application alike the one considered by DECA.

1.2.2 Objectives and Contributions: CS coded measurements in WSNs

The second energy conservation technique considered in this work explores another paradigm, being based on Compressive Sensing. We consider a quantized CS measurements framework, aiming at reducing the amount of transmissions in a WSN that is monitoring environmental data, as in the case also considered for DECA. In doing so, one expects that network nodes save energy, thus increasing WSN autonomy.

As the transmitted measurements are quantized, the rate–distortion behavior of the reconstruction of monitored signals is evaluated. We consider different environmental data: temperature, humidity and illumination, gathered by the WSN Research Lab [18]. For reconstruction, we investigate the performance of three distinct CS reconstruction schemes. We compare the rate–distortion performance of the Newton with log–barrier, A*OMP and LASSO methods for several number of CS measurements and bit–depths. We also evaluate the impact of packet losses in their rate–distortion performance.

Finally, we propose a scheme to incrementally transmit CS measurements in order to provide quantization refinement in the decoder, resulting in a successive approximation CS measurements. In other words, for a desired increment in rate and starting from any given point on its rate–distortion operational curve, a sensor node just needs to transmit extra CS measurements and/or quantization refinement bits for the measurements. For the proposed scheme, we aim at reducing the amount of transmissions (bits, packets and bursts) carried out by each sensor node, while keeping the rate–distortion performance sufficiently close to the optimal.

1.3 Organization

This work is structured as follows. In Chapter 2, we present a model for WSN based monitoring applications. This model is used to size-up the impacts of putting nodes to sleep in WSNs, and in continuation employed to derive the proposed algorithm for saving energy in monitoring applications (DECA). Furthermore, we describe the

DECA algorithm implementation. In Chapter 3, we present the sensor node energy model used in this work, the simulation aspects and results obtained with simulations considering environmental data for DECA. Chapter 4 presents Compressive Sensing fundamentals and the reconstruction methods investigated in this work. In Chapter 4, we also analyze the rate–distortion performance of the reconstruction of environmental data with three distinct recovery strategies for CS measurements and perform a comparison between CS and DECA. Chapter 5 presents the impact of packet losses in the rate–distortion performance of the reconstructed signals. The proposed scheme for incremental transmissions (successive approximation) in the proposed CS–quantized framework is presented and evaluated in Chapter 6. At last, conclusions and future directions are discussed in Chapter 7.

Chapter 2

Distributed Energy Conservation Algorithm for WSNs–DECA

2.1 Introduction

In this work, we consider that a WSN is employed to sense a physical variable, a field – a process whose value depends on space coordinates (x, y, z) and on time t. Each sensor S_i samples the monitored process at its position (x_i, y_i, z_i) measuring $s(x_i, y_i, z_i, t)$, and, eventually, transmits the measurement to a sink node (for sake of simplicity in the reminder we ignore the coordinate z and use the notation $s_i(t) = s(x_i, y_i, t)$). In a WSN, the sink is usually more robust than the sensor nodes, and it can be used as a gateway [1].

The problem that we consider is: how can we make an energy-efficient usage of the WSN while providing an acceptable reconstruction of the sensed field for the monitoring application? As acceptable reconstruction, we assume that the reconstruction error is kept below a predefined value, while energy efficiency involves improving the network autonomy, by increasing its lifetime. More specifically, we assume that the network lifetime is the time until the first node dies, i.e., when its energy ends [45].

Several researchers deal with the problem of energy management in sensor networks [46, 47]. We consider two main aspects as design premises that differ from their proposals: 1) we propose an algorithm in the application layer such that off-the-shelf components can be used to build the network. Thus, the proposed algorithm can be applied to the sensor nodes regardless of other methods used in lower layers, as MAC (Medium Access Control) and routing protocols; and 2) we aim at keeping the reconstruction error of the sensed variable within an acceptable distortion criterion. We achieve energy saving by putting nodes to sleep between successive signal sampling and its transmissions. A sleeping node is not capable to sense, to process,

to receive neither to transmit until it wakes up.

Figure 2.1 depicts the reasoning behind the proposed algorithm. The sensor S_i, at an instant $t_i[n]$, measures the quantity $s_i[n] = \mathrm{S}_i(t_i[n])$. Then, based on the past of the quantity (known), the node sleeps for a period IP_i (the Inactivity Period of node S_i). Once this time has passed, the node awakes at $t_i[n+1]$ for sensing, processing and transmitting the data, and then computes a new inactivity period for switching to sleep mode again.

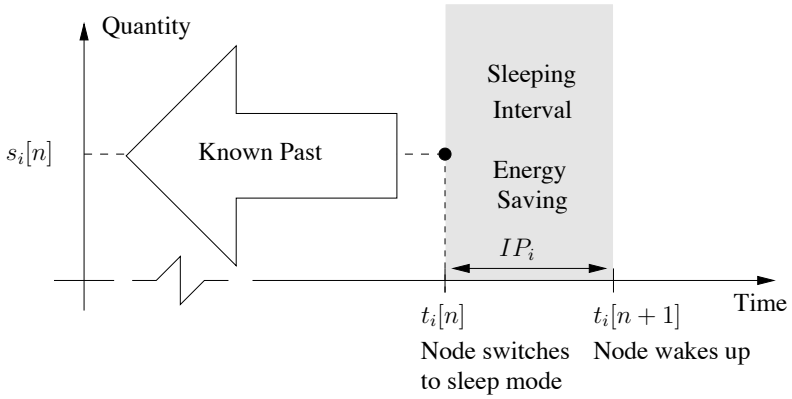

Figure 2.1: Idea developed in DECA.

We consider that each node in the WSN should decide whether to sleep or not by itself, i.e., in a distributed fashion. That is, we consider that the decision to sleep or not to sleep should not be bounded or controlled by any central node, i.e., it should be decentralized. Nevertheless, a node may route messages from other nodes to the sink. Therefore, each and every node must consider information collected by neighboring nodes for deciding to sleep and for how long.

2.2 Sensing and Reconstructing a Process with a WSN

Figure 2.2 summarizes the problem of using a WSN to sense a field. Sensor nodes are spread in a region of interest and measure a collection of samples $\mathbf{s}_i^{(M)}$ of a physical quantity – i indicates the sensor and (M) is used to indicate that these values are in principle known only by the measurement node. As sensor nodes use an algorithm to save energy by managing the necessity of communication, a subset of their measurements is transmitted while other may not be. We denote the set of transmitted samples as $\mathbf{s}_i^{(T)}$. The sink node receives the set $\mathbf{s}_i^{(R)}$, since a wireless channel is used, some packets may be lost, and therefore $\mathbf{s}_i^{(R)} \subset \mathbf{s}_i^{(T)}$. The sink reconstructs the monitored field using only the received samples, and generates an estimate $\hat{s}$.

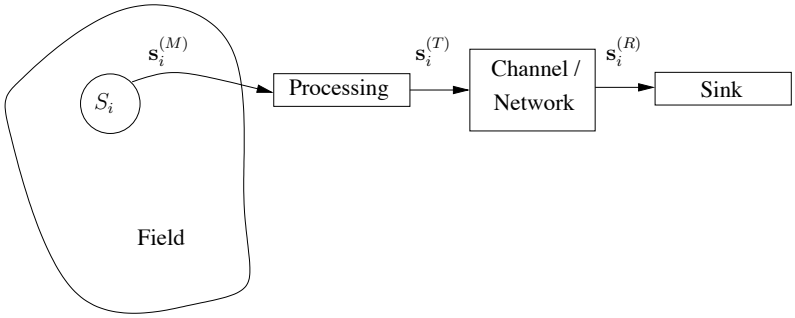

Figure 2.2: Framework considered in this work.

It is worth emphasizing that:

a) A measured sample $s_i^{(M)}$ may be transmitted or not, i.e., it may generate a transmitted sample $s_i^{(T)}$ or not, therefore $\mathbf{s}_i^{(T)} \subset \mathbf{s}_i^{(M)}$;

b) A transmitted sample $s_i^{(T)}$ may be received or not, i.e., it may generate a received sample $s_i^{(R)}$ or not, therefore $\mathbf{s}_i^{(R)} \subset \mathbf{s}_i^{(T)}$.

In the previous discussion, time has been omitted for simplicity. Hereafter, we describe its inclusion for WSN based monitoring application.

2.2.1 Measurements Flow

Consider a sensor node S_i and a sink node S_r. The measurements collected by S_i can be organized as a vector

$$\mathbf{s}_i^{(M)} = \left[s_i^{(M)}\left(t_i^{(M)}[1]\right), s_i^{(M)}\left(t_i^{(M)}[2]\right), \ldots, s_i^{(M)}\left(t_i^{(M)}[N]\right) \right], \tag{2.1}$$

where the $t_i^{(M)}[n]$, $n \in [1 \ldots N]$ are the measurement/sampling times and are such that $t_i(M)[j] < t_i(M)[k]$ if $j < k$. For simplicity of notation we define

$$\mathbf{s}_i^{(M)} = \left[s_i^{(M)}[1], s_i^{(M)}[2], \ldots, s_i^{(M)}[N] \right], \text{and } \mathbf{t}_i^{(M)} = \left[t_i^{(M)}[1], t_i^{(M)}[2], \ldots, t_i^{(M)}[N] \right] \tag{2.2}$$

for the vector composed of the measuring instants of the elements of $\mathbf{s}_i^{(M)}$.

At the sink node, the reconstruction of the process is done from the available information, which is in principle a subset of the measured samples. Since $\mathbf{s}_i^{(R)} \subset \mathbf{s}_i^{(T)} \subset \mathbf{s}_i^{(M)}$, reconstruction employs the received samples $\mathbf{s}_i^{(R)}$ and the corresponding times $\mathbf{t}_i^{(R)}$, i.e.,

$$\mathbf{s}_i^{(R)} = \left[s_i^{(R)}[1], s_i^{(R)}[2], \ldots, s_i^{(R)}[K] \right] \text{ and } \mathbf{t}_i^{(R)} = \left[t_i^{(R)}[1], t_i^{(R)}[2], \ldots, t_i^{(R)}[K] \right]. \tag{2.3}$$

The values of the elements of $\mathbf{t}_i^{(R)}$ are the measurement times of the samples that are transmitted by the sensors, that are actually received by the sink. At any given time of the network life the cardinality of $\mathbf{s}_i^{(R)}$ is smaller than or equal to the cardinality of $\mathbf{s}_i^{(M)}$, that is $K \leq N$. It should be noted that $\mathbf{t}_i^{(R)}$ is a subset of elements of $\mathbf{t}_i^{(M)}$ and $\mathbf{s}_i^{(R)}$ is a subset of elements of $\mathbf{s}_i^{(M)}$. However, a priori, there is no simple mapping rule from the time index n in $s_i^{(M)}[n]$ to the index k in $s_i^{(R)}[k]$. This follows as a direct consequence of the items a) and b) above.

2.2.2 Reconstruction of the Process

At the sink node S_r, we aim at obtaining a reconstruction $\hat{s}\,(x, y, t)$ of the process $s\,(x, y, t)$ by using the measurements received from the sensor nodes. Moreover, we want this reconstruction to be within an acceptable error criterion.

For reconstructing $\hat{s}\,(x, y, t)$, a first order predictor from the samples received at the sink node is considered. This kind of predictor leads to smaller reconstruction errors as compared to a zero order predictor, such as the one used in [14]. The reconstruction error is defined as

$$e\,(\mathrm{S}_i, t) = e\,(x_i, y_i, t) = s_i\,(t) - \hat{s}_i\,(t)\,. \tag{2.4}$$

2.3 Problem Model

In this section, we discuss a general model for the estimation and the reconstruction of the physical variable monitored by the WSN. The sink node S_r knows just the set of received samples from node S_i, $\mathbf{s}_i^{(R)}$. It is important to emphasize that $\mathbf{s}_i^{(T)}$ is a subset of $\mathbf{s}_i^{(M)}$ since nodes do not transmit all measured samples, in order to save energy. In addition, since some of the transmitted packets may even be lost or with errors, $\mathbf{s}_i^{(R)}$ is a subset of $\mathbf{s}_i^{(T)}$. For the moment, we will assume that $\mathbf{s}_i^{(R)}$ and $\mathbf{s}_i^{(T)}$ are equal, but the impact of packet loss in the proposed algorithm will be discussed in Section 3.3.

At a given instant τ, we can consider that a set of past measurements

$$\tilde{\mathbf{s}}_i^{(M)}(\tau) = \left\{ s_i^{(M)}(t_i^{(M)}[j]) \right\}_{t_i^{(M)}[j]<\tau} = \left\{ s_i^{(M)}[j] \right\}_{t_i^{(M)}[j]<\tau} \tag{2.5}$$

is known. From the known measurements, we can estimate / predict $s_i^{(M)}(\tau)$ (guess its value) at instant τ using a predictor

$$\hat{s}_i^{(M)}(\tau) = \mathcal{P}\left\{ \tilde{\mathbf{s}}_i^{(M)}(\tau) \right\}. \tag{2.6}$$

Above, we do not explicitly describe the prediction rule $\mathcal{P}\{\cdot\}$, but rather the in-

formation it depends on. In addition, we are not assuming any specific prediction or model, just that it should be causal, i.e., it uses just past known samples. $\mathcal{P}\{\cdot\}$ may incorporate the instants of sample measurements to improve its performance. Defining

$$\tilde{\mathbf{t}}_i^{(M)}(\tau) = \left\{ t_i^{(M)}[j] \right\}_{t_i^{(M)}[j]<\tau}, \tag{2.7}$$

we have

$$\hat{s}_i^{(M)}(\tau) = \mathcal{P}\left\{ \tilde{\mathbf{s}}_i^{(M)}(\tau),\ \tilde{\mathbf{t}}_i^{(M)}(\tau) \right\}. \tag{2.8}$$

The same strategy can be applied at the sink to reconstruct the process from the received samples. This provides

$$\hat{s}_i^{(R)}(\tau) = \mathcal{P}\left\{ \tilde{\mathbf{s}}_i^{(R)}(\tau),\ \tilde{\mathbf{t}}_i^{(R)}(\tau) \right\}. \tag{2.9}$$

It is important to note that since not every measurement is transmitted the sets $\tilde{\mathbf{s}}_i^{(R)}(\tau)$ and $\tilde{\mathbf{t}}_i^{(R)}(\tau)$ are defined similarly to eqs. (2.5) and (2.7), but they are not the same.

2.3.1 Error Criterion

Let $s_i(t) = s(x_i, y_i, t)$ be the value collected by sensor S_i placed at (x_i, y_i) over time and its reconstruction version be $\hat{s}_i(t) = \hat{s}(x_i, y_i, t)$. The reconstruction error is defined as eq. (2.4). We assume that the ratio between the predicted sample and its actual value must be smaller than a fraction δ of the actual value. For the predicted measurement value at instant t, $\hat{s}_i^{(M)}(t)$, and the reconstructed one (done at the sink), $\hat{s}_i^{(R)}(t)$, the above error criteria are

$$\left| \hat{s}_i^{(M)}(t) - s_i^{(M)}(t) \right| \leq \delta \left| s_i^{(M)}(t) \right| \quad \text{and} \quad \left| \hat{s}_i^{(R)}(t) - s_i^{(R)}(t) \right| \leq \delta \left| s_i^{(R)}(t) \right|. \tag{2.10}$$

Using eqs. (2.8) and (2.9), the errors of the estimated field in the sensor node and at the sink at a given instant t are given by

$$e\left(s_i^{(M)},\ t\right) = \left| \mathcal{P}\left\{ \tilde{\mathbf{s}}_i^{(M)}(t),\ \tilde{\mathbf{t}}_i^{(M)}(t) \right\} - s_i^{(M)}(t) \right|, \text{and} \tag{2.11}$$

$$e\left(s_i^{(R)},\ t\right) = \left| \mathcal{P}\left\{ \tilde{\mathbf{s}}_i^{(R)}(t),\ \tilde{\mathbf{t}}_i^{(R)}(t) \right\} - s_i^{(R)}(t) \right|. \tag{2.12}$$

From the accepted error criterion in eq. (2.10), these should be bounded respectively by

$$e\left(s_i^{(M)},\ t\right) \leq \delta \left| s_i^{(M)}(t) \right| \text{ and } e\left(s_i^{(R)},\ t\right) \leq \delta \left| s_i^{(R)}(t) \right|. \tag{2.13}$$

Sensor nodes may estimate a future measurement $\hat{s}_i^{(M)}[n] = \hat{s}_i^{(M)}\left(t_i^{(M)}[n]\right)$ from some known samples and their measurement times. Similarly, we may use the time

interval between $\hat{s}_i^{(M)}[n]$ and $s_i^{(M)}[n-1]$, given by $\hat{t}_i^{(M)}[n]-t_i^{(M)}[n-1]$, for the node to sleep and save energy. Putting this altogether, using the restriction $e\left(s_i^{(M)},\ t\right) \leq \delta|s_i^{(M)}(t)|$ we can estimate the inactivity period / sleeping time of node i. This resembles the approach on "signal compression" in [15]. However, our approach does not imposes nor assumes a uniform sampling ($t_i[n] = n\Delta T$).

Nonetheless, the problem acquires a new complexity when we consider that the reconstruction at the sink should closely match the samples taken from the actual process $s_i(t)$ instead of the ones that are known at the sink. Therefore, we now consider this aspect when a first order prediction rule is employed in the sensor to predict future values and in the sink to reconstruct the process.

2.3.2 First Order Prediction

Let us consider that a first order model for $\mathcal{P}\{\cdot\}$ is employed. That is, the last two samples are used to estimate a future value and we employ the following notation: for the last two measured samples and instants of measurements, we have $s_i^{(M)}[n-1]$ and $s_i^{(M)}[n-2]$, $t_i^{(M)}[n-1]$ and $t_i^{(M)}[n-2]$. Using these definitions, the predicted value (see Figure 2.3) is given by

$$\begin{aligned}\hat{s}_i^{(M)}[n] &= \mathcal{P}\left\{\tilde{\mathbf{s}}_i^{(M)}(\hat{t}_i(M)[n]),\ \tilde{\mathbf{t}}_i^{(M)}(\hat{t}_i(M)[n])\right\} = \\ &= s_i^{(M)}[n-1] + \frac{s_i^{(M)}[n-1]-s_i^{(M)}[n-2]}{t_i^{(M)}[n-1]-t_i^{(M)}[n-2]}\left(\hat{t}_i^{(M)}[n]-t_i^{(M)}[n-1]\right).\end{aligned} \tag{2.14}$$

Above, we use $\hat{t}_i^{(M)}[n]$ to denote that this time has not passed yet, but is instead estimated. A similar equation applies for the reconstruction at the sink, $\hat{s}_i^{(R)}[k]$, if we assume that all transmissions are successful, i.e., $\mathbf{s}_i^{(R)} \equiv \mathbf{s}_i^{(T)}[k]$ (although this may not be always true, as some transmissions may not be successful, we evaluate the impact of breaching this assumption in Section 3.3).

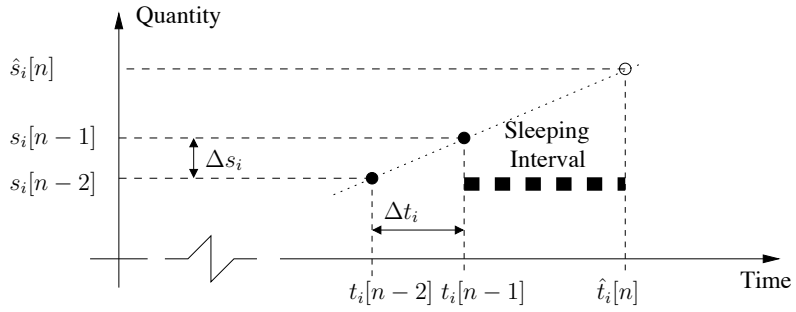

Figure 2.3: First order prediction.

We can define

$$\Delta s_i^{(M)}[n-1] = s_i^{(M)}[n-1] - s_i^{(M)}[n-2], \tag{2.15}$$
$$\Delta t_i^{(M)}[n-1] = t_i^{(M)}[n-1] - t_i^{(M)}[n-2], \tag{2.16}$$
$$\Delta s_i^{(R)}[k-1] = s_i^{(R)}[k-1] - s_i^{(R)}[k-2], \text{ and} \tag{2.17}$$
$$\Delta t_i^{(R)}[k-1] = t_i^{(R)}[k-1] - t_i^{(R)}[k-2] \tag{2.18}$$

which correspond to the differences between the two last measured samples, between the sample times of the two last measured samples, between the two last transmitted samples and between the sample times of the two last transmitted samples, respectively. From these, we can define the variation rates

$$\alpha_i^{(M)}[n-1] = \frac{\Delta s_i^{(M)}[n-1]}{\Delta t_i^{(M)}[n-1]}, \text{ and } \alpha_i^{(R)}[k-1] = \frac{\Delta s_i^{(R)}[k-1]}{\Delta t_i^{(R)}[k-1]}. \tag{2.19}$$

Using the above notations, the first order prediction rules become

$$\hat{s}_i^{(M)}[n] = s_i^{(M)}[n-1] + \alpha_i^{(M)}[n-1]\left(\hat{t}_i^{(M)}[n] - t_i^{(M)}[n-1]\right), \text{ and} \tag{2.20}$$
$$\hat{s}_i^{(R)}[k] = s_i^{(R)}[k-1] + \alpha_i^{(R)}[k-1]\left(\hat{t}_i^{(R)}[k] - t_i^{(R)}[k-1]\right). \tag{2.21}$$

A future sample is guessed using linear interpolation which requires very low computational complexity and memory requirements, welcome aspects for algorithms running on sensors. As depicted in Figure 2.3, this rule could be used to estimate a time interval for the sensor node to sleep. However, as previously stated, we want to impose an additional constrain that needs to be considered: we want to control / limit the reconstruction error at the sink. This leads to the requirement of modeling the influence of the quantities known at the sink on the sensor decision to sleep or not. That is, the sensor (local) should evaluate its impacts on the sink reconstruction (global). The next subsection evaluates the influence of the quantities known at the sink on the sensor behavior and, following, we present how to use this to estimate the sleeping period at sensor nodes.

2.3.3 How does Information Cognized at the Sink Influence Sensor Nodes?

All information that is available at the sink about sensor measurements is known by the sensor itself. On the other hand, the contrary is hardly true, i.e., what the sensor knows about its measurements is not necessarily known by the sink. Let us resort the indices of the last two transmitted (received) samples and their instants of transmissions with respect to the sample indices at the sensor. In doing so,

we have $s_i^{(R)}[n-j]$ (a measurement corresponding to j samples before the current measurement sample) and $s_i^{(R)}[n-j-l]$ (corresponding to $j+l$ samples before the current measurement sample), and their corresponding measurement times $t_i^{(R)}[n-j]$ and $t_i^{(R)}[n-j-l]$. That is, we define the sample indices $n-j$ and $n-j-l$, since not all measured samples are transmitted nor received at the sink, and j and $j+l$ are used to express the indices time lags. Using this notation, the superscripts for measured ((M)) and transmitted ((T)) samples in the previous can be eliminated, referencing all the samples and over all sensing time instants with respect to the sensor measurement process / clock. From the discussed indexing strategy one derives for eqs. (2.15)–(2.18)

$$\Delta s_i^{(M)}[n-1] = s_i[n-1] - s_i[n-2], \tag{2.22}$$

$$\Delta t_i^{(M)}[n-1] = t_i[n-1] - t_i[n-2], \tag{2.23}$$

$$\Delta s_i^{(R)}[n-1] = s_i[n-j] - s_i[n-j-l], \text{and} \tag{2.24}$$

$$\Delta t_i^{(R)}[n-1] = t_i[n-j] - t_i[n-j-l]. \tag{2.25}$$

From the above differences definition, one obtains

$$\alpha_i^{(M)}[n-1] = \frac{s_i[n-1] - s_i[n-2]}{t_i[n-1] - t_i[n-2]}, \text{ and } \alpha_i^{(R)}[n-1] \ = \frac{s_i[n-j] - s_i[n-j-l]}{t_i[n-j] - t_i[n-j-l]}. \tag{2.26}$$

Therefore, we have that (2.20) and (2.21) can be expressed as

$$\hat{s}_i^{(R)}[n] = s_i[n-j] + \alpha_i^{(R)}[n-1](\hat{t}_i[n] - t_i[n-j]), \text{and} \tag{2.27}$$

$$\hat{s}_i^{(M)}[n] = s_i[n-1] + \alpha_i^{(M)}[n-1](\hat{t}_i[n] - t_i[n-1]). \tag{2.28}$$

Equation (2.27) provides an estimate of the next measurement to be received by the sink node $\hat{s}_i^{(R)}[n]$, which would be transmitted by sensor node S_i, from the last received measurements $s_i^{(R)}[n-j]$ and $s_i^{(R)}[n-j-l]$. Meanwhile, eq. (2.28) provides an estimate of the next sample to be collected $\hat{s}_i^{(M)}[n]$, obtained from the last ones that were measured, $s_i^{(M)}[n-1]$ and $s_i^{(M)}[n-2]$.

It is worth mentioning that for $\hat{s}_i^{(M)}[n]$ only data available at the sensor are used. That is, eq. (2.28) is applicable only at the sensor. Therefore, it describes the local behavior of the first order predictor, i.e., at the sensor. On the other hand, the data employed for $\hat{s}_i^{(R)}[n]$ is expected to be available both at the sensor and at the sink for computing the estimate at the sink at the time instant $\hat{t}_i[n]$. We emphasize again that we assume that all transmitted data arrive at the sink. In this sense, eq. (2.27) is the prediction of the reconstruction that is made at the sink computed at the sensor node. This estimates are illustrated in Figure 2.4.

Ideally, the next reconstructed sample should be equal to the next sample to be measured. Since this will rarely occur, we try to keep the error within a fractional error criterion. Therefore,

$$\left|\hat{s}_i^{(R)}[n] - \hat{s}_i^{(M)}[n]\right| \leq \delta \left|\hat{s}_i^{(M)}[n]\right|. \tag{2.29}$$

The error in the reconstructed value at the sink is bounded by the accepted error criterion with respect to the prediction of the next measurement.

Considering that measurements are always positive (without any loss of generality), from eq. (2.29), two situations arise:
I) If $\hat{s}_i^{(R)}[n] > \hat{s}_i^{(M)}[n]$ then we have

$$\hat{s}_i^{(R)}[n] - \hat{s}_i^{(M)}[n] \leq \delta \hat{s}_i^{(M)}[n] \;\Rightarrow\; \hat{s}_i^{(R)}[n] \leq \hat{s}_i^{(M)}[n](1+\delta). \tag{2.30}$$

II) Else (if $\hat{s}_i^{(R)}[n] \leq \hat{s}_i^{(M)}[n]$)

$$-\hat{s}_i^{(R)}[n] + \hat{s}_i^{(M)}[n] \leq \delta \hat{s}_i^{(M)}[n] \;\Rightarrow\; \hat{s}_i^{(R)}[n] \geq \hat{s}_i^{(M)}[n](1-\delta). \tag{2.31}$$

These can be combined into

$$1 - \delta \leq \frac{\hat{s}_i^{(R)}[n]}{\hat{s}_i^{(M)}[n]} \leq 1 + \delta. \tag{2.32}$$

Therefore, two situations can be verified, that bound the ratio between the process estimated at the sink for reconstruction and at the sensor for the future measured value. Since δ is the value that defines the error bounds and may be application dependent, it can be used to define for how long sensors can be put in sleeping mode.

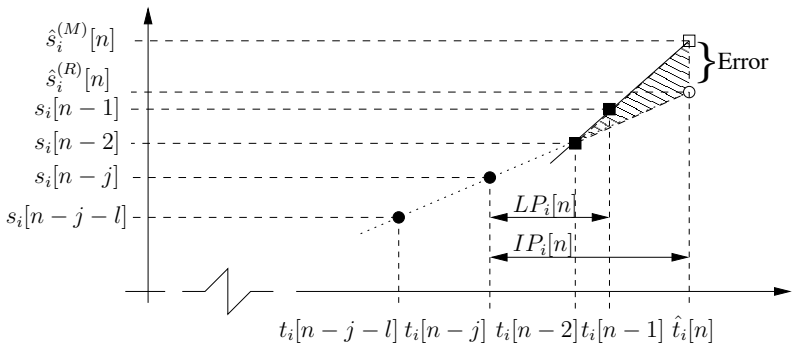

Figure 2.4: Illustration of the computed estimates and inactivity period.

2.3.4 Inactivity Period Computation

We define the Inactivity Period IP_i as the time interval between the last transmitted sample and the future one; and LP_i as the time interval between the last transmitted sample and the last measured sample.

$$\mathrm{IP}_i[n] = \hat{t}_i[n] - t_i[n-j], \text{and} \tag{2.33}$$

$$\mathrm{LP}_i[n] = t_i[n-1] - t_i[n-j]. \tag{2.34}$$

It is worth mentioning that all the data used in these definitions are known at the sensor but not at the sink. These data are illustrated in Figure 2.4. We use $\mathrm{IP}_i[n]$ and $\mathrm{LP}_i[n]$ in the sensor node to estimate for how long the sensor can switch into the sleeping mode and then wake up and measure a new sample. This allows to save energy and augment sensor autonomy.

Using $\mathrm{IP}_i[n]$ and $\mathrm{LP}_i[n]$, eqs. (2.27) and (2.28) become

$$\hat{s}_i^{(R)}[n] = s_i[n-j] + \alpha_i^{(R)}[n-1]\mathrm{IP}_i[n], \text{and} \tag{2.35}$$

$$\hat{s}_i^{(M)}[n] = s_i[n-1] + \alpha_i^{(M)}[n-1](\mathrm{IP}_i[n] - \mathrm{LP}_i[n]). \tag{2.36}$$

Replacing eqs. (2.35) and (2.36) in eq. (2.32), we obtain

$$1-\delta \leq \frac{s_i[n-j] + \alpha_i^{(R)}[n-1]\mathrm{IP}_i[n]}{s_i[n-1] + \alpha_i^{(M)}[n-1](\mathrm{IP}_i[n] - \mathrm{LP}_i[n])} \leq 1+\delta. \tag{2.37}$$

Taking each side separately, we have that

$$\mathrm{IP}_i[n] \leq \frac{s_i[n-j] - s_i[n-1](1-\delta) + \mathrm{LP}_i[n]\left[\alpha_i^{(M)}[n-1](1-\delta)\right]}{\alpha_i^{(M)}[n-1](1-\delta) - \alpha_i^{(R)}[n-1]}, \text{and} \tag{2.38}$$

$$\mathrm{IP}_i[n] \geq \frac{s_i[n-j] - s_i[n-1](1+\delta) + \mathrm{LP}_i[n]\left[\alpha_i^{(M)}[n-1](1+\delta)\right]}{\alpha_i^{(M)}[n-1](1+\delta) - \alpha_i^{(R)}[n-1]}. \tag{2.39}$$

Equations (2.38) and (2.39) show that the acceptable range for the inactivity period for a given sensor node depends on: i) the relationship of the variation rates at the sink $\alpha_i^{(R)}[n-1]$ (data transmitted by the sensor to the sink) and at the sensor $\alpha_i^{(M)}[n-1]$ (computed from data that are available at the sensor); ii) the accepted maximum error (distortion) δ; and iii) the time interval between the last transmitted sample and the last measured sample, $\mathrm{LP}_i[n]$.

The bounds for the inactivity period of a given sensor node $\mathrm{IP}_i[n]$ provided by eqs. (2.38) and (2.39) keep the reconstruction error at the sink within an acceptable error. From the computed bound, the most restrictive case for $\mathrm{IP}_i[n]$, i.e., the smallest

value, can be chosen. As the sensor knows everything that it has transmitted to the sink, one can apply the sink reconstruction at the sensor to compute the acceptable range for the sensor inactivity period with respect to the obtainable reconstruction error. This interval can be computed by each sensor with information generated and available just at the sensor.

In this subsection, we presented a strategy to compute the inactivity period by comparing the prediction at the sink to the actual behavior of the signal measured by the sensor. This was done using linear predictors/interpolators, these allow to easily find bounds on the sleeping periods for each node, that is they allow to evaluate for how long the sensor can be put into sleep mode. It is worth noticing that other more sophisticated interpolators as splines or polynomial ones could also be evaluated. We use the analysis discussed so far to propose DECA hereafter.

2.4 Distributed Energy Conservation Algorithm for Wireless Sensor Networks – DECA

DECA works directly in the application layer of each sensor node, regardless, for example, of the routing protocol used. The steps of the algorithm are showed in **Algorithm 1**. In DECA, n represents the current instant of the sensor. The algorithm is presented for a node S_i and considers the sensor energy $\mathcal{E}_i$ and that the sensor runs until its energy ends. That is, at each step of the algorithm $\mathcal{E}_i$ is decreased accordingly to the task that is performed (see Section 3.2.1). The inactivity period is initially set to $T_{\text{granularity}}$ seconds and it is updated by DECA at each sensor. After that, the sensing-processing-transmission-sleeping procedure occurs (see lines 4–13).

The sleeping period reduction factor γ ($0 < \gamma \leq 1$) is used to increase the probability of S_i to be awake to forward packets from its neighbors. Experiments evaluating the impact of γ on DECA are presented in Section 3.3.2. More details on how DECA considers neighbors (when the sensor has to route data from other sensors) and the reasons for the use of γ are presented in the following.

In this work, we do not consider the synchronization among the clocks of sensor nodes. In order to synchronize their clocks, the nodes must exchange messages, spending energy with this communication task, and one of the main objectives of DECA is to reduce the amount of transmissions in the network. Thus, the γ factor is also used statistically to compensate for the lack of synchronization of the clocks of sensor nodes.

ALGORITHM 1: DECA running at node S_i.

1 $n \leftarrow 1$;
2 $\mathrm{IP}_i[n] \leftarrow T_{\text{granularity}}$;
3 **while** $\mathcal{E}_i > 0$ **do**
4 measure $s_i[n]$ at $t_i[n]$ ($s_i[n] = \mathrm{S}_i(t_i[n])$);
5 **if** $n > 1$ **then**
6 $\mathrm{IP}_i[n] \leftarrow$ minimum between eq. (2.38) and eq. (2.39);
7 **if** *there are packets to forward from neighbors* **then**
8 forward the data from the neighbors;
9 $\mathrm{IP}_i[n] \leftarrow \min\left(\mathrm{IP}_i[n], \{\mathrm{IP}_k[n]\}_{k \in \#\mathrm{SN}_i}\right)$;
10 **end**
11 **end**
12 transmit $s_i[n]$ and $t_i[n]$ together with $\mathrm{IP}_i[n]$;
13 sleep for $\gamma \mathrm{IP}_i[n]$ seconds;
14 $n \leftarrow n + 1$;
15 **end**

2.4.1 Neighbors, Routing and their Inclusion in DECA

A given node may act as router for others. This is exemplified in Figure 2.5. One observes from this example that the sleeping period of node S_i must consider the impact it may provoke on the transmission of the neighbor sensors that use it as a router. Each sensor node includes its Inactivity Period within the packet that it has to transmit. In lines 7–10 of **Algorithm 1**, before the node transmits and enters in the sleeping mode, it verifies if it has to forward packets from neighbors. If there are no neighbors using S_i, the inactivity period $\mathrm{IP}_i[n]$ depends solely on the quantity the sensor measures and the ones available at the sink for reconstruction. However, if S_i is used as a router for packets from other sensors then its inactivity period computation must consider the sleeping of its neighbors.

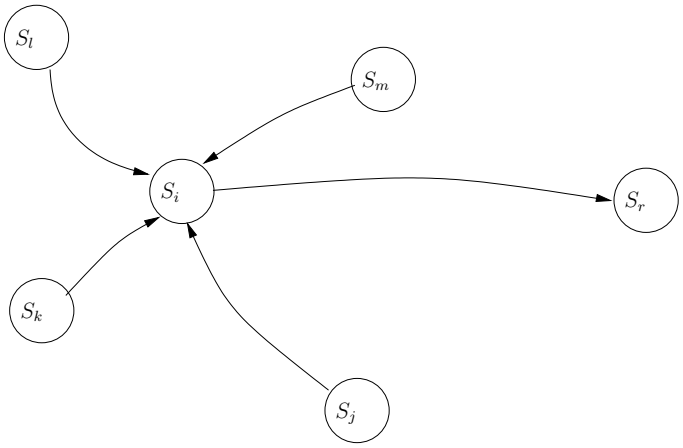

Figure 2.5: Example of a topology with a sensor and its neighbors in a WSN.

Therefore, for router nodes, the inactivity period IP_i of each network node S_i

must consider the inactivity periods of its neighbors (nodes that use S_i as a router), because S_i has to be awake to forward packets from them. DECA considers that by making the inactivity period of a sensor S_i to be defined by

$$IP_i[n] = \min\left(IP_i[n], \{IP_k[n]\}_{k \in \#SN_i}\right), \tag{2.40}$$

where each $IP_k[n]$ represents the inactivity period of each neighbor of S_i and $\#SN_i$ represents the set of neighbors of S_i. In this work, eq. (2.40) is applied at each sensor node in the WSN, since any of them can act as a router. The impact and effectiveness of such an strategy are evaluated in following sections.

2.4.2 Some Comments on DECA and its Parameters

DECA aims at tracking the variation rate of the monitored process. Nodes should sleep for shorter periods when the monitored process varies more rapidly, because they have to transmit more packets. Otherwise, nodes tend to sleep for larger periods when the variation of the monitored process reduces.

DECA has a very low complexity, as each node S_i basically computes the bounds on IP_i from eqs. (2.38) and (2.39) and chooses the smallest value. If the node also acts as router then it computes the minimum among that bound and the inactivity periods received from neighbors. Each sensor needs to store just the last two measurements of the process and their corresponding measurements instants as well as the two last transmitted measurements and corresponding measurements instants. In the case the node is a router, it is also required to store a list with the inactivity periods of nodes that use it as a router.

As discussed, DECA employs three parameters: 1) an initial sampling interval ($T_{\text{granularity}}$) or inactivity period (which is adjusted locally in each sensor regarding what it measures); 2) δ, the acceptable maximum fraction of distortion; and 3) the parameter γ used to augment the probability of the node to be awaken to forward packets from its neighbors. S_i sleeps for $\gamma IP_i[n]$ seconds after executing its tasks (line 13 of DECA).

The parameter $T_{\text{granularity}}$ is the initial sampling interval. It is adjusted by DECA as time goes by. The inactivity period is initially defined to be equal to $T_{\text{granularity}}$. As the nodes sleep during their inactivity periods, samples are taken when the nodes awake and, therefore its inherent to DECA sampling the process at a non-uniform sampling rate whose elapsed time between samples is adaptively adjusted. One should note that this time lapse is not the period of the sensor clock, which, in actual deployments, for an awake sensor, is orders of magnitude shorter than this time lapse between successive samples.

The parameter δ is defined considering the maximum percentage error allowed in

the reconstruction process. The parameter γ is also defined for DECA deployment. The influences of both on DECA are thoroughly evaluated in the sequel.

2.5 Conclusions

In this chapter, the problem of using a WSN to make an energy-efficient sensing of a given signal of interest was presented and analytically modeled. Moreover, we have defined a simple first–order predictor that uses information available at sensors and sink nodes to estimate a period of inactivity of sensor nodes. A Distributed Energy Conservation Algorithm (DECA) for WSNs was proposed. DECA aims at keeping the reconstruction error of the monitored signal lower than a given threshold. In the next chapter, DECA is evaluated and simulation results are presented.

Chapter 3

Performance Evaluation and Simulation Results for DECA

3.1 Introduction

In this chapter, we present the methodology for the performance evaluation of proposed DECA. The energy model used by sensor nodes is presented. Some static simulation parameters and the simulation environment considered are also presented. The criteria used to evaluate the algorithm are defined, and simulation results are presented and discussed.

3.2 Performance Evaluation

3.2.1 Energy Model

We assume that each node operates switching between two states: i) Inactive – an energy saving state (sleep mode) [8]; or ii) Active. In active state a node is in one of four operation modes: a) Measuring mode; b) Processing mode; c) Transmission mode; and d) Receiving mode – each mode is related to a specific task performed by the node. Therefore, a state-based energy model is adopted in this work. However, we extend the energy model in [48] – an empirical model for TELOS commercial hardware – to consider the different operation modes.

The energy consumption of a node is estimated as a function of the period of time in which the node stays in the inactive and the active state and in the different operation modes. Therefore, one can use

$$\begin{aligned}\widehat{EC} =& t_I C_I + t_A C_A + t_M(C_A + C_M) + t_P(C_A + C_P) \\ &+ t_R(C_A + C_R) + t_T(C_A + C_T),\end{aligned} \tag{3.1}$$

to evaluate energy consumption, in which t_I, t_A, t_M, t_P, t_R and t_T are, respectively, the cumulative sum of intervals in which the node remains in inactive and active states, and in measuring, processing, receiving, and transmitting operation modes.

In the active state, a fixed consumption of 10 mJ/s is taken into account. This consumption does not considers performing any specific task. In the transmission mode, a linear relationship between energy consumption and the payload size of the transmitted packet was observed, while for reception, energy consumption was observed to be independent of payload size [49]. Moreover, a consumption of 0.034 mJ for transmitting 1 byte during 0.58 ms was observed [48]. Meanwhile, for reception and sleep modes, 62.4 mJ/s and 1.8 mJ/s of consumption are reported [48], respectively. Summarizing, the associated consumptions to each state and mode of the energy model employed are presented in Table 3.1.

Table 3.1: Energy consumption parameters employed in the simulations.

Node initial energy (J)	2.00
Transmission power (dBm)	-5
Reception sensibility (dBm)	-66
Radio range (m)	40
C_I: Inactive state Consumption (mJ/s)	1.80
C_A: Active state Consumption (mJ/s)	10.00
C_M: Measuring mode Consumption (mJ/s)	18.00
C_P: Processing mode Consumption (mJ/s)	18.00
C_R: Rx mode Consumption (mJ/s)	62.40
C_T: Tx mode Consumption (mJ/s)	58.62
Payload size (byte)	1

3.2.2 Other Simulation Parameters

WSNs monitoring environmental data (temperature and humidity signals) are considered. The data employed comes from the WSN of the Intel Berkeley Research Lab, where for more than a month 54 sensor nodes sensed environmental data [18, 44]. These data correspond to temperature and humidity signals (among others), which indeed depend on the sensor node coordinates x_i and y_i, and on time t (i is the index of a given node S_i). We use $sT_i(t) = s_T(x_i, y_i, t)$ and $sH_i(t) = s_H(x_i, y_i, t)$, to refer to them, respectively.

A multi-hop communication model is considered, using the Ad-hoc On Demand Distance Vector (AODV) routing protocol [50, 51], in order to forward packets hop-by-hop from sources to the sink. In simulations, a WSN with 15 of the 54 nodes of the Berkeley WSN is considered. The sink node is located in the position of node S_{20} of [18], (0.5,17) in meters. Coordinates are relative to the upper right corner

of the lab. In each simulation run, the positions of the fourteen sensor nodes are drawn from the remaining fifty-three nodes. We have performed 10 simulation runs.

The presented simulations consider as benchmark a WSN at which nodes sample and transmit the sensed variable at $R_s = 1/T_{\text{granularity}}$. DECA starts with the same value for sensors inactivity periods. However, this value is adjusted for each and every sensor by DECA, as samples are taken and packets are transmitted. In addition, sensors are considered to have an initial energy of 2J stored in their batteries.

We consider that all sensor values are correct, without noise. As the sensed process is reconstructed at sink, there is a constraint in DECA imposing that the reconstruction error has to be kept within an acceptable distortion criterion, defined by δ.

Simulations were performed using TrueTime 1.5 [52], an environment based in MatLab/Simulink, and the network standard was the IEEE 802.15.4 [53] [54].

3.2.3 Evaluation Criteria

The criteria used to evaluate the results obtained with DECA are:

- **Reconstruction Error:** This is defined in eq. (2.4). In addition, to obtain a more precise evaluation of the reconstruction error, we employ the Cumulative Distribution Function (CDF) of it in our analyses;

- **Lifetime:** As pointed out in Section 1, we adopt as network lifetime the elapsed time until the first node exhaust its energy. Network lifetime is defined as

$$\text{LI} = \arg\min_t \{\mathcal{E}_i(t) = 0,\ \forall\ S_i \text{ in the network}\}, \tag{3.2}$$

  in which $\mathcal{E}_i(t)$ is the energy of node S_i at instant t;

- **Transmission Decrease:** This evaluates the reduction in the amount of transmissions (in %) as compared to a network without any energy management – for identical node positions and operation starting time. It is defined as

$$\text{TX}_d = 100\frac{\text{TX}_{\text{Tot}} - \text{TX}_{\text{DECA}}}{\text{TX}_{\text{Tot}}}, \tag{3.3}$$

  in which TX_{Tot} is the overall amount of transmissions in the network without any energy saving mechanism and TX_{DECA} is the amount of transmissions with DECA;

- **Lifetime Increase:** This evaluates the increase of the network lifetime in percentage, by comparing the network lifetime using DECA against the lifetime of

the network (with identical node positions and operation starting time) without any energy management scheme. It is important to highlight that transmission decrease (TX_d) and lifetime increase (LI) are computed with respect to simulations without any energy saving method, in which nodes measures samples and transmit them at the granularity setting ($T_{\text{granularity}}$);

- **Packet Delivery Ratio:** This is the ratio between the amount of received packets and the amount of transmitted packets.

$$\mathrm{PDR} = \frac{\text{received packets}}{\text{transmitted packets}}; \tag{3.4}$$

- **Success Ratio:** This is the ratio between the amount of packets that a node S_i has forwarded to the sink S_r and the amount of packets that its neighbors S_j forwarded to it. That is, the success ratio considers all packets that S_i has to forward, even if it is sleeping. Therefore, one defines

$$\mathrm{SR}_i = \frac{\sum_{j=1}^{N} \text{forwarded packets}_i^{j \ldots N}}{\sum_{j=1}^{N} \text{forwarded packets}_{j \ldots N}^{i}}, \tag{3.5}$$

  in which i is the index of a relay node S_i, and $j \ldots N$ are the indexes of each neighbor of S_i. This metric aims at evaluating the amount of packets that may have been lost by a given relay S_i.

The success ratio of S_i in forwarding packets from its neighbors may vary with its inactivity period IP_i. If S_i sleeps for a long period, in order to save energy, then a reduction in the success ratio may be observed due to possible packet losses. From **Algorithm 1**, one can see that a node S_i estimates its inactivity period and sleeps for IP_i seconds. To calculate IP_i, S_i uses information it knows about how long its neighbors will be inactive, and a sleeping period reduction factor γ ($0 < \gamma < 1$) is used to increase the probability of this node being awake to forward packets. Then, a direct influence of γ on the success ratio is expected. The γ factor is also considered to compensate for the lack of synchronization among the clocks of sensor nodes. Another relevant issue that must be investigated is the influence of the quantity of neighboring nodes $\#\mathrm{SN}_i$ that use S_i as a router on the success ratio. One can expect the success ratio to be affected by $\#\mathrm{SN}_i$. Figure 2.5 presents an example in which node S_i forwards packets from its neighbors $\mathrm{S}_{i,\#} = \{\mathrm{S}_j, \mathrm{S}_k, \mathrm{S}_l, \mathrm{S}_m\}$ to a sink node S_r. We employ such scenarios both to evaluate the success ratio and to analyze the network connectivity. More specifically, the effects caused by: 1) the factor γ; and 2) the amount of nodes using S_i as a router, in both success ratio and packet delivery ratio are addressed.

3.3 Simulation Results

Initially, we consider a fifteen-node WSN used to sense temperature and humidity real data. For the presented results, $\gamma = 0.5$ (the effects of γ are investigated in Section 3.3.2) and threshold δ values of 1%, 2%, 5%, and 8% are used. Each simulation scenario is run ten times (randomly selecting the 15 nodes composing the WSN from the 54 possible ones), and 95% confidence intervals for the mean are used when applicable in result graphs, where these intervals are represented by vertical bars.

The results are presented using four subsections, aiming at evaluating different aspects. In the first subsection, the objective is to evaluate the proposed algorithm in augmenting network lifetime, while keeping the error within a predefined fraction. In the second subsection, one aims at evaluating how relay nodes behave when executing the algorithm, that is the node has to act also as a router, an important aspect for multi-hop sensor networks. In the third, it is investigated how the algorithm behaves in terms of network scalability; networks of different sizes are considered. In the fourth, one aims at investigating the impact of the time granularity in the simulations with DECA, so that possible bias in the evaluation criteria can be overcome.

3.3.1 Overall Algorithm Evaluation

Figure 3.1 presents the percentage reduction in the amount of transmissions and the network lifetime increase, as a function of δ, for the monitored environmental data: temperature and humidity. The percentages are computed with respect to networks without any energy management strategy. Nodes take measures and transmit them periodically at $T_{\text{granularity}}$. We consider a $T_{\text{granularity}}$ of 0.1 s. It should be noted that the actual sampling rate used in the WSN in [18] was 31 s. Here, we assume that the data from [18] is originally sampled at 0.1 s instead of 31 s, that is as if the process was faster than it originally is. Results for other sampling rates are presented in Section 3.3.4. Figure 3.1 shows that increasing δ leads to a reduction of transmissions and therefore augments the network lifetime. Moreover, increasing δ makes nodes to transmit measured samples with greater variation rates, because a larger reconstruction error is allowed. Therefore, nodes may have larger sleeping periods and then transmit fewer measurements. Figure 3.1 shows a gain in the network lifetime of up to 1,000% as compared to a WSN without any energy management.

Figure 3.2 shows the Cumulative Distribution Function (CDF) of the reconstruction error of the monitored signals (temperature and humidity) for the different δ considered. These curves allow to evaluate if the reconstruction error is kept within the desired threshold δ. The increase of δ leads to the increase of the largest recon-

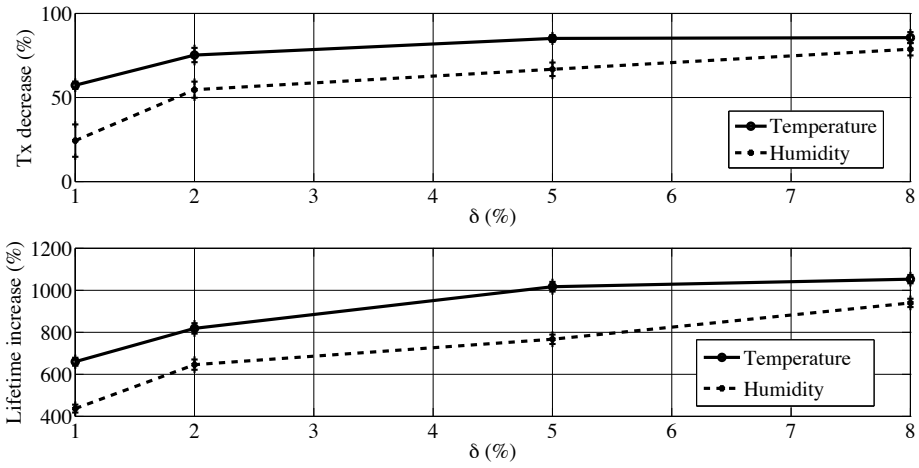

Figure 3.1: Transmission decrease and lifetime increase $\times$ δ for temperature and humidity signals.

struction error. This behavior is in consonance with results presented in Figure 3.1, and derives from the fact that as δ increases, fewer samples are used to reconstruct the monitored signals. However, more important is to observe in Figure 3.2 that the reconstruction error is always smaller than the set threshold (δ). In order to show that explicitly, Table 3.2 shows the maximum percentage reconstruction error, as a function of the parameter δ, for the two monitored fields. It can be verified that the imposed constraint (reconstruction error within a desired fractional margin) is satisfied in all cases.

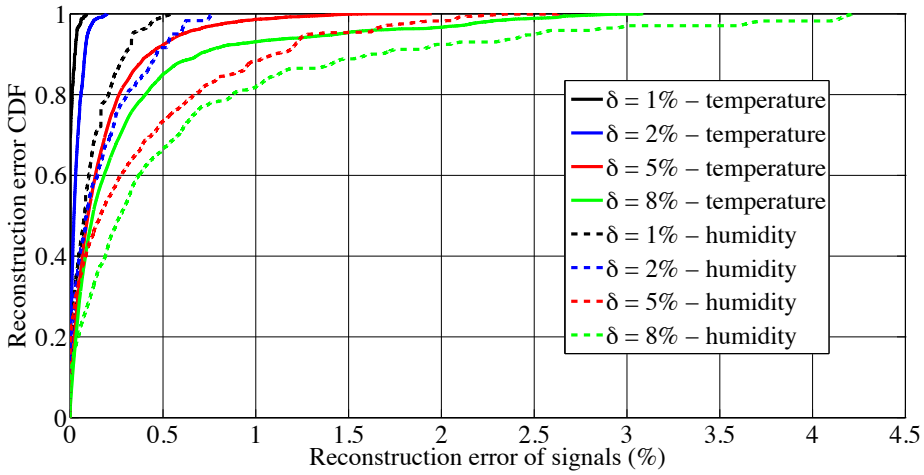

Figure 3.2: Reconstruction error CDF of temperature and humidity signals.

We can observe from Figure 3.2 and from the data in Table 3.2 that the reconstruction error for humidity is greater than for temperature. In order to understand this, we evaluate the variation rate of these monitored signals. Using the partial

Table 3.2: Maximum percentage reconstruction error (emax) $\times$ δ for temperature and humidity signals.

δ (%)	emax (%) – Temperature	emax (%) – Humidity
1	0.1987	0.5430
2	0.2526	0.7615
5	1.9479	2.6297
8	3.0879	4.2182

derivative of both signals with respect to time, we have

$$sT_i'(t) = \frac{\partial sT_i(t)}{\partial t} \text{ and } sH_i'(t) = \frac{\partial sH_i(t)}{\partial t}. \tag{3.6}$$

This provides an idea of the process variation rates. Figure 3.3 shows the variation rate of the monitored signals, for one node (S_1) in one simulation run, for $\delta = 2\%$. Considering all sensor nodes and sensing time, for the temperature field, the maximum of this rate is 0.0294^oC/s and, for the humidity field, the maximum is 0.202%/s. The means of these variation rates are 0.00002^oC/s and 0.00051%/s, respectively. The humidity signal presents a larger variation rate than the temperature signal, being thus a faster process. As DECA considers the variation rate of the signal, the greater the variation rate, the higher the number of required transmissions. Thus, DECA requires more transmissions for monitoring humidity than for temperature, an inherent feature of the adaptiveness of DECA. This explains the less significant LI when monitoring humidity, as shown in Figure 3.1. In addition, when using DECA, as the variation rate decreases, nodes tend to sleep for larger time periods. For example, considering $\delta = 2\%$, the means of the inactivity periods over all nodes and time are 3.7118 s and 1.1659 s when DECA is applied for monitoring temperature and humidity respectively. The maximum values of the inactivity periods are of 14.6594 s and 10.4643 s, respectively (see Figure 3.4). These values show that DECA correctly adapts the sleeping periods of sensor nodes according to the behavior of the physical quantity being monitored.

Figure 3.5 shows monitored and reconstructed temperature and humidity signals for one node in one simulation run, when $\delta = 1\%$ and $\delta = 2\%$. One can see that the reconstructed signals closely follow the monitored ones. It is also observed that a more precise reconstruction is obtained for the smaller value of δ, since DECA imposes a smaller fractional reconstruction error.

Table 3.3 shows the Packet Delivery Ratio (PDR) for different δ when considering the monitoring of both signals. A 95% confidence interval for the mean is also presented. PDRs above 90% were obtained in all cases. For larger δ, nodes sleep

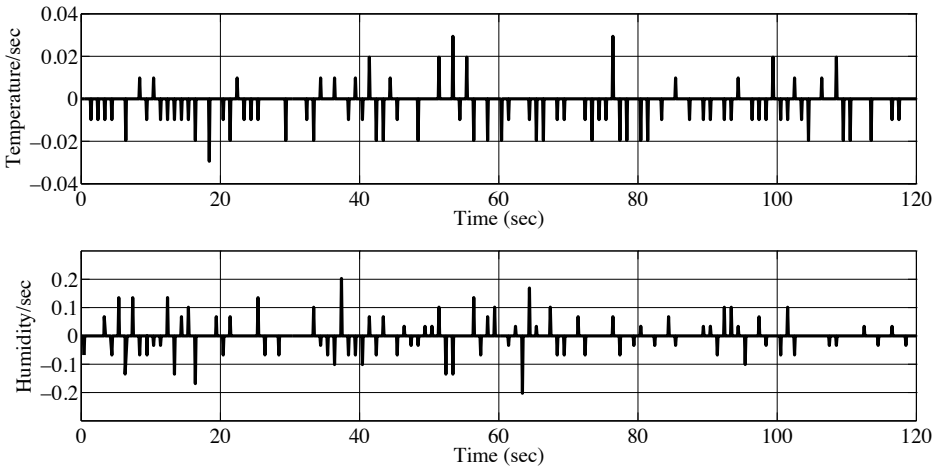

Figure 3.3: Variation rate of monitored temperature and humidity signals, for $\delta = 2\%$.

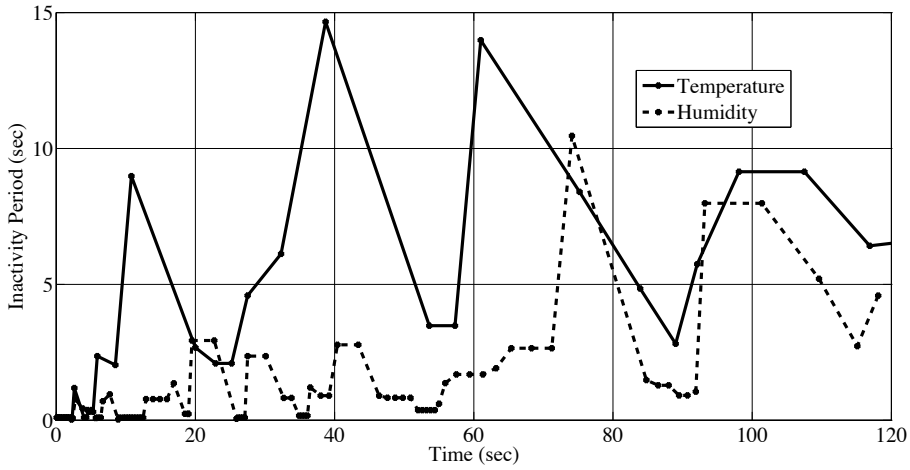

Figure 3.4: Inactivity periods for S_1 node with $\delta = 2\%$ for temperature and humidity signals.

more, which may impact network connectivity. Moreover, we verified smaller values of PDR, for the case in which the nodes were monitoring humidity. Since this signal presents a larger variation rate, then more transmissions are required for it, which may cause more contention to access the medium and consequently collisions.

3.3.2 Routing Nodes

We now analyze how does DECA impact WSN nodes that route packets for other nodes, i.e., they are relays. In this scenario, the topology considered is exemplified in Figure 2.5, in which node S_i has to forward packets from its neighbors to the sink node S_r. We consider cases in which S_i has 1, 5 and 13 neighbors. We con-

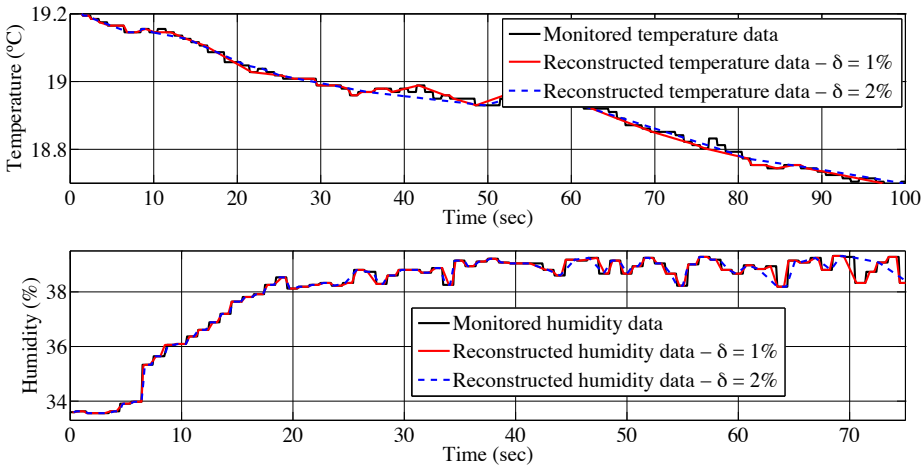

Figure 3.5: Monitored and reconstructed temperature and humidity signals, for $\delta = 1\%$ and $\delta = 2\%$.

Table 3.3: Packet Delivery Ratio (PDR) $\times$ δ for temperature and humidity signals.

δ (%)	PDR (Temperature)	PDR (Humidity)
1	0.9774 ± 0.0045	0.9546 ± 0.0050
2	0.9750 ± 0.0045	0.9500 ± 0.0030
5	0.9420 ± 0.0050	0.9333 ± 0.0100
8	0.9180 ± 0.0150	0.9166 ± 0.0080

sider that the WSN is monitoring temperature and δ is set as 2%. Since γ makes nodes sleep for a fraction of the computed inactivity periods, the values in the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ are examined. Figure 3.6 shows the evaluation of network connectivity in two perspectives: i) micro, by analyzing the local connectivity of the relay with the success ratio metric; and ii) macro, by analyzing the packet delivery ratio metric. In each case, the presented results are means for ten runs and the 95% confidence intervals are also shown.

As γ increases, the node S_i sleeps for a period of time closer to the minimum inactivity period among the ones of its neighbors. This is expected to augment the probability of S_i not to forward packets from its neighbors, because it may sleep for longer periods. In Figure 3.6, we can see this behavior. As γ increases, there is a decrease in SR. Furthermore, if S_i has to forward packets from a larger number of neighbors, this situation worsens. For the same reason, it is also observed that as γ increases, PDR decreases, meaning that the network is losing connectivity.

Since in this work one aims at the energy-efficient reconstruction of a monitored process by a WSN, it is important to evaluate how γ affects the energy consumption of sensor nodes. For the topology exemplified in Figure 2.5, where S_i has to forward packets from its neighbors, one knows that the IP of S_i (IP_i) is a function of the

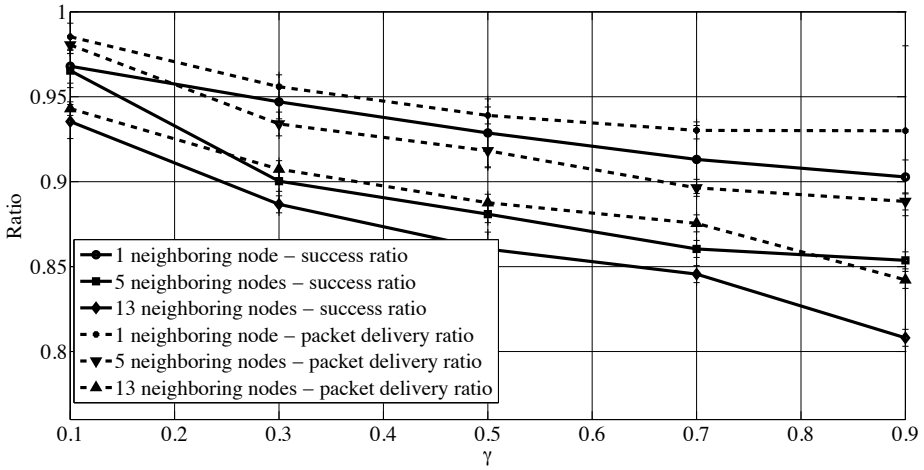

Figure 3.6: Success ratio and packet delivery ratio $\times$ γ for $\delta = 2.0\%$ for a temperature signal.

inactivity period of its neighbors and also of γ, as presented in eq. (2.40). As $\gamma \in (0, 1)$, from eq. (2.40), one observes that the smaller the γ value, the shorter the IP_i, and, as a consequence, the lower the energy savings. Figure 3.7 shows LI as a function of γ, for $\delta = 2\%$ when 1, 5 and 13 neighboring nodes ($\#\text{SN}_i$) are served by S_i as router. As expected, one observes that increasing γ leads to an increase of the network lifetime independently of $\#\text{SN}_i$, because nodes can sleep for longer periods. When $\#\text{SN}_i$ increases, the sleeping task becomes more critical as there are more packets to forward, and the gain in network lifetime decreases, although these gains are still significant.

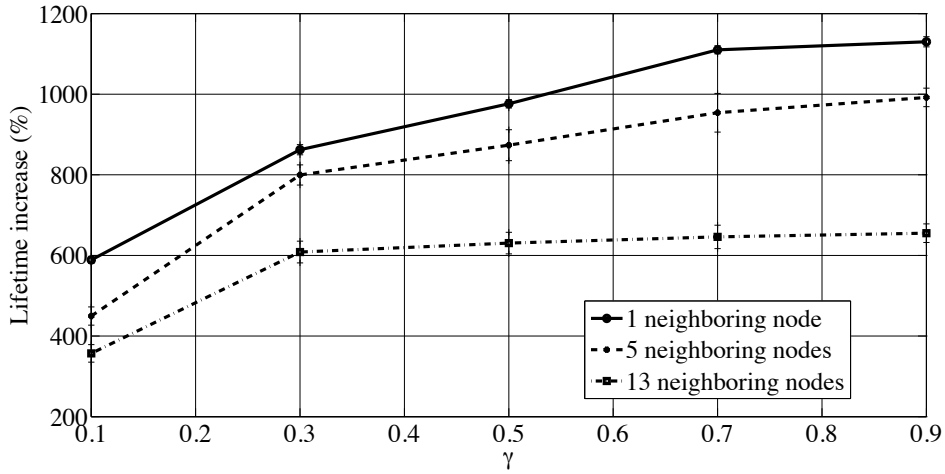

Figure 3.7: Lifetime increase $\times$ γ for $\delta = 2.0\%$ for a temperature signal.

However, one should still ponder if connectivity and network lifetime are somehow related. Let $\mathbf{SR}_\gamma$, $\mathbf{PDR}_\gamma$, and $\mathbf{LI}_\gamma$ be vectors containing the values of SR,

PDR, and LI, respectively, for a given γ obtained with different runs. One may compute the correlation coefficient between $\mathbf{SR}_\gamma$ and $\mathbf{LI}_\gamma$, between $\mathbf{PDR}_\gamma$ and $\mathbf{LI}_\gamma$, and between $\mathbf{PDR}_\gamma$ and $\mathbf{SR}_\gamma$, to evaluate the joint behavior of each variable pair. The correlations coefficients are given by

$$\mathrm{Corr}_\gamma(\mathrm{SR}, \mathrm{LI}) = \frac{\langle \mathbf{SR}_\gamma, \mathbf{LI}_\gamma \rangle}{\|\mathbf{SR}_\gamma\| \ \|\mathbf{LI}_\gamma\|}, \tag{3.7}$$

$$\mathrm{Corr}_\gamma(\mathrm{PDR}, \mathrm{LI}) = \frac{\langle \mathbf{PDR}_\gamma, \mathbf{LI}_\gamma \rangle}{\|\mathbf{PDR}_\gamma\| \ \|\mathbf{LI}_\gamma\|}, \text{and} \tag{3.8}$$

$$\mathrm{Corr}_\gamma(\mathrm{PDR}, \mathrm{SR}) = \frac{\langle \mathbf{PDR}_\gamma, \mathbf{SR}_\gamma \rangle}{\|\mathbf{PDR}_\gamma\| \ \|\mathbf{SR}_\gamma\|}. \tag{3.9}$$

In these, $\langle \mathbf{x}, \mathbf{y} \rangle$ is the inner product between $\mathbf{x}$ and $\mathbf{y}$ and $\|\mathbf{x}\|$ is the norm of $\mathbf{x}$.

Figure 3.8 shows these correlations for DECA with $\delta = 2\%$ with different neighboring nodes for the reconstruction of a temperature signal. The solid and dashed curves present the correlations between the Success Ratio and Lifetime, and between Packet Delivery Ratio and Lifetime, respectively. In these curves, it is observed that the absolute value of the correlation reaches the maximum at $\gamma = 0.5$, where the statistical relationship is stronger. This indicates a tradeoff between connectivity and lifetime increase and, therefore, energy saving.

The dotted curves in Figure 3.8 show the correlations between the Packet Delivery Ratio and Success Ratio. Since both metrics decrease as a function of γ (as seen in Figure 3.6), their correlation is positive.

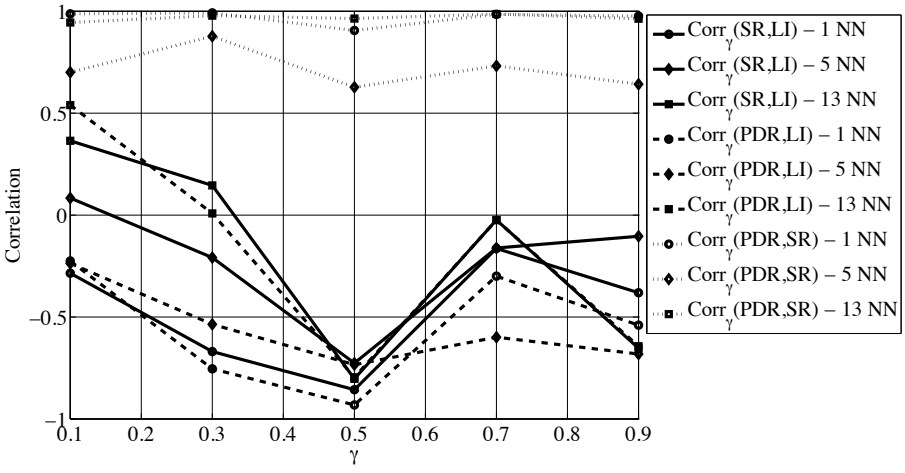

Figure 3.8: Correlations between SR and LI, between PDR and LI, and between PDR and SR for the reconstruction of a temperature signal. NN refers to the number of Neighboring Nodes.

3.3.3 Network Scalability

The increase in the number of nodes in a WSN may cause more transmissions and more contention for the physical medium. This may influence energy conservation by sensor nodes and network connectivity, and also the reconstruction of the monitored process. Thus, we evaluate network lifetime, packet delivery ratio, and reconstruction error of the sensed field, considering WSNs having different quantities of sensors. The impact of the network scalability on DECA is evaluated considering the monitoring of temperature [18] with 15, 30, and 50–nodes, which translate into varying node density.

Figure 3.9 shows LI as a function of δ for the 15, 30, and 50-node WSNs. As expected, in the three cases, as δ increases also does LI (this was already observed in Figure 3.1). One can also observe that for a given δ there is an increment in LI as the sensor density increases. Since network lifetime is defined here as the time period until the energy of the first node ends, the energy of routers may end sooner as the sensor density diminishes. If the WSN has more nodes then its sensor density increases and packet forwarding tasks can be shared among more routers, resulting in a better balance of energy consumption, redounding in the network lifetime increase.

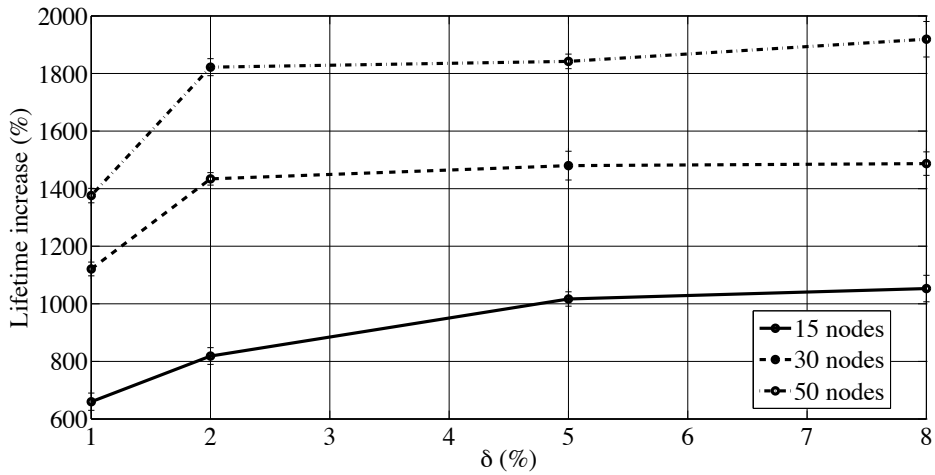

Figure 3.9: Lifetime increase $\times$ δ for a temperature signal.

Another aspect that can be impacted by sensor density is network connectivity. The greater is the number of nodes in the monitored area, the greater is the number of transmissions, which could lead to an increase in the contention for the medium. This behavior may result in more collisions, thus affecting network connectivity. In Figure 3.10, this is addressed by presenting the PDR for the three different network densities, where a decrease in the PDR can be observed as the quantity of nodes (network density) increases.

Moreover, the impact of the scalability in network connectivity may affect the reconstruction of the monitored field. As packets are lost, there are fewer samples available to reconstruct the field. Corroborating this, one observes in Figure 3.11 that the reconstruction error increases with the quantity of sensors in the network – the network density (Figure 3.11 considers $\delta = 1\%$). However, from the graphs in Figure 3.11, it is also verified that the constraint imposed by the algorithm of keeping the reconstruction error within a prescribed fraction is satisfied, even for the 50-node WSN.

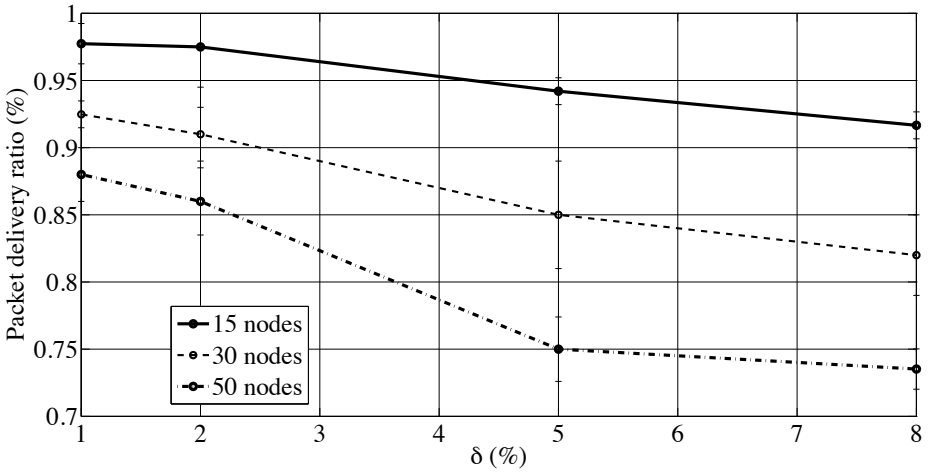

Figure 3.10: Packet delivery ratio $\times$ δ for a temperature signal.

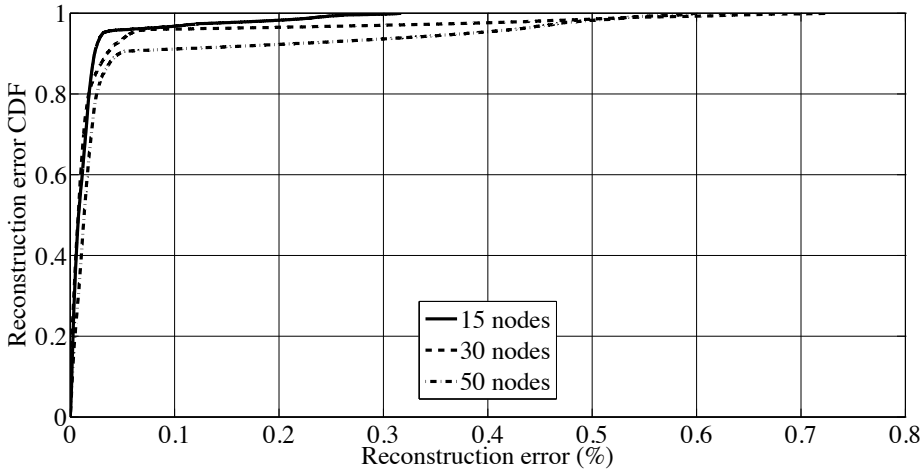

Figure 3.11: Reconstruction error CDF for the reconstruction of a temperature signal and for $\delta = 1.0\%$.

3.3.4 Time Granularity

For the previous analyses, time granularity was 0.1 seconds. The process is assumed to be sampled at this rate, and the nodes of the WSN that we have compared DECA against simply sample the signal of interest at 0.1 s and transmit it to the sink. In this section, we investigate the performance of DECA when other time–base is considered. The aim of such study is to evaluate the performance of DECA for processes having different speeds as a function of time. We do that by using three scenarios of different signal sampling rates 0.1 s, 1 s, and 31 s. For this experiment, one reconstructs the temperature signal and a fifteen–node WSN is used. We also set $\gamma = 0.5$ and thresholds $\delta \in \{1\%, 2\%, 5\%, 8\%\}$ are employed.

Figure 3.12 shows the percentage reduction in the amount of transmissions and the network lifetime increase, as a function of δ, with respect to a WSN in which the sensors simply sample and transmit measurements to the sink, with $T_{\text{granularity}} = 1$ s. A significant reduction in the amount of transmissions on the network and a significant lifetime increase (of more than 120%) are observed. These are similar to the results observed for $T_{\text{granularity}} = 0.1$ s. Albeit the transmission decrease is very similar in both cases, the gains in network lifetime are now smaller. This reduction in the gain of network lifetime is explained by the fact that the initial energy of nodes is the same (2J) for the two values of $T_{\text{granularity}}$.

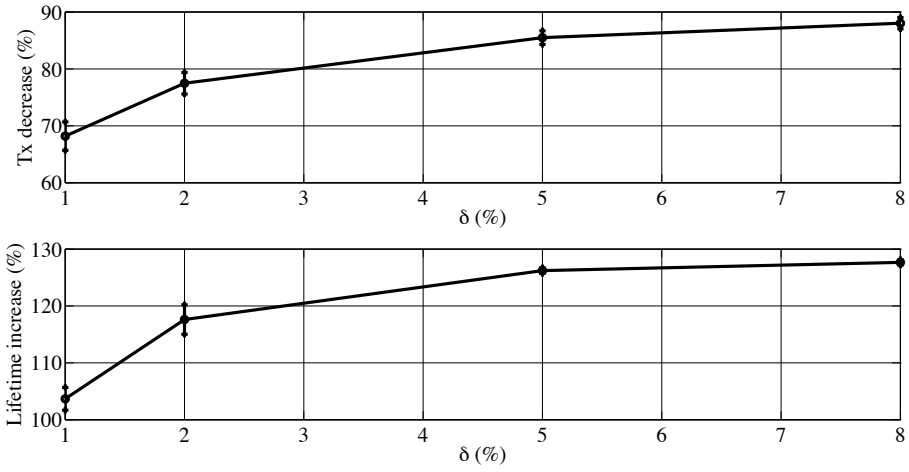

Figure 3.12: Transmission decrease and lifetime increase $\times$ δ for a temperature signal and for a time-base value of 1 s.

The Cumulative Distribution Function (CDF) of the reconstruction error of the monitored temperature signal for different δ is presented in Figure 3.13. It can be observed in Figure 3.13 that the reconstruction error for $T_{\text{granularity}} = 1$ s is always smaller than the threshold, obeying the proposed constraint. The constraint imposed by DECA that this error must be kept within the desired threshold (δ) is

satisfied.

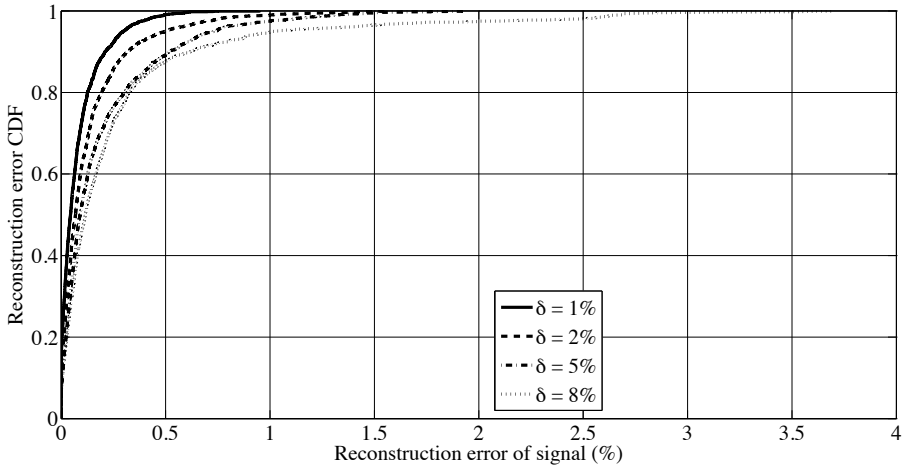

Figure 3.13: Reconstruction error CDF of temperature signal, for a time-base value of 1 s.

We increased the time granularity up to 31 seconds obtaining similar results, as presented in Figures 3.14 and 3.15. Table 3.4 shows the Packet Delivery Ratio (PDR) for time–base values of 0.1, 1 and 31 seconds and different δ. 95% confidence intervals for the mean are presented. PDRs around 90% and above are obtained, showing that the network connectivity is sustained for the different scenarios.

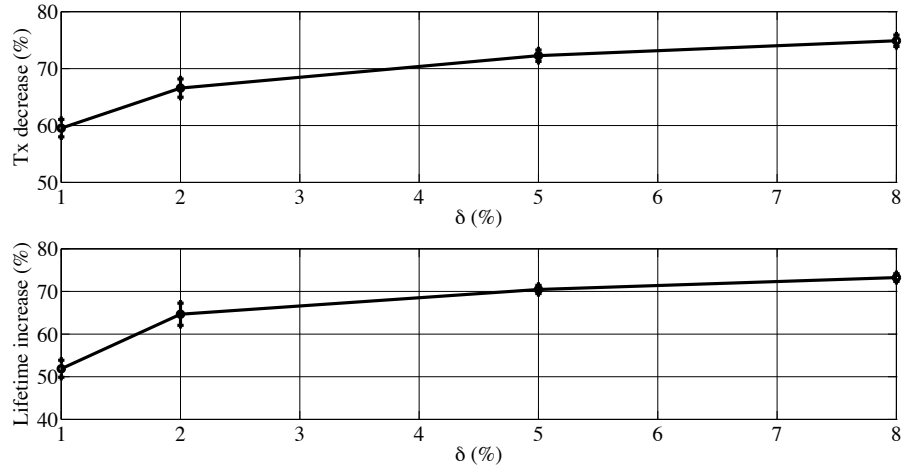

Figure 3.14: Transmission decrease and lifetime increase $\times$ δ for a temperature signal and for a time-base value of 31 seconds.

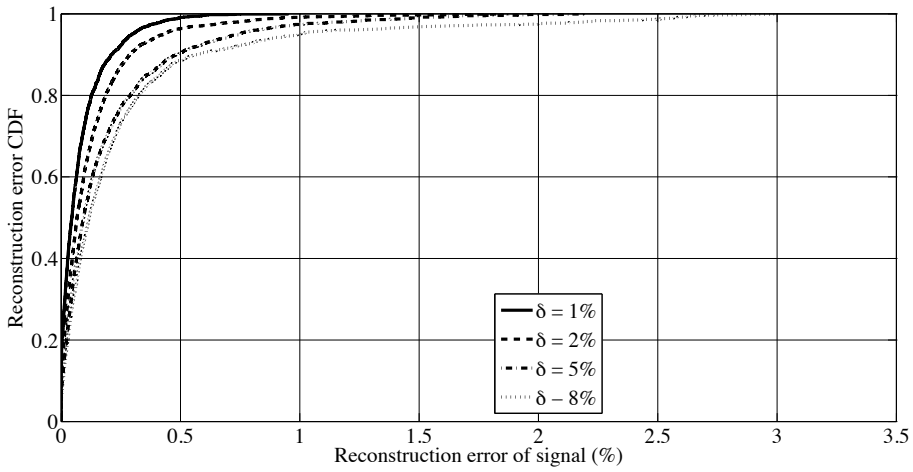

Figure 3.15: Reconstruction error CDF of a temperature signal, for a time-base value of 31 seconds.

Table 3.4: Packet Delivery Ratio (PDR) $\times$ δ, for a temperature signal and for time–base values of 0.1, 1 and 31 seconds.

δ (%)	PDR ($T_{\text{granularity}} = 0.1$ s)	PDR ($T_{\text{granularity}} = 1$ s)	PDR ($T_{\text{granularity}} = 31$ s)
1	0.9774 $\pm$ 0.0045	0.9579 $\pm$ 0.0018	0.9182 $\pm$ 0.0013
2	0.9750 $\pm$ 0.0045	0.9543 $\pm$ 0.0012	0.8959 $\pm$ 0.0017
5	0.9420 $\pm$ 0.0050	0.9341 $\pm$ 0.0020	0.8944 $\pm$ 0.0013
8	0.9180 $\pm$ 0.0150	0.9106 $\pm$ 0.0033	0.8815 $\pm$ 0.0014

3.4 Conclusions

In this chapter, we evaluated DECA. The energy model, simulation parameters and metrics used to evaluate the algorithm were presented. Real temperature and humidity signals were reconstructed, with errors that are within a predefined (desired) error range (the errors are kept lower than the predefined threshold). Significant gain in the network lifetime was obtained with DECA. A metric to evaluate the connectivity between routing nodes was proposed, and the influence of the network scalability at DECA was investigated. We also studied the impact of the time granularity in DECA, and verified significant gains in network lifetime, while keeping the reconstruction error of the monitored signal within the predefined range. In the next chapter, another paradigm is considered for energy conservation in WSNs and the Compressive Sensing framework is presented for this purpose.

Chapter 4

Compressive Sensing Applied to WSNs Collected Data

4.1 Introduction

This chapter starts with some fundamental ideas about Compressive Sensing (CS). We perform simulations to evaluate the rate–distortion behavior of the reconstruction of environmental data (temperature, humidity and illumination) using the CS framework. A comparison among three distinct recovery methods is presented: Newton with log–barrier, A*OMP and LASSO. Finally, an initial comparison between DECA and Compressive Sensing strategies is performed.

4.2 Fundamental Ideas

The Compressive Sensing framework relies on two principles: *sparsity* and *incoherence* [55] [56]. The first one explores the fact that some natural signals have a concise representation in proper bases. The second CS principle relies on the idea that signals that have a sparse representation in some basis must be spread out in the domain in which they are acquired.

Let a signal $\mathbf{x} \in \mathbb{R}^N$ (a column vector) be sparse in some domain and $\mathbf{s}$ be its sparse representation

$$\mathbf{s} = \mathbf{\Psi}\mathbf{x}, \tag{4.1}$$

in which $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$ is a transform that provides a sparse $\mathbf{s}$, for example, DCT or Wavelet. Consider that

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x}, \tag{4.2}$$

in which $\mathbf{y} \in \mathbb{R}^M$ $(M < N)$ is the vector that contains the coefficients (linear measurements) of $\mathbf{x}$ and $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is called *measurement (or sensing) matrix* [57] [58]. The entries of the measurement vector $\mathbf{y}$ are the inner products between signal $\mathbf{x}$ and test functions $\boldsymbol{\phi_m}$, which are the rows of the *measurement matrix*

$$y_1 = \langle \mathbf{x}, \boldsymbol{\phi}_1 \rangle , y_2 = \langle \mathbf{x}, \boldsymbol{\phi}_2 \rangle , \ldots , y_M = \langle \mathbf{x}, \boldsymbol{\phi_M} \rangle . \tag{4.3}$$

Once $\mathbf{y}$ is known, a reconstruction procedure would aim at finding the sparsest solution for $\mathbf{y} = \Phi\Psi^*\mathbf{s}$, the smallest l_0 norm for $\mathbf{s}$ (the amount of its non–zero entries), in which Ψ^* is the complex conjugate transpose of Ψ. Unfortunately, this is an ill posed problem requiring a combinatorial approach [55].

Alternatively, in order to circumvent the combinatorial problem, one minimizes the l_1 norm of the reconstructed signal [59]. In this case, the optimization problem is given by

$$\hat{s} = \min_{\mathbf{s}} \|\mathbf{s}\|_1 \qquad \text{s.t.} \qquad \mathbf{y} = \mathbf{\Phi x}, \qquad \mathbf{x} = \mathbf{\Psi}^*\mathbf{s}, \tag{4.4}$$

where $\|\mathbf{s}\| = \sum_{i=1}^{n} |s_i|$ is the l_1 norm of the signal $\mathbf{s}$, and s_i are components of $\mathbf{s}$.

There are some convex optimization algorithms that can be used to solve (4.4). In the next subsection, three distinct recovery methods considered in this work are briefly discussed: the Newton algorithm with a log–barrier method, used in the L1–magic MatLab toolbox [60]; the greedy–based algorithm called A*OMP [61]; and the shrinkage–based algorithm, called Least Absolute Shrinkage and Selection Operator (LASSO) [62].

4.3 Reconstruction Methods

4.3.1 Newton + Log–barrier

This recovery procedure uses a Newton algorithm with a logarithmic barrier (log–barrier) method [63] to solve the convex optimization problem

$$\min_x \ f_0(x) \qquad \text{s.t.} \qquad f_i(x) \leq 0, \ i = 1, \ldots, m. \tag{4.5}$$

The problem in eq. (4.5) is equivalent to

$$\min_x \ f_0(x) + \sum_{i=1}^{m} I_{-}(f_i(x)), \tag{4.6}$$

where

$$I_{-}(u) = \begin{cases} 0, & \text{if } u \leq 0 \\ \infty, & \text{otherwise.} \end{cases} \tag{4.7}$$

As $I_{-}(u)$ is difficult to optimize, in the log-barrier method, this function is replaced by a logarithmic-type function

$$\hat{I}_{-}(u) = \begin{cases} -\frac{1}{t}\log(-u), & \text{if } u < 0 \\ \infty, & \text{if } u \geq 0, \end{cases} \tag{4.8}$$

in which t is a parameter that defines the approximation between $I_{-}(u)$ and $\hat{I}_{-}(u)$. Then, the log-barrier function is defined

$$\phi(t) = \begin{cases} \sum_{i=1}^{m} -\log(-f_i(x)), & \text{if } f_i(x) \leq 0, i = 1, \ldots, m \\ \infty, & \text{otherwise.} \end{cases} \tag{4.9}$$

Thus, an alternative to eq. (4.5) is given by

$$\min_{x} \ \hat{f}_0(x; t) := f_0(x) + \frac{1}{t}\phi(x). \tag{4.10}$$

This problem can be translated into a CS recovery problem, that aims at minimizing the l_1 norm of $\mathbf{s}$ with quadratic constraints, i.e.,

$$\hat{s} = \min_{s} \|\mathbf{s}\|_1 \qquad \text{s.t.} \qquad \|\mathbf{y} - \mathbf{\Phi}\mathbf{\Psi}^{*}\mathbf{s}\|_2 \leq \epsilon, \tag{4.11}$$

in which the quadratic constraint is applied to solve the optimization problem in eq. (4.4).

This optimization is done by recasting the problem in eq. (4.11) as a *Second-Order Cone Program* (SOCP)[60]. Following, a Newton algorithm with a logarithmic barrier (log–barrier) method [63] is used to solve the optimization problem in eq. (4.11). This recovery procedure is implemented in the L1–magic package [60].

4.3.2 A* Orthogonal Matching Pursuit

The A*OMP is an algorithm based on the *Orthogonal Matching Pursuit* (OMP) algorithm [64], that uses atoms from a dictionary to expand $\mathbf{x}$. At each iteration, the expansion uses the dictionary atom having the largest inner-product with the residue. After each iteration, the orthogonal projection of the residue onto the

selected atoms is computed. We employ the notation:

- $V_n \in \mathbb{R}^N$ and $n = 1, 2, \ldots, N$ for dictionary atoms (columns of the dictionary Φ);
- $\mathbf{r}^l$ for the residue after the l'th interaction;
- $\mathbf{S}$ for the matrix (or set) of atoms selected from Φ for representing $\mathbf{y}$ (measurement vector);
- $\mathbf{c}$ for the vector of coefficients.

The initialization of OMP is

$$\mathbf{r}^0 = \mathbf{y}, \tag{4.12}$$
$$\mathbf{S} = \{\}, \tag{4.13}$$
$$\mathbf{c} = \mathbf{0}. \tag{4.14}$$

At iteration l, $\mathbf{S}$ is increased by the dictionary atom that best matches $\mathbf{r}^{l-1}$ doing

$$\mathbf{s} = \arg\min_{\mathbf{V}_n \in \Phi} \left\langle \mathbf{r}^{l-1}, \mathbf{V}_n \right\rangle, \tag{4.15}$$
$$\mathbf{S} = \mathbf{S} \cup \mathbf{s}, \tag{4.16}$$
$$\mathbf{c} = \arg\min_{\tilde{\mathbf{c}} \in \mathbb{R}^l} \|\mathbf{y} - \mathbf{S}\tilde{\mathbf{c}}\|_2, \tag{4.17}$$
$$\mathbf{r}^l = \mathbf{y} - \mathbf{S}\mathbf{c}. \tag{4.18}$$

At the end, $\mathbf{S}$ contains the support of $\mathbf{x}$ (the original signal to be coded), and $\mathbf{c}$ contains their nonzero entries (that define the sparsity of the signal).

The A*OMP considers the best-first search tree [65] [66], in which multiple paths can be evaluated simultaneously improving the reconstruction accuracy. It employs the A* search to look for the sparsest solution on a tree whose paths grow similar to the OMP [61] [67].

At the end of this iterative procedure, A*OMP aims at finding the smallest l_0 norm for s, solving the problem in eq. (4.4).

4.3.3 LASSO

The problem in eq. (4.4) can be expressed as

$$\min_{\mathbf{x}} \; k(\mathbf{x}) \qquad \text{s.t.} \qquad \|\Phi\mathbf{x} - \mathbf{y}\|_2 \leq \sigma, \tag{4.19}$$

in which k is convex, and σ is a measure of error [62]. According to [68], the problem in eq. (4.19) is equivalent to

$$\phi(\tau) := \min_{x} \ \|\Phi x - y\|_2 \qquad \text{s.t.} \qquad \mathrm{k(x)} \leq \tau. \tag{4.20}$$

The function $\phi(\tau)$ gives the optimal objective value as a function of the parameter τ. If $k(x) = \|x\|_1$, then eq. (4.20) is called the *Least Absolute Shrinkage and Selection Operator problem* [62]. Thus, the reconstruction of CS measurements can be expressed for LASSO algorithm as

$$\min_{\mathbf{x}} \ \|\Phi\mathbf{x} - \mathbf{y}\|_2 \qquad \text{s.t.} \qquad \|\mathbf{x}\|_1 \leq \tau. \tag{4.21}$$

The *Least Absolute Shrinkage and Selection Operator* (LASSO) was first developed as a least squares method for estimation in linear models. According to [62], it aims at minimizing the residual sum of squares, subject to the sum of the absolute value of the coefficients being less than a constant. In addition, LASSO shrinks some coefficients, and sets others to 0, in order to improve the prediction accuracy. This can be translated into signal sparsity, as we assume that $\mathbf{x}$ is compressible in some basis.

There is a connection between shrinkage methods and a minimal l_1-norm penalty [69]. Elad *et al.* [70] argues that a shrinkage method could be interpreted as the first iteration of an algorithm that solves the basis pursuit denoising (BPDN) problem. Thus, shrinkage-based algorithms, like LASSO, can be used to solve the problem in eq. (4.19). In this work, we implement LASSO using the SPGL1 software packet [71].

4.4 Rate–Distortion Analysis

We consider the application of CS for the sensing of environmental data in WSNs. This is expected to save sensor nodes energy by reducing the amount of transmissions from sensing nodes to the sink. Sensor nodes transmit quantized CS measurements. Thus, the deterioration in the rate–distortion performance caused by quantization must be analyzed. We evaluate the rate–distortion performance for the three reconstruction methods in Section 4.3 for three different environmental data: temperature, humidity and illumination.

4.4.1 Problem Model

We consider that a given node S_i takes samples from the environment, and stores them in a vector $\mathbf{x}_i \in \mathbb{R}^N$,

$$\mathbf{x}_i = [x_i[1], x_i[2], \ldots, x_i[N]]\,. \tag{4.22}$$

Compressive Sensing is used to generate $\mathbf{y}_i \in \mathbb{R}^M$ $(M < N)$ measurements,

$$\mathbf{y}_i = [y_i[1], y_i[2], \ldots, y_i[M]]\,. \tag{4.23}$$

These compressive sensed measurements are quantized using a uniform scalar quantizer of bit-depth B (that is, with 2^B reconstruction levels) generating a quantized vector $\mathbf{y}_{q_i}$. The quantized measurements are the ones that are transmitted to the sink node. In the best scenario (without any channel impairment) the vector

$$\mathbf{y}_{r_i} = [y_{r_i}[1], y_{r_i}[2], \ldots, y_{r_i}[M]] \tag{4.24}$$

is available at the sink node with the M received measurements for signal recovery.

We initially assume that $\mathbf{y}_{q_i} = \mathbf{y}_{r_i}$, i.e., all transmitted measurements are received at the sink node, with no distortion. We postpone the analysis of the impact of packet losses in the rate–distortion performance of the reconstruction of the monitored signal to Chapter 5.

4.4.2 Simulation Set-up

We consider a WSN with a single hop communication, where a sensor node S_i can transmit packets directly to the sink node, that is an ad hoc network and use real temperature, humidity and illumination signals, gathered by the 54–node WSN located in the Intel Berkeley Research Lab [18].

However, this single hop and single sensor node model can be easily extended to a scenario with multihop communication. It can be observed that the i index in eqs. (4.22), (4.23) and (4.24) represents measurements and transmissions of a sensor node S_i, and receptions (at sink S_r) from the same sensor node, respectively.

Suppose an example of a multihop communication with three nodes (S_i, S_j and S_r). We consider in this example that S_i and S_j are sensor nodes and S_r is the sink. It is also considered that S_i is a router (relay), and it forwards packets from S_j to S_r.

In this multihop communication example, the S_i and S_j nodes measure temperature samples from the environment

$$\mathbf{x}_i = [x_i[1], x_i[2], \ldots, x_i[N]] \text{ and} \tag{4.25}$$

$$\mathbf{x}_j = [x_j[1], x_j[2], \ldots, x_j[N]]\,. \tag{4.26}$$

The S_j node transmits its quantized measurements to S_r

$$\mathbf{y}_{q_j} = \left[y_{q_j}[1], y_{q_j}[2], \ldots, y_{q_j}[M]\right], \tag{4.27}$$

and the S_i node has to transmit its quantized measurements ($\mathbf{y}_{q_i}$) and forward the quantized measurements ($\mathbf{y}_{q_j}$) from S_j to the sink node.

The sink node receives measurements from both S_i and S_j sensor nodes

$$\mathbf{y}_{r_i} = [y_{r_i}[1], y_{r_i}[2], \ldots, y_{r_i}[M]] \text{ and} \tag{4.28}$$

$$\mathbf{y}_{r_j} = \left[y_{r_j}[1], y_{r_j}[2], \ldots, y_{r_j}[M]\right], \tag{4.29}$$

and reconstructs the original signals $\mathbf{x}_i$ and $\mathbf{x}_j$, separately.

Considering a multihop WSN application, the *Medium Access Control Sublayer* (MAC) deals with possible collision problems with the *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) protocol [53], when both S_i and S_j nodes want to transmit. In the network layer, the *Ad-hoc On Demand Distance Vector* (AODV) routing protocol [51] can be used to deal with the forwarding task of S_j sensor node.

For sensing matrix construction, M random waveforms of an $N \times N$ Noiselet [72] are used. An example of sensing matrix ($\mathbf{\Phi}$) for $N = 4$ is

$$\mathbf{\Phi} = \frac{1}{2}\begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}.$$

It is assumed that the sensed signal is sparse in the DCT domain [17]. Quantizers of different bit–depths are tested, in order to investigate the impact of quantization in the reconstruction of the signal. We suppose that an entropy coder [73] is used, so it is reasonable to assume that the average rate spent to encode each coefficient is given by

$$R = \frac{M}{N} \times H(\mathbf{y}), \tag{4.30}$$

where N is the dimension of the measured vector, M is the number of coefficients transmitted and $H(\mathbf{y})$ is the entropy (in bits per sample) of the quantized measured data $\mathbf{y}$. For entropy computation, we consider that each possible quantizer output value occurs at least once. Doing so, unused reconstruction values are adequately taken into account.

Distortion is defined in terms of the Mean Squared Error (MSE) normalized by

the signal energy,

$$\text{Normalized MSE} = \frac{E[(\mathbf{x} - \hat{\mathbf{x}})^2]}{||\mathbf{x}||^2}. \tag{4.31}$$

In eq. (4.31), $\mathbf{x}$ is the sensor output samples, $\hat{\mathbf{x}}$ is the reconstructed sensor output samples and $E[\cdot]$ is the expected value operator.

Table 4.1 presents the definition of some parameters used in simulations.

Table 4.1: Definition of some simulation parameters used in CS applied to WSNs collected data .

N	Length of the signal block
M	Quantity of CS measurements
B	Bit–depth used in quantization

At the presented rate–distortion curves each point represents the average over 300 runs, each with a different seed. The normalized MSEs are presented in dB.

4.4.3 Rate–Distortion Results

The RD results are presented using four subsections: the first subsection shows some instabilities observed in L1–magic results, possible causes for these unexpected behavior and some experiments are employed to mitigate these problems; after that, a comparison among the three recovery methods is exposed; moreover, the rate–distortion behavior of the reconstruction of the three environmental signals with LASSO algorithm is presented; finally, the impact of the length of the signal block in the reconstruction is discussed.

L1–magic instabilities

The recovery procedure for the L1–magic is based on eq. (4.11), and the ϵ parameter used to solve eq. (4.11) was chosen empirically. For each number of measurements and bit–depth $\{M_i, B_i\}$ the ϵ parameter was exhaustively varied, and the ϵ value that gives the lower distortion is employed. Figure 4.1 shows an example of this procedure, for the reconstruction of a signal block with $N = 512$ samples of a temperature signal, with $M = 128$ measurements and $B = 6$ bits. Table 4.2 presents some values of ϵ for $B = 4$ and $B = 6$.

This optimization on ϵ was done because of two main aspects: to achieve the lowest distortion for each (M, B) pair; and to mitigate some instabilities observed in the rate–distortion curves, for the L1–magic results. Figure 4.2 presents the rate-distortion curve for the reconstruction of $N = 512$ samples of a temperature signal, considering CS measurements varying within a set $\{8, 16, 32, 64, 128, 256\}$

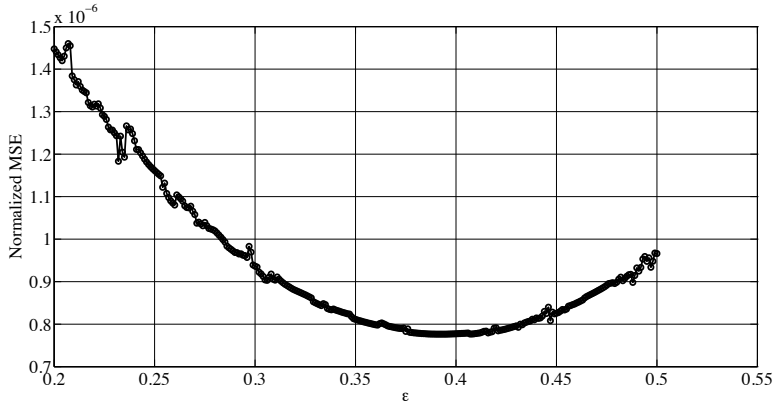

Figure 4.1: Value of the normalized MSE as a function of ϵ for reconstruction of a temperature signal with L1–magic with $N = 512$ samples, $M = 128$ measurements and $B = 6$ bits.

Table 4.2: ϵ values used in the reconstruction of $N = 512$ samples of a temperature signal with L1–magic, for $B = 4$ and $B = 6$ bit–depths.

B (bits)	M	ϵ
4	8	0.0001
4	16	0.0001
4	32	0.091
4	64	0.741
6	8	0.0001
6	16	0.0001
6	32	0.001
6	64	0.047

and bit–depth $B = 6$ bits, without optimization on ϵ. An undesirable peak in the distortion can be observed. This undesirable behavior is not expected, because it means that there is a worsening in distortion with the increase in the bit rate. To increment the rate means that there is a larger number of measurements to be used in the reconstruction of the original signal. Thus, there should be an improvement of distortion with the increase in bit rate.

Figure 4.3 shows the rate–distortion curves of the reconstruction of $N = 512$ samples of a temperature signal with L1–magic, after the optimization on ϵ (to choose the ϵ that leads to a minimal distortion). As expected, a decrease in the distortion can be observed with the increase in the amount of reconstruction levels (L), leading to an improvement of the signal reconstruction. However, the instabilities above mentioned can still be observed. Figure 4.4 presents the same curves from Figure 4.3, but with the number of CS measurements used ranging from $M = 8$ to $M = 400$, in steps of 2. This way, we consider a more dense set of measurements to

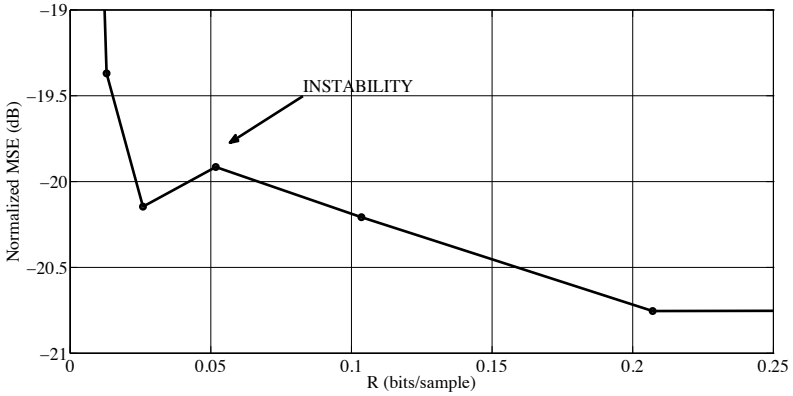

Figure 4.2: Instability of L1–magic, observed in the reconstruction of $N = 512$ samples of a temperature signal, for 6 bits.

better visualize the curves. The same unstable behavior can be observed.

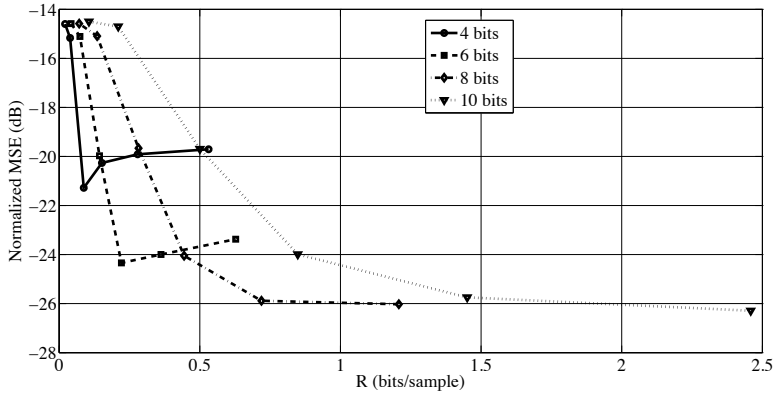

Figure 4.3: Rate–distortion curves for the reconstruction of $N = 512$ samples of a temperature signal with L1–magic.

Figure 4.5 presents the rate–distortion convex hulls for the reconstruction of a temperature signal block with dimension set to $N = 512$, when different amount of CS measurements (M) are considered. The convex hull (or operational curve) of the rate–distortion curves represents the points that lead to the better trade–off between rate and distortion, or the optimal quantizers observed in the curves. For the solid curve, one uses the number of measurements within the set $\{8, 16, 32, 64, 128, 256\}$ and the ϵ parameter is not optimized, and the dashed curve represents the results when M ranges from $M = 8$ to $M = 256$, in steps of 2, and the optimization of ϵ is considered. The bit–depths (B) varied within the set $\{4, 6, 8, 10\}$ bits, for both operational curves. It can be observed that the optimization on the ϵ parameter combined with a more complete set of CS measurements improved the rate–distortion performance of the reconstruction of the monitored signal.

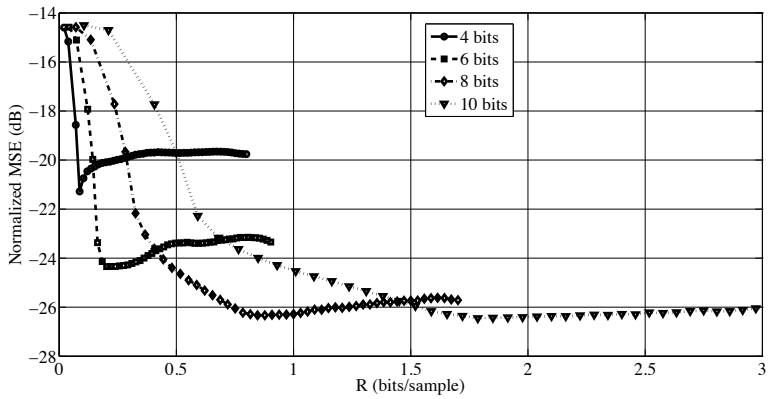

Figure 4.4: Rate–distortion curves for the reconstruction of $N = 512$ samples of a temperature signal with L1–magic with densely sampled number of measurements.

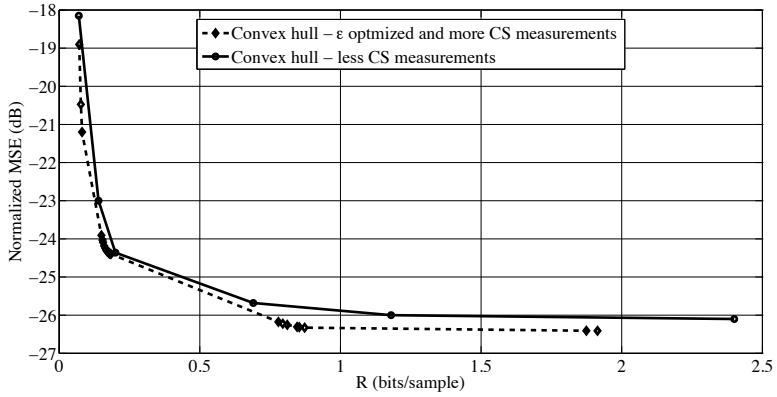

Figure 4.5: Rate–distortion convex hulls for the reconstruction of $N = 512$ samples of a temperature signal with L1–magic when ϵ is optimized and more number of measurements are used at reconstruction.

The L1–magic is a package widely used for image compression applications [25]. This work considers the reconstruction of signals with different characteristics of an image. Thus, an issue that can be investigated is the reconstruction of another environmental signal, in order to evaluate possible relations between the instability and features of the sensed signal. Figures 4.6 and 4.7 present the rate–distortion and operational curves (convex hulls) for the reconstruction of humidity and illumination signals, respectively. Signal blocks of $N = 512$ samples are considered and CS measurements ranging from $M = 2$ to $M = 256$ in steps of 1 are used. When these results are compared with those previously obtained for the reconstruction of temperature signal (Figure 4.4), one can verify a smaller instability in the rate–distortion curves for humidity and illumination signals.

From the obtained results, one can argue that the instability observed in the rate–distortion curves depends on the nature of the reconstructed signal (monitored

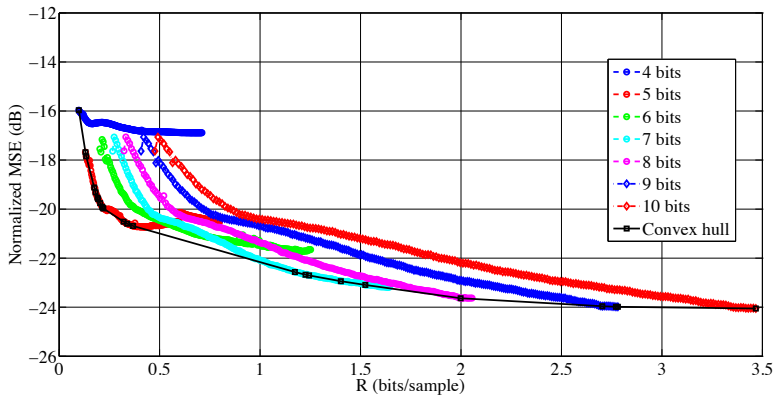

Figure 4.6: Rate–distortion curves and convex hull for the reconstruction of $N = 512$ samples of humidity signal with L1–magic.

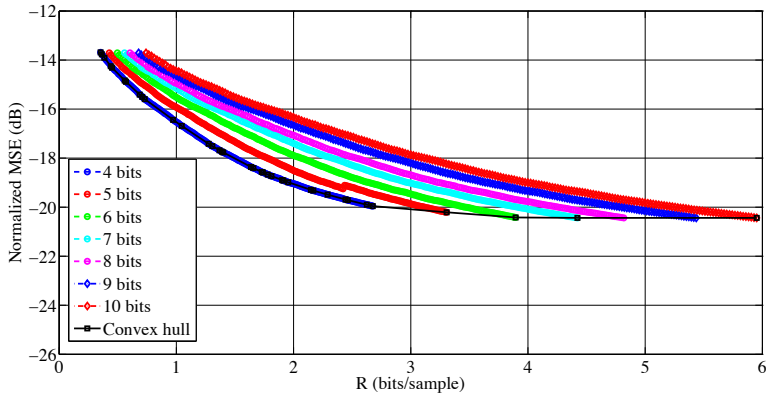

Figure 4.7: Rate–distortion curves and convex hull for the reconstruction of $N = 512$ samples of illumination signal with L1–magic.

by the WSN). In order to better visualize the impact of the characteristics of the monitored signal in the instability impaired by the recovery algorithm used by L1–magic package in the rate–distortion performance, Figures 4.8, 4.9, 4.10 and 4.11 are presented. These figures show a comparison of rate–distortion curves between temperature and humidity signals, for some bit–depths. The presence of instabilities is more visible on the temperature signal.

The number of representation levels used in the quantization of the CS measurements may also cause an increase in this unstable behavior, as showed in Figures 4.4, 4.6 and 4.7. There is more instability when less bits are used in quantization.

This thesis do not intend to investigate the causes for these instabilities or even to solve this problem. However, there are some issues to be highlighted: i) the optimization on ϵ is important to achieve a better rate–distortion performance and to mitigate the instability; ii) the characteristics of the monitored signal can be

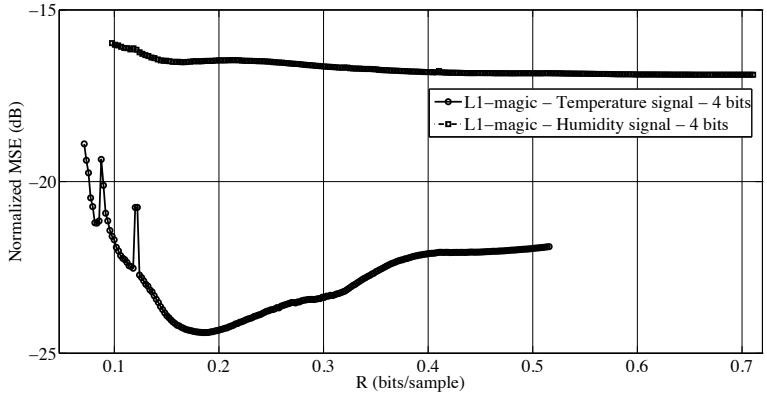

Figure 4.8: Rate–distortion curves for reconstruction of $N = 512$ samples of temperature and humidity signals with L1–magic and 4 bits.

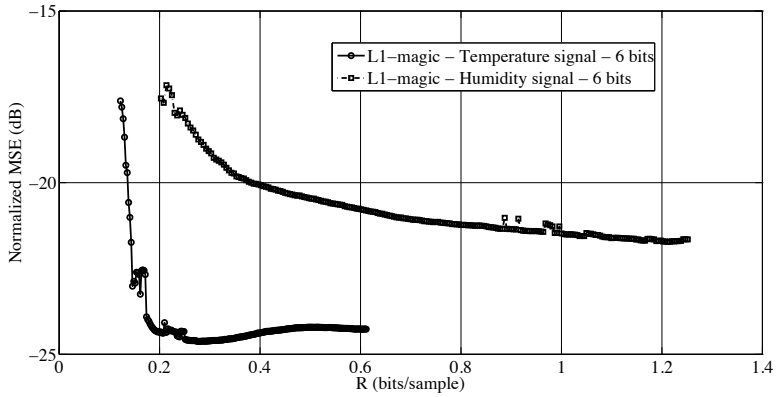

Figure 4.9: Rate–distortion curves for reconstruction of $N = 512$ samples of temperature and humidity signals with L1–magic and 6 bits.

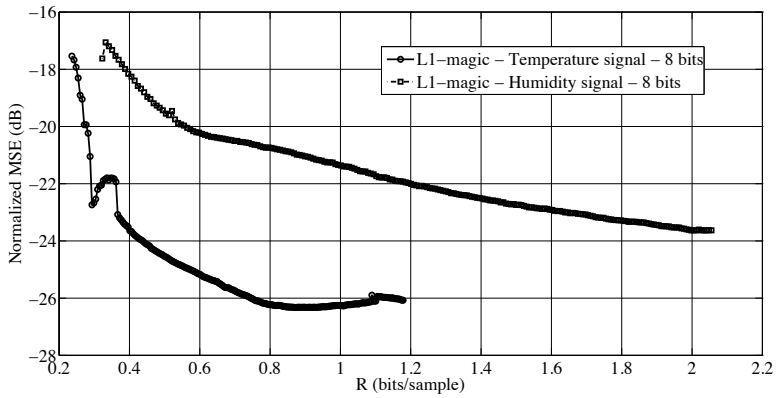

Figure 4.10: Rate–distortion curves for reconstruction of $N = 512$ samples of temperature and humidity signals with L1–magic and 8 bits.

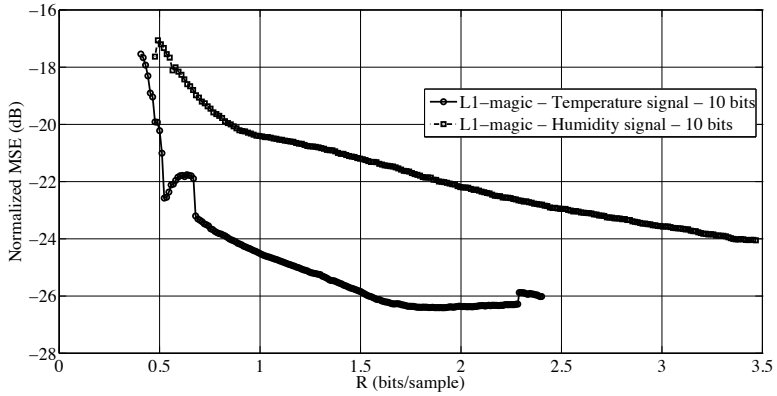

Figure 4.11: Rate–distortion curves for reconstruction of $N = 512$ samples of temperature and humidity signals with L1–magic and 10 bits.

related to these unexpected behavior; and iii) the number of representation levels used in quantization impairs these non–monoticities.

Comparison among recovery methods

Figure 4.12 presents the RD performance of the reconstruction of the temperature signal using L1–magic, A*OMP and LASSO. The dimension of the signal block is set to $N = 512$. We consider quantizers bit-depths (B) of $\{4, 6, 8, 10\}$ and the N samples are encoded using $M \in \{8, 16, 32, 64, 128, 256\}$ CS measurements. We consider the RD curves for the different B (as M increases) and for each of the three reconstruction algorithms a different distortion is obtained. Thus we present an RD curve for each algorithm and quantizer. As rate increases (in bits/sample), in general there is an improvement in the reconstruction of the signal (decrease of the normalized MSE) for the three recovery algorithms. This behavior is expected since more CS measurements are used in the reconstruction as the rate increases. These results also show that smaller reconstruction errors are obtained as the bit–depths of the quantizers increase.

The reconstruction using L1–magic presented a better rate–distortion performance than the ones for both A*OMP and LASSO. However, for L1–magic the obtained distortion does not always decrease as the rate increases. Such non-monotonic behavior has been observed in various simulation settings (previously discussed), and was found to be very sensitive to the ϵ values. The A*OMP and LASSO do not show such non-monotonic behavior. It also can be verified that the LASSO presents a better rate–distortion performance than the A*OMP.

In order to better investigate the reconstruction of the signals of interest, more dense RD curves were generated. The number of CS measurements used ranges from $M = 8$ to $M = 256$, in steps of 2; $N = 512$; and the bit–depths (B) varied within the

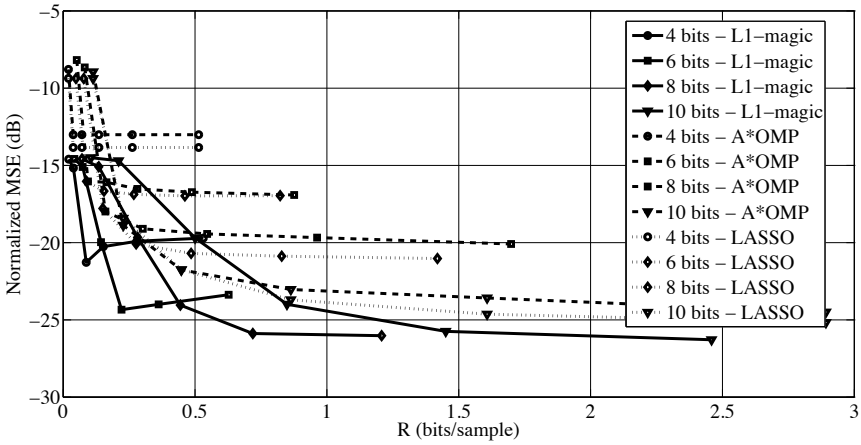

Figure 4.12: Rate–distortion results for L1–magic, A*OMP and LASSO for the reconstruction of $N = 512$ samples of a temperature signal, for several quantization bit–depths.

set {4,5,6,7,8,9,10}. The rate–distortion performances by means of their operational curves (convex hulls) of the temperature signal reconstruction with the L1–magic, A*OMP and LASSO methods are compared in Figure 4.13. It can be observed that the reconstruction with L1–magic is better than the ones with the other two reconstruction methods. However, as we verified previously in the above subsection, there are some non–monoticities in the rate–distortion curves of L1–magic. One can verify that the operational curve for the LASSO outperforms the operational curve for the A*OMP.

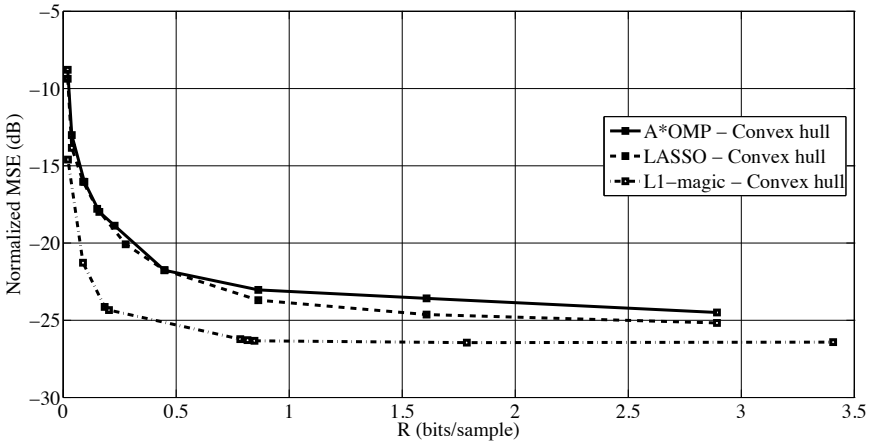

Figure 4.13: Rate–distortion convex hulls for reconstruction of $N = 512$ samples of a temperature signal with L1–magic, A*OMP and LASSO.

Rate–distortion results with LASSO

Based on the above results, we present rate–distortion performance of the reconstructions of $N = 512$ samples of temperature, humidity and illumination signals with LASSO. The number of CS measurements used ranged from $M = 8$ to $M = 256$, in steps of 2; and the bit depths (B) varied within the set {4,5,6,7,8,9,10}. Figure 4.14 shows the rate–distortion curves of the reconstruction of the temperature signal, for each bit–depth.

Similar behavior to the one observed in Figure 4.12 is verified in the results in Figure 4.14. There is a decrease in the normalized MSE with the increment of the rate, in bits per sample, and the more reconstruction levels used in the quantizer, the better is the rate–distortion performance. We present, in Figure 4.14, the rate–distortion curves for 4, 6, 8 and 10 bits, and the convex hull. The curves for 5, 7 and 9 bits show similar behavior. All curves were used to generate the convex hull.

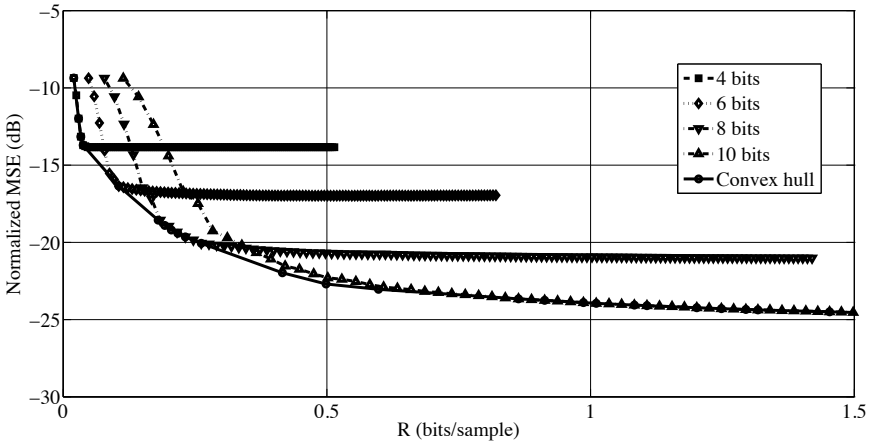

Figure 4.14: Rate–distortion curves and RD convex hull for the reconstruction of $N = 512$ samples of temperature signal with LASSO for several quantization bit–depths.

Figures 4.15 and 4.16 present the rate–distortion curves and the convex hulls for the reconstruction for the other two data: humidity and illumination signals, respectively. As expected, results show the same behavior observed for the reconstruction of temperature signal. While Figures 4.15 and 4.16 explicitly present the rate–distortion curves for 4, 6, 8 and 10 bits, the curves for 5, 7 and 9 bits have similar behavior. The curves for all these different values of B were used to generate the convex hulls.

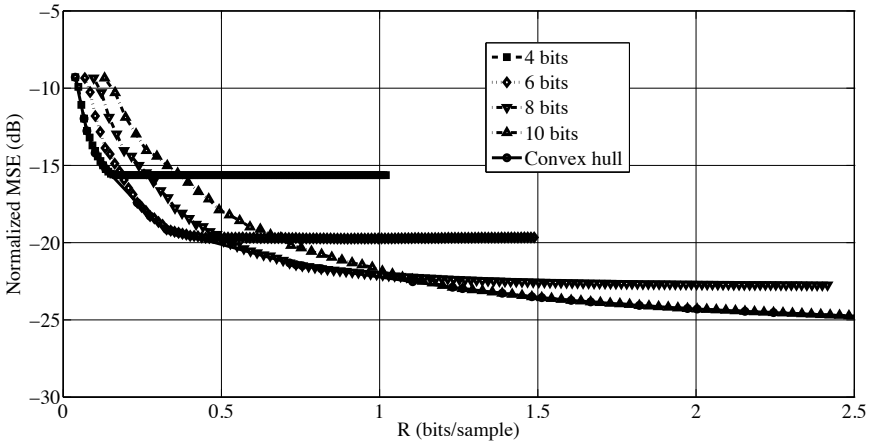

Figure 4.15: Rate–distortion curves and RD convex hull for reconstruction of $N = 512$ samples of humidity signal with LASSO for several quantization bit–depths.

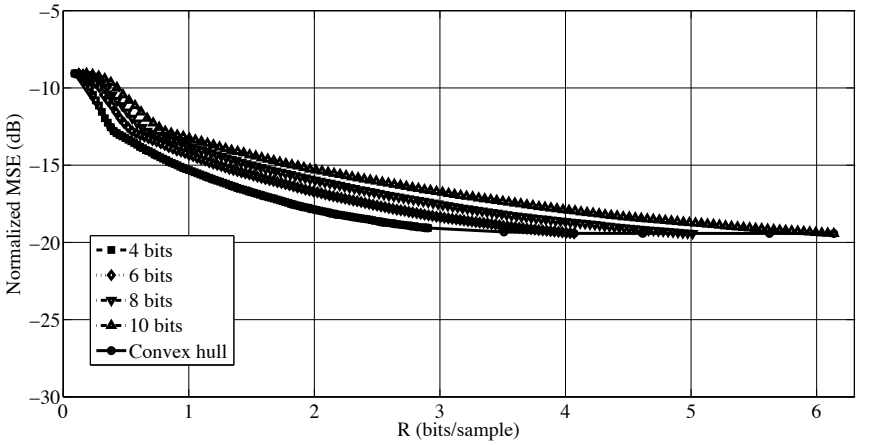

Figure 4.16: Rate–distortion curves and RD convex hull for reconstruction of $N = 512$ samples of illumination signal with LASSO for several quantization bit–depths.

Impact of the length (time support) of the monitored signal block (N)

Sensor nodes firstly store N samples of the signal of interest. After that, the CS framework is applied providing $M < N$ measurements. The M CS measurements are then quantized and subsequently transmitted to the sink node. In the above described experiments, we considered signal block of $N = 512$ samples in each sensor node.

Considering the same sample rate, the larger N is, then the higher is the consumption of sensor nodes battery. Therefore, we investigate the impact of the length of the signal block (N) in the rate–distortion behavior of the reconstruction of the temperature signal. The number of CS measurements, ranges from $M = 8$ to $M = 256$, in steps of 2 and the bit–depths (B) are within the set {4,5,6,7,8,9,10}.

We also consider three different signal block lengths, $N = \{128, 256, 512\}$. The reconstruction was obtained with LASSO.

Figure 4.17 presents the results obtained for this experiment. One can observe a better rate–distortion performance for larger signal blocks, for the same sample rate. However, a trade–off between the length of signal block and the energy consumption of sensor nodes has to be considered.

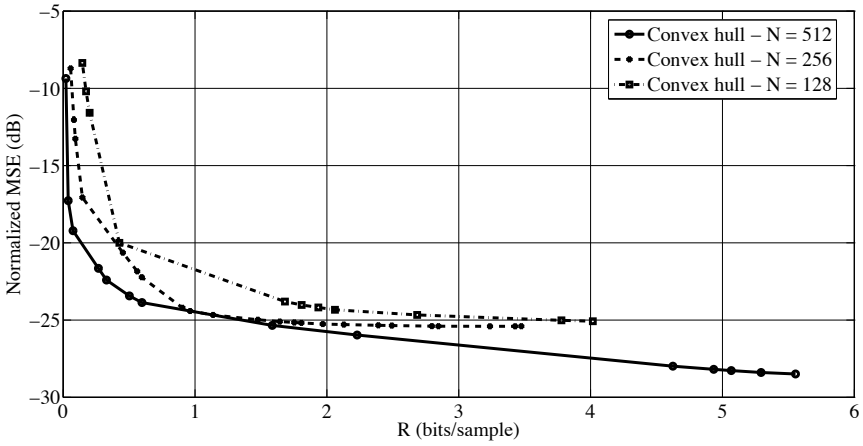

Figure 4.17: Rate–distortion convex hulls for the reconstruction of temperature signal with LASSO varying the length of monitored signal block (N).

4.5 An Energy Consumption Comparison Between DECA and CS in WSNs

Both DECA and CS are welcome strategies to be considered for energy conservation in WSNs, because they shift the computational complexity from the sensor nodes to the sink, as sensor nodes generally have memory and energy restrictions.

In this experiment, we consider that a WSN is monitoring a temperature signal. We also consider the same node positions of [18], and used for the evaluation of DECA in Chapter 3. A single sensor node, located in the position of node S_{47}, $(39.5, 14)$ in meters, collects temperature samples of the environment and transmits the measurements to the sink node, located in the $(0.5, 17)$ coordinates. We remind that these coordinates (in meters) are relative to the upper right corner of the lab. The other thirteen nodes are used as routers to forward the packets from the sensor node to the sink.

The TrueTime 1.5 is used to perform the simulations, the ZigBee standard is considered and the AODV routing protocol is used by routing nodes. The energy model is the same considered in Section 3.2.1. In each simulation run, the position of

the thirteen routers are drawn from the remaining twelve nodes (the original WSN in [18] has fifty–four nodes). We have performed 10 simulation runs.

We verified in previous simulations that 14 measurements are transmitted by the sensor node in order to reconstruct a block of 512 samples, with DECA when $\delta = 1\%$ is considered. Thus, these parameters (a signal block with $N = 512$ samples of a temperature signal and $M = 14$ coefficients) are used in the CS simulations. Figure 4.18 shows the MSE (in dB) of the reconstruction of $N = 512$ samples of a temperature signal, and the energy consumption (in Joules) of the sensor node (transmitter) after the transmission of the $M = 14$ measurements/coefficients. We consider that the transmitted measurements in DECA are densely quantized (as if these measurements had infinite precision), while we use quantizers with bit–depths (B) varying within the set $\{4, 6, 8, 10\}$ bits before transmissions in CS scheme.

As expected, the obtained results show that the sensor node spends less energy when running DECA. This is so, because DECA uses a simple linear predictor, while the CS requires more calculation (as inner products, for example). Moreover, in the Compressive Sensing strategy, the sensor node has to store the $N = 512$ samples of the monitored signal before the CS be applied, generating the $M = 14$ coefficients. On the other hand, there are specific recovery algorithms used in the CS scheme, while the sink uses a simple first–order interpolator to reconstruct the received signal in DECA. Thus, we also observed in Figure 4.18 that the CS provides a better reconstruction of the monitored signal (when more than 4 bits are used to quantize the transmitted coefficients), in this simulation scenario.

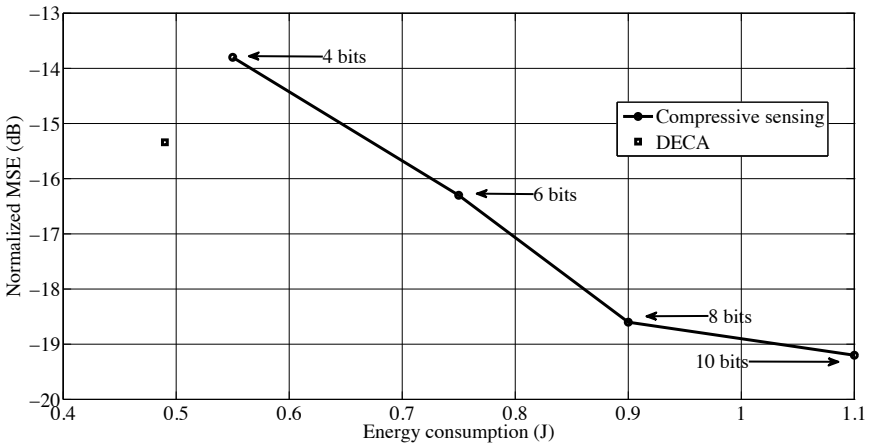

Figure 4.18: A comparison of the energy consumption between both DECA and CS schemes, applied to the reconstruction of $N = 512$ samples of a temperature signal, with 14 transmitted measurements.

4.6 Conclusions

In this chapter, the Compressive Sensing (CS) framework was applied to environmental data collection by Wireless Sensor Networks (WSNs). The rate–distortion analysis of the reconstruction of environmental signals was discussed, with a comparison among three distinct signal reconstruction methods. Some instabilities observed in the rate–distortion behavior for the L1–magic were investigated and the ϵ parameter was optimized in order to mitigate this effect. The influence of the length of the signal block of the monitored signal in the reconstruction was also studied and an example of reconstruction of temperature data was shown.

We observed that the L1–magic provided a better rate–distortion performance compared to the both A*OMP and LASSO. However, some instabilities were verified when the L1–magic was used to reconstruct the monitored signals. We also observed better rate–distortion results when the LASSO was considered, in comparison with the A*OMP algorithm. In the next chapter, the impact of packet loss in the rate–distortion performance of the reconstruction is evaluated.

An initial comparison between DECA and CS schemes was performed, considering a specific network scenario, and it was verified that DECA provided a lower energy consumption, while CS yielded a better reconstruction of the monitored signal.

Chapter 5

Robustness of CS Coded Data in WSNs

5.1 Introduction

In this chapter, the impact of packet losses in Compressive Sensing (CS) applied to a Wireless Sensor Network (WSN) is investigated. In this sense, one tries to evaluate how it affects the reconstruction of the monitored signal. In addition, the brunt of these losses when more than one measurement per packet is transmitted by sensor nodes is also evaluated. This analysis brings a more realistic WSN scenario, in which data packets can be lost, and extends the investigation performed in the previous chapter.

5.2 Problem Model

Sensor nodes compress a vector $\mathbf{x}_i \in \mathbb{R}^N$ containing the i–th block of contiguous samples of a signal, generating $\mathbf{y} \in \mathbb{R}^M$ measurements, with $M < N$. After quantization, sensor nodes transmit the $\mathbf{y}_{q_i}$ measurements, and the sink node receives $\mathbf{y}_{r_i}$ measurements. Previously, we investigated the rate–distortion performance when $\mathbf{y}_{q_i} = \mathbf{y}_{r_i}$, meaning that all transmitted packets are correctly received by the sink. However, in a real scenario, packet losses may occur (these are inherent to wireless channels [74] [75]). This means that $\#\mathbf{y}_{r_i} \leq \#\mathbf{y}_{q_i}$, in which $\#\mathbf{y}_{r_i}$ and $\#\mathbf{y}_{q_i}$ represent the quantity of received and transmitted measurements, respectively. In this section, we investigate the impact of packet losses in the CS–WSN strategy, and how does it affect the reconstruction of the monitored signal.

As defined in Section 4.4.1, a sensor node S_i transmits its quantized measurements

$$\mathbf{y}_{q_1} = [y_{q_1}[1], y_{q_1}[2], \ldots, y_{q_1}[M]]\,. \tag{5.1}$$

We consider that some transmitted coefficients are lost, because of the wireless channel of the WSN. Thus, the sink node receives L measurements, in which $L \leq M < N$, generating a vector

$$\mathbf{y}_{r_i} = [y_{r_i}[1], y_{r_i}[2], \ldots, y_{r_i}[M]]\,. \tag{5.2}$$

of which $M - L$ measurements are unknown due to packet losses.

If each entry of $y_{q_i}[k]$ $(k = 1, \ldots, M)$ is transmitted within a data packet, then a sensor node transmits M packets. This transmission strategy has the advantage that in the case some packet being lost then only one measurement is missed. Conversely, if all M measurements were transmitted within a single packet and this packet was lost, the whole signal would be missed.

Once transmitting a measurement within a packet, the number of lost packets is equal to the number of erased coefficients [76]. In [77], orthogonal projections are used by the transmitter, in order to compensate for erasures. However, authors in [77] consider a transmitter–aware scheme, in which the occurrence of erasures is known by the transmitter in order to project the erasure error and compensate for them. We highlight that, unlike [76] and [77], in this work the transmitter is not aware of losses, and the sink simply reconstructs the signal with the measurements that it receives from sensor nodes.

A simple way to mitigate transmission losses is by using *acknowledgments* (ACK) [78]. When a transmission is received, the sink may send an ACK, confirming the receipt of the involved measurement. If the sensor node receives the ACK, then the subsequent data can be sent. Otherwise, a retransmission is required. In this work, ACKs are not employed. We intend to reduce the amount of transmissions by sensor nodes, and to retransmit packets means to consume more energy.

Authors in [79] investigate the characteristics of lossy links in WSNs, and apply the CS framework to overcome this problem. Experimental results show that the WSN can transmit with high quality, while reducing energy consumption because of the CS characteristics, i.e., if there is a sufficient number of received packets and if the sink is capable of identifying which coefficients (packets) were lost.

In [80], for mitigating the impact of missing sensor data, authors propose an adaptive estimation algorithm. The spatial correlation among sensor data is used, and a regression model that considers information of neighboring nodes is applied to estimate data losses.

The IEEE 802.15.4 standard [54], referred to as ZigBee and widely used in WSNs

is considered for communication between nodes. In this standard, there are sequence numbers that are used to enumerate packet transmissions [81] [82]. These sequence numbers allow to identify lost frames in the link layer or to reorder segments in the transport layer. In the considered CS framework, the sequence numbers are used in the reconstruction, in order to identify which measurements were lost.

The rate–distortion performance (varying the bit–depths of the quantizers and the number of measurements transmitted by sensor nodes) is evaluated under several packet loss conditions. We also run experiments when considering that sensors transmit more than one CS measurement per packet.

5.3 Set-up and Results

We consider a WSN monitoring the temperature signal to evaluate the ad hoc WSN communication link. We consider the following percentages of packet loss: 0%, 10%, 20%, 30%, 40% and 50%. The 0% of packet loss means that all packets were received and, for example, the 10% of packets loss means that one in ten packets is lost. These percentages are applied to each combination of bit–depth (B) and number of coefficients (M). The dimension of the signal block is set to $N = 512$; the number of CS measurements used ranges from $M = 8$ to $M = 256$, in steps of 2; and bit–depths B vary within the set {4,5,6,7,8,9,10}. At the presented rate–distortion curves each point represents the average over 300 runs, each with a different seed.

Initially, we assume that each CS measurement is transmitted within a packet that has its corresponding sequence number. When reconstructing the signal, the sink knows which measurements were received and which measurements were lost. Lost measurements are then ignored by sink in signal reconstruction. At the end of this section, we evaluate the impact of transmitting more than one measurement per packet in the rate–distortion performance of the CS framework.

Figure 5.1 shows, along with the optimal rate-distortion curve (obtained from the convex hull of the rate distortion points), the curves resulting from each level of packet loss. Figures 5.2, 5.3, 5.4, 5.5 and 5.6 show the same information as in Figure 5.1, but comparing the original performance (0% packet loss) with the ones for each level of packet loss (10%, 20%, 30%, 40% and 50% packet loss), separately. We consider, in this experiment, the transmission of one single CS measurement per packet.

Table 5.1 presents the increase in the average Bjontegaard Delta (ΔBD) [83] of the Mean Squared Error (MSE) between the curves in Figure 5.1, in dB. The Bjontegaard Delta metric is a metric that is widely used in audio and video coding in order to evaluate average rate-distortion performance differences between two encoders [84] [85]. The computation of the ΔBD between curves takes place between:

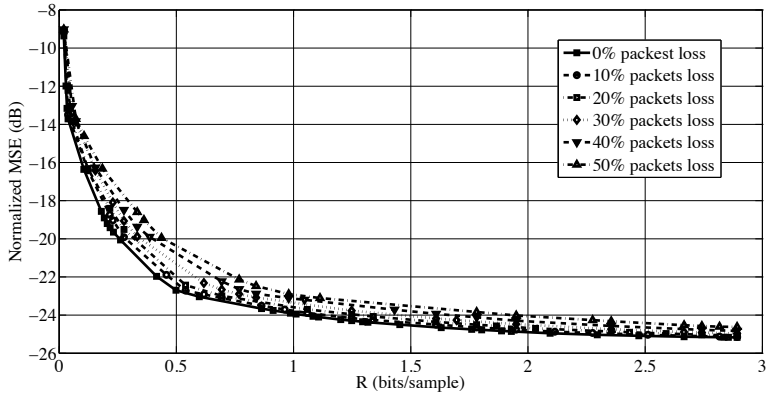

Figure 5.1: Rate–distortion convex hulls for each packet loss percentage for the reconstruction of $N = 512$ samples of a temperature signal with LASSO, considering the transmission of 1 CS measurement per packet.

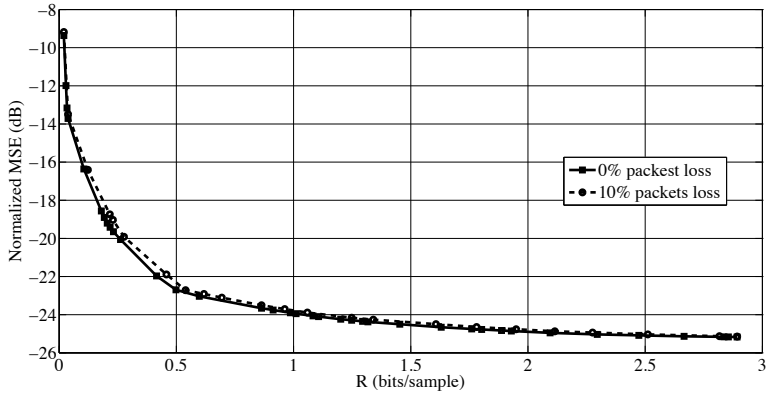

Figure 5.2: Rate–distortion convex hulls for 0% and 10% packet loss percentage for the reconstruction of $N = 512$ samples of a temperature signal with LASSO, considering the transmission of 1 CS measurement per packet.

0% and 10% packet loss; 0% and 20% packet loss; 0% and 30% packet loss; 0% and 40% packet loss; and 0% and 50% packet loss. Even for a large packet loss of 50%, we verified a small deterioration in the reconstructed signal with an MSE increase that is smaller than 2 dB.

One can see, from Figure 5.1 and Table 5.1, that the Compressive Sensing framework is robust towards packet loss (coefficient erasure). This result is significant from an energy–efficiency point of view. The CS democracy property states that all measurements contribute with the same amount of information [86]. The results corroborate it. If all CS measurements contribute with the same amount of information and if the sink knows the indices of the missing coefficients, then it is enough to ignore the losses when reconstructing the original signal, even if there are burst errors on the network.

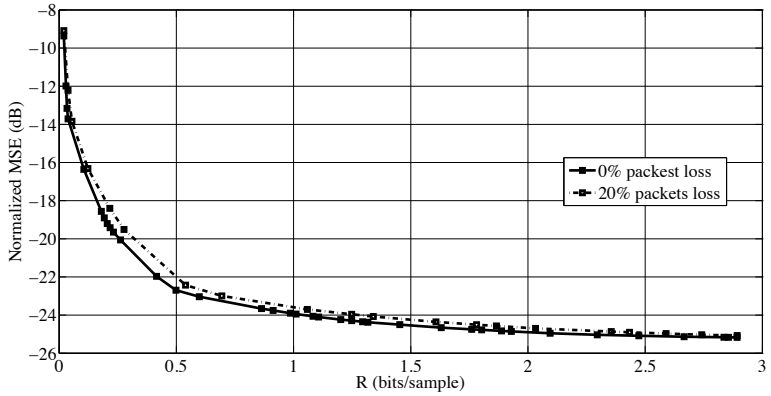

Figure 5.3: Rate–distortion convex hulls for 0% and 20% packet loss percentage for the reconstruction of $N = 512$ samples of a temperature signal with LASSO, considering the transmission of 1 CS measurement per packet.

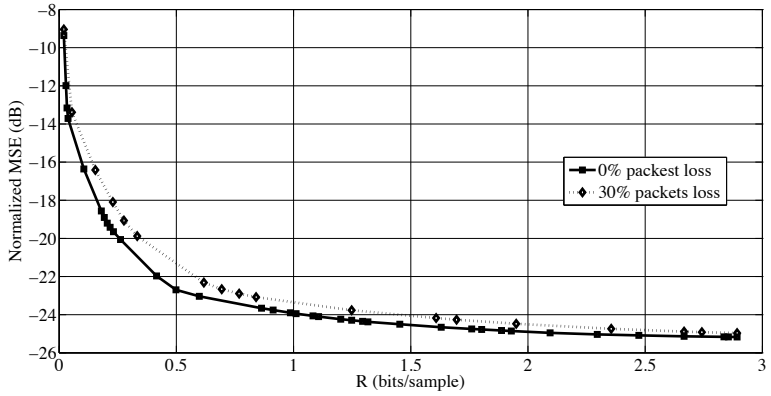

Figure 5.4: Rate–distortion convex hulls for 0% and 30% packet loss percentage for the reconstruction of $N = 512$ samples of a temperature signal with LASSO, considering the transmission of 1 CS measurement per packet.

Table 5.1: ΔBD between RD curves for each packet loss percentage for the reconstruction of $N = 512$ samples of a temperature signal with LASSO, considering the transmission of 1 CS measurement per packet.

Percentage	ΔBD (dB)
0%–10%	0.36
0%–20%	0.64
0%–30%	1.07
0%–40%	1.30
0%–50%	1.73

Finally, in Figure 5.7 one redraws the RD curves shown in Figure 5.1 (resulting from some levels of packet losses), considering the effective rates (R_{eff}). For example,

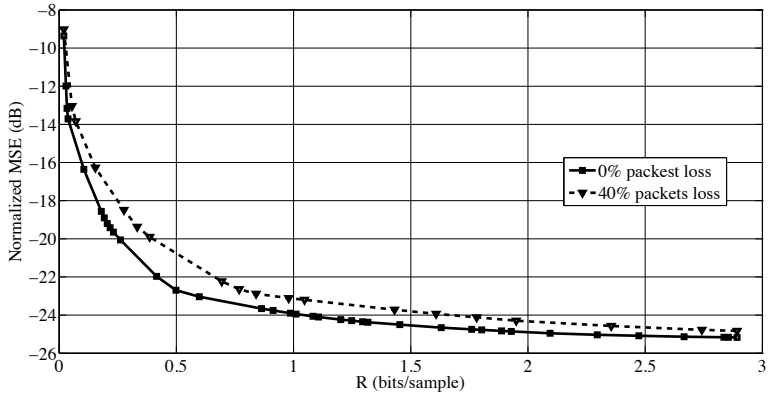

Figure 5.5: Rate–distortion convex hulls for 0% and 40% packet loss percentage for the reconstruction of $N = 512$ samples of a temperature signal with LASSO, considering the transmission of 1 CS measurement per packet.

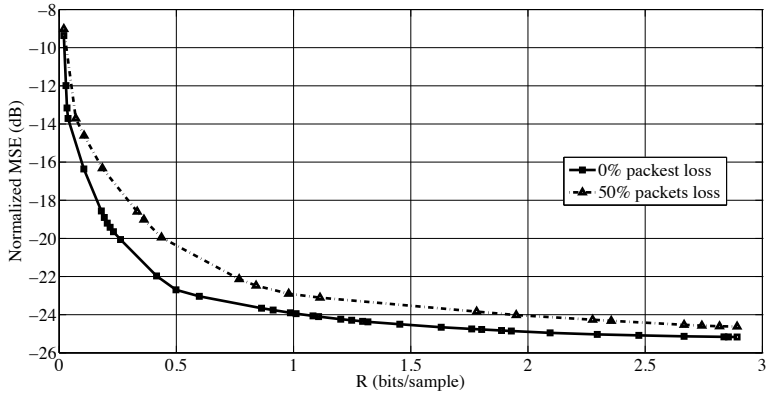

Figure 5.6: Rate–distortion convex hulls for 0% and 50% packet loss percentage for the reconstruction of $N = 512$ samples of a temperature signal with LASSO, considering the transmission of 1 CS measurement per packet.

if a node transmits 1 bit/sample and 50% of packets are lost, there is an effective rate of 0.5 bit/sample arriving at the sink node. This confirms the expected result that the RD performances as a function of R_{eff} closely match the RD performance without packet loss.

5.4 Evaluation of the Influence of the Amount of Transmitted Information

The above experiments considered sensor nodes transmitting a single CS measurement per packet. If a packet is lost, then just one coefficient is missed. However, if sensor nodes transmit only one measurement per packet, then the wireless channel

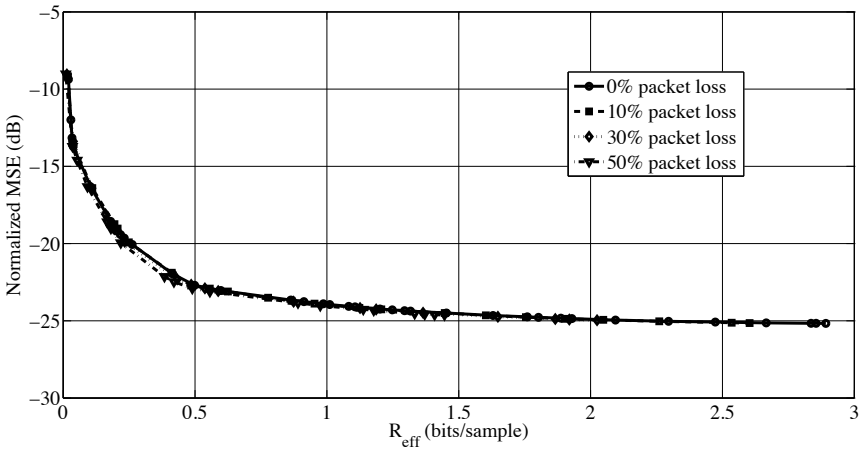

Figure 5.7: Rate–distortion convex hulls for each packet loss percentage for the reconstruction of $N = 512$ samples of a temperature signal with LASSO, considering the effective arrival rates when nodes transmit 1 CS measurement per packet.

may be underused. There may be a trade–off between the packet loss impact on the rate–distortion performance and the wireless channel usage.

One starts this study considering that sensor nodes transmit more than one CS measurement per packet and a single packet is lost. Then a worsening in the distortion is expected when sensor nodes transmit packets with a large amount of information. This awaited behavior can be observed in Figure 5.8, that presents the convex hulls for the rate–distortion curves of the reconstruction of a temperature signal with LASSO. For this experiment, it is consider the transmission of 2, 4, 8, 16 and 32 measurements per packet by sensor nodes, and one can verify a greater impact on distortion when the lost packet carries more measurements. It is also observed that this impact worsens for smaller rates. This is so, because the percentage loss of information is greater for lower rates. For example, if a sensor node has to transmit 16 CS measurements with 2 measurements per packet and one packet is lost, then 12.5% of the information to be transmitted is lost. On the other hand, if this node has to transmit 100 measurements with 2 measurements per packet and one packet is lost, then only 2% of the information is lost.

Now, instead of assuming that the sensor nodes have lost a single packet, a percentage of packet loss is assigned to the wireless communication channel, as presented in the experiments discussed in the Section 5.3. Figure 5.9 shows the rate–distortion behavior of the reconstruction of a temperature signal with LASSO, for 10% of packet loss. To evaluate the trade–off between the packet loss impact on the rate–distortion performance and the wireless channel usage, it is also consider the transmission of 2, 4, 8, 16 and 32 measurements per packet by sensor nodes. One can notice the impact of the number of measurements per packet in the rate–

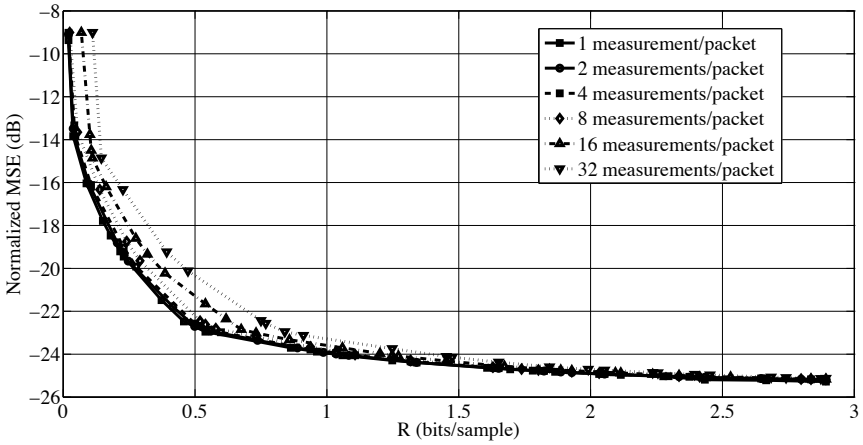

Figure 5.8: Rate–distortion convex hulls of the rate–distortion curves for the reconstruction of a temperature signal with LASSO varying the amount of CS measurements per packet.

distortion performance, when each transmission carries more than 4 coefficients. As expected, if there is a larger number of measurements per packet, then any loss leads to a worsening in the reconstruction of the original signal.

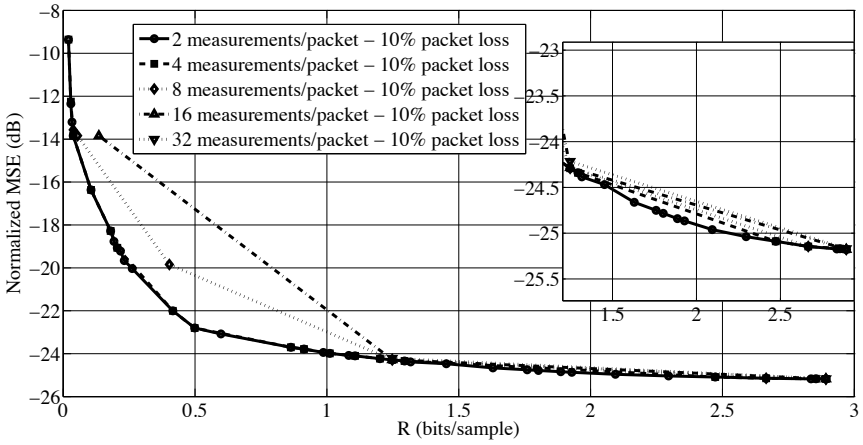

Figure 5.9: Rate–distortion curves for reconstruction of temperature signal with LASSO, varying the number of measurements per packet with 10% packet loss, along with its details.

5.5 Conclusions

In this chapter, we evaluated the impact of packet losses in the RD performance of the reconstruction of a monitored signal, when a WSN is used to sense environmental data and the CS framework is used to code data to be transmitted by sensor

nodes. The Bjontegaard Delta metric was used to compare the RD curves with some levels of packet losses and the RD curve without packet loss and we observed, as expected, that CS framework is robust against coefficient erasures. In the next chapter, we provide ways for sensor nodes perform transmissions incrementally, using a successive approximation scheme. Doing so, we intend to save sensor nodes energy, while keeping the rate–distortion performance near to the optimal one (operational curve).

Chapter 6

Successive Approximation of CS Coded Data in WSNs

6.1 Introduction

So far we have seen that it is possible to use CS for coding measurements taken by WSN nodes within a desired RD performance. However, it could be the case that one requires a given sensor to transmit extra bits for improving the reconstructed signal at the sink node. This is what we address now. How do we setup an incremental transmission framework for CS-WSN that guarantees good RD performance while saving sensor energy? Note that the incremental feature denotes that one wants to continue using the quantized measurements that are already known by the sink in the reconstruction and to transmit only extra-bits, improving signal reconstruction while keeping RD performance as good as possible.

6.2 Problem Model

An incremental scheme allows to save sensor nodes energy because nodes only need to transmit refinements of previously transmitted measurements and/or new measurements. That is, any extra bit transmitted by nodes is used in the reconstruction of the signal and not as substitute of previously known values. We propose for nodes to incrementally transmit quantized CS measurements by using a successive approximation scheme. Suppose that a node has already transmitted M_i CS measurements, each one of them quantized with B_i bits. We refer to such a setting as the operational setup point (M_i, B_i) in the $M \times B$ space. In subsequent transmissions, a node can send remaining measurements (the ones not transmitted yet), or quantization refinements (increasing the quantization bit–depth, then reducing the measurement quantization error), or both. The selected increment must improve RD performance

while remaining as close as possible to the optimal RD points. In any case, the rate increment corresponds to the selection of a new setup point $(M_i + \Delta M_i, B_i + \Delta B_i)$, where ΔM_i and ΔB_i are the increments in the quantity of measurements and in the bit–depth, respectively.

Let a given node code a set of M_i measurements using B_i bits, transmitting these, spending a total rate R_i. Suppose now that the sink node (decoder) needs more information, i.e., it requires an increment in rate (ΔR_i), in order to generate a better version of the reconstructed signal. ΔB_i and ΔM_i increase the rate by

$$\Delta R_i = M_i \Delta B_i + \Delta M_i B_i + \Delta M_i \Delta B_i, \tag{6.1}$$

in which M_i and B_i are known, and ΔR_i may be seen as the target.

Obviously, it is desirable that the sequence of incremental transmissions does provide a monotonically decreasing path along the RD curve. At each incremental transmission, the coder should reduce distortion, and the next (M_j, B_j) should have to satisfy

$$(M_j, B_j) = \min_{k \in \mathcal{K}|_{M_i, B_i}} \mathcal{D}\left(M_k, B_k\right), \tag{6.2}$$

in which $\mathcal{D}(.)$ is the distortion function, $\mathcal{K}$ is the set of new candidate pairs, and the $k \in \mathcal{K}$ defines the possible shifts in the operational setup, i.e., from (M_i, B_i) to (M_j, B_j) corresponding to a given increment in rate.

For this experiment, Figure 6.1 is presented. This figure shows the convex hull of the rate–distortion curves of the reconstruction of the temperature signal, considering the number of measurements ranging from $M = 8$ to $M = 256$, in increments of 2, quantized bit–depths values (B) varying within the set {4,5,6,7,8,9,10,11,12,13} and a signal block of length $N = 512$. The LASSO method is considered in the experiment. Furthermore, Figure 6.2 presents the variation of the optimal quantizers, as function of the bit rate, considering these reconstruction levels. Each (M_i, B_i) pair represents the best tradeoff between rate and distortion.

To better investigate this problem, a large enough number of points in the rate–distortion curves had to be used. For each (M_i, B_i) pair of the convex hull from Figure 6.1, we varied B_i from $\frac{1}{M_i}$ to $\frac{1}{M_i}$ (around B_i), generating other $2M_i$ values of B. The procedure to increase the granularity of B, is shown in **Algorithm 2**, for each (B_i, M_i).

ALGORITHM 2: Increasing the granularity of B.

```
for k ← 1 to 2M_i do
    B_ia(k) = B_i − 1 + k/M_i;
end
```

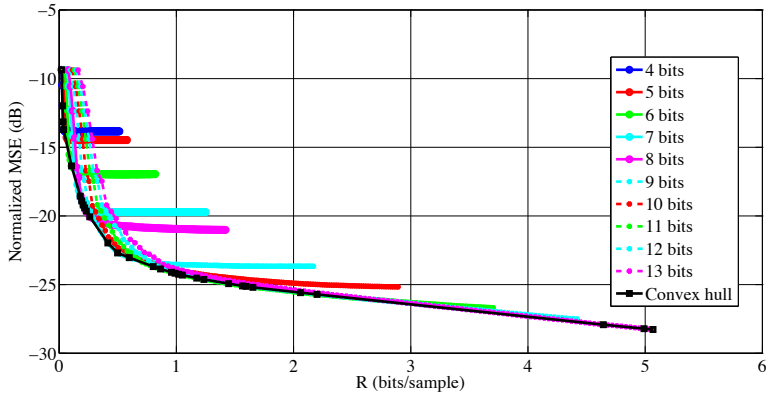

Figure 6.1: Rate–distortion curves and the convex hull for the reconstruction of $N = 512$ samples for a temperature signal considering more bit–depths.

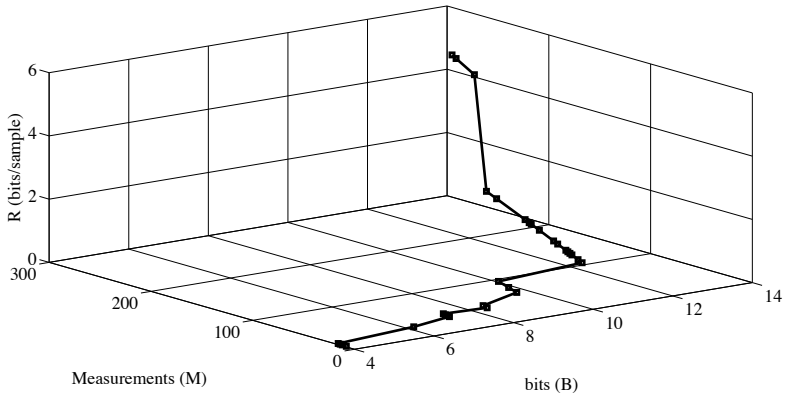

Figure 6.2: Optimal quantizers variation as a function of rate for the reconstruction of $N = 512$ samples for a temperature signal considering more bit–depths.

Figure 6.3 shows the rate–distortion convex hull for the reconstruction of $N = 512$ samples of a temperature signal with LASSO generated with the combinations between B_i and M_i with a larger granularity. The number of measurements ranges from $M = 8$ to $M = 256$, in increments of two; to the bit depths B belonging to the set {4,5,6,7,8,9,10,11,12,13} we have added others generated by **Algorithm 2**. Figure 6.4 presents the optimal coding path on the $(M \times B)$ plane along this convex hull, along with the (M_j, B_j) pairs that have the better trade–of between rate and distortion. The LASSO method is considered in the experiment.

We consider the use of a successive approximation scheme for sensor nodes to incrementally transmit the CS measurements, in which coder moves along (M_i, B_i) pairs keeping RD performance very close to the optimal. This makes it possible to have incremental transmissions while saving as much as possible sensor node energy. The proposed scheme is detailed hereafter.

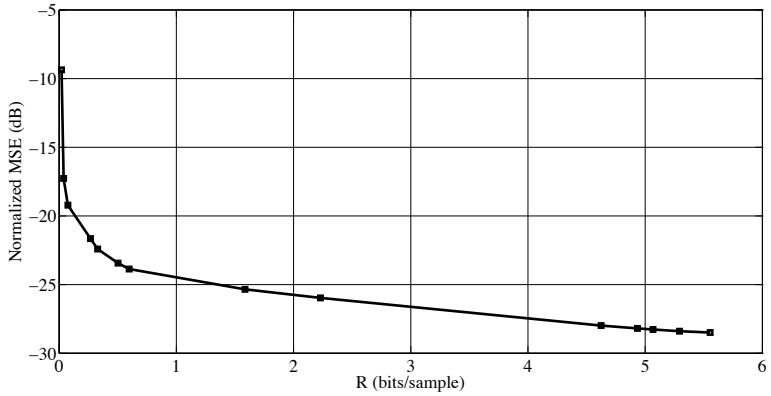

Figure 6.3: Rate–distortion convex hull for the reconstruction of $N = 512$ samples of a temperature signal when more combinations of M and B are used.

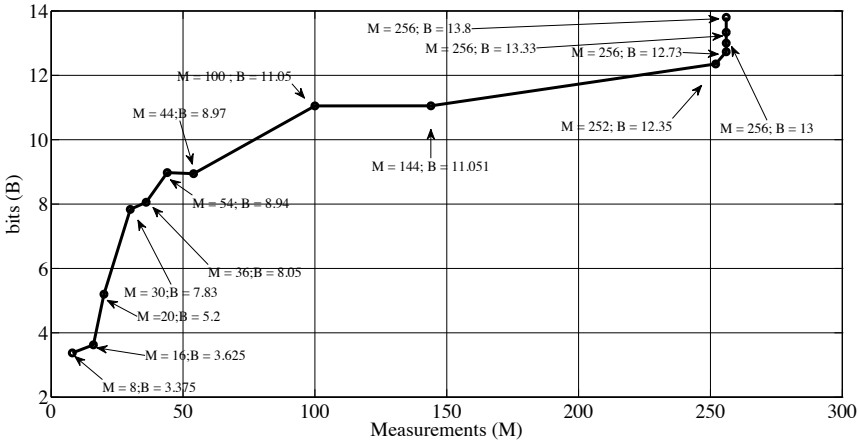

Figure 6.4: Optimal coding path on the $M \times B$ plane along the RD convex hull for the reconstruction of a temperature signal with LASSO.

6.3 Moving Along (M, B) Pairs

A special and restrictive case would be to consider that the increments (ΔM_i and ΔB_i) do not occur simultaneously. At each (M_i, B_i) pair, there would be two possibilities: incrementing the number of measurements, keeping the bit–depth unaltered, or refining the quantization, keeping the number of measurements unaltered. This corresponds to a path along either horizontal or vertical directions on the $M \times B$ plane. Sensor nodes shall choose the directions with lower distortion, based on eq. (6.2). For each increment in rate, as there are only two possible directions, $\mathcal{K} = 2$ in eq. (6.2).

In the more general case, instead of being restricted to either horizontal or vertical directions on the $M \times B$ plane, both M and B could be varied together in order to provide the desired increment in rate. Therefore, starting from a given (M_i, B_i)

setup corresponding to an RD point in the convex hull, the coder should move to a new (M_j, B_j) pair leading to the minimal distortion for the resulting increment in the rate.

Figure 6.5 presents an example of the pairs generated with the successive approximation scheme. In this example, one starts from a given (M_i, B_i) operational point. Then, the first curve of candidate pairs is generated, varying both M_i and B_i together. This procedure is done iteratively at each new refinement, thus a new curve is generated and another operational setup is chosen. In each operational point shift there is an increase as in eq (6.1) and the new point should be selected satisfying eq. (6.2).

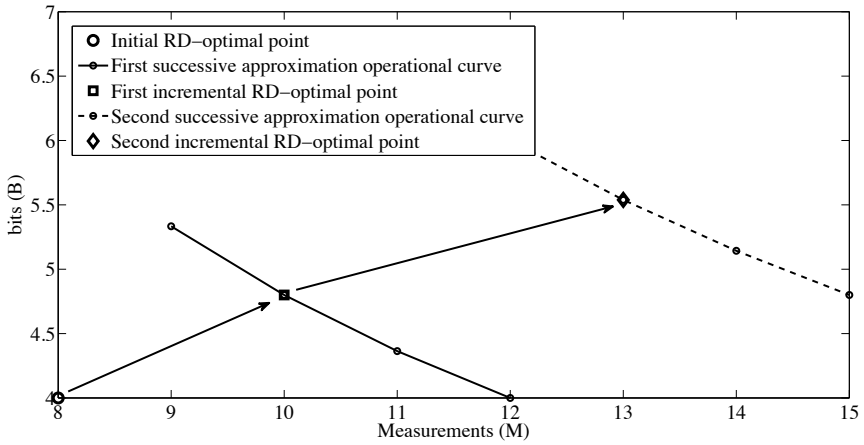

Figure 6.5: Example of the successive approximation scheme, varying both B and M together.

However, the above procedure can not be performed online, since sensor nodes energy would be exhausted due to the extensive RD points computations required. Therefore, we propose using a successive approximation scheme close to the optimal one that relies on previously analyzing RD performance, that is, on offline computations. We use, for each sensor, a training set composed of N previously recorded samples. Using these measurements, we generate a table with the path on the $M \times B$ plane to be followed by the coder. Sensor nodes must have this pairs stored before effectively starting transmitting the measurements. The selected pair, which defines the allocation of the refinement bits, is the one that leads to the minimal distortion. The sequence of (M, B) points to be followed could be informed by the sink to sensor nodes. This approach would reduce sensor nodes memory requirements and marginally increase the data sent by the sink when asking for more data for signal reconstruction.

The initial RD–optimal setup point (M_0, B_0) is defined from the initial rate. The M_0 measurements are quantized and transmitted with B_0 bits. If there is room for

more bits, then a new RD–optimal point is to be used. The new point (M_1, B_1) is selected from the operational setup table according to the desired rate increase. That is, for the operational point (M_i, B_i) the table contains several possible operational setups to use, the one to used (M_{i+1}, B_{i+1}) depends on the actual increase in rate desired. Then the sensor sends the $M_{i+1} - M_i$ extra measurements quantized with B_{i+1} bits and the extra $B_{i+1} - B_i$ bits refining the already transmitted M_i measurements.

6.4 Successive Approximation Scheme: Table Construction

Given that M_i coefficients have been already transmitted, one can define the set of new possible quantities of CS coefficients as $\{M_k\}_{k\in\mathcal{K}}$, $M_k > M_i$. From eq. (6.1) one can compute the corresponding possible bit–depths $\{B_k\}_{k\in\mathcal{K}}$, for a predefined target increment in rate ΔR_i. Dividing both sides of eq. (6.1) by $\Delta M_i = M_k - M_i$, we have

$$\frac{\Delta R_i}{\Delta M_i} = \frac{M_i \Delta B_i}{\Delta M_i} + B_i + \Delta B_i, \tag{6.3}$$

yielding

$$\Delta B_i = \frac{\frac{\Delta R_i}{\Delta M_i} - B_i}{1 + \frac{M_i}{\Delta M_i}}. \tag{6.4}$$

From the previous discussion, we want to build a table that at a given RD–optimal setup (M_i, B_i), and for a given increment in rate ΔR_i, provides the increments ΔM_i and ΔB_i to the next RD–optimal point. The sequence of steps for constructing the successive approximation table to be used by sensor node S_i is presented in **Algorithm 3**. In this procedure, (M_i, B_i) is the current RD–optimal pair of node S_i; d_R is the percent increment in the rate; c_n is the amount of (M_i, B_i) points generated in each operational curve (Figure 6.5); d_N is a factor that limits the amount of operational curves that can be generated. The generation of the candidate pairs–(M_j, B_j)–is shown in lines 7 to 9, and the choice of the new pair depends on minimizing the incurred distortion as presented in line 11. All c_n points are tested for each (M_k, B_k) setup, and one stores that c_n point leading to the minimum for eq. (6.2), for the target increase in rate ΔR_i.

The complexity of the table construction is mainly limited by d_R and c_n. At each operational point, the d_R factor defines a rate increment. c_n defines the number of possible operational points to be tested at a given (M_i, B_i) to chose the next operational point $(M_{i+1}, B_{i+1}) = (M_i + \Delta M_i, B_i + \Delta B_i)$. In other words, c_n defines

the amount of candidate setups for the given percent increment in rate d_R. Note that the larger c_n is, the more trials are required. Now suppose that one starts at a rate R_i and in the usage of the WSN one wants increment it to $R_i + \Delta R_i$. As d_R increases, the sensor traverses less operational points to go from a rate R_i to a rate $R_i + \Delta R_i$. Therefore, d_R restricts the amount of generated curves that have to be tested. The smaller that d_R is the more precise the rate can be adjusted, at expenses of a more complex training process.

The overall complexity comes from the construction the table of (M, B) setups defining the successive approximation path. However, as mentioned above, this does not impact the energy consumption of the sensor nodes because it is performed offline. Sensor nodes need to be either previously aware of the table or to be informed of each new operational setup. This provides an "incremental path" for encoding and transmission or successive approximation of CS quantized measurements.

ALGORITHM 3: Construction of the successive approximation table.

$M_j \leftarrow M_i$;
$B_j \leftarrow B_i$;
$\Delta M_j \leftarrow [1 : c_n]$;
while $M_j \leq N d_N$ **do**
    $\Delta R_j \leftarrow M_j B_j d_R$;
    for $n \leftarrow 1$ **to** c_n **do**
        $M_k[n] \leftarrow M_j + \Delta M_j[n]$;
        $\Delta B_j[n] \leftarrow \frac{\frac{\Delta R_j}{\Delta M_j[n]} - B_j}{1 + \frac{M_k[n]}{\Delta M_j[n]}}$;
        $B_k[n] \leftarrow B_j + \Delta B_j[n]$;
    end
    $(M_j, B_j) \leftarrow \min_{k \in \mathcal{K}} \mathcal{D}\left(M_k[n], B_k[n]\right)$;
end

As an illustration of the procedure for building the successive approximation table, we starts from an initial operational point $(M_0 = 8, B_0 = 4)$. For the next point (the first refinement point), consider $c_n = 10$, i.e., there are at most ten (M, B) pairs in the next generated operational curve, each one with a respective distortion, based on a fixed rate increment defined by d_R. Then, each one of the operational points are tested and the one that leads to the minimal distortion is chosen, based on eq. (6.2). As a result the subsequent operational point is selected $(M_1 = 10, B_1 = 5)$. After that, a second operational curve is now generated for the same fraction of rate increase and tested, and the configuration incurring the smaller distortion is chosen, for example, it may be $(M_2 = 13, B_2 = 6)$. The above procedure is iterated until $M \leq N d_N$. This last criteria imposes a maximum number of measurements $(d_N < 1)$. This is just a way to inflict a limiter on the number of transmitted measurements for the N–lengthed signal block, on its sparsity.

Figure 6.6 illustrates the transmitted data within the successive approximation coding method. Sensor node starts transmitting M_i measurements, each one quantized with B_i bits. Thus, we have the quantities M_i and B_i, and the bit stream with the M_i CS measurements. After that, it moves to the next point, $(M_i + \Delta M_j, B_i + \Delta B_j)$. Sensor node transmits the quantities ΔM_j and ΔB_j, the bit stream with the remaining CS measurements (ΔM_j) quantized with B_j bits, and quantization refinements of the previously transmitted measurements. Next, sensor node transmits ΔM_k and ΔB_k, the bit stream with the last CS measurements (ΔM_k) quantized with B_k bits, and quantization refinements of the previously transmitted measurements.

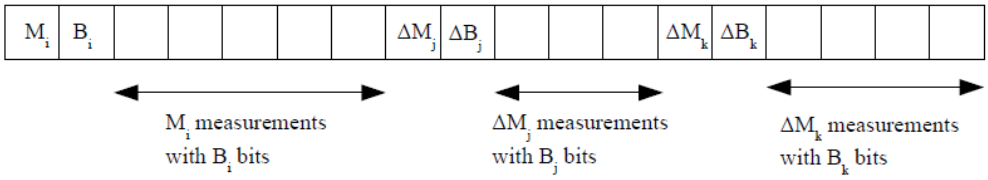

Figure 6.6: Necessary data within the successive approximation scheme.

6.5 Simulation Results

We consider the reconstruction of temperature, humidity and illumination signals, using a block length of $N = 512$ samples and LASSO for reconstruction. From each (M_i, B_i) point, we consider rates increments of 10%, i.e., $d_R = 0.1$. We also impose $d_N = 0.5$ $(M \leq \frac{N}{2})$ and $c_n = 10$. The Bjontegaard Delta metric (ΔBD) is used to evaluate the RD performance of the successive approximation scheme with respect to the exhaustively generated convex hull. Note that, for the simulations, the sensor output samples used to evaluate the RD performance are different from the ones used for computing the successive approximation table, as described in Section 6.3.

Figure 6.7 shows the paths of the operating points (M, B) that provide the RD convex hull and the best RD performance with the successive approximation scheme for a temperature signal. The correspondent rate–distortion curves are shown in Figure 6.8, the Bjontegaard Delta between them is $\Delta BD = 0.32$ dB. It can be observed that when we use the successive approximation scheme, the resulting operational RD curve closely matches the RD convex hull. We also verify that the curve for the incremental transmission scheme has more RD points than the convex hull does, meaning that it allows a finer adjustment of the rate.

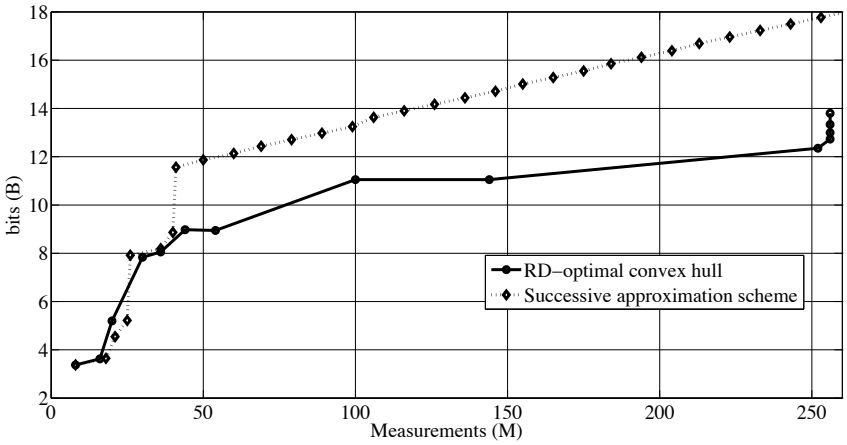

Figure 6.7: (M, B) path used by sensor node in the successive approximation scheme, for the reconstruction of a temperature signal with LASSO.

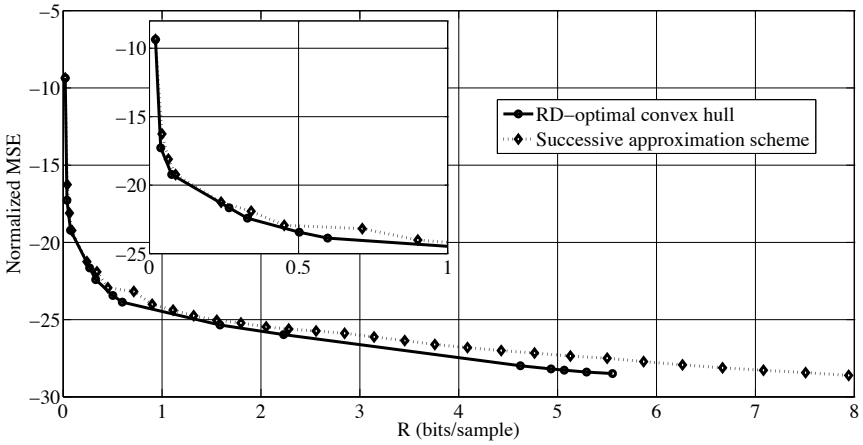

Figure 6.8: RD–optimal convex hull and rate–distortion curves for the successive approximation scheme, for the reconstruction of a temperature signal with LASSO, along with its detail between 0 and 1 bit/sample. The block length N is equal to 512.

The same experiment is performed with humidity signal. As we did for temperature signal, we consider the reconstruction of a $N = 512$ samples of signal block of the original humidity signal. LASSO is used for reconstruction, and we increase rate in increment of 10%.

Figure 6.9 shows the (M, B) path of the convex hull for humidity signal and the optimal path of the successive approximation scheme varying both M and B together at each point. There is a superior bound in the bit–depths in 10 bits. This bound exists because we considered values for the bit–depths varying within the set $B = \{4, 5, 6, 7, 8, 9, 10\}$ to generate the convex hull for the humidity signal. It can be observed in Figure 6.9 that the successive approximation scheme closely tracks

the convex hull. The rate–distortion curves for the successive approximation scheme is presented in Figure 6.10. It can be seen, from the obtained results, that when the proposed scheme is used, the rate–distortion curve closely reaches the convex hull. A value of $\Delta BD = 0.72$ dB is calculated by using the Bjontegaard Delta metric, in order to compare the RD curve of the proposed incremental transmission scheme with respect to the operational curve for the humidity signal.

The previous behavior is also observed for the illumination signal, as shown in Figures 6.11 and 6.12. The successive approximation varying both CS measurements quantity and bit–depths presents a rate–distortion curve that closely matches the attainable rate–distortion convex hull. We also calculated the Bjontegaard Delta in order to compare the RD performances of the successive approximation scheme and the convex hull for the illumination signal, and a value of $\Delta BD = 0.67$ dB was found.

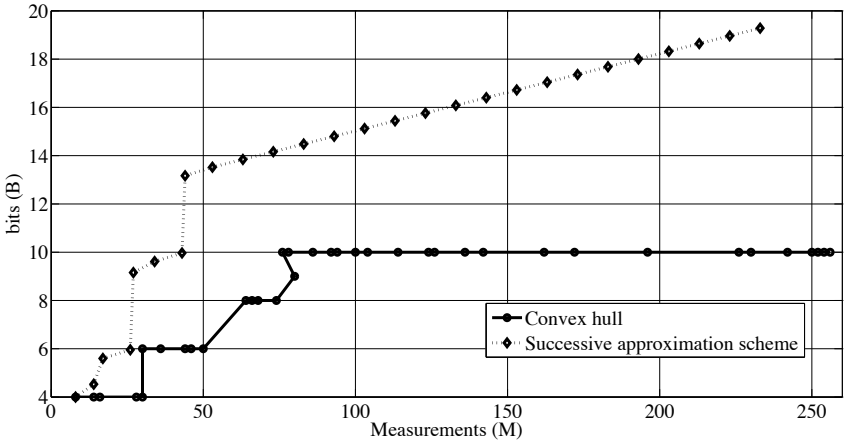

Figure 6.9: (M, B) path used by sensor node in the successive approximation scheme, for the reconstruction of a humidity signal with LASSO.

This shows that the proposed successive approximation CS coding provides a good compromise for all three signals considered.

6.6 Conclusions

In this chapter, we propose a successive approximation scheme that allows sensor nodes to incrementally transmit CS measurements. The scheme was employed in a WSN monitoring application, in which nodes measure environmental data (temperature, humidity and illumination). We observed that the proposed scheme closely follows the best attainable operational curve for the reconstruction of the considered monitored signals.

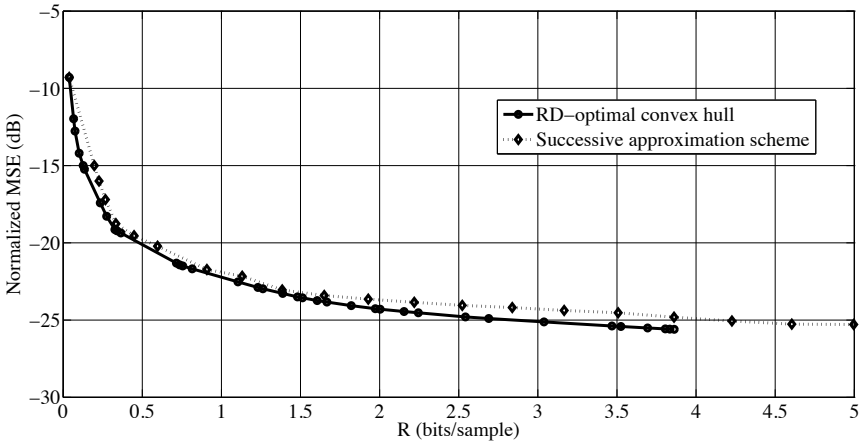

Figure 6.10: RD–optimal convex hull and rate–distortion curves for the successive approximation scheme, for the reconstruction of a humidity signal with LASSO. The block length N is equal to 512.

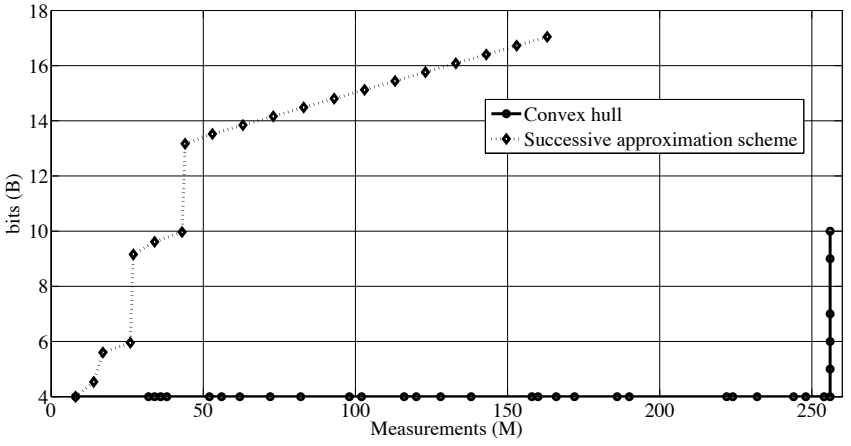

Figure 6.11: (M, B) path used by sensor node in the successive approximation scheme, for the reconstruction of an illumination signal with LASSO.

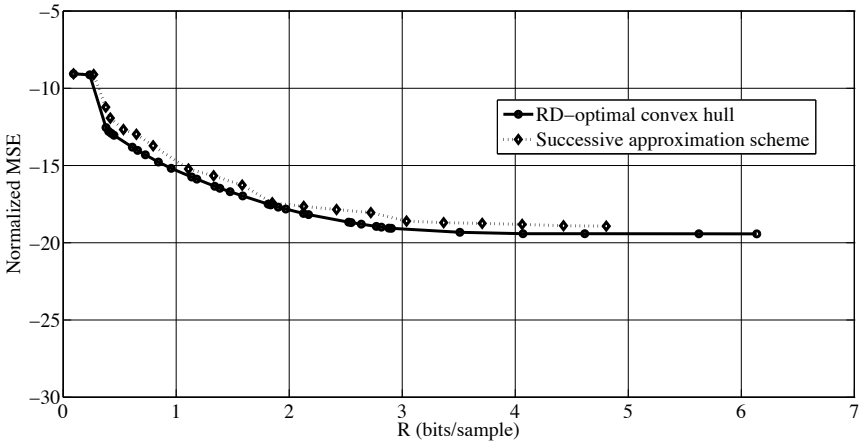

Figure 6.12: RD–optimal convex hull and rate–distortion curves for the successive approximation scheme, for the reconstruction of a illumination signal with LASSO. The block length N is equal to 512.

Chapter 7

Conclusions, Future Directions and List of Publications

In this work a WSN is employed to sense real environmental signals: temperature, humidity and illumination. The main objective of this work is to increase the autonomy of the network, saving energy of its sensor nodes. Thus, two frameworks were proposed: the first one is based on a sleep/wakeup protocol, and the second uses the paradigm of Compressive Sensing (CS) to reduce the amount of transmissions by sensor nodes, then saving their energy.

7.1 A Brief Comparison Between DECA and CS in WSNs

Some general comments comparing both DECA and CS are presented hereafter:

- Both strategies provide a reduction in the amount of transmissions by sensor nodes, thus improving the autonomy of the WSN;
- Both strategies can be applied in WSNs in application layer, regardless to existent routing protocols;
- DECA considers a simple linear predictor to estimate samples to be transmitted, while in CS the measurements to be transmitted are coded;
- In DECA, the nodes are put in the inactive state between transmissions, based on the variation rate of the monitored signal. In CS strategy, nodes store a block of N samples of the monitored signal, then the CS framework is applied to generate $M < N$ CS coded measurements. These M coded measurements are then transmitted, and nodes do not necessarily enter in the inactive state;

- The sink node considers a first order interpolator to reconstruct the monitored signal in DECA, while some specific recovery algorithms (as the Newton with log–barrier, in L1–magic; A*OMP; or LASSO) are used by sink to reconstruct the monitored signal in the CS coded framework;

- DECA provides a small packet loss, while the CS framework is robust against packet loss.

Simulations were performed considering a specific network scenario, in which $M = 14$ measurements/coefficients are transmitted and a block of $N = 512$ samples of a temperature monitored signal is reconstructed at the sink node. We observed the MSE of the reconstruction of the sensed signal by the WSN and the energy consumption of the sensor node that collected and transmitted these measurements/coefficients after the fourteenth transmission, for both DECA and CS strategies. As DECA uses a simple linear predictor and a first–order interpolator, in sensor and sink nodes respectively, the sensor node has spent less energy by using this algorithm. Furthermore, a better reconstruction of the monitored signal was verified with CS, as specific recovery procedures are used in the Compressive Sensing framework.

The conclusions for each one of these methods (DECA and CS) will be presented hereafter, together with some discussions about future directions.

7.2 Conclusions on DECA

We proposed a Distributed Energy Conservation Algorithm (DECA) for WSNs, in order to increase the network autonomy in monitoring applications. DECA uses data about the sensed process that is available at a node, both the last collected data and the last one sent to the sink, to estimate measured values and sleeping periods. By design, DECA imposes the reconstruction error of the monitored process to be within a predefined range. This is done by using linear predictors, and these allow to easily find bounds on the sleeping periods for each node.

In DECA, the decision for how long a node can be put to sleep is taken locally by each sensor node. That is, it is distributed and decentralized. To consider the possibility of a node to route packets from neighboring nodes, nodes transmit their sleeping periods together with the monitored process data so that routers can account for that in their decisions for how long they will be inactive. A sleeping period factor is used in DECA to increase the probability of a routing node to be awake to forward packets from its neighbors.

Network relay (as performed by routing nodes) is a key challenge for energy conservation is WSN. In order to investigate if relay nodes manage their routing

and sleeping tasks appropriately when running DECA, we proposed to evaluate the success ratio, which accounts for network connectivity in a micro perspective (locally). We should notice that DECA can be used together with other energy-aware routing algorithms to further improve the WSN autonomy.

DECA was evaluated by simulation using actual signals. The reconstruction error, the transmission decrease, and the network lifetime were obtained to evaluate DECA performance. Results corroborate that: i) the error of the reconstructed process at the sink is kept within the desired fraction imposed; and also ii) that network lifetime is increased as compared to WSNs were the nodes simply sleep periodically. The results indicate that for larger networks, the network lifetime increases more. The larger the network, the more possible the emergence of energy balancing, as there are more relays for forwarding packets from sources to the sink. In addition, we observed a decrease in both success ratio and packet delivery ratio with the increase of the sleeping factor and also with the increase of the amount of neighboring nodes. At last, we observed that as the sleeping factor increases, energy conservation also increases, because nodes can sleep for larger periods of time.

7.3 Conclusions on CS Coded Data in WSNs

In a distinct approach from DECA, the Compressive Sensing framework is considered for saving sensor nodes energy. This technique aims at acquiring a given signal at its information rate, in many cases, smaller than the *Nyquist* limit. Then, after sensor nodes collect N samples from the environment, $M < N$ measurement coefficients are generated. As a consequence, sensor nodes make less transmissions, saving energy. After receiving the M measurements, the sink node recovers the original signal with some convex optimization algorithm.

We considered the quantization of measurements before their transmission. Rate–distortion analysis took place in this scenario, evaluating three distinct reconstruction methods. We performed simulations with real temperature, humidity and illumination signals, varying the values of CS measurements and bit–depths (M and B, respectively). We compared the three recovery methods in terms of of their rate–distortion behavior in the reconstruction of the three signals. We also evaluated the impact of packet losses in the rate–distortion performance of the monitored signal and observed, as expected, that CS framework is robust against packet losses. Moreover, we verified that may be a trade–off between the packet loss impact on the rate–distortion performance and the channel usage, varying the amount of measurements per packet.

Finally, we proposed a successive approximation scheme, that allows sensor nodes to incrementally transmit CS measurements. We observed that the proposed scheme

closely follows the best attainable operational curve for the reconstruction of the considered monitored signals.

7.4 Future Directions

For DECA, we intend to generalize the algorithm for a multi–signal application. In this scenario, a node may collect data from several sensors, like temperature and humidity. Both values can be transmitted in the same packet. With two signals, the sensor node performs two predictions, and calculates two inactivity periods, chosen the smaller and more restrictive one. The sink receives both signals and reconstructs them.

For the CS coded data scheme and considering multiple hops with routing protocol, some data fusion technique can be applied in relays nodes, performing aggregation between the data sensed by the relay and by its neighbors. This may lead to the reduction in the amount of transmissions in the network.

The democracy property can be explored to predict some of the lost packets within the transitions, in order to improve the robustness of CS towards packet losses.

Furthermore, we intend to perform more simulations, improving the comparison between both DECA and CS schemes. A more general network scenario can be considered, with more sensor nodes performing transmissions. Moreover, the measurements can be quantized in DECA simulations with the same bit–depths used in CS, and the number of measurements/coefficients to be transmitted (M) can be varied in simulations of both schemes.

Finally, it is intended to implement both energy conservation frameworks in a real WSN, and compare the results against simulated ones.

7.5 List of Publications

During the development of this thesis, some parts of this work have been published and/or submitted to conferences and journals, listed hereafter:

7.5.1 Conference Publications

1. HENRIQUES, F. R.; LOVISOLO, L.; RUBINSTEIN, M. G., "Reconstrução com Eficiência Energética de um Processo usando Redes de Sensores sem Fio," In Anais do XXIX Simpósio Brasileiro de Telecomunicações–SBrT 2011, Curitiba, 2011.

2. HENRIQUES, F. R.; LOVISOLO, L.; RUBINSTEIN, M. G., "Avaliação da micro Conectividade de uma Rede de Sensores Sem Fio com Protocolo Dormir/Acordar," In Anais do XXX Simpósio Brasileiro de Telecomunicações–SBrT 2012, Brasília, 2012.

3. HENRIQUES, F. R.; LOVISOLO, L.; RUBINSTEIN, M. G., "Energy-Efficient Reconstruction of Environmental Data with a Multihop Wireless Sensor Network," In Anais do XXXI Simpósio Brasileiro de Telecomunicações–SBrT 2013, Fortaleza, 2013.

4. HENRIQUES, F. R.; LOVISOLO, L.; RUBINSTEIN, M. G., " Algorithms for Energy Efficient Reconstruction of a Process with a Multihop Wireless Sensor Network," In Proceedings of the Latin American Symposium on Circuits and Systems–LASCAS2013, Cusco, Peru, 2013.

7.5.2 Journal Submissions

1. HENRIQUES, F. R.; LOVISOLO, L.; RUBINSTEIN, M. G., "DECA: Distributed Energy Conservation Algorithm for Wireless Sensor Networks," submitted to EURASIP Journal on Wireless Communications and Networking, 2015.

2. HENRIQUES, F. R.; LOVISOLO, L.; DA SILVA, E. A. B., "Rate–Distortion Performance of Compressive Sensed Measurements Over Wireless Sensor Networks," Submitted to Digital Signal Processing, 2015.

Bibliography

[1] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., et al. "Wireless Sensor Networks: A Survey", *Computer Networks*, v. 38, n. 4, pp. 393–422, March 2002.

[2] BECKWITH, R., TEIBEL, D., BOWMEN, P. "Pervasive Computing and Proactive Agriculture". In: *Proceedings of the2nd International Conference on Pervasive Computing*, Vienna–Austria, April 2004.

[3] MAINWARING, A., POLASTRE, J., SZEWCZY, R., et al. "Wireless Sensor Networks for Habitat Monitoring". In: *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, New York–USA, September 2002.

[4] RAZA, U., CAMERRA, A., MUTPHY, A. L., et al. "What does Model-Driven Data Acquisition Really Achieve in Wireless Sensor Networks?" In: *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications*, pp. 19–23, Lugano, March 2012.

[5] SZEWCZYK, R., MAINWARING, A., POLASTRE, J., et al. "An Analysis of a Large Scale Habitat Monitoring Application". In: *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 214–226, New York–USA, 2004.

[6] SZEWCZYK, R., OSTERWEIL, E., POLASTRE, J. "Habitat Monitoring", *Communications of the ACM*, v. 47, n. 6, pp. 393–422, 2002. doi: 10.1016/S1389-1286(01)00302-4.

[7] SACHAN, V. K., IMAM, S. A., BEG, M. T. "Energy-efficient Communication Methods in Wireless Sensor Networks: A Critical Review", *International Journal of Computer Applications*, v. 39, n. 17, 2012.

[8] SAUSEN, P. S., SOUSA, J. R. B., SPOHN, M. A., et al. "Dynamic Power Management with Scheduled Switching Modes in Wireless Sensor Networks". In: *Proceedings of the 15th Annual Meeting of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer*

and Telecommunication Systems, pp. 1–8, Istanbul, October 2007. IEEE. doi: 10.1109/MASCOTS.2007.20.

[9] AKYILDIZ, I. F., VURAN, M. C., AKAN, O. B. "On Exploiting Spatial and Temporal Correlation in Wireless Sensor Networks". In: *Proceedings of the Modeling and Optimization in Mobile, Ad Hoc and Wireless Sensor Networks*, pp. 71–80, UK, March 2004.

[10] CUNHA, D. O., LAUFER, R. P., MORAES, I. M., et al. "A Bio-Inspired Field Estimation Scheme for Wireless Sensor Networks", *Annals of Telecommunications - Special Issue in Wireless Sensor Networks*, v. 60, n. 7/8, pp. 806–818, 2005. doi: 10.1007/BF03219948.

[11] GAO, D., ZHANG, L., WANG, H. "Energy Saving with Node Sleep and Power Control Mechanism for Wireless Sensor Networks", *The Journal of China Universities of Posts and Telecommunications*, v. 18, n. 1, pp. 49–59, 2011.

[12] YANG, O., HEINZELMAN, W. "An Adaptative Sensor Sleeping Solution Based on Sleeping Multipath Routing and Duty-Cycled MAC Protocols", *ACM Transactions on Sensor Networks*, v. 10, n. 1, pp. 1–30, 2013. doi: 10.1145/2529977.

[13] YANG, O., HEINZELMAN, W. "Modeling and Performance Analysis for Duty-Cycled MAC Protocols with Applications to S-MAC and X-MAC", *IEEE Transactions on Mobile Computing*, v. 11, n. 6, pp. 905–921, 2012. doi: 10.1109/TMC.2011.121.

[14] HENRIQUES, F. R., LOVISOLO, L., RUBINSTEIN, M. G. "An Innovated-Based Algorithm for Energy Conservation in Multihop Wireless Sensor Networks". In: *7th International Telecommunications Symposium - ITS 2010*, Manaus–Brazil, 2010.

[15] MAH, R., TAMHANE, A., TUNG, S., et al. "Process Trending with Piecewise Linear Smoothing", *Computers & Chemical Engineering*, v. 19, n. 2, pp. 129–137, 1995. doi: 10.1016/0098-1354(94)E0042-L.

[16] HENRIQUES, F. R., LOVISOLO, L., RUBINSTEIN, M. G. "Algorithms for energy efficient reconstruction of a process with a multihop wireless sensor network". In: *Circuits and Systems (LASCAS), 2013 IEEE Fourth Latin American Symposium on*, pp. 1–4. IEEE, 2013.

[17] CHEN, W., RODRIGUES, M. R. D., WASSEL, I. J. "Distributed Compressive Sensing Reconstruction Via Common Support Discovery". In: *Proceed-*

ings of the IEEE International Conference on Communications, pp. 1–5, Kyoto–Japan, June 2011.

[18] LAB WSN, I. B. "http://db.csail.mit.edu/labdata/labdata.html". 2004.

[19] MAHMUDIMANESH, M., KHELIL, A., SURI, N. "Reordering for Better Compressibility: Efficient Spatial Sampling in Wireless Sensor Networks". In: *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing*, pp. 50–57, Newport Beach–CA–USA, June 2010.

[20] JACQUES, L., LASKA, J. N., BOUFOUNOS, P. T., et al. "Robust 1-Bit Compressive Sensing via Binary Stable Embeddings of Sparse Vectors", *IEEE Transactions on Information Theory*, v. 59, n. 4, pp. 2082–2102, April 2013.

[21] XIONG, J., TANG, Q. "1-Bit Compressive Data Gathering for Wireless Sensor Networks", *Journal of Sensors*, v. 2014, pp. 1–8, May 2014.

[22] DAI, W., MILENKOVIC, O. "Information Theoretical and Algorithmic Approaches to Quantized Compressive Sensing", *IEEE Transactions on Communications*, v. 59, n. 7, pp. 1857–1866, July 2011.

[23] CANDÈS, E. J., ROMBERG, J. K., TAO, T. "Stable signal recovery from incomplete and inaccurate measurements", *Communications on Pure and Applied Mathematics*, v. 59, n. 8, pp. 1207–1223, August 2006.

[24] DAI, W., MILENKOVIC, O. "Subspace Pursuit for Compressive Sensing Signal Reconstruction", *IEEE Transactions on Information Theory*, v. 55, n. 5, pp. 2230–2249, May 2009.

[25] SCHULZ, A., VELHO, L., DA SILVA, E. A. B. "On the Empirical Rate-Distortion Performance of Compressive Sensing". In: *Proceedings of the 16th IEEE Conference on Image Processing*, pp. 3049–3052, Cairo–Egypt, November 2009.

[26] LINDBERG, C., AMAT, A. G. I., WYMEERSCH, H. "Distributed Compressed Sensing for Sensor Networks with Packet Erasures". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 13–19, Austin–Texas–USA, December 2014.

[27] FELEMBAN, E., SHEIKH, A. A., MANZOOR, M. A. "Improving Response Time in Time Critical Visual Sensor Network Applications", *Ad Hoc Networks*, v. 23, pp. 65–79, December 2014.

[28] DA SILVA, E. A. B., CRAIZER, M. "Generalized bit-planes for Embedded Codes". In: *Proceedings of the International Conference on Image Processing (ICIP)*, pp. 317–321, Chicago–Illinois–USA, October 1998.

[29] DA SILVA, E. A. B., SAMPSON, D. G., GHANBARI, M. "A Successive Approximation Vector Quantizer for Wavelet Transform Image Coding", *IEEE Transactions on Image Processing*, v. 11, n. 12, pp. 1337–1348, February 1996.

[30] DA SILVA, E. A. B., LOVISOLO, L., DUTRA, A. J. S., et al. "FIR Filter Design Based on Successive Approximation of Vectors", *IEEE Transactions on Signal Processing*, v. 62, n. 15, pp. 3833–3848, May 2014.

[31] HENRIQUES, F. R., LOVISOLO, L., RUBINSTEIN, M. G. "DECA: Distributed Energy Conservation Algorithm for Wireless Sensor Networks", *Submitted to EURASIP Journal on Wireless Communications and Networking*, 2015.

[32] HENRIQUES, F. R., LOVISOLO, L., DA SILVA, E. A. B. "Rate–Distortion Performance of Compressive Sensed Measurements Over Wireless Sensor Networks", *Submitted to Digital Signal Processing*, 2015.

[33] ANASTASI, G., CONTI, M., FRANCESCO, M. D., et al. "Energy Conservation in Wireless Sensor Networks: A Survey", *Ad Hoc Networks*, v. 7, n. 3, pp. 537–568, maio 2009. doi: 10.1016/j.adhoc.2008.06.003.

[34] NGUYEN-XUAN, S., OH, S., SUNSHIN, A. "EE-MAC: Energy efficient-medium access control for periodic applications in border surveillance wireless sensor networks". In: *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, pp. 693–697. IEEE, 2014.

[35] SHA, M., HACKMANN, G., LU, C. "Energy-efficient low power listening for wireless sensor networks in noisy environments". In: *Proceedings of the 12th international conference on Information processing in sensor networks*, pp. 277–288. ACM, 2013.

[36] TANG, L., SUN, Y., GUREWITZ, O., et al. "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks". In: *INFOCOM, 2011 Proceedings IEEE*, pp. 1305–1313. IEEE, 2011.

[37] TANG, L., SUN, Y., GUREWITZ, O., et al. "EM-MAC: a dynamic multichannel energy-efficient MAC protocol for wireless sensor networks". In:

Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing, p. 23. ACM, 2011.

[38] HAO, X.-C., WANG, M.-Q., HOU, S., et al. "Distributed Topology Control and Channel Allocation Algorithm for Energy Efficiency in Wireless Sensor Network: From a Game Perspective", *Wireless Personal Communications*, pp. 1–21, 2014.

[39] BARCELÓ-LLADÓ, J. E., MORELL, A., SECO-GRANADOS, G. "Conditional downsampling for energy-efficient communications in wireless sensor networks", *EURASIP Journal on Advances in Signal Processing*, v. 2013, n. 1, pp. 1–16, 2013.

[40] PRADHAN, S. S., KUSUMA, J., RAMCHANDRAN, K. "Distributed compression in a dense microsensor network", *Signal Processing Magazine, IEEE*, v. 19, n. 2, pp. 51–60, 2002.

[41] SUN, N., WU, J. "Optimum sampling in spatial-temporally correlated wireless sensor networks", *EURASIP Journal on Wireless Communications and Networking*, v. 2013, n. 1, pp. 1–18, 2013.

[42] BARCELÓ-LLADÓ, J. E., PÉREZ, A. M., SECO-GRANADOS, G. "Enhanced correlation estimators for distributed source coding in large wireless sensor networks", *Sensors Journal, IEEE*, v. 12, n. 9, pp. 2799–2806, 2012.

[43] SHUMAN, D. I., NAYYAR, A., MAHAJAN, A., et al. "Measurement scheduling for soil moisture sensing: From physical models to optimal control", *Proceedings of the IEEE*, v. 98, n. 11, pp. 1918–1933, 2010.

[44] DESHPANDE, A., GUESTRIN, C., MADDEN, S. R., et al. "Model-driven data acquisition in sensor networks". In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 588–599. VLDB Endowment, 2004.

[45] WANG, Z. M., BASAGNI, S., MELACHRINOUDIS, E., et al. "Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime". In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 3–6, Hawaii, January 2005. IEEE. doi: 10.1109/HICSS.2005.259.

[46] POLASTRE, J., HILL, J., CULLER, D. "Versatile Low Power Media Access for Wireless Sensor Networks". In: *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 95–107, New York USA, 2004. ACM. doi: 10.1145/1031495.1031508.

[47] CHALLEN, G. W., WATERMAN, J., WELSH, M. "IDEA: Integrated Distributed Energy Awareness for Wireless Sensor Networks". In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, pp. 35–48, New York USA, 2010. ACM. doi: 10.1145/1814433.1814439.

[48] PANTHACHAI, Y., KEERATIWINTAKORN, P. "An Energy Model for Transmission in Telos-Based Wireless Sensor Networks". In: *Proceedings of the 4th International Joint Conference on Computer Science and Software Engineering*, 2007.

[49] POLASTRE, J., SZEWCZYK, R., CULLER, D. "Telos: Enabling Ultra-Low Power Wireless Research". In: *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, pp. 364–369. IEEE, April 2005. doi: 10.1109/IPSN.2005.1440950.

[50] CAMPISTA, M. E. M., RUBINSTEIN, M. G. "Advanced Routing Protocols for Wireless Networks", *ISTE/Wiley*, 2014.

[51] PERKINS, C., BELDING-ROYER, E., DAS, S. "Ad Hoc On Demand Distance Vector Routing (AODV)". In: *RFC 3561*, July 2003.

[52] CERVIN, A., HERINKSSON, D., LINCOLN, B., et al. "How Does Control Timing Affect Performance? Analysis and Simulation of Timing using Jitterbug and TrueTime", *IEEE Control Systems Magazine*, v. 23, n. 3, pp. 16–30, maio 2003. doi: 10.1109/MCS.2003.1200240.

[53] ZHENG, J., LEE, M. J. "A Comprehensive Performance Study of IEEE 802.15.4", *Sensor Network Operations, IEEE Press, Wiley Interscience*, v. Chapter 4, pp. 218–237, 2006.

[54] BARONTI, P., PILLAI, P., CHOOK, V., et al. "Wireless Sensor Networks: a Survey on the State of Art and the 802.15.4 and ZigBee Standards", *Computer Communications*, v. 30, n. 7, pp. 1655–1695, maio 2007. doi: 10.1016/j.comcom.2006.12.020.

[55] DONOHO, D. L. "Compressed Sensing", *IEEE Transactions on Information Theory*, v. 52, n. 4, pp. 1289–1306, April 2006.

[56] CANDÈS, E. J., ROMBERG, J. K. "Sparsity and incoherence in compressive sampling", *Inverse Problems*, v. 23, n. 3, pp. 1–20, April 2007.

[57] ABOLGHASEMI, V., FERDOWSI, S., MAKKIABADI, B., et al. "On Optimization of the Measurement Matrix for Compressive Sensing". In:

18th European Signal Processing Conference (EUSIPCO), pp. 427–431, Aalborg–Denmark, August 2010.

[58] LOVISOLO, L., PEREIRA, M. P., DA SILVA, E. A. B., et al. "On the Design of Maximally Incoherent Sensing Matrices for Compressed Sensing and its Extension for Biorthogonal Bases Case", *Digital Signal Processing*, v. 27, pp. 12–22, April 2014.

[59] CANDÈS, E., ROMBERG, J., TAO, T. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information", *IEEE Transactions on Information Theory*, v. 52, n. 2, pp. 489–509, February 2006.

[60] CANDÊS, E., ROMBERG, J. "L1–magic". In: *http://www.l1–magic.org*, 2005.

[61] KARAHANOGLU, N. B., ERDOGAN, H. "A* Orthogonal Matching Pursuit: Best-First Search for Compressed Sensing Signal Recovery", *Digital Signal Processing*, v. 22, n. 4, pp. 555–568, 2012.

[62] TIBSHIRANI, R. "Regression Shrinkage and Selection via the LASSO", *J. R. Statist.*, v. 58, pp. 267–288, 1996.

[63] BOYD, S., VANDENBERGHE, L. "Convex Optimization", *Cambridge University Press*, 2004.

[64] PATI, Y. C., REZAIIFAR, R., KRISHNAPRASAD, P. S. "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition". In: *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44, Pacific Grove–CA–USA, November 1993.

[65] KOENIG, S., LIKHACHEV, M., Y., et al. "Incremental Heuristic Search in AI", *AI Magazine*, v. 25, n. 2, pp. 99–112, 2004.

[66] DECHTER, R., PEARL, J. "Generalized Best-First Strategies and the Optimality of A*", *Journal of the ACM*, v. 32, n. 3, pp. 505–536, 1985.

[67] KARAHANOGLU, N. B., ERDOGAN, H. "A Comparison of Termination Criteria for A*OMP". In: *Proceedings of the 20th European Signal Processing Conference*, pp. 1449–1453, Buchartest, August 2012. IEEE.

[68] BERG, E. V. D., FRIEDLANDER, M. P. "Sparse Optimization with Least-Squares Constraints", *SIAM J. Optimization*, v. 21, n. 4, pp. 1201–1229, 2011.

[69] DONOHO, D., JOHNSTONE, I. "Ideal Spatial Adaptation by Wavelet Shrinkage", *Biometrika*, v. 81, pp. 425–455, 1994.

[70] ELAD, M. "Why Simple Shrinkage is Still Relevant for Redundant Representations?" *IEEE Transactions on Information Theory*, v. 52, n. 12, pp. 5559–5569, 2006.

[71] VAN DEN BERG, E., FRIEDLANDER, M. P. "SPGL1: A solver for large-scale sparse reconstruction". In: *http://www.cs.ubc.ca/labs/scl/spgl1*, 2011.

[72] COIFMAN, R., GESHWIND, F., MEYER, Y. "Noiselets", *Applied and Computational Harmonic Analysis*, v. 10, n. 1, pp. 27–44, January 2001.

[73] ZHAO, D., CHAN, Y. K., GAO, W. "Low-Complexity and Low-Memory Entropy Coder for Image Compression", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 11, n. 10, pp. 1140–1145, October 2001.

[74] CERPA, A., WONG, J. L., KUANG, L., et al. "Statistical Model of Lossy Links in Wireless Sensor Networks". In: *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 81–88, Los Angeles–California–USA, April 2005.

[75] BACCOUR, N., KOUBÂAL, A., MOTTOLA, L., et al. "Radio Link Quality Estimation in Wireless Sensor Networks: A Survey", *ACM Transactions on Sensor Networks*, v. 8, n. 4, pp. 1–34, September 2012.

[76] GOYAL, V. K., KOVACEVIC, J., KELNER, J. A. "Quantized Frames Expansions with Erasures", *Applied and Computational Harmonic Analysis*, v. 10, n. 3, pp. 203–233, May 2001.

[77] BOUFOUNOS, P., OPPENHEIM, A. V., GOYAL, V. K. "Causal Compensation for Erasures in Frame Representations", *IEEE Transactions on Signal Processing*, v. 56, n. 3, pp. 1071–1082, March 2008.

[78] ZHANG, H., ARORA, A., CHOI, Y., et al. "Reliable bursty convergecast in wireless sensor networks", *Computer Communications*, v. 30, pp. 2560–2576, June 2007.

[79] WU, L., YU, K., DU, T., et al. "Efficient Information Transmission under Lossy WSNs Using Compressive Sensing". In: *Proceedings of the IEEE 9th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 493–498, Hangzhou–China, June 2014.

[80] PAN, L., GAO, H., GAO, H., et al. "A Spatial Correlation Based Adaptive Missing Data Estimation Algorithm in Wireless Sensor Networks", *International Journal of Wireles Information Networks*, v. 21, n. 4, pp. 280–289, October 2014.

[81] BHAR, J. "A Mac Protocol Implementation for Wireless Sensor Network", *Journal of Computer Networks and Communications*, v. 2015, pp. 1–12, March 2015.

[82] MALAV, K., GUPTA, D., MURRAY, V. "Energy Efficient Routing in Zigbee Wireless Sensor Network - A Review", *International Journal of Advanced Research in Computer and Communication Engineering*, v. 4, n. 4, pp. 336–343, April 2015.

[83] BJONTEGAARD, G. "Calculation of average PSNR differences between RD–curves", *Technical Report VCEG–M33*, v. ITU–T SG16/Q6, 2001.

[84] HANHART, P., EBRAHIMI, T. "Calculation of average coding efficiency based on subjective quality scores", *Journal of Visual Communication and Image Representation*, v. 25, n. 3, pp. 555–564, April 2014.

[85] ZHOU, C., LU, Y., XIONG, C., et al. "Macroblock Mode Pre-classification Algorithms based on Motion Vectors Filtering for H.264/AVC", *Journal of Multimedia*, v. 8, n. 2, pp. 129–136, April 2013.

[86] LASKA, J. N., BOUFOUNOS, P. T., DAVENPORT, M. A. "Democracy in Action: Quantization, Saturation and Compressive Sensing", *Applied and Computational Harmonic Analysis*, v. 31, n. 3, pp. 429–443, 2011.