Compressed Sensing for Real-Time Energy-Efficient ECG Compression on Wireless Body Sensor Nodes

Hossein Mamaghanian*, *Student Member, IEEE*, Nadia Khaled, *Member, IEEE*, David Atienza, *Member, IEEE*, and Pierre Vandergheynst, *Senior Member, IEEE*

***Abstract*—Wireless body sensor networks (WBSN) hold the promise to be a key enabling information and communications technology for next-generation patient-centric telecardiology or mobile cardiology solutions. Through enabling continuous remote cardiac monitoring, they have the potential to achieve improved personalization and quality of care, increased ability of prevention and early diagnosis, and enhanced patient autonomy, mobility, and safety. However, state-of-the-art WBSN-enabled ECG monitors still fall short of the required functionality, miniaturization, and energy efficiency. Among others, energy efficiency can be improved through embedded ECG compression, in order to reduce airtime over energy-hungry wireless links. In this paper, we quantify the potential of the emerging compressed sensing (CS) signal acquisition/compression paradigm for low-complexity energy-efficient ECG compression on the state-of-the-art Shimmer WBSN mote. Interestingly, our results show that CS represents a competitive alternative to state-of-the-art digital wavelet transform (DWT)-based ECG compression solutions in the context of WBSN-based ECG monitoring systems. More specifically, while expectedly exhibiting inferior compression performance than its DWT-based counterpart for a given reconstructed signal quality, its substantially lower complexity and CPU execution time enables it to ultimately outperform DWT-based ECG compression in terms of overall energy efficiency. CS-based ECG compression is accordingly shown to achieve a 37.1% extension in node lifetime relative to its DWT-based counterpart for "good" reconstruction quality.**

***Index Terms*—Compressed sensing (CS), ECG compression, real-time ambulatory ECG monitoring, wireless body sensor nodes.**

I. Introduction

OUR modern society is today threatened by an incipient health care delivery crisis caused by the current demographic and lifestyle trends. On the one hand, the world's population is fast aging resulting into an increased prevalence of cardiac disorders. On the other hand, our busy and often unhealthy lifestyles are fueling the rise of the number of people unsuspectingly developing or living with chronic cardiovascular conditions for decades. As a matter of fact, according to the World Health Organization, cardiovascular diseases are the number one cause of death worldwide, responsible for an estimated 17.1 million deaths in 2004 (i.e., 29% of all deaths worldwide) and economic fallout in billions of dollars [1]. These increasingly prevalent cardiac diseases are requiring escalating levels of supervision and medical management, which are contributing to skyrocketing health care costs and, more importantly, are unsustainable for traditional health care infrastructures. Wireless body sensor network (WBSN) technologies promise to offer large-scale and cost-effective solutions to this problem. These solutions consist in outfitting patients with wearable, miniaturized and wireless sensors able to measure and wirelessly report cardiac signals to telehealth providers. They are poised to enable the required personalized, real-time and long-term ambulatory monitoring of chronic patients, its seamless integration with the patient's medical record and its coordination with nursing/medical support.

While the resting ECG monitoring is standard practice in hospitals, its ambulatory counterpart is still facing many technical challenges. For instance, the three-lead ECG is still nowadays recorded on a rather bulky and obtrusive commercial data-logging (Holter) device during one to five days of normal daily activities of a patient. These systems suffer from important limitations: limited autonomy, bulkiness, and no or limited wireless connectivity. Recently, the realization of wireless-enabled low-power ECG monitors for ambulatory use has received significant industrial and academic interest. The most important highlights of these research and development efforts are: 1) Toumaz's Sensium Life Pebble TZ203082 [2], an ultra-small and ultra-low-power monitor for heart rate, physical activity, and skin temperature measurements with a reported autonomy of five days on a hearing aid battery; 2) Intel's Shimmer [3], a small wireless wearable sensor platform able to record and wirelessly transmit three-lead ECG data as well as accelerometer, gyroscope, and galvanic skin response information; (3) IMEC's wireless single-lead bipolar ECG patch [4] for ambulatory monitoring claiming over ten days of monitoring on a 160 mAh Li-ion battery (for undisclosed use conditions). The clinical relevance of the first system is still being validated, as Toumaz aims to achieve more than the system's so far established accurate measurement of heart rate. The second system, which is based on commercial off-the-shelf components such as the TI MSP430 microcontroller and the CC2420 radio chipset, operates on a Li-ion battery that provides about 1 Wh of energy. According to our measurements, it is able to support a

Manuscript received September 23, 2010; revised March 8, 2011; accepted April 12, 2011. Date of publication May 19, 2011; date of current version August 19, 2011. This work was supported in part by the the Swiss Confederation under the Nano-Tera.ch NTF Project BioCS-Node. This paper was presented in part at the Conference on Design, Automation and Test, Europe, 2011. *Asterisk indicates corresponding author.*

*H. Mamaghanian is with the School of Engineering, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: hossein.mamaghanian@epfl.ch).

N. Khaled, D. Atienza, and P. Vandergheynst are with the School of Engineering, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: nadia.khaled@epfl.ch; david.atienza@epfl.ch; pierre.vandergheynst@epfl.ch).

Digital Object Identifier 10.1109/TBME.2011.2156795

maximum of 6.5-day single-lead raw ECG sensing and storage on local memory. This autonomy figure is reduced by 25%, when the raw ECG data are wirelessly streamed using the ultra-low-power CC2420 in a perfect point-to-point link with no wireless protocol overhead. More importantly, this autonomy figure will undoubtedly dramatically decrease under realistic ambulatory monitoring. Finally, IMEC ultra-low-power wireless biopotential sensor node achieves its enhanced autonomy due to a proprietary customized ultra-low-power analog read-out ASIC [signal acquisition, amplification, and analog-to-digital conversion (ADC)], a proprietary ultra-low-power ultra-wideband wireless transceiver, and more importantly, dedicated signal processors to preprocess and compress the sensed data using state-of-the-art techniques, in order to reduce the airtime over power-hungry wireless links.

Based on these premises, it is today acknowledged that the achievement of truly WBSN-enabled ambulatory monitoring systems requires more breakthroughs not only in terms of ultra-low-power read-out electronics and radios, but also and increasingly so, in terms of ultra-low-power dedicated digital processors and associated embedded feature extraction and data compression algorithms. In this study, we explore a novel and promising approach based on the emerging compressed sensing (CS) framework to tackle the challenge of ultra-low-power embedded compression of ECG signals. More specifically, we quantify the potential of the CS signal acquisition/compression paradigm for low-complexity energy-efficient ECG compression and show that it represents a competitive alternative to state-of-the-art ECG compression solutions in the context of WBSN-based monitoring systems.

Since its early days in the 1970s, ECG data compression has witnessed remarkable advances fueled by, on the one hand, the everincreasing wish of physicians to store the huge amounts of ECG data acquired in clinical practice, and on the other hand, the emergence of mobile telecardiology over bandwidth-limited public wireless networks. Until recently (and still practically today), "mobile" telecardiology, however, referred to wireless ECG communication from an ambulance or a patient's home to the hospital. In these scenarios, the device implementing ECG data compression in real time is assumed to be a portable device with enough processing power and storage space to deploy powerful, but, complex algorithms. This explains the rich body of ECG compression techniques with ever-greater compression ratios (CRs) at the cost of ever-higher computational complexity. A concise overview of the most relevant techniques can be found in [5]–[7]. Since our focus is on ECG compression solutions implementable in real time on WBSN nodes for future "personal" or "wearable" telecardiology, we will limit our review to those state-of-the-art algorithms that were shown to achieve competitive CRs while being amenable to efficient, fast, and low-memory-footprint implementation on our target platforms. Interestingly, these algorithms are all based on the digital wavelet transform (DWT), namely, the embedded zerotree wavelet (EZW) [8], the set partitioning in hierarchical trees (SPIHT) [9], and thresholding-based algorithms [10], [11]. Furthermore, the latter thresholding-based algorithm [11], where a fixed percentage of wavelet coefficients are zeroed, was shown to outperform EZW and SPIHT at a lower computational cost. Therefore, it is the baseline algorithm against which the performance, computational complexity, and energy consumption of CS-based ECG compression are benchmarked.

The prior art established the largely compressible nature of the ECG, as it can be very well approximated by a compact representation in the wavelet domain. Capitalizing on this sparsity, we thus propose to apply the emerging CS approach to the acquisition and compression of this class of signals. This approach promises significant CRs while using computationally light linear encoders. It is particularly attractive for our target energy-constrained WBSN-based ECG monitoring systems because: 1) the sensor node can efficiently compress the acquired ECG signal through a small number of linear signal-independent measurements while preserving their underlying information; 2) only this small number of measurements will be wirelessly transmitted to the remote telecardiology center, where the full record can be accurately reconstructed using complex, yet computationally feasible and numerically stable, nonlinear decoding.

To the best of our knowledge, CS has never been applied to ECG. It has, however, been recently considered for efficient EEG acquisition and compression [12]–[14]. The two first works have only focused on the sparse modeling of EEG signals and on assessing the efficiency of CS-based compression in terms of signal reconstruction errors. The work in [14] tried to estimate the low-power potential of CS for portable EEG systems using datasheet-extracted power consumption figures for the various components as well as estimates for the required amount of processing and wireless transmission. The measurement results reported in the present work for CS-based single-lead ECG compression on the commercial Shimmer indicate that their estimates are inaccurate.

To the best of our knowledge, the present work introduces $three$ main contributions: 1) It is the first to thoroughly quantify the potential of CS for low-complexity energy-efficient ECG compression on resource-constrained WBSN platforms; 2) to quantify this potential, it provides an exhaustive system-level comparison between CS-based and state-of-the-art DWT-based embedded ECG compression. This system-level comparison is based on embedded implementations of the two considered ECG compression algorithms, which have been optimized for real-time implementation on a representative state-of-the-art WBSN mote, namely, Shimmer. In addition to the careful motivation of the various underlying tradeoffs and implementation choices, this study proposes a comparative study of these algorithms in terms of signal reconstruction metrics, embedded memory usage, CPU execution time, and energy consumption; 3) beyond the comparative study, our results reveal that the widely used general-purpose MSP430 microcontroller [15] fails to efficiently support embedded processing and as such effectively limits the WBSN lifetime. More interestingly, our results show that outfitting WBSN platforms with more competitive ultra-low-power processors with advanced signal processing capabilities can substantially boost their energy efficiency and lifetime.

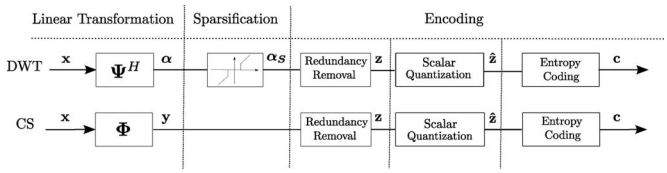

Fig. 1. Block diagram of the two ECG compression schemes implemented on the Shimmer wireless mote.

II. Methods

This study proposes a thorough system-level performance comparison between the state-of-the-art thresholding-based DWT algorithm of [11] and a CS-based algorithm in the context of ECG data compression on the embedded Shimmer wireless mote. These two algorithms are schematized in Fig. 1. They both consist of three processing stages: a linear transformation is first applied to the original ECG signal, followed by an optional "sparsification" stage, and a final encoding stage outputs the compressed signal to be wirelessly transmitted. As highlighted in Fig. 1, the fundamental difference between the two approaches lies in the fact that the former algorithm explicitly exploits the sparsity of the ECG signal by computing its sparse expansion and *adaptively* encoding the coefficients of this expansion, while the latter algorithm *nonadaptively* acquires a few random measurements of the ECG signal and only implicitly relies on ECG signal sparsity to guarantee accurate reconstruction. This section describes further in detail the two algorithms and introduces the corresponding data models.

Notation: In all the following, normal letters designate scalar quantities, boldface lower-case letters indicate column vectors, and boldface capitals represent matrices. $\langle \mathbf{a}, \mathbf{b} \rangle$ stands for the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$. Moreover, m_i and $\mathbf{M}_{i,j}$ are the ith entry of vector $\mathbf{m}$ and the (i,j)th entry of matrix $\mathbf{M}$, respectively. Finally, $(.)^H$ and $||.||_p$ denote the conjugate transpose, and the l_p-norm of a vector, respectively.

A. Thresholding-Based DWT Compression Algorithm

Fig. 1 (top) depicts the block diagram of this algorithm, as originally introduced in [11]. Due to the limited on-chip memory and real-time constraints, the digitized ECG signal is processed in nonoverlapping windows of N samples. Let $\mathbf{x}$ be the real-valued N-dimensional ECG signal vector corresponding to such a window (i.e., $\mathbf{x} \in \mathbb{R}^N$). $\mathbf{x}$ expands in an orthonormal Daubechies[1] wavelet basis $\mathbf{\Psi} = [\boldsymbol{\psi}_1 | \boldsymbol{\psi}_2 | \cdots | \boldsymbol{\psi}_N]$ as follows:

$$\mathbf{x} = \mathbf{\Psi}\boldsymbol{\alpha} \tag{1}$$

where $\boldsymbol{\alpha}$ represents the N-dimensional coefficient vector. Given the highly sparse nature of $\mathbf{x}$ in the wavelet domain, most coefficients in $\boldsymbol{\alpha}$ can be zeroed without much signal quality loss. Accordingly, for a target CR, the thresholding-based DWT compression algorithm [11] computes $\boldsymbol{\alpha}_S$ the S-sparse approximation of $\boldsymbol{\alpha}$, where all coefficients have been zeroed except the S largest values ($S \ll N$), as illustrated in Fig. 1. Then, the final encoding stage further removes any remaining redundancy between consecutive $\boldsymbol{\alpha}_S$ through a redundancy removal module. The output of this module, $\mathbf{z}$, is subsequently uniformly quantized yielding the N-dimensional vector $\hat{\mathbf{z}}$. The latter is encoded using lossless Huffman coding to produce the encoded vector $\mathbf{c}$.

B. CS-Based Compression Algorithm

Sensing and processing information have traditionally relied on the Shannon sampling theorem, one of the central tenets of digital signal processing. This theorem states that, given a signal of bandwidth Ω, it is sufficient to sample it at 2Ω samples per second (i.e., the Nyquist rate) to ensure faithful representation and reconstruction. However, this traditional ADC paradigm has been challenged lately. First, there are many situations where Ω is so large that constraints put on sampling architectures are simply unbearable. Second, even for relatively low signal bandwidths such as our target wearable ECG application, given the established sparsity of the ECG signal, (above) Nyquist-rate sampling produces a large amount of redundant digital samples, which are costly to wirelessly transmit, and severely limit the sensor nodes lifetime. If one sets course to design energy-efficient embedded ECG sensors, it is desirable to reduce the number of acquired ECG samples by taking advantage of the sparsity, or, the reduced "information rate" of the ECG signal.

Compressed sensing is a methodology that has been recently proposed to address this problem [17]–[19]. But, as we argue herein, it is also particularly well suited for low-power implementations because it dramatically reduces the need for resource- (both processing and storage) intensive digital signal processing operations. The main idea behind CS is relatively simple and will be subsequently illustrated using the discrete-time data model introduced in Section II-A and Fig. 1. As aforementioned, the original ECG signal $\mathbf{x}$ has a sparse approximation, i.e., it can be represented by a linear superposition of S elements of an orthonormal wavelet basis, $\mathbf{x} \approx \sum_{k=1}^{S} \alpha_k \boldsymbol{\psi}_k$[2], with $S \ll N$. Conventionally (as in Section II-A), one would collect ECG samples at the Nyquist rate forming $\mathbf{x}$ and *then* compress it using nonlinear digital compression techniques. CS offers a striking alternative by showing that you can collect roughly S samples using simple *analog* measurement waveforms, thus sensing/sampling and compressing at the same time. Moreover, by merging the sampling and compression steps, CS removes a large part of the digital architecture. This so-called "analog CS," where the compression occurs in the analog sensor read-out electronics prior to ADC is our ultimate goal. Its demonstration still requires extensive work on the analog sensor read-out electronics. Consequently, in the present work, we propose to approach it through "digital CS," where the linear CS compression is applied after the ADC.

Accordingly, as depicted in Fig. 1 (bottom), we collect M samples using simple measurement vectors $\{\boldsymbol{\phi}_i\}_{1 \le i \le M}$ as $y_i = \boldsymbol{\phi}_i^H \mathbf{x} = \langle \boldsymbol{\phi}_i, \mathbf{x} \rangle$, $i = 1, \ldots, M$. Consequently, the CS linearly compressed data vector $\mathbf{y} \in \mathbb{R}^M$ is described by $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$,

[1]While [11] used the biorthogonal bior4.4 wavelet, we herein utilize the orthogonal Daubechies wavelets (db 10) as the most popular wavelet family for ECG compression [16].

[2]For notational simplicity, the entries of $\boldsymbol{\alpha}$ are considered to be in a decreasing order.

where $\mathbf{\Phi}$ denotes the $M \times N$ measurement or sensing matrix with the vectors $\phi_1^H, \cdots, \phi_M^H$ as rows. Then, the measurement vector $\mathbf{y}$ is directly encoded following the same steps as the DWT-based compression scheme of Section II-A. It is important to notice that the sensing matrix $\mathbf{\Phi}$ does not depend on the signal: CS proposes a simple linear sampling strategy that is only marginally off the optimal but complex best adaptive strategy. To guarantee the robust and efficient recovery of the S-sparse signal $\boldsymbol{\alpha}_S$, the sensing matrix $\mathbf{\Phi}$ must obey the key *restricted isometry property* (RIP) [20], [21]

$$(1-\delta_S)\,||\boldsymbol{\alpha}||_2 \leq ||\mathbf{\Phi}\mathbf{\Psi}\boldsymbol{\alpha}||_2 \leq (1+\delta_S)\,||\boldsymbol{\alpha}||_2 \qquad (2)$$

for all S-sparse vectors $\boldsymbol{\alpha}$. δ_S is the isometry constant of matrix $\mathbf{\Phi}$, which must be not too close to one. This property is difficult to verify. Practically, it is replaced by the requirement that the sensing matrix $\mathbf{\Phi}$ must be such that its coherence $\mu(\mathbf{\Phi}, \mathbf{\Psi})$ with the sparsity basis $\mathbf{\Psi}$ defined as [22], [23]:

$$\mu(\mathbf{\Phi}, \mathbf{\Psi}) = \sqrt{N}. \max_{1 \leq k,j \leq N} |\langle \boldsymbol{\phi}_k, \boldsymbol{\psi}_j \rangle| \qquad (3)$$

is small enough. A universal good choice for the sensing matrix $\mathbf{\Phi}$ are random matrices, such as random matrices with independent identically distributed (i.i.d.) entries formed by sampling: 1) a Gaussian distribution $\mathcal{N}(0, 1/N)$; (2) a symmetric Bernoulli distribution $(P(\mathbf{\Phi}_{i,j} = \pm 1/\sqrt{N}) = 1/2)$. Interestingly, many efficient sensing matrices can be constructed with simple pseudorandom design that can be generated and applied using a small amount of onboard memory and computational power. The price to pay for these advantages is a more complex decoder: the state of the art consists in recovering the original signal by solving a convex optimization problem. One important benefit of decoding by optimization, though, is that CS decoders are notably robust to noise and quantization errors. This robustness makes it possible to work with lower precision digital arithmetic, and further enhances compression and reduces demands on digital back end and onboard memory. This is a feature that will be extensively exploited in our hardware implementation.

If the RIP holds, then accurate reconstruction can be accomplished by solving the following convex optimization problem:

$$\min_{\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^N} ||\tilde{\boldsymbol{\alpha}}||_1 \qquad \text{subject to} \qquad ||\mathbf{\Phi}\mathbf{\Psi}\tilde{\boldsymbol{\alpha}} - \mathbf{y}||_2 \leq \sigma \qquad (4)$$

where σ bounds the amount of noise unavoidably corrupting the data. Many algorithms were introduced to solve this reconstruction problem, including interior-point algorithms [24], [25], gradient projection [26], iterative thresholding [27], and greedy approaches such as orthogonal matching pursuit (OMP) [28], [29]. Our results are based on the basis pursuit denoise algorithm provided in the SPGL1 solver [30].

III. ECG Database and Performance Metrics

To validate the performance of the considered compression schemes, we use the MIT-BIH Arrhythmia Database [31] that is the most commonly used database for the comparative study of ECG compression algorithms. This database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory. The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10-mV range.

TABLE I
PRD and corresponding quality class [32]

PRD	Reconstructed Signal Quality
$0 \sim 2\%$	"Very good" quality
$2 \sim 9\%$	"Very good" or "good" quality
$\geq 9\%$	Not possible to determine the quality group

Moreover, to quantify the compression performance while assessing the diagnostic quality of the compressed ECG records, we employ the two most widely used performance metrics, namely, *the compression ratio* and *percentage root-mean-square difference* (PRD) [5]. CR is defined as

$$\text{CR} = \frac{b_{\text{orig}} - b_{\text{comp}}}{b_{\text{orig}}} \times 100 \qquad (5)$$

where b_{orig} and b_{comp} represent the number of bits required for the original and compressed signals, respectively. The PRD, and associated SNR, quantifies the percent error between the original signal vector $\mathbf{x}$ and the reconstructed $\tilde{\mathbf{x}}$:

$$\text{PRD} = \frac{||\mathbf{x} - \tilde{\mathbf{x}}||_2}{||\mathbf{x}||_2} \times 100 \qquad (6)$$

$$\text{SNR} = -20 \log_{10} (0.01\text{PRD}). \qquad (7)$$

The link between the measured PRD and the diagnostic distortion is established based on the work of Zigel *et al.* on the weighted diagnostic measure for ECG signal compression [32], which classifies the different values of PRD based on the signal quality perceived by a specialist. Table I reports the resulting different quality classes and corresponding PRD. It is worthwhile mentioning that $\mathbf{x}$ consistently refers to the original ECG signal with its dc component removed, which guarantees the relevance of the classification of Table I.

IV. Real-Time Embedded Compression

This section first describes the target embedded sensor platform for which the two compression algorithms of Section II are optimized, implemented, and demonstrated. Then, it reports the most relevant platform-dependent optimizations introduced on the two original algorithms to achieve real-time execution and optimized processing and memory footprint.

A. The Shimmer Embedded Platform

From the hardware viewpoint, the Shimmer mainboard includes the low-power Texas Instrument 16-bit MSP430F1611 microcontroller, a low-power CC2420 IEEE 802.15.4-compliant radio, and a Bluetooth module. The MSP430 microcontroller runs at 8 MHz, has 10 kB of RAM, 48 kB of Flash and includes a fast hardware multiplier, but does not include a floating-point unit. A three-lead ECG daughterboard may be connected to the internal connector pin of the Shimmer mainboard, with application to the skin via four conventional disposable electrodes. The ECG daughterboard was validated for ambulatory ECG for research purposes [3]. Since our aim is to comparatively study the two compression algorithms on

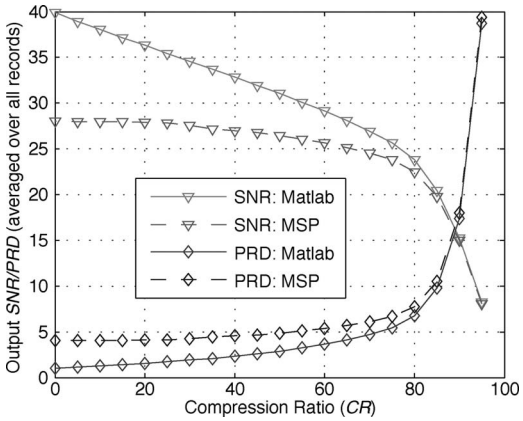

Fig. 2. Comparison of the output SNR and PRD versus CR of the MATLAB version and the MSP430 implementation for DWT-based compression.

the standard MIT-BIH arrhythmia database, we did not perform real-time ECG acquisition via the ECG daughterboard, but alternatively used the Shimmer serial port to read in the database records resampled at 256 Hz, and to read out the encoded ECG data.

From the software viewpoint, the open-source FreeRTOS kernel [33] was used to implement the real-time wireless communication protocol. FreeRTOS is a light, real-time operating system for embedded devices that allows a strict control over the timing of tasks and their relative priorities. We also chose Code Composer Essentials v3.2.0.24.2 [34] from Texas Instruments to generate the binaries of both compression algorithms, for its superior and customizable optimization levels for assembly code generation. The compiler uses automatically the hardware multiplier of the MSP430 for multiplication operations, but since the MSP430 does not include a floating-point unit, the compiler has to replace many arithmetic operations in the DWT-based compression algorithm by software emulation code, which implies a significant performance penalty if the proposed algorithm is simply compiled for Shimmer. Thus, we performed several implementation optimizations for improved execution time, as subsequently explained.

B. Thresholding-Based DWT Compression Algorithm

1) Linear transformation and sparsification: The ECG samples in $\mathbf{x}$ are stored using 12-bit resolution, their original and sparse wavelet expansion coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}_S$, respectively, are both represented using 12 bits. Due to the limited 10-kB RAM memory, it was determined that the largest processing window is of length $N = 512$ ECG samples acquired at the above-Nyquist sampling rate of 256 Hz. Moreover, due to the inefficient implementation of floating-point operations on the MSP430, the DWT had to be implemented on the platform using 16-bit integer operations. This integer implementation was instrumental to achieve real-time operation; it, however, unavoidably introduced additional quantization errors. Fig. 2 depicts the output SNR and PRD of the double-precision MATLAB version and the MSP430 integer implementation of the DWT-based compression algorithm, respectively. It clearly shows that the quantization error introduced in the MSP430 yields a reduction in signal quality. However, this reduction is marginal for high CRs that are the ones of interest for this study.

2) Interpacket redundancy removal: Since no beat synchronization was performed[3], there is no interpacket redundancy to be exploited between consecutive $\boldsymbol{\alpha}_S$.

3) Huffman coding: Since the N-dimensional vector $\boldsymbol{\alpha}_S$ is exactly S-sparse; we can either directly encode it or only encode its S nonzero entries and their corresponding indexes. The latter approach requires different codebooks for the coefficients and indexes, whereas the former obviously avoids the index codebook. Our simulations (omitted for lack of space) showed that the first approach has better performance for higher CRs due to the larger number of zeros in the coefficient vector, which are more efficiently encoded. The implementation of thresholding-based DWT compression requires 4.6 kB of RAM memory and 10 kB of Flash memory, 5 kB of which is used for Huffman codebook storage.

C. CS-Based Compression Algorithm

1) Linear transformation: The implementation of Gaussian random sensing with matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ requires the implementation of a Gaussian-distributed random number generator on the embedded platform and the computation of a large matrix multiplication. This is too complex, time consuming, and is certainly not a real-time task for the MSP430. To address this problem, we explored three different approaches to the implementation of the random sensing matrix $\boldsymbol{\Phi}$.

a) Quantized Gaussian random sensing: We implemented an 8-bit quantized version of a normal random number generator to form $\boldsymbol{\Phi}$. Our simulations showed no meaningful loss in signal quality between the quantized and the original floating-point normal random number generation. While this quantized version can be implemented on the MSP430, it was discarded for its important drawbacks: 1) it uses the complex $\log$ and sqrt functions; 2) for each input ECG vector, it requires the generation of and the multiplication by a large number of normal random numbers; 3) it is clearly not real time, as it requires over 1 min to process a 2-s ECG vector (i.e., $N = 512$ samples at the rate of 256 Hz).

b) Pseudorandom sensing: We try to circumvent the onboard generation of the normal random numbers by storing them on the platform. Due to the memory constraints, which make it impossible to store the full Gaussian sensing matrix of size $(M \times N)$, we instead store one normal random column vector and generate the other columns of our sensing matrix by shuffling the positions of the entries of this vector. The shuffling process is as follows: The generated random vector is sorted, and the sorted index vector is used for successively reordering the original vector. Unfortunately, this process is also time consuming as it summons a sorting algorithm in each iteration; a 2-s ECG vector is processed in 16 s. This is why we end up generating a random index vector for shuffling the original vector.

[3]Beat synchronization entails a higher complexity and does not make sense for CS-based compression which in essence makes no attempt to comprehend the acquired signal.

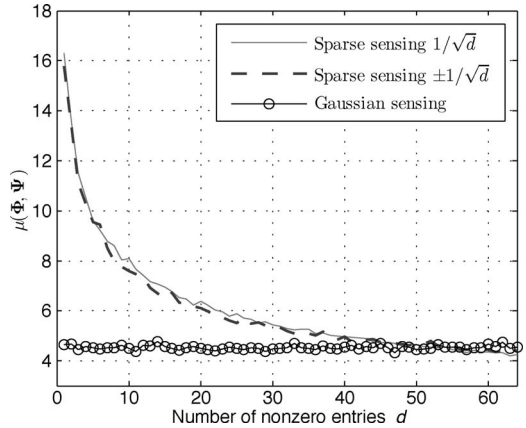

Fig. 3. Mutual coherence $\mu(\mathbf{\Phi}, \mathbf{\Psi})$ versus d.

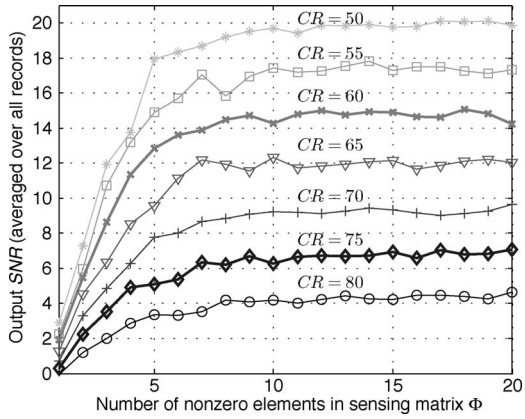

Fig. 4. Output SNR versus d for different CR.

Interestingly, since both approaches use the same (but shuffled) entries in each column, the norm of each column of the sensing matrix is constant (versus the original approach where each column had to be normalized on the platform), and the normalization can be moved to the reconstruction side. Furthermore, this implementation does not need an embedded Gaussian number generator and its underlying complex and time-consuming functions. Although this suboptimal procedure can lead to repeated or missed entries, it was verified that the output signal quality is unchanged while the execution time is significantly improved. More specifically, a 2-s ECG vector now takes 1.9 s to be CS sampled, which amounts to more than 90% of CPU execution time.

c) Sparse binary sensing: To address the shortcomings of the previous two approaches, we herein introduce an innovative approach to CS implementation on embedded sensor platforms. As aforementioned in Section II-B, it is possible to use sub-Gaussian random matrices such as the one formed by ± 1 entries. To further decrease execution time, we explore sparse binary sensing. For a sparse and binary matrix $\mathbf{\Phi}$ (i.e., each column has exactly d nonzero entries equal to 1, with $d \ll N$), the RIP property of (2) is not valid. However, such a sensing matrix satisfies a different form of this property, so-called RIP_p property. An $M \times N$ matrix, $\mathbf{\Phi}$ is said to satisfy RIP$_p$, if for any S-sparse vector $\boldsymbol{\alpha}$, we have

$$(1-\delta)\,||\boldsymbol{\alpha}||_p \leq ||\mathbf{\Phi}\mathbf{\Psi}\boldsymbol{\alpha}||_p \leq (1+\delta)\,||\boldsymbol{\alpha}||_p \qquad (8)$$

it was proven in Theorem 2 of [35] that the RIP$_1$ property of a sparse binary sensing matrix with exactly d ones on each row suffices to guarantee a good sparse approximation recovery by a linear program.

Since sparse sensing matrices allow a very fast and efficient implementation of the large matrix multiplication required by CS, we herein explore the use of sparse sensing matrices to decrease execution time. We consider two alternatives: 1) sparse sensing matrices with nonzero entries equal to $\pm 1/\sqrt{d}$; 2) sparse sensing matrices with nonzero entries equal to $1/\sqrt{d}$. Fig. 3 plots the mutual coherence of these two alternatives with the used Daubechies $db10$ wavelet basis (i.e., sparsity basis), as defined in (3). As a baseline, the mutual coherence corresponding to Gaussian sensing matrices is also reported. The mutual coherence is plotted versus the number of nonzero elements d for the two sparse sensing alternatives. The positions of the d nonzero elements are randomly chosen to keep the incoherence between the columns of the sensing matrix. Obviously, the choice of the number of nonzero elements depends on the sparsity of the signal. Fig. 3 shows that there is hardly any difference between the two sparse sensing modalities, and these suboptimal solutions fast approach the optimal Gaussian sensing modalities as d increases. The second sparse sensing modality corresponding to a sparse sensing matrix $\mathbf{\Phi}$ with exactly d nonzero entries equal to $1/\sqrt{d}$ on each column will thus be retained due to its simple implementation. This sensing modality will be subsequently referred to as *sparse binary sensing*. Furthermore, we are interested in identifying the minimum value of d that strikes the optimal tradeoff between execution time and signal reconstruction error. To do so, sparse binary sensing matrices are applied to all the records of the MIT-BIH Arrhythmia ECG database, and the output SNR of the reconstructed signals is measured. Fig. 4 reports the resulting average output SNR versus the number of nonzero elements d in the sparse binary sensing matrix $\mathbf{\Phi}$. Clearly, the output SNR saturates after $d = 12$ nonzero elements, which is the value retained for the rest of our hardware implementation on the Shimmer.

As aforementioned, all our experimental results were generated using the SPGL1 solver [30] in combination with the SPARCO toolbox [36] in MATLAB to solve the sparse recovery problem of (4). Fig. 5 shows the average output SNR versus CR for the three different approaches to the implementation of the random sensing matrix $\mathbf{\Phi}$ explored in this section, namely, (1) Gaussian random sensing: the double-precision MATLAB version; 2) pseudorandom sensing: the version based on the generation of a random index vector implemented on the MSP430; 3) sparse binary sensing: the version with $d = 12$ and all nonzero entries equal to $1/\sqrt{12}$ also implemented on the MSP430. These results were obtained for an input vector of $N = 512$ samples and a 12-bit resolution for the input vector $\mathbf{x}$ and the measurement vector $\mathbf{y}$. Interestingly, the obtained results validate that there is no meaningful performance difference between these three approaches, while sparse binary sensing offers the shortest execution time (a 2-s vector is now CS-sampled in 8 ms), the

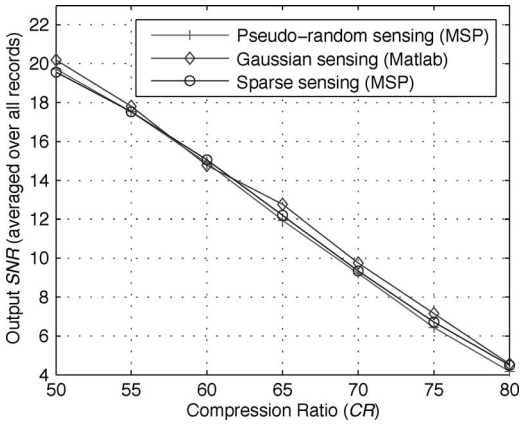

Fig. 5. Performance comparison between various CS implementations.

simplest operation and the smallest memory footprint, and as such will be our implementation of choice. Note that Fig. 5 also illustrates that CS exhibits excellent robustness with respect to quantization errors, unlike DWT (see Fig. 2).

2) Interpacket redundancy removal: The use of a fixed binary sensing matrix, combined with the periodic nature of the ECG signal, yields very similar consecutive measurement vectors $\mathbf{y}$. This is confirmed by Fig. 6(a), which plots the measured mean and variance on each of the 103 entries of 1296 consecutive measurement vectors $\mathbf{y}$ in 12-bit resolution, for a CR of CR $= 20\%$. Clearly, there is a large interpacket redundancy that must be removed prior to encoding and wireless transmission. Consequently, the redundancy removal module computes the difference between consecutive vectors, and only this difference is further processed. Fig. 6(b) further shows the pdf of the difference signal between two consecutive measurement vectors. It is thus sufficient to represent the difference signal using 9 bits, instead of the 12 bits required for the measurement vector. This observation translates into a larger compression performance.

3) Huffman coding: Interestingly, Fig. 6(b) shows that the distribution of the difference signal at the output of the redundancy removal module are far from uniform. Consequently, Huffman encoding can be used for further compression. Since the range of the difference signal just before encoding is between $[-256 : 255]$, a complete Huffman codebook of size 512 is needed with a maximum codeword length of 16 bits, for a given CR. The storage of such an offline-generated codebook requires 1 kB for the codebook itself and 512 B for its corresponding codeword lengths. The CS implementation requires 6.5 kB of RAM memory and 13.5 kB of Flash, 1.5 kB of which are for Huffman codebook storage.

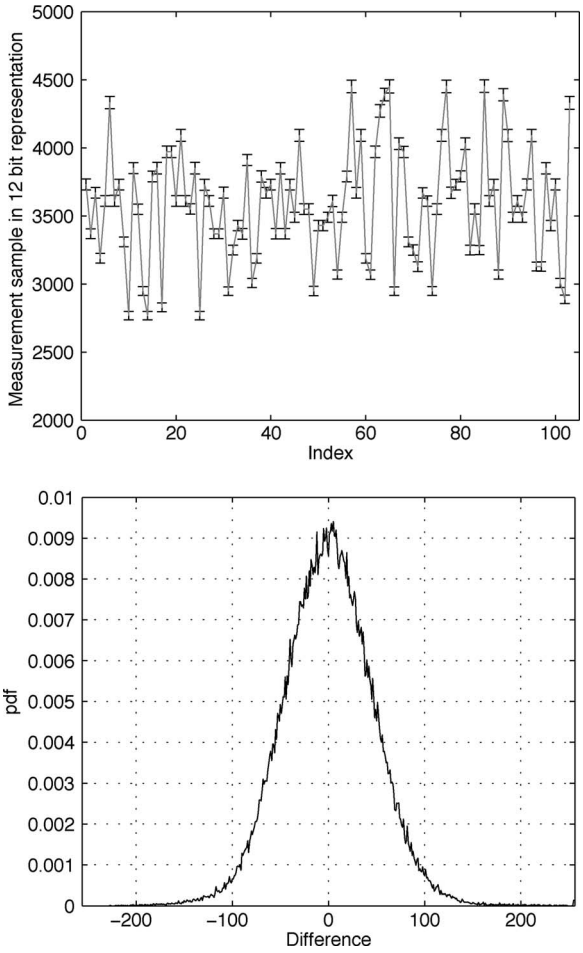

Fig. 6. (Top) Mean and variance (around each entry) of the measurement vector $\mathbf{y}$ over 1296 consecutive measurement windows. (Bottom) Pdf of the difference signal between two consecutive measurement vectors.

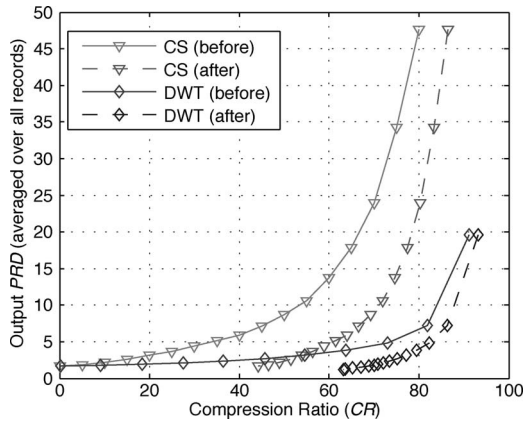

Fig. 7. Output PRD versus CR for CS and DWT before and after interpacket redundancy removal and Huffman coding.

D. Comparison Between the Two Algorithms

Fig. 7 compares the output PRD, averaged over all database records, for CS and DWT-based compression before and after interpacket redundancy removal and Huffman coding for different CRs. It confirms the crucial role of the redundancy removal module and the careful design of the Huffman encoding. This figure shows the average PRD, but there is a large variance across the individual records. Alternatively, Fig. 8(a) and (b) show the box plots for both algorithms. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers. Record 107 produces the best results for both CS and DWT: "very good" signal reconstruction quality (corresponding to PRD below 2%, see Table I) can

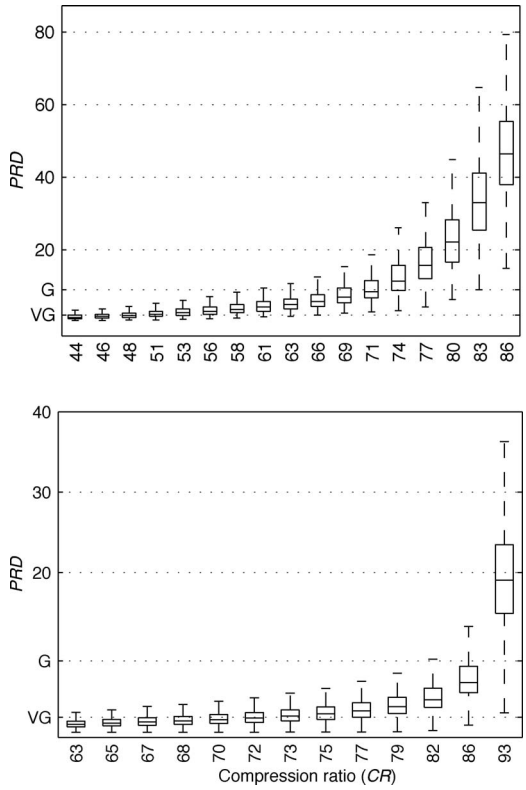

Fig. 8. Box plots for all database records for CS (top) and DWT (bottom).

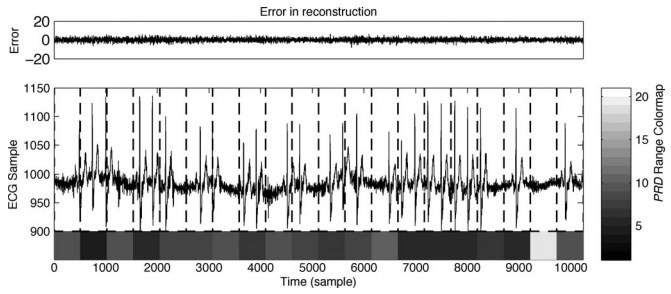

Fig. 9. Fragment of record 232 and corresponding PRD for each window.

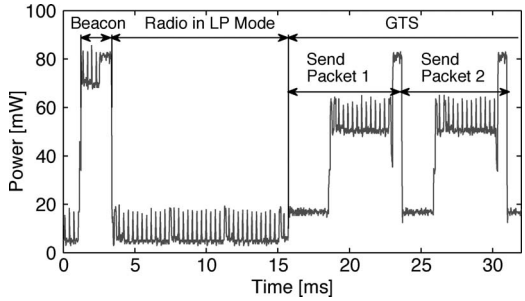

Fig. 10. Power dissipated in Shimmer during sampling, processing, and wireless transmission.

be reached for CRs of up to 61% and 75% for CS and DWT respectively, while "good" signal recovery is reached for CRs of up to 74% and 90%. On average, "very good" quality of reconstruction for DWT-based compression is achieved up to CR $= 73\%$, and up to CR $= 51\%$ for CS. Moreover, "good" reconstruction quality is reached on average for up to CR $= 90\%$ and CR $= 71\%$ for DWT and CS, respectively. As expected, the *signal-adaptive* DWT-based compression outperforms the *nonadaptive* CS-based compression. However, the measured performance of CS compression reported for the first time in this study is certainly promising in view of its very low computational complexity and limited resource needs. In fact, this measured performance is strikingly promising since the sparse reconstruction algorithm used in this study is the default basis pursuit denoise, where no attempt is made to exploit the highly structured nature of the ECG signal. Finally, it is worthwhile mentioning that the used metrics PRD and SNR may not always reflect the reconstruction quality. To illustrate their shortcoming, Fig. 9 plots a fragment of the original record 232 and its corresponding CS-compressed and reconstructed signal. The PRD value corresponding to each 512-sample window is shown underneath, while the related reconstruction error is plotted earlier. It is easily seen that window 19 exhibits a very large PRD value (i.e., very bad reconstruction). In fact, this window contains no ECG rhythm, just background noise that does not fulfill our sparsity assumption. Consequently, while bearing no relevant information, this window leads to a detrimental reduction of the measured signal quality in terms of PRD and SNR.

V. Power and Energy Consumption Measurements

The previous section described in details the characteristics of the embedded implementations of the two considered ECG compression algorithms and carefully motivated the various underlying tradeoffs and implementation choices. It also proposed a comparative study of these algorithms in terms of the signal reconstruction metrics (i.e., PRD and SNR) and their embedded memory usage. The present section further characterizes their execution time on the embedded Shimmer sensor node and the corresponding energy consumption.

Our experimental setup consists of a Shimmer node implementing embedded ECG compression and wirelessly transmitting the compressed data using its IEEE $802.15.4$-compliant radio to a remote base station (a desktop PC to which a second Shimmer node is connected). The communication between the node and the base station takes place according to the beacon-enabled mode of the IEEE $802.15.4$ MAC protocol [37] implemented in FreeRTOS [33]: the base station periodically sends a beacon to maintain node synchronization and to notify the ECG sensing node its guaranteed time slots (GTS), during which the node may send its data. A data packet is 127-Byte long with 13 Bytes of MAC overhead.

To measure the power consumption during operation, we placed a $10.3\,\Omega$ resistor in the power path of the platform. The voltage was then measured using an oscilloscope, and the corresponding current and power consumption were calculated. Fig. 10 shows the power consumption trace of a sensing node running a simple ECG streaming application, in which the raw ECG signal is forwarded without any onboard processing. Regarding the radio, three main phases can be distinguished: periodic beacon reception, low power (LP) mode, and transmission.

TABLE II
AVERAGE POWER CHARACTERIZATION (IN mW) OF SHIMMER

	CPU	Radio	Total
Radio reception (μC idle)	6.60	65.79	72.39
Radio reception (μC active)	16.80	65.79	82.59
Packet transmission (from μC to radio)	16.80	0.00	16.80
Packet transmission (actual radio transmission)	6.60	45.32	51.92
Sampling (μC and radio idle)	6.60	0.00	6.60
Sampling (μC active and radio idle)	16.80	0.00	16.80

TABLE III
AVERAGE TIMING CHARACTERIZATION (IN MS) OF SHIMMER

Beacon reception (radio in reception and μC idle)	1.39
Beacon reception (radio in reception and μC active)	0.97
Packet transmission (from μC to radio)	2.85
Packet transmission (actual radio transmission)	4.51
ACK reception	0.75

TABLE IV
NODE LIFETIME FOR "VERY GOOD" RECONSTRUCTION QUALITY WITH EMBEDDED COMPRESSION

	DWT	CS	No Comp.
Compression Ratio (%)	73	51	0
Code execution time (ms)	580	25	0
Packet Ready every ... (ms)	1099	605.9	296.9
Beacon Interval (ms)	3932	1966	983
Energy Consumption (mJ)	9.86	7.29	7.75
Life time (h) (280 mAh@3.7 V)	105.02	142.1	133.74

TABLE V
NODE LIFETIME FOR "GOOD" RECONSTRUCTION QUALITY WITH EMBEDDED COMPRESSION

	DWT	CS	No Comp.
Compression Ratio (%)	90	71	0
Code execution time (ms)	580	25	0
Packet Ready every ... (ms)	2968	1023.7	296.9
Beacon Interval (ms)	7864	3932	983
Energy Consumption (mJ)	9.67	7.05	7.75
Life time (h) (280 mAh@3.7 V)	107.05	146.8	133.74

During beacon reception, the radio first switches to reception mode before the beacon is expected, then receives it. After the beacon reception, the node's radio enters an LP mode until the start of its assigned GTS. Finally, during its GTS, the node transmits the ECG signal to the coordinator. Fig. 10 depicts the transmission of two data packets. After each packet transmission, the radio is seen to automatically go to reception mode, while it waits for an acknowledgment from the WBSN coordinator. Tables II and III, respectively, report the power consumption of the different working modes of the node and their associated timing information.

Additionally, to compress an ECG data vector of 512 samples (corresponding to a duration of 2 s at sampling rate of 256 Hz), the embedded CS code executes in 25 ms, whereas the DWT-based code requires 580 ms. Accordingly, Table IV compares the power consumption and the resulting node lifetime (calculated for a 280 mAh battery at 3.7 V) of the two considered ECG algorithms for a "very good" reconstruction quality and without compression. Table V does the same for a "good" reconstruction quality.

Interestingly, the reported results show that CS achieves a relative node lifetime extension of 35.3% and 37.1% over DWT-based compression, for "very good" and "good" reconstruction quality, respectively. These attractive figures confirm the superior energy efficiency of CS for a given reconstructed signal quality. This superior energy efficiency is achieved due to CS's lower complexity and shorter processing time. Furthermore, the results of Tables IV and V establish that DWT-based compression actually fails to achieve any lifetime extension, compared to the default alternative of streaming uncompressed ECG data. Alternatively, CS shows a 6.25% and 9.7% extension in node lifetime for "very good" and "good" reconstruction quality, respectively.

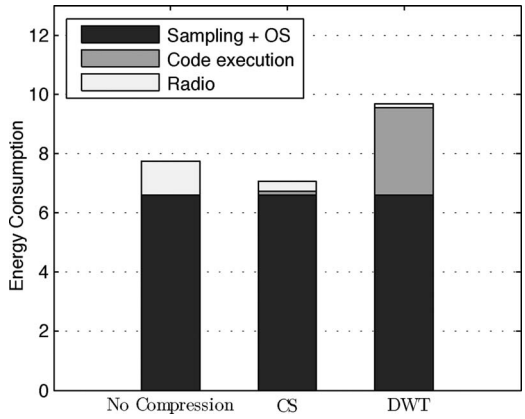

Fig. 11. Breakdown of energy consumption of Shimmer.

The latter results surprisingly evidence that even the computationally light CS-based ECG compression, which only consumes 1.25% of the CPU time while reducing the amount of data to transmit by 71% for "good" reconstruction quality, offers the limited lifetime extension of 9.7% with respect to raw ECG streaming. These results highlight that, for WBSN applications, the radio is not always responsible for most of the energy consumption of the node, as widely assumed. To understand the limited lifetime extension provided by embedded ECG compression, we further characterized the energy consumption breakdown of the Shimmer platform. In particular, Fig. 11 depicts the breakdown of the total energy consumption between the three main processes on the platform: 1) the wireless communication through the radio; 2) the code execution on the CPU; and 3) the sampling and the operation system (OS). As shown in Fig. 11, most of the energy consumption is due to sampling the ECG signal. Therefore, the sampling process can be very expensive in terms of energy consumption depending on the hardware implementation and the sampling frequency. In this particular case, for the considered sampling frequency of 256 Hz, it is not possible to turn off the internal reference voltage of the A/D converters between two consecutive samples, because the time that the internal voltage regulator takes to reach the appropriate level for the conversion is longer than the sample duration. These findings suggest the importance of ultra-low-power A/D converter design for wearable ECG WBSN systems. More interestingly, they confirm the relevance of pursuing "analog CS," which promises to significantly reduce the required ADC (i.e., sampling) rate, thus enabling a significant reduction in the energy consumption.

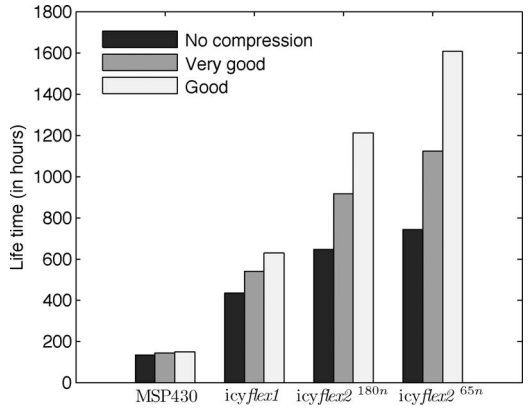

Fig. 12. Node lifetime estimation for CS compression on various processors.

Additionally, the promising results exhibited by CS in Tables IV and V could have been even more impressive had the MSP430 been optimized for ultra-low-power DSP operations. Since this generic low-power microcontroller is not, it effectively limits the achievable node lifetime extension through embedded signal preprocessing prior to wireless communication.

To investigate the full potential of CS to extend node lifetime, we consider its implementation on more competitive state-of-the-art ultra-low-power processors. Fig. 12 shows the node lifetime for the MSP430 and the recently introduced ultra-low-power DSP/MCUs icy$flex1$ [38] and icy$flex2$ [39]. These processors are designed in technologies that are more power efficient and are customized for DSP applications. The reported results for embedded CS on icy$flex$ series are calculated based on their published specifications, while wireless transceiver consumption is the same measured on the Shimmer. Strikingly, Fig. 12 shows that the node lifetime can be extended by 76% and 46% for "good" and "very good" reconstruction quality, respectively, on the currently available icy$flex1$. Furthermore, the lifetime extension is further increased up to 92% and 54% by the upcoming icy$flex2^{65\text{nm}}$ for "good" and "very good" reconstruction quality, respectively. These results establish the relevance of energy-efficient embedded CS-based ECG compression on sensor motes as a powerful means to extend their autonomy.

VI. Conclusion

This paper proposed a complete system-level comparison between a new CS-based and the state-of-the-art DWT-based embedded ECG compression algorithms. As expected, *nonadaptive* CS-based compression was found to exhibit inferior compression performance compared to its *signal-adaptive* DWT-based counterpart for a given reconstructed signal quality. The presented results, however, were obtained using the default basis pursuit denoise algorithm for sparse reconstruction, where no attempt has been made to exploit the highly structured nature of the ECG signal. Despite its inferior compression performance, CS-based compression was found to exhibit the best overall energy efficiency due to its lower complexity and reduced CPU execution time for Shimmer WBSN platform. Our results validate the suitability of CS for real-time energy-efficient ECG compression on resource-constrained WBSN motes. More importantly, they suggest the relevance of pursuing an "analog CS" implementation for the joint sampling and compression of ECG in the context of WBSN applications.

References

[1] World Health Organization. (2009). *Cardiovascular diseases* [Online]. Available: http://www.who.int/topics/cardiovascular_diseases/ 2011.
[2] Toumaz Technology. (2009). [Online]. Available: http://www.toumaz.com/public/news.php?id=92
[3] Shimmer Research. (2008). [Online]. Available: http://shimmer-research.com
[4] R. F. Yazicioglu, T. Torfs, J. Penders, I. Romero, H. Kim, P. Merken, B. Gyselinckx, H. J. Hoo, and C. Van Hoof, "Ultra-low-power wearable biopotential sensor nodes," in *Proc. IEEE EMBS (EMBC)*, Sep. 2009, pp. 3205–3208.
[5] S. M. S. Jalaleddine, C. G. Hutchens, R. D. Stranttan, and W. A. Coberly, "ECG data compression techniques: A unified approach," *IEEE Trans. Biomed. Eng.*, vol. 37, no. 4, pp. 329–343, Apr. 1990.
[6] L. Sörnmo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*: Elsevier/Academic Press: New York, 2005.
[7] M. B. Velasco, "Compression of electrocardiograms using cosine modulated filter banks and multi-resolution analysis (in Spanish)" Ph.D. dissertation, Univ. Alcalá, Spain, Dec. 2004.
[8] M. Hilton, "Wavelet and wavelet packets compression of electrocardiogram," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 5, pp. 394–402, May 1997.
[9] Z. Lu, D. Y. Kim, and W. A. Pearlman, "Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 7, pp. 849–856, Jul. 2000.
[10] B. A. Rajoub, "An efficient coding algorithm for the compression of ECG signals using the wavelet transform," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 4, pp. 355–362, Apr. 2002.
[11] R. Benzid, F. Marir, A. Boussaad, M. Benyoucef, and D. Arar, "Fixed percentage of wavelet coefficients to be zeroed for ECG compression," *Electron. Lett.*, vol. 39, no. 11, pp. 830–831, 2003.
[12] S. Aviyente, "Compressed sensing framework for EEG compression," in *Proc. IEEE Workshop Stat. Signal Proc. (SSP)*, Aug. 2007, pp. 181–184.
[13] S. Şenay, L. F. Chaparro, M. Sun, and R. J. Sclabassi, "Compressive sensing and random filtering of EEG signals using slepian basis," in *Proc. EURASIP EUSIPCO*, Aug. 2008.
[14] A. M. Abdulghani, A. J. Casson, and E. Rodriguez-Villegas, "Quantifying the feasibility of compressive sensing in portable electroencephalography systems," in *Proc. Int. Conf. Foundations Augmented Cognition, Neuroergonomics Oper. Neurosci. (FAC)*, 2009, pp. 319–328.
[15] (2009). [Online]. Available: http://focus.ti.com/docs/prod/folders/print/msp430f1611.html
[16] P. S. Addison, "Wavelet transforms and the ECG: A review," *Physiol. Meas.*, vol. 26, no. 5, pp. 155–199, 2005.
[17] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
[18] D. L. Donoho, "Compressed sensing," *IEEE Trans. on Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[19] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
[20] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
[21] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, pp. 1207–1223, 2006.
[22] D. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2845–2862, Nov. 2001.
[23] J. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 1030–1051, Mar. 2006.
[24] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1999.

[25] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.
[26] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Selec. Topics Signal Proc.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.
[27] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2004.
[28] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Nov. 1993, pp. 40–44.
[29] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.
[30] E. v. Berg and M. P. Friedlander. (June 2007). *SPGL1: A solver for large-scale sparse reconstruction* [Online]. Available: http://www.cs.ubc.ca/labs/scl/spgl1
[31] MIT-BIH arrhythmia database. (2005). [Online]. Available: http://www.physionet.org/physiobank/database/mitdb/
[32] Y. Zigel, A. Cohen, and A. Katz, "The weighted diagnostic distortion (WDD) measure for ECG signal compression," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 11, pp. 1422–1430, Nov. 2000.
[33] (2009). [Online]. Available: http://www.freertos.org
[34] (2009). [Online]. Available: http://focus.ti.com/docs/toolsw/folders/print/msp-cce430.html
[35] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *Allerton Conf. Commun., Control Comput.*, 23–26, 2008, pp. 798–805.
[36] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, Ö. Yılmaz. (2007, Oct). Sparco: A testing framework for sparse reconstruction. Dept. Comput. Sci., Univ. British Columbia, Vancouver, Tech. Rep. TR-2007-20, [Online]. Available: http://www.cs.ubc.ca/labs/scl/sparco/
[37] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE 802.15.4 Std., 2003.
[38] C. Arm, S. Gyger, J.-M. Masgonty, M. Morgan, J.-L. Nagel, C. Piguet, F. Rampogna, and P. Volet, "Low-power 32-bit dual-mac 120 w/mhz 1.0 v icyflex1 dsp/mcu core," *Solid-State Circuits, IEEE J.*, vol. 44, no. 7, pp. 2055–2064, Jul. 2009.
[39] C. Arm *et al.*, "The icyflex processor family," CSEM Scientific and Technical ReportTech. Rep., 2008.

Author's photographs and biographies not available at the time of publication.