Rate-Adaptive Compressed-Sensing and Sparsity Variance of Biomedical Signals

Vahid Behravan, Neil E. Glover, Rutger Farry,
Patrick Y. Chiang
Oregon State University
Corvallis, OR, USA
{behravav,gloverne,farryr,pchiang}@onid.oregonstate.edu

Mohammed Shoaib
Microsoft Research
Redmond,WA, USA
mohammed.shoaib@microsoft.com

Abstract—**Biomedical signals exhibit substantial variance in their sparsity, preventing conventional a-priori open-loop setting of the compressed sensing (CS) compression factor. In this work, we propose, analyze, and experimentally verify a rate-adaptive compressed-sensing system where the compression factor is modified automatically, based upon the sparsity of the input signal. Experimental results based on an embedded sensor platform exhibit a 16.2% improvement in power consumption for the proposed rate-adaptive CS versus traditional CS with a fixed compression factor. We also demonstrate the potential to improve this number to 24% through the use of an ultra low power processor in our embedded system.**

Keywords—adaptive compressed-sensing; low power wireless sensors; sparsity variance; embedded system

I. INTRODUCTION

Power consumption is one of the most critical constraints for sensor networks and future Internet-of-Things devices. The most power consuming block is the radio (transmitter), typically responsible for 60-80% of the total sensor's energy usage [1]. A known method to reduce the radio power consumption is transmitter duty cycling, sending data in quick bursts before rapidly powering down [2]. To obtain the largest benefit from transmitter duty cycling, the amount of sensor data transmitted with the radio must be reduced, such as using data compression.

Signal specific compression methods do exist (*e.g.* JPEG) that very efficiently decrease the amount of data required to recover the original signal [3]; However they typically are specific and cannot be applied to multiple sensor data types. Due to these limitations, a compression method generalizable to many different types of data is highly desirable.

Compressed-Sensing (CS) is a subsampling method that requires no a-priori information about the input signal as long as the input shows a property called sparsity (explained in detail below). Fig. 1 shows both traditional and compressed-sensing systems and their energy consumption specifications. The complexity of the receiver in a CS system is higher than its traditional counterpart; however the receiver typically does not have as tight of a power consumption constraint. On the other hand, the CS sensor node will exhibit significant energy

savings in the transmitter (radio), outweighing any additional energy consumed by implementing the compression block.
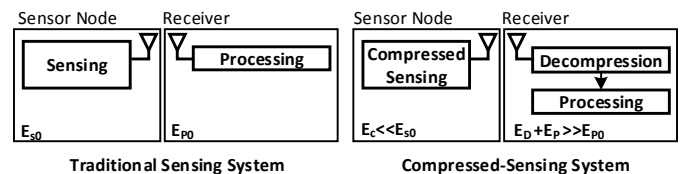

Fig. 1. A compressed-sensing system vs. a traditional sensing system and their energy consumption

Various methods have been previously proposed that implement a CS TX framework [4]. Unfortunately, all previous CS works assume a fixed and predetermined input signal sparsity. This assumption implies that the compression factor is statically fixed, and is not a function of a varying input signal; however, there is a trade-off between input sparsity and optimum compression factor [5]. An input signal with time-varying sparsity must adjust its compression factor continuously in order to achieve the lowest error rate during reconstruction. Using the signal sparsity to determine the sensor's optimal compression factor is critical if the sparsity of the signal varies significantly over time, across different sensor types. To our knowledge, there is no prior work that has studied the sparsity variance for several different biomedical signals.

In this paper, we first conduct a feasibility study on adaptive compressed sensing, investigating the sparsity variance of electrocardiogram (ECG), electroencephalogram (EEG) and electromyogram (EMG), using the physionet database [6]. Next, we propose an adaptive CS framework, adjusting the compression rate based upon the input signal's sparsity on-the-fly. Finally, we implement this rate-adaptive CS using a miniature, low-power wearable platform that captures ECG signals under varying sparsity conditions, calculate the sparsity variance, and adapt the CS rate.

This paper is organized as follows. Section II covers the theory of compressed-sensing, reconstruction and key system requirements. The requirements of an adaptive compressed-sensing system are then described in Section III. The miniature

wearable system developed to capture test signals and implement adaptive CS framework is then outlined in Section IV, and conclusions are summarized in Section V.

II. CS FRAMEWORK

A. Compressed-Sensing Background

Compression using CS is achieved using a simple matrix multiplication, where an uncompressed input vector f of size N multiplied by a measurement matrix $\mathbf{\Phi}$ of size M-by-N that produces a measurement vector y of size M. $\mathbf{\Phi}$ is a matrix of random numbers (*e.g.*, Bernoulli, Gaussian, uniform, etc.), such that y is a vector of random linear projections of f on $\mathbf{\Phi}$. CS can compress bio-signals with compression factors as large as 10x, which reduces the amount of data transmitted and therefore the total power dissipation of the sensor node by a similar factor [8]. To reconstruct the original signal f from the measured signal y in the receiver, we need to solve an underdetermined set of equations where the number of equations is much less than the number of unknown variables. In general, there is not a unique solution for these types of equations. However, CS theory shows that if the signal f is sparse in any basis (*e.g.* fourier transform, wavelet transform), then there are optimized methods to reconstruct the original signal with minimum error based on convex optimization [5]. In other words, if there is a transformation matrix $\mathbf{\Psi}$ such that $f = \mathbf{\Psi}x$ (and therefore $y = \mathbf{\Phi}f = \mathbf{\Phi\Psi}x$) and x is sparse, then reconstruction is possible. Fig. 2 summarizes the complete CS framework [9].
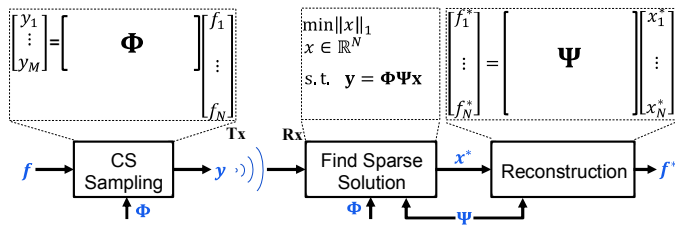

Fig. 2. Complete CS framework

B. Sparsity Number

In theory, sparsity of a vector x is defined as the number of non-zero values of x (this is called the norm-0 of x and represented as $\|x\|_0$). It is shown that if the number of measurements (M) is approximately $\mu\|x\|_0\log(N)$, an optimum reconstruction can be achieved with high probability where μ is a constant related to $\mathbf{\Phi}$ and $\mathbf{\Psi}$ [5][6]. The reconstruction process in the receiver, as shown in Fig. 2, consists of two steps. The first step finds an approximation of the sparse vector x by minimizing norm-1 of x, subject to the main equation $y = \mathbf{\Phi\Psi}x$ (norm-1 also known as l1-norm is defined as the sum of absolute values of vector x and is represented by $\|x\|_1$). The problem of minimizing the l1-norm has been shown to be solvable efficiently and requires only a small set of measurements (M<<N) to enable perfect recovery. After finding the optimized approximation of x (x^*), the second step reconstructs the original signal using matrix multiplication $f^* = \mathbf{\Psi}x$.

Fortunately, some biomedical signals have sparse representation in either gabor or wavelet domains [11], [12]. In general, any type of sensor data can be represented in its sparse form by using a method called dictionary learning. Based on the ideal definition of sparsity, it should be very easy to find the sparsity of a signal simply by counting the number of non-zero elements of the signal in the sparse domain. Unfortunately, for all real signals the representation of the input signal in sparse domain has non-zero elements. There are a few large values with other elements that are very small but not zero. To address this issue, Lopes in [13] introduced a new metric called *sparsity number*, as defined in Eq. (1) below, which is the lower bound of the ideal definition of sparsity:

$$sx = \frac{\|x\|_1^2}{\|x\|_2^2} = \frac{(\sum |x_i|)^2}{\sum x_i^2} \qquad (1)$$

To find the sparsity number of a signal we need to first transfer that signal to the sparse domain. For some signals the sparse domain is known; however, for other signals there is not a specific domain for this purpose and we must use dictionary learning to train a dictionary to transfer the signal f to its sparse representation $\mathbf{x}$.

C. Dictionary Learning

While ECG signal has known domain of sparsity, most other types of sensor signals do not have well-known domains where the signal is sparse. To extend the application of CS to different type of signals, it is necessary to have a general method for converting signals to a sparse representation. Fortunately, a method called dictionary learning exists for this purpose. Using a dictionary $\mathbf{D}$, a vector f (that is input signal) is represented as approximation of a linear combination of a few *atoms* where *atoms* are columns of $\mathbf{D}$ (such that the dictionary in fact is a matrix). To generate this dictionary, a training process called dictionary learning is required. Dictionary learning is the problem of finding $\mathbf{D}$ such that the approximation of many known vectors (the training set) are as good as possible given a sparseness criterion on the coefficients (*i.e.* allowing only a small number of non-zero coefficients for each approximation) [14]. Different methods are available for dictionary learning, such as KSVD [14] and MOD [15]. In this paper, we use the KSVD method to check the sparsity variance of EMG and EEG signals. Having $\mathbf{D}$ any new signal in the same family of training set can be represented in the sparse domain by solving Eq. (2), where γ controls the sparseness of the answer:

$$x^* = arg\ \min_x \|x\|_p + \gamma\|f - Dx\|_2^2 \qquad p \in \{0,1\} \quad (2)$$

III. SPARSITY-AWARE COMPRESSED SENSING

The number of measurements (M), which also defines the compression factor (M/N), is related to the sparsity level of the input signal. A trade-off exists between the compression factor and the accuracy of the reconstructed signal. Increasing the compression factor creates more errors in the reconstructed signal, while lowering the compression factor increases the amount of transmitted data without any benefit to the reconstruction. Fig. 3 shows the output of a sparse optimization block (x^*) for a CS system with a fixed compression factor of 16 for two different input signals. The first signal is a single tone (that is sparse in the Fourier domain) and the second signal is the summation of 5 sine waves with different

frequencies. It is clear that by changing the statistics of the input signal we will see a large difference between the reconstructed sparse signal ($\mathbf{x}^*$) and actual sparse representation of the input signal in the sparse domain. This results in a large error in the output of the reconstruction algorithm.
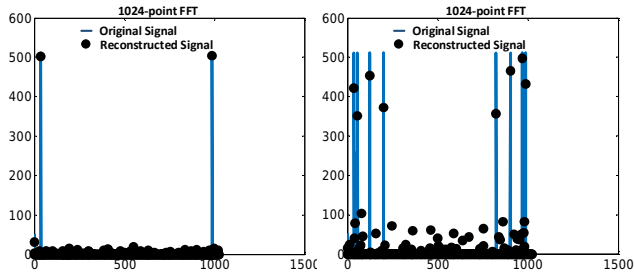

Fig. 3. Effect of sparsity on reconstruction error for two different signals

Obtaining updated information about the sparsity of the input signal will help us to adjust the compression factor to the optimum value. Using sparsity number as a key to find the optimum compression factor is only beneficial if the sparsity of the input signal changes sufficiently over time and for different sensor data types. This means that sparsity variance is a key feature for adaptive compressed sensing.

A. Sparsity Variance

If sparsity of an input signal does not change significantly over time, we can use a fixed compression factor that corresponds to the type of input signal. To the best of our knowledge, we have found no prior works have investigated and quantified the sparsity variance for different biomedical data types. In this work, we analyze the sparsity variance for three different types of clinical biomedical signals (ECG, EEG and EMG), using the nsrdb, mitdb, chbmit and drivedb databases of physionet [7]. We also obtained the sparsity variance of ECG data captured by own embedded system, platform. In each case, several signals divided in blocks of 256 samples and then transferred to sparse domain to calculate the sparsity numbers. Finally we plot the histogram of the sparsity number and calculate mean and standard deviation of distribution. Fig. 4 shows the simulation results, showing that the sparsity changes drastically and that a fixed compression factor is not a good choice.

B. System Models for Sparsity-Aware Compressed-Sensing

Fig. 5 shows two possible methods for implementing rate-adaptive compressed-sensing, using the input signal sparsity to adjust the compression factor. In the first system (Method 1), the receiver calculates the sparsity and transmits it to the sensor node of the system. The sensor then adjusts its compression factor based on this information. Because the receiver already has the sparse representation of signal, this system can easily implement CS rate adaption in the sensor node; however, one limitation with this system is the round trip delay, or the latency required to calculate the sparsity in the receiver and the transmission delay to send the result back to the sensor. If this latency is large, the sensor node must buffer recently acquired data until it has received the most up-to-date estimate of sparsity from the receiver. Another problem is that the calculation of sparsity is performed on the current data block;

however this result is then used to compress the next data block, which may have a different sparsity
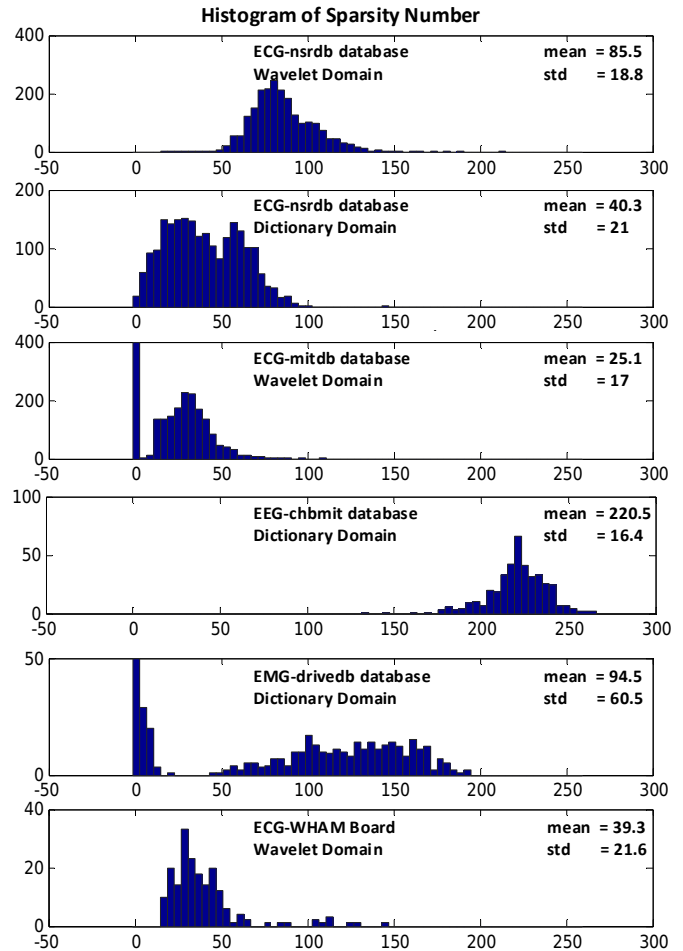

Fig. 4. Sparsity variance for different type of biomedical signals
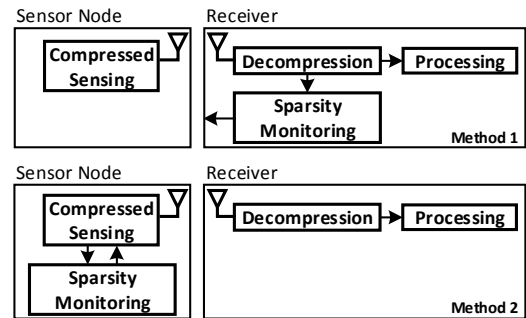

Fig. 5. Two possible methods for adaptive CS system implementation

The second system (Method 2) calculates the sparsity of the input data on the sensor node of the system. This eliminates possible problems associated with the round trip delay, and also allows for the calculated sparsity of each data block to be used in the compression of that same block. Unfortunately, this method requires additional power to operate the sparsity monitoring block, which may not be available on the energy-constrained sensor node. In addition, this system has no direct access to the sparse representation of the input signal, and therefore requires a sparsity estimation block that obtains the

sparsity of the input data directly from the time domain signal. To our knowledge, this is currently an unsolved problem and hence, we evaluate the feasibility and performance of only Method-1 here.

IV. Experimental Results

To analyze the performance of this proposed rate-adaptive CS system, we developed an experimental wireless monitoring platform, called the Wearable Health and Activity Monitor (WHAM). We implemented the compression on the WHAM sensor node, while the corresponding reconstruction algorithm is performed in an iOS iPhone/Mac application used as the receiver device. We implemented the GPSR algorithm [16] for reconstruction, which is faster than other reconstruction algorithms like LASSO [17]. We particularly were interested in investigating the energy consumption of the wearable sensor portion of the system with and without the use of adaptive data compression. Minimizing the delay associated with the reconstruction algorithm is critical because it directly affects the size of the buffer needed at the sensor node, before the receiver can calculate and send back the sparsity number needed to compress the next data block on the sensor.

A. WHAM System Overview

The WHAM platform is a low-cost, reconfigurable, wearable, wireless platform designed specifically to collect, process and display data related to the user's activity and health. Utilizing wireless Smartphone connectivity using Bluetooth, the WHAM was designed to be operated by a wide variety of users. The complete system includes a functional custom hardware, a fully functioning iOS application, which can control and display data from the sensor from a variety of Apple products. The system weighs just 9.5 grams and is small enough to be easily stitched into clothing. Using a small CR2032 coin cell battery, the WHAM can provide continuous streaming ECG data collection for over 100 hours. The system currently consists of 3 physical components: The main board, the battery, and the flexible ECG board. The dime-sized main board consists of a stack of two miniature two-layer PCBs which contain the SoC, accelerometer, printed circuit board antenna, and other necessary passive components. The flexible ECG board contains the electrodes and instrumentation amplifier necessary for ECG data acquisition. Because of the flexibility and simplicity of its development platform, iOS was chosen as the operating system. The iOS application plots the real-time ECG graph and Beats per Minute (BPM) calculation. Fig. 6 shows the WHAM main sensor board, and top and bottom sides of the flexible electrode ECG patch.
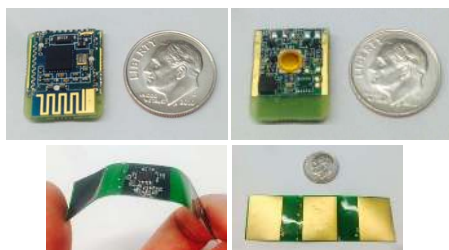

Fig. 6 WHAM main board (top), Flexible electrode patch (bottom).

B. Implementation Methodology

As illustrated in Fig. 2, CS compression is the product of a block of N input samples (here N=256) to a random matrix. We implemented this matrix multiplication at the sensor node. For simplicity, the random matrix Φ has a fixed size of 192×256, so it always generates 192 measured values from every 256 input samples. Based on the feedback from the receiver, the sensor only transmits the required number of measured samples. For example, if M=128 then only the first 128 samples will be transmitted. This is equivalent to multiplying the input vector with the first 128 rows of Φ. In the receiver we also use the first 128 rows of Φ for reconstruction. The receiver constantly checks the sparsity number of the data it is receiving. If the sparsity of the input signal changes by more than 50% when compared to the previous block of data, it will change the value of M. Even for a normal ECG signal the sparsity number may change in a range of 40%. Analysis of the data sets investigated in this work found that a 50% variation in sparsity number is a significant change in the input signal (*e.g.* ECG abnormality such as atrial fibrillation).

For our wireless radio protocol, we adopted Bluetooth Low Energy (BLE) as it is one of the most commonly used low-power sensor radio standards, and heavily utilizes duty cycling. A BLE connection between two devices is made every T seconds where the devices exchange some data. After this exchange both devices disconnect and power down their radios for a period of time. The receiver portion of the system (*e.g.* the App running on the iPhone) is able to set T. Unfortunately, the system we developed for this work has limitations on the possible values of T and the number of samples that each BLE transmission is able to send. For this reason we only use a limited number of values for M, as shown in TABLE I below, where CF stands for the compression factor.

TABLE I. Possible Options For Choosing M

T	N=256, ADC Sample Rate=200 sample/sec								
	1 sample per BLE Transm.			2 samples per BLE Transm.			3 samples per BLE Transm.		
	Tx Rate	CF	M	Tx Rate	CF	M	Tx Rate	CF	M
20ms	50	4	64	100	2	128	150	1.33	**192**
30ms	33	6	42	66	3	84	99	2	**128**
40ms	25	8	32	50	4	64	75	2.66	**96**
50ms	16.6	10	25	40	5	50	60	3.33	75

In our implementation of adaptive CS, we transmit three samples for each BLE transmission such that the possible values of M are 192, 128, 96, 75 and 63. Initially the system starts with a default value of M (*e.g.* M=96 that is equal to a compression factor of 256/96=2.7). During normal operation, anytime the receiver detects a 50% change in the sparsity number it adjusts T appropriately such that the sensor automatically switches to the next higher or lower M value.

C. Measurments Results

To calculate the average power consumption of the system, we used a 1.3Ω resistor in series with the battery to measure the voltage drop across it. Fig. 7 contains the system setup for

power measurements as well as real-time signal acquisition. Transmission of data occurs in very short time durations but with high current consumption.
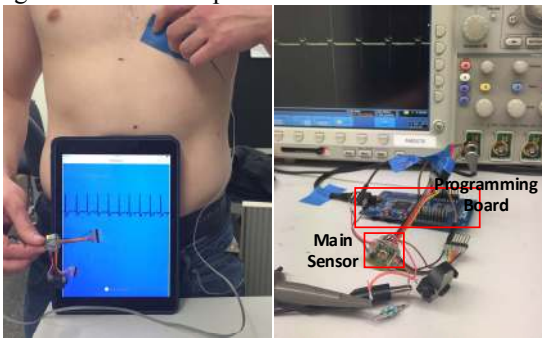

Fig. 7. System setup for power measurement and signal acquisition

Our measurements show that the average current changes negligibly when sending 1, 2 or 3 samples with each BLE transmission. For this reason we decided to control the CS compression factor by controlling the turn-on period (T). To demonstrate and analyze the performance of adaptive CS system, we selected two sections of an ECG signal captured by the WHAM system that showed a significant change in their sparsity number. After this signal is stored in the sensor, the data is then compressed and transmitted to the receiver, where it is then reconstructed and analyzed. Analysis of the reconstructed data in the receiver provides a new estimate of the sparsity number, where the sparsity number is used to set the new transaction time (T) of the BLE connection. The reconstructed signal is then appropriately scaled and analyzed using MATLAB in order to compare it with the original signal to calculate the reconstruction error. The reconstruction error is determined using the equation for Signal to Error Ratio (SER), as defined in Eq. (3) where f, f^* are the original and reconstructed signals, respectively:

$$SER = 10 \log_{10}(\frac{\|f\|_2^2}{\|f - f^*\|_2^2}) \qquad (3)$$

Fig. 8 shows the original and reconstructed signals. The first two blocks are normal with low sparsity where the system transmits 3 samples every 40ms, equivalent to a compression factor of 2.7. Based on the information in TABLE I, in order to transmit these two blocks, M must be set to 96. In the 3rd block of ECG, motion artifact noise is observed. Due to this noise, its sparsity number changes by more than 50% when compared with ECG block 2. The receiver senses this change and adjusts the BLE transmission time to 30ms, equivalent to a compression factor of 2 (M=128), to compensate for this sparsity change. The sensor then applies this lower compression factor, obtaining a more accurate reconstruction in the receiver. Note that this system always exhibits a one block delay, as the 3rd block is compressed with the compression factor calculated from the 2nd block. Because of this the system experiences some error upon reconstruction.
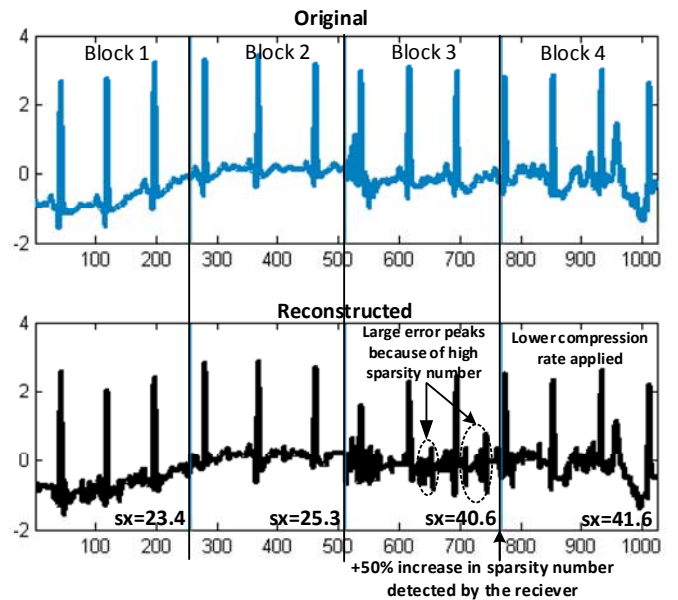

Fig. 8. Original vs. reconstructed signal for adaptive CS system

TABLE II summarizes the measurement results. First we measure the average power consumption in 3 modes: Adaptive CS versus traditional CS with fixed compression factors of 2 and 2.7. We calculate the average power consumption of the adaptive CS system assuming 50% of the time the system experiences signals with a high sparsity number (*e.g.* noisy or abnormal). These results show that when targeting more than 10dB of SER, adaptive CS exhibits the same performance as a fixed compression rate system (with M=128) but consumes 14% less power. If the target is a minimum SER of 16dB for all input blocks, we will then need a fixed value of M=192 (TABLE III). In this case, using adaptive CS that switches between M=192 and M=128 gives us the same performance but with 16.2% less power consumption. The average power consumption of the adaptive system is based on the assumption that the signal input is a normal signal with low sparsity number for a long time duration, until it abruptly changes to a signal with high sparsity number (*e.g.* because of abnormalities or motion artifact) and remains this way for a long time period. In this case, we can approximate the average power of the adaptive system using the formula in Eq. (4), where $P_{avg-fixed1}$ and $P_{avg-fixed2}$ are the average power of the CS system with old and new compression factors:

$$P_{avg-adaptive} = \frac{1}{2}(P_{avg-fixed1} + P_{avg-fixed2}) \qquad (4)$$

TABLE II. PERFORMANCE OF THE ADAPTIVE CS SYSTEM FOR SER$_{min}$=10dB

CS method	SER(dB)				Average Power
	Block1	Block2	Block3	Block4	
Adaptive	13.2	12.5	4.9	**10.5**	**2.5mW**
Fixed (M=128)	14.9	14.7	10.2	**10.5**	2.9mW
Fixed (M=96)	13.2	12.5	4.9	**6**	2.1mW

TABLE III. Performance of the Adaptive CS System for SER$_{min}$=16dB

CS method	SER(dB)				Average Power
	Block1	Block2	Block3	Block4	
Adaptive	14.9	14.7	10.2	**16.3**	**3.6mW**
Fixed (M=192)	22	19	16.1	**16.3**	4.3mW
Fixed (M=128)	14.9	14.7	10.2	**10.5**	2.9mW

Fig. 9 shows the measured average power consumption of the sensor for different transmission rates of BLE. Unfortunately, the results show that the power consumption of the embedded processor dominates after some point, such that decreasing transmission rate (increasing T) does not improve power consumption proportionally. Using an ultra low power processor we could expect 1.5mW power consumption for T=40ms giving average power of 2.2mW for the adaptive CS system of TABLE II, equivalent to a 24% improvement when compared with a fixed CS system with M=128.
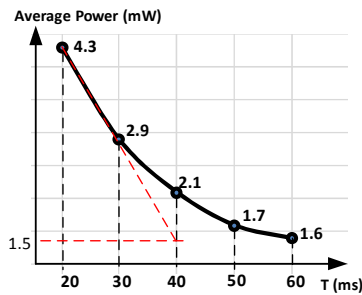

Fig. 9. Measured average power for different BLE transmission rates

Finally, we measured the execution time of the reconstruction algorithm on the receiver (a MacBook Air), where the worst case reconstruction latency was found to be 24 ms, or less than the time required by the ADC to generate 5 samples.

V. Conclusion

In this paper we propose an adaptive compressed-sensing system that enables real-time adjustment of the compression factor, depending on the variance in the sparsity number of the input signal. We also investigated the sparsity variance of different biomedical signals to provide justification for implementing an adaptive compressed-sensing system. We then validated our proposed system through experiments conducted on a custom embedded hardware. We showed that our adaptive compressed-sensing system can reduce power consumption by 16.2%, without any major change on the sensor node. Despite this initial success, the criteria for which the adaptive system should change the compression factor has not been fully explored and is the subject of future work Furthermore, the relationship between sparsity and the desired compression factor is also still unknown for real biomedical signals, and requires future research.

References

[1] F. Chen, "Energy-efficient Wireless Sensors: Fewer Bits, More MEMS," PhD Thesis, Dept. Electr. Eng. Comp. Sc., Massachusetts Inst. Technol., Cambridge, MA, USA, Sep. 2011.

[2] J. Cheng; et al., "A near-threshold, multi-node, wireless body area sensor network powered by RF energy harvesting," Custom Integrated Circuits Conference (CICC), 2012 IEEE , vol., no., pp.1,4, 9-12 Sept. 2012

[3] S.K. Mukhopadhyay, M. Mitra, "An ECG data compression method via R-Peak detection and ASCII Character Encoding," Computer, Communication and Electrical Technology (ICCCET), 2011 International Conference on , vol., no., pp.136,141, 18-19 March 2011.

[4] F. Chen, A.P. Chandrakasan, V. Stojanovic, "A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors," Custom Integrated Circuits Conference (CICC), 2010 IEEE , vol., no., pp.1,4, 19-22 Sept. 2010

[5] D. Donobo, "Compressed Sensing," Trans. on Information Theory, 2006.

[6] Candes, E.J.; Tao, T., "Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?," *Information Theory, IEEE Transactions on* , vol.52, no.12, pp.5406,5425, Dec. 2006

[7] A. L. Goldberger, et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals,[Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215], 2000.

[8] http://www.physionet.org/physiobank/database/

[9] D. Gangopadhyay, et al., "Compressed Sensing Analog Front-End for Bio-Sensor Applications," Solid-State Circuits, IEEE Journal of , vol.49, no.2, pp.426,438, Feb. 2014.

[10] O. Abari, F. Chen, F. Lim, V. Stojanovic, "Performance trade-offs and design limitations of analog-to-information converter front-ends," Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on , March 2012.

[11] S. Miaou and S. Chao, "Wavelet-Based Lossy-to-Lossless ECG Compression in a Unified Vector Quantization Framework," IEEE Transactions on Biomedical Engineering , vol. 52, pp. 539-543, 2005.

[12] S. Aviyente, "Compressed Sensing Framework for EEG Compression, " IEEE 14th Workshop on Statistical Signal Processing , 2007.

[13] M. Lopes, "Estimating Unknown Sparsity in Compressed Sensing," International Conference on Machine Learning (ICML), 2013.

[14] M. Aharon, M. Elad, A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," Signal Processing, IEEE Transactions on , vol.54, no.11, pp.4311,4322, Nov. 2006.

[15] http://www.ux.uis.no/~karlsk/dle/

[16] http://www.lx.it.pt/~mtf/GPSR/

[17] http://statweb.stanford.edu/~tibs/lasso.html