Energy Scaling in Multi-tiered Sensing Systems Through Compressive Sensing

Mohammed Shoaib, Jie Liu, and Matthai Phillipose
Microsoft, One Microsoft Way, Redmond WA 98052
Email: {moshoaib,luij,mathaip}@microsoft.com

***Abstract*— High functional complexity is leading us towards new architectures for sensing systems. Multi-tiered design is one among the many emerging alternatives. Such architectures bring new opportunities for effective system-level power management. For instance, varying one/more tier-level parameters can provide substantial end-to-end energy scaling. In this paper, we review an existing approach that shows how one such parameter, namely data compression, can help us scale energy at the cost of algorithmic accuracy. The methodology is driven by a case study of inferring the onset of seizure events directly from compressively-sensed electroencephalograms. Results from an integrated circuit implementation have shown tier-level computational energy scaling in the range 1.2-214 μJ depending on the amount of compression (2-24×) and inference accuracy (sensitivity, latency, and specificity of 91-96%, 4.7-5.3 sec., and 0.17-0.30 false-alarms/hr., respectively). The projections we make in this paper show that for similar systems, compressive sensing, through this approach, has the potential to prolong battery lives of all tiers by up to 5×.**

I. Introduction

Sensing systems are starting to monitor increasingly complex physical entities, such as the human body and the brain. They serve the purpose of not only enabling offline analytics but also providing real-time feedback to control tools [1]. A key characteristic of such systems is that they aggregate enormous amounts of data through continuous sensing. Thus, it is important that these systems distill and present only informative instances of data to the applications they drive.

In a typical sensing system, sensors that collect data are placed close to natural phenomena since they then provide richer- and higher-quality signals. However, this placement often imposes strict storage and energy constraints [2]. Thus, sensing nodes themselves can support little to no computation. A more effective way of handling the sensed data is thus to offload it to another device whose storage- and processing-energy constraints are somewhat relaxed. This device is also a part of the sensing system. Custom or general-purpose processing platforms such as smartphones are typical options. If the end-goal of the system is to support offline analytics, we can either store the data on these second-tier devices or relay it to a cloud server or remote base station. However, if the end-goal is to provide real-time feedback, we need to compute on either the device itself or at the base station. These devices typically employ long-haul radio links such as a cellular network (or even WiFi) to communicate with the base station. The latter option thus entails huge overheads in latency and energy consumption. The device is also energy constrained (but not as severely as the sensors), which make even the former option less feasible. The middle ground is where we compute partially on the device just enough to select informative data instances, which are then relayed to a base station for further analysis. This trade-off leads to a multi-tiered architecture for the sensing system that is shown in Fig. 1. There arise three distinct tiers. The first (sensing) tier is the most energy constrained. The second (processing) tier has a somewhat higher energy budget than the first but is still energy constrained. The third (storage) tier is faced with the least (or no) energy constraint.

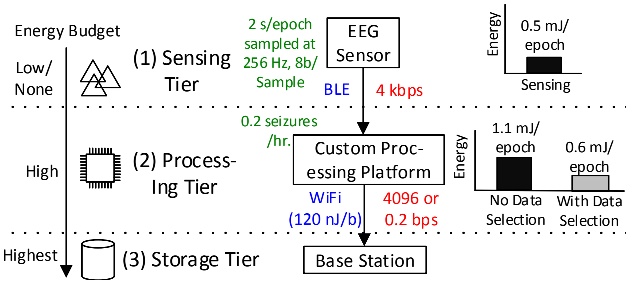


Fig. 1. In a multi-tiered seizure-detection system, data selection reduces energy consumption of the processing tier by about 2×..

Impact of data on energy consumption. To explore this aspect, we consider a representative sensing system that acquires electroencephalogram (EEG) signals from a head-worn device and processes them on a custom-computing platform. The end goal in this system is to detect seizure events in real time. Since only the first two tiers are energy constrained, we do not consider the third tier in the rest of the discussion.

Fig. 1 shows the energy consumption of each tier. We compare two cases: one where all data is relayed by the processing tier to the storage tier and the other where the processing tier only transmits selected data instances to the base station. This system uses bluetooth low energy (BLE) [3] and WiFi [4] for communication. We observe that the data rates are higher on the second link (between the processing and storage tiers) in the former case leading to higher energy consumption of the processing tier. Clearly, *the amount of data that moves between tiers has a significant impact on energy consumption*. Thus, reduction of data on all links has the potential to reduce the overall system energy. In emerging systems, the processing tier already performs data selection that helps reduce data on the link between the processing and storage tiers [5]. Reducing data between the sensing and processing tiers can further reduce the energy consumption. One promising approach that can help us achieve this reduction is data compression [6]. In this paper, we review existing work that exploits data compression to reduce system energy.

Energy scalability. Reduction in energy consumption is not always free. It comes at the cost of one or more of the system-level parameters. For instance, selecting data instances

978-1-4799-3286-3/14/$31.00 ©2014 IEEE

to transmit from the processing tier to the storage tier allows us to save energy at the cost of information content. At each tier, voltage/frequency scaling allows us to trade energy for performance [7], while techniques like approximate computing allow us to trade energy for accuracy [8]. There are also approaches that employ selective sensor/power gating to save energy at the cost of data quality [9], [10]. Thus, from an applicaiton point of view, energy reduction may not always be desirable. In fact, *it is more powerful if we have knobs that provide us control over the system energy consumption*. Thus, depending on application needs, a designer can invoke low/high energy modes of operation thereby scaling the system energy consumption up or down at whim.

Scalability through data compression. Data compression provides us with a specific knob (*i.e.*, amount of compression) that allows us to save communication energy between all tiers at the cost of some extra computation (required for the compression process). In certain cases, there is also an energy cost for decompression, which may be necessary to perform prior to data-selection in the processing tier. Compressive sensing is an interesting data-compression technique that does not require a lot of energy to compress the data, thus it is ideally suited to be used in the sensing tier [11]–[13]. This technique is applicable to a broad range of signals and allows us to compress data by large amounts [14]. However, it raises with two major issues: (1) the data get altered in the compressed domain due to the random projections involved and (2) the data reconstruction energy is very high [15]. Thus, after data compression in the sensing tier, if we want to perform computations for data selection in the processing tier (*e.g.*, to reduce the communication energy of the processing tier), we would need to reconstruct the signal. However, the high energy overheads for reconstruction can rapidly eclipse the gains provided by compressive sensing.

Recent work has shown that through new linear transformations, the processing tier can extract useful features directly from compressively-sensed data [16]. Further, such features can also be used effectively in machine-learning algorithms to perform data selection. Thus, using this approach, data are compressed in the sensing tier and remain compressed throughout. In fact, these transformations enable us to achieve energy scaling at the cost of not only some extra computation (required for compression) but also at the cost of some loss in data-selection accuracy. The energy savings come not only because the decompression energy overheads are eliminated but also since the number of data samples that need be analyzed by the processing tier get reduced (*i.e.*, by an amount equal to the compression factor, ξ). In this paper, we review this work along with an IC implementation that demonstrates a processor that takes advantage of the potential energy scalability through various circuit-design techniques [17]–[19]. Thus, the IC that we discuss illustrates a new approach to system-level power management.

II. Background

In this section, we present background on the BLE communication protocol, compressive sensing, and seizure detection using EEG signals. We also describe transformations that allow direct analysis of compressively-sensed EEG.

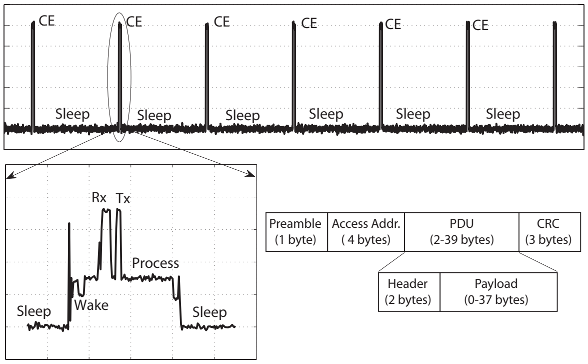


Fig. 2. BLE communication entails multiple CEs. Each CE comprises nine states and is able to handle a payload of up to 37 bytes.

A. Bluetooth Low-energy Radio Protocol

BLE is a 2.4 GHz, industrial sensing and medical (ISM) band radio protocol that uses Gaussian frequency shift keying (GFSK) modulation across 40 channels, spaced at 2 MHz [3]. The protocol allows periodic receiption/transmission of data packets to achieve an over the air data rate of up to 1 Mbps. A BLE packet breakdown is shown in Fig. 2. It comprises a 1 byte preamble, 4 byte sync word, 1 byte protocol data unit (PDU) header, 1 byte PDU length, 0-37 byte payload and 3 byte cyclic redundancy check (CRC) code. Further, a connection event (CE) is defined as an event during which one BLE packet is transmitted/received. To transmit/receive more than 37 bytes of data, multiple CEs are established between a BLE receiver and transmitter (see Fig. 2). A CE comprises the following nine states: (1) *wake-up*: radio wakes from sleep, (2) *pre-processing*: radio prepares to send or receive data, (3) *pre-Rx*: radio turns on in preparation of reception/transmission, (4) *Rx*: radio receiver listens for packets, (5) *Rx-to-Tx transition*: receiver stops, and radio prepares to transmit a packet, (6) *Tx*: radio transmits a packet, (7) *post-processing*: radio processes the received packet and sets up the sleep timer in preparation for the next CE, (8) *pre-sleep*: radio prepares to go into sleep mode, and (9) *sleep*: radio goes into sleep mode.

B. Compressive Sensing

Compressive sensing is a technique that allows us to multiply an N-sample signal by an $M \times N$ projection matrix $\mathbf{\Phi}$ to create an M-sample signal (with $M \ll N$, $\xi = N/M$) [14]. Further, a $\mathbf{\Phi}$ whose elements are set to ± 1 randomly with uniform probability allows us to recover the N samples from the M samples with high probability [20]. Such a choice for $\mathbf{\Phi}$ enables low-energy compression, applicable to a broad range of signals; this has recently been exploited in biomedical sensing systems [13]. However, reconstruction is energy intensive [15], which has typically limited the functionality of the processing tier to relaying raw compressed data to the storage tier [13].

C. Seizure Detection using EEG Sampled at Nyquist Rate

Fig. 3 (top panel) shows a multi-tiered sensing system for seizure detection that we consider in this paper. The Nyquist-domain algorithm is based on [21]. Fig. 3 (a) shows the computations involved in the processing tier. A two-second epoch from one EEG channel is processed using eight band-pass finite-impulse-response filters (BPFs) with passbands of 0-3 Hz, . . ., 21-24 Hz. The spectral energy from each filter is

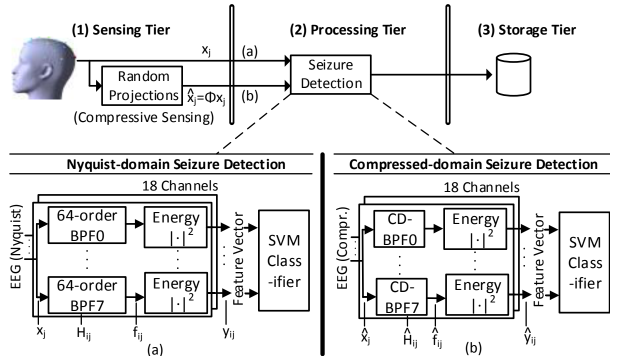


Fig. 3. A multi-tiered sensing system for seizure detection. Random projections in the sensing tier enable data compression. The processing tier involves feature extraction and classification using an SVM. To enable compressed-domain detection, BPFs $\mathbf{H_{ij}}$ are transformed to the CD-BPFs $\mathbf{\hat{H}_{ij}}$.

then represented by summing the squared value of the output samples to form a feature vector (FV), which is then used for classification by an support-vector machine (SVM) classifier.

In [21], the detector has been validated on 558 hrs. of EEG (corresponding to 148 seizures) from 21 patients in the CHB-MIT database [22]. For every patient, up to 18 channels were processed using eight BPFs per channel, leading to an FV of length 144. The performance of the detector was characterized using the metrics of specificity, sensitivity, and latency. Latency refers to the delay between an expert-identified electrographic onset and the seizure onset recognized by the detector. The detector achieved an average latency, sensitivity, and specificity of 4.59 sec., 96.03%, and 0.1471 false alarms per hour, respectively. We next describe the transformations proposed in [17] that allow us to perform seizure-detection directly with compressively-sensed EEG.

D. Seizure Detection using Compressively-sensed EEG

Fig. 3 (top panel) shows the use of random projections in the sensing stage to achieve data compression [*i.e.*, to obtain the compressively-sensed signal $\mathbf{\hat{x}_j}$ (= $\mathbf{\Phi x_j}$)]. Recall that compression saves communication energy between the sensing and processing tiers. Fig. 3 (b) shows the computations that must be performed in the processing tier to enable seizure detection without signal reconstruction. The key aspect that distinguishes the compressed-domain algorithm from the Nyquist domain [Fig. 3 (a)] is the use of compressed-domain band-pass filters (CD-BPFs) in place of the BPFs. We next describe, how to obtain the CD-BPFs from the BPFs.

In the Nyquist domain, the computations performed by the i^{th} BPF on every epoch of the j^{th} EEG channel to compute the filtered signal $\mathbf{f_{ij}}$ can be formulated as a matrix multiplication, namely of an input signal $\mathbf{x_j}$ by a matrix $\mathbf{H_{ij}}$ [Fig. 3 (a)]. An FV is then derived using the inner product $\mathbf{f_{ij}^T f_{ij}}$. Given this feature-extraction process, the authors in [17] demonstrate how to derive corresponding CD-BPF matrices $\mathbf{\hat{H}_{ij}}$ such that the resulting compressed-domain inner products $\mathbf{\hat{f}_{ij}^T \hat{f}_{ij}}$ are approximately equal to the corresponding Nyquist-domain FVs [*i.e.*, $\mathbf{f_{ij}^T f_{ij}}$].

The approach that the authors follow aims to construct a matrix $\mathbf{\hat{H}_{ij}}$ that operates on $\mathbf{\hat{x}_j}$ (= $\mathbf{\Phi x_j}$) to obtain a projection $\mathbf{\hat{f}_{ij}}$ (= $\mathbf{\Theta f_{ij}}$), where $\mathbf{\Theta}$ is an auxiliary matrix of dimensionality $N/\nu \times N/\xi$, with $\nu \geq \xi$. By invoking the Johnson-Lindenstrauss (JL) guarantees [23], they demonstrate that the resulting inner product, $\mathbf{\hat{f}_{ij}^T \hat{f}_{ij}}$, forms a good estimate of $\mathbf{f_{ij}^T f_{ij}}$ if: (1) $\mathbf{\Theta} = \mathbf{\hat{H}\Phi H^{-1}}$ and $\mathbf{\hat{H}_{ij}} = \mathbf{S^{-1}V^T}$ (exact solution), or (2) $\mathbf{\Theta} = (\mathbf{\Phi H^{-1}})^\mathbf{T}\mathbf{\hat{H}}$ and each row of $\mathbf{\hat{H}}$ is derived from the normal distribution $N(0, \mathbf{\Sigma})$ (approximate solution). In these solutions, $\mathbf{\Sigma} = \mathbf{VS^{-2}V^T}$; $\mathbf{S}$ and $\mathbf{V}$ are diagonal and unitary matrices, respectively, obtained from the following singular value decomposition (SVD): $(\mathbf{\Phi H^{-1}})^\mathbf{T} = \mathbf{USV^T}$. Further, in both cases, the CD-BPFs $\mathbf{\hat{H}_{ij}}$ are of dimensionality $N/\nu \times N/\xi$, with $\nu \geq \xi$.

Algorithmic Performance. Fig. 4 shows that for the exact solution, performance very close to the Nyquist-domain seizure detector is retained up to large values of ξ. Fig. 5 shows that in the case of the approximate solution, for any given value of ξ, the performance begins to degrade gradually as we increase ν. Note that the multiple local minima shown in the contour plots occur since performance metrics are dependent on one another; attempting to optimize one metric results in a degradation in others. Ideally, it is preferred to optimize all metrics simultaneously, which is achieved by the exact solution.

III. Energy Scaling through Compressed-domain Processing

In this section, we present circuits for compressive sensing and CD-BPF computation that help achieve energy scalability in the sensing and processing tiers, respectively. The measurement results are from an IC that was prototyped in a 0.13μm

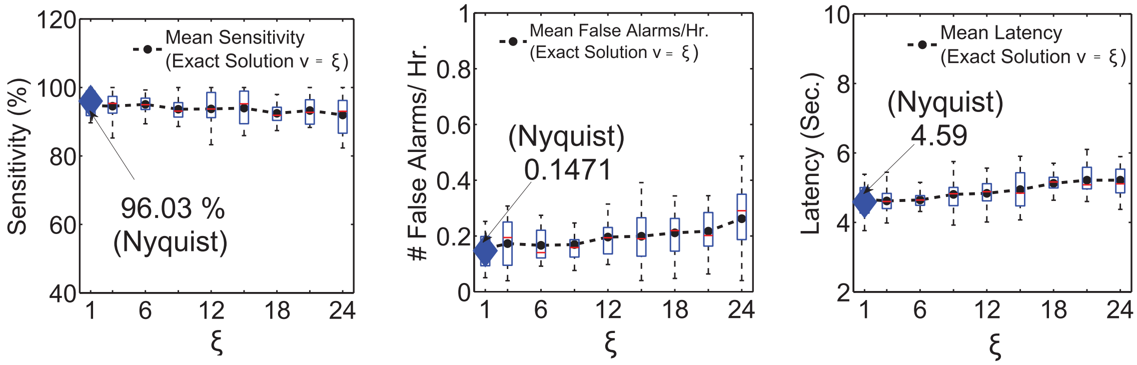


Fig. 4. Figure reproduced from [19]: Performance of the compressed-domain seizure detection algorithm using the exact solution (shown over 558 Hrs. of EEG data from 21 patients) is maintained up to large ξ.

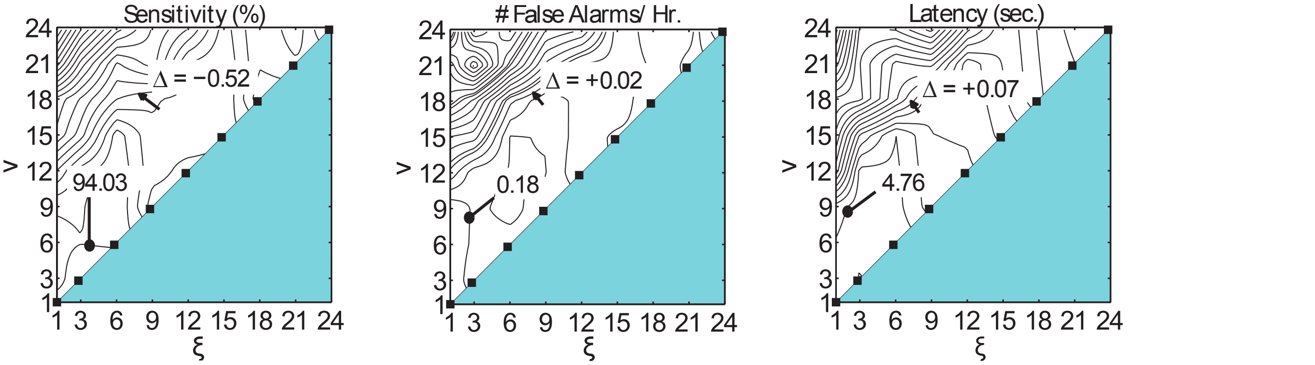


Fig. 5. Figure reproduced from [19]: For the approximate solution, performance degrades gradually due to the JL-approximation at higher values of ν.

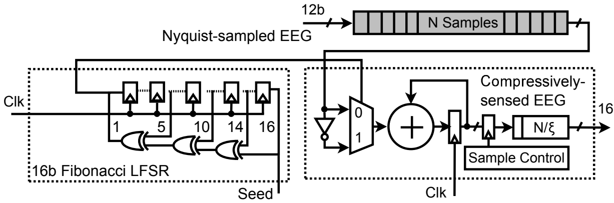


Fig. 6. Compressive sensing using random projections. The energy of the sensing tier scales linearly with the compression factor ξ.

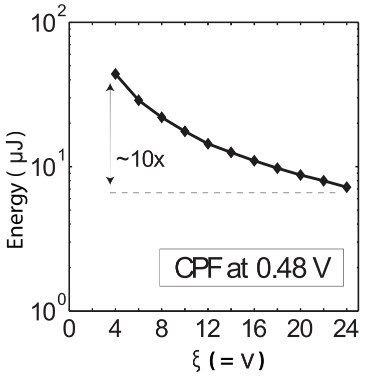


Fig. 7. Compression by 2-24× enables CPF energy scaling by an order of magnitude.

CMOS process from IBM. The circuits and measurement results were previously presented in [18] and [19]. 18 channels of Nyquist EEG signals were sampled at a rate of 256 Hz, and eight CD-BPFs were derived corresponding to eight Nyquist-domain BPFs, each of order $k = 64$ (based on the filter specifications required for seizure detection).

A. Scalability in the Sensing Tier through Random Projections

Fig. 6 shows the compressive-sensing front-end (CPF) module that was integrated on the IC. It comprises two computational blocks: a pseudo-random number generator (PRNG) and the projection logic. A sequence of ±1 values were obtained using a PRNG based on a 16 bit Fibonacci linear feedback shift register with the characteristic polynomial $x^{16} + x^{14} + x^{10} + x^5 + 1$. Based on the sequence of values, each compressively-sensed signal sample was computed serially as $\hat{\mathbf{x}}_\mathbf{i} = \mathbf{x_1} \pm \mathbf{x_2} \pm \ldots \pm \mathbf{x_N}$. This process was repeated N/ξ times to provide the compressively-sensed signal $\hat{\mathbf{x}}$. Some other potential implementations of the CPF are proposed in [13] and [12]. To achieve higher compression factors (larger ξ values), the number of random projections [and thus, the number of multiply-accummulate (MAC) operations] in the CPF decrease linearly with ξ. Thus, depending on the amount of compression, we can achieve energy scalability in the sensing tier.

Fig. 7 shows the scaling in the energy of the CPF implemented in [19]. The CPF was operated at its minimum energy point of 0.48 V [24] and it permits EEG compression by a factor of ξ = 2-24×, consuming 7.3-85 pJ of energy.

B. Scalability in the Processing Tier through SRAM Rationing

An important consequence of the algorithmic construction proposed in [17] is that the CD-BPF matrices $\hat{\mathbf{H}}_\mathbf{i}$ (which are of dimensionality $\frac{N}{\xi} \times \frac{N}{\xi}$ and $\frac{N}{\nu} \times \frac{N}{\xi}$ for the exact and approximate solution, respectively) do not retain the regularity of $\mathbf{H_i}$. Even though $\mathbf{H_i}$ are of dimensionality $N \times N$, as shown in Fig. 8, the rows of $\mathbf{H_i}$ are simply selected to implement convolution, and thus are shifted versions of the impulse response of the same FIR filter. As a result, very few unique filter coefficients are required, and many of the coefficients are zero, as determined by the FIR-filter order k. However, in deriving $\hat{\mathbf{H}}_\mathbf{i}$, the shifted impulse responses and zero entries are disrupted. As shown in Fig. 8, the number of multiplications required thus no longer depends on the filter order, but rather (1) *quadratically* on the compression factor ξ for the exact solution and (2) *linearly* on both ξ and ν for the approximate solution. This scaling helps reduce the number of multiplications required in the processing tier.

Due to the disruption in regularity, the $\hat{\mathbf{H}}_\mathbf{i}$ matrices need

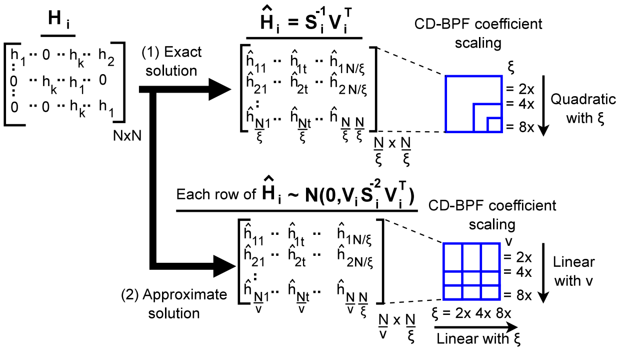

H i
$h_1 \cdot\cdot\ 0 \cdot\cdot\ h_k \cdot\cdot\ h_2$
$0 \cdot\cdot\ h_k \cdot\cdot\ h_1 \cdot\cdot\ 0$
$0 \cdot\cdot\ 0 \cdot\cdot\ h_k \cdot\cdot\ h_1$
NxN
(1) Exact solution
$\hat{H}_i = S_i^{-1} V_i^T$
$\hat{h}_{11} \cdot\cdot\ \hat{h}_{1t} \cdot\cdot\ \hat{h}_{1N/\xi}$
$\hat{h}_{21} \cdot\cdot\ \hat{h}_{2t} \cdot\cdot\ \hat{h}_{2N/\xi}$
$\hat{h}_{\frac{N}{\xi}1} \cdot\cdot\ \hat{h}_{\frac{N}{\xi}t} \cdot\cdot\ \hat{h}_{\frac{N}{\xi}\frac{N}{\xi}}$
$\frac{N}{\xi} \times \frac{N}{\xi}$
CD-BPF coefficient scaling
ξ
= 2x
= 4x
= 8x
Quadratic with ξ
Each row of $\hat{H}_i \sim N(0, V_i S_i^{-2} V_i^T)$
$\hat{h}_{11} \cdot\cdot\ \hat{h}_{1t} \cdot\cdot\ \hat{h}_{1N/\xi}$
$\hat{h}_{21} \cdot\cdot\ \hat{h}_{2t} \cdot\cdot\ \hat{h}_{2N/\xi}$
$\hat{h}_{\frac{N}{v}1} \cdot\cdot\ \hat{h}_{\frac{N}{v}t} \cdot\cdot\ \hat{h}_{\frac{N}{v}\frac{N}{\xi}}$
$\frac{N}{v} \times \frac{N}{\xi}$
(2) Approximate solution
CD-BPF coefficient scaling
v
= 2x
= 4x
= 8x
Linear with v
ξ = 2x 4x 8x
Linear with ξ

Fig. 8. Figure reproduced from [19]: CD-BPF matrices $\hat{\mathbf{H}}_\mathbf{i}$, derived using $\mathbf{H_i}$ and $\mathbf{\Phi}$, disrupt the regularity and zeros in $\mathbf{H_i}$. The complexity of the CD-BPFs thus scales (a) quadratically with ξ for the exact solution and (b) linearly with ξ and ν for the approximate solution.

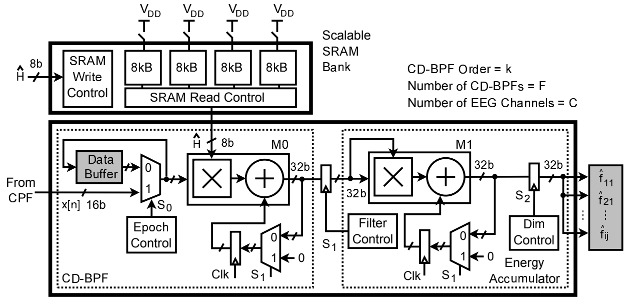


Fig. 9. SRAM bank in the CD-FE enables energy scaling in the processing tier.

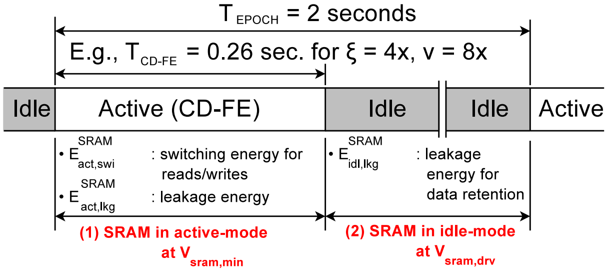

T_{EPOCH} = 2 seconds
E.g., T_{CD-FE} = 0.26 sec. for ξ = 4x, ν = 8x
Idle
Active (CD-FE)
Idle
Idle
Active
• $E^{SRAM}_{act,swi}$: switching energy for reads/writes
• $E^{SRAM}_{act,lkg}$: leakage energy
• $E^{SRAM}_{idl,lkg}$: leakage energy for data retention
(1) SRAM in active-mode at $V_{sram,min}$
(2) SRAM in idle-mode at $V_{sram,drv}$

Fig. 10. Figure reproduced from [19]: Summary of energy components contributing to total SRAM energy (the $\xi = 4\times$, $\nu = 8\times$ case is shown for illustration).

a larger number of distinct coefficients to be stored, potentially increasing the memory requirements. The authors use a scalable SRAM bank to do the power management. Multiple subarrays of SRAM memory enable fine-grained power-gating as well as reduced bit-line and word-line access energy. The filter coefficients were represented using 8 bits of precision and the total bank size in their implementation was 32kB, which was partitioned into four subarrays of 8kB each. Fig. 9 shows the circuits used in the compressed-domain feature extractor (CD-FE). The CD-FE includes a CD-BPF and energy-accumulator block. The coefficients for the CD-BPF are pulled from SRAM.

SRAM Energy Analysis. The SRAM energy per access (E^{sram}_{acc}) was reduced by choosing four smaller-sized subarrays (each of size 8 kB) [25]. The detector processed an EEG epoch every $T_{EPOCH} = 2$ sec. However, the optimal operating frequency (and supply voltage $V_{dd,opt}$) for the CD-FE logic was determined by minimizing the overall CD-FE energy, while ensuring a minimum throughput that allowed the active CD-FE computations to be completed in T_{CD-FE} (< 2) seconds for each value of ξ and ν. For the remainder of the epoch (*i.e.*, $T_{EPOCH} - T_{CD-FE}$), the logic and SRAMs were placed in low-energy idle modes.

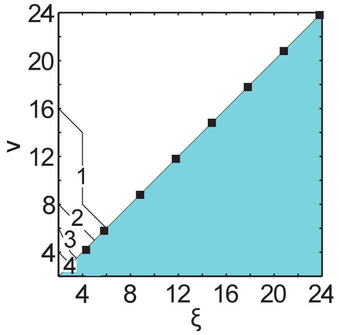


Fig. 11. Figure reproduced from [19]: N_{sub} scales with ξ and ν, affecting the SRAM leakage energy.

Fig. 10 summarizes the SRAM operating modes and energies [25]. The total SRAM energy was the sum of the active-mode (E^{SRAM}_{act}) and idle-mode (E^{SRAM}_{idl}) energies for each subarray (numbering N_{sub}) that was enabled; under the assumption that the SRAMs cannot by fully power-gated in order to ensure data retention, E^{SRAM}_{idl} was not zero. During the active mode, the SRAM operated at the minimum operational supply voltage ($V_{sram,min}$) of 0.7 V for reads and writes; at this voltage, it operated at 920 kHz; this was sufficient performance for all design points (ξ, ν) of the CD-FE. This allowed the SRAM voltage to remain at 0.7 V. During the idle mode, the SRAM operated at its minimum data-retention voltage ($V_{sram,drv}$) of 0.42 V.

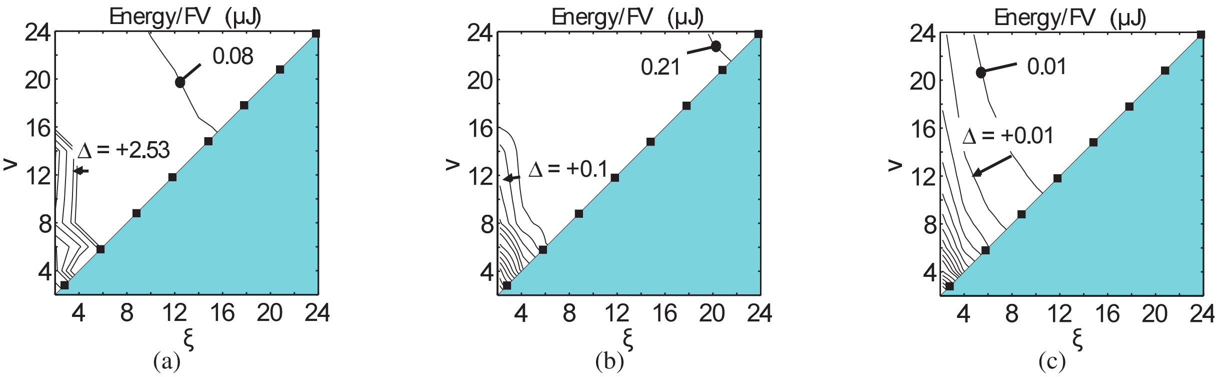
In the active mode, while set to a supply voltage of $V_{sram,min}$, E^{SRAM}_{act} comprised active-switching ($E^{SRAM}_{act,swi}$) and leakage ($E^{SRAM}_{act,lkg}$) energies for a period of T_{CD-FE}. In the idle mode, while set to a supply voltage of $V_{sram,drv}$, E^{SRAM}_{idl} comprised only the leakage energy ($E^{SRAM}_{idl,lkg}$) for the duration ($T_{EPOCH} - T_{CD-FE}$). Thus, the SRAM energy components were represented as follows:

$$\begin{aligned} E^{SRAM}_{lkg} &= E^{SRAM}_{act,lkg} + E^{SRAM}_{idl,lkg} \\ &= N_{sub} T_{CD-FE} \{ I_{V_{sram,min}} V_{sram,min} \} \\ &\quad + N_{sub} (T_{EPOCH} - T_{CD-FE}) \{ I_{V_{sram,drv}} V_{sram,drv} \} \end{aligned} \tag{1}$$

$$E^{SRAM}_{act,swi} = E^{sram}_{acc} \times \#\text{accesses}. \tag{2}$$

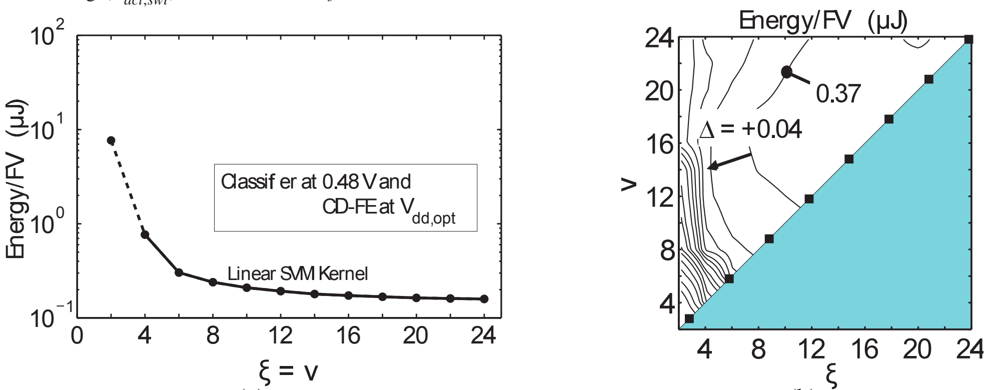
The duration of the active mode (T_{CD-FE}) in Eq. (1) depended on ξ, ν, and the optimum logic voltage $V_{dd,opt}$. For smaller (larger) values of ξ and ν, there were more (fewer) coefficients in $\mathbf{\hat{H}_i}$ and T_{CD-FE} (the active CD-FE time) was higher (lower). For instance, T_{CD-FE} was 0.26 sec. for $\xi = 4\times$ and $\nu = 8\times$, as shown in Fig. 10(b). It increased to 0.52 sec. at $\xi = \nu = 4\times$ and reduced to 0.13 sec. at $\xi = 4\times$ and $\nu = 16\times$. Further, the number of active subarrays (N_{sub}) was also a function of ξ and ν; Fig.11 shows this dependence. Eqs. (1) and (2) also show that although $E^{SRAM}_{act,swi}$ remained invariant to changing values of V_{dd}, it was impacted by ξ and ν (since #accesses changes with ξ and ν). Note that in Eq. (2), E^{sram}_{acc} denotes the active-switching energy per access, which remained invariant to changing values of V_{dd}, ξ, and ν. Similar to $E^{SRAM}_{act,swi}$, the SRAM leakage energy E^{SRAM}_{lkg} also scaled substantially with ξ and ν. Consequently, the optimal logic voltage $V_{dd,opt}$, which minimized the SRAM and logic CD-FE energy, varied (in the range 0.44 − 0.5 V) with respect to ξ and ν.

Figs. 12(a) and (b) show the SRAM leakage energies in the idle and active modes and Fig. 12(c) shows the SRAM switching energy in the active mode, versus ξ and ν. We can see from the figures that for smaller values of ξ and ν, since the size of $\mathbf{\hat{H}_i}$ is larger, T_{CD-FE} is higher and the SRAM active energy dominated the idle-mode energy. This is also consistent with a higher value of $V_{dd,opt}$ at these values of ξ and ν, which enables the CD-FE computations to finish sooner. In contrast, at larger values of ξ and ν, however, there were



(a) (b) (c)

Fig. 12. Figures reproduced from [19]: Each of the SRAM energy subcomponents, *i.e.,* (a) idle-mode leakage ($E^{SRAM}_{idl,lkg}$), (b) active-mode leakage ($E^{SRAM}_{act,lkg}$), and (c) active-mode switching ($E^{SRAM}_{act,swi}$) scales with both ξ and ν.



(a) (b)

Fig. 13. Figures reproduced from [19]: Total processor energy scales substantially with ξ and ν for the (a) exact and (b) approximate solution using a linear SVM kernel.

fewer coefficients in $\hat{\mathbf{H}}_\mathbf{i}$ and the SRAM spent most of the time in the idle mode.

Figs. 13 (a) and (b) show the total processor energy for the exact and approximate solutions, respectively. The SVM classifier operated at the minimum energy point (0.48 V), while the CD-FE logic was at $V_{dd,opt}$. From the figure, we conclude that by using the methodology presented in [17], we can achieve substantial energy scalability in the processing tier with respect to both ξ and ν. Next, we explore The impact of this energy scalability on the end-to-end energy of the multi-tiered system.

IV. Scaling in System-level Energy

In this section, we present an analysis of the end-to-end energy-scaling characteristics of the multi-tiered EEG sensing system. For comparisons, we consider three system models. First is Nyquist Analysis (NA), which is the usual approach wherein EEG signals remain sampled at the Nyquist rate through all three tiers. Second is reconstructed analysis (RA) in which compressively-sensed EEG signals, transmitted from the sensing tier to the processing tier, are first reconstructed and then analyzed in the processing tier. In the event of a seizure, the compressed epochs are relayed to the storage tier. Third is compressed analysis (CA), which is the approach described in the previous section where EEG signals remain compressed through all three tiers. To simplify analysis, we consider only the case of the exact solution (*i.e.*, when $\xi = \nu$).

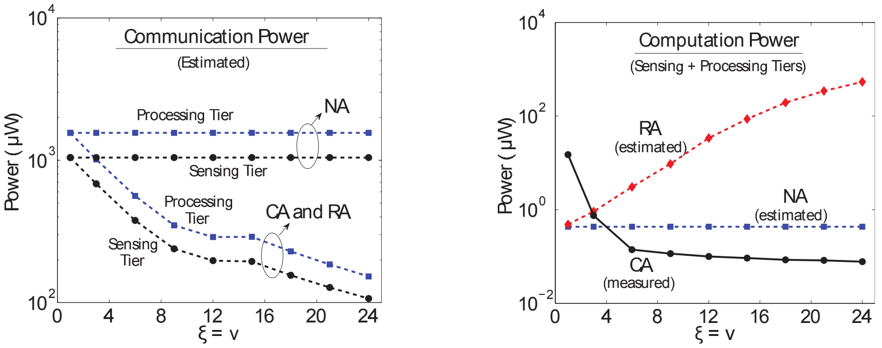
To determine the compression energy (for CA and RA) in the sensing tier, we use measurements from the previous section. For computations in the processing tier, we use measurements (for CA) and estimations (for RA and NA) based on the IC implementation described in the previous section. To estimate the communication energy, we assume BLE radio links between all tiers. Further, we estimate communication energy based on the following time and current values for the various states of a CE for 1 byte of payload: (1) wake-up: 400 μs, 6.0 mA, (2) pre-processing: 340 μs 7.4 mA, (3) pre-Rx: 80 μs 11.0 mA, (4) Rx: 190μs, 17.5 mA, (5) Rx-to-Tx: 105 μs, 7.4 mA, (6) Tx: 115 μs, 17.5 mA, (7) post-processing: 1280 μs, 7.4 mA, (8) pre-sleep: 160 μs, 4.1 mA, and (9) sleep: 1 μA [26]. Given the above nine states, we estimate the energy per CE (E^{tx-rx}_{CE}) of a BLE radio as follows:

$$E^{tx-rx}_{CE} = \left[\sum_{\substack{i=1\\ i\neq 4,5,6}}^{8} T_i I_i + B_{PDU} \left(T_4 I_4 + T_6 I_6 \mathbf{I}_{tx} \right) + T_5 I_5 \mathbf{I}_{tx} \right] V_b$$

where, T_i and I_i, $i \in [1, 9]$ are the time duration and current consumption values, respectively, for state i, B_{PDU} is the number of bytes in the PDU payload, $\mathbf{I}_{tx}$ is an indicator variable that is a 1 or 0 depending or whether the radio performs data transmission or reception, and V_b is the voltage of the battery that powers the radio. Given the above relationship, we estimate the total power consumption of the radio as follows:

$$P_{BLE} = \left[E^{tx-rx}_{CE} N_{CE} + I_9 V_b \left(T_{epoch} - T_{CE} N_{CE} \right) \right] / T_{epoch} \quad (3)$$

where T_{epoch} and T_{CE} are the durations of an EEG epoch and all CEs, respectively, N_{CE} is the number of CEs per epoch.



(a) (b)

Fig. 14. When ξ varies in the range $2 - 24\times$, CA saves power by approximately an order of magnitude in both (a) computation and (b) communication. At higher values of ξ, although RA saves communication energy, reconstruction is expensive leading to high costs in computational energy.

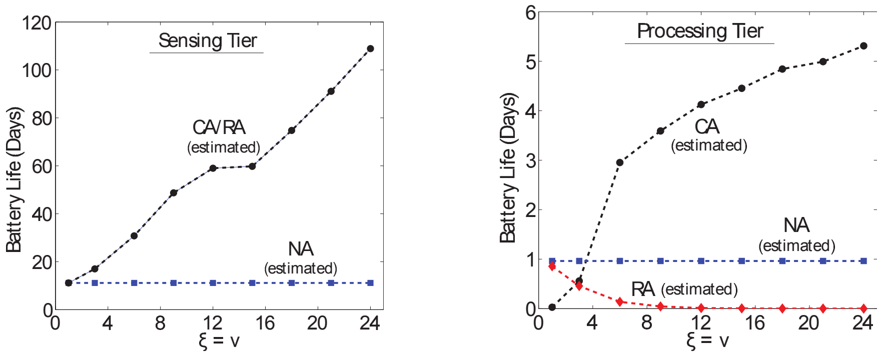
For NA, we assume $V_b = 1.5$ V and 2 sec. EEG epochs with a sampling rate of 256 Hz and 8 bits per EEG sample. Thus, for this case, $N_{CE} = 14$ with 13 of them having $B_{PDU} = 37$ and one with $B_{PDU} = 31$. For both CA and RA, we assume 16 bits per EEG sample. Thus, N_{CE} varies in the range 2 (one with $B_{PDU} = 37$ and the other with $B_{PDU} = 11$) to 14 (13 with $B_{PDU} = 37$ and one with $B_{PDU} = 31$), when ξ varies in the range $2-24\times$. Given, these values, we use Eq. (3) to estimate the power consumption for communication. Note that in all system models, the sensing tier only transmits data while the processing tier receives as well as transmits the EEG signals. The transmission from the processing tier occurs only in the event of a seizure. For our analysis, we assume an average event occurence of 0.2 seizures/hr.

Fig. 14 (a) shows the estimated communication power for both the sensing and processing tiers in NA, RA, and CA. With increasing amounts of data compression (*i.e.*, increasing ξ), both RA and CA provide substantial energy scaling as compared to NA. This is because both of these approaches transmit/receive only compressively-sensed signals. Fig. 14 (b) shows similar comparisons for computational energy. To determine the computational energy in RA, we first estimate the number of operations (OPs) required for signal reconstruction using the Lightspeed toolbox [27]. We then assume values of 0.27 GOPs per second and 29 pW per OP for the processing tier. These assumptions are based on circuits presented in [28] and [29]. The figure shows that, unlike NA, CA and RA both provide computational energy scaling. However, power consumption increases with more compression in RA ($\approx$ 2 orders of magnitude higher at $\xi = 21\times$). This is due to the complex signal reconstruction process, which incurs high computational costs. Note that in RA, reconstruction has to be performed on every EEG epoch. Thus, we conclude that CA is the only approach that simultaneous saves energy in both computation and communication depending on the amount of compression.

Figs. 15 (a) and (b) show projections about battery life for the sensing the processing tiers. We assume Lenmar 1.55 V, 180 mAh [30] and Xeno Lithium 3.6 V, 2400 mAh [31] batteries for the two tiers, respectively. When ξ varies in the range $2-24\times$, the CA system models allows us to prolong battery recharge intervals to about 4 months for the sensing tier and about 5 days for the processing tier. In the NA model, the corresponding recharge intervals are about 2 weeks and 1 day, respectively.

V. Conclusions and Future Work

Multi-tiered architectures provide new opportunities for designing sensing systems that support energy scalability. Scalability can be achieved at each tier separately using various techniques like power/clock gating, voltage/frequency scaling, *etc.* While such techniques are promising, they increase the system complexity due to the required tier-level control. A simpler approach is to build architectures where we modulate a global system-level parameter, such as the amount of data compression, to achieve energy scaling. Compressive sensing is one technique of data compression that is ultra lightweight and suitable for energy-constrained sensing systems. It exploits



(a) (b)

Fig. 15. When ξ varies in the range $2-24\times$, CA prolongs battery recharge times by up to 5× in both the (a) sensing and (b) processing tiers.

signal sparsity in a secondary basis to achieve very low-energy compression. The random projections in compressive sensing, however, affect the sensed signals, preventing the use of Nyquist-domain algorithms for signal analysis. In this paper, we reviewed an existing approach that allows us to transform linear signal-processing computations so that they can be applied directly to compressively-sensed signals. However, due to the JL approximations, this approach introduces some processing error that increases with the amount of data compression. However, this error can be very low (below 5%) up to large compression factors ($\xi = 15\times$). Using a previously presented IC, we showed that if an application can tolerate a small performance hit, we can achieve substantial energy scaling in the end-to-end system energy (up to one order of magnitude energy scaling when ξ varies in the range $2-24\times$). These energy savings can potentially increase battery-recharge times in all tiers by over 5×.

Although the projections made in this paper demonstrate substantial benefits of processing data in the compressed domain, much needs to be done to generalize this methodology. For instance, more work is required to develop new methods that can enable non-linear computations on compressed data. It is well known that the JL approximation improves with high-dimensional data vectors. In the presented application, the dimensionality of data vectors that were compressed at a time was limited by the epoch length. New formulations that can aggregate signals into high-dimensional vectors (perhaps through orthogonal projections) can permit much higher compression factors, while retaining the detection performance. At an application level, an algorithm designer can take advantage of the scaling characteristics in CA by designing intelligent software. For instance, new two-stage algorithms can be developed that take advantage of data compression to perform coarse-grain signal detection in the first stage. In case of a suspected event, a second-stage, that does not use as much compression, can be engaged to resolve ambiguities. Dynamic on-chip power management and high-speed voltage regulation are other areas that need more attention.

References

[1] A. Csavoy, G. Molnar, and T. Denison, "Creating support circuits for the nervous system: Considerations for brain-machine interfacing," in *Proc. Int. Symp. VLSI Circuits*, Jun. 2009, pp. 4–7.

[2] A. P. Chandrakasan, N. Verma, and D. Daly, "Ultralow-power electronics for biomedical applications," *Annual Review: Biomedical Engineering*, vol. 4, pp. 247–274, Aug. 2008.

[3] Texas Instruments,, "2.4 GHz Bluetooth Low Energy and Proprietary System-on-Chip," [Online]. Available: http://www.ti.com/lit/ds/ symlink/cc2541.pdf, Jun. 2013.

[4] ——, "Low Power Advantage of 802.11a/g *vs.* 802.11b," [Online]. Available: http://focus.ti.com/pdfs/bcg/80211_wp_lowpower.pdf.

[5] N. Verma *et al.*, "A micro-power EEG acquisition SoC with integrated seizure detection processor for continuous patient monitoring," in *Proc. Int. Symp. VLSI Circuits*, Jun. 2009, pp. 62–63.

[6] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proc. Int. Conf. Embedded Networked Sensor Systems*, Oct. 2006, pp. 265–278.

[7] A. P. Chandrakasan *et al.*, "Technologies for ultradynamic voltage scaling," *Proc. IEEE*, vol. 98, pp. 191–214, 2010.

[8] V. K. Chippa *et al.*, "Analysis and characterization of inherent application resilience for approximate computing," in *Proc. Design Automation Conf.*, Jun. 2013, pp. 113–119.

[9] B. Priyantha, D. Lymberopoulos, and J. Liu, "Littlerock: Enabling energy-efficient continuous sensing on mobile phones," *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 12–15, Apr. 2011.

[10] R. LiKamWa *et al.*, "Energy proportional image sensors for continuous mobile vision," in *Proc. Int. Conf. Mobile Systems, Applications, and Services*, Jun. 2013, pp. 467–468.

[11] J. Haupt *et al.*, "Compressed sensing for networked data," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92–101, Mar. 2008.

[12] H. Mamaghanian *et al.*, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Trans. Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, Sep. 2011.

[13] F. Chen, A. P. Chandrakasan, and V. M. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE J. Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, Mar. 2012.

[14] D. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[15] 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. Sel. Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[16] M. Shoaib, "Design of energy-efficient sensing systems with direct computations on compressively-sensed data," Ph.D. Thesis, Electrical Engineering, Princeton University, Princeton, NJ, Sep. 2013.

[17] M. Shoaib, N. K. Jha, and N. Verma, "Signal processing with direct computations on compressively-sensed data," *IEEE Trans. VLSI Systems, to appear*.

[18] ——, "A compressed-domain processor for seizure detection to simultaneously reduce computation and communication energy," in *Proc. IEEE Conf. Custom Integrated Circuits*, Sep. 2012, pp. 1–4.

[19] M. Shoaib *et al.*, "A 0.6-106 μW energy scalable processor for seizure detection with compressively-sensed EEG," *IEEE Trans. Circuits and Systems - I, to appear*.

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

[21] A. H. Shoeb and J. Guttag, "Application of machine learning to seizure detection," in *Proc. Int. Conf. Machine Learning*, Jun. 2010, pp. 975–982.

[22] Physionet, "CHB-MIT Physionet database," [Online]. Available: http://www. physionet.org/physiobank/database, Jun. 2000.

[23] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson-Lindenstrauss lemma," *Random Structures and Algorithms*, vol. 22, no. 1, pp. 60–65, Jun. 2002.

[24] A. Wang and A. P. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *J. Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.

[25] N. Verma, "Analysis towards minimization of total SRAM energy over active and idle operating modes," *IEEE Trans. VLSI Systems*, vol. 19, pp. 1695–1703, Sep. 2011.

[26] Texas Instruments,, "Measuring Bluetooth Low Energy Power Consumption," [Online]. Available: http://www.ti.com/lit/an/swra347a/ swra347a.pdf.

[27] T. Minka, "The lightspeed MATLAB toolbox," [Online]. Available: http://www. research.microsoft.com/~minka/software/lightspeed, May. 2011.

[28] V. Karkare, S. Gibson, and D. Markovic, "A 130 μW, 64-channel neural spike-sorting DSP chip," *IEEE J. Solid State Circuits*, vol. 46, no. 5, pp. 1214–1222, May 2011.

[29] S. Gibson, J. W. Judy, and D. Marković, "Technology-aware algorithm design for neural spike detection, feature extraction, and dimensionality reduction," *IEEE Trans. Neural Systems and Rehabilitation Engineering*, vol. 18, no. 5, pp. 469–478, Oct. 2010.

[30] Lenmar Enterprises,, "Lenmar SR44W, 357 Silver Oxide Battery," [Online]. Available: http://www.lenmar.com.

[31] Xeno Energy,, "Xeno Original XL-060F Lithium/SOCl2 Battery," [Online]. Available: http://xenousa.com/.