Signal Processing With Direct Computations on Compressively Sensed Data

Mohammed Shoaib, *Member, IEEE*, Niraj K. Jha, *Fellow, IEEE*, and Naveen Verma, *Member, IEEE*

***Abstract*— Sparsity is characteristic of a signal that potentially allows us to represent information efficiently. We present an approach that enables efficient representations based on sparsity to be utilized throughout a signal processing system, with the aim of reducing the energy and/or resources required for computation, communication, and storage. The representation we focus on is compressive sensing. Its benefit is that compression is achieved with minimal computational cost through the use of random projections; however, a key drawback is that reconstruction is expensive. We focus on inference frameworks for signal analysis. We show that reconstruction can be avoided entirely by transforming signal processing operations (e.g., wavelet transforms, finite impulse response filters, etc.) such that they can be applied directly to the compressed representations. We present a methodology and a mathematical framework that achieve this goal and also enable significant computational-energy savings through operations over fewer input samples. This enables explicit energy-versus-accuracy tradeoffs that are under the control of the designer. We demonstrate the approach through two case studies. First, we consider a system for neural prosthesis that extracts wavelet features directly from compressively sensed spikes. Through simulations, we show that spike sorting can be achieved with 54× fewer samples, providing an accuracy of 98.63% in spike count, 98.56% in firing-rate estimation, and 96.51% in determining the coefficient of variation; this compares with a baseline Nyquist-domain detector with corresponding performance of 98.97%, 99.69%, and 97.09%, respectively. Second, we consider a system for detecting epileptic seizures by extracting spectral-energy features directly from compressively sensed electroencephalogram. Through simulations of the end-to-end algorithm, we show that detection can be achieved with 21× fewer samples, providing a sensitivity of 94.43%, false alarm rate of 0.1543/h, and latency of 4.70 s; this compares with a baseline Nyquist-domain detector with corresponding performance of 96.03%, 0.1471/h, and 4.59 s, respectively.**

***Index Terms*— Compressed sensing, digital signal processing, linear systems, machine learning, spike sorting, seizure detection.**

I. Introduction

IN NATURE, signals often do not represent the information of interest in the most efficient manner. Sparsity is a property of signals that potentially allows us to represent information much more efficiently. Sparse representations capture most or all information in a signal via a small number of samples. Such representations can significantly benefit several functions, such as communication, storage, and potentially computation. Compressive sensing is one specific technique that exploits sparsity in a transform basis to efficiently represent signals using simple random projections [1]. It is starting to find application in several resource-constrained sensing systems, such as environmental sensors, visual networks, and patient monitors [2]–[5]. However, compressive sensing significantly alters the Nyquist-domain samples; Fig. 1, for instance, shows how an electroencephalogram (EEG), which is sparse in the Gabor basis, is altered as a result of random projections in compressive sensing. Consequently, before signal processing can be performed using conventional frameworks, it becomes necessary to reconstruct the original Nyquist-domain signal. The challenge is that reconstruction from random projections, can be extremely costly. In this paper, we focus on transforming signal-processing operations so that they can be applied directly to the compressed signals. We focus on data-driven inference frameworks for analyzing sensor signals. In this case, our approach enables the use of compressively-sensed signals while completely avoiding signal reconstruction. Our transformations also significantly reduce computational energy by enabling processing over fewer input samples. Our specific contributions are as follows.

1) For the first time, we present a mathematical framework to derive compressed-domain equivalents of linear signal-processing functions. We consider both rate-preserving systems [e.g., finite impulse response (FIR) filtering] as well as multirate systems (e.g., wavelet transforms, downsampling, etc.).

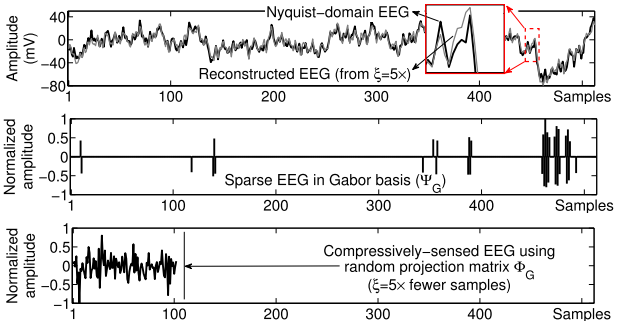

Fig. 1. Nyquist-domain EEG is sparse in the Gabor basis (center panel), enabling substantial compression (bottom panel); although accurate reconstruction is possible (top panel), reconstruction is computationally intensive, motivating signal analysis directly using the compressed signal.

Manuscript received March 3, 2013; revised November 28, 2013; accepted January 12, 2014. This work was supported in part by the Harold W. Dodds Honorific Fellowship from Princeton University, in part by the National Science Foundation under Grant CCF-1253670, in part by the Systems on Nanoscale Information Fabrics, and in part by the Six SRC STARnet Centers through the MARCO and DARPA.

M. Shoaib is with Microsoft Research, Redmond, WA 98052 USA (e-mail: mohammed.shoaib@microsoft.com).

N. K. Jha and N. Verma are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: jha@princeton.edu; nverma@princeton.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2014.2301733

2) We present an approach that solves for random projections of a processed signal, thereby introducing important designer-controllable knobs for system-level tradeoffs. First, the projections can be used to derive an exact solution for the compressed-domain equivalents. Second, the projections can be used to derive approximate solutions, wherein fewer signal-processing operations are required, thereby enabling a new knob for computational power management. This enables energy savings while ensuring the required end-to-end performance for the system.
3) To illustrate our approach for rate-preserving systems, we derive compressed-domain equivalents of wavelet computations for neural prosthesis. We use the compressed-domain features to sort spikes and infer statistical parameters, which can be used to synthesize control function for prosthetics. We show that we can achieve system performance similar to an approach where features are extracted from signals that are first reconstructed.
4) To illustrate our approach for multirate systems, we derive compressed-domain equivalents for downsampling and FIR filtering in a seizure-detection system. We provide an exact solution for the compressed-domain operations and demonstrate a significant improvement in performance compared with a least-squares approximate solution [6], which limited the performance that was previously achievable.

The rest of this paper is organized as follows. In Section II, we present background on compressive sensing and related work in compressed-domain analysis. In Section III, we present system models that will enable a quantitative evaluation of the approach. In Section IV, we describe our approach for deriving compressed-domain processing functions. In Section V, we describe the specific metrics used to evaluate system-level tradeoffs. We then provide experimental results from two case studies: 1) a neural-prosthesis system, described in Section VI and 2) a seizure-detection system described in Section VII. Finally, we conclude in Section IX.

II. Background

In this section, we present background on compressive sensing. We also review related work that focuses on processing applied to compressed data vectors.

A. Compressive Sensing

Compressive sensing is a technique that can be used to compress an N-sample signal $\mathbf{x}$, which is sparse in a secondary basis Ψ; e.g., EEG is sparse in the Gabor basis [7] and spike data are sparse in the wavelet basis [8]; the sparse dictionary Ψ can be learned by training on the data, and such data-driven bases often outperform predefined fixed dictionaries [9], [10]. Thus, if we can represent $\mathbf{x}$ as $\Psi\mathbf{s}$, where $\mathbf{s}$ is a vector of C-sparse coefficients, we can use a projection matrix $\mathbf{\Phi}$ to transform $\mathbf{x}$ to a set of M $[O\{\text{Clog}(N/C)\} < M \ll N]$ compressed samples (denoted by $\hat{\mathbf{x}}$) as follows:

$$\hat{\mathbf{x}}_{\mathbf{M}\times\mathbf{1}} = \Phi_{\mathbf{M}\times\mathbf{N}}\mathbf{x}_{\mathbf{N}\times\mathbf{1}}. \tag{1}$$

The compression factor $\xi = N/M$ quantifies the amount of compression achieved by the projection. For accurate recovery of $\mathbf{x}$ from $\hat{\mathbf{x}}$, $\mathbf{\Phi}$ needs to be incoherent with Ψ; an $M \times N$ dimensional matrix $\mathbf{\Phi}$, whose entries are independent identically distributed (i.i.d) samples from the uniform distribution $U(+1, -1)$ or from the normal distribution $N(0, 1)$, is often maximally incoherent with Ψ [1]. Deriving $\mathbf{\Phi}$ from $U(+1, -1)$ also leads to low-energy compression since the projection is reduced to simple additions and subtractions.

Although sensing can thus incur very little energy, the reconstruction of $\mathbf{x}$ from $\hat{\mathbf{x}}$ can be costly. We see from (1) that $\hat{\mathbf{x}}$ is underdetermined (i.e., knowing only $\hat{\mathbf{x}}$ and $\mathbf{\Phi}$, there are an infinite number of possible solutions for $\mathbf{x}$ and, hence, for $\mathbf{s}$). However, since $\mathbf{x}$ is sparse in Ψ, the sparsest solution for $\mathbf{s}$ is often the correct solution with high probability. One common approach used to determine the sparse solution is to solve the following convex optimization problem:

$$\text{minimize } \|\mathbf{s}\|_1 \text{ subject to } \hat{\mathbf{x}} = \mathbf{\Phi}\Psi\mathbf{s}. \tag{2}$$

The reconstructed signal is then given by $\mathbf{x_R} = \Psi\mathbf{s}^*$, where $\mathbf{s}^*$ is the optimal solution to (2). Although (2) requires only a small number of measurements ($M \ll N$) to enable accurate recovery, even with the most efficient approach, the complexity of solving the optimization problem can be prohibitive on typical power-constrained platforms, such as sensor nodes [11]–[14].

B. Related Work

Due to the challenges associated with signal reconstruction, compressive sensing has primarily been used in applications where the sensor nodes acquire and relay data to a remote base station for analysis [5], [7], [15]–[17]. Emerging applications, however, have emphasized the need for on-node analysis [18], [19]. Random projections, which are used in compressive sensing, preserve inner products of signals. Since inner products are at the core of several machine-learning algorithms, the application of machine-learning frameworks to compressively-sensed data has thus recently received some attention. Theoretical error bounds for using generative and discriminative classifiers with compressively-sensed signals have been explored in [20] and [21], respectively. Similar work has shown bounds of $\mathcal{O}(\sqrt{M})$ and $\mathcal{O}(\log M/\sqrt{M})$ for compressed-domain classification and regression, respectively [21]–[23]. Applying principal component analysis and manifold learning to compressively-sensed signals has also been explored [24], [25]. An important aspect of exploiting inference frameworks for signal analysis, however, is to develop an ability to process signals before classification in order to create data representations based on critical signal features [26]. In [6], we presented an approach to extract specific signal features from compressively-sensed EEG. Our approach was based on a least-squares solution to the compressed-domain functions. We demonstrated accurate seizure detection up to $\xi = 10\times$. In this paper, we introduce a new auxiliary matrix in the regularized equations to achieve an exact solution for any linear signal-processing function in the compressed

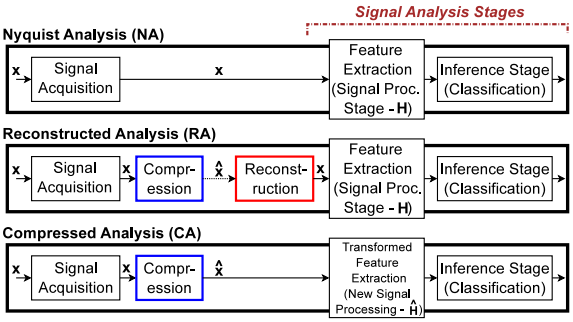

Fig. 2. Reconstruction is bypassed in CA to provide significant savings in computational energy.

domain. This approach retains performance up to much higher values of ξ than in [6]. The new auxiliary matrix also introduces previously unexplored knobs, which are under the control of a designer, to significantly reduce computational energy based on the accuracy needs of the system.

III. Overview of the System Models

Before describing the proposed methodology, we define three system models that we use for comparisons. Fig. 2 shows the generalized system approaches. First, Nyquist analysis (NA) is defined as the usual approach wherein the embedded signals are time-domain representations obtained through Nyquist sampling. Second, reconstructed analysis (RA) is defined as an alternate approach wherein a compressed signal representation is initially received, but is then reconstructed before processing (which is represented by the matrix transformation $\mathbf{H}$). RA corresponds to the system model most commonly used with frameworks like compressive sensing [5], [27]. Third, compressed analysis (CA) is defined as our targeted approach wherein the end-to-end embedded signals are representations based on compressive sensing. Since the proposed methodology aims to explicitly enable energy-accuracy tradeoffs in signal processing, comparisons between CA and RA will enable us to isolate the impact on accuracy due to these tradeoffs from the impact on accuracy due to compression of the initial signal. Fig. 2 also shows the processing stages used in each of the three approaches. Embedded signals are first processed by a feature-extraction stage. Then, the extracted features are used to develop classification models and perform classification via an inference stage. Note that in CA, we need to derive new signal-processing operations ($\hat{\mathbf{H}}$) that enable us to obtain a representation of the targeted features (with minimal distortion errors) directly from the compressively-sensed signals. Thus, CA completely avoids signal reconstruction. From a computational complexity point of view, note that in CA, the derivation of $\hat{\mathbf{H}}$ needs to be done offline only once as opposed to doing signal reconstruction online in RA for every incoming data vector, which can be extremely energy intensive. Next, we describe our approach of deriving the signal-processing operations required in the feature-extraction stage of CA.

IV. Processing Data in the Compressed Domain

In this section, we first show the feasibility of compressed-domain equivalents $\hat{\mathbf{H}}$ for any signal-processing function, which can be represented as a matrix operation $\mathbf{H}$. We aim to minimize the error in the inner product between feature vectors (FVs), since, as described below, this is a key computation in kernel functions for inference stages like classifiers. We show that $\hat{\mathbf{H}}$ permits very low distortion errors with respect to the inner products between FVs.

Many powerful inference frameworks from the domain of machine learning transform data into the Euclidean space by employing signal-processing functions for feature extraction. These frameworks then use linear or nonlinear classification to perform inference over the data. The classification step commonly utilizes a distance metric (e.g., 2-norm or inner product in the Euclidean space) between FVs [28], i.e., classification can be achieved with only inner-product information, rather then complete feature data. Connecting this with the systems in Fig. 3, in NA, an N-dimensional signal $\mathbf{x}$ is multiplied with an $N \times N$ matrix operator $\mathbf{H}$ to perform linear processing that derives an FV $\mathbf{y}$ in the Euclidean space; as we describe below, representing signal processing as a matrix transformation enables the mathematics required to develop the CA methodology. As an example, to realize FIR filtering of time-domain signals (as in NA), $\mathbf{H}$ is simply selected to implement convolution, wherein each row is a shifted version of the system impulse response, thus realizing the multiply-accumulate-shift operations required. The inner product of $\mathbf{y}$ with other FVs is then used as the distance metric for classifying the input signal $\mathbf{x}$. Extending this to CA, we now aim to process compressed representations of the input signal, namely $\hat{\mathbf{x}} = \mathbf{\Phi x}$, where $\mathbf{\Phi}$ represents the $M \times N$ random-projection matrix used for compressive sensing. We seek to find a matrix transformation $\hat{\mathbf{H}}$ that leads to a representation of a signal with the intended signal processing, but derived by directly using $\hat{\mathbf{x}}$.

Need for Regularization: Suppose we can process each vector $\hat{\mathbf{x}}$ in CA by a matrix operator $\hat{\mathbf{H}}$ to derive the compressed-domain FV $\hat{\mathbf{y}}$. A naive approach might be to find $\hat{\mathbf{H}}$ such that the output vector $\hat{\mathbf{y}}$ equals $\mathbf{y}$ from NA. This gives the following formulation:

$$\mathbf{y} = \hat{\mathbf{y}} \;\Rightarrow\; \mathbf{Hx} = \hat{\mathbf{H}}\hat{\mathbf{x}} \;\Rightarrow\; \mathbf{Hx} = \hat{\mathbf{H}}\mathbf{\Phi x} \Rightarrow \underset{N\times N}{\mathbf{H}} = \underset{N\times M}{\hat{\mathbf{H}}}\underset{M\times N}{\mathbf{\Phi}}, \tag{3}$$

However, with $M \ll N$, matrix $\hat{\mathbf{H}}$ above corresponds to $N \times M$ variables constrained by $N \times N$ equations. Such a system with fewer variables than equations is overdetermined and has no exact solution. In [6], we proposed an approach to regularize the left hand side of (3) through $\mathbf{\Phi}$. The resulting solution for $\hat{\mathbf{H}}$ was accurate only in the least-squares sense. We show next how an auxiliary matrix $\mathbf{\Theta}$, which can be used instead of $\mathbf{\Phi}$ for regularization, introduces additional degrees of freedom in (3) and allows us to solve for $\hat{\mathbf{H}}$ exactly. Instead of solving for $\mathbf{y} = \hat{\mathbf{y}}$ [6] [as in (3)], we solve for some K-dimensional projection $\mathbf{\Theta y}$ of $\mathbf{y}$. The elements of the $K \times N$ auxiliary matrix $\mathbf{\Theta}$ are now design variables along with $\hat{\mathbf{H}}$. Thus, we need to solve for $\mathbf{\Theta}$ and $\hat{\mathbf{H}}$ simultaneously in the following equation:

$$\mathbf{\Theta y} = \hat{\mathbf{y}} \;\Rightarrow\; \mathbf{\Theta H x} = \hat{\mathbf{H}}\mathbf{\Phi x} \Rightarrow \underset{K\times N}{\mathbf{\Theta}}\underset{N\times N}{\mathbf{H}} = \underset{K\times M}{\hat{\mathbf{H}}}\underset{M\times N}{\mathbf{\Phi}}, \tag{4}$$

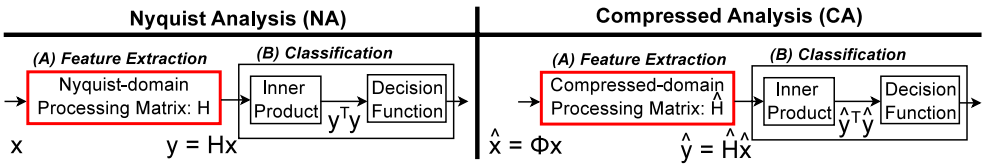

Fig. 3. For any signal-processing function, which can be represented as a matrix $\mathbf{H}$, we derive an equivalent operator $\hat{\mathbf{H}}$ in CA. Since we are not interested in the exact value of $\mathbf{y}$, but in its distance from other processed signals, we solve for a random projection of $\mathbf{y}$, which preserves the inner product of vectors.

With $M \ll N$, $\boldsymbol{\Theta}$ and $\hat{\mathbf{H}}$ together correspond to $K \times (N + M)$ variables constrained by $K \times N$ equations. Thus, with more variables than constraints, (4) will have an infinite number of solutions. This lets us set constraints for finding unique solutions that make several useful design options available.

1) It enables us to solve exactly for the compressed-domain processing matrix $\hat{\mathbf{H}}$, avoiding additional error sources in the processing.
2) By using a smaller value of K, it also permits us to solve for an approximate $\hat{\mathbf{H}}$ of smaller size. This solution provides us with a knob to scale the number of computations performed in CA based on the required accuracy for solving (4).

In addition, by introducing $\boldsymbol{\Theta}$, (4) allows us to extend our methodology from signal-processing operations where $\mathbf{H}$ is a square matrix to those where $\mathbf{H}$ is a nonsquare matrix (e.g., multirate system). We consider the above cases in the Appendices and summarize the solution in the subsections below.

Before proceeding, we parameterize the dimensionality of $\boldsymbol{\Theta}$ and relate it to the dimensionality of $\hat{\mathbf{H}}$; this will ease our consideration of the scaling tradeoffs related to accuracy and energy. The size of the compressed-domain processing matrix $\hat{\mathbf{H}}$ is governed by the size of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ [see (4)]. Thus, in addition to the compression factor $\xi = N/M$, we define a parameter called projection factor ν for $\boldsymbol{\Theta}$ as follows:

$$\nu = N/K. \tag{5}$$

Note that $\nu > 1$ (< 1) denotes a compressive (expansive) projection $\boldsymbol{\Theta}$. Similarly, $\xi > 1$ (< 1) denotes a compressive (expansive) projection $\boldsymbol{\Phi}$. These, in turn, imply fewer (more) computations associated with $\hat{\mathbf{H}}$.

A. Exact Solution for $\hat{\mathbf{H}}$ When $\mathbf{H}$ is Square (for Highest Accuracy)

Assuming $\mathbf{H}$ is a square matrix [e.g., discrete wavelet transform (DWT) in NA], we present a solution for (4) in Appendix A. We find that setting $K = M$ (or $\nu = \xi$) leads to a minimum error solution and results in the following relationships:

$$\hat{\mathbf{H}} = \mathbf{S}^{-1}\mathbf{V}^{\mathbf{T}} \quad \text{and} \quad \boldsymbol{\Theta} = \hat{\mathbf{H}}\boldsymbol{\Phi}\mathbf{H}^{-1}. \tag{6}$$

The solutions for $\boldsymbol{\Theta}$ and $\hat{\mathbf{H}}$ have dimensionality $M \times N$ and $M \times M$ ($M \ll N$ due to compression), respectively. Processing vectors in CA (with an $\hat{\mathbf{H}}$ that is smaller than $\mathbf{H}$) would thus reduce the number of computations as compared with NA.

B. Approximate Solution for $\hat{\mathbf{H}}$ when $\mathbf{H}$ is Square (for Designer-Controllable Energy Savings)

In Appendix B, we present a solution for $\boldsymbol{\Theta}$ and an approximate $\hat{\mathbf{H}}$ to save more computational energy in CA.

To derive the approximate solution, we take help of the Johnson–Lindenstrauss (JL) lemma [29], which states that the inner product of vectors is preserved under random projections. Our results show that $\boldsymbol{\Theta} = (\boldsymbol{\Phi}\mathbf{H}^{-1})^{\mathbf{T}}\hat{\mathbf{H}}$ and each row of $\hat{\mathbf{H}}$ needs to be derived from the normal distribution $N(0, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \mathbf{V}\mathbf{S}^{-2}\mathbf{V}^{\mathbf{T}}$; $\mathbf{S}$ is a diagonal and $\mathbf{V}$ is a unitary matrix obtained from the following singular value decomposition (SVD): $(\boldsymbol{\Phi}\mathbf{H}^{-1})^{\mathbf{T}} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}}$.

In this case, the solutions for $\boldsymbol{\Theta}$ and $\hat{\mathbf{H}}$ have dimensionality $K \times N$ and $K \times M$ (where K can be chosen to be smaller than M or $\nu > \xi$), respectively. Such an approach (with a much smaller $\hat{\mathbf{H}}$ matrix) would reduce the number of computations in CA below those required for the exact solution and save additional computational energy. This energy saving comes at the cost of accuracy in solving (4). However, we will present a case study ahead that suggests that this cost can be small and, in fact, we can reliably employ $K \ll M$ ($\nu \gg \xi$).

Next, we show that our approach is also applicable to multirate signal-processing systems, and we solve (4) when $\mathbf{H}$ is a nonsquare matrix.

C. Solution for $\hat{\mathbf{H}}$ When $\mathbf{H}$ is Nonsquare

For the case when $\mathbf{H}$ is of dimensionality $L \times N$ ($L \neq N$), we again invoke the JL lemma to derive a near-orthogonal matrix $\boldsymbol{\Theta}$ and solve for $\hat{\mathbf{H}}$ using the SVDs of $\mathbf{H}$ and $\boldsymbol{\Phi}$. The derivation is presented in Appendix C, where we show that $\boldsymbol{\Theta}$ is of dimensionality $K \times L$ and its elements are drawn from $N(0, 1)$. We also show that $\hat{\mathbf{H}} = \boldsymbol{\Theta}\mathbf{H}\mathbf{U}\mathbf{S}^{-1}\mathbf{V}^{\mathbf{T}}$, where $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ are derived from the SVD: $\boldsymbol{\Phi} = \mathbf{V}\mathbf{S}\mathbf{U}^{\mathbf{T}}$.

Algorithm 1 shows the pseudocode (with the correct scaling constants) that summarizes our approach of simultaneously solving for $\boldsymbol{\Theta}$ and $\hat{\mathbf{H}}$ under the three conditions described in this section. For the case of a nonsquare $L \times N$ ($L > N$) processing matrix $\mathbf{H}$, the algorithm also shows (on line 15) an optional step of orthogonalization (e.g., by the Gram–Schmidt process) before deriving $\mathbf{B}$, $\mathbf{A}$, and $\hat{\mathbf{H}}$. This ensures a perfectly orthonormal $\boldsymbol{\Theta}$ when its row rank is greater than the column rank. Next, we describe system-level metrics that will be used to evaluate our approach in CA.

V. Metrics Used to Evaluate the Proposed Approach

The approach of the previous section opens up many system design options. To understand the associated accuracy

Algorithm 1 Find Compressed-Domain Processing Matrix $\hat{\mathbf{H}}$

Require: projection dimension K and matrices $\mathbf{\Phi}$ and $\mathbf{H}$
Ensure: $\mathbf{\Theta}$ and $\hat{\mathbf{H}}$ with $\mathbf{\Theta H} = \hat{\mathbf{H}}\mathbf{\Phi}$

Init: $N \leftarrow$ # cols($\mathbf{\Phi}$); $M \leftarrow$ # rows($\mathbf{\Phi}$); $L \leftarrow$ # rows($\mathbf{H}$)
if $L = N$ **then**
  $\mathbf{D^T} := \mathbf{\Phi H^{-1}}$; $\mathbf{USV^T} \leftarrow$ SVD($\mathbf{D}$); {for $\theta_\mathbf{i} = \mathbf{D}\hat{\mathbf{h}}_\mathbf{i}$}
  if $K = M$ **then**
    $\hat{\mathbf{H}} = \sqrt{(N/M)}\left(\mathbf{S^{-1}V^T}\right)$; $\mathbf{\Theta} = \sqrt{(N/M)}\left(\hat{\mathbf{H}}\mathbf{\Phi H^{-1}}\right)$;
  else
    for $i = 1$ **to** K **do**
      $\mathbf{x_i} \sim N(0, \mathbf{I_M})/\sqrt{(K)}$; {for $\hat{\mathbf{h}}_\mathbf{i} \sim N\left(0, \mathbf{VS^{-2}V^T}\right)$}
      $\hat{\mathbf{h}}_\mathbf{i} = \mathbf{VS^{-1}x_i}$; $\theta_\mathbf{i} = \mathbf{Ux_i}$;
    end for
    $\mathbf{\Theta} = \sqrt{(N/M)}\left(\theta_1^T; \ldots; \theta_K^T\right)$; $\hat{\mathbf{H}} = \sqrt{(N/M)}\left(\hat{h}_1^T; \ldots; \hat{h}_K^T\right)$;
  end if
else
  $\mathbf{PQR^T} \leftarrow$ SVD($\mathbf{H}$); $\mathbf{VSU^T} \leftarrow$ SVD($\mathbf{\Phi}$);
  $\mathbf{\Theta} \sim N(0,1)/\sqrt{(NK/M)}$; {ortho($\mathbf{\Theta}$) if $K > L$}
  $\mathbf{B} = \mathbf{\Theta PQ}$; $\mathbf{A} = \mathbf{BR^T U}$; $\hat{\mathbf{H}} = \sqrt{N/M}\left(\mathbf{AS^{-1}V^T}\right)$;
end if

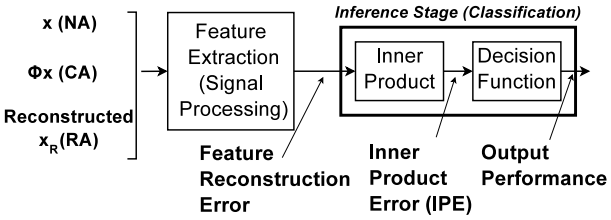

Fig. 4. Metrics used to evaluate the performance of NA, RA, and CA.

tradeoffs, in this section, we discuss precise metrics that are relevant in inference applications. In addition to comparing the proposed CA with NA as a baseline approach, we also compare it with RA in which the sensor node transmits compressed data to an external platform to reduce the amount of data transmitted (hence, saving communication energy and/or alleviating bandwidth constraints); the data are reconstructed on the external platform before performing signal processing. Fig. 4 shows the metrics we use. Since, in CA, we solve for a random projection $\mathbf{\Theta}$ of the FV [see (4)], we expect to be able to reconstruct the signal features accurately. Thus, we reconstruct the FVs in CA and compare them with the features extracted from reconstructed signals in RA. We also compare the variation in the inner-product error (IPE) and the accuracy of the inference stage with respect to both ξ and ν.

A. Reconstruction Signal-to-Noise Ratio With Respect to ξ

Since CA solves for a projection of the processed signal ($\mathbf{\Theta y}$) in NA, the accuracy of processing in CA is expected to be correlated with our ability to recover the $\mathbf{y}$ features from $\mathbf{\Theta y}$. If we denote the reconstructed features as $\mathbf{y}^*_{\mathrm{CA}}$, we can define the SNR in CA as follows:

$$\mathrm{SNR_{CA}} = 10 \cdot \log\left[\|\mathbf{y}\|_2^2 \big/ \left(\|\mathbf{y}^*_{\mathrm{CA}} - \mathbf{y}\|_2^2\right)\right] \text{ dB}. \tag{7}$$

Similarly, the performance in RA is governed by our ability to recover the $\mathbf{y}^*_{\mathrm{RA}}$ features. However, since reconstruction occurs before processing in RA, the reconstructed features $\mathbf{y}^*_{\mathrm{RA}}$ are related to the reconstructed signal $\mathbf{x}^*_{\mathrm{RA}}$ as $\mathbf{y}^*_{\mathrm{RA}} = \mathbf{H}\mathbf{x}^*_{\mathrm{RA}}$. Thus, the SNR in RA can be defined as follows:

$$\mathrm{SNR_{RA}} = 10 \cdot \log\left[\|\mathbf{y}\|_2^2 \big/ \left(\|\mathbf{H}\mathbf{x}^*_{\mathrm{RA}} - \mathbf{y}\|_2^2\right)\right] \text{ dB}. \tag{8}$$

We will investigate how close the SNR in CA is with respect to the SNR in RA for the spike-sorting case study in Section VI-C.

B. IPE in Feature Extraction With Respect to ξ

For feature extraction and classification, a primary concern is how the IPE of FVs scales with ξ. For any two FVs $\mathbf{y_i}$ and $\mathbf{y_j}$, IPE between the inner product in CA (i.e., $\hat{\mathbf{y}}_\mathbf{i}{}^T\hat{\mathbf{y}}_\mathbf{j}$) and the inner product in NA (i.e., $\mathbf{y_i^T y_j}$) is given by

$$\mathrm{IPE} = \left|\hat{\mathbf{y}}_\mathbf{i}{}^T\hat{\mathbf{y}}_\mathbf{j} - \mathbf{y_i^T y_j}\right| \big/ \left(\mathbf{y_i^T y_j}\right). \tag{9}$$

We study the scaling characteristics of IPE with respect to the dimensionality of $\mathbf{\Theta}$. We explore this tradeoff for the spike-sorting application in Section VI-C and for the seizure detection application in Section VII-A.

C. Inference Performance With Respect to ξ

Recall that $\xi = N/M$ quantifies the amount of compression achieved by compressive sensing. As ξ becomes larger, we expect the performance of RA and CA to deteriorate with respect to NA. The first question that arises is: till what value of ξ do RA and CA remain competitive with NA for an application of interest? The second question is: as we increase the value of ξ, does CA remain competitive with RA? If it does, then computations can viably be performed on the sensor node, with the additional benefit of computational energy reduction (due to the fewer operations required in CA). This suggests a new design approach to energy-constrained sensor nodes, wherein devices can be more computationally powerful, thanks to energy savings enabled by the explicit use of efficient representations for the embedded signals; this approach to energy reduction can be exploited alongside algorithmic and architectural optimizations. We explore these questions for the two case studies in Sections VI-C and VII-B, respectively.

D. Inference Performance With Respect to ν

Recall that $\nu = N/K$ provides us with a knob to obtain additional computational energy savings in our CA approach since the approximate solution permits a smaller $\hat{\mathbf{H}}$ matrix. These savings come at the cost of accuracy. The first question is what would the impact on performance and computational energy be if we simultaneously turn the ν and ξ knobs? The second question is how the accuracy and energy savings compare to the case where an exact solution is used for $\hat{\mathbf{H}}$? For the two case studies, we explore these scaling trends in Sections VI-C and VII-B, respectively.

VI. Case Study I: Neural Prosthesis With Compressively Sensed Spikes

In this section, we use the case study of a neural-prosthesis system to validate the system-level tradeoffs arising from the proposed approach. We derive the compressed-domain equivalent of a square matrix $\mathbf{H}$, which computes the DWT of spike signals. We study the impact of the exact and approximate solutions for $\hat{\mathbf{H}}$ on system performance. We also analyze the IPE and SNR trends in NA and CA.

A. Neural Prosthesis System

In a control system for neural prosthesis, a passive transponder is used to transmit spike data serially, thus requiring buffering over all channels, at data rates up to 1 Mb/s [31]. Many modern systems thus detect and align spikes on the implant before transmission [32]. This can significantly reduce the data rates. Spikes are then sorted on an external head stage before analysis. This step comprises feature extraction and clustering [33]. DWT and K-means are two widely used algorithms for feature extraction and clustering, respectively [34]. After sorting, the data rates can become significantly lower. Spike trains from each sorted cluster can then be analyzed to extract statistical parameters, such as the spike count (SC), neuron firing rate (FR), interspike interval (ISI), coefficient of variation (CV), and so on. These parameters eventually steer an algorithm for prosthesis control [35], [36].

In CA, we first detect and align spikes on the implant. We then compressively sense each detected spike through random projections. This process can potentially help alleviate the bandwidth requirements of a passive transponder. We then perform spike sorting directly on compressively-sensed data. This can be done either on the external head stage or on the implant itself. If done on the implant, it permits real-time operation by avoiding reconstruction, while potentially reducing the computational energy of spike sorting. Our results below suggest that the computational energy can be reduced substantially. If done on the head stage, CA can reduce the communication constraints of the implant drastically (due to compressive sensing). This implies that low-energy or zero-energy communication links (e.g., based on passive impedance modulation [31]) may be viable. The cost, however, is a small increase in computational energy (for the random projection of data) on the implant. In Section VI-C, we evaluate the benefits in computational energy delivered by CA in this context. Next, we formulate feature extraction as a matrix operation to enable a transformation to CA.

B. Formulating DWT as a Matrix Operation

Fig. 5 shows the computations we focus on for spike sorting and analysis. The DWT function shown can be implemented as a filter bank [37]. To enable a transformation to CA, however, we require DWT to be formulated as a matrix operation $\mathbf{H}$. We describe the formulation of DWT as a matrix operation in Appendix D. Given the DWT formulation in NA, we can derive the corresponding DWT transformation $\hat{\mathbf{H}}$ in the compressed domain based on the approach presented in Section IV.

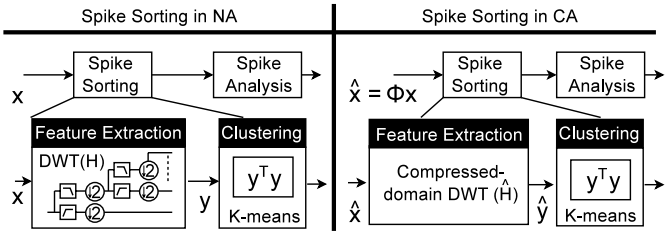

Fig. 5. DWT feature extraction in the neural-prosthesis system; the DWT operation is formulated as a matrix $\mathbf{H}$ in NA, and a corresponding matrix $\hat{\mathbf{H}}$ is derived for CA.

C. Experimental Results

In this section, we compare the experimental results for NA, RA, and CA. We begin by describing our spike sorting framework and discussing how the representative parameters (SC, CV, and FR) for prosthetic control are computed.

The spike sorting and analysis system of Fig. 5 are implemented in MATLAB. For our experiments, we use four records (named as E1, E2, D1, and D2) from the data set in [34]. We first process each record to detect and align spikes using the thresholding algorithm described in [34]. This process results in a window of 64 samples per spike (denoted by vector $\mathbf{x}$). In NA, we then process the detected spikes by a matrix $\mathbf{H}$ to extract specific signal features. $\mathbf{H}$ corresponds to the DWT matrix, which is derived from four levels of decomposition of a Haar mother wavelet. In CA and RA, however, we first project the detected spikes using a matrix $\mathbf{\Phi}$ to obtain the compressively-sensed signal $\hat{\mathbf{x}} = \Phi\mathbf{x}$. We choose each entry of $\mathbf{\Phi}$ from a uniform distribution $U(-1, +1)$ to facilitate an efficient implementation. In RA, before performing computations, we reconstruct signal $\mathbf{x_R}$ from $\hat{\mathbf{x}}$ and then apply $\mathbf{H}$. In CA, however, we directly apply matrix $\hat{\mathbf{H}}$ to compressed signal $\hat{\mathbf{x}}$. We then sort the extracted wavelet features (in NA, RA, and CA) into three clusters using the K-means algorithm. Finally, for each spike cluster, we derive SC, CV, and FR.

1) Baseline Performance of the Spike Sorting Algorithm: Next, we describe how SC, CV, and FR can be computed and what their values are using the various analyses (NA, CA, and RA). SC is simply determined by counting the number of spikes in each cluster after K-means. The first step in computing CV is to determine the ISI histogram. We then model the envelope of the histogram as a Poisson distribution. We directly use this model to determine CV, which is defined as the ratio of the standard deviation to the mean of the distribution function of the ISI histogram. To compute FR for each class, we first determine the number of spikes, which occur in nonoverlapping windows—each of width 300 ms. We then use a Gaussian filter with a length (L) of 30 and variance (σ) of 3 to smooth the binned FR estimates. The bin-width and smoothing filter parameters are chosen empirically to avoid discontinuities in the FR curve. The mean FR is then computed from the smoothed curve. Fig. 6 shows the performance of the spike sorting approach in comparison with the ground truth (GT) values. The GT values are obtained using annotations that identify the true cluster association for each spike in the database. The end-to-end performance values for CA and RA (with no compression) are also shown. We observe that the

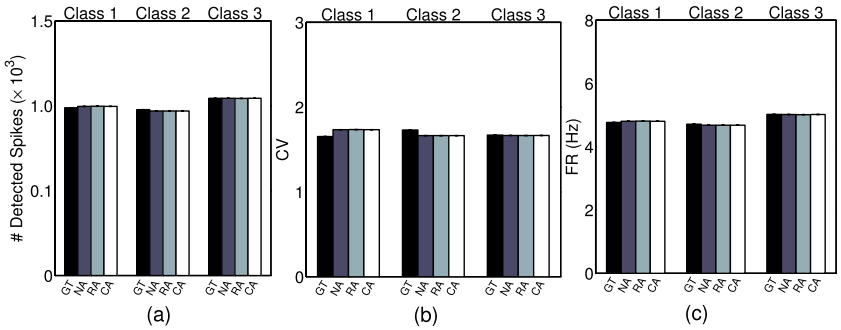

Fig. 6. Accuracy of the spike sorting algorithm in NA is close to the ground truth (GT). Also shown are performance values for CA and RA at $\xi = 1\times$ (these are close to NA).

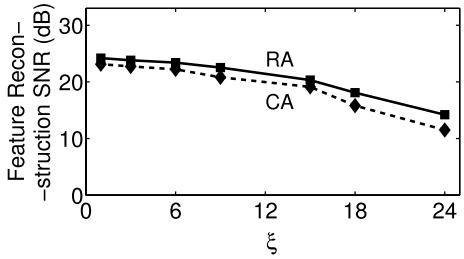

Fig. 7. SNR in CA is close to the SNR in RA, thus validating the similarity in performance trends.

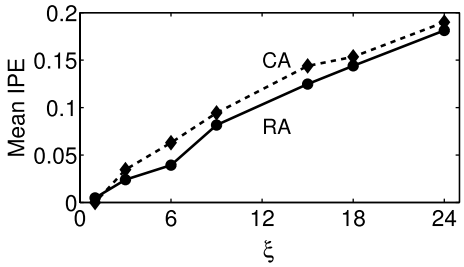

Fig. 8. Mean IPE in CA is close to the mean IPE in RA.

performance of all four approaches are close to one another. Next, we study the performance metrics of Section V (namely SNR, IPE, and performance of CA, RA, and NA) when we scale ξ and ν.

2) Scaling in SNR With Respect to ξ: Since the performance in CA and RA is related to our ability to reconstruct the FVs, we analyze the error introduced in each approach. Fig. 7 shows the mean SNR computed over spikes in all the four records. We observe that the SNR in RA is close to the SNR in CA. Although our ability to reconstruct features governs the performance trends in CA and RA, the inner-product between the features is the key parameter used in the K-means algorithm. Next, we study the IPE scaling of FVs with increasing ξ.

3) IPE With Respect to ξ: Fig. 8 shows the IPE evaluated from the entire spike database. We see that the IPE in CA is only 19% even at $\xi = 24\times$ (RA has a similar error). At $\xi = 24\times$, only three compressively sensed samples per spike are used for CA processing (compared with 64 samples for NA). In RA, we used gradient projection to reconstruct a sparse representation of spikes [14]. Also, we obtained IPE using tenfold cross-validation on the total spike data. In each iteration, we learnt a new sparse dictionary Ψ from K-SVD using 90% of the total spike data [10].

4) Inference Performance With ξ: With fewer compressively-sensed samples (i.e., larger ξ), we expect the accuracy of SC, CV, and FR estimates to deteriorate in RA and CA. Since $\mathbf{H}$ is a square processing matrix in the neural prosthesis application, we use the exact solution for $\hat{\mathbf{H}}$ (from Section IV-A). Fig. 9 and Table I show the mean and standard deviation of the estimation errors for CA and RA, respectively. We use three methods for reconstructing the spikes from $\hat{\mathbf{x}}$, namely basis pursuit [14], l_1-Magic [38], and SPG-Lasso [39]. However, the results for only basis pursuit, which performs better than the other two algorithms, are shown in Fig. 9 and Table I. Each performance metric is obtained for the exact solution (i.e., $\xi = \nu$ case). The estimation errors are with respect to GT and averaged over one hundred different choices of $\mathbf{\Phi}$. We observe that the performance trends for both CA and RA show only a small variance across ξ. The performance degrades gracefully, e.g., even at $\xi = 24\times$, the mean estimation errors with respect to GT for SC, CV, and FR are 8.65%, 5.06%, and 9.96% (standard deviation: 0.81%, 0.30%, and 0.89%) in CA and 6.66%, 4.91%, and 7.54% (standard deviation: 0.69%, 0.12%, and 0.92%) in RA, respectively. Thus, the exact solution enables CA to perform nearly as well as RA. Since compression does not introduce significant errors, we can significantly compress the spikes before transmitting them to the external head stage. We next study the performance trends in CA under an approximate solution for $\hat{\mathbf{H}}$ (from Section IV-B).

5) Inference Performance With ν: Since the approximate solution permits a smaller $\hat{\mathbf{H}}$ matrix, it enables additional savings in computational energy. However, as described in Section IV-B, due to the approximation required in $\hat{\mathbf{y}}$, this can impose a performance cost. Fig. 10 shows trends in performance as we simultaneously scale the projection factor ν and compression factor ξ. The points corresponding to the exact solution for $\hat{\mathbf{x}}$ are shown as black squares along the diagonal. Fig. 10 also indicates that the degradation in the three parameters (i.e., SC, CV, and FR) is small with increasing values of ν. For example, for both SC and FR (left and right plots), $\nu = 9\times$ (at $\xi = 6\times$) incurs very little error, yet enables $54\times$ reduction in the size of the transformation matrix for CA processing; CV incurs somewhat higher error, but still quite small ($\leq$3.5%). Note that there are several local minima shown in the contour plots of Fig. 10. These minima occur since the performance metrics

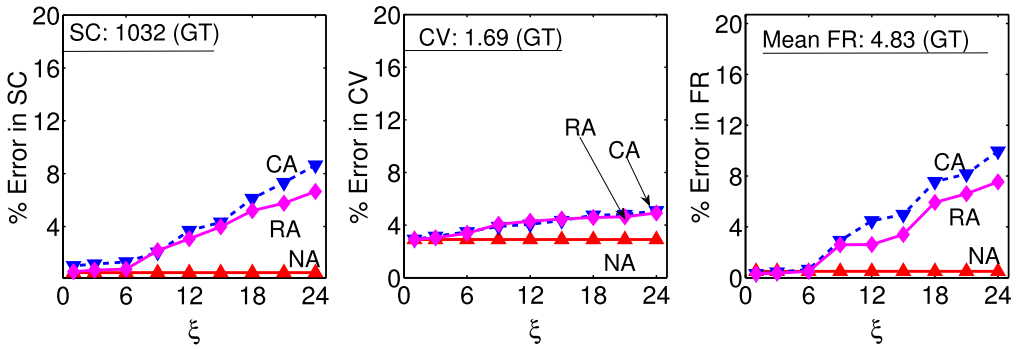

Fig. 9. Scaling with ξ: trends for CA and RA are similar—they slowly degrade with increasing ξ, enabling us to compress the spikes significantly.

TABLE I

STANDARD DEVIATION IN THE PERFORMANCE OF SPIKE SORTING ACROSS ONE HUNDRED DIFFERENT CHOICES OF $\mathbf{\Phi}$

	% Error in SC									% Error in CV									% Error in FR								
	$\xi = 1\times$	$3\times$	$6\times$	$9\times$	$12\times$	$15\times$	$18\times$	$21\times$	$24\times$	$\xi = 1\times$	$3\times$	$6\times$	$9\times$	$12\times$	$15\times$	$18\times$	$21\times$	$24\times$	$\xi = 1\times$	$3\times$	$6\times$	$9\times$	$12\times$	$15\times$	$18\times$	$21\times$	$24\times$
CA	0.03	0.04	0.10	0.25	0.65	0.73	0.62	0.84	0.81	0.02	0.01	0.02	0.08	0.12	0.11	0.09	0.26	0.30	0.10	0.15	0.23	0.42	0.56	0.69	1.00	0.97	0.89
RA	0.03	0.04	0.02	0.16	0.20	0.24	0.56	0.49	0.69	0.01	0.01	0.01	0.03	0.05	0.10	0.08	0.13	0.12	0.11	0.09	0.14	0.52	0.39	0.58	0.79	0.87	0.92

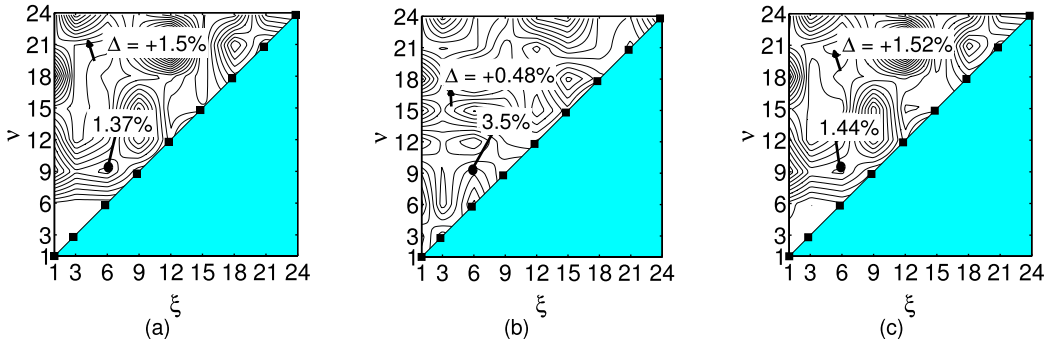

Fig. 10. Scaling with ν: SC and FR performance in CA is retained up to $\nu = 6\times$. The exact solution is also shown in (b) with dark boxes. The cases of $\nu < \xi$, which are not favorable for low-energy operation, are blocked out. Δ shows the separation between adjacent contours. (a) % error in mean SC (CA). (b) % error in mean CV (CA). (c) % error in mean FR (CA).

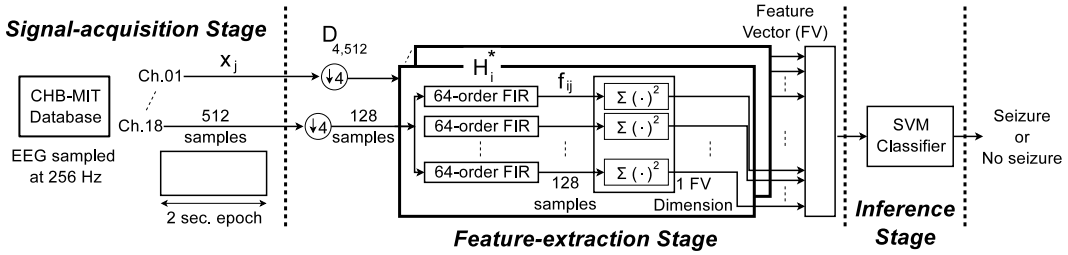

Fig. 11. Spectral-energy feature extraction in NA can be represented as a product of the decimation matrix $\mathbf{D_{4512}}$ and an FIR bandpass filter (BPF) $\mathbf{H}^*$.

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. For fixed ξ and ν, the local minima tend to remain practically unchanged with different choices of $\hat{\mathbf{H}}$ and $\mathbf{\Theta}$. This behavior is observed since the JL lemma provides low variance in the IPE values [29].

VII. Case Study II: Epileptic Seizure Detection Using Compressively-Sensed EEG

In this section, we present a second case study. The Nyquist-domain processing matrix $\mathbf{H}$ we consider is non-square. We thus derive the compressed-domain equivalent matrix $\hat{\mathbf{H}}$ using the solution in Section IV-C. First, we describe the Nyquist-domain algorithm for seizure detection, which employs patient-specific classifier training [40].

Fig. 11 shows the baseline algorithm for seizure detection. A two-second epoch from each EEG channel is processed using eight BPFs with passbands of 0–3 Hz, 3–6 Hz, …, 21–24 Hz. The spectral energy from each filter is then represented by summing the squared value of the output samples to form a FV, which is then used for classification by a support-vector machine (SVM) classifier. The feature-extraction process represents a special case of nonlinear processing (i.e., we handle this by deriving a random projection and use the JL lemma to represent the signal energy). Further, note that since the feature-extraction process for this particular application involves spectral-energy extraction after

filtering, the energy in the filtered EEG signal from each filter corresponds to one dimension of the FV. This operation can be represented as an inner-product computation: $\mathbf{y_{ij}} = \mathbf{f_{ij}^T f_{ij}}$. Relating the entire feature-extraction process with the stages in Fig. 4, we observe that there is an additional inner-product computation involved before classification. Thus, for this case study, the IPE metric defined in Section V directly represents the error in the signal features.

The baseline detector in NA is validated on 558 h of EEG data from 21 patients (corresponding to 148 seizures) in the CHB-MIT database [41]. For every patient, up to 18 channels of continuous EEG is processed using eight BPFs, leading to an FV dimensionality of 144. The Nyquist-domain detector has been demonstrated to achieve an average latency, sensitivity, and specificity of 4.59 s, 96.03%, and 0.1471 false alarms/h, respectively [40].

A. Formulating Feature Extraction as a Matrix Operation

To enable a transformation to the compressed domain, we focus on computations in the feature-extraction stage of Fig. 11. Note that to enable efficient processing with a low-order FIR filter, we downsample the EEG signals before filtering. Since the BPFs in the filter bank have a maximum cutoff frequency of 24 Hz and EEG signals in the CHB-MIT database are sampled at 256 Hz, we downsample the data from each channel by a factor of 4. For each data channel, one EEG epoch (corresponding to 512 Nyquist samples) thus results in 128 decimated samples. These samples are then processed with eight BPFs of order 64. To represent the BPF computations as a matrix operation, we generalize our formulation in (21) [Section VI-B handled decimation of an N-sample signal by $2\times$]. We define a new decimation matrix $\mathbf{D_{4512}}$, which acts upon a 512-sample EEG epoch to give 128 decimated samples. Suppose we represent one EEG epoch from the jth channel as $\mathbf{x_j}$. Then, $\mathbf{D_{4512}}$ acts upon $\mathbf{x_j}$ to give us 128 samples. Further, suppose we represent each 64-order BPF before energy accumulation as a convolution matrix $\mathbf{H_i^*}$, $0 \leq i \leq 7$, of dimensionality 128×128—observe that in $\mathbf{H_i^*}$ each row is a shifted version of the previous row. We can then represent the decimation plus filtering operation in the feature-extraction stage as the following cascaded operation:

$$\mathbf{f_{ij}} = \mathbf{H_i^*}\ \mathbf{D_{4512}}\ \mathbf{x_j} \tag{10}$$

where $\mathbf{f_{ij}}$ is the filtered EEG data derived from the ith filter acting upon the jth EEG channel. The Nyquist-domain processing matrix for each BPF can thus be defined as $\mathbf{H_i} = \mathbf{H_i^* D_{4512}}$. This matrix is rectangular and has a dimensionality of 128×512.

Next, we investigate the variation of IPE with ξ as well as its correlation with the performance of the end-to-end algorithm.

B. Experimental Results

In this section, we study the error in the FVs (represented by IPE) and the performance of the end-to-end system. We will see that the performance does not correlate directly with the IPE because the information content of the features is what controls the performance of the system [6]. This behavior, which is unlike the previous case study, is due to the presence of the spectral-energy operation in the feature-extraction stage. We thus also study the variation in mutual information with respect to ξ in CA and compare it with that in RA.

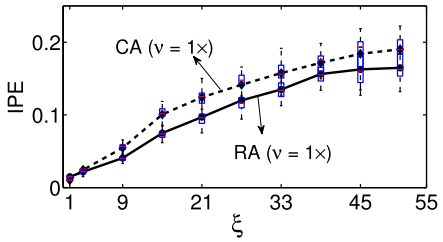

Fig. 12. Mean IPE computed across ten different pairs of $\mathbf{\Theta}$ and $\mathbf{\Phi}$. The variation in the mean IPE values shown in the figure corresponds to the 21 patients in the CHB-MIT database. Mean IPE in CA is close to the mean IPE in RA.

In CA, we derive compressed-domain processing matrices $\mathbf{\hat{H}_i}$ from the corresponding rectangular NA matrices $\mathbf{H_i}$ using the solution in Section IV-C. Note that $\mathbf{\hat{H}_i}$ has $K \times M$ [or $N(1/\nu + 1/\xi)$] entries. As in NA, we then obtain the processed signal from each filter as $\mathbf{\hat{f}_{ij}} = \mathbf{\hat{H}_i}\ \mathbf{\Phi}\ \mathbf{x_j}$, where the processing matrix $\mathbf{\hat{H}_i}$ acts directly on the compressively-sensed signal $\mathbf{\Phi x_j}$. We then derive a CA estimate of the spectral energy as $\mathbf{\hat{y}_{ij}} = \mathbf{f_{ij}^T f_{ij}}$.

1) IPE With Respect to ξ: The error in the FVs (IPE) is defined as $\text{IPE} = \|\mathbf{\hat{y}_{ij}} - \mathbf{y_{ij}}\|/\mathbf{y_{ij}}$. We expect the error to increase with increasing compression ($\xi > 1\times$). For our experiments, we keep $\nu = 1\times$ and scale ξ. The computational savings in CA thus increase with ξ [$\mathbf{\hat{H}_i}$ has $N(1/\nu + 1/\xi)$ entries]. Fig. 12 shows the trend in the IPE. The plot also shows the variation in the IPE across all patients in the database. We observe that the IPE is $<19.5\%$ upto $\xi = 51\times$, at which point we only transmit and process 10 EEG samples per epoch. The figure also shows the IPE in RA, where we reconstruct each epoch using gradient projection [14]. For each patient, we learn a new sparse dictionary Ψ from K-SVD using tenfold cross-validation [10]. We observe that IPE in CA is close to the IPE in RA, thus validating the solution for $\mathbf{\hat{H}_i}$ in Section IV-C. We next proceed to study the impact of the error in the FVs on overall system accuracy.

2) Inference Performance With Respect to ξ and ν: To evaluate the performance of the compressed-domain detector, we derive FVs from the CHB-MIT database. We use these FVs to train and test the SVM classifier in a patient-specific manner. We employ a leave-one-out cross-validation scheme for measuring the performance of the detector. Fig. 13 and Table II show the scaling in mean and standard deviation of the performance metrics in CA computed across ten different pairs of $\mathbf{\Theta}$ and $\mathbf{\Phi}$. The performance of the compressed-domain detector is very close to the Nyquist case at $\xi = 1\times$. For CA, at a compression of $1\times$, the sensitivity is 95.53%, latency is 4.59 s, and the number of false alarms is 0.1538/h. These performance numbers begin to degrade with both ξ and ν. The corresponding numbers at $\xi = 21\times$ are 94.43%, 4.70 s, and 0.1543/h, respectively. As described in Section IV-C, $\nu = 1\times$ gives the lowest approximation error to the JL lemma. Thus, the performance is highest at $\nu = 1\times$ and begins to degrade at higher values of ν. At higher values of ξ (which also

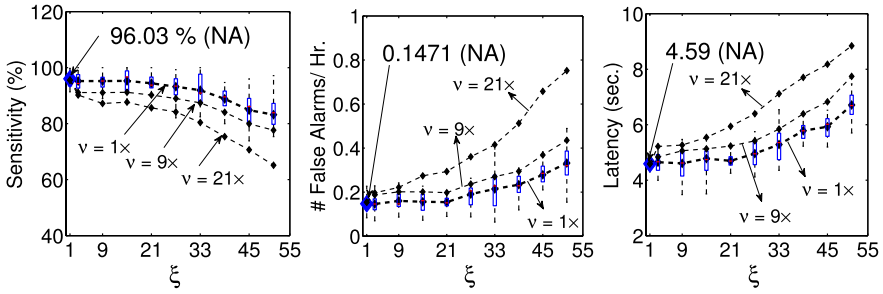

Fig. 13. Mean performance of the seizure-detection algorithm in CA computed across ten different pairs of $\mathbf{\Theta}$ and $\mathbf{\Phi}$. The variation in the mean performance values shown in the figure corresponds to the 21 patients in the CHB-MIT database. Performance is maintained up to $\xi = 21\times$.

TABLE II

STANDARD DEVIATION ACROSS TEN DIFFERENT PAIRS OF $\mathbf{\Theta}$ AND $\mathbf{\Phi}$ IN THE MEAN INFERENCE PERFORMANCE (MEAN COMPUTED ACROSS THE 21 PATIENTS) OF THE SEIZURE DETECTOR IN CA AND RA

	Sensitivity (%)										**False Alarms/ Hr.**										**Latency (seconds)**									
$\xi =$	1×	3×	9×	15×	21×	27×	33×	39×	45×	51×	1×	3×	9×	15×	21×	27×	33×	39×	45×	51×	1×	3×	9×	15×	21×	27×	33×	39×	45×	51×
CA	0.43	0.57	1.22	1.84	2.77	2.36	3.36	3.24	3.99	4.79	0.01	0.01	0.03	0.03	0.05	0.05	0.06	0.07	0.08	0.11	0.10	0.11	0.19	0.52	0.54	0.73	1.09	1.27	1.42	1.47
RA	0.88	1.10	1.65	2.14	2.25	2.96	3.45	3.54	3.50	4.44	0.00	0.02	0.02	0.02	0.06	0.06	0.09	0.11	0.11	0.15	0.09	0.23	0.35	0.69	0.76	1.22	1.17	1.50	1.43	1.79

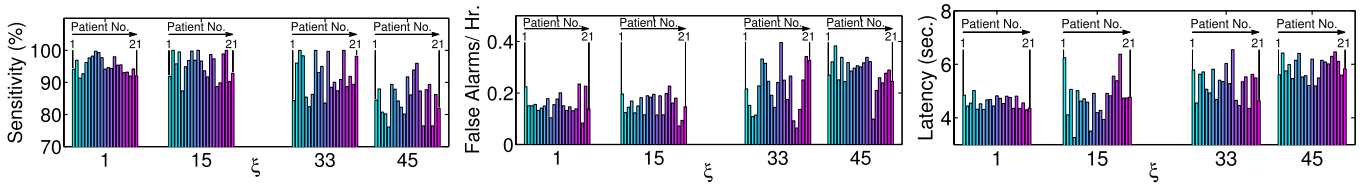

Fig. 14. Variation in the performance of seizure detector across different patients at $\nu = 1\times$. The stochastic nature of the JL lemma results in nonuniform error values.

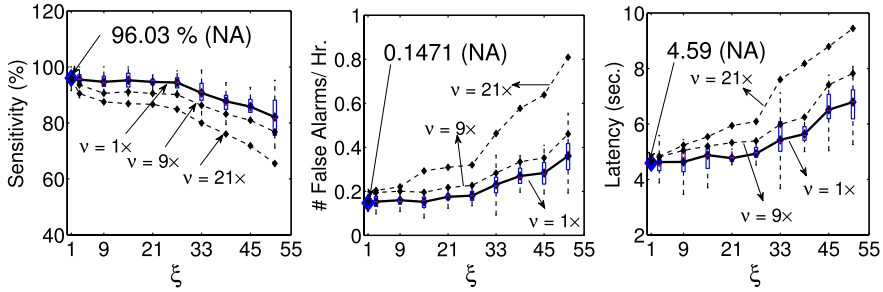

Fig. 15. Mean performance in RA computed across ten different pairs of $\mathbf{\Theta}$ and $\mathbf{\Phi}$ is close to the performance in CA. The variation in the mean performance values shown in the figure corresponds to the 21 patients in the CHB-MIT database.

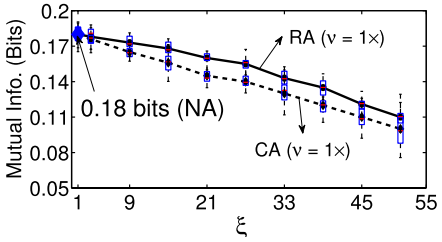

Fig. 16. Mean mutual information across ten different pairs of $\mathbf{\Theta}$ and $\mathbf{\Phi}$. The variation in the mean information values shown in the figure corresponds to the 23 patients in the CHB-MIT database. Mean mutual information in CA and RA follows the performance trends.

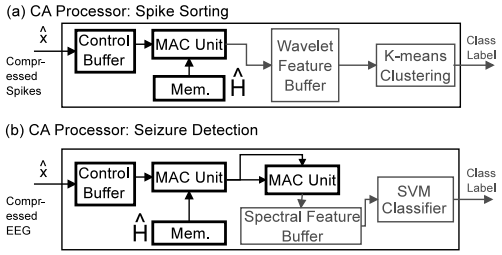

Fig. 17. Architectural block diagrams of the processors in CA.

give corresponding energy savings), the degradation in sensitivity is modest. For instance, at $\nu = 1\times$, the degradation is 1.1% when $\xi = 21\times$, beyond which it begins to drop more significantly. The scaling in the number of false alarms per hour and the latency also follows a similar trend. The mean latency of detection increases by 2.41% while the specificity of the algorithm degrades by only 0.33% at $\xi = 21\times$. Fig. 14 shows the variation in the performance at $\nu = 1\times$ for one pair of $\mathbf{\Phi}$ and $\mathbf{\Theta}$. We observe that the stochastic nature of the JL lemma leads to nonuniform errors in the performance metrics. Fig. 15 shows the performance in RA. We observe that it is close to the performance of CA. These trends, however, do not correlate with the IPE in Fig. 12. For example, at $\xi > 39\times$, IPE in RA is almost constant ∼16%, but the difference in

performance for values of $\xi > 33\times$ is significant. Next, we study the information content in the FVs, which has been shown to be a metric that directly indicates the end-to-end performance of the detector [6].

3) Mutual Information With Respect to ξ: Mutual information between the FVs and the class labels acts as an indicator for the performance of a classifier [6]. High mutual information results in better performance. Fig. 16 shows the variation in mutual information of the FVs in RA and CA versus ξ. We see that the inference performance is high until a mutual information falls below a specific level. This value is specific to the application data and the classifier used. For instance, from Figs. 13, 15, and 16, we observe that the performance degrades only minimally up to $\xi = 21\times$, which corresponds to a mutual information value of $\sim$0.15 bits. Hence, based on the mutual information results for the multirate system presented in this section, we can conclude that very limited degradation is seen up to large compression factors. Thus, this limited degradation in information supports the CA system model.

VIII. Hardware Analysis

In this section, we present an analysis of the hardware complexity of CA and compare it with that of NA. We observe that the number of computations required in CA can be substantially lower. However, there is an increased cost in storage that is required to accommodate the extra coefficients in $\hat{\mathbf{H}}$.

Fig. 17(a) and (b) show architectural block diagrams of CA processors for spike sorting and seizure detection, respectively. In these processors, different values of ξ and ν impact only the multiply-accumulate (MAC) units and the memory where the $\hat{\mathbf{H}}$ coefficients reside. Recall from Sections VI-C and VII-A that four levels of decomposition with a (order-2) Haar wavelet and eight 64-order BPFs are required for the two applications, respectively.

Fig. 18(a) and (b) show the scaling in the estimated number of MAC operations and the memory required, respectively, for the spike sorting application. The estimates for NA are given using different pairs of wavelet order and decomposition level. We observe that the number of MAC operations required by the exact solution ($\xi = \nu$) is below that of NA for $\xi > 5\times$. This is true for wavelets of all orders and for all decomposition levels. However, the memory required in CA is higher than that required by low-order wavelets. For instance, for two levels of decomposition using an order-3 wavelet, ξ is greater than $11\times$ for the exact solution. This increase in memory can be addressed by using a higher value of ν, which results in smaller $\hat{\mathbf{H}}$ matrices; there is thus a tradeoff between memory size and accuracy. Fig. 18(c) and (d) show the estimated MAC operations and the memory requirement for the seizure-detection application, respectively. We again observe a similar computation-memory tradeoff. In this case, since we process multiple data channels, substantial scaling in ν ($\gg 21\times$) is required to match the memory of NA. However, as shown in Section VII-B, such a high scaling leads to a very suboptimal detector performance.

We observe from this section that although CA provides substantial savings in computation and communication energies, it potentially requires more data storage than NA. Consequently, architectures and technologies that address the memory energy and footprint can play an important role in the proposed approach.

IX. Conclusion

Sparsity of signals provides an opportunity to efficiently represent sensor data. Compressive sensing is one technique that exploits signal sparsity in a secondary basis to achieve very low-energy compression at the cost of high complexity in signal reconstruction. The energy for reconstruction can present a significant barrier to signal analysis, which is becoming increasingly important in emerging sensor applications. In this paper, we presented an approach to derive computations that can be performed directly on compressively-sensed signals. Our approach not only circumvents the energy imposed by signal reconstruction, but also enables computational energy savings by processing fewer signal samples. Through analytical validations, we observed that this approach can achieve error bounds in feature estimates that are very close to the expected lower limit. We validated our approach with two case studies, namely spike sorting for neural prosthesis and EEG classification for seizure detection. For the neural-prosthesis application, our experimental results suggest that we can process up to $54\times$ fewer samples while restricting detection errors to under 3.5%. Using our approach, the reduction in the communication energy can also be significant. For instance, in the seizure-detection application, the detection error was under 2.41% when we used $\sim 21\times$ fewer transmitted EEG samples. The proposed approach thus provides an approach for signal-processing systems that addresses system-resource constraints, such as energy and communication bandwidth, through efficient signal representation. This is in contrast to efforts that focus on efficient architectures and algorithms alone.

Appendix A
Derivation of Exact Solution for $\hat{\mathbf{H}}$ When $\mathbf{H}$ is Square

The intuition behind solving for a projection of $\mathbf{y}$ instead of $\mathbf{y}$ itself in (4) is that many machine-learning stages (e.g., SVMs) that act after feature extraction do not use the exact value of $\mathbf{y}$ but only its distance from other vectors. Thus, the Euclidean distance between FVs is the metric we should aim to preserve. The distance between any two FVs, $\mathbf{y_1}$ and $\mathbf{y_2}$, is given by the inner product, $\mathbf{y_1^T y_2}$. The corresponding distance in the compressed domain is given by

$$\hat{\mathbf{y_1}}^{\mathrm{T}}\hat{\mathbf{y_2}} \Rightarrow (\mathbf{\Theta y_1})^{\mathbf{T}}(\mathbf{\Theta y_2}) \Rightarrow \mathbf{y_1^T}(\mathbf{\Theta^T\Theta})\mathbf{y_2}. \tag{11}$$

The right-hand side will be equal to the inner product $\mathbf{y_1^T y_2}$ of NA if $\mathbf{\Theta^T\Theta}$ is equal to the $N \times N$ identity matrix $\mathbf{I}$. Thus, to solve for $\mathbf{\Theta}$ and $\hat{\mathbf{H}}$ exactly in (4), we have to solve the following constrained optimization problem:

$$\underset{\mathbf{\Theta}}{\operatorname{argmin}} \|\mathbf{\Theta^T\Theta} - \mathbf{I}\|_2^2 \text{ such that } \mathbf{\Theta H} = \hat{\mathbf{H}}\mathbf{\Phi}. \tag{12}$$

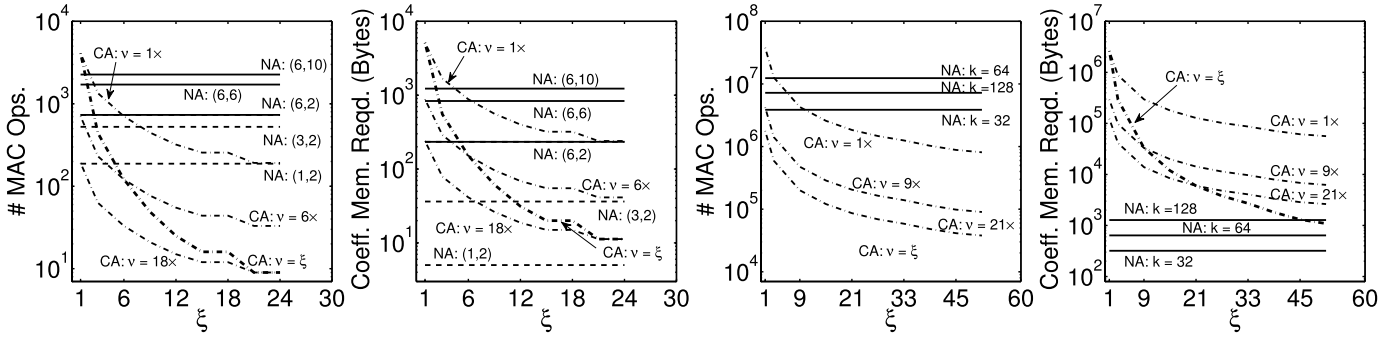

Fig. 18. Number of MAC operations and memory costs, shown for spike sorting [(a) and (b), respectively] and seizure detection [(c) and (d), respectively]. CA incurs fewer computations than NA at the cost of increased storage.

Assuming $\mathbf{H}$ is a square matrix, we can obtain the SVD of $\mathbf{\Phi H^{-1}}$ as $\mathbf{VSU^T}$, where $\mathbf{V}$ and $\mathbf{U}$ are orthogonal matrices (i.e., $\mathbf{U^TU = V^TV = I}$) and $\mathbf{S}$ is an $M \times M$ diagonal matrix formed by the singular values of $\mathbf{\Phi H^{-1}}$. We thus have the following relationship for $\mathbf{\Theta^T\Theta}$:

$$\mathbf{\Theta^T\Theta} = (\hat{\mathbf{H}}\mathbf{\Phi H^{-1}})^{\mathbf{T}}\hat{\mathbf{H}}\mathbf{\Phi H^{-1}} = \mathbf{U(SV^T\hat{H}^T\hat{H}VS)U^T}. \quad (13)$$

The distance from the above matrix to the identity will be at least the rank deficiency of $\mathbf{U}$. The lower bound in (12) will thus be achieved if we set $K = M$ (or $\nu = \xi$)

$$\hat{\mathbf{H}} = \mathbf{S^{-1}V^T} \quad \text{and} \quad \mathbf{\Theta} = \hat{\mathbf{H}}\mathbf{\Phi H^{-1}}. \quad (14)$$

Appendix B
Derivation of Approximate Solution for $\hat{\mathbf{H}}$ When $\mathbf{H}$ is Square

According to the JL lemma [29], $\hat{\mathbf{y_1}}^{\mathbf{T}}\hat{\mathbf{y_2}}$ in (11) will be approximately equal to $\mathbf{y_1^Ty_2}$, if the entries of the auxiliary matrix $\mathbf{\Theta}$ are drawn from the normal distribution $N(0, 1)$ [30]. Thus, we can solve the following modified problem:

Find $\mathbf{\Theta}$ and $\hat{\mathbf{H}}$ such that $\mathbf{\Theta H} = \hat{\mathbf{H}}\mathbf{\Phi}$ and $\mathbf{\Theta} \sim N(0, 1)$.

Suppose $\mathbf{\Theta}$ and $\hat{\mathbf{H}}$ comprise row vectors $\mathbf{\Theta_i^T}$ and $\hat{\mathbf{h}}_\mathbf{i}^\mathbf{T}$, $i \in [1, K]$, where $\mathbf{\Theta_i^T} \in \mathbb{R}^\mathbf{N}$ and $\hat{\mathbf{h}}_\mathbf{i}^\mathbf{T} \in \mathbb{R}^\mathbf{M}$. We have the following representation:

$$\mathbf{\Theta} = \begin{bmatrix} — & \boldsymbol{\theta}_\mathbf{1}^\mathbf{T} & — \\ & \vdots & \\ — & \boldsymbol{\theta}_\mathbf{K}^\mathbf{T} & — \end{bmatrix}_{(K\times N)} \quad \hat{\mathbf{H}} = \begin{bmatrix} — & \hat{\mathbf{h}}_\mathbf{1}^\mathbf{T} & — \\ & \vdots & \\ — & \hat{\mathbf{h}}_\mathbf{K}^\mathbf{T} & — \end{bmatrix}_{(K\times M).}$$

Given the above formulation, we can simplify and represent the ith row of (4) as follows:

$$\boldsymbol{\theta}_\mathbf{i}^\mathbf{T}\mathbf{H} = \hat{\mathbf{h}}_\mathbf{i}^\mathbf{T} \;\Rightarrow\; \boldsymbol{\theta}_\mathbf{i} = \mathbf{D}\hat{\mathbf{h}}_\mathbf{i} \quad (15)$$

where $\mathbf{D^T} = \mathbf{\Phi H^{-1}}$. Note that $\mathbf{D}$ in the above equation is of dimensionality $N \times M$. Suppose the SVD of $\mathbf{D}$ is $\mathbf{USV^T}$, where orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ are of dimensionality $N \times M$ and $M \times M$, respectively, and the diagonal matrix $\mathbf{S}$, comprising the singular values of $\mathbf{D}$, is of dimensionality $M \times M$. Then, we can simplify (15) as follows:

$$\theta_i = \mathbf{D}\hat{\mathbf{h}}_\mathbf{i} = \mathbf{USV^T}\hat{\mathbf{h}}_\mathbf{i}. \quad (16)$$

Since we seek $\theta_i \sim N(0, I_\mathbf{N})$, to preserve the inner products according to the JL lemma, we draw $\hat{\mathbf{H}}_i$ from $N(0, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = \mathbf{VS^{-2}V^T}$. Then, we derive each row of $\mathbf{\Theta}$ based on (16). This choice of $\hat{\mathbf{H}}_i$, in fact, gives the exact JL solution for $\hat{\mathbf{H}}$ according to the following corollary.

Corollary 1 (JL Solution for $\hat{\mathbf{H}}$): Given orthogonal matrices $\mathbf{U}$, $\mathbf{V}$ of dimension $N \times M$ and $M \times M$, respectively, and an $M \times M$ diagonal matrix of singular values $\mathbf{S}$. Then, $\hat{\mathbf{h}}_\mathbf{i} \sim N(0, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = \mathbf{VS^{-2}V^T}$ and $\hat{\mathbf{h}}_\mathbf{i} \in \mathbb{R}^\mathbf{M}$, gives the solution for $\theta_i = \mathbf{USV^T}\hat{\mathbf{h}}_\mathbf{i}$ such that the entries of the row vector $\mathbf{\Theta_i}$ are drawn i.i.d from the multivariate normal $N(0, \mathbf{I_N})$.

Proof: We complete the proof by deriving the mean and variance of $\hat{\mathbf{h}}_\mathbf{i}$ under the assumption of $\theta_\mathbf{i} \sim N(0, I_M)$. Consider the following equation:

$$\theta_i = \mathbf{USV^T}\hat{\mathbf{h}}_\mathbf{i} = \mathbf{Uz_i} \quad (17)$$

where $\mathbf{z_i} = \mathbf{SV^T}\hat{\mathbf{h}}_\mathbf{i}$ is an M-dimensional vector of random variables. Since $\theta_i \sim N(0, I_M)$ and $\mathbf{U}$ is a constant matrix, $z_i \sim N(0, \mathbf{I_N})$. Further, since $\hat{\mathbf{h}}_\mathbf{i} = \mathbf{VS^{-1}z_i}$, we can compute the mean of $\hat{\mathbf{h}}_\mathbf{i}$ as $\mathbb{E}\left[\hat{\mathbf{h}}_\mathbf{i}\right] = \mathbb{E}\left[\hat{\mathbf{z}}_\mathbf{i}\right] = 0$, and the variance of $\hat{\mathbf{h}}_\mathbf{i}$ as follows:

$$\begin{aligned}\mathbb{E}\left[\hat{\mathbf{h}}_\mathbf{i}\hat{\mathbf{h}}_\mathbf{i}^\mathbf{T}\right] &= \mathbb{E}\left[\mathbf{VS^{-1}z_iz_i^TS^{-1}V^T}\right] \\ &= \mathbf{VS^{-1}}\,\mathbb{E}\left[\mathbf{z_iz_i^T}\right]\mathbf{S^{-1}V^T} \\ &= \mathbf{VS^{-2}V^T}. \end{aligned}$$ ■

Thus, the approximate solution for matrix $\hat{\mathbf{H}}$ is of dimension $K \times M$, where $K < M$ (or $\nu > \xi$).

Appendix C
Solution for $\hat{\mathbf{H}}$ When $\mathbf{H}$ is Nonsquare

To solve (12) for $\mathbf{\Theta}$ and $\hat{\mathbf{H}}$, we take the transpose of (4) and multiply with itself, obtaining the following relationship:

$$\begin{aligned}(\mathbf{\Theta H})^\mathbf{T}(\mathbf{\Theta H}) &= (\hat{\mathbf{H}}\mathbf{\Phi})^\mathbf{T}(\hat{\mathbf{H}}\mathbf{\Phi}) \\ \mathbf{H^T\Theta^T\Theta H} &= \mathbf{\Phi^T\hat{H}^T\hat{H}\Phi} \\ \mathbf{RQP^T\Theta^T\Theta PQR^T} &= \mathbf{USV^T\hat{H}^T\hat{H}VSU^T} \end{aligned} \quad (18)$$

where $\mathbf{H} = \mathbf{PQR^T}$ and $\Phi = \mathbf{VSU^T}$ are the SVDs of $\mathbf{H}$ and $\mathbf{\Phi}$, respectively. Since $\mathbf{H}$ is of dimensionality $L \times N$ ($L < N$), $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{R}$ is of dimensionality $L \times L$, $L \times L$, and $N \times L$, respectively. Similarly, since $\mathbf{\Phi}$ is of dimensionality $M \times N$ ($M < N$), $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ are of dimensionality $N \times M$, $M \times$

M, and $M \times M$, respectively. If we let $\mathbf{\Theta} = \mathbf{BQ^{-1}P^T}$ and $\hat{\mathbf{H}} = \mathbf{AS^{-1}V^T}$ in (18), we have the following relationship:

$$\mathbf{RB^TBR^T} = \mathbf{UA^TAU^T}$$
$$\Rightarrow \mathbf{U^TRB^TBR^TU} = \mathbf{A^TA}$$

where $\mathbf{A}$ and $\mathbf{B}$ are unknown matrices that need to be determined. We can invoke the JL lemma and draw the $K \times L$ elements of $\mathbf{\Theta}$ from $N(0, 1)$. We can thus solve for the $K \times L$ matrix $\mathbf{B} = \mathbf{\Theta PQ}$ and use the above equation to derive the $K \times M$ matrix $\mathbf{A} = \mathbf{BR^TU}$. Finally, we obtain the $K \times M$ matrix $\hat{\mathbf{H}} = \mathbf{AS^{-1}V^T}$.

Appendix D
Formulating DWT as a Matrix Operation

In the filter bank implementation, the DWT of a signal is derived by passing it through a series of filters. First, vector $\mathbf{x}$ is passed simultaneously through a low pass filter (LPF) and a high pass filter (HPF). Each of these filters has half the bandwidth of the signal. Thus, the output of these filters are downsampled by $2\times$ without the risk of aliasing. This comprises one level of wavelet decomposition. The process is repeated with the LPF outputs to achieve higher levels of decomposition [37]. To formulate the entire process as a matrix operation in NA, we note that the processing between a vector of filter coefficients $\mathbf{g}$ and the N-sample spike vector $\mathbf{x}$ can be represented as a convolution operation

$$\mathbf{z} = \mathbf{g} * \mathbf{x} = \sum_{k=-\infty}^{\infty} g[n-k]x[k] = \mathbf{G_N x} \tag{19}$$

where $\mathbf{z}$ is the filtered signal of N samples and $\mathbf{G_N}$ is the $N \times N$ convolution matrix whose rows are shifted versions of the coefficient vector $\mathbf{g}$. For the DWT algorithm, $\mathbf{G_N^L}$ and $\mathbf{G_N^H}$ can be used to represent the LPF and HPF operations, respectively. After the filtering process, we can then implement downsampling by $2\times$ at each level of decomposition through an $N/2 \times N$ matrix $\mathbf{D_{2,N}}$

$$\mathbf{D_{2,N}} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{(N/2 \times N).}$$

Using a cascade of **D**-**G** operators, we can thus represent the full DWT operation in NA as the following linear transformation:

$$\mathbf{y} = \mathbf{Hx} = \begin{bmatrix} \mathbf{H_1} \\ \mathbf{H_2} \\ \vdots \\ \mathbf{H_{L+1}} \end{bmatrix}_{N \times N} \begin{bmatrix} \mathbf{x} \end{bmatrix}_{N \times 1} \tag{21}$$

where $\mathbf{y}$ is the N-sample DWT of spike samples $\mathbf{x}$. For L levels of decomposition, sub-matrices $\mathbf{H_n}$ $(1 \le n \le L+1)$ are given by

$$\mathbf{H_n} = \begin{cases} \mathbf{D_{2,N} G_N^H} & \text{if } n = 1 \\ \prod_{\mathbf{k=0}}^{\mathbf{n-2}} \left(\mathbf{D_{2,N/2^k} G_{N/2^k}^L}\right)\left(\mathbf{D_{2,N/2^{n-1}} G_{N/2^{n-1}}^H}\right) & \text{if } 2 \le n \le L \\ \prod_{\mathbf{k=0}}^{\mathbf{n-1}} \left(\mathbf{D_{2,N/2^k} G_{N/2^k}^L}\right) & \text{if } n = L+1. \end{cases}$$

Each pair of matrices, $\mathbf{G_{N/2^j}^L}$ and $\mathbf{G_{N/2^j}^H}$, in the above equation is designed to be a quadrature mirror filter based on standard mother wavelets, e.g., Haar, Daubechies, Coiflet, biorthogonal wavelet, and so on [37].

Acknowledgment

The authors would like to thank X. Cheng and A. Singer from the Applied and Computational Mathematics Department at Princeton University for valuable discussions on low-dimensional error bounds for the compressed-domain equations.

References

[1] E. J. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
[2] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 92–101, Mar. 2008.
[3] F. Fazel, M. Fazel, and M. Stojanovic, "Random access compressed sensing for energy-efficient underwater sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1660–1670, Sep. 2011.
[4] L.-W. Kang and C.-S. Lu, "Distributed compressive video sensing," in *Proc. IEEE ICASSP*, Apr. 2009, pp. 1169–1172.
[5] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2456–2466, Sep. 2011.
[6] M. Shoaib, N. K. Jha, and N. Verma, "Enabling advanced inference on sensor nodes through the direct use of compressively-sensed signals," in *Proc. IEEE DATE Conf.*, Mar. 2012, pp. 437–443.
[7] S. Aviyente, "Compressed sensing framework for EEG compression," in *Proc. IEEE/SP 14th Workshop SSP*, Aug. 2007, pp. 181–184.
[8] S. Ganguli and H. Sompolinsky, "Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis," *Annu. Rev. Neurosci.*, vol. 35, pp. 485–505, Jul. 2012.
[9] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *J. Neural Comput.*, vol. 15, no. 2, pp. 349–396, Feb. 2003.
[10] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[11] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.
[12] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
[13] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst., Comput.*, Nov. 1993, pp. 40–44.
[14] 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 Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.
[15] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007.
[16] P. K. Baheti and H. Garudadri, "An ultra low power pulse oximeter sensor based on compressed sensing," in *Proc. 6th Int. Workshop Wearable Implant. Body Sensor Netw.*, Jun. 2009, pp. 144–148.
[17] A. S. Alvarado, C. Lakshminarayan, and J. C. Principe, "Time-based compression and classification of heartbeats," *IEEE Trans. Biomed. Eng.*, vol. 6, no. 1, pp. 1–9, Jan. 2007.

[18] N. Verma, A. Shoeb, J. Bohorquez, J. Dawson, J. Guttag, and A. P. Chandrakasan, "A micro-power EEG acquisition SoC with integrated feature extraction processor for a chronic seizure detection system," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 804–816, Apr. 2010.

[19] M. Shoaib, N. K. Jha, and N. Verma, "A low-energy computation platform for data-driven biomedical monitoring algorithms," in *Proc. 48th ACM/EDAC/IEEE DAC*, Jun. 2011, pp. 591–596.

[20] R. J. Durrant and A. Kaban, "Compressed Fisher linear discriminant analysis: Classification of randomly projected data," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, Jul. 2010, pp. 1119–1128.

[21] R. Calderbank, S. Jafarpour, and R. Schapire, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," Dept. Comput. Sci., Princeton Univ., Princeton, NJ, USA, Tech. Rep., Dec. 2009.

[22] S. Zhou and J. Lafferty, "Compressed and privacy-sensitive sparse regression," *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 846–866, Feb. 2009.

[23] O. A. Maillard and R. Munos, "Compressed least-squares regression," in *Proc. NIPS*, 2009, pp. 1–9.

[24] J. E. Fowler, "Compressive-projection principal component analysis," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2230–2242, Oct. 2009.

[25] C. Hegde, M. B. Wakin, and R. G. Baraniuk, "Random projections for manifold learning," in *Proc. Conf. Neural Inf. Process. Syst.*, Dec. 2007, pp. 15–23.

[26] N. Verma, K. H. Lee, and A. H. Shoeb, "Data-driven approaches for computation in intelligent biomedical devices: A case study of EEG monitoring for chronic seizure detection," *J. Low Power Electron. Appl.*, vol. 1, no. 1, pp. 150–174, Apr. 2011.

[27] 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.

[28] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Statist.*, vol. 36, no. 3, pp. 1171–1220, Jun. 2008.

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

[30] M. Rudelson and R. Vershynin, "Non-asymptotic theory of random matrices: Extreme singular values," in *Proc. Int. Congr. Math.*, vol. 3, pp. 1576–1602, Aug. 2010.

[31] S. Mandal and R. Sarpeshkar, "Power-efficient impedance-modulation wireless data links for biomedical implants," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 4, pp. 301–315, Dec. 2008.

[32] R. R. Harrison, P. T. Watkins, R. J. Kier, R. O. Lovejoy, D. J. Black, B. Greger, *et al.*, "A low-power integrated circuit for a wireless 100-electrode neural recording system," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 123–133, Jan. 2007.

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

[34] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004.

[35] K. V. Shenoy, D. Meeker, S. Cao, S. A. Kureshi, B. Pesaran, C. A. Buneo, *et al.*, "Neural prosthetic control signals from plan activity," *Neuroreport*, vol. 14, no. 4, pp. 591–596, Mar. 2003.

[36] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen, "Cognitive control signals for neural prosthetics," *Science*, vol. 305, no. 5681, pp. 258–262, Jul. 2004.

[37] T. K. Sarkar, C. Su, R. Adve, M. Salazar-Palma, L. Garcia-Castillo, and R. R. Boix, "A tutorial on wavelets from an electrical engineering perspective I: Discrete wavelet techniques," *IEEE Mag. Antennas Propag.*, vol. 40, no. 5, pp. 49–68, Oct. 1998.

[38] E. Candes and J. Romberg. (2005, Oct.). *l_1-MAGIC: Recovery of Sparse Signals Via Convex Programming* [Online]. Available: http://www.acm.caltech.edu/l1magic/downloads/l1magic.pdf

[39] E. V. D. Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada, Tech. Rep. TR-2008-01, Jan. 2008.

[40] A. Shoeb and J. Guttag, "Application of machine learning to seizure detection," in *Proc. ICML*, Jun. 2010, pp. 1–8.

[41] Physionet. Cambridge, MA, USA. (2013, Oct.). *CHB-MIT Physionet Database* [Online]. Available: http://www.physionet.org/physiobank/database

Mohammed Shoaib (S'08–M'13) received the B.Tech. and M.Tech. degrees in electrical engineering from IIT, Madras, India, in 2007 and 2008, respectively, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA, in 2010 and 2013, respectively.

He is a Researcher with Microsoft Research, Redmond, WA, USA. He has authored or co-authored more than 20 technical papers and holds 4 U.S. patents. His current research interests include the design of energy-efficient platforms for sensing and computing.

Niraj K. Jha (S'85–M'85–SM'93–F'98) received the B.Tech. degree in electronics and electrical communication engineering from IIT, Kharagpur, India, in 1981, the M.S. degree in electrical engineering from the State University of New York at Stony Brook, Stony Brook, NY, USA, in 1982, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 1985.

He is a Professor of electrical engineering with Princeton University, Princeton, NJ, USA. He has co-authored or co-edited 5 books, 12 book chapters, and more than 390 technical papers. He has co-authored 14 papers, which have won various awards, and another six papers that have been nominated for best paper awards. He holds 14 U.S. patents. He has given several keynote speeches in the area of nanoelectronic design and test. His current research interests include FinFETs, low power hardware/software design, computer-aided design of integrated circuits and systems, quantum computing, and secure computing.

Dr. Jha is a fellow of ACM. He has served as an Editor-in-Chief of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, and an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, and the *Journal of Electronic Testing: Theory and Applications*. He is currently serving as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTERS, the *Journal of Low Power Electronics*, and the *Journal of Nanotechnology*. He has served as the Program Chairman of the 1992 Workshop on Fault-Tolerant Parallel and Distributed Systems, the 2004 International Conference on Embedded and Ubiquitous Computing, and the 2010 International Conference on VLSI Design. He has served as the Director of the Center for Embedded System-on-a-Chip Design funded by the New Jersey Commission on Science and Technology. He is a recipient of the AT&T Foundation Award and the NEC Preceptorship Award for research excellence, the NCR Award for teaching excellence, and the Princeton University Graduate Mentoring Award.

Naveen Verma (M'09) received the B.A.Sc. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada, in 2003, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2005 and 2009, respectively.

He has been an Assistant Professor of electrical engineering with Princeton University, Princeton, NY, USA, since 2009. His focus spans low-voltage digital logic and SRAMs, low-noise analog instrumentation and data-conversion, and integrated power management. His current research interests include ultralow-power integrated circuits and systems with an emphasis on sensing applications. Of particular importance is the use of emerging devices for the creation of functionally-diverse systems and the use of advanced signal-analysis frameworks for low-power inference over embedded signals.

Dr. Verma was a co-recipient of the 2008 ISSCC Jack Kilby Award for Outstanding Student Paper and the 2006 DAC/ISSCC Student Design Contest Award. He was a recipient of the Alfred Rheinstein Junior Faculty Award from Princeton University and the 2013 National Science Foundation CAREER Award.