

# Ultra-Low Power Dynamic Knob in Adaptive Compressed Sensing Towards Biosignal Dynamics

Aosen Wang, *Student Member, IEEE*, Feng Lin, *Member, IEEE*, Zhanpeng Jin, *Member, IEEE*, and Wenyao Xu, *Member, IEEE*

**Abstract**—Compressed sensing (CS) is an emerging sampling paradigm in data acquisition. Its integrated analog-to-information structure can perform simultaneous data sensing and compression with low-complexity hardware. To date, most of the existing CS implementations have a fixed architectural setup, which lacks flexibility and adaptivity for efficient dynamic data sensing. In this paper, we propose a dynamic knob (DK) design to effectively reconfigure the CS architecture by recognizing the biosignals. Specifically, the dynamic knob design is a template-based structure that comprises a supervised learning module and a look-up table module. We model the DK performance in a closed analytic form and optimize the design via a dynamic programming formulation. We present the design on a 130 nm process, with a 0.058 mm<sup>2</sup> fingerprint and a 187.88 nJ/event energy-consumption. Furthermore, we benchmark the design performance using a publicly available dataset. Given the energy constraint in wireless sensing, the adaptive CS architecture can consistently improve the signal reconstruction quality by more than 70%, compared with the traditional CS. The experimental results indicate that the ultra-low power dynamic knob can provide an effective adaptivity and improve the signal quality in compressed sensing towards biosignal dynamics.

**Index Terms**—Biosignal dynamics, compressed sensing, dynamic knob, dynamic programming.

## I. INTRODUCTION

COMPRESSED SENSING (CS) is a rapidly growing field that has attracted considerable attention in diverse domains. Since its initial introduction [1], [2] in 2006, an avalanche of results have been obtained, both of theoretical and practical nature. CS offers a framework for simultaneous sensing and compression of analog front-end design for energy-efficient wireless sensing. Specifically, CS exploits signal structure and samples the information in the signal, namely

Manuscript received May 29, 2015; revised August 26, 2015; accepted October 20, 2015. Date of publication January 19, 2016; date of current version March 04, 2016. This work was supported in part by NSFCNS-1423061/1422417, ECCS-1462498/146247 and CNS-1547167. This paper was recommended by Associate Editor S. Chakrabarty.

A. Wang, F. Lin and W. Xu are with the Department of Computer Science and Engineering, State University of New York (SUNY) at Buffalo, Buffalo, NY 14260 USA (e-mail: aosenwan@buffalo.edu; flin28@buffalo.edu; wenyaoxu@buffalo.edu).

Z. Jin is with the Department of Electrical and Computer Engineering, Binghamton University, State University of New York (SUNY), Binghamton, NY 13902 USA (e-mail: zjin@binghamton.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/TBCAS.2015.2497304

*analog-to-information* conversion. The CS-based analog-to-information convertor usually has a low hardware complexity, and can provide sub-Nyquist sampling rate. Currently, CS has been widely applied to diverse and *energy-constrained* wireless sensing-involved applications from large-scale networks to personal implantable and wearable devices [3]–[6]. The ultra-low power sensing technique also provides a promising way to promote the new *self-powered* medical devices [7], [8].

Biosignals from the human body tend to change dynamically, and these signals are governed by the physics of the human body. These properties endow the time-varying sparsity nature of biosignals, also known as biosignal dynamics. Therefore, it seems promising to obtain a significant gain (such as a better sensing efficiency in CS architecture) if the sensing mechanism is coherent with the dynamic wave and self-adapts to the setup. However, there is no sophisticated research yet on CS towards biosignal dynamics, and the state-of-the-art CS structure is rigid and lacks adaptivity, which considerably degrades the energy-efficiency in structure-variational signal acquisition.

In this paper, we propose a dynamic knob (DK) in CS to improve the adaptivity towards biosignal dynamics. We first analyze the design of the DK, which requires an effective configuration control and an ultra-low power design. According to these design aims, we develop a template-based structure of the DK based on a supervised learning algorithm. Specifically, we design a support vector machine (SVM)-based cascaded signal analyzer to recognize the signal among different categories and configure CS architecture via a pre-defined configuration look-up table. For circuit-level implementation of the DK, we model the design performance in closed form and formulate it into a dynamic programming problem. We present our design at a 130nm standard cell library which consumes the energy of 187.88 nJ/event, with a 0.058 mm<sup>2</sup> footprint. We benchmark the performance of the adaptive CS architecture with the dataset from Physionet [9], in which signals include electroencephalography (EEG), electrocardiography (ECG) and electrooculography (EOG). The evaluation results indicate that compared with the traditional CS, our CS architecture with the proposed DK can improve signal reconstruction quality by more than 70% with dynamic data sampling under the given energy constraint of wireless sensing.

In this paper, our main contribution is four-fold:

- We propose and design a dynamic knob structure to enable the adaptivity in CS towards biosignal dynamics. The dynamic knob is a template-based structure that includes a supervised learning model and a configuration look-up table.

- We model the design performance in the circuit level and formulate it into a dynamic programming problem. The design optimization can be accomplished within polynomial time.
- We present the design and implementation of dynamic knob in 130nm standard cells. This design is ultra-low power with high efficiency.
- We conduct a comprehensive evaluation of the adaptive CS architecture that reveals the necessity and feasibility of the new CS architecture in dynamic data sensing.

The remaining of the paper is organized as follows: Section II describes prior work of adaptive sensing and CS. The basics of the conventional CS theory and its architecture in wireless sensing are described in Section III. Our proposed efficient dynamic knob design towards biosignal dynamics is introduced in Section IV. Section V presents the implementation of the dynamic knob on an application-specific integrated circuit (ASIC), with its problem formulation and corresponding solution. Evaluations and experimental results are discussed in Section VI. Section VII is the conclusion of our paper and the plan for future work.

## II. RELATED WORK

Traditional CS theory is a non-adaptive sampling scheme. The reconstruction algorithms are mostly referring to the sparsity assumption, such as the  $\ell_1$ -based approach [10] and orthogonal matching pursuit (OMP) [11]. They can give a good performance according to the prior knowledge. However, to deal with the data dynamics, time-varying sparsity, the condition becomes more complex.

Prior research has proposed several methods to deal with the data dynamics. Their adaptivity assumptions are mainly based on two aspects: sensing matrix and compression ratio, also known as the compressive measurement number. For the sensing matrix, Malloy and Nowak proposed a compressive adaptive sense and search (CASS) [12] method to update the matrix on each iteration. Although it gains better performance, it needs to obtain the signal sparsity ahead of time. The research work of [13] presented an algorithm called adaptive compressed sensing (ACS) to optimize the measurement number. This method traverses possible measurement dimensions to compare the reconstruction signal qualities and determine the optimal compression ratio, which is computationally intensive. To address this challenge, Wang *et al.* [14] presented another algorithm with similar concerns of ACS, called adaptive measurement adjustment (AMA). The measurement dimensions are changed through an approach similar to the bisection method, and reconcile the inner contradiction between the interval and the accuracy. There is also some work dealing with these two factors together. Zahedi *et al.* [15] developed a multi-step look-ahead design to simultaneously consider the sensing matrix and compression ratio. Experiments verify its significant advantage over non-adaptive designs. These works are all simulation-based without referring to the circuit-level details. Also, they do not provide any specific scheme towards the data dynamics.

## III. BACKGROUND AND PRELIMINARY

### A. Compressed Sensing Theory

CS theory is a newly emerging analog-to-information sampling scheme for the signals that are known to be sparse or compressible under certain basis. We assume  $\mathbf{x}$  lies in an  $N$ -dimension vector space and is sampled using  $M$ -measurement vector  $\mathbf{y}$

$$\mathbf{y} = \Phi \mathbf{x} \quad (1)$$

where  $\Phi \in R^{M \times N}$  is the sensing array, which models the linear encoding, and  $M$  is defined as the sampling rate in  $N$ -dimensional CS. The elements in  $\Phi$  are either Gaussian random variables or Bernoulli random variables [16]. Because of  $M \ll N$ , the formulation in (1) is undetermined, and signal  $x$  cannot be uniquely retrieved from the sensing array  $\Phi$  and measurements  $\mathbf{y}$ . However, under certain sparsity-inducing basis  $\Psi \in R^{N \times N}$ , the signal  $x$  can be represented by a set of sparse coefficients  $\mathbf{u} \in R^N$

$$\mathbf{x} = \Psi \mathbf{u} \quad (2)$$

that is, the coefficient vector  $\mathbf{u}$ , under the transformation  $\Psi$ , has few non-zero elements. Therefore, based on (1) and (2),  $\mathbf{y}$  can be represented as follows:

$$\mathbf{y} = \Phi \Psi \mathbf{u} = \Theta_{M \times N} \mathbf{u} \quad (3)$$

where  $\Theta_{M \times N} = \Phi \Psi$  is an  $M \times N$  array, called the measuring matrix. Due to the prior knowledge that the unknown vector,  $\mathbf{u}$ , is sparse, it is possible to estimate the value,  $\mathbf{u}$ , using the  $\ell_0$  minimization formulation as follows:

$$\mathbf{u} = \min \|\mathbf{u}\|_0 \text{ s.t. } \|\mathbf{y} - \Theta \mathbf{u}\| < \epsilon \quad (4)$$

where  $\epsilon$  is the reconstruction error margin. The formulation in (4) is a determined system with a unique solution. However, the  $\ell_0$  minimization is an NP-hard problem [17]. One of the methods to solve (4) is to approximate the  $\ell_0$  minimization formulation to the  $\ell_1$  minimization formulation

$$\mathbf{u} = \min \|\mathbf{u}\|_1 \text{ s.t. } \|\mathbf{y} - \Theta \mathbf{u}\| < \epsilon. \quad (5)$$

Under the condition of restricted isometry property (RIP) [18], the  $\ell_1$  problem has been theoretically proven equivalent to minimizing the  $\ell_0$  problem. The  $\ell_1$  minimization is convex and can be solved within a polynomial time.

However, the above theory did not take the quantization into account. In the practical applications, original signals are analog in nature and need to be quantized before transmission and processing. Therefore, the compressed signal,  $\mathbf{y}$ , should be processed by a quantization model formulated as follows:

$$\hat{\mathbf{y}} = Q_b(\mathbf{y}) \quad (6)$$

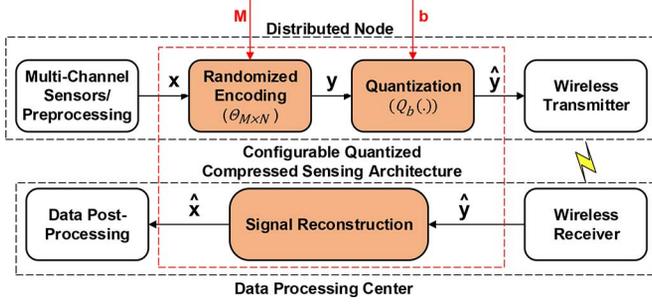


Fig. 1. The conventional configurable quantized CS architecture.

where  $Q_b(\cdot)$  is the quantization function [19], and  $\hat{y}$  is the quantized representation of  $y$  with  $b$  bits. When considering the quantization process into the CS architecture, the  $\ell_1$  formulation can be reformulated

$$\hat{\mathbf{u}} = \min \|\mathbf{u}\|_1 \text{ s.t. } \|\hat{\mathbf{y}} - \Theta \mathbf{u}\| < \epsilon. \quad (7)$$

By solving the formulation in (7), we can obtain  $\hat{\mathbf{u}}$ , the sparse representation in the quantized measure  $\hat{\mathbf{y}}$ . Therefore, the reconstructed signal,  $\hat{\mathbf{x}}$ , is retrieved by

$$\hat{\mathbf{x}} = \Psi \hat{\mathbf{u}}. \quad (8)$$

### B. Quantized Compressed Sensing Architecture

The entire sensing framework based on the formulation in (8) is named quantized compressed sensing (QCS) architecture, which is illustrated as Fig. 1.

We can see that the QCS architecture consists of randomized encoding, quantization and signal reconstruction modules. Original analog signals, which usually denote the raw analog data,  $\mathbf{x} \in R^N$ , coming from sensors, are encoded into the  $M$ -dimensional vector,  $\mathbf{y} \in R^M$ , by linear encoding  $\Phi$ . Through the quantization scheme  $Q_b(\cdot)$ , every measurement becomes a certain  $b$ -bit digital representation,  $\hat{y}$ . A wireless transmitter streams these measurements data to the receiver. When the wireless receiver gets the data extracted from the bit stream, it performs reconstruction algorithms to recover the  $N$ -dimension original input signal  $\mathbf{x}$  from the quantized  $M$ -dimension compressed measurements  $\mathbf{y}$ . The reconstructed signal  $\hat{\mathbf{x}}$  is sent to the data post-processing module for specific applications.

In this part, we also introduce the models of energy and performance [20], [21] in traditional QCS architecture. For distributed nodes, the power consumption is dominated by the volume of the data stream in wireless communication. So the energy model can be formulated as follows:

$$E = C \times M \times b \quad (9)$$

where  $M$  is the sampling rate,  $b$  is the bit resolution in quantization, and  $C$  is the energy per bit which is determined by the wireless communication protocol and usually a constant. We use the signal reconstruction error as the performance metric in the

data center. Therefore, the performance model in the architecture can be defined as follows:

$$P(M, b) = \frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \times 100\% \quad (10)$$

where  $P(M, b)$  denotes the performance metric involving the sampling rate  $M$  and the resolution bit  $b$ .  $\hat{\mathbf{x}}$  denotes the reconstructed signal, and  $\mathbf{x}$  is the original input signal.

## IV. DYNAMIC KNOB IN QUANTIZED COMPRESSED SENSING ARCHITECTURE

In this section, we discuss the dynamic knob design for the QCS architecture. We first discuss the significance and design consideration for the dynamic knob. Then we elaborate the entire dynamic knob architecture structure considering these concerns.

### A. Design Consideration

*Adaptive Control:* The CS theory is a prevalent sampling scheme to acquire the signals by a rate proportional to their intrinsic information. However, in practice, the volume of intrinsic information in biomedical signals always varies. We name it *Data Dynamics*. The biosignal dynamics is a significant factor for an efficient biomedical sensing task. Thus, dynamic parameter adjustment of CS architecture, such as the sampling rate and bit resolution in quantization described in the preliminary section, is in great demand to accommodate biosignal dynamics. For the characteristics of streaming data processing, the knob design needs to make the parameter judgement by high accuracy of the optimal configuration parameters estimation, which directly determines the final performance of the entire CS architecture.

*Ultra-Low Power Design:* To collaborate with other components in the CS architecture, the dynamic knob functions as the central control. The sensor node in wireless sensing is powered by limited-life battery, so the knob should be an ultra-low power design to maintain the long-term data acquisition. Thus, our dynamic knob design should reduce the energy consumption from both hardware and algorithm levels. We need to choose hardware with low energy consumption, such as an ASIC platform. For the algorithm level, our dynamic knob design should search for a perfect trade-off between power and accuracy.

### B. Dynamic Knob Framework

Towards biosignal dynamics in wireless sensing, we propose a template-based supervised learning dynamic knob framework to configure parameters in randomized encoding module and quantization module. The entire framework consists of two key components, the signal structure analyzer module and the parameter look-up table module. The signal structure analyzer identifies the input signal based on supervised learning algorithm. By identifying the input signal, the optimal parameter estimation can be queried from the look-up table module according to the pre-defined templates. The entire block diagram of the framework is shown in Fig. 2.

We can see that the dynamic knob controls the other two functional modules, randomized encoding and quantization, by

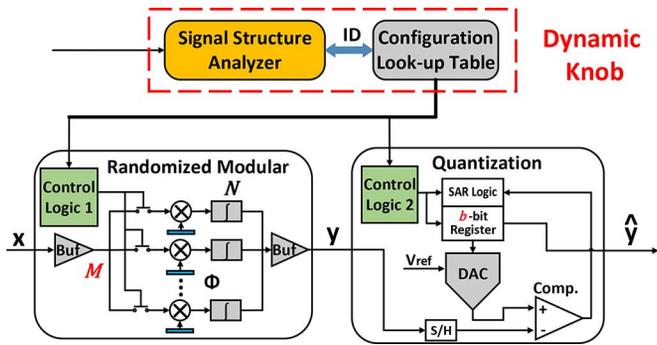


Fig. 2. The block diagram of dynamic knob including circuit-level details of randomized encoding and quantization.

analyzing the input signal waveform. We provide the circuit-level details of these two modules in Fig. 2 for the sake of indicating how our knob works. The analog  $N$ -dimension raw sensor signal  $\mathbf{x}$  is compressed into  $M$ -dimension  $\mathbf{y}$  in the randomized encoding module. It consists of many branches to compress the signal, each of which completes a linear combination of input signals. Every branch includes a multiplier, a column vector in the sensing array  $\Phi$ , and an integrator to accumulate the intermediate results. The logic control module controls the switches before the multipliers, which enables the branch when they are closed. These switches decide how many branches to participate in the compression calculation. Thus, the sampling rate  $M$  can be configured dynamically by this control logic. Consequently, the compressed signal  $\mathbf{y}$  must be quantified for wireless transmission which can only communicate the digitalized signal.

The quantization module comprises a successive approximation register (SAR) [22] logic, a  $b$ -bit register, a digital-to-analog converter (DAC) and a comparator. When the compressed measurements come, the SAR logic sets the valid bit for the  $b$ -bit register. Then the digital register is transformed to analog value by DAC and compared with the analog input signal by the comparator. The comparing result provides feedbacks for SAR logic for next-round bit setting. This procedure iteratively continues from the MSB to LSB of the  $b$ -bit register to approximate the analog input signal. We also add another control logic to enable modifications of bit width  $b$  in the SAR logic and its register. Thus, the quantization bit resolution  $b$  can be reconfigurable by this control logic. Therefore, to manipulate the randomized encoding module and the quantization module, our dynamic knob directly configures their control logics. After the input signal waveform comes to the dynamic knob, the signal structure analyzer judges its category information based on supervised learning algorithms. Once obtaining the signal category information, the optimal parameter estimation can be queried from the parameter look-up table module to configure the corresponding components.

It is noted that the most significant part in dynamic knob design is the signal structure analyzer module. The more accurate the parameter configuration is, the better system performance can be obtained. For the dynamic knob design, the performance of look-up table module is directly related to the memory management techniques, which can be very low-power. Therefore,

our biggest challenge in this framework is how to implement an ultra-low power consumption, as well as a fast, and high accuracy signal structure analyzer.

## V. ENERGY-AWARE DESIGN AND OPTIMIZATION OF DYNAMIC KNOB

In this section, we present the SVM-based primitive model of the signal structure analyzer in dynamic knob. For optimal design under this primitive model, we formulate an optimization problem, taking accuracy and energy consumption into account, and solve it by a polynomial-time algorithm.

### A. Support Vector Machine

SVM [23] is one of the most widely applied supervised machine learning algorithms. It divides high-dimension feature space into two subspaces and classifies the data by searching for the optimal hyperplane as the decision plane. The SVM is widely researched and used in many fields for its fewer tuning parameters yet high recognition rate. The formulation of training phase of SVM is a standard quadratic programming problem, whose form is as the following:

$$\min \frac{1}{2} \|\mathbf{w}\|^2, \text{ s.t. } y_i(K(\mathbf{w}, \mathbf{x}_i) - b) \geq 1 \quad (11)$$

where  $\mathbf{w}$  is the perpendicular vector to the hyperplane's direction,  $\mathbf{x}_i$  is the training data,  $y_i$  is the label of the training data  $\mathbf{x}_i$ ,  $K(\cdot, \cdot)$  is the kernel function and  $b$  is the offset to the origin. The training task is to calculate the perpendicular vector  $\mathbf{w}$ . There are many methods presented for this problem. The sequential minimal optimization (SMO) algorithm [24] is one of the most favored methods for its robust performance and fast speed. After finishing training the SVM model, the classification task on new testing data can be implemented by the following:

$$f(x) = \text{sgn} \left( \sum_i y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \right) \quad (12)$$

where,  $\text{sgn}(\cdot)$  is the sign function,  $\mathbf{x}$  is the new input testing feature vector,  $\mathbf{x}_i$  is the supporting vector,  $y_i$  is its corresponding label, and  $\alpha_i$  is the weight of the supporting vector. Therefore, the testing phase can give its binary decision for the classification of new testing data.

### B. Signal Structure Analyzer

Our signal structure analyzer model takes binary SVM as the basic supervised learning algorithm. However, the analyzer is faced with the multi-class classification problem. We adopt a cascade binary SVM classifier to accomplish the multi-class task. There are four popular strategies to solve multi-class classification problems based on the SVM algorithm, one-to-one SVM, one-to-all SVM, directed acyclic graph (DAG) SVM, and binary-tree SVM. The one-to-one scheme [25], [26] indicates to build SVM classifier between any two categories in the training data. In the testing phase, every classifier gives its classification decision and the winner is the one with the most votes. The

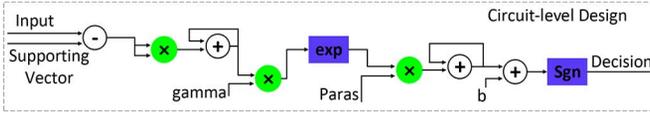


Fig. 3. The data flow of SVM testing phase on the circuit level.

one-to-all SVM [25], [26] identifies one class for each classifier until all the classes are classified. DAG SVM scheme [27] first trains all the one-to-one SVM classifiers, and then builds a directed acyclic graph by these classifiers for multi-class classification. The binary-tree SVM organizes all the basic SVMs by the form of binary tree and makes decisions hierarchically.

In this paper, we provide a design of a cascade SVM scheme for our signal structure analyzer, whose primitive element is a binary SVM classifier. For the analyzer design, low energy is a compulsory requirement and our goal is different with the previous binary tree SVM [28]. We explore the optimal binary classifier combination, not the direct connection of all the basic classifiers. We need to take two factors, i.e., accuracy and energy consumption, into consideration simultaneously. The accuracy can be calculated by recognizing the training data using the corresponding classifier. For the energy consumption, we design a circuit-level implementation for the basic binary SVM classifier to simulate its energy consumption.

We first illustrate our circuit-level implementation of binary SVM. Because the main task of dynamic knob is to deal with the multi-class classification problem, we only need to implement the testing phase of SVM according to the formulation in (12). In addition, we need to specify a kernel function for the SVM classifier. The radial basis function (RBF) [29] is a good alternate, which has a robust performance in the usual classification problem. Its formulation is as follows:

$$K(\mathbf{x}, \mathbf{x}_i) = e^{-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2} \quad (13)$$

where  $\gamma$  is a penalty parameter,  $\mathbf{x}$  is the input feature vector and  $\mathbf{x}_i$  is the feature vector of support vectors. It is computationally complicated to calculate exponential value on a hardware platform. Here, we adopt Cordic algorithm [30] to compute the kernel function. Specifically, the Cordic method transforms the complex computation into some basic operations, such as addition, shift, and look-up table. These computations have efficient implementation in hardware design. Therefore, the circuit-level details of the binary SVM implementation is illustrated in Fig. 3.

From Fig. 3, when an input feature vector comes, we first calculate the subtraction values between two feature vectors, respectively. Then a square operation is executed by a multiplier for individual components in the feature vector. Subsequent addition operation sums all the squared values together to a scalar value and prepares for multiplication with the parameter  $\gamma$ . Then the data go through the exponential calculation module (exp) which applies the Cordic algorithm. Finally, the sum result of the multiplication between exponential value and  $paras$  can be

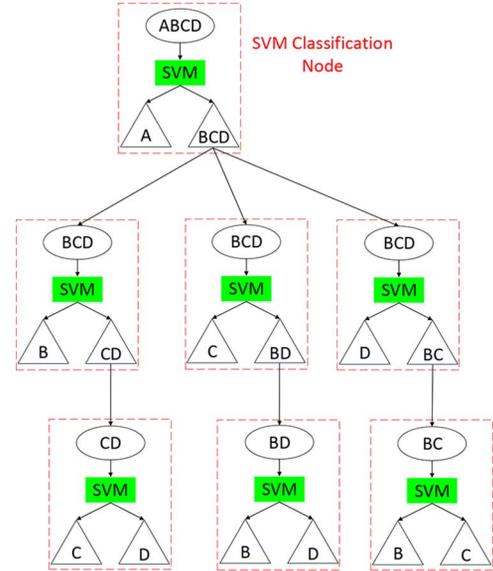


Fig. 4. SVM classification tree: tree-shaped organization of the SVM classification nodes.

taken as the indicator for final binary decision. We can pre-compute the  $paras$  as the following:

$$paras_i = y_i \times \alpha_i. \quad (14)$$

Based on the above circuit-level design, we can obtain the energy consumption of each trained binary SVM classifier. Our biggest challenge becomes how to find the optimal binary SVM cascades under the constraint of accuracy and energy consumption.

### C. Problem Formulation

We have a set of trained binary SVM classifiers, each of which aims to do a binary division for the input data categories. Also, each binary SVM has two key attributes, i.e., classification accuracy and energy consumption. The key question is how to find an optimal SVM-cascade classifier from all elementary binary SVM combined. This is a challenging combinatorial optimization problem [31], which cannot be efficiently solved by the brute force [32] or Monte Carlo [33] methods. In this work, we propose a tree-based method and formalize the problem into a solvable form within polynomial time. First, we would like to provide some basic definitions in our problem:

*Definition 1:* An SVM classification node is a binary SVM classifier with its seven attributes, *Parent\_category*, *LChild\_category*, *RChild\_category*, *LChild\_domain*, *RChild\_domain*, *Accuracy* and *Energy*.

In Definition 1, we emphasize that each SVM classification node has two child domains (left and right), and each child domain can connect to other arbitrary SVM nodes, as shown in the dashed rectangle in Fig. 4. Another emphasis is that the attribute *energy* indicates the average energy consumption for a single classification event. To simplify the original unit nJ/event, we directly use nJ as SVM node's energy unit. The detailed attribution description is illustrated in Table I.

TABLE I  
DESCRIPTION OF SEVEN ATTRIBUTES IN SVM CLASSIFICATION NODE

Attributes	Notation	Description
<i>Parent_Category</i>	$P_c$	The input category set of this classifier
<i>LChild_Category</i>	$LC_c$	The subset of <i>Parent_Category</i> when the decision of classifier is -1
<i>RChild_Category</i>	$RC_c$	The subset of <i>Parent_Category</i> when the decision of classifier is 1
<i>LChild_domain</i>	$LC_d$	The SVM node ID set that connects to <i>LChild_Category</i>
<i>RChild_domain</i>	$RC_d$	The SVM node ID set that connects to <i>RChild_Category</i>
<i>energy</i>	$nrg$	The average energy consumption to identify an input feature vector
<i>accuracy</i>	$acc$	The recognition rate of applying the trained SVM classifier to identify the training data

**Definition 2:** An SVM classification tree is a tree consisting of the SVM classification nodes, with its leaf node indivisible. The connectivity between two nodes is built only when the *Parent\_category* attribute in child node is equal to the *LChild\_category* or *RChild\_category* attribute in parent node.

According to Definition 2, we can know that an SVM classification tree divides the data set until the data set cannot be divided, and only one category is left. We provide an example of building a four-class SVM classification tree in Fig. 4. There are four classes in the figure, referred as  $A$ ,  $B$ ,  $C$ , and  $D$ . Each dashed rectangle indicates an SVM classification node. In the SVM classification node, the top ellipse represents the data needed to be classified, the *parent\_category*, and the rectangle represents the SVM classifier. After the decision of SVM classifier, we obtain two category subsets, *LChild\_category* and *RChild\_category*. For example, the top SVM node is designed to classify  $A$  category from all other three classes, thus its *parent\_category* is  $ABCD$ , *LChild\_category* is  $A$  and *RChild\_category* is  $BCD$ . Since *LChild\_category* is indivisible with only  $A$  category, we continue to proceed the classification for the right child set. Here, we have three choices of how to classify *RChild\_category* into two groups. We list all the possible binary classifiers,  $B$  and  $CD$ ,  $C$  and  $BD$ ,  $D$  and  $BC$ . We connect all these three classifiers to the right child domain of the previous top node. So the *LChild\_domain* attribute of the top node is  $\emptyset$ , and *RChild\_domain* includes all the three nodes dividing  $BCD$  set. There are still three groups with two categories needed to be divided further. We continue to train the subsequent classifiers until there is no node to be split. Therefore, we finished an SVM classification tree building.

If the purpose is to solve the four-class classification problem, there are still six other SVM classification trees to be built by this rule. For a global optimal solution, we define a super dummy root node  $S$  to connect all these SVM classification trees into its left child.

**Definition 3:** A complete SVM classification tree is a tree with a super dummy node connecting all the possible SVM classification trees for a specific multi-class classification problem.

Based on Definition 3, we can build a complete SVM classification tree, including all the possible SVM classification nodes as its tree node, as illustrated in Fig. 5. In this complete tree, we define a new delimiter “|” to simplify the symbol of our SVM classification node. This is a binary relation symbol, with the variables before the delimiter classified as left child and the elements after the delimiter identified as right child. The union of

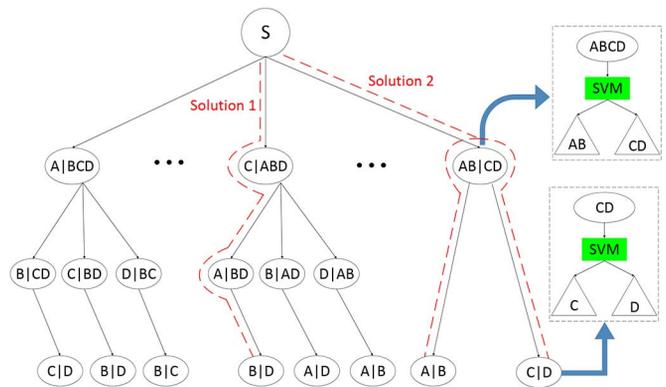


Fig. 5. The complete SVM classification tree.

both children is the entire input data set. For example, we have shown two detailed SVM classifier for the  $AB|CD$  and  $C|D$  nodes as the dashed rectangles in Fig. 5. Therefore, the problem changes to how we search for the optimal SVM cascade classifier on the complete SVM classification tree.

#### D. Problem Solution

Before we discuss the solution of this optimization problem, we first have a look at some properties of the complete SVM classification tree.

**Lemma 1:** If an SVM classification node subset from the complete SVM classification tree can form a tree, whose root is the super dummy node and leaves are unique and indivisible, this SVM node subset is a solution for the multi-class classification problem.

**Proof:** According to Definition 3 about the complete SVM classification tree, any child of the super dummy node includes all the categories for the multi-classification problem. In addition, the SVM classification node subset is with the unique and indivisible leaves of the new tree. That is to say that each leaf holds a unique category for the classification problem. Therefore, such SVM node subset forms a solution for multi-classification task.

According to Lemma 1, we can find some cascade classifier solutions for a 4-class problem as shown in Fig. 5, such as the cascade of  $C|ABD$ ,  $A|BD$ ,  $B|D$  and the combination of  $AB|CD$ ,  $A|B$ ,  $C|D$ . So the solution may be a single path from root to specific leaf. It is also possible that the multi-path combination, as the  $AB|CD$  case, which needs multiple paths to construct the solution. We can see from the above lemma that the set of classifiers, located on a single path from root node to leaf node, is the minimal structural unit to construct the solution for the multi-class classification problem.

**Lemma 2:** The complete SVM classification tree is a complete solution space for the multi-class classification problem.

**Proof:** According to Definition 3 about the complete SVM classification tree, the super dummy node connects all the possibilities of the first binary division of the original classification problem. Subsequently, based on Definition 2, the building procedure of the SVM classification tree traverses all the possible binary divisions for the current classification problem. If we iteratively build the tree according to this rule, we will cover all the

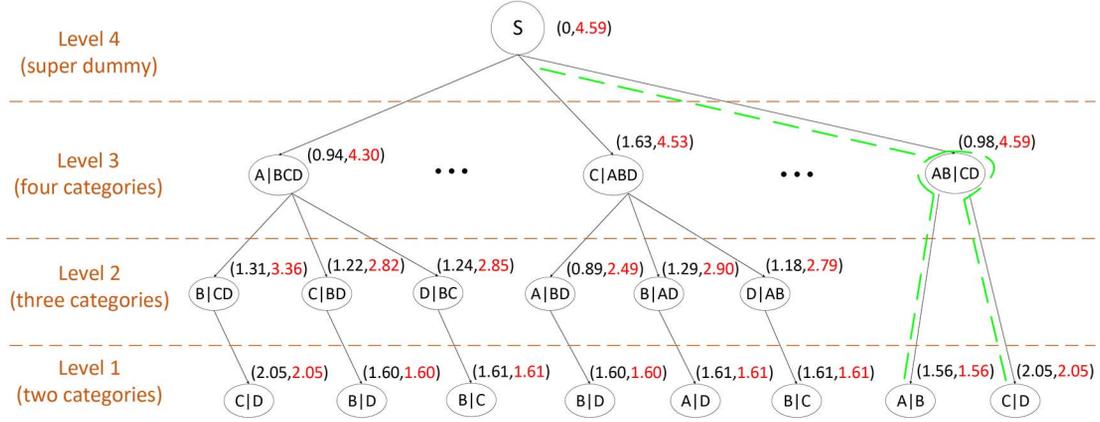


Fig. 6. An example of dynamic programming on the SVM classification tree. The former text is the  $Ad$  of a specific SVM node and the latter text indicates the intermediate result in dynamic programming.

solutions of multi-classification problem in the complete SVM classification tree.

From Lemma 2, we know that the complete SVM tree is the complete solution space of all the possible solutions of multi-class classification. In addition to Lemma 1, with paths from root to leaf node, we can conclude that the optimal cascade classifier must be a single path or multiple path combination on the complete SVM classification tree.

Our purpose is to find the optimal cascade classifier to perfectly trade off the accuracy and energy consumption. We define an accuracy density  $Ad$  as the criterion to evaluate the relationship between the accuracy and energy consumption

$$Ad = \frac{\text{accuracy}}{\text{energy}}. \quad (15)$$

From the definition of (15), we can see that  $Ad$  is actually an indicator of how much accuracy gain can be obtained by consuming one unit energy for an event, which indicates to identify a single segment by the classifier. Under this situation, big  $Ad$  means that more accuracy can be reached by consuming the same energy. Therefore, if we have the complete solution set  $Q_s$ , our objective is to maximize the  $Ad$  sum for the cascade SVM classifiers  $C_p$  from  $Q_s$ , as the following:

$$C_p = \arg \max_{C_p \in Q_s} \left( \sum_{i \in C_p} Ad_i \right). \quad (16)$$

The most straight forward method is to traverse all the possible paths from dummy node to leaf node. Then the brute force algorithm can find the maximal sum of  $Ad$  in the path set. The time complexity of this method is  $O(2^n)$ , where  $n$  is the node number of the entire complete SVM classification tree. However, if multi-class tasks aim to solve larger category number problems, the huge computation burden is intolerable.

If we examine the complete SVM classification tree, we find that there are some same sub-trees between different SVM classification trees connected to the super dummy root. Thus, we don't directly search for the specific traversing path. On the contrary, we divide the complete SVM classification tree

into several levels according to the category number of nodes'  $Parent\_category$  attribute. Then the original combinatorial optimization problem can be reformulated as the multi-stage decision problem. We define  $dp[i][j]$  as the max sum of  $Ad$  at the  $j$ -th node on the  $i$ -th level. We apply a dynamic programming algorithm [34] on this complete SVM classification tree to obtain the optimal solution, with its recursive formula as follows:

$$dp[i][j] = Ad(i, j) + \underbrace{\max_{LC\_d} dp[l_1][k_1]}_{\text{Left\_Child}} + \underbrace{\max_{RC\_d} dp[l_2][k_2]}_{\text{Right\_Child}}. \quad (17)$$

From the formulation in (17), we can know that the programming path is proceeding by the connectivity among nodes, not trying to find a specific path. Each node updates its accuracy density accumulation by the sum of the max  $Ad$  of the left child, the max  $Ad$  of the right child and the  $Ad$  in its own node. The time complexity of dynamic programming on our complete SVM tree is  $O(n^2)$ , where  $n$  is the total number of all the SVM classification nodes. After updating all the accuracy density accumulation of the SVM node, we can obtain the biggest  $Ad$ , whose programming path is the optimal SVM cascade classifier solution. Here, we provide an example of this dynamic programming on the tree. In fact, our formulation can deal with multiple tree branches. For simplicity, we take a two-branch tree in our example, as illustrated in Fig. 6.

Due to the space limitation, only a tree with a super dummy and three SVM classification trees are shown in Fig. 6. This tree is not a real complete SVM tree, but we just use it to explain our dynamic programming on the SVM tree. In Fig. 6, all the SVM classification nodes are divided into four levels according to the category number of their  $Parent\_category$  attribute. The first level has two categories and other levels are with an ascending order of classes. The hierarchical framework ends by the top super dummy root node. Each node has two values, one is the accuracy density  $Ad$  of this node, appearing to be the former text, and the other is the  $Ad$  accumulation value, shown as the latter text, which is updated by dynamic programming. We take

a bottom-up scheme to implement the dynamic programming algorithm, according to (17). Here, we would like to emphasize two detailed updating procedures. The first is to update node  $A|BCD$  on the level 3. Since this node does not have left children, we calculate the max  $Ad$  accumulation value among all its right children on level 2. Then the node  $B|CD$  is chosen for its highest  $Ad$  value 3.36, compared with other two candidates. Thus we update the node  $A|BCD$  by the sum of the max contribution of its right children and the  $Ad$  of its own node. The other procedure we need to explain is the updating of node  $AB|CD$ . As we can see, this node does not have children nodes on Level 2, but with left child and right child on Level 1. So this node can be updated by the sum from its two children and its own  $Ad$ , because the two children are both with a single child node.

After we finish dynamic programming on this SVM tree, we can find that the optimal cascade SVM classifier comes from the sub-tree with  $AB|CD$  as its root node. We can trace the updating path, and find the optimal solution, as the large-pitch dashed line shows in Fig. 6. Therefore, the cascade SVM classifier, including  $AB|CD$ ,  $A|B$  and  $C|D$ , provides the optimal solution in this SVM tree example.

## VI. EXPERIMENTS

In this section, we verify the performance of our proposed dynamic knob framework. We elaborate on the characterization of the dynamic knob itself, and then provide a comprehensive comparison between traditional CS and CS architecture with the dynamic knob on different biosignal benchmarks to demonstrate its effect on the CS framework.

### A. Experimental Setup and Dataset

We aim to demonstrate the characterization of the dynamic knob and its impact on the energy efficiency of CS architecture by the experiments. For the dynamic knob, we define the multi-class classification problem based on the biosignal with data dynamics. Then we illustrate the accuracy, area and energy consumption of the optimal cascade SVM-based ASIC design. For its impact on CS architecture, we provide a comprehensive comparison between CS architecture with the dynamic knob to reconfigure the parameters and traditional CS framework with a static setup configuration in several aspects, such as signal reconstruction quality and the configuration deviation from the optimal case.

Body sensor network is a prevalent field applying wireless sensing technology. Monitoring the challenging biosignals is very helpful for diagnosing human health conditions. These signals have strong data dynamics, because the human body is a dynamically stable entity. In our experiments, we choose EEG, ECG, and EOG signals from Physionet [9] as our test bench. The waveforms of these three signals are shown in Fig. 7. We use a continuous 120-segment signal for each signal category, with 128 samples in each segment. We take the Libsvm tool [35] in Matlab to train the basic binary SVM classifier by 10-fold cross validation scheme [36], with 60 continuous segments as the training set and the other 60 continuous segments as the testing set.

In the ASIC implementation, we use Synopsys Design Suit [37] to accomplish the dynamic knob design. We take TSMC

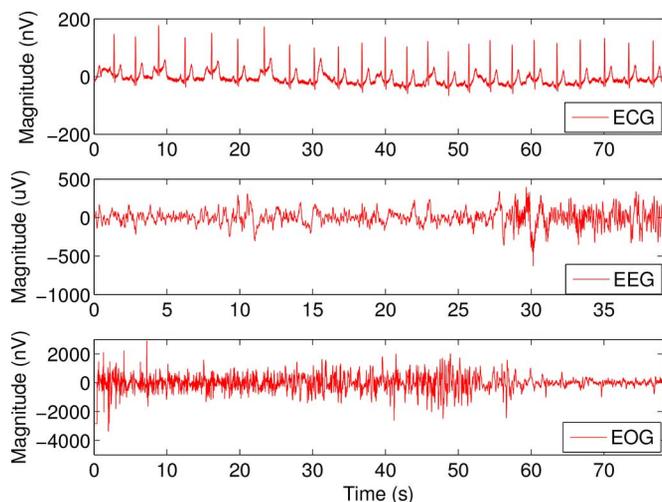


Fig. 7. The waveforms of ECG, EEG, and EOG signals.

130nm standard cell library [38]. We implement our dynamic knob design by verilog in its Verilog Compile Simulator (VCS). We also carry out logic simulation for our design and record the logic toggles, the SAIF file, in the simulation of VCS to achieve an accurate power estimation. Then we adopt Design Compiler (DC) to synthesize our verilog design and report the power consumption by the Power Compiler.

### B. Dynamic Knob Characterization

In this part, we elaborate on the characterization of our dynamic knob design. First, we provide proofs to determine the number of biosignal classification problems. Then, we illustrate the detailed attributes of our dynamic knob, such as the area and energy consumption of each component. For simplicity, we only take the EEG benchmark as an example to describe the above two procedures. The processing of other biosignals is similar to the EEG case. At last, we provide a comparison of the three dynamic knobs from different biosignal types.

1) *Classification Number Exploration*: For our template-based dynamic knob approach, we need to determine how many categories, or templates, are in the training set. The common method is to extract the feature of training data and cluster the feature vector. However, the ultimate goal of our CS architecture is to improve the performance-energy trade-off for the entire framework towards biosignal dynamics. Therefore, we adopt the Pareto's curve [39] as our input for the clustering, as it indicates the optimal relationship between performance and energy.

The Pareto's curve always consists of arbitrary number trade-off points. Directly implementing the clustering algorithm on the Pareto's curve is difficult, because there exists the misalignment between different curves. We adopt B-Spline plus K-means method [40] to cluster the Pareto's curves from the training set. We first fit each Pareto's curve into the same vector space using the B-spline method, whose parameters with the same dimension are taken as the feature for clustering. This step finishes the alignment operation of all the curves. Then the K-means algorithm is applied to cluster the extracted feature from the fitting step. Because of the randomness of

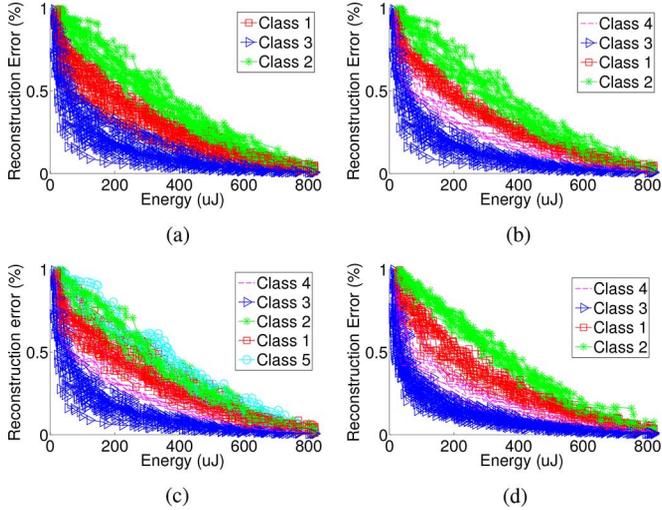


Fig. 8. Clustering results of B-spline and K-means method. (a), (b), (c) correspond to  $K = 3, 4$  and  $5$ , respectively, and (d) is the result of the testing set. (a)  $K = 3$ . (b)  $K = 4$ . (c)  $K = 5$ . (d) Test Set:  $K = 4$ .

starting point in K-means, we may use manual correction to the obvious error clustering segment. In this experiment, we set  $\text{knots} = [80 \ 80 \ 80 \ 80 \ 320 \ 480 \ 640 \ 819 \ 819 \ 819]$  and  $\text{degree} = 3$  for the B-spline fitting. We try three centroid values, 3, 4 and 5, for the K-means algorithm. Related clustering results are illustrated as Fig. 8(a)–(c).

In Fig. 8, we can see that the clustering results of  $K = 3$  and  $K = 4$  are very clear to clarify each category. Every group is close to another one, but they do not have severe overlapping. However, for the case of  $K = 5$ , its result is a failure clustering. There are different degrees of overlap among categories 1, 2, 4 and 5. Especially to category 2 and 5, they are almost blending into each other. In other words, these two clusters should be merged into a bigger cluster. In our template-based dynamic knob design, more templates mean better optimal parameter estimation accuracy. Therefore, we choose 4 as the category number in the EEG training set to characterize the dynamic knob design. Related clustering results are taken as the ground truth of the classification problem.

2) *Energy Consumption and Accuracy*: In the dynamic knob, the energy consumption of parameter look-up table depends on a specific memory technique, usually a subtle amount of energy. Thus, our optimization focus of dynamic knob is the optimal SVM cascade classifier in the signal structure analyzer module. Before we start to solve the optimization problem of the optimal cascade classifier in the dynamic knob, we carry out experiments to obtain the two significant attributes, *accuracy* and *energy*, for each SVM classification node. Based on the experiment in the Classification Number Extraction, our dynamic knob is faced with a four-class classification problem for the EEG signal.

We train the basic SVM classifiers by LibSVM tool on Matlab. We train SVMs for each possible binary classification category combination in the training set. In the experimental setup, there are 25 category combinations in total of the four-class classification. In all the SVM training phases, we take RBF as the kernel function. We also apply a grid searching

TABLE II  
CHARACTERIZATION OF BINARY SVM CLASSIFIER NODES

ID	$P\_c$	$LC\_c$	$RC\_c$	$LC\_d$	$RC\_d$	<i>acc</i> (%)	<i>nrg</i> (nJ)
1	1234	1	234	□	[17,18,19]	85	90.59
2	1234	2	134	□	[14,15,16]	86.67	62.45
3	1234	3	124	□	[11,12,13]	100	91.06
4	1234	4	123	□	[8,9,10]	100	110.89
5	1234	12	34	[20]	[25]	100	101.82
6	1234	13	24	[21]	[24]	100	121.05
7	1234	14	23	[22]	[23]	83.33	124.72
8	123	1	23	□	[23]	89.58	79.62
9	123	2	13	□	[21]	87.50	47.31
10	123	3	12	□	[20]	100	74.70
11	124	1	24	□	[24]	80.43	90.25
12	124	2	14	□	[22]	82.61	64.08
13	124	4	12	□	[20]	100	84.80
14	134	1	34	□	[25]	100	76.67
15	134	3	14	□	[22]	100	68.66
16	134	4	13	□	[21]	100	79.92
17	234	2	34	□	[25]	97.62	74.37
18	234	3	24	□	[24]	100	81.78
19	234	4	23	□	[23]	95.24	76.95
20	12	1	2	□	□	76.47	48.87
21	13	1	3	□	□	100	61.99
22	14	1	4	□	□	100	62.18
23	23	2	3	□	□	100	62.04
24	24	2	4	□	□	92.86	57.84
25	34	3	4	□	□	100	48.76

method [41] to find the optimal parameter setup for SVM, such as cost variable  $C$  and kernel function coefficient  $\gamma$ . After completing the training phase, we use each training data for binary SVM to verify the corresponding trained classifier, taking the recognition rate as the accuracy, *acc*, of the trained classifier. On the other hand, we use Synopsys design suit with 130 nm TSMC cell library to measure the average energy consumption, *nrg*, of each binary SVM classifier to complete the classification task of a specific segment. Table II lists the accuracy and energy consumption of all the SVM classification nodes in this four-class classification problem.

As shown in Table II, it is not such an obvious rule to describe the accuracy or energy consumption for all the SVM classification nodes. For accuracy, the recognition rate of two-category  $P\_c$ , whose *ID* is between 20 and 25, ranges from 76.47% to 100%, while the accuracy of four-category  $P\_c$ , with *ID* from 1 to 7, is between 83.33% and 100%. The phenomenon seems to be that SVM suffers poor performance when dealing with two-category divisions, which contradicts with the advantage of the SVM classifier. In fact, this accuracy is related to the characteristics of data distribution. For example, the classifier with 12 as its  $P\_c$  is with the lowest recognition rate. We can find from Fig. 8(b) that Category 1, with the squared marker, and Category 2, with the star marker, have a little overlapping. Compared with other clear boundaries among two-category nodes, this small overlapping confuses the SVM classifier to find the optimal hyperplane, resulting in a relatively lower accuracy.

Reducing energy consumption relates to the number of supporting vectors in the SVM classifier. In a similar way, data distribution is a significant factor to influence the supporting vector number. For example, the node whose  $P\_c$  is 1234,  $LC\_c$  is 2, and  $RC\_c$  is 134, consumes little energy, 62.45 nJ. In contrast, the node whose  $P\_c$  is 1234,  $LC\_c$  is 14, and  $RC\_c$  is 23, consumes 124.72 nJ, more than two times than the previous node.

TABLE III  
NODES CHARACTERIZATION IN THE OPTIMAL CASCADE SOLUTION

ID	$P_c$	$LC_c$	$RC_c$	acc (%)	nrg (nJ)	Ad (%/nJ)
2	1234	2	134	86.67	62.45	<b>1.39</b>
14	134	1	34	100	76.67	<b>1.30</b>
25	34	3	4	100	48.76	<b>2.05</b>
Total	—	—	—	—	187.88	<b>4.74</b>

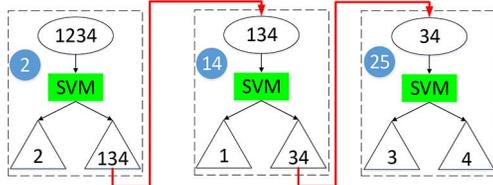


Fig. 9. The structure of the optimal cascade SVM classifier. It consists of three SVM classification nodes, NO. 2, 14, and 25.

Therefore, it is very difficult to directly obtain the optimal cascade SVM classifier, while considering two orderless attributes, accuracy and energy consumption, simultaneously.

3) *Characterization of Signal Structure Analyzer*: In this part, we first solve the optimization problem and obtain the optimal cascade classifier solution for the dynamic knob. We also show the post-layout and corresponding characterization of the optimal cascade classifier on ASIC. Then we validate the performance of the cascade classifier on all three biosignal testing sets.

After obtaining the accuracy and energy attributes, we can build seven SVM classification trees for the 4-class classification problem, as  $ID$  from 1 to 7 as the root node shown in Table II. Connecting all these SVM trees into the super dummy node, we can build a complete SVM classification tree for the optimal solution. We execute dynamic programming on the complete SVM classification tree, and the programming path corresponding to the largest  $Ad$  sum value is the optimal cascade classifier combination. The final optimal solution includes SVM nodes 2, 14 and 25, with the highest  $Ad$  sum value 4.74 %/nJ. Related node characteristics and their relationship are illustrated in Table III and Fig. 9, respectively.

From Fig. 9, we can see that the optimal classifier is a sequential cascade SVM for this EEG application. The input feature vector first goes through the NO. 2 SVM node. If the decision is Category 2, the recognition is terminated, labeling the input signal as Category 2. For the contrasting decision, the input feature vector continues to be verified by NO. 14 SVM node, as the arrow line shows. If the result is Category 1, the classification task is over. Otherwise, the input feature needs to be identified by the last SVM node, NO. 25. Based on its decision, the input feature can be recognized as Category 3 or Category 4. If the analyzing procedure involves all the three SVM nodes, the total energy consumption is 187.88 nJ, as shown in Table III.

Next, we implement the above design on ASIC by Synopsys IC Compiler with a physical library from the 130nm TSMC standard cell library, whose post-layout is illustrated as Fig. 10. The entire layout consists of four key parts, NO. 2 SVM node, NO. 14 SVM node, NO. 25 SVM node and the control logic to order these three SVM classifier nodes. The SVM nodes are

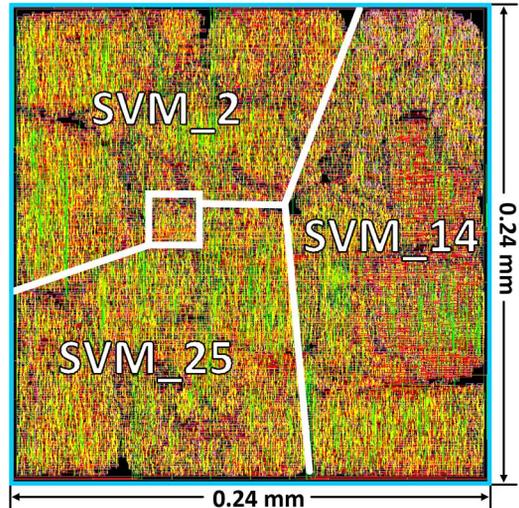


Fig. 10. Layout of the optimal cascade classifier on ASIC design.

TABLE IV  
CHARACTERIZATION OF DYNAMIC KNOB POST-LAYOUT ON ASIC DESIGN

Resource or Module	Area in $\mu m^2$	(%)	Power in $\mu W$	(%)
Knob	57732.25		300.1	
Registers	27239.75	(47.18%)	158.7	(52.88%)
Combinational	30492.5	(52.82%)	141.4	(47.12%)
SVM (2)	19643.0	(34.02%)	105.0	(34.99%)
SVM (14)	18118.0	(31.38%)	93.0	(30.99%)
SVM (25)	19449.0	(33.69%)	99.7	(33.22%)
Other logic	522.25	(0.91%)	2.4	(0.80%)

labeled in their own layout areas, and the squared rectangle reveals the combinational logic to organize this dynamic knob. We also show the characterization of our post-layout in Table IV. We detail the area and energy consumption utilization from the resource type and component levels. We can see that three SVM nodes are sharing the similar area and energy usage due to their similar implementation only with the support vector difference. The control logic to keep the dynamic knob in perfect order takes up minor energy consumption, 0.8%, and area, 0.91%. It is noted that this small resource occupation is the reason why we use the energy estimation from logic simulation to find the optimal solution, neglecting the cost from the control logic part.

4) *Dynamic Knob Comparison of Different Biosignals*: For a comprehensive verification of biosignal sensing, we take all *three* type of signals, EEG, ECG and EOG, to apply our dynamic programming based design method. We take the same condition as the EEG case for ECG and EOG signals. Here, EEG and EOG are facing the 4-class classification problem, while ECG needs to deal with the 3-class problem. We first calculate the optimal design for each signal type via their 60-segment training set. Then we use the implementation of the optimal design on ASIC to recognize the new testing segments. We obtain the area, energy consumption and testing set accuracy of three dynamic knobs, which are shown in Fig. 11.

First, let us have a look at the accuracy comparison. The recognition rate of three knobs are 83.3%, 88.3%, and 93.3%, respectively. These results demonstrate the high performance of our trained cascade classifier, whose solid capability can

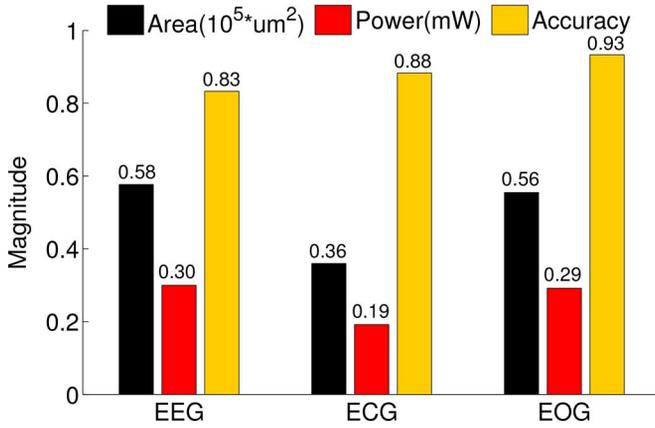


Fig. 11. Characterization of the optimal dynamic knob design of three biosignals and corresponding accuracy in testing set.

improve the accuracy for parameter configuration in wireless sensing. Then, if we take a closer look at the resource utilization, we find that the energy consumption and area of EEG and EOG are similar. These two cases both deal with the 4-class problem, which needs three classifiers. For the ECG case, its energy and area are obviously smaller than EEG and EOG, because ECG completes its 3-class classification task only by two classifiers. The waveform of ECG is more periodical than EEG and EOG, so that EEG and EOG show stronger data dynamics than ECG. Therefore, it is demonstrated that our optimal dynamic knob design can achieve high accuracy over all challenging biosignals in the experiment under low power conditions.

### C. CS With Dynamic Knob versus Traditional CS

After characterizing the dynamic knob, we check into the performance improvement when CS architecture equips with our dynamic knob. The dynamic knob can affect the reconstruction error and configuration energy of the CS architecture for its dynamic reconfiguration. In this part, we investigate the impact on performance and energy from the dynamic knob. We make comprehensive comparisons in several aspects, such as architecture performance and energy deviation.

In this experiment, we simulate the CS architecture by Matlab on PC with a 3.4 GHz 8-core processor and 8 GB RAM. The dynamic knobs solved in the characterization section are taken, which are simulated on ASIC design. For the CS architecture, we take inverse discrete wavelet transform (IDWT) as its sparsity-inducing transformation basis,  $\Psi$ , [42]. All of our experiments use Bernoulli random variables [16] as sensing array and use a uniform quantization strategy. We also take the other 60 biosignal segments as our testing set. For the traditional CS architecture, we set its bit resolution as 16 [43]–[45] for a better reconstruction and range the sampling rate  $M$  from 10 to 128.

For practical wireless sensing applications, we adopt an IPv6-based communication model over Bluetooth Low Energy (BLE) on real device [46] to model energy consumption of wireless data transmission. The throughput of three biosignals is 0.5 kBytes/s, and the 128-length segment can be transmitted

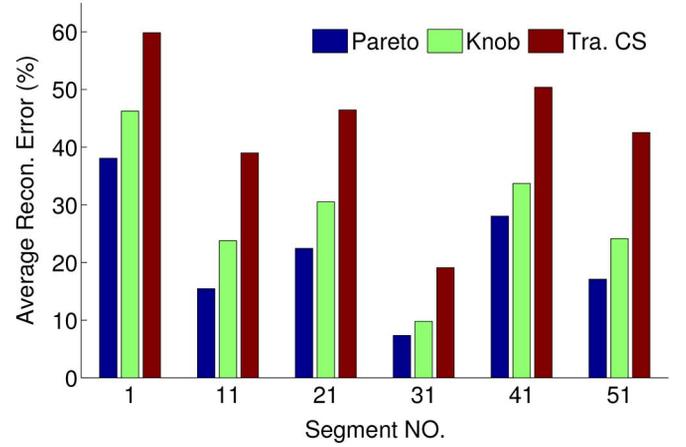


Fig. 12. Average reconstruction Error of some EEG segments under three cases: 1) the traditional CS, 2) the CS with dynamic knob, and 3) the optimal case.

in one packet. Thus, according to the experiment of connection mode of BLE, the energy consumption of this setup is 325 kBytes/J, that is,  $C = 0.4$  uJ/bit in the energy model.

1) *Reconstruction Error Comparison*: We carry out this experiment to check the reconstruction accuracy improvement by our proposed dynamic knob. We calculate the reconstruction errors of both dynamic knob case and traditional CS case with energy bound between 64 and 819. To the traditional CS architecture, we first brute force the Pareto's curve under its constant 16-bit quantization setup. Then, the responsive CS architecture with dynamic knob analyzes the input signal waveform and outputs the optimal parameter configuration estimation, sampling rate  $M$  and bit resolution  $b$ . We adopt all the biosignals, EEG, ECG and EOG, with the 60 segments in each testing set to go through the two architectures. We also calculate the error from the Pareto's curve considering both parameters together, which is the optimal solution in the entire performance-energy space.

We evaluate the entire reconstruction quality improvement based on all the 60 testing segments. For simplicity of data representation, we define an average reconstruction error ARE to indicate the average reconstruction error on all the energy bounds for a specific testing segment, as follows:

$$\text{ARE}_i = \frac{1}{|\text{EB}|} \sum_{j \in \text{EB}} P(i, j) \quad (18)$$

where EB is the energy bound set,  $|\text{EB}|$  is the total number of the bounds and  $P(i, j)$  refers to the reconstruction error under energy bound  $j$  of segment  $i$ . Based on (18), we can calculate all the ARE values of all the biosignals. The related results of the EEG testing segments are presented in Fig. 12.

As shown in Fig. 12, there are *six* groups of average reconstruction errors, with traditional CS, dynamic knob and the optimal case from the brute force algorithm. In each group, the traditional CS suffers from the largest average reconstruction error, and the Pareto's case is with the least distortion rate, which is also the lower bound of the reconstruction. Our dynamic knob is between these two cases, approximating the reconstruction error

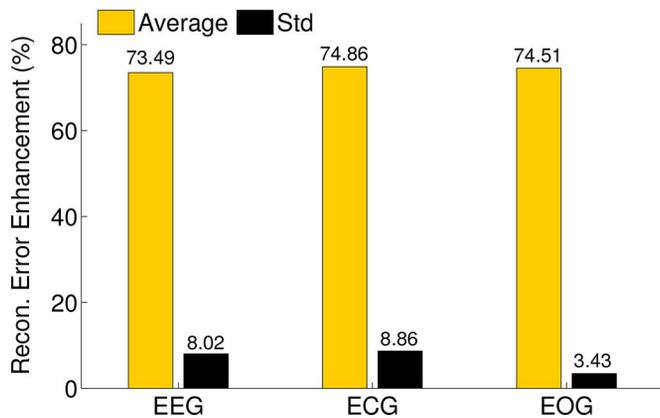


Fig. 13. Reconstruction error enhancement over all biosignal segments with the traditional CS, the CS with dynamic knob and the optimal case.

to the optimal case. From the Pareto's curve, the average reconstruction error of EEG signal fluctuates from less than 10% to almost 40% as the segment number changes. This also demonstrates the data dynamics of the EEG signal. To adjust to this situation, our knob provides a reconfiguration strategy for different segments. From the results of comparison, our dynamic knob can promote more than a half improvement towards the optimal solution than the traditional CS case, which only has a startup configuration.

To quantitatively evaluate the improvement by our dynamic knob design, we define a relative enhancement indicator  $RE_{\text{enhance}}$ , whose form is as the following:

$$RE_{\text{enhance}} = \frac{ARE_{TCS} - ARE_{DK}}{ARE_{TCS} - ARE_{OPT}} \quad (19)$$

where  $ARE_{TCS}$  is the average reconstruction error of traditional CS,  $ARE_{TCS}$  and  $ARE_{OPT}$  refer to the ARE from dynamic knob case and the optimal case. Because the average reconstruction error has its lower bound as the optimal case, the improvement should be a ratio by the distance in  $ARE_{TCS}$  and  $ARE_{OPT}$  as the denominator, which is the maximum value in theory, and the improvement contributed by our dynamic knob,  $ARE_{TCS} - ARE_{DK}$ , as the numerator. Once we have this metric, we can calculate the average and standard deviation of  $RE_{\text{enhance}}$  for each biosignal over all their segments. The related results are shown in Fig. 13.

We can find from Fig. 13 that the three signals are with the average  $RE_{\text{enhance}}$  as 73.49%, 74.86% and 74.51%, respectively. This demonstrates that our dynamic knob can greatly improve the traditional CS to the optimal solution by more than 70%. Also, if we examine the standard deviation, 8.02%, 8.86% and 3.43%, all the fluctuations are less than 10%, which indicates that our dynamic knob has a very stable performance for the signals with data dynamics. Therefore, this experiment indicates that our dynamic knob framework is a critical module in the CS architecture towards biosignal dynamics, with more than 70% reconstructed signal fidelity promotion.

2) *Energy Configuration Deviation Comparison*: We carry out this experiment to check the energy consumption by wireless communication in the CS architecture. Different from the

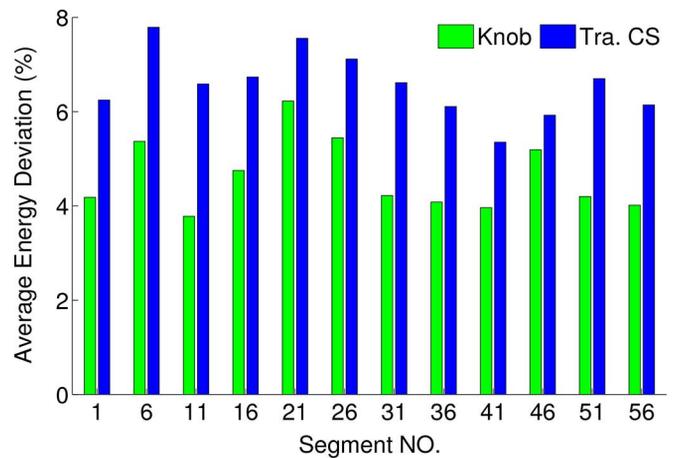


Fig. 14. Average energy configuration deviation of some EEG segments under traditional CS and the CS with dynamic knob.

analysis of reconstruction error, which has the best value in its optimal case, the energy consumption analysis focuses on the deviation between current energy configuration and the optimal energy configuration. We have the Pareto's curve, retrieved by the brute force algorithm, as our best performance-energy trade-off. Thus, getting closer to the optimal energy configuration means a larger chance to approximate to the optimal trade-off. We define the variable  $DevE$  to describe this significant deviation as follows:

$$DevE = \frac{|E_c - E_{\text{opt}}|}{E_{\text{opt}}} \quad (20)$$

where  $E_{\text{opt}}$  stands for the optimal configuration and  $E_c$  is the current energy configuration. Consequently, we continue to define average energy deviation AED to describe the deviation degree for a specific testing segment over all the energy bounds

$$AED = \frac{1}{|EB|} \sum_{j \in EB} DevE(j). \quad (21)$$

According to the definition of AED, we can calculate the average energy deviation over all energy bounds. In this experiment, we use all the biosignals with 60 segments as the testing data and set all the energy bounds between 64 and 819. We calculate the AED of CS with dynamic knob architecture and traditional CS. We also take some EEG segments to check the energy deviation of traditional CS and CS with dynamic knob, as shown in Fig. 14.

We can find from Fig. 14 that the CS with dynamic knob has less energy configuration deviation than the traditional CS architecture, because traditional CS uses a one-pass configuration strategy, which is rigid to data dynamics. However, our dynamic knob can adjust the configuration parameter towards the optimal case. Thus, it has less energy configuration deviation, which means our dynamic knob approximates to the optimal solution further than traditional CS, reflecting the flexibility of our dynamic knob to deal with biosignal dynamics. Consequently, we

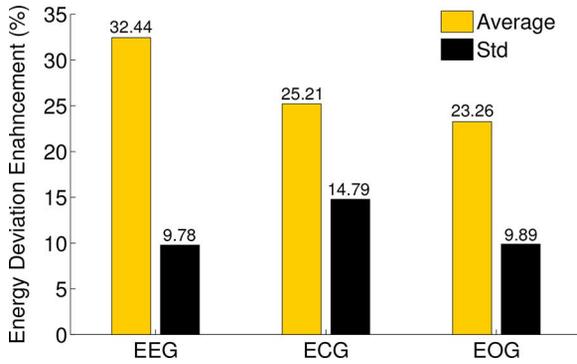


Fig. 15. Energy deviation enhancement over all biosignal segments under traditional CS and the CS with dynamic knob.

define  $E_{d_{\text{enhance}}}$  as the energy deviation enhancement between our dynamic knob and the traditional CS architecture.

$$E_{d_{\text{enhance}}} = \frac{AED_{\text{TCS}} - AED_{\text{DK}}}{AED_{\text{TCS}}} \quad (22)$$

We can calculate the deviation enhancement  $E_{d_{\text{enhance}}}$  of all testing segments for each type of biosignals, i.e., EEG, ECG and EOG. The average and standard deviation of this enhancement are illustrated in Fig. 15.

In Fig. 15, the average energy deviation enhancements of all three signals are similar, 32.44%, 25.21% and 23.26%. For the standard deviation, ECG shows a larger fluctuation trend, 14.79%. For the entire template space, EEG and EOG divide it into four areas, while ECG can only use three levels of the parameter setting. So the ECG case is with larger energy deviation enhancement fluctuations. This demonstrates that our dynamic knob can accurately configure the architecture parameters to ensure a larger chance that the architecture will gain a better performance-energy trade-off over several representative biosignals.

## VII. CONCLUSION AND FUTURE WORK

In this work, we introduced a dynamic knob design for wireless sensing architecture towards biosignal dynamics. We first described the traditional CS architecture to solve the energy consumption bottleneck of the sensor node design. Then we proposed a dynamic knob framework to dynamically control other components to achieve a better energy efficiency. We explained our structure characters of dynamic knob for the designing concerns of wireless sensing. Moreover, we discussed the implementation of our dynamic knob based on a supervised learning algorithm on ASIC, and formulated a tree-based optimization problem to search for the optimal dynamic knob design. Eventually, the optimization problem was solved by dynamic programming. We also verified our proposed dynamic knob design by experiments on continuous biosignals. We analyzed the characterization of the dynamic knob and discussed its impact on CS architecture. We demonstrated that our dynamic knob provides a basis towards biosignal dynamics.

In the future work, we will consider integrating the specific ADC optimization scheme, such as dynamic range in ADC [47], to further improve the energy efficiency. Another direction is to

apply other design criterion to deeply optimize the signal reconstruction quality towards biosignal dynamics, e.g., rakesness [48].

## REFERENCES

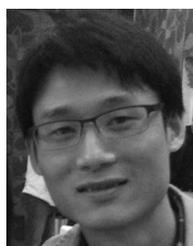
- [1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [2] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [3] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vanderghenst, "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, 2011.
- [4] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing of eeg for wireless telemonitoring with low energy consumption and inexpensive hardware," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 221–224, 2013.
- [5] M. Shorran, M. H. Kamal, C. Pollo, P. Vanderghenst, and A. Schmid, "Compact low-power cortical recording architecture for compressive multichannel data acquisition," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 6, pp. 857–870, Dec. 2014.
- [6] P. K. Baheti and H. Garudadri, "An ultra low power pulse oximeter sensor based on compressed sensing," in *Proc. 6th IEEE Int. Workshop Wearable and Implantable Body Sensor Networks*, 2009, pp. 144–148.
- [7] T. Feng, K. Aono, T. Covassin, and S. Chakrabarty, "Self-powered monitoring of repeated head impacts using time-dilation energy measurement circuit," vol. 9, no. 2, pp. 217–226, Apr. 2015.
- [8] N. Lajnef, N. G. Elvin, and S. Chakrabarty, "A piezo-powered floating-gate sensor array for long-term fatigue monitoring in biomechanical implants," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 3, pp. 164–172, 2008.
- [9] G. B. Moody, R. G. Mark, and A. L. Goldberger, "Physionet: A web-based resource for the study of physiologic signals," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 70–75, 2001.
- [10] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.
- [11] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [12] M. L. Malloy and R. D. Nowak, "Near-optimal adaptive compressed sensing," in *Proc. 46th IEEE Asilomar Conf. Signals, Systems and Computers, Conf. Rec.*, 2012, pp. 1935–1939.
- [13] X. Wang, W. Guo, Y. Lu, and W. Wang, "Adaptive compressive sampling for wideband signals," in *Proc. IEEE 73rd Vehicular Technology Conf.*, Spring, 2011, pp. 1–5.
- [14] X. Wang, T. Shi, D. Lei, and W. Guo, "Adaptive measurement adjustment for sparse streaming signal," in *Proc. Int. Conf. Information and Communications Technologies*, 2014, pp. 1–7.
- [15] R. Zahedi, L. W. Krakow, E. K. Chong, and A. Pezeshki, "Adaptive compressive measurement design using approximate dynamic programming," in *Proc. IEEE Amer. Control Conf.*, 2013, pp. 2442–2447.
- [16] N. R. Chaganty and H. Joe, "Range of correlation matrices for dependent bernoulli random variables," *Biometrika*, vol. 93, no. 1, pp. 197–206, 2006.
- [17] M. Hyder and K. Mahata, "An approximate l0 norm minimization algorithm for compressed sensing," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2009, pp. 3365–3368.
- [18] A. Zymnis, S. Boyd, and E. Candes, "Compressed sensing with quantized measurements," *IEEE Signal Process. Lett.*, vol. 17, no. 2, pp. 149–152, 2010.
- [19] A. Wang, W. Xu, Z. Jin, and F. Gong, "Quantization effects in an analog-to-information front end in eeg telemonitoring," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 2, pp. 104–108, 2015.
- [20] A. Wang, C. Song, and W. Xu, "A configurable quantized compressed sensing architecture for low-power tele-monitoring," in *Proc. IEEE Int. Green Computing Conf.*, 2014, pp. 1–10.
- [21] A. Wang, Z. Jin, C. Song, and W. Xu, "Adaptive compressed sensing architecture in wireless brain-computer interface," in *Proc. 52nd Annu. Design Automation Conf.*, 2015, p. 173.
- [22] S. Yuan, H. Huang, Q. Liang, and Q. Li, "Energy efficient comparator for successive approximation register adcs with application to wireless sensor networks," *Int. J. Sensor Netw.*, vol. 17, no. 2, pp. 122–129, 2015.
- [23] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

- [24] J. C. Platt *et al.*, "Using analytic qp and sparseness to speed training of support vector machines," *Adv. Neural Inf. Process. Sys.*, pp. 557–563, 1999.
- [25] K.-B. Duan and S. S. Keerthi, "Which is the best multiclass svm method? an empirical study," in *Multiple Classifier Systems*. New York, NY, USA: Springer, 2005, pp. 278–285.
- [26] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, 2002.
- [27] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Nips*, 1999, vol. 12, pp. 547–553.
- [28] B. Fei and J. Liu, "Binary tree of SVM: A new fast multiclass training and classification algorithm," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 696–704, 2006.
- [29] R. J. Howlett and L. C. Jain, *Radial Basis Function Networks 2: New Advances in Design*. Physica, 2013, vol. 67.
- [30] P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of cordic: Algorithms, architectures, and applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 1893–1907, 2009.
- [31] C. Blum and G. Ochoa, in *Proc. 14th Eur. Conf. Evolutionary Computation in Combinatorial Optimization*, Granada, Spain, 2014, vol. 8600, Apr, 23–25, 2014, Revised Selected Papers, Springer.
- [32] A. M. A. Ibrahim and M. E. Mustafa, "Comparison criteria between matching algorithms texts application on (Horspool's and brute force algorithms)," *J. Adv. Comput. Sci. Technol.*, vol. 4, no. 1, pp. 175–179, 2015.
- [33] Y. A. Shreider, *The Monte Carlo Method: The Method of Statistical Trials*. New York, NY, USA: Elsevier, 2014, vol. 87.
- [34] E. V. Denardo, *Dynamic Programming: Models and Applications*. Mineola, NY, USA: Courier Dover, 2003.
- [35] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.
- [36] T. Fushiki, "Estimation of prediction error by using k-fold cross-validation," *Stat. Comput.*, vol. 21, no. 2, pp. 137–146, 2011.
- [37] H. Bhatnagar, *Advanced ASIC Chip Synthesis: Using Synopsys® Design Compiler (TM) Physical Compiler (TM) and PrimeTime®*. New York, NY, USA: Springer Science & Business Media, 2007.
- [38] M. Sanie, M. Côté, P. Hurat, and V. Malhotra, "Practical application of full-feature alternating phase-shifting technology for a phase-aware standard-cell design flow," in *Proc. IEEE Design Automation Conf.*, 2001, pp. 93–96.
- [39] I. Kacem, S. Hammadi, and P. Borne, "Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic," *Math. Comput. Simul.*, vol. 60, no. 3, pp. 245–276, 2002.
- [40] C. Abraham, P.-A. Cornillon, E. Matzner-Løber, and N. Molinari, "Unsupervised curve clustering using b-splines," *Scandinavian J. Stat.*, vol. 30, no. 3, pp. 581–595, 2003.
- [41] A. Janz, S. van der Linden, B. Waske, and P. Hostert, "images-vma user-oriented tool for advanced classification of hyperspectral data using support vector machines," in *Proc. EARSeL SIG Imaging Spectroscopy*, Bruges, Belgium, 2007.
- [42] H. Ocak, "Automatic detection of epileptic seizures in eeg using discrete wavelet transform and approximate entropy," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 2027–2036, 2009.
- [43] H. Kim, Y. Kim, and H.-J. Yoo, "A low cost quadratic level ECG compression algorithm and its hardware optimization for body sensor network system," in *Proc. 30th Annu. Int. Conf. IEEE Engineering in Medicine and Biology Soc.*, 2008, pp. 5490–5493.
- [44] J. Altenburg, R. J. Vermeulen, R. L. Strijers, W. P. Fetter, and C. J. Stam, "Seizure detection in the neonatal eeg with synchronization likelihood," *Clin. Neurophys.*, vol. 114, no. 1, pp. 50–55, 2003.
- [45] A. T. Vehkaoja *et al.*, "Wireless head cap for eeg and facial ERG measurements," in *Proc. IEEE 27th Annu. Int. Conf. Engineering in Medicine and Biology Soc.*, 2006, pp. 5865–5868.
- [46] M. Siekkinen, M. Hienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4," in *Proc. IEEE Wireless Communications and Networking Conf. Workshops*, 2012, pp. 232–237.
- [47] B. Malki, T. Yamamoto, B. Verbruggen, P. Wambacq, and J. Craninckx, "A 70 db dr 10 b 0-to-80 ms/s current-integrating sar adc with adaptive dynamic range," *IEEE J. Solid-State Circuits*, vol. 49, no. 5, pp. 1173–1183, 2014.
- [48] M. Mangia, R. Rovatti, and G. Setti, "Rakeness in the design of analog-to-information conversion of sparse and localized signals," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 5, pp. 1001–1014, 2012.



**Aosen Wang** (S'15) received the B.S. degree in electrical engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2011.

After earning the B.S. degree, he joined Vimicro as an Algorithm and Software Engineer. Currently, he is working toward the Ph.D. degree in computer science and engineering at the State University of New York (SUNY) at Buffalo, Buffalo, NY, USA. His research interests include low-power computer architecture and energy-efficient machine learning.



**Feng Lin** (S'11–M'15) received the Ph.D. degree from the Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN, USA, in 2015.

Currently, he is a Research Scientist in the Department of Computer Science and Engineering, State University of New York (SUNY) at Buffalo, Buffalo, NY, USA. His research interests include biometrics, human-computer interface, wireless communications, and their applications in wireless health and body sensor networks.



**Zhanpeng Jin** (S'07–M'10) received the Ph.D. degree in electrical engineering from the University of Pittsburgh, Pittsburgh, PA, USA, in 2010.

Currently, he is an Assistant Professor in the Electrical and Computer Engineering Department and Biomedical Engineering Department as well as the Director of the Cyber-Med Laboratory at Binghamton University, State University of New York (SUNY), Binghamton, NY, USA. He was a Postdoctoral Research Associate at the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2011. He was a Visiting Faculty Fellow in the Air Force Research Laboratory.

His research interests include mobile and wearable computing in smart and connected health, cognitive biometrics, neuromorphic computing, and ultra-low-power sensing.

Dr. Jin. is on the journal editorial boards of *Computers in Biology and Medicine*, *Informatics in Medicine Unlocked*, *Computers and Electrical Engineering*, and *International Journal of Monitoring and Surveillance Technologies*. He is a member of the Sigma Xi, IEEE Computer Society, and IEEE Engineering in Medicine and Biology Society.



**Wenyao Xu** (M'13) received the Ph.D. degree from the Electrical Engineering Department, University of California, Los Angeles, Los Angeles, CA, USA, in 2013.

Currently, he is an Assistant Professor in the Computer Science and Engineering Department, State University of New York (SUNY) at Buffalo, Buffalo, NY, USA. His research foci include embedded systems, computer architecture, wireless health, low-power technologies, and their applications in biomedicine, healthcare and security. He owned

five licensed U.S. and international patents, and has authored more than 70 peer-reviewed journal and conference papers.

Dr. Xu received the Best Paper Award of the IEEE Conference on Implantable and Wearable Body Sensor Networks in 2013, and the Best Demonstration Award of ACM Wireless Health Conference in 2011.