REDUCING QUANTIZATION ERROR IN LOW-ENERGY FIR FILTER ACCELERATORS

Zhuo Wang *Jintao Zhang* *Naveen Verma*

Department of Electrical Engineering, Princeton University, Princeton, NJ, USA 08544

ABSTRACT

Computational energy versus computational precision represents a critical implementation-level tradeoff facing embedded DSP systems. Focusing on multiply-accumulate (MAC) hardware, which is used extensively in DSP implementations (e.g., FIR filtering), this paper proposes an approach that exploits floating-point representation of multipliers to enable optimization of their quantization error. The approach introduces a parameter α for coefficient scaling, and optimizes α to minimize the output error. Applied to FIR filters with coefficient representation of 6 bits, the approach reduces the quantization error by 37×, compared to traditional, linear-quantized fixed-point coefficient representation and by 28×, compared to unoptimized floating-point coefficient representation. Further, the energy and hardware gate-count of a MAC unit is reduced by 1.4× and 1.2×, respectively, compared to an implementation based on fixed-point representation.

Index Terms— embedded systems, digital filter, floating point, quantization error, low energy

1. INTRODUCTION

The emergence of advanced sensing technologies has enabled acquisition of a wide range of physical signals. Combining embedded sensing with local signal analysis capabilities is leading to high-value systems in a range of application domains, including medical, environmental, industrial, etc. [1, 2, 3, 4]. However, energy constraints play a critical role in embedded sensing systems. This leads to tradeoffs, whereby the functions and/or computational precision through which local analysis can be performed is limited. Such limitations are of particular concern with advanced algorithms, which typically require specific operations performed with specific level of precision [2].

This paper focuses on enabling a high-level of precision for FIR filtering, which is among the most widely performed operations in embedded sensing systems. Given its prominence, FIR filter accelerators are often employed [5, 6]. Their design is driven by energy-precision tradeoffs, in the context of the entire system. Namely, for ultra-low-energy sensors, fixed-point representation (and corresponding computation) is typically used, to avoid energy overheads at the low dynamic range usually required. This paper proposes the use of mixed fixed-point and floating-point representation for an FIR accelerator at the low-dynamic range levels (∼6 bits). A common approach explored for achieving low-energy digital filters is through circuit-level knobs, such as reducing supply voltage [7, 8], which can lead to low throughput and reliability issues. Here we show how energy-versus-precision trade-offs in digital filters can be substantially improved by judicious digital representation and corresponding optimization of the coefficients. The proposed implementation leads to increased precision through two mechanisms: (1) non-linear quantization, favoring low-valued multipliers; and (2) explicit optimization of the filter co-efficients to minimize the quantization error.

Support is provided by SRC, NSF (CCF-1253670), as well as Center for Future Architectures Research (C-FAR) and Systems on Nanoscale Information fabriCs (SONIC), two of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

Together these enable precision at the level of >10-b computation, using 6-b hardware, having complexity simpler than a typical fixed-point multiplier.

2. APPROACH

To present the proposed approach, we consider, as a representative application, EEG-based seizure detection. Feature extraction corresponds to spectral energy computation from each EEG channel over a one-second epoch, with three epochs then combined for classification [9]. The corresponding system is shown in Fig. 1, consisting of front-end decimation filtering, followed by band-pass filtering (via a bank of FIR filters) and output-sample accumulation. We note that the input data stream, provided by an ADC, has fixed-point representation, as is typical. In the following subsections, we first analyze the quantization error of the filter coefficients, and then describe the benefits of mixed fixed-/floating-point multiplication, proposing a corresponding implementation. Then, we describe how the implementation enables an optimization that substantially reduces the effects of quantization error.

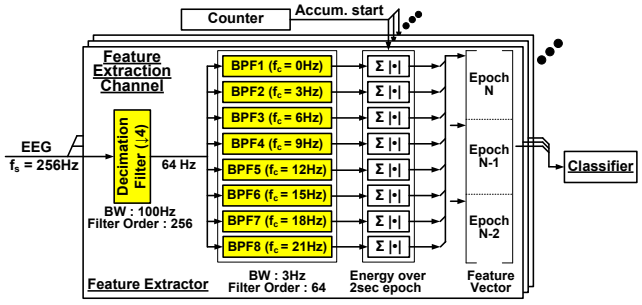

Fig. 1. Example system, corresponding to an EEG-based seizure detector, employing FIR filtering for a decimation filter and eight band-pass filters (BPF1-8).

2.1. Filter Coefficient Quantization Error

The familiar FIR filtering computation can be represented as in Eq. 1 (where y represents an output sample, h_i represents i^{th} filter coefficient, x_i represents samples of the input signal, and N represents the filter order):

$$y = \sum_{i=0}^{N-1} h_i \cdot x_i. \quad (1)$$

Typically computed via multiply-accumulate (MAC) hardware, Fig. 2 shows the effect of linear coefficient quantization at the 6-b level: (a) and (d) show the coefficients for the decimation filter and the last band-pass filter, respectively; (b) and (e) show histograms of the coefficients with bins corresponding to the quantization levels; and, (c) and (f) show histograms of the corresponding quantization error, normalized to the least-significant bit. As seen, the coefficients fall in a small number of bins, implying inefficient quantization, and exhibit substantial quantization error. Further, the dominant bins correspond to small-valued coefficients, implying large percentage error.

We point out that the characteristics observed for the last band-pass filter are typical across the filter bank, since all filters have coefficients determined by the same envelop but modulated by a sinusoid at the corresponding center frequency.

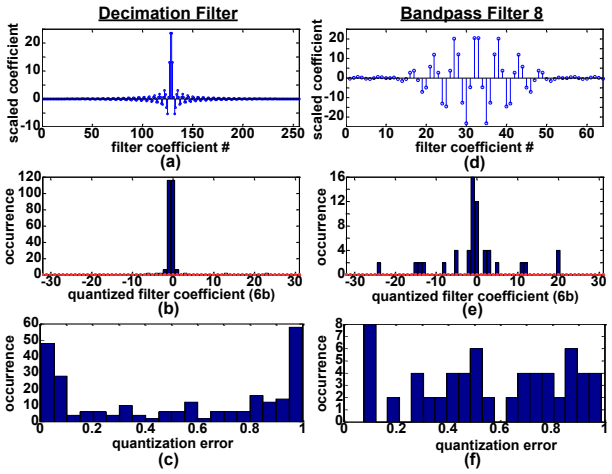

Fig. 2. Traditional fixed-point linear quantization of filter coefficients to the 6-b level for (a)-(c) decimation filter and (d)-(f) last band-pass filter used for feature extraction in the seizure detector.

2.2. Mixed Fixed-point/Floating-point Multiplication

The issue above occurs for two reasons: (1) large-valued multipliers, which tend to occur infrequently, raise the need for a large linear dynamic range; (2) linear quantization has particularly severe impact on the percentage error of small-valued multipliers, which tend to occur with high frequency in many applications of interest (e.g., that considered). To address this we leverage a floating-point representation for the multiplier h_i, while retaining fixed-point representation for the multiplicand:

$$h_i = l_i \times (1 + m_i) \times 2^{s_i}; \quad (2)$$

$$l_i = sign(h_i); \quad (3)$$

$$s_i = \lfloor log_2 |h_i| \rfloor ; \quad (4)$$

$$m_i = \frac{|h_i|}{2^{\lfloor log_2 |h_i| \rfloor}} - 1. \quad (5)$$

As defined in Eq. 3-5: l_i represents the sign of the multiplier h_i; s_i represents the number of bits that $|h_i|$ must be shifted by to give a number in the range $[1, 2)$; and $1 + m_i$ represents the resulting number within that range. Among the parameters l_i, m_i, and s_i, only m_i can take on continuous values. Thus, values of m_i in the range $[0, 1)$ are quantized. Once again limiting ourselves to a 6-b representation, Fig. 3 shows the effect of coefficient quantization for the decimation filter (with l_i, m_i, and s_i allocated 1, 1, 4 bits, respectively) and the last band-pass filter (with l_i, m_i, and s_i allocated 1, 2, 3 bits, respectively). The bit allocation is chosen to ensure the dynamic range $\frac{max|h_i|}{min|h_i|}$ can be represented (i.e., dynamic range of $\sim 2^{15}$ for the decimation filter necessitates 4 bits for s_i and 1 bit for m_i, while dynamic range of $\sim 2^8$ for the band-pass filters necessitates 3 bits for s_i and 2 bits for m_i). Fig. 3 (a) and (c) show histograms of the coefficients with bins corresponding to the quantization levels, and (b) and (d) show histograms of the quantization error. Comparing with Fig. 1, we see that the quantization levels are used more uniformly, implying more efficient quantization. However, the quantization error, while small in most cases, can now be larger than 1 (which is the maximum with linear quantization) and is so for a few cases. In fact, with floating-point representation, the maximum value of the error is proportionate with the multiplier value. This somewhat mitigates the negative effect because it implies reduced percentage error. Nonetheless, with larger errors now possible, we propose the optimization presented in the next section to substantially reduce the quantization error of the output samples.

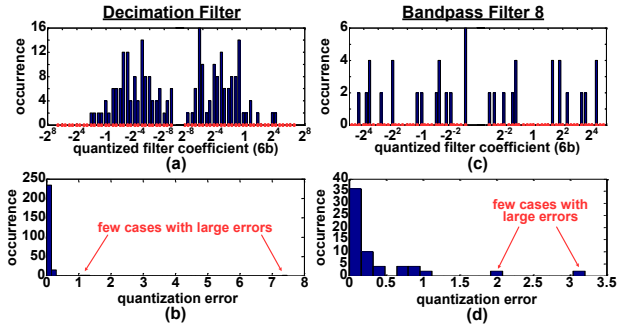

Fig. 3. Floating-point quantization of filter coefficients to the 6 bit level for (a)-(b) decimation filter and (c)-(d) last band-pass filter for feature extraction in the seizure detector.

Beyond optimizing quantization, a key benefit enabled by floating-point multiplier representation is that it can lead to very low complexity implementation of multiplication hardware. In particular, multiplication hardware at only the level of precision of m_i and a sign bit (to account for l_i) is required, along with multiplicand addition to implement the offset $(m_i + 1)$. In a traditional floating point multiplier, hardware for addition of the exponents (s_i) and rescaling of the mantissa product is also required. However, with fixed-point multiplicands, only a barrel shifter is necessary to apply the exponent s_i. Thus, very simple implementation is possible (with energy and area implications considered in Section 3.2).

2.3. Coefficient Quantization Error Optimization

To address the large quantization errors possible, we now propose an approach by which floating-point representation for the filter coefficients can be exploited to reduce the quantization error of output samples. The approach is based on introducing a scaling parameter α, which can be applied to all filter coefficients. This has the effect of simply scaling the output samples, but, as shown in Fig. 4, it gives us a knob whereby the coefficients can be mapped to values that yield lower quantization error. With α corresponding to a single scaling parameter, this requires quantization levels whose separation is appropriately scaled, as in the case of the exponential spacing achieved with floating-point representation. Below, we present the optimization framework by which a suitable α can be chosen.

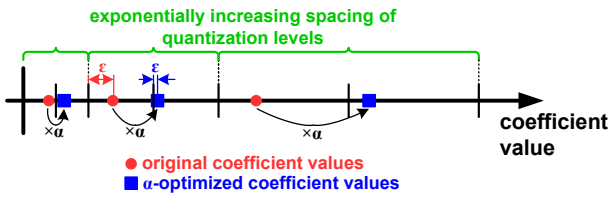

Fig. 4. Illustration considering three coefficients, showing how the values after α optimization (blue squares) can lead to lower quantization error compared to the original values (red dots).

First, consider X_i to be a random variable corresponding to the i^{th} input sample from Eq. 1. We assume that all X_i's in Eq. 1 are identically and independently distributed (IID). Note that identicalness should hold given that all X_i's are drawn from the same signal. However, independence may not hold in some cases. Nonetheless, the assumption is made here for the convenience of the derivation, and results from a practical application are presented, validating the approach.

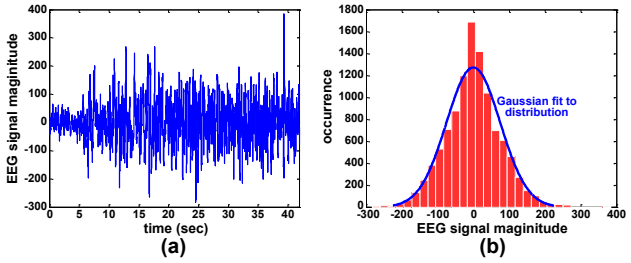

Fig. 5. (a) A 42-second segment of EEG from one channel, sampled at 256 Hz, and (b) a histogram of the sample values, fit to a normal distribution.

Next, to derive the output quantization error ϵ_Y, we must first model the distribution of X_i. Generally, this distribution depends on the application signal. We proceed with EEG data from the application case study. Fig. 5(a) shows a 42-second segment of an EEG channel sampled at 256Hz. A histogram of the samples is shown in Fig. 5(b), suggesting the distribution can be modeled as a normal $X_i \sim N(0, \sigma^2)$. Following from Eq. 1 and the IID assumption, we have $Y \sim N(0, \sum_i h_i^2 \cdot \sigma^2)$, showing that the output of filter is also normally distributed. The variable of interest is the output sample error due to quantization of the filter coefficients. This error is defined as $\epsilon_Y = Y - \hat{Y}$, where $\hat{Y}$ is the output obtained with quantized coefficients $\hat{h}$ (note, this corresponds to the negative of error as typically defined, and is adopted here for convenience to yield positive values for quantization error). This gives us:

$$\epsilon_Y = Y - \hat{Y} = \sum_i h_i \cdot X_i - \sum_i \hat{h}_i \cdot X_i = \sum_i \epsilon_{h_i} \cdot X_i, \quad (6)$$

from which we can conclude that $\epsilon_Y \sim N(0, \sum_i \epsilon_{h_i}^2 \cdot \sigma^2)$. Thus, to minimize the output quantization error, we must minimize the cost function $C_{\vec{h}=(h_0,h_1,\cdots,h_{n-1})} = \sum_i \epsilon_{h_i}^2$. Notice, this cost function essentially states that the quantization error of the filter coefficients must be minimized. To facilitate this minimization, we introduce the scaling parameter $\alpha \in R^+$, which merely has the effect of scaling the filter output samples by a constant factor (this is acceptable in most systems). Applying this scaling factor to the coefficients, represented by the vector $\vec{h}$, we now have a cost function that can be optimized over the parameter α:

$$min_\alpha C_{\vec{h}}(\alpha) \quad (7)$$

To solve this, we derive ϵ_h. Supposing we quantize m to the k-bit level. Thus, the quantized $\hat{m}$ can be expressed as in Eq. 8, ϵ_h can be expressed as in Eq. 9, and, with scaling by α applied, $\epsilon_h(\alpha)$ can then be expressed as in Eq. 10:

$$\hat{m}_i = \left\lfloor m_i \times 2^k \right\rfloor / 2^k; \quad (8)$$

$$\epsilon_{h_i} = h_i - \hat{h}_i = l_i(1 + m_i)2^{s_i} - l_i(1 + \hat{m}_i)2^{s_i}; \quad (9)$$

$$\epsilon_{h_i}(\alpha) = \frac{\alpha h_i - (\alpha \hat{h}_i)}{\alpha} = l_i \frac{2^{s_i(\alpha)}}{\alpha}(m_i(\alpha) - \hat{m}_i(\alpha)). \quad (10)$$

We note that, α only appears in Eq. 10 via a function having the following form: $f(\alpha) = 2^{\lfloor log_2(\alpha|h_i|) \rfloor}/\alpha$. For this function, $f(\alpha) = f(2\alpha)$. We thus have $\epsilon_h(\alpha) = \epsilon_h(2\alpha)$ and $C_{\vec{h}}(\alpha) = C_{\vec{h}}(2\alpha)$, implying that there exists a global minimum for the cost function $C_{\vec{h}}(\alpha)$ in the range $\alpha \in [1, 2)$. Consequently, the optimization can be easily solved by searching for the optimal α in this range. Doing this, Fig. 6(a) and (b) show the coefficient quantization error distributions that result for the decimation and last band-pass filters, respectively. Comparing with Fig. 3, we see that the coefficient error is substantially reduced.

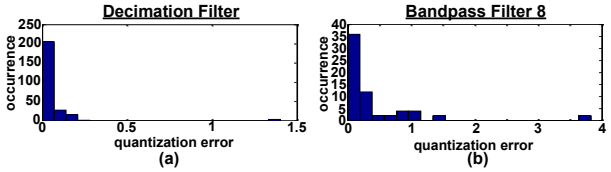

Fig. 6. Floating-point quantization of filter coefficients after α-optimization to a 6 bit level for (a) decimation filter and (b) last band-pass filter used for feature extraction in the seizure detector.

3. DEMONSTRATION AND ANALYSIS

In this section, we analyze the impact on quantization error, energy, and area of a filter using the proposed approach to multiplication within an accelerator. For quantitative analysis, we consider the seizure-detection application described in Section 2, composed of FIR filters. Results are presented using >210 sec. of EEG data sampled at 256Hz [9, 10]. Three filter implementations are considered, and each implementation is analyzed with coefficient representation from 6-10 bits. The implementations are as follows:

1. Traditional fixed-point multiplication.
2. Mixed fixed-/floating-point multiplication, with direct quantization of filter coefficients based on floating-point representation (with l_i allocated 1 bit, s_i allocated 4 bits, and m_i allocated the remaining bits).
3. Mixed fixed-/floating-point multiplication, with α-optimized quantization of filter coefficients based on floating-point representation.

All implementations are developed in both MATLAB and RTL Verilog. For analysis of quantization error, the MATLAB implementations are employed. For energy and area analysis, the Verilog implementations are synthesized to standard cells in a 32nm CMOS technology, and simulations are performed at the gate level using a high-capacity simulator (NanoSim).

3.1. α-Optimization of Filter Coefficients

Before showing results for the various filter implementations, we provide details pertaining to α-optimization of the filter coefficients, by using the decimation filter with 6-b coefficient representation as an example (the approach is similar for other cases). Fig. 7(a) shows the cost function (Eq. 7) for $\alpha \in [1, 1024]$. In particular, we see that the function is indeed periodic on a log scale, thus enabling us to restrict our focus to a range $\alpha \in [1, 2)$ for finding the global minimum. Fig. 7(b) shows the cost function in this range and identifies the minimum point at α=1.023 (blue point). Compared to the cost-function value obtained without explicit optimization, i.e., α=1 (green point), we see a 10× reduction in the cost-function value, which corresponds to the variance of quantization error for the filter coefficients.

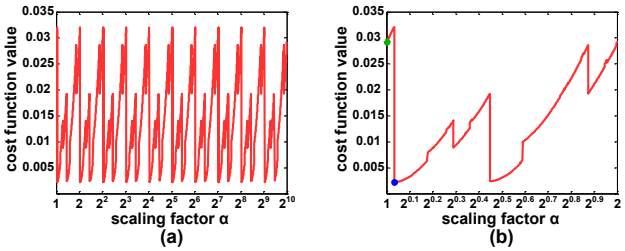

Fig. 7. Cost function for 6-b quantized decimation filter coefficients with (a) $\alpha \in [1, 1024]$, and (b) $\alpha \in [1, 2]$, showing value at global minimum (blue dot) and value without optimization (green dot).

Next, having found the optimal α, we examine the impact on the quantization error of the filter output samples. Fig. 8 shows the quantization error ϵ_Y of the three implementations (for 6-b coefficient representation). Here, the outputs Y have explicitly been sorted in order of increasing value to aid visualization, and, for reference, the output sample values have also been plotted, derived using double-precision computation. As seen, fixed-point coefficient representation results in substantial error. While floating-point coefficient representation reduces this, the error is substantially reduced further with the optimization.

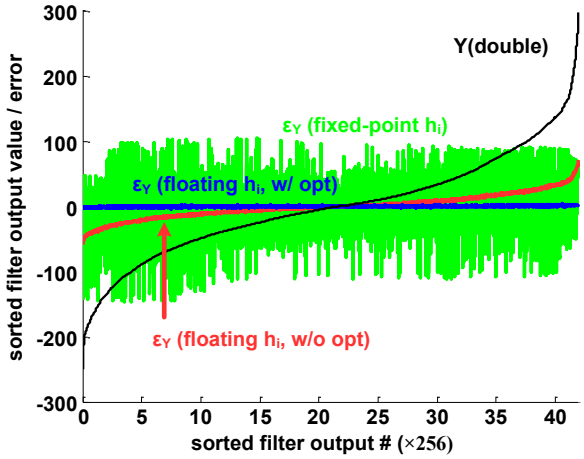

Fig. 8. Comparison of output-sample quantization error for decimation filter with 6-b coefficient representation in three implementations: (green) traditional fixed-point multiplication; (red) mixed fixed-/floating-point implementation without α optimization; (blue) mixed fixed-/floating-point implementation with α optimization. Points in the curves are explicitly sorted in order of increasing value of the output sample Y.

3.2. Performance of Implementations

Having demonstrated the α-optimization approach in one example, here we compare the three implementations in terms of three metrics: (1) computational error, quantified by the root-mean-square error of the filter output samples, normalized to the ideal values; (2) energy consumption, quantified by the average filter energy consumed per clock cycle (i.e., per output sample), as derived from NanoSim simulation; and (3) hardware complexity, quantified by the number of equivalent NAND gates required in the implementation, as derived from RTL synthesis. We point out that although the reduced complexity of the floating-point implementations can enable faster clock speeds, all implementations run at 100MHz.

Fig. 9a shows the computational error for the three implementations for the decimation filter. We see that the proposed implementation with α optimization leads to substantially lower quantization error; in particular, at the 6-b level, the error is 37× lower than an implementation based on fixed-point coefficient representation and 28× lower than an implementation without α optimization. As shown in Section 2.3, with normally distributed input samples, the filter output samples also follow a normal distribution. Thus, the same optimization approach can be applied to the band-pass filters, which are fed by the decimation filter. The computational error thus achieved at the output of the band-pass filters is shown in Fig. 9b. We see that, at 6 bit level, with floating-point coefficient representation and α optimization, the output error is 9× lower than an implementation based on fixed-point coefficient representation and 7× lower than an implementation without α optimization.

Fig. 9 (c) and (d) show the energy consumption and hardware complexity, respectively, for the fixed-point and floating-point based implementations (the energy and hardware complexity are roughly equivalent with and without α optimization in the floating-point based implementations). Since floating-point coefficient representation requires multiplication involving fewer bits (only those used for $\hat{m}_i$) lower energy and area are observed compared to the case with fixed-point coefficient representation. The remaining operations required with floating-point coefficient representation (barrel shifting and sign application) consume much less energy and hardware compared to multiplication. In particular, at the 6-b level, the proposed implementation leads to 1.4× lower energy and 1.2× lower hardware complexity.

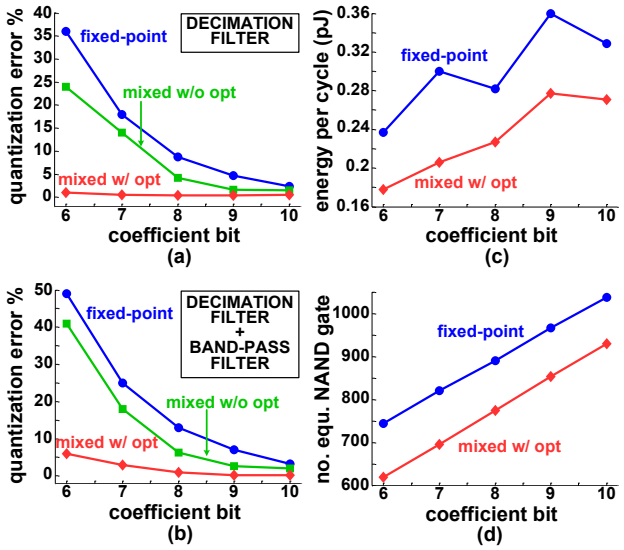

Fig. 9. Comparison of (a)-(b) computational error, (c) energy consumption, and (d) hardware complexity of the various filter implementations within the seizure-detection system.

4. CONCLUSION

This work proposes an optimization for FIR filter coefficients, made possible when floating-point representation is used, whereby the quantization error of filter output samples is substantially reduced. The optimization approach is presented, and its application in FIR filters within an EEG-based seizure detector is demonstrated and evaluated. Further, with fixed-point coefficient representation, the multiplication hardware required is reduced (to just the bits designated for the mantissa m_i). Thus, both the energy and hardware complexity, evaluated via gate-level synthesis and simulation, are also reduced.

5. REFERENCES

[1] C. Lin, Y. Chen, T. Huang, T. Chiu, L. Ko, S. Liang, H. Hsieh, S. Hsu, and J. Duann, "Development of wireless brain computer interface with embedded multitask scheduling and its application on real-time driver's drowsiness detection and warning," *IEEE Trans. Biomedical Engineering*, vol. 55, no. 5, pp. 1582–1591, 2008.

[2] 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 Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 804–816, Apr. 2010.

[3] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM Int. Workshop on Wireless Sensor Networks and Applications*, 2002, pp. 88–97.

[4] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.

[5] J. Kwong and A. Chandrakasan, "An energy-efficient biomedical signal processing platform," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 7, pp. 1742–1753, 2011.

[6] S. Sridhara, M. DiRenzo, S. Lingam, S. Lee, R. Blazquez, J. Maxey, S. Ghanem, Y. Lee, R. Abdallah, and P. Singh, "Microwatt embedded processor platform for medical system-on-chip applications," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 4, pp. 721–730, 2011.

[7] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Transactions on VLSI Systems*, vol. 9, no. 6, pp. 813–823, 2001.

[8] C. Kim, H. Soeleman, and K. Roy, "Ultra-low-power DLMS adaptive filter for hearing aid applications," *IEEE Transactions on VLSI Systems*, vol. 11, no. 6, pp. 1058–1067, 2003.

[9] A. Shoeb, *Application of machine learning to epileptic seizure onset detection and treatment*, Ph.D. thesis, Massachusetts Institute of Technology, 2009.

[10] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. Ivanov, R. Mark, J. Mietus, G. Moody, C. Peng, and H. Stanley, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.