

Hibernets: Energy-Efficient Sensor Networks Using Analog Signal Processing

Brandon Rumberg

David W. Graham

Vinod Kulathumani

Lane Department of Computer Science and Electrical Engineering

West Virginia University

brumberg@mix.wvu.edu, {david.graham, vinod.kulathumani}@mail.wvu.edu

ABSTRACT

In-network processing is recommended for many sensor network applications to reduce communication and improve energy efficiency. However, constraints on memory, speed, and energy currently limit the processing capabilities within a sensor network. In this paper, we describe how ultra-low-power analog circuitry can be integrated with sensor nodes to create energy-efficient sensor networks. We present a custom analog front-end which performs spectral analysis at a fraction of the power used by a digital counterpart. We then show that the front-end can be combined with existing sensor nodes to (1) selectively wake up the mote based upon spectral content of the signal, thus increasing battery life without missing interesting events, and to (2) achieve low-power signal analysis using an analog spectral decomposition block, freeing up digital computation resources for higher-level analysis.

Categories and Subject Descriptors

C3 [computer systems and organization: special purpose and application-based systems: signal processing]: B8:Hardware: performance and reliability J2:Computer applications: physical sciences and engineering

Keywords

Analog Signal Processing, Energy-Efficient, Selective Wake Up, In-Network Processing, Sensor Networks

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

1. INTRODUCTION

Wireless sensor networks (WSN) hold great promise for use in applications such as environmental monitoring, protection of borders and resources against intruders, and monitoring critical infrastructure like bridges and power grids [2, 3, 18, 4, 34]. However, a combination of factors continues to prevent wide-scale deployment of sensor networks for these applications. Key among these factors is energy efficiency [32] and the ability to last for long durations on small power sources such as batteries and energy harvesting systems.

One strategy to conserve energy locally is to perform minimal computation at each node while transmitting most of the data, thereby leaving a majority of the computation and any necessary decision-making to one or more centralized units. However, this strategy leads to increased communication overhead. On the other hand, local processing and in-network aggregation are recommended for reducing power consumption due to the high cost of communication. However, the amount of processing that a node can perform is restricted by both its power budget and its limited processing resources. These constraints restrict the amount of signal processing that the node can perform and also limit the highest sampling frequency at which processing can be sustained. One solution for performing more advanced signal processing and working with higher frequency signals is to use a mote with a faster processor such as the Intel Imote2 [24] or the Stargate platform, but technique comes at the expense of higher power consumption and cost.

1.1 Contributions and Main Results

In this paper, we suggest augmenting sensor nodes with ultra-low-power analog signal processing (ASP). These ASP systems extend the capabilities of the mote while contributing little to the node's required power budget. We describe an ASP/WSN framework (Figure 1), in which the ASP consists of two parts: signal analysis and event detection / classification. The analysis portion serves dual purposes: (1) generate features for

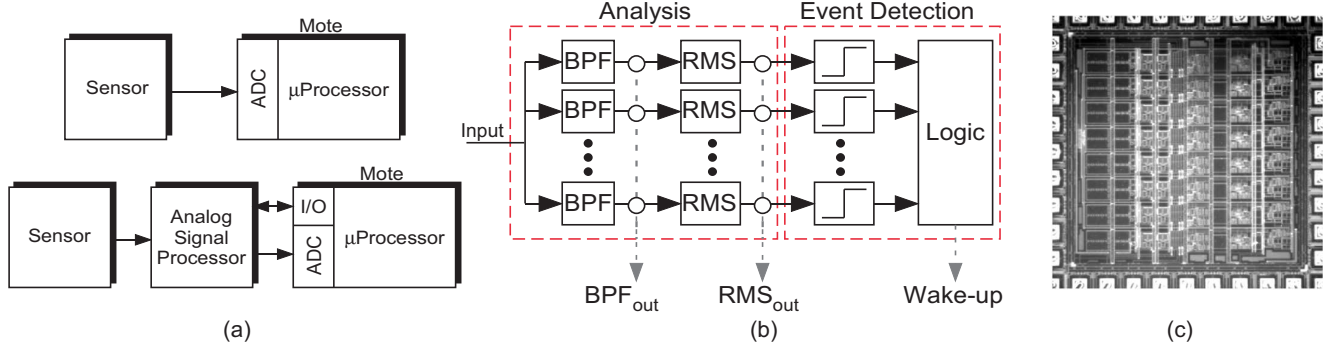


Figure 1: (a) In contrast to many WSN designs in which sensor data are directly converted into the digital domain by a mote, we introduce an intermediate stage composed of analog circuits for pre-processing of the sensor data. This analog pre-processor allows us to compress the sensor data into relevant characteristics, improve the performance of event detection (while letting the mote sleep), add processing capabilities, and save considerable amounts of power. (b) Our analog signal processor is capable of performing spectral decomposition of the incoming signal and generating interrupt signals to wake a sleeping mote. This integrated circuit is composed of a bank of bandpass filters (BPFs) for spectral decomposition, a magnitude/RMS detection circuit for determining power levels within a subband, and comparators for generating interrupt signals. (c) Integrated circuit die photograph.

use in event detection and (2) perform pre-processing to free up the mote’s computing resources. The classifier wakes the mote when it detects events of interest, and allows the mote to operate at a higher abstraction level, dealing with sensor data at the level of classes.

To demonstrate the potential of the ASP/WSN framework, we have designed and fabricated an analog integrated circuit (Figure 1) for use in wireless sensor networks. The analysis portion of the system performs spectral decomposition using a constant relative-bandwidth filter bank with subband root-mean-square (RMS) detection circuits. Event detection is performed with a comparator on the RMS output of each subband, followed by digital logic which asserts a hardware interrupt when the signal spectrum matches a user-defined binary template. The core of this chip operates at an average power of $1 - 3\mu W$, which is less than the power consumed by a TelosB mote in its lowest-power sleep mode.

We note that spectral decomposition is a crucial first-step for many sensor network applications such as acoustic or seismic object classification, event detection, and vibration monitoring [34, 4, 2]. Therefore, these analog circuits hold great promise for use in wireless sensor networks, and in this paper, we show different ways to leverage these circuits.

In order for an ASP/WSN system to be practical, the use of the ASP must be as straight-forward as writing programming code in a higher-level programming language. Also, there should be some flexibility in controlling the parameters of the circuit at run-time and after

deployment. With these requirements in mind, we have interfaced the ASP to a TelosB mote [26]. All signal analysis outputs are multiplexed to a single analog-to-digital converter (ADC) pin on the mote, allowing the mote to sample these outputs using standard TinyOS [19] sensor interfaces. The event detector can be set to generate an interrupt when activity has been detected in a user-selected combination of channels. Additionally, the frequency range and spacing of the filter bank can be varied using the mote’s built-in digital-to-analog converters (DACs).

1.2 Related Work

Many WSN applications require some form of spectral analysis for detection and classification of events [4, 7, 22, 13, 5, 2]. All of these applications have discussed the need for processing within the network in order to decrease communication requirements. Our analog front-end would complement all such systems by providing low-power processing capabilities.

Hardware-based event detection, in contrast to sensor polling, has been suggested for reducing power consumption in sensor networks. Jevtic, et al. [27] reported a crack monitoring device which uses a comparator to trigger a wake-up signal based on the amplitude of the signal. Their complete wake-up circuit consumes $16.5\mu W$, and they describe the use of both a passive sensor for event detection and a high-precision sensor for event recording. Malinowski, et al. [23] developed a cargo-monitoring tag with a total quiescent current of approximately $5\mu A$. In their event detection circuits,

they prepend peak detectors to the comparators, triggering interrupts based on the envelope of the signal. They also describe a dynamically adjustable threshold scheme to achieve a post-event refractory period. Goldberg, et al. [9] presented an acoustic surveillance system, which uses a digital VLSI periodicity detector (with a core power consumption of $835nW$) to wake up the system. In this paper, we present an analog event detector which goes beyond amplitude-based event detection. We also show how the signal analysis performed for event detection can contribute to the mote's processing needs.

As power constraints on various types of systems are becoming more stringent, analog circuits are being re-investigated for use in low-power systems, such as hearing prostheses [30], implantable electronics [15], and signal-processing systems [16, 8, 31]. While digital circuits have been used in most settings because of their ease of use through programming, noise robustness, and scalable dynamic range, analog circuits are able to operate in real-time and perform many computations inherently that would require significant overhead in the digital domain (e.g. multiplication) [17]. Furthermore, as was shown by Sarpeshkar in [28], analog circuits are able to perform many computations more efficiently than digital systems for given levels of signal-to-noise ratio (SNR) – typically for all but the highest-fidelity systems. Additionally, recent developments in programmable / reconfigurable analog systems enable analog integrated circuits to be general-purpose and easy to use [21, 14]. In this work, we demonstrate how these analog circuits and systems are capable of making a significant impact on WSN designs by performing sophisticated computations, including spectral decomposition, all while consuming less power than a sleeping mote.

1.3 Experimental Methodology

Our prototype setup combines our custom ASP with external logic (for event detection), connected to a Telos mote. Since our analog/mixed-signal system will use the same sensors as an exclusively digital approach, we bypassed the use of sensors and used a standalone DAC to supply arbitrary waveforms for the demonstrations. We point out that in order for the ultra-low-power of the custom integrated circuit to reduce overall system power, it is necessary that ultra-low-power sensors be used when the mote is asleep. Additionally, the sensors should generate frequencies greater than $1Hz$, otherwise a polling approach would be more appropriate. The system we describe is based on spectral analysis, so low-power vibration sensors (e.g. piezo) are a good match the system. Except where noted otherwise, all plots in this paper depict the actual measurements of the experiments done with the setup described above.

1.4 Outline of the Paper

In Section 2, we describe the design of our analog signal processor. In Section 3, we describe the integration of the ASP with a TelosB mote. In Section 4, we describe the performance evaluation of the integrated system and quantify the impacts on WSN design.

2. DESIGN OF ANALOG COMPUTATIONAL ELEMENTS

2.1 System Overview

Our analog signal processor, shown in Figure 1, is fabricated on a $0.5\mu m$ standard CMOS process available through MOSIS. This integrated circuit is $2.25mm^2$ and consumes only $3\mu W$. The intent is to make a low-power, but discriminating, event detector which can call attention to compelling characteristics of a signal. The detection approach is to identify when the signal matches a binary spectral template. This integrated circuit (Figure 1) has two stages: (1) a spectral analysis stage, and (2) an array of comparators, which, combined with external logic form the event detector.

2.2 Spectral Analysis Front-End

The spectral decomposition front-end is composed of a filter bank with subband RMS circuits. This spectral analysis system is used for frequency-dependent event detection and for offloading some of the signal processing which would otherwise be performed by the mote. Since the outputs of all the filters and RMS circuits are multiplexed to a single pin, a mote can select the filter output or RMS output of any frequency band in order to acquire a frequency-domain representation of the signal.

An array of eight bandpass filters (BPFs) is used to create the constant relative-bandwidth filter bank. The BPFs are based on the compact filters we presented in [11, 10], shown in Figure 2(a), and have been modified to increase the linear range of operation. These are electronically-tunable continuous-time filters which are suitable for low-frequency, low-power applications. The corner frequencies are orthogonally tunable through voltages V_{τ_l} and V_{τ_h} , which control, g_{m1} and g_{m4} , respectively, in (1). The transfer function for the filter is

$$\frac{V_{out}}{V_{in}} = \frac{C_1}{C_2} \frac{s\tau_l(1 - s\tau_f)}{1 + s\left(\tau_l + \tau_f\left(\frac{C_O}{C_2} - 1\right)\right) + s^2\tau_h\tau_l} \quad (1)$$

where the time constants are given by

$$\tau_h = \frac{C_T C_O - C_2^2}{C_2 g_{m4}} \quad \tau_l = \frac{C_2}{g_{m1}} \quad \tau_f = \frac{C_2}{g_{m4}} \quad (2)$$

and where $C_T = C_1 + C_2 + C_W$ and $C_O = C_2 + C_L$. Since these circuits are operated in weak inversion, the

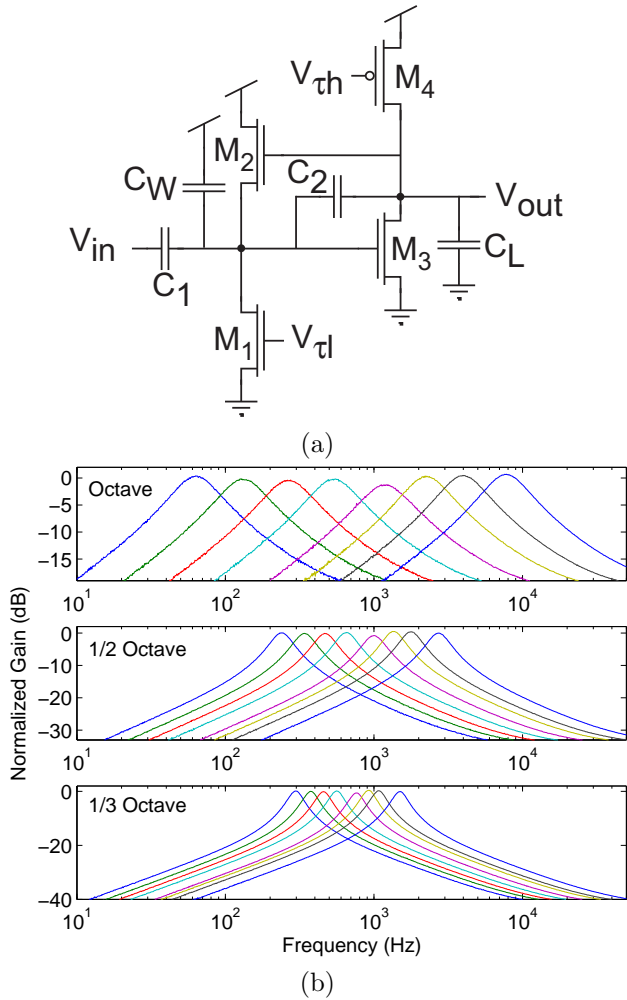


Figure 2: (a) Simplified BPF circuit (b) AC response of the filter bank for (1) octave spacing, (2) 1/2 octave spacing, and (3) 1/3 octave spacing.

transconductance values (g_{m1} , g_{m4}) vary exponentially with the bias voltage. This exponential relationship between voltage and frequency allows us to achieve the desired log-frequency spacing across the whole filter bank using a simple resistive divider, internal to the chip. The voltages on either end of the resistive divider can be tuned to cover different frequency ranges and spacings, similar to [33, 29]. We use the $1/N$ octave spacing convention [1], which is common in vibrational and acoustical analyses. In fractional octave spacing, there are N filters per octave, and the filters cross at their $-3dB$ points. Figure 2(b) demonstrates the ability to set the filter bank for one, two, or three filters per octave.

Our RMS circuit is used to extract magnitude information from each subband via peak detection and fil-

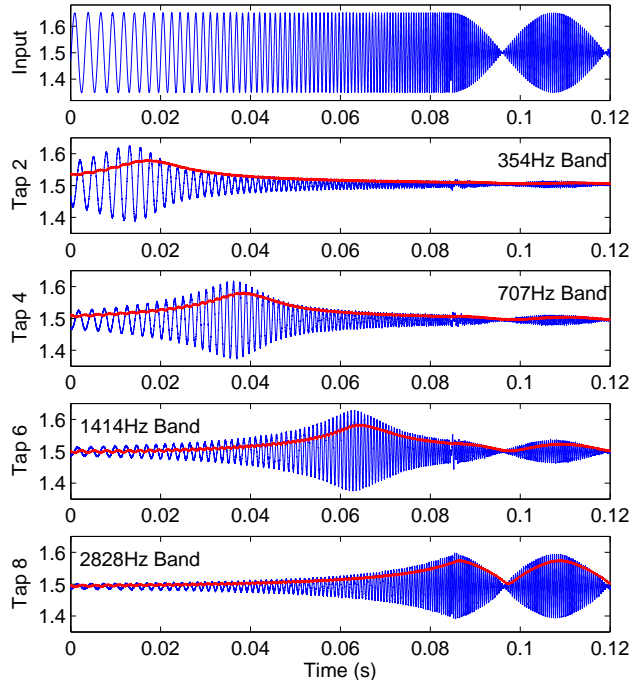


Figure 3: Demonstration of the spectral decomposition front-end. The input, top plot, is a logarithmic chirp. The rest of the plots show the response of four of the subbands. Note that each channel’s response is frequency dependent, and that the RMS outputs represent spectral characteristics of the signal.

tering of the subbanded signal. The RMS circuit is a continuous-time circuit, based on [20], that is used to rectify and lowpass filter an incoming signal. The RMS circuit has three adjustable parameters: attack rate, decay rate, and filtering time constant. The attack and decay rates can be adjusted to track the RMS or envelope of the signal. The filtering time constant is adjusted to obtain the best compromise between responsiveness (phase lag) and accuracy (rejecting ripple from the peak detector). Figure 3 demonstrates the combination of the filter bank and RMS detector. In this demonstration, our spectral decomposition system is set for 1/2-octave spacing, starting at 250Hz. The input to the filter bank is a logarithmic chirp signal. Below the input are the responses of the second, fourth, sixth, and eighth bands of the decomposition system. As the chirp sweeps to higher frequencies, the response of the higher subbands increases and the response of the lower subbands decreases. Note that the output of the RMS circuit is the signal content in that band. Voltage biases for the RMS circuits are set by a resistive divider in the same way as for the filters, allowing the frequency range of the entire

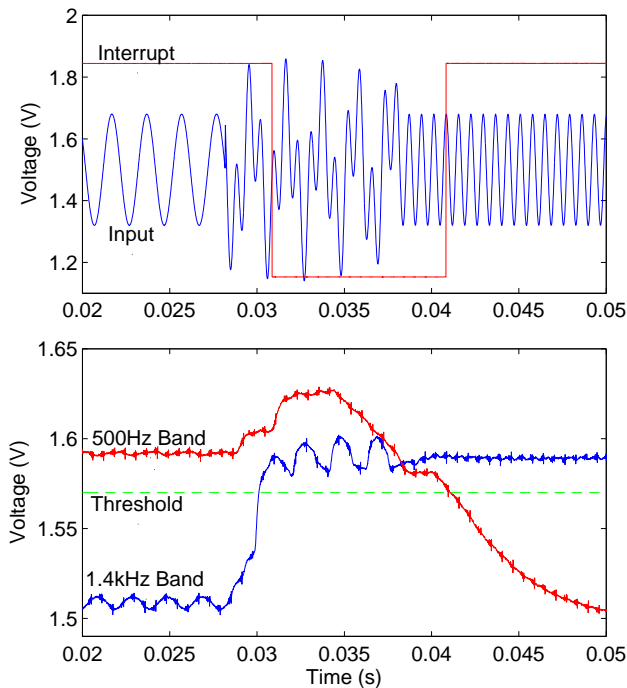


Figure 4: Demonstration of multi-band detection using an exclusive-or template. The input consists of two overlapping sine waves. The bottom plot shows the magnitude outputs for the 500Hz and 1.4kHz bands. The comparator outputs of those two bands are combined via an exclusive-or to generate an interrupt when only one band exceeds the threshold.

spectral analysis system to be controlled by two bias voltages.

While using a resistive divider to bias the filter bank makes the ASP easy to use, there are several drawbacks. First, the accuracy of the filter parameters depends on the matching of the resistors, which is generally poor. The effects of this mismatch can be observed by looking at the variation in gain across the AC sweeps in Figure 2(b). Second, if using the mote's DAC to permit run-time bias changes, resistive biasing will require the mote's DAC to remain on all the time, adding to quiescent power draw. Both of these issues can be solved by using floating-gate transistors for parameter biasing [12]. Floating-gate transistors allow precise programming of each parameter. Also, since floating-gate transistors are non-volatile, they do not require any external biasing once they have been programmed.

2.3 Event Detection

By combining the spectral analysis system with com-

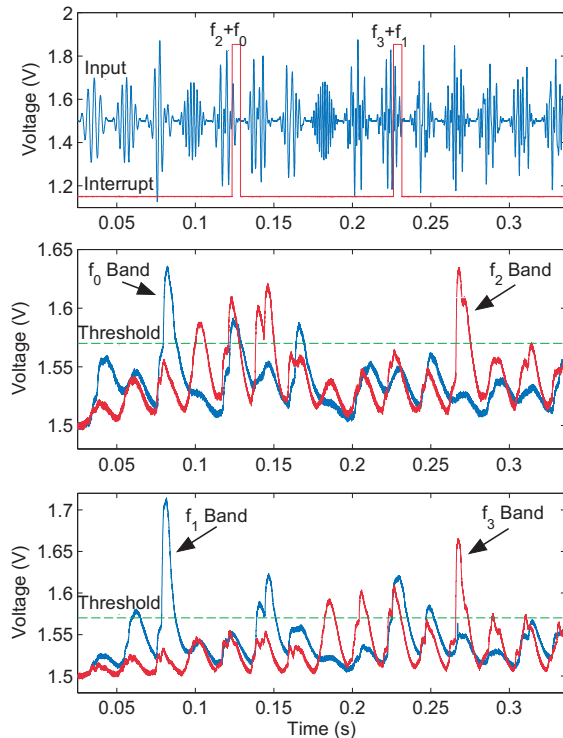


Figure 5: In this demonstration an interrupt is generated when the signal consists of only harmonically spaced frequencies. The filter bank is set for half octave spacing and we are only concerned with the bottom four bands, f_0 , f_1 , f_2 , and f_3 .

parators and digital logic, we form a simple yet selective event-detection system, with the flexibility to define what constitutes an event. In this section, two examples are used to illustrate multi-band event detection.

For the first example, (Figure 4) an event is defined as occurring when signal content is present in one of two channels, but not both. The two bands being compared are 500Hz and 1.4kHz. The input consists of a 500Hz sine wave and a 1.4kHz sine wave which overlap for 10 milliseconds. The wakeup signal is generated by combining the comparator outputs for those two bands using an exclusive-or (XOR) operation, so that the interrupt is asserted only when one band exceeds the threshold.

In the second example, we detect harmonically spaced content. With the filter bank set for $1/N$ octave spacing, harmonic detection can be performed by checking every N bands. In this demonstration we are only looking at the bottom four bands ($f_0 = 250Hz$, $f_1 = 354Hz$, $f_2 = 500Hz$, and $f_3 = 707Hz$) and have defined an event as $Event = f_0\bar{f}_1\bar{f}_2\bar{f}_3 \vee \bar{f}_0f_1\bar{f}_2f_3$, where f_i de-

notes signal presence in the i -th band. The results of this test are shown in Figure 5. The input consists of the 15 different combinations of signal presence in those four bands. The middle plot shows the detection of harmonic spacing off of the 250Hz band while the bottom plot shows detection off of the 354Hz band. In the top plot we see that the front-end successfully generates interrupts for the desired frequency combinations.

In the above examples, we observe that there is some lag-time between when the event occurs and when the interrupt signal is generated. The lag-time is caused by the RMS circuit, and is a result of filtering the peak-detected signal. By adjusting the parameters of the RMS circuit, the phase-lag can be reduced, at the expense of reduced RMS tracking accuracy. This lag time is related to the frequency, f , of the subband, and is approximately $4/f$ for the RMS circuit biasing used in this paper. For an application where the mote should record the event, this phase-lag could cause the onset of the event to be overlooked. Regardless of how small the phase lag is, we will miss the prelude to the event. This problem will be present in all systems which wake up based on event detection. To solve the phase-lag problem, the designer can include a memory buffer. This buffer may take the form of an analog delay line (continuous-time continuous-value), an array of sample-and-holds (discrete-time continuous-value), or low-power ADC and RAM (discrete-time discrete-value). This memory can also have a second use of adding memory to the event detector.

2.4 Power Consumption

The power consumed by our analog integrated circuit is dominated by the bandpass filters, and to a lesser extent, the magnitude circuits. As we presented in [11], which describes the circuit that this paper’s BPF is based upon, the power consumed by the BPF is linearly proportional to its center frequency. This relationship is shown in Figure 6 for a filter tuned to a 1/2-octave bandwidth. This relationship enables the system designer to determine the maximum frequency of operation available at a given power budget.

The power consumption of the RMS circuit also scales with frequency. However, the RMS circuit can be tuned in various fashions within a given frequency band f_0 ; for example, this circuit can follow the envelope or RMS of a signal. Therefore, this circuit has a range of power-consumption values for a given f_0 . Figure 6 shows the worst-case scenario (i.e. highest power consumption) for operation within a given frequency band, f_0 .

The overall power consumption of our analog spectral-decomposition block is set by the center frequency of

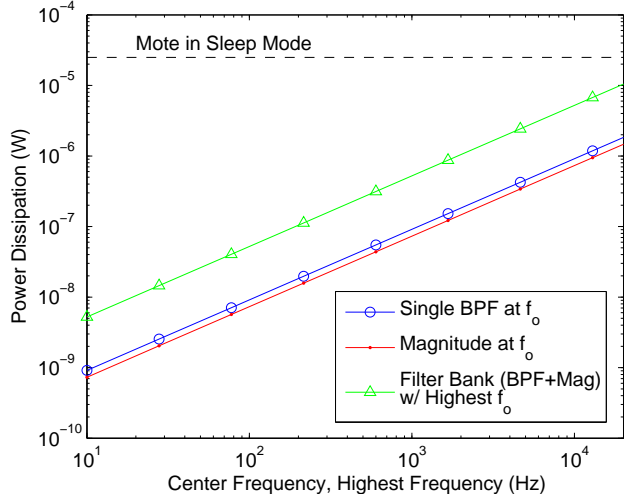


Figure 6: The power consumed by our analog spectral-decomposition block depends primarily on the center frequency of the bandpass filter of the highest-frequency subband. The x-axis shows the center frequency for the filter and RMS circuit, and also shows the center frequency of the highest-frequency subband for an array that performs spectral analysis. These are extrapolated from circuit simulations.

the highest filter tap. The power consumption of the entire spectral-decomposition system is described by a geometric series, resulting in a total power consumption of

$$P_{tot} = \frac{P_{BPF,high} + P_{RMS,high}}{1 - 2^{1/N}} \quad (3)$$

where $P_{BPF,high}$ and $P_{RMS,high}$ represent the power consumed by the BPF and RMS circuits in the highest-frequency subband, and N indicates the number of filters per octave. The total amount of power consumed by the analog block is shown in Figure 6 for the case of 1/2-octave spacing. Included in this figure is the measured power consumption of the TelosB mote in sleep mode ($25.4\mu W$, which is within the specified bounds of $15 - 60\mu W$). For the entire audio frequency band, our spectral-decomposition block consumes less power than a sleeping mote.

3. INTERFACING WITH THE TELOS MOTE

To evaluate the potential of the ASP/WSN framework, we interfaced the integrated circuit described in section 2.1 with the TelosB mote. The Telos mote was chosen for its low-power sleep mode and fast wakeup time, which make it suitable for a hardware-based wake

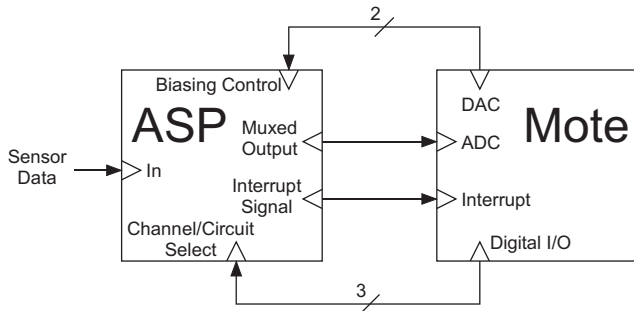


Figure 7: Interfacing of the analog signal processor with the mote. The analog hardware generates an interrupt signal for an interrupt-enabled general I/O pin to wake the mote. The mote can sample the processed output of the analog hardware with its ADC. The output of the analog signal processor is chosen via the mote’s digital I/O pins. Also the parameters of the analog signal processor are set via the mote’s DAC.

up system. We make use of the mote’s general I/O, ADC, and DAC for interfacing, illustrated in Figure 7.

From the ASP, the output of each subband’s bandpass filter and magnitude circuits are multiplexed to a single ADC pin on the mote. The subband to be sampled is selected via three digital selection lines which are controlled by the mote’s general I/O pins. To acquire the entire spectral representation, the mote selects a new subband between each sample. Due to the low-frequency nature of the RMS outputs, the mote is able to cycle through all channels without experiencing aliasing.

The spectral analysis front-end’s parameters (frequency range and spacing) are set by two bias voltages. These bias voltages can be set as constants or controlled at run-time by software using the DAC. The comparator thresholds are set by a bias voltage, which can also be manipulated at run-time for dynamic thresholds.

4. PERFORMANCE EVALUATION

In this section, we demonstrate how we can leverage the computational capabilities of the analog integrated circuit for WSN applications and quantify the performance gains achieved.

4.1 Selective Wake-Up and Energy Savings

First we compare the power consumed by the analog circuit for spectral decomposition with a digital implementation. For this comparison, we first implement a second-order Butterworth bandpass filter on a TelosB mote running TinyOS and measure the power consumed. In Figure 8, we show the power consumed by this digital

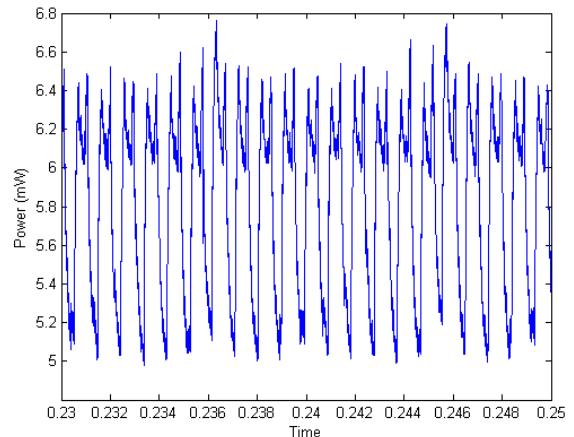


Figure 8: Measured power consumed by TelosB mote while running an IIR digital filter at 1kHz sampling frequency

filter implementation at a 1kHz sampling frequency. We use the *ReadStream* interface to buffer 100 samples at a time and then compute the filter outputs after every 100ms. The average power is determined to be 6.3mW. Similarly, we determine the average power over a range of frequencies from 10Hz to 1kHz. In Figure 9, we compare the average power consumed by a digital filter over different sampling frequencies with the power drawn by an analog bandpass filter over different center frequencies. We point out that the energy consumed by our entire spectral analysis system is over 1000 times lower than the power consumed by a single digital filter.

A mote can be operated in a duty-cycle-based polling mode where it periodically samples and then runs the digital filter to look for events of interest. This approach will decrease the power consumed by the mote. However, as the polling interval increases, the likelihood of overlooking events increases as well.

We can leverage the low-power processing capability of analog circuits, by placing the mote into long periods of hibernation and then selectively waking the mote when a specific frequency band has been detected. This is done by keeping the analog circuit always-on and monitoring for a threshold in the selected band. Figure 10 demonstrates single-band event detection. The band of interest is 1kHz and the filter has a quality factor of 2.8. Signal content appears in the band at 2.6 seconds but has noise added to it. This noise is a combination of white noise and tones at 100Hz, 600Hz, and 10kHz. The bandpass filter focuses on the frequency of interest and the comparator trips once the RMS reaches the threshold. This demonstrates that a detection scheme as simple as a hard threshold on the RMS of one subband is

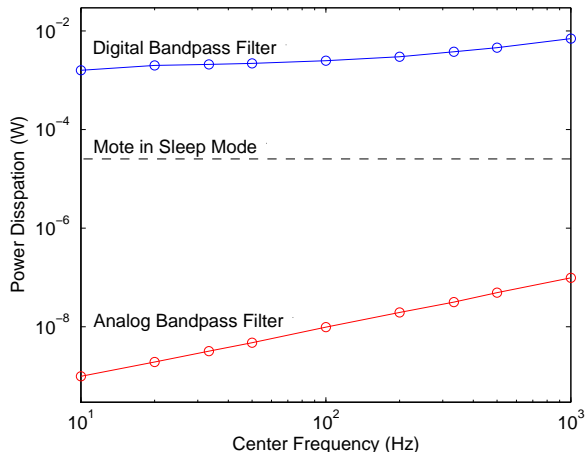


Figure 9: Comparison of the average power consumed by our analog bandpass filter and the digital bandpass filter shown in Figure 8. Power consumption is shown as a function of center frequency for the analog filter and sampling frequency for the digital filter. The analog trace comes from circuit simulation.

much more discriminating than just a comparator operating on the entire signal. Note that the subband event is detected despite having much lower amplitude than the noise and other frequencies. In this demonstration, the mote samples the sensor signal when it wakes up and transmits it to the basestation. The signal received by the basestation is shown in the bottom trace. While this demonstration shows the mote sampling for a fixed duration after the interrupt, we can also have the mote sample for the duration that the event detection interrupt is asserted.

4.2 Selective Sampling and Computational Savings

In this subsection, we highlight the ASP’s ability to perform pre-ADC signal analysis. In Figure 11(a), we repeat the demonstration in Figure 10, but with the mote only sampling the frequencies of interest. The ASP extracts this subset of frequencies, a process which would otherwise be done by the mote. Thus, we save computational resources at the mote, making room for higher-level signal analysis.

In the following experiment, we take advantage of the full spectral analysis performed by the front-end. To acquire the complete spectral characteristics, the mote samples the RMS output of each subband in succession. In the experiment (Figure 11(b)), the input signal consists of two 1kHz pulses followed by a chirp signal. The 1 kHz pulse is used to trigger the mote into sampling the

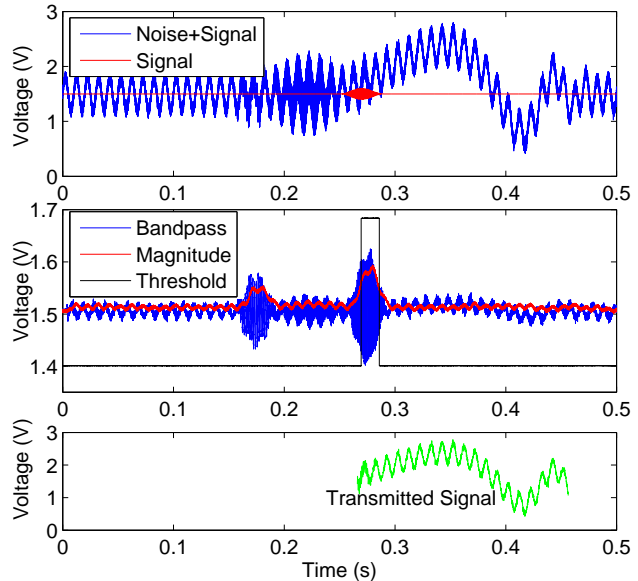


Figure 10: Single-band event detection. The frequency of interest is 1kHz and is the “Signal” trace in the top plot. Broadband noise and tones at 100Hz, 600Hz, and 10kHz are added to “Signal”. “Noise+Signal” is the input to the analog signal processor. The middle plot shows the response of the three stages of the processor within the 1kHz subband. The bandpass filter cuts the undesired frequencies, while the RMS circuit tracks the magnitude. Once the magnitude exceeds the threshold, an interrupt is generated to wake the mote. The mote then samples the output of the sensor and transmits it to the basestation. The received signal is plotted in the bottom plot.

RMS energy of each subband in succession, for a specified period of time. The mote scans through subbands by writing to the GIO port’s output register between each sampling operation. The frequency-decomposed RMS data obtained by the mote is transmitted to a base station and is displayed in the bottom plot. We note that by scanning through the energy of all the subband channels in succession, a complete spectral decomposition can be obtained at the mote in real time using the analog circuit.

Due to the limited resources of sensor nodes, the rate at which the mote can continuously sample is quite low. We demonstrate here that an analog frontend can perform dimensionality reduction on a high frequency signal, reducing it to a signal which is slow enough to sample. In this demonstration, (Figure 12) we obtain the RMS of a 12kHz input using the RMS circuit. This

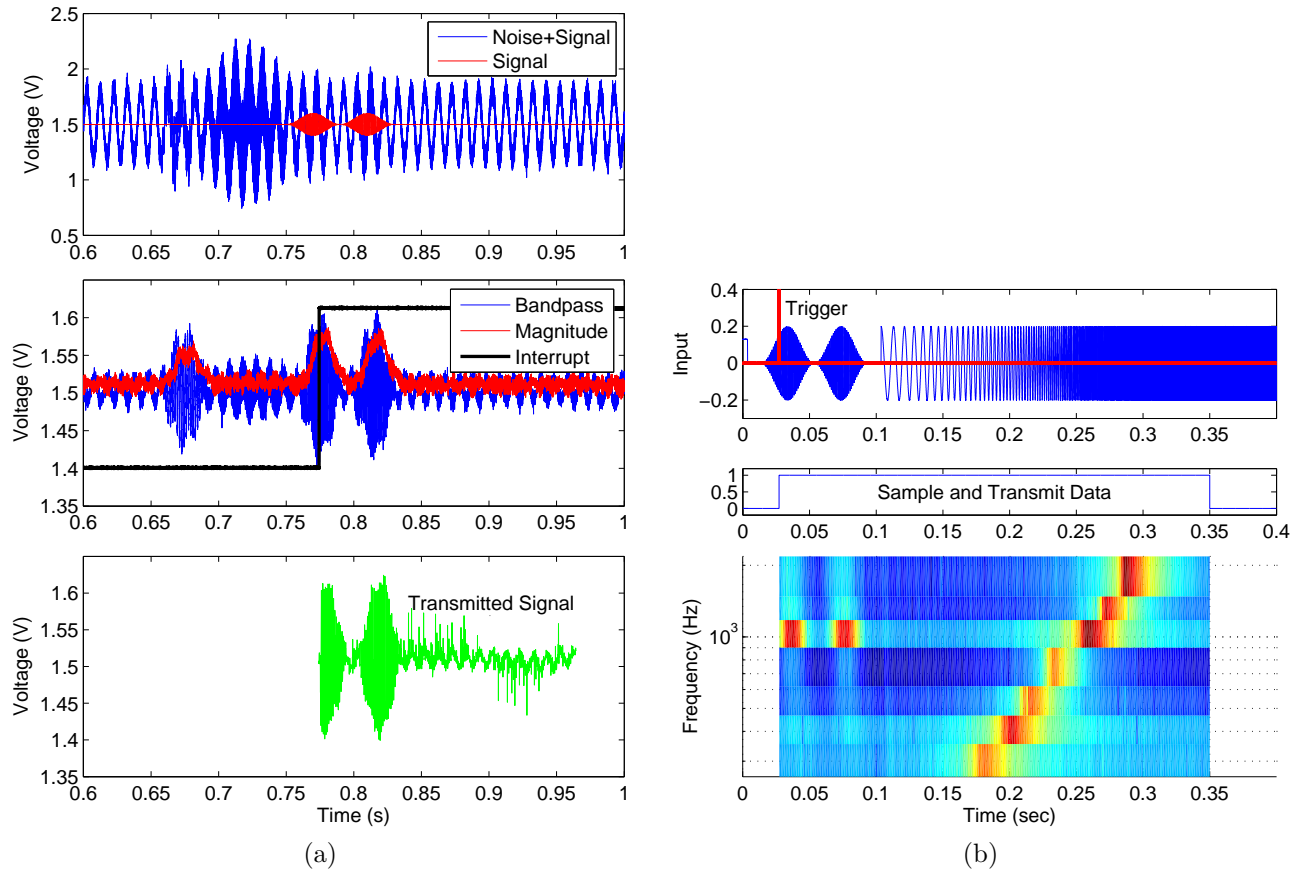


Figure 11: (a) Sampling a single frequency band of interest. Once the 1kHz band RMS exceeds a user-defined threshold, an interrupt signal is generated to wake the mote. The mote then samples the 1kHz subband and transmits it to the basestation. The received signal is plotted in the bottom plot. (b) Spectral analysis performed by the analog IC. (Top) The input signal is composed of two 1kHz pulses followed by a logarithmic chirp signal. (Middle) Once the trigger goes high, the ADC samples all 8 channels for a user-specified amount of time (e.g. 300msec). (Bottom) Spectrogram of the transmitted frequency-dependent magnitude data received by the base station.

process reduces the signal to just its amplitude information, which is of low enough frequency to sample at 2kHz, within the reach of the Telos mote.

5. CONCLUSIONS

In this paper, we described how ultra-low-power analog circuitry can be integrated with sensor nodes to reduce the node-level power consumption. We have shown the ability to interface these circuits with existing sensor platforms, and have presented demonstrations to illustrate how analog hardware can reduce node resource usage and increase performance in the following ways. First, detecting significant sensor events while the mote is sleeping extends the node’s battery life. Our event detector consumes less power than the mote’s sleep power and can be configured to trigger on a variety of signal

characteristics. Second, the ASP can be used to perform pre-sampling analysis, providing a spectral representation of the signal for the mote. The mote can sample the entire spectral representation, removing the need for spectral analysis by the mote, or sample only select portions of the signal, thereby reducing the sampling requirements.

The proof-of-concept system we have presented is based upon spectral analysis, which is appropriate for acoustic and vibration sensing. However, an ASP need not be application specific. In the future, incorporating reconfigurable analog circuits [14] into the front-end will allow the system developer to use a generic front-end which is configured at run-time for the application. Our system uses a very simple classifier, but we are exploring alternative approaches. Several low-power classifiers have

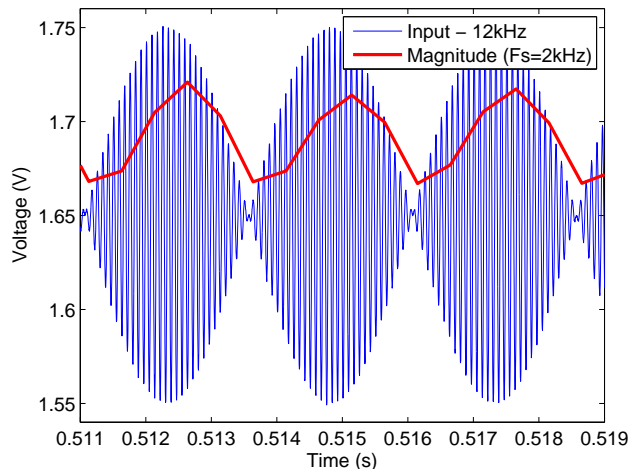


Figure 12: Performing dimensionality reduction on a high-frequency signal (for instance, only keeping the amplitude information) allows us to reduce the data rate so that it can be handled by a sensor node.

been reported [6, 35, 25].

This paper focuses on the impact of analog computational systems on WSN applications at a nodal level. In the future we would like to consider the impact at a multi-hop network level. Communication consumes a significant portion of the energy and we would like to quantify how the increased computational ability at a node translates to network-wide communication savings. This is a subject of our current research.

6. ACKNOWLEDGMENTS

We would like to thank Anvesh Singireddy for his help with the layout of the analog integrated circuit.

7. REFERENCES

- [1] American National Standards Institute. *American National Standard Specification for Octave-Band and Fractional-Octave-Band Analog and Digital Filter*, ansi s1.11-1986 edition.
- [2] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, and V. N. et. al. A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking. *Computer Networks, Special Issue on Military Communications Systems and Technologies*, 46(5):605–634, July 2004.
- [3] A. Arora and R. Ramnath. ExScal: Elements of an Extreme Wireless Sensor Network . In *The 11th International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 102–108, 2004.
- [4] M. Duarte and Y.-H. Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 2004.
- [5] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *Information Processing in Sensor Networks (IPSN)*, 2006.
- [6] R. T. Edwards and G. Cauwenberghs. Mixed-mode correlator for micropower acoustic transient classification. *IEEE Journal of Solid-State Circuits*, 34(10):1367–1372, Oct 1999.
- [7] J. Frigo, V. Kulathumani, S. Brennan, and E. Raby. Sensor network based vehicle classification and license plate identification system. In *International Conference on Networked Sensing Systems (INSS)*, 2009.
- [8] R. Genov and G. Cauwenberghs. Kerneltron: support vector “machine” in silicon. *Neural Networks, IEEE Transactions on*, 14(5):1426–1434, Sept. 2003.
- [9] D. G. Goldberg, A. G. Andreou, P. Julian, P. O. Pouliquen, L. Riddle, and R. Rosasco. A wake-up detector for an acoustic surveillance sensor network: Algorithm and VLSI implementation. In *Proceedings of IPSN’04*.
- [10] D. W. Graham. *A Biologically Inspired Front End for Audio Signal Processing Using Programmable Analog Circuitry*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2006.
- [11] D. W. Graham, P. Hasler, R. Chawla, and P. D. Smith. A low-power, programmable bandpass filter section for higher-order filter applications. *IEEE Transactions on Circuits and Systems I*, 54(6):1165–1176, June 2007.
- [12] D. W. Graham, P. D. Smith, R. Chawla, and P. Hasler. A programmable bandpass array using floating-gate elements. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 1, pages I-97 I-100, Vancouver, BC, Canada, May 2004.
- [13] B. Greenstein, C. Mar, A. Pesterev, S. Farshchi, E. Kohler, J. Judy, and D. Estrin. Capturing High-Frequency Phenomena Using a Bandwidth-Limited Sensor Network. In *ACM SenSys*, 2006.
- [14] T. Hall, C. Twigg, J. Gray, P. Hasler, and D. V. Anderson. Large-scale field-programmable analog arrays for analog signal processing. *IEEE Transactions on Circuits and Systems I*, 52(11):2298–2307, November 2005.

- [15] R. Harrison and C. Charles. A low-power low-noise CMOS amplifier for neural recording applications. *Solid-State Circuits, IEEE Journal of*, 38(6):958–965, June 2003.
- [16] P. Hasler and D. V. Anderson. Cooperative analog-digital signal processing. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 3972–3975, Orlando, FL, May 2002.
- [17] P. Hasler, P. D. Smith, D. W. Graham, R. Ellis, and D. V. Anderson. Analog floating-gate, on-chip auditory sensing system interfaces. *IEEE Sensors Journal*, 5:1027–1034, October 2005.
- [18] T. He, S. Krishnamurthy, and J. Stankovic. VigilNet: An Integrated Sensor Network System for Energy-Efficient Surveillance. *ACM Transactions on Sensor Networks*, 2(1):1–38, 2006.
- [19] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.
- [20] M. Kruiskamp and D. Leenaerts. A CMOS peak detect sample and hold circuit. *IEEE Transactions on Nuclear Science*, 41(1):295–298, Feb 1994.
- [21] M. Kucic, J. Dugger, P. Hasler, and D. Anderson. Programmable and adaptive analog filters using arrays of floating-gate circuits. In *Proceedings of the 21st Conference on Advanced Research in VLSI*, page 148, Atlanta, GA, March 2001.
- [22] B. Malhotra, I. Nikolaidis, and J. Harms. A simple vehicle classification framework for wireless audio-sensor networks. *Journal of Telecommunications and Information Technology*, pages 43–50, Jan. 2008.
- [23] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. Paradiso. Cargonet: A low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asynchronous monitoring of exceptional events. In *ACM Sensys*, 2007.
- [24] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling. IMOTE2: Serious Computation at the Edge. In *Wireless Communications and Mobile Computing Conference (IWCMC)*, 2008.
- [25] S. Peng, P. Hasler, and D. V. Anderson. An analog programmable multidimensional radial basis function based classifier. *IEEE Transactions on Circuits and Systems I*, 54(10):2148–2158, Oct 2007.
- [26] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *Information Processing in Sensor Networks (IPSN)*, 2005.
- [27] S. Jevtic and M. Kotowsky and R. Dick and P. Dinda and C. Dowding. Lucid dreaming: reliable analog event detection for energy-constrained applications. In *IPSN*, pages 350–359, 2007.
- [28] R. Sarpeshkar. Analog versus digital: Extrapolating from electronics to neurobiology. *Neural Computation*, 10:1601–1608, 1998.
- [29] R. Sarpeshkar, R. F. Lyon, and C. A. Mead. An analog VLSI cochlea with new transconductance amplifiers and nonlinear gain control. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 3, pages 292–296, Atlanta, GA, 1996.
- [30] R. Sarpeshkar, C. Salthouse, J.-J. Sit, M. Baker, S. Zhak, T.-T. Lu, L. Turicchia, and S. Balster. An ultra-low-power programmable analog bionic ear processor. *Biomedical Engineering, IEEE Transactions on*, 52(4):711–727, April 2005.
- [31] T. Serrano-Gotarredona, B. Linares-Barranco, and A. Andreou. A general translinear principle for subthreshold MOS transistors. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 46(5):607–616, May 1999.
- [32] V. Raghunathan and C. Schurgers and M. Srivatsava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, Mar. 2002.
- [33] A. van Schaik, E. Fragnière, and E. Vittoz. Improved silicon cochlea using compatible lateral bipolar transistors. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 8*, pages 671–677, Cambridge, MA, 1996. MIT Press.
- [34] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, R. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *ACM Sensys*, 2004.
- [35] T. Yamasaki and T. Shibata. Analog soft-pattern-matching classifier using floating-gate MOS technology. *IEEE Transactions on Neural Networks*, 14(5):1257–1265, Sept 2003.