Chapter 1

Introduction

1.1 Introduction

In the late 1980's, as the computer revolution was getting into full swing, Carver Mead wrote a book called *Analog VLSI and Neural Systems* [1] espousing a view of analog circuit design which was different from the generally accepted norm (which might be described as "better amplifiers and D/A converters"). The underlying premise is that biological systems outperform electrical/electronic systems by many orders of magnitude in terms of both compactness of design and consumption of power resources. Loss of speed is compensated by massive parallelism of architecture, and precision is not compromised. Indeed, the raw computational power and often precision [4] of many biomechanical and neurological systems is unparalleled.

The reasoning behind the dichotomy in power dissipation between biological and electronic computers is relatively simple. At one extreme, biological systems compete for resources and the forces of natural selection drive them towards the extremes of processing power (smarter is better) and energy efficiency (the least energy efficient creatures tend to be the first to starve). These requirements are strongly interdependent, as the process of thinking requires considerably more energy than, for instance, being a potted plant. But there are obvious competitive advantages to being perceptive of one's environment, perceptive of one's condition, mobile and, at the top of the heap, rational. But regardless of the tradeoff made between processing power and energy efficiency, natural selection always ensures that in biological systems, one is continuously minimized with respect to the other. Small size and compact design is a variable closely related to power efficiency, as larger organisms consume more resources, but larger bodies support larger brain cases housing more powerful brains.

A digital system has what is known as "restoring logic." The system is powered by positive and negative (ground) supplies which also represent the ideal logic values. Whenever a digital operation occurs, a node capacitance is forced to the value of one of the power rails by charging or discharging completely. At a glance, there is seemingly little difference between the discharging capacitor of a digital operation and the rush of current through a sodium channel in a spiking neuron. The difference is one of method, scale, dimension, and material. In the end, it is the material of which electronic circuits are made that determines the ultimate constraints on scale and dimension. As integrated electronic devices get smaller, power consumption per operation does drop by leaps and bounds. But the operation of digital circuits demands that each circuit be able to maintain two distinct states, and the particular nonlinearity of the transistor (exponential for bipolar devices and also for MOS devices below the threshold voltage) requires a voltage margin to keep the binary states distinct. Attempts to lower the transistor device threshold only shorten the voltage margin by shifting the exponential curve downward, decreasing the dynamic power dissipated but increasing the static power dissipated. For a circuit of a given size (number of transistors) and frequency, there is an optimum threshold and power supply voltage (there is still much room for improvement in today's digital fabrication processes), which minimizes power consumption, but the fundamental limit is imposed by the physical properties of the silicon p-n junction itself, and the power savings cannot be continued indefinitely. For silicon circuits to get power efficiency above this limit, a difference in method is required. Analog processing is one way to go about it. If the system is continuous-valued and not binary, then there is no need for restoring logic and the system can operate at rates of power dissipation which are orders of magnitude below those of digital systems. Of course there is a tradeoff in moving to low-power in the analog domain, and that is the presence of noise (thermal noise, shot noise, and $1/f$ noise), nonlinear distortion, mismatch, drift, slewing, parasitic inductive and capacitive coupling, and temperature dependence. These unwanted effects have to be addressed in every circuit design.

Today, however, as digital systems become increasingly fast and densely packed and decreasingly expensive (though expense reflects more the economy of scale than the cost of production or operation), it is necessary to choose analog applications wisely. There is no point in creating an analog system which will be outperformed by an equivalent digital (microprocessor or DSP) system in a year or two when digital fabrication processes move to the next smaller feature size and system designs move to a lower power supply standard. The incredible advances of CMOS fabrication processes (as per Amdahl's Law) [7] remain virtually unchecked as yet by their fundamental limits. These advances and the prospect of their continuation have the tendency to reduce the subject of

analog design to the margins, and even that typically meaning operational amplifier design, and A/D (analog-to-digital) conversion. The modern engineer deals with the real world through a transducer and an A/D, after which the sky's the limit on signal processing possibilities. Except for simple CMOS imagers and some newer applications based on capacitive sensors, transducers are the realm of Microelectromechanical systems (MEMS) research and tend to require complicated or exotic fabrication processes that are usually incompatible with analog CMOS or BiCMOS circuit layout. Thus the sought-after application of "smart" analog-to-digital conversion, with sophisticated analog processing occurring at the source on the analog signal transducer, remains elusive.

On the other hand, the problem of power consumption and heat dissipation works against the prospect of digital systems increasing in size and complexity without bound. In addition, ultra-low power circuits are becoming increasingly necessary as mobile applications increase in popularity and use, especially in consumer electronics (mobile phones, laptop computers, GPS receivers), but also in space electronics (micro-satellites, deep-space probes) military electronics, and medical electronics (pacemakers, hearing aids). Power constraints determine the difference between a battery life of hours and one of days, or one of months *vs.* one of years, or can mean the difference between a device which can operate exclusively off of solar power and one which needs alternate sources of power. Micropower electronics keep digital watches running for years; this kind of performance should and will be demanded of many more applications. Even for non-mobile applications, power regulation and heat dissipation become major problems when electronics begin consuming dozens of watts, forcing designers to find ways to cut back on power use.

Analog VLSI designers often are able to overcome some of the drawbacks of the power-hungry digital domain by emulating biological processes in electronic circuits which make optimal use of the analog domain. Mimicking neurological processes in electronics is sometimes referred to as "neuromorphic engineering" [2], a term which can be applied to processes on the level of individual neurons up to entire systems based on psychophysical data, and which is irrespective of the medium of implementation, be it analog custom integrated circuits, DSPs, microcontrollers, or computer software. There is a seemingly endless debate as to whether the human brain is most accurately described as a digital or analog signal processing system (it contains elements of both), and while it is possible that quasi-digital, spike-based processing may be the ultimate solution to approaching the efficiency and processing power of biological neural systems, as of now only specially designed analog systems come close to this goal. Neuromorphic systems have the capability to couple massively parallel computation with low power consumption and a large degree of robustness in the presence of both noise and mismatch or failure of components. Achieving this goal is a

daunting but often greatly rewarding endeavor. This thesis explores one small corner of the space of neuromorphic engineering design: I hope that in addition to educating the reader, it conveys a sense of the scope of possibilities contained in the field, and the sense of excitement and exploration felt by those whose research touches the neuromorphic engineering community.

I have directed my research, which began while I was in the Masters degree program at Stanford, towards investigating methods in auditory signal processing which lend themselves to implementation in analog hardware, specifically large-scale fabrication in silicon. A general goal of all my projects, and of the research groups of which I have been a part, has been to examine systems which are not easily implemented in a digital environment due to size, scalability, or power constraints, and find systems in the analog or mixed-mode domain which perform the same task at low power and high robustness without compromising performance. Typical methods include using current-domain subthreshold circuits for ultra-low power operation [1, 3], and using translinear circuits to compute many arithmetic functions of current and voltage in a minimal area [34]. These methods, along with thoughtful system design, make efficient use of space on silicon. In order to compare analog and digital systems fairly, it is critically important not to stop at merely simulating these systems, but to fabricate and test them as well. Each system described in this thesis has been put onto silicon and characterized by measurements made in the laboratory.

My contributions to the field of electrical engineering and associated fields include:

- An analog architecture for implementing the Continuous Wavelet Transform using the method of complex demodulation, and the use of a cascade of lowpass filters in conjunction with complex demodulation to realize a Gaussian-shaped bandpass filter [17].

- A mixed-mode architecture which used oversampling methods to perform linear analog multiplications [20].

- A mixed-mode architecture for a Continuous Wavelet Transform processor, demonstrating the use of the oversampling method for sine wave generation and complex modulation and achieving low power consumption [18, 21].

- A method for synthesis of log-domain filter circuits, especially as relates to audio-frequency applications [48].

- Circuit architectures for current-mode audio-frequency filterbanks using log-domain filter techniques [49].

- A mixed-mode architecture for implementing efficient template correlation, especially for use in conjunction with the frontend filterbank for acoustic transient classification [55, 56, 57].
- Analysis and optimization of acoustic transient classification algorithms using template correlation methods, and investigation of training methods for such classifiers [58].
- Fabricated VLSI circuits for each of the indicated architectures, including results and measurements which are detailed in this thesis, as well as test hardware and software which is beyond the scope of the thesis to describe, but which has been made publicly available and has been of benefit to many others.

Due to the variety of material encountered in my research, I have organized this dissertation into four major topics:

1. Chapter 2. Mapping the time-frequency plane with efficient analog and mixed-signal computation: The Continuous Wavelet Transform Processor.
2. Chapter 3. Continuous-time, current-mode circuits for acoustic-band filterbanks: Log-Domain Circuits, Filter Design and Synthesis.
3. Chapter 4. Algorithms, architectures, and mixed-signal circuits for pattern recognition using template correlation: The Acoustic Transient Classifier.
4. Chapter 5. Methods for training the acoustic transient classifier, extensions and directions of the research.

1.2 Mapping the Time-Frequency Plane

The time-frequency plane is a powerful concept in signal processing [8, 13, 15]. All acoustic signals can be represented as functions of both time and frequency; speech recognition and acoustic pattern classification systems require that acoustic information be mapped into a function of both time and frequency. In Chapter 2, I summarize the problem of mapping the time-frequency plane, and show how different mappings can yield different information about acoustic signals.

A signal can be divided into discrete units (samples) which adequately describe the signal; *i.e.*, the original signal can be reconstructed from its component units. This is true only if all samples cover the total area of interest (total bandwidth and timespan of the signal) in the time-frequency plane, where *coverage* is determined by the area of uncertainty of each sample,

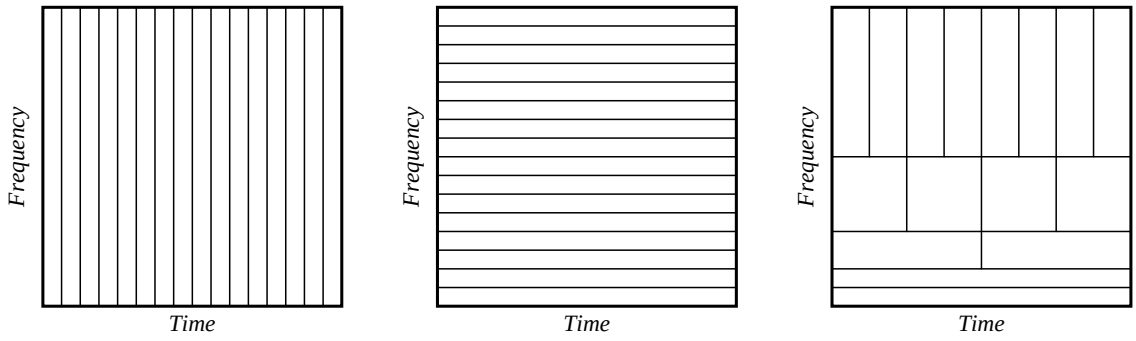
$$\Delta f \Delta t \geq \frac{1}{2}, \tag{1.1}$$

the acoustic corollary of Heisenberg's uncertainty principle in which matter waves are replaced by acoustic pressure waves. Every sampled output of an acoustic signal-processing system can be viewed as covering a small area in the time-frequency plane. The representation of the area of support of a system as a rectangle with width Δt and height Δf is an idealization of real systems which may be primarily localized in a small area of time-frequency space but spread well beyond the boundaries. The Gaussian is in fact the only function satisfying the minimum $\Delta f \Delta t$ area [8], yet it is nonzero to infinity: a true Gaussian is in fact noncausal, but causal approximations to the Gaussian profile have near-minimum area.

The two most common divisions of the time-frequency plane are the *Dirac map* and the *Fourier map* [15], shown in Figures 1.1(a) and (b). The Dirac map corresponds to a sampled signal, and the area covered by each sample is the frequency spread times the sample period. The Fourier map comes from the Fourier transform, which divides the frequency plane into even parts. The non-windowed Fourier transform assumes an infinite timespan for the signal, and therefore does not have minimum area samples in the time-frequency plane.

Each of these methods yields different information: The Dirac map yields the *best* time resolution but the worst frequency resolution; the Fourier map achieves the best frequency resolution at the expense of having the worst time resolution. Overview: Continuous wavelet decomposition

This view of the time-frequency plane naturally leads to ideas of ways to create time-frequency maps which achieve the optimal tradeoff between time and frequency resolution. One of these is the *wavelet map* (Figure 1.1(c)) [15]. An important property of the wavelet map is that the



(a) Dirac (b) Fourier (c) Wavelet

Figure 1.1: Some mappings of the TF plane.

filter center frequencies f_c are spaced exponentially, with the bandwidth of each filter proportional to its center frequency. If we use a simple filterbank of bandpass filters to generate the wavelet map, then to adequately capture the output in a sampled-data system we must sample each filter channel at twice (*i.e.*, the Nyquist rate of) the upper cutoff frequency of its transfer function. However, in this architecture the upper cutoff is typically twice the value of the bandwidth, which results in an output bandwidth which is larger than the input bandwidth. The wavelet decomposition, however, describes a *transform*, so no information has been added to the system and it should not be necessary to increase the bandwidth of the system. Preferably, the system should shift the frequency content of each channel toward zero, allowing it to be sampled at the Nyquist rate corresponding to the channel bandwidth rather than the channel upper cutoff frequency. Thus the total output bandwidth is equal to the input bandwidth.

We adopted the method of shifting the center frequency f_c of the band to zero using *complex demodulation* [9]. The result of this process is two orthogonal outputs each of which can be sampled at twice the lowpass filter cutoff frequency. Re-modulation of the two outputs by the same process is equivalent to bandpass-filtering the original signal. We can approximately reconstruct the original input by applying this manipulation to each filterbank channel output and adding them together.

The Gaussian function has the most compact representation in the time-frequency plane, and so it forms the core filtering process of the Continuous Wavelet Transform processor. An approximation to a half-Gaussian-shaped lowpass function is quite easy to create using a series of

simple lowpass filters in cascade: the total transfer function converges to a Gaussian function as the cascade length is increased.

We have designed and fabricated a series of analog and mixed-signal chips to perform the wavelet transform. Two major obstacles to overcome were determining 1) how to generate a sine wave with low harmonic distortion and 2) how to implement a highly linear multiplier. We developed a robust system by using a hybrid analog and digital architecture [18, 20, 21]. We solved both problems of the original design by using an *oversampled binary representation* of the modulating sine wave [20].

1.2.1 Current-mode filterbanks

It is not always necessary to preserve phase information in order to perform acoustic recognition and classification tasks. In such cases a simple filterbank of bandpass filters may suffice, with the outputs rectified and smoothed to yield an estimate of signal energy in a particular band at a specific instant in time. In Chapter 3, I consider the problem of designing such a filterbank using *current-mode* circuits as a frontend processor for the acoustic recognition system described in Chapter 4. The filter transfer functions in a current-mode filterbank are the ratio of output to input *current* rather than voltage. I chose a current-mode architecture partly because it is convenient to interface to the backend systems, which are also current-mode circuits, but more importantly because current-mode filters have a larger dynamic range (on the order of 10^6) than voltage-mode circuits which have similar power-consumption and layout area. Subthreshold analog voltage-mode filterbanks have been well-researched and well-developed [1, 5, 51], but remain subject to low dynamic range.

Current-mode filterbanks, on the other hand, are only beginning to be extensively investigated. Recent work on the relatively new subject of *log-domain filters* [39, 47] has provided some circuit designs and methods for current-mode filters. Working with circuit ideas from Philippe Pouliquen and Wolfgang Himmelbauer, I developed a new method of generating log-domain filter circuits given a transfer function, and fabricated some low-power BiCMOS designs on a test chip which proved to implement the filter functions correctly. Figure 3.11 shows one bandpass filter design.

A useful aspect of log-domain designs is that filter parameters such as center frequency f_0 and resonance Q are controlled directly by bias currents, permitting the same circuit to be used for each filter of the bank, with bias currents appropriately scaled for each one.

1.2.2 Acoustic Transient Recognition

Simple patterns often can be successfully recognized (classified) by directly correlating each input pattern with a stored template: compute the distance between each input and template entry using an appropriate metric, sum the result over all components of the feature space, and compare the totals among all templates to select a winner. This method is a linear classifier encoding one separating hyperplane per template. For acoustic signal processing, the feature space is typically a time-frequency map.

Speech signals are normally too complicated to enable a successful recognition system using correlation, due to the necessity of coping with large variance in pitch, timbre, and time for the same target output. *Acoustic transients*, on the other hand, are signals which by definition occur over very short periods of time (less than approximately 1/10 second). Examples of acoustic transients are the sound of a handclap, a door closing, sonar echos, and certain events in speech which occur on sub-phonetic time scales, such as the sudden bursts and silences associated with hard consonants like 'p' and 't'. Time-frequency decomposition gives a relatively stable description of an acoustic transient from instance to instance.

Together with Dr. Fernando Pineda, at the Johns Hopkins University Applied Physics Laboratory (JHU-APL), we have experimented with acoustic transient recognition by correlating time-frequency-mapped outputs with templates formed by averaging together examples of the transients from a training set [54, 55]. In addition, we have experimented with ways of reducing the complexity of the computation and the amount of data stored in each template. The result is a system which lends itself very nicely to efficient implementation in analog hardware [57].

We begin with a baseline algorithm which is the direct correlation between an input and a template

$$c_z[\,t\,] = \sum_{m=1}^{M} \sum_{n=1}^{N} x[t-n, m]\, p_z[n, m] \tag{1.2}$$

where M is the number of frequency channels of the input, N is the maximum number of time bins in the window (enough to cover the transient event at 1 to 2 ms per bin), x is the array of input signals split into frequency bands, p_z is the matrix of template pattern values for pattern z, and t is the current time. This formula produces a running correlation $c_z[\,t\,]$ of the input array with the template z. A system implementing the correlation in this form requires an analog (or multibit, if digital) storage for $M \times N$ values for *each* template, and $M \times N$ multiplications at every time step, also for each template. The computational requirements are formidable: for example, a 10-template classifier with a 16-channel filterbank frontend, operating at 2 ms per time step and allowing approx-

imately 1/10 second (64 bins at 2 ms) per template requires over 10 million multiplies per second. This is feasible using dedicated software. However, we view the template correlation as a fundamental low-level task of potentially many kinds of acoustic recognition and classification systems, and as such it should be made as simple, compact, and power-efficient as possible.

We simplified the correlation equation from the standpoint of analog hardware implementation through a series of steps, arriving at an algorithm with requirements for memory storage reduced to a single bit per template value (thus, $M \times N$ bits per template) plus N analog values stored in a shift-and-accumulate register. The accuracy of the algorithm in classification tasks remains little affected by the sparse encoding of information in the template.

1.2.3 Analog VLSI implementation of the transient classifier

Figure 4.7 shows a block diagram of the correlator architecture we developed. The details of the circuits which implement the frontend section, normalizer, and correlator are explored in Chapter 4. Simple memory circuits such as those found in RAM chips provide the binary template storage. Currents from each channel input are switched onto N common lines depending on the template value for channel m and time n. Merging currents together performs the summation across the M channels. Two *bucket brigade devices* (BBDs), circuits usually used for analog delay lines in high-fidelity audio and other acoustic applications [60, 62], perform summation across time n. Summed currents are converted to a voltage on each bucket brigade node by integrating onto the bucket brigade capacitor for a fixed period of time, and we compute the difference between the last bucket brigade cell of the two BBDs with a simple switched capacitor circuit. The bucket brigades are fully pipelined so that for each time t the system produces one complete correlation computation at its output node.

We have made a simple estimate of power consumption based on detailed knowledge about the circuit components used in the VLSI layout. This power consumption typically should be less than 10 microwatts for each template. We expect a system with a large "vocabulary" of 200 transient events using the normal timestep of 2 ms to consume less that 2 mW of power. This system computes all correlations on all templates in parallel, and therefore does not need to employ any measures to reduce the search space of the solution; and it produces a complete correlation at every timestep, so segmentation of the input is not necessary, though it may be desired for robustness, particularly in instances where the transients are presented to the system in isolation.

I made an analysis of the performance of the algorithm in simulation under different archi-

tectures. The architecture described above represents only one of a number of different possibilities. First, any combination of input or template can be made binary or trinary. Second, although a zero-mean transformation is the best way to get a simple decision boundary for choosing a binary value, the choice of pairwise frequency channel differences is only one of many such transformations. Other possibilities include time differences between samples, center-surround computation for each frequency channel, or some combination of time and frequency differences. The architecture of Figure 4.16 was chosen as the simplest representation maintaining acceptable performance on the classification task. However, results of the analysis of different architectures showed that the use of a channel difference computation in addition to a binary representation of *both* input and template resulted in similar classification rates to the proposed mixed-mode architecture. A system with both binary inputs and templates can be implemented entirely in the digital domain (except for the front-tend processing of the input prior to binarization). The optimal form of a binary-binary correlator would be a custom digital architecture, using a parallel architecture similar to that proposed for the analog-binary correlation. The slow speed at which the system operates (1 ms per sample) allows the digital system to be operated at extremely low power. On the other hand, the speed is slow enough to allow the architecture to be completely serial, which precludes low-power operation but permits the system to be built with discrete IC parts. A system based on discrete parts and field-programmable gate arrays (FPGAs) can be designed and built within the space of about two months at a cost similar to that of a custom chip. I designed and constructed a digital ATP system based on a trinary-trinary correlation method (an extension of the binary-binary case), described in detail in Chapter 4, both as a way to have a working real-time version of the algorithm, and as a way to directly compare area, power consumption, and other critical properties between the digital and the mixed-mode systems.

1.2.4 Overview: Learning and Continuous Speech Recognition

While Chapter 4 presents in detail the algorithms and architectures which perform robust acoustic transient classification, they do not cover the subject of how the templates are generated, which is part of the broad topic of *machine learning*. Chapter 5 takes a look at the average-value method used to train the classifier in simulation. The baseline algorithm calls for simple learning of template values by aligning and averaging examples from the training dataset. The method requires a segmentation algorithm to detect transient events in the input and determine the point at which the correlation output should be examined for a classification result. More intelligent methods

of template generation can eliminate the need for a segmenter, allowing the correlation system to operate continuously and increasing system robustness to situations such as overlapping transient events. In Chapter 5, we investigate training methods based on information-theoretic and neural-network-like methods. Another step is to incorporate a learning algorithm into the system so that template values can be trained by on-chip adaptation. In this thesis, however, I will only consider the intermediate step of "chip in the loop" training, in which a computer evaluates the template adaptation based on correlation results of the hardware system itself.

Currently, the most successful and most widely used method for automated speech recognition is the Hidden Markov Model (HMM) [65]. Systems based on Hidden Markov Models must incorporate many intricate strategies to reduce the search space in order to make computation times reasonable. Digital systems incorporating large-scale parallelism can investigate many possibilities in real time, but are bound to be power-consuming, especially as real-time speech recognition systems constantly push the limits of available technology. The human brain, however, remains the ideal against which all of today's state-of-the-art systems are measured, and it succeeds at this amazing task using a tiny fraction of the power of the digital systems which it unfailingly outperforms. The potential gains of realizing speech processing systems in analog hardware make investigating them worthwhile. Template correlation algorithms and architectures have the potential to provide a fundamental layer of processing upon which a speech processing system may be built. The thesis concludes with a look at two different approaches to achieving this goal. The first is an architecture proposed by Unnikrishnan, Hopfield, and Tank [68, 69]. Like the acoustic transient processor, the algorithm was developed with analog hardware implementation in mind. Also like the acoustic transient processor, it is based on template correlation. It is presented as a neural network system, but after a small amount of algebraic manipulation, one can arrive at a description which bears strong resemblance to the template correlation equations. The main difference between the two is that the Unnikrishnan *et al.* system is a continuous-speech digit recognizer, able to cope with some of the problems of complex signals of long duration, but not requiring the solution of complicated syntactic and semantic issues faced by more general speech recognition systems. In simulation, the system achieves accuracies of 98 to 99% on the TIDIGITS database. Its requirements of the hardware are formidable, however, which is one reason that the system has never been implemented outside of computer simulation, where it runs very slowly and inefficiently due to the analog nature of the design.

The second approach is a biological model of auditory processing in the human brain, based on physiological data about the neural connections in the auditory cortex [70]. There are areas

of the auditory cortex which encode maps of the time-frequency domain. Certain neurons in this region of the brain respond strongly to wavelet-like regions in time-frequency space, of differing size, scale, and angular orientation. Thus they are more efficient than the wavelet processor at capturing specific auditory events. It is likely that they encode the independent components of all auditory events which are meaningful to humans. The wavelet-like regions of response include areas of excitation and other areas of inhibition. The complex responses of the neurons are a result of many synaptic connections and are not easy to describe in neural network terms. However, they can be easily approximated by templates. A trinary encoding scheme such as that used by the digital transient processor allows encoding regions of excitation as positive bits and regions of inhibition as negative bits. Zero values mask out the regions of little or no response and fill in the corners of the artificially rectangular template, leaving elliptical areas like those encountered in auditory cortex neurons. Although the binary inputs and templates are only an approximation to the information-rich spike trains delivering auditory information through the brain and the continuous-valued response of the neurons, the template correlator should be able to provide additional insight into the processing of auditory signals in the brain, and may shed light on how these complicated feature detectors in the brain can work together to enable the robust sound and speech recognition of which the brain is so magnificently capable.