A LOW-POWER SYSTEM FOR AUDIO NOISE SUPPRESSION: A COOPERATIVE ANALOG-DIGITAL SIGNAL PROCESSING APPROACH

David Anderson, Paul Hasler, Rich Ellis, Heejong Yoo, David Graham, Mat Hans

School of Electrical and Computer Engineering, Georgia Institute of Technology
777 Atlantic Dr., Atlanta, GA 30332-0250

ABSTRACT

In this paper, we introduce the concept of cooperative analog-digital signal processing, and its application in the development of a real–time noise suppression system. The algorithm implemented is designed to reduce stationary background noise while preserving the non–stationary signal component. The algorithm is based on digital signal processing foundations that are slightly adjusted for use in the continuous-time domain. Because the system relies on analog computation rather than digital, it has benefits such as extremely low power consumption and real-time computation. The analog circuit elements are based on new floating-gate circuits that are small, efficient, and programmable—making it possible to set and tune bias points, offsets, and filter parameters under digital control.

1. COOPERATIVE ANALOG-DIGITAL SIGNAL PROCESSING

New advances in analog VLSI circuits have made it possible to perform operations that more closely reflect those done in DSP applications, or that are desired in future DSP applications [1, 2, 3, 4, 5, 6, 7]. Further, analog circuits and systems can be *programmable*, reconfigurable, adaptive, and at a density comparable to digital memories (for example, 100,000+ multipliers on a single chip) [8, 9, 10, 11, 5]. These properties have been almost exclusively associated with digital processing, but the addition of small, dense, programmable analog circuits provides a framework in which to create cooperative analog/digital signal processing systems that benefit from both approaches to make something better than the sum of its parts. We define cooperative analog-digital signal processing (CADSP) as the research field using combinations of programmable analog signal processing and digital signal processing techniques for real-world processing.

Neither analog signal processing nor digital signal processing can exist in current technologies without the other; that is, real-world signals are analog while much of the modern control and communication is digital. Therefore, a primary question is where to partition the analog–digital boundry to enhance the overall functionality of a system by utilizing analog/digital computations in mutually beneficial way (Figure 1). CADSP allows more freedom of movement for the partition between the analog and digital computation. CADSP is a superset of mixed-signal research in that it focuses heavily on algorithms as well as circuit implementation. By adding functionality to our analog systems, we enhance the capabilities of the controlling digital system, and therefore, the entire product under consideration. A full discussion of this partition problem can and will encompass several research papers. The range of applications for these approaches reaches from auditory and speech processing, to beam-forming, multidimensional signal processing, and radar computations, communications processing, and image processing and recognition.

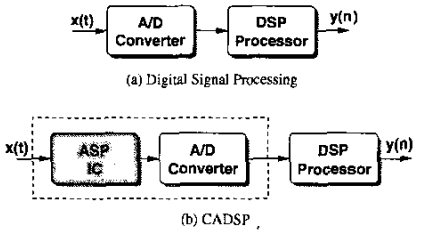


Fig. 1. Illustration of the tradeoffs in cooperative analog/digital signal processing. We assume the typical model of signals coming from real-world sensors, which are analog in nature, that need to be utilized by digital computers. The inverse problem, digital signals going to real-world actuators, is similar in nature. One approach is to put an analog-to-digital (A/D) converter as close to the sensor signals as possible, and allow the remainder of the computations to be performed digitally. An alternate approach is to perform some of the computations using analog signal processing, requiring simplier A/D converters, and reducing the computational load of resulting digital processors. One could group this analog computation and A/D converter as a specialized A/D converter that gives more refined information (Fourier coefficients, phonemes, etc.) than a literal map of the incoming signal. The question of where to put this boundry line strongly depends upon the particular requirements of an application.

The following sections briefly discuss CADSP advantages and disadvantages followed by a discussion of the development and implementation of a background noise suppression system based on the CADSP approach.

2. THE CASE FOR COMBINING ANALOG AND DIGITAL SIGNAL PROCESSING

One might wonder why introduce analog signal processing at all, since the current framework of immediately digitizing incoming signals, illustrated as the top half in Fig. 1, seems to be working well in current practice. As more computational power is required to implement more digital functionality, the needs have largely been met by transistor scaling and the advantages flexibility in programming. However, current trends are, and have been, finding unique challenges that provide an opportunity for a new perspective. Several factors suggest using analog signal processing in conjunction with digital systems to meet these challenges:

1. power consumption / efficiency requirements,
2. A/D converter requirements,
3. size constraints, and
4. problem "fit."

The analog VLSI signal processing circuits are programmable and adaptive using floating–gate circuits. The programmable and adap-

0-7803-8116-5/02/$17.00 ©2002 IEEE.

tive aspects of floating–gate analog VLSI are significant and make practical a variety of signal processing applications. One system that holds promise for the implementation of a large class of signal processing application in analog VLSI is the field programmable analog array presented in [12].

Power consumption in DSP microprocessors, as measured in mW/MIPS, has been decreasing by half about every 18 months—a phenomena known as Gene's law [13]. This trend of decreasing power consumption has been keeping pace with Moore's law and has helped make possible the increasing proliferation of portable electronics. Even so, device functionality, and the amount of signal processing, is often primarily constrained by a fixed power budget. A custom analog approach can often achieve an increased efficiency (= Bandwidth·Power) of a factor of 10^4 over a custom digital approach [14]. If Gene's law continues to hold, digital systems will catch up to current analog efficiencies within about 20 years! Therefore, migrating some applications to analog processing could provide up to a 20 year jump in functionality relative to a purely digital roadmap. The greatest efficiency occurs when matching physics of problem to physics of silicon medium. This increase in efficiency can be used for reducing the power efficiency of a given problem, or addressing computational problems that are considered *intractable* by the current digital roadmap.

Analog to digital (A/D) converters also can be a limiting factor in signal processing systems and A/D requirements are becoming an increasingly large part of the system design constraints. The system demands on many current systems require very high resolution / high performance A/D converters; the resulting A/D converter block is often consuming a large fraction of the power budget, as well as system design time. While digital processor efficiency may be increasing rapidly, partially as a result of transistor scaling; scaling is not helping as much for A/D converters. A/D converters have roughly been increasing resolution at 1.5 bits / 5 years at the same performance, and quickly running into additional physical limits which might further slow this progress.[1] In many problems the need for enormous speed and resolution originates in recovering "low-information" signals over a wide dynamic range or in a *noisy* background (e.g. CMOS imaging, software radio). By utilizing analog signal processing at the front-end, we can significantly reduce the A/D converter complexity, and the overall system complexity.

Circuit size constraints also favor analog VLSI circuits in many cases and it is often possible to take advantage of device physics to perform complex operations with only a very few transistors. For example, an analog multiplier can store the coefficient and perform the multiplication using only as many transistors as would be needed to store a four bit coefficient in digital memory. Besides multiplication there are many elemental operations that can be efficiently implemented in floating–gate analog VLSI. These include filters, adaptive multipliers, a large range of non-linear functions, decision circuits, and others (see also [5, 11]).

3. NOISE SUPPRESSION SYSTEM

The recent prosperity of portable computing and communication devices has resulted in a renewed interest in audio signal enhancement by suppressing additive background noise from corrupted noisy signal. While most noise suppression methods focus on processing sampled audio signals, Because noise suppression is typically performed before other processing such as compression, transmission, or storage, it fits well into the CADSP framework shown in Fig. 1 [15, 16]. Therefore, we present here an analog noise suppression system as a demonstration of a typical CADSP application.

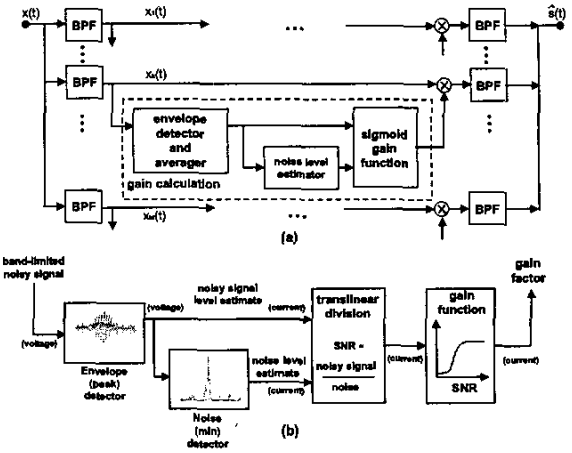


Fig. 2. (a) Block diagram of continuous-time noise suppression system. The center frequency of filter bank is spaced exponentially. At each sub-band, gain is calculated from non-linear gain function. To achieve consistent noise suppression for different noise variance, noisy signal estimate, instead of signal estimate, is used as an input of non-linear gain function. The gain is then multiplied with sub-band signal and summed to build full-band signal estimate. (b) Details of the sub-band gain calculation—Within each frequency band, the noisy signal envelope is estimated using a peak detector. Based on the voltage output of the peak detector, the noise level is estimated using a minimum detector. The currents representing the noisy signal and noise levels are input to a translinear division circuit, which outputs a current representing the estimated signal–to–noise ratio. A non-linear function is applied to the SNR current, and the resulting gain factor is multiplied with the band–limited noisy signal to produce a band–limited "clean" signal. Finally, the output of all of the bands are summed to reconstruct the signal with the noise components significantly reduced.

We assume a noisy signal, $x(t)$, consisting of a signal, $s(t)$, corrupted by additive noise, $n(t)$, that is uncorrelated with the signal

$$x(t) = s(t) + n(t). \tag{1}$$

The goal is to design a real–time system that generates some optimal estimate, $\hat{s}(t)$, of $s(t)$ from $x(t)$. We assume that the additive noise is stationary over a long time period relative to the short-term stationary patterns of normal speech. The signal estimate, $\hat{s}(t)$, may be found in the frequency domain using spectral subtraction or by applying a Wiener filter gain. The basic Wiener gain can be expressed in terms of the frequency–dependent signal–to–noise ratio (SNR) as

$$H(\omega) = \frac{\Gamma^2(\omega)}{1 + \Gamma^2(\omega)}, \tag{2}$$

where $\Gamma^2(\omega) = \Phi_s(\omega)/\Phi_n(\omega)$ with $\Phi_s(\omega)$ and $\Phi_n(\omega)$ representing the power spectral densities of the signal and noise respectively.

Frequency domain processing is accomplished using a one-third octave filter bank. The filter bank separates the noisy signal into narrow-band signals. In each band, the envelope of the noisy signal is detected and smoothed. From the smoothed sub-band signal envelope the noise envelope is estimated in each sub-band.

[1] The A/D converter limitations on a power constrained systems is somewhat similar to the problems generated as processor speeds progressed much faster than memory speeds in recent years.

The SNR in each band is estimated from the noisy signal and noise envelopes. A non-linear (sigmoid) gain function is used to approximate the Wiener gain. Finally, the original band-limited signal in each band is multiplied by the respective gain and the result is summed to construct the full-band "clean" signal estimate. The overall structure of the system is shown in Fig. 2(a) with a more detailed view of the gain calculation block of a single band shown in Fig. 2(b).

4. IMPLEMENTATION

This section describes the implementation of the noise suppression algorithm on a 0.5 micron CMOS VLSI chip. The gain calculation circuits make up the processing block for the system. Each block operates independently of the others in the array, so effectively we have 32 parallel signal processors operating simultaneously on 32 band-limited signals. The block diagram shown in Fig. 2(b) summarizes the gain calculation algorithm. The following sections will elaborate on the algorithm and relate the circuits that perform the underlying functions. Note that the analog systems described here are largely based on an analog floating gate circuit technology which allows each circuit to be programmable/tunable. The result is that the individual circuit elements can be very small since they do not require the extensive overhead of building perfectly matched circuits or of designing circuits tuned via external inputs.

4.1. Frequency Decomposition

The first structure on the left side of Fig. 2(a) is a filter bank that separates the noisy signal into 32 bands that are logarithmically spaced in frequency, similar to the human auditory system, for frequency domain processing. This results in approximately one-third octave spacing in the cut-off frequencies of the filter bank. By using one-third octave filters, any frequency distortions—whose bandwidth is on the order of the bandwidth of the each band—also lie almost within the same critical band and can be minimized for the perceptual impact [17]. The filters used in the filter bank are the Capacitively-Coupled Current Conveyor Second–Order Sections (C^4 SOS), shown in Fig. 3(a). The C^4 SOS is composed of three C^4s (described in [18, 19, 20]) in which the feedback capacitor of the first stage filter is removed in order to make that stage a high gain amplifier. This filter can have a frequency response of any defined bandwidth, and outside that bandwidth, slopes of ± 40 dB/decade or greater occur. By adjusting the voltage biases, the response at either corner can be tuned to have a sharp transition or even a Q peak. A high Q peak is useful because it helps isolate the respective center frequency. More information on the details of the C^4 SOS is available elsewhere [21].

After the incoming noisy signal has been band–limited by the filter bank, a gain factor is calculated based on the characteristics of each band-limited noisy signal.

4.2. Envelope Estimation

The first step in the gain calculation is to estimate the envelope of the noisy signal. Next, the noise envelope is estimated using a technique closely related to the minimum statistics method where the noise is found using a minimum detector on the envelope of the noisy signal. The SNR is then estimated from the noisy signal envelope and noise envelope in each subband.

$$\hat{\Gamma}(k,t) = \frac{\hat{e}_s(k,t)}{\hat{e}_n(k,t)} \approx \frac{e_x(k,t) - \hat{e}_n(k,t)}{\hat{e}_n(k,t)} = \frac{e_x(k,t)}{\hat{e}_n(k,t)} - 1 \quad (3)$$

where k is the subband index and $\hat{e}_x(k,t)$ is the noisy signal envelope estimate that can be represented as the sum of the actual signal envelope $e_s(k,t)$ and the noise envelope estimate $\hat{e}_n(k,t)$.

4.2.1. Peak Detector

A peak detector is used to estimate the envelope of each noisy subband signal. When a speech signal is input, the peak detector will follow the envelope of the signal, rising rapidly with the increasing signal amplitude and decaying slowly enough to result in a smooth envelope. The peak detector will also follow the level of the additive noise, particularly in times where the signal is absent. The circuit outputs both a voltage and current that are representative of the noisy–signal level (envelope). Each subband contains a signal of a different bandwidth so the peak detectors each have a programmable time constant so that an appropriate smoothed envelope can be determined for each of the bands.

4.2.2. Minimum Detector

One effective method of noise estimation is the minimum statistics approach [22]—an approximation to this approach is to use an inverted peak detector or minimum detector with a long time constant on the averaged noisy signal envelope estimate. The inverted peak detector operates on the estimated envelope of the noisy signal, keeping an estimate of the minimum value which is assumed to be the noise floor. A "minimum detector" circuit is therefore used to estimate the noise level in the subband signals (i.e. the noise envelope). The input to the minimum detector is the voltage output from the peak detector. In this way, we estimate the noise level by following the minimum values of the noisy–signal envelope. A bias voltage sets the attack time constant; it is set to run much slower than the peak detector in order to follow the slow changes found in the amplitude of relatively stationary noise. When the signal is present, the output will maintain a slowly rising level; when the signal is not present, the minimum detector will track the noise level.

4.2.3. SNR Calculation

Multiplication and division operations can be performed using the translinear principle [23]. The circuit shown in Fig. 4 performs a division operation where the output, I_{SNR}, can be represented by the following equation:

$$I_{SNR} = I_{scale}\frac{I_{signal}}{I_{noise}} - I_{scale}$$

and represents the estimated SNR. The current I_{scale} is set by the bias voltage V_{scale} and is used to put the output current into the proper range for the gain function.

4.3. Non-linear Gain Function

The final elements of the gain calculation algorithm are the gain function and multiplier. The Wiener gain (Eq. 2) is widely used in noise suppression since it minimizes error energy. In the circuit implementation, the gain function and the multiplier comprise one circuit, shown in Fig. 5(a). The transistors at the top of the schematic are the differential pair where the actual multiplication takes place; the multiplier will be described shortly. The transistors below the differential pair create the behavior of the gain function. The output of the gain function circuit is I_{gain}, which can be

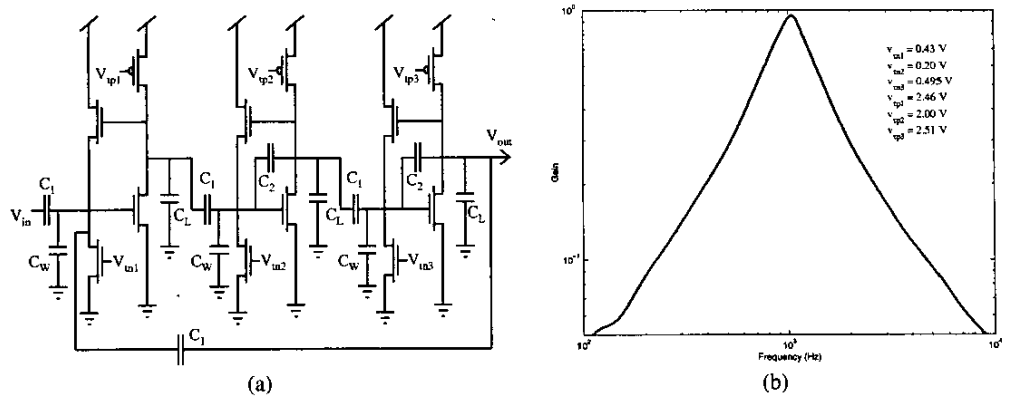


Fig. 3. **(a)** The schematic showing the C^4 SOS band-pass filter. The bias voltages are set using a resistive-divider network, which creates a logarithmically spaced filter bank. Future versions will be implemented using floating gate elements instead of resistors to allow for programming and improved tuning. **(b)** This plot shows the frequency response of a single filter in a 32-band filter bank. In this configuration, the band-pass rolloff is 40 dB. The noted bias voltages refer to the voltages set at each end of the resistive network. This data is from a circuit fabricated in a 0.5 μm process available through MOSIS.

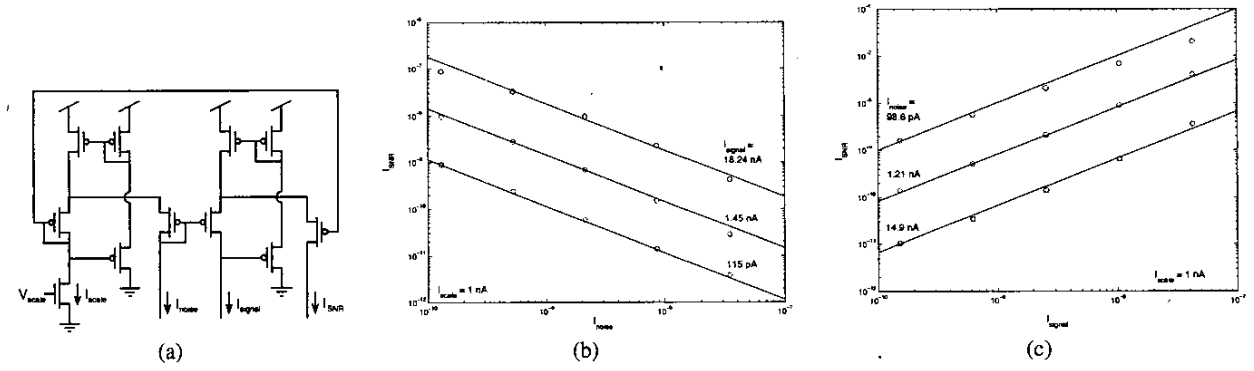


Fig. 4. **(a)** The translinear circuit shown in this schematic implements the division of I_{noise} into I_{signal}, yielding a current representing the signal-to-noise ratio, I_{SNR}. We subtract an I_{scale} from the output current, I_{SNR} to mirror the function described in Equation 3. The voltage V_{scale} sets the bias current I_{scale}, which is used to put I_{SNR} into the proper range for the gain function circuit. **(b)** & **(c)** The theoretical and measured outputs for the divider circuit using fixed signal and noise currents respectively.

approximated by

$$I_{gain} = \frac{(I_{SNR}^2 + I_{min1})I_{max}}{(I_{SNR}^2 + I_{min1}) + I_{max}}.$$

This has almost the exact form of the Wiener gain with the addition of a bias current that can be used to tune circuit operation as needed and a scaling factor. I_{max} and I_{min1} are set by the voltage biases V_{max} and V_{min1} and effectively create the upper and lower bounds of the gain factor output. The voltage bias V_{min2} can be used to further adjust the range of input current that the circuit will accept. The diode-connected transistor on the input branch of the circuit causes the current that is mirrored onto the gain branch to be approximately squared; this is important to ensure a quick transistion from low to high gain as I_{SNR} increases.

The multiplication portion of the circuit results from the interaction between the subband signal presented at the inputs as a pair[2] V_{in1} and V_{in2} and the gain factor current I_{gain} from the gain function circuit in Fig. 5(a). The circuit is biased to operate in the linear range of the following equation:

$$I_{out1} - I_{out2} = I_{gain} \tanh\left(\frac{\kappa(V_{in1} - V_{in2})}{2U_T}\right),$$

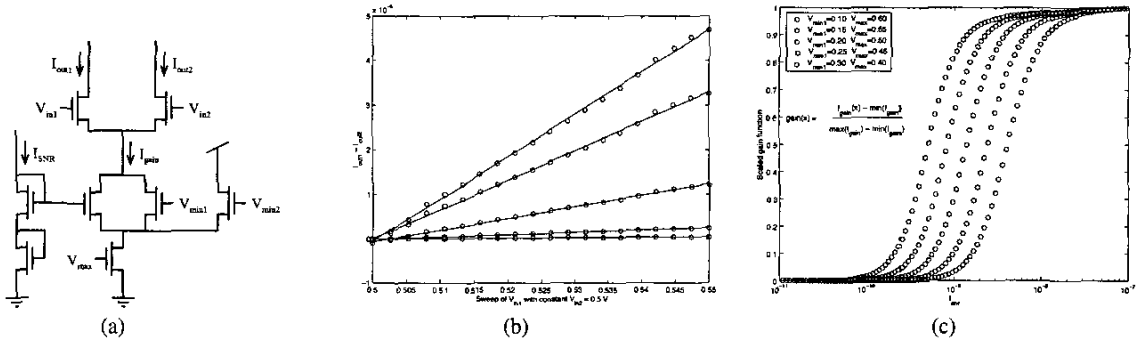
where U_T is the thermal voltage of the transistors. The input to the multiplier is the band-limited noisy signal, i.e. the output of the C^4 SOS band-pass filter. The data in Fig. 5(b) and (c) show the functionality of the multiplier and the form of the gain function.

5. SYSTEM RESULTS

5.1. Simulation

It is important to have a "signal processing simulation" of the analog VLSI system prior to fabricating the circuit. Circuit or device level simulation can provide accurate results but they take too long

[2]Because the current implementation is single-ended rather than differential, the "negative" input voltage, V_{in2}, is held constant.



(a) (b) (c)

Fig. 5. **(a)** The schematic of the multiplication and gain function circuit. The output current, $I_{out1} - I_{out2}$, is the product of the band-limited input signal, $V_{in1} - V_{in2}$, and the gain factor, I_{gain}. Each voltage bias sets a current that forms the overall behavior of the gain function circuit. **(b)** The output current, $I_{out1} - I_{out2}$, varies with its two factors: the gain factor and the differential input voltage. This plot shows sweeps of the input voltage for several values of I_{SNR}. **(c)** The gain function of the circuit depicted in (a) is plotted versus I_{snr}. At the extremes of the signal-to-noise ratio, either a low gain factor or a unity gain factor is the result. Multiple curves show the bias voltages being adjusted.

to effectively include in an iterative design process. Therefore, the initial continuous-time noise suppression algorithm was functionally simulated in Matlab. The simulated system contained the same functional blocks as shown in Fig. 2 and as implemented in the analog system with continuous–time transfer functions converted to discrete–time functions using the bilinear transform. The sampling rate in the simulator was chosen to be at least four times the required Nyquist rate to avoid the frequency warping that occurs near the Nyquist rate with the bilinear transform. Multi-rate signal processing was used to improve efficiency but the high oversampling rule was always followed. In order to anticipate accurate performance when implemented with circuits, we tried to include physical limitations as much as possible. For example, the filters were implemented as high Q second-order sections that conformed to previously measured circuit behavior for the C^4 filters.

The simulation gain function parameters were chosen so that the attenuation was limited to less than 40 dB—leaving some residual background noise. Representative output from the simuluation is shown in Fig. 6a. The perceived quality of the noise–suppressed signal is remarkably free from artifacts normally associated with Wiener filtering and spectral subtraction. We attribute this to the method of frequency decomposistion that matches human auditory critical bands, and the proportional bandwidths of the subband envelopes.

5.2. Implementation Results

The experimental results presented in this paper are from tests on individual components that have not yet been integrated into a larger system. Figure **??**b shows a noisy speech signal that has been processed by the components in our system. The system is effective at adaptively reducing the amplitude of noise-only portions of the signal while leaving the desired portions relatively intact.

The power consumption of the noise suppression circuit is very small. The core circuitry, assuming 32 frequency bands and a 3.3 V power supply, consumes less than 50 μW. Therefore, integrating this circuit into an existing system will not likely have a measureable impact on battery performance. In systems that provide signal amplification, the removal of background noise may decrease amplifier power consumption, thereby increasing the overall battery life.

5.3. Circuit Noise and Distortion

Any noise or distortion created by the gain calculation circuits minimally affects the output signal because these circuits are not directly in the signal path. While the bandpass filters and the multipliers will inject a certain amount of noise into each frequency band, this noise will be averaged out by the summation of the signals at the output of the system. Distortion in the signal path will arise from the bandpass filters and the multiplier. In the bandpass filter array, the C^4 SOS structure is not cascaded as in cochlea models, therefore there is no distortion or noise accumulation. The distortion level in each frequency band for a 30 mV single-ended input signal is 2nd harmonic limited at -30 dB at peak. A differential filterbank will eliminate 2nd harmonic distortion and reduce the 3rd harmonic level to -40 dB at peak. The distortion introduced by the multiplier is dependent on the output levels of the bandpass filters: if the signal is near 30 mV, 3rd harmonic distortion will be -20 dB down; however, if the signal is near 7.5 mV, the 3rd harmonic distortion will approach -46 dB. In speech, particularly in noisy environments, the signal is more evenly distributed across a broad frequency range than a simple tone, therefore distortion is significantly reduced.

6. CONCLUSIONS

This paper describes an approach to designing signal processing systems called Cooperative Analog and Digital Signal Processing (CADSP) in which significant portions of the signal processing are done in programmable analog VLSI circuits. We then present noise suppression system for single-channel background audio noise suppression designed using the CADSP framework. The system is unique in that it is implemented in programmable analog VLSI circuits. The analog nature of the system has several novel implications:

1. The system is extremely low–power, operating on a wide-band signal while using only a few milliwatts of power.

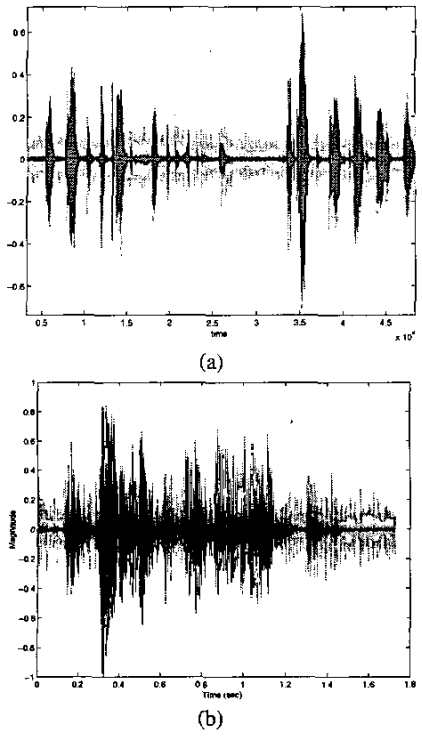


Fig. 6. (a) Time-domain waveform of original noisy signal (gray) and noise-suppressed signal (dark) from our functional circuit simulation. (b) Noise suppression from analog VLSI circuit in one frequency band. The light gray data is the subband noisy speech input signal; the black waveform is the corresponding subband output, after the gain function has been applied.

2. The noise suppression quality appears to be comparable in quality to that of state-of-the-art digitally implemented adaptive Wiener filters.
3. The system provides a proof of concept for the potential usefulness of signal processing using analog VLSI circuits.

We believe that this circuit is one of the first of its kind—an analog VLSI signal processing circuit with wide applicability. Our choice of filters, noise estimator, and gain function yield a robust system for a variety of noise levels and conditions. Also, we demonstrated circuit implementations of each of the algorithmic components.

7. REFERENCES

[1] Carver Mead, *Analog VLSI and Neural Systems*, Addison-Wesley, Reading, MA, 1989.

[2] K. Boahen and A. Andreou, "A contrast-sensitive retina with reciprocal synapses," in *Advances in Neural Information Processing Systems 4*, J.E. Moody, Ed. Morgan Kaufman Publishers, San Mateo, CA, 1991.

[3] L. Watts, D.A. Kerns, and R.F. Lyon, "Improved implementation of the silicon cochlea," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 5, pp. 692–700, 1992.

[4] J. Lazzaro and C. A. Mead, "A winner-take-all circuit in o(n) complexity," in *Advances in Neural Information Processing Systems 1*, Gerald Tesauro and David S. Touretzky, Eds., pp. 817–824. MIT Press, Cambridge, MA, 1989.

[5] Matt Kucic, Paul Hasler, Jeff Dugger, and David V. Anderson, "Programmable and adaptive analog filters using arrays of floating-gate circuits," in *2001 Conference on Advanced Research in VLSI*, Erik Brunvand and Chris Myers, Eds. IEEE Computer Society, March 2001, pp. 148–162.

[6] Asad A. Abidi G. Tyson Tuttle, Siavask Fallahi, "An 8b CMOS vector A/D converter," in *Proceedings of the IEEE International Solid State Circuits Conference*, Monterey, CA, 1993, pp. 257–259.

[7] Jeremy Lubkin and Gert Cauwenberghs, "A micropower learning vector quantizer for parallel analog-to-digital data compression," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Monterey, CA, 1998, vol. III, pp. 58–61.

[8] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network with 10240 'floating gate' synapses," in *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C., 1989, vol. II, pp. 191–196.

[9] P. Hasler, C. Diorio, B. A. Minch, and C. A. Mead, "Single transistor learning synapses," in *Advances in Neural Information Processing Systems 7*, Gerald Tesauro, David S. Touretzky, and Todd K. Leen, Eds., pp. 817–824. MIT Press, Cambridge, MA, 1995.

[10] Paul Hasler, Bradley A. Minch, and Chris Diorio, "Floating-gate devices: They are not just for digital memories anymore," in *IEEE International Symposium on Circuits and Systems*, Orlando, Florida, 1999, vol. II, pp. 399–391.

[11] P. Hasler and T.S. Lande, "Special issue on floating-gate devices, circuits, and systems," *IEEE Journal of Circuits and Systems*, vol. 48, no. 1, Jan. 2001.

[12] Tyson S. Hall Paul Hasler and David V. Anderson, "Field-programmable analog arrays: A floating-gate approach," in *12th International Conference on Field Programmable Logic and Applications*, Montpellier, France, Sept. 2002.

[13] Gene Franz, "Digital signal processor trends," *IEEE Micro*, vol. 20, no. 6, pp. 52–59, Nov–Dec 2000.

[14] Carver A. Mead, "Neuromorphic electronic systems," *IEEE Proceedings*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[15] Paul Hasler and David V. Anderson, "Cooperative analog-digital signal processing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, FL, May 2002, vol. IV, pp. 3972–5.

[16] David V. Anderson and Paul Hasler, "Cooperative analog/digital signal processing," in *World Conference on Systemics, Cybernetics, and Informatics*, Orlando, FL, July 2001.

[17] David V. Anderson, "Model based development of a hearing aid," M.S. thesis, Brigham Young University, Provo, Utah, 1994.

[18] Paul Hasler, Matt Kucic, and Bradley A. Minch, "A transistor-only circuit model of the autozeroing floating-gate amplifier," in *Midwest Conference on Circuits and Systems*, Las Cruces, NM, 1999.

[19] Matt Kucic, AiChen Low, Paul Hasler, and Joe Neff, "A programmable continuous-time floating-gate fourier processor," *IEEE Transactions on Circuits and Systems II*, vol. 48, no. 1, pp. 90–99, Jan. 2001.

[20] Rich Ellis, Heejong Yoo, David Graham, Paul Hasler, and David Anderson, "A continuous-time speech enhancement front-end for microphone inputs," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Phoenix, AZ, 2002, vol. II, pp. 728–31.

[21] David Graham and Paul Hasler, "Capacitively-coupled current conveyer second-order section for continuous-time bandpass filtering and cochlea modeling," in *Proceedings of the International Symposium on Circuits and Systems*, 2002, vol. to appear.

[22] Ranier Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, July 2001.

[23] T. Serrano-Gotarredona, B. Linares-Barranco, and A. G. Andreou, "A general translinear principle for subthreshold MOS transistors," *IEEE Transactions on Circuits and Systems*, vol. 46, no. 5, pp. 607–616, May 1999.