CONTINUOUS-TIME AUDIO NOISE SUPPRESSION AND REAL-TIME IMPLEMENTATION

Heejong Yoo, David V. Anderson, and Paul Hasler

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332

ABSTRACT

In this paper, we propose a continuous-time audio noise suppression algorithm for reducing stationary background noise in a single microphone signal. The algorithm is targeted for implementation in sub-threshold analog floating-gate transistor circuits for extremely low-power systems. The noise suppression algorithm divides the noisy signal into exponentially spaced sub-bands and estimates the noisy signal envelope and the noise envelope to calculate a time-varying gain for each sub-band based on each band's *a posteriori* SNR. A sigmoidal gain function in for use in the analog system is introduced and a normalized SNR is also proposed for consistent noise suppression for different input noise variances. Simulations of the system show promising results.

1. INTRODUCTION

Audio signal enhancement by removing additive background noise from corrupted noisy signal is not a new idea; but, with the prosperity of the portable communication devices, it has received much attention recently. While most noise suppression methods focus on the processing of discrete-time audio signals, we propose a continuous-time noise suppression method in which the processing is performed on the microphone signal prior to the A/D conversion, as illustrated in Fig. 1. By performing this significant portion of the processing in low-power analog circuits [1], we hope to enhance the overall functionality of an entire system, by utilizing analog/digital computation in mutually beneficial way. A review of the cooperative analog/digital computing paradigm with its benefits is presented in [2].

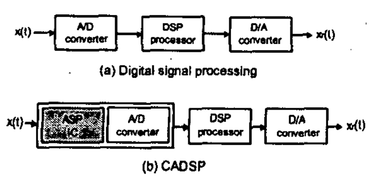

Fig. 1. (a) Typical signal processing system where processing is done between the A/D and D/A converters. (b) Cooperative analog and digital signal processing (CADSP). Continuous-time noise suppression is done before the A/D converter.

The proposed noise suppression algorithm consists of a one-third octave filter bank with each band containing an envelope detector, a recursive noise estimator, and a non-linear gain function. The filter bank separates the noisy signal into narrow-band signals; at each band we perform envelope detection and averaging. From the smoothed sub-band signal envelope we estimate the noise envelope in each sub-band. The gain function is calculated from the noisy signal level (envelope) estimate as well as noise level (envelope) estimate using non-linear sigmoid gain function. 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 details of analog components and the implementation of this algorithm are addressed in [3].

This paper is organized as follows. In Section 2, we present the continuous-time noise suppression algorithm which exploits the characteristics of the sub-threshold operation of floating-gate circuits for real-time and low-power implementation. In Section 3, we present our simulation result, and we conclude this paper in Section 4.

2. CONTINUOUS-TIME NOISE SUPPRESSION

A common model for a noisy signal, $x(t)$, is a signal, $s(t)$, plus additive noise, $n(t)$, that is uncorrelated with the signal

$$x(t) = s(t) + n(t). \quad (1)$$

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 audio signal is non-stationary and the additive noise is stationary at least for a long time relative to the speech. Figure 2 shows the structure

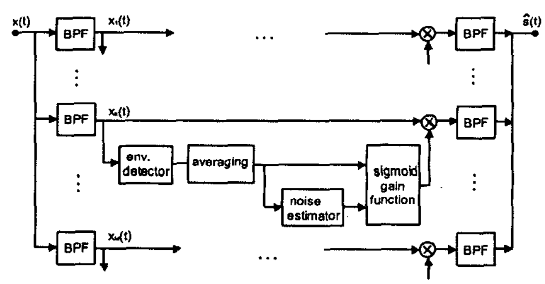

Fig. 2. The block diagram of continuous-time noise suppression system. The center frequency of filter bank is spaced exponentially. At each sub-band, gain is calculated from non-linear gain function. To achieve consistent noise suppression for different noise variance, noisy signal estimate, instead 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.

of a continuous-time noise suppression system for real-time implementation. A bank of band-pass filters break the input noisy signal

Research sponsored by the Hewlett-Packard Corporation and the National Science Foundation under Grant CISE-1068549.

into sub-band signals. At each sub-band, the envelope detection of band-limited signal $x_k(t)$ is performed and averaged to provide a smooth estimation of the envelope and followed by noise level estimation. Gain computation at each band is performed using non-linear gain function with each band's *a posteriori* SNR. For simplicity, we abbreviate *a posteriori* SNR as SNR. The unmodified band-limited signal, $x_k(t)$, is then multiplied by the calculated gain to obtain the noise suppressed band-limited signal, $\hat{s}_k(t)$. All of the $\hat{s}_k(t)$ sub-band signals are summed to build the full-band signal estimate, $\hat{s}(t)$, after the synthesis filter bank. The details of each steps are discussed below.

2.1. Band-pass Filter Bank

The analysis and synthesis filter banks are implemented with Fourier processor IC [4] that has been developed in our previous research. It can easily divide the signal into frequency bands spaced exponentially instead of linearly, as in the typical DFT algorithm. We set the bandwidth of each band-pass filter equal to the bandwidth of the repective critical band in the human ear. By adopting one-third octave spacing in the filter bank 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 [5]. Each continuous-time filter is the transistor-only version of autozeroing floating-gate amplifier(AFGA), which also referred to as the capacitively coupled current converyer(C^4) [3]. This filter allows both the low-frequency and high-frequency cutoffs to be controlled electronically by changing the appropriate bias currents.

2.2. Envelope Extraction and Averaging

An audio signal, $x_k(t)$, also can be represented as

$$x_k(t) = e_k(t)v_k(t), \tag{2}$$

where $v_k(t)$ is a band-limited signal with an approximately constant magnitude and $e_k(t)$ represents the envelope variation over time. Figure 3 shows $e_k(t)$ and $x_k(t) = e_k(t)v_k(t)$ for a single

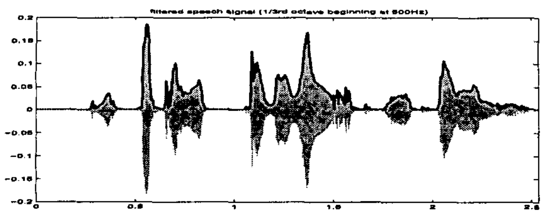

Fig. 3. The gray area is $x_k(t) = e_k(t)v_k(t)$. The dark line is $e_k(t)$, the envelope. Since we use $e_k(t)$ as an input of non-linear gain function, we apply averaging to $e_k(t)$ for smoothing gain function.

band from a sentence spoken by a female. Many different algorithms may be used in the simulation to estimate $e_k(t)$ from $x_k(t)$ including

- half-wave or full-wave rectification followed by low-pass filtering of $x_k(t)$
- using the magnitude of the output of Hilbert transform using FIR filters
- low-pass filtering of $x_k^2(t)$

An initial envelope estimate $\hat{e}_k(t)$ is found, for this system, using a simple diode–capacitor AM demodulation circuit. This envelope is then smoothed via a low-pass filter. This smoothing helps to make the final noise–reduced signal more free from artifacts like musical noise and missing consonants. The averaged noisy signal envelope estimate in the k-th subband, $\bar{e}_{x,k}(t)$ is described by

$$\alpha \frac{d\bar{e}_{x,k}(t)}{dt} + \bar{e}_{x,k}(t) = \hat{e}_k(t), \tag{3}$$

where α is a time constant for noisy signal envelope estimation, and $\bar{e}_{x,k}(t)$ is the (smoothed) envelope of the noisy signal. Note, if we average $\hat{e}_k(t)$ over too long of a time interval, it might degrade some accuracy of the level estimate of the noisy signal which is assumed to be non-stationary. However, if we estimate noise level from this averaged noisy signal estimate, we can expect that averaging might be benefical to noise level estimate which is assumed to be stationary. Therefore, the estimate of the noise level is obtained by further filtering of $\bar{e}_{x,k}(t)$.

2.3. Noise Level Estimation

The average noise envelope estimate, $\bar{e}_{n,k}(t)$, may be obtained in several ways. One way to is by using an inverted peak detector or "min" detector with a long time constant on the average envelope signal $\bar{e}_{x,k}(t)$. This method is discussed more in [3]. Another method is to use single-pole recursive averaging with a time-varying pole as

$$\beta(t)\frac{d\bar{e}_{n,k}(t)}{dt} + \bar{e}_{n,k}(t) = \bar{e}_{x,k}(t), \tag{4}$$

where $\beta(t)$ is a time constant for noise estimation. Variations of Eq. 4 are introduced in [6]. Equation 4 has the same structure as Eq. 3, but while Eq. 3 uses fixed time constant, α, for the averaging, Eq. 4 uses time constant function, $\beta(t)$, for the estimation.

2.3.1. Discrete Time Constants

For the best estimation of stationary background noise level, two-sided single-pole recursion has been investigated [7]. It generally has separate attack and decay time constants and uses the attack (decay) time constant if input signal is greater (smaller) than the noise estimate.

$$\beta(t) = \begin{cases} \tau_a & \text{if} \quad \bar{e}_{x,k}(t) > \bar{e}_{n,k}(t) \\ \tau_d & \text{if} \quad \bar{e}_{x,k}(t) < \bar{e}_{n,k}(t) \end{cases} \tag{5}$$

Typically the attack time constant, τ_a, is on the order of second and the decay time constant, τ_d, is on the order of milli-second.

2.3.2. Continuous Time Constants

Unlike the discrete-time domain, instantaneous time constant switching, especially one is much bigger than the other, can be hard to implement in the continuous-time domain since the time constant is generally controlled by the bias current [8] and its maximum change rate might be limited by slew-rate when the current is large signal. To minimize the quality deterioration of noise suppressed audio signal in the presence of physical slew-rate limitation, we propose asymmetric time constant function for the noise level estimation, as shown in Fig. 4(b).

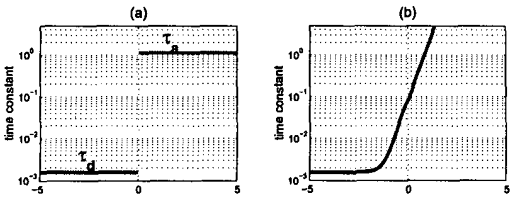

Fig. 4. Time constants, $\beta(t)$, for noise estimation. The abscissa is $\bar{e}_{x,k}(t) - \bar{e}_{n,k}(t)$. (a) Attack($\tau_a$) and decay time constant(τ_d) in discrete-time domain. (b) Asymmetric time constant function in continuous-time domain. The asymmetric time constant is used to incorporate small $\beta(t)$ when $\bar{e}_{x,k}(t) \ll \bar{e}_{n,k}(t)$ and a large $\beta(t)$ when $\bar{e}_{x,k}(t) \gg \bar{e}_{n,k}(t)$ and slew-rate limitation in transition region.

We experimentally observed that the sensitivity of time constant change to the quality of sound increases as the SNR decreases. To keep the quality of noise suppressed signal from degrading caused by the time constant function, we should make continuous time constant function, when $\bar{e}_{x,k}(t) < \bar{e}_{n,k}(t)$, be as small as possible in order for estimator to adapt quickly to the input noisy signal. When $\bar{e}_{x,k}(t) > \bar{e}_{n,k}(t)$, however, the maximum change rate of continuous function might be limited by physical slew-rate. As a model of continuous-time constant when $\bar{e}_{x,k}(t) > \bar{e}_{n,k}(t)$, we attempt polynomial function or exponential function, which is parametrized so that the maximum change rate does not exceed the physical slew-rate limitation. Unlike the discrete time-constant, the continuous time-constant function is not designed to have saturation region as $\bar{e}_{x,k}(t)$ increases. This simplifies the implementation and its effect on the resulting sound quality negligible since quality is insensitive to change of time constant in this region.

2.4. Non-linear Gain Function

In the discrete-time domain, the Wiener gain is widely used in noise suppression for its superiority in performance. However, many other gain functions are possible (e.g. see [9]) and calculating the Wiener gain can be difficult in the target system. Another factor is that Wiener filtering for noise suppression is usually performed using block processing whereas our system must operate on the continuous signal in real-time. We simulated the Wiener gain and two other gain functions to evaluate them for both performance and ease of implementation. The first of the other two non-linear gain functions is the simple bi-linear gain function [7, 6], H_{bi}.

$$H_{bi}(t) = min\Big[1, \Big\{ \frac{\bar{e}_{x,k}(t)}{\gamma \bar{e}_{n,k}(t)} \Big\} \Big] \quad (6)$$

where, γ is the threshold that specifies the certainty of speech is declared. A disadvantage of bi-linear gain function is that if we increase γ in order to achieve high noise suppression rate at low SNR, then we also sacrifice audio signal magnitude up to the region where the SNR equals γ. To minimize this signal reduction, we propose sigmoid gain function, H_{sig}, which is quite easy to implement with current mirror circuits in continuous time.

$$H_{sig}(t) = \frac{1}{2}\Big\{ \tanh\Big(\alpha \Big(\frac{\bar{e}_{x,k}(t)}{\bar{e}_{n,k}(t)} - \phi\Big)\Big) \Big\} + \delta \quad (7)$$

where, α, ϕ, and δ are parameters for slope, horizontal, and vertical shift, respectively.

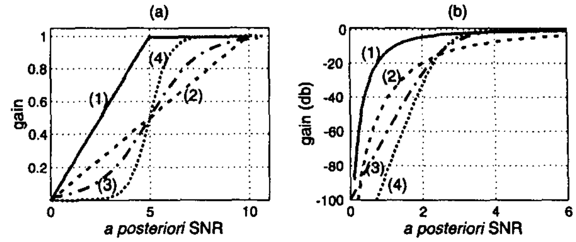

Fig. 5. (a) Bi-linear gains and sigmoid gains. (1) bi-linear with $\gamma = 5$, (2) bi-linear with $\gamma = 10$, (3) sigmoid gain with $\phi = 5, \delta = 0.5$, and $\alpha = 0.4$, and (4) sigmoid gain with $\phi = 5, \delta = 0.5$, and $\alpha = 1$. (b) Wiener gains and sigmoid gains. (1) Wiener gain with oversubtraction factor = 1, (2) Wiener gain with oversubtraction factor = 7, (3) sigmoid gain with $\phi = 2.5, \delta = 0.5$, and $\alpha = 1$, and (4) sigmoid gain with $\phi = 2.5, \delta = 0.5$, and $\alpha = 1.4$.

The sigmoidal gain function is superior to bi-linear gain function in that it not only reduces more noise when SNR $\leq \phi$, but also preserves more signal magnitude when $\phi \leq$ SNR $\leq \gamma$. Figure 5(a) shows sigmoid gain functions with $\delta = 0.5$, $\phi = \gamma/2 = 5$ and bi-linear gain functions with the maximum noise suppression rate is 20dB ($\gamma = 10$) and 14dB ($\gamma = 5$). Figure 5(b) shows the distinction between Wiener gains with oversubtraction and sigmoid gains. We can see that sigmoid gains can achieve more noise suppression than Wiener gains when SNR ≈ 1 (normalized SNR), while preserving as much signal as Wiener gain at higher SNR.

It is often desirable perceptually to leave some background noise to minimize the loss of unvoiced consonants that may be indistinguishable from the added noise. Typical δ should satisfy $\delta > 1/2$ to leave some residual background noise.

2.5. Normalized SNR

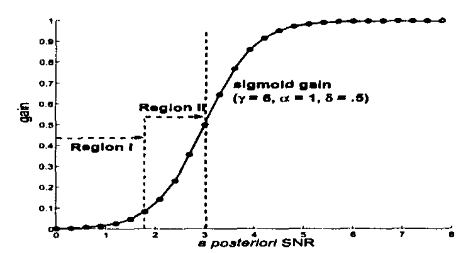

Fig. 6. Dynamic range of gain function when the signal is absent. For the consistent noise suppression when different input noise variances are applied, SNR, whenever signal is not present, should always be within the Region I. If it goes into the Region II, or, in worst case, above the Region II, the probability of false alarm increases and subsequently it generates artifacts.

The gain function is a function of the instantaneous SNR, but estimating the SNR is a noisy process. Especially when the added noise has a high variance. Figure 6 shows two possible dynamic ranges of SNR when the signal is not present. If the estimated SNR varies within the region I, most of the background noise will be suppressed. However, if it goes into or beyond the region II, the algorithm will preserve considerable background noise and generate noticeable artifacts. This can be called a *false alarm*. False alarm shows that gain function optimized for specific input noise level could fail to suppress the noise effectively for diffrent noise level. To avoid or severely reduce the false alarm, we should guarantee that the SNR, when the signal is not present, should be normalized to stay within the region I.

Diethorn [6] estimated the envelope of the noisy and the noise envelope in parallel from band-limited signal and we experimentally observed this could tends to generate false alarms, as it can be seen from Fig. 7(b). To overcome this problem, as shown in Fig. 2, we estimate the noisy signal envelope from band-limited signal and estimated noise envelope not from the band-limited signal but from the averaged envelope of the noisy signal. Since the noise estimator with time-constant function, $\beta(t)$, is analogous to minimum detector, we can achieve more normalized SNR than [6] and this guarantees reliable noise suppression independent of input noise variance.

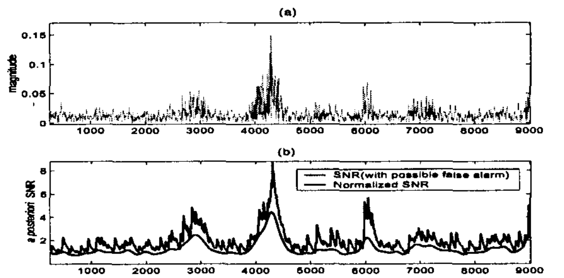

Fig. 7. Normalized SNR. (a) Noisy signal envelope of a single band. (b) Normalized SNR (lower line) calculated from proposed method and SNR calculated from [6] (upper line). When the signal is assumed to be not present, normalized SNR have SNR $\approx$ 1.

3. SIMULATION

The continuous-time noise suppression algorithm has been functionally simulated and is being implemented with floating-gate transistors for low-power noise suppression IC [3]. In order to anticipate accurate performance when implemented with circuits, we tried to include physical limitations as much as possible.

In Fig. 8, original noisy signal, which was corrupted with background noise, is drawn overlapped with noise suppressed output signal. While it is possible to further suppress the noise by changing parameters, we experimentally observed that severe noise suppression can deteriorate the intelligibility of signal.

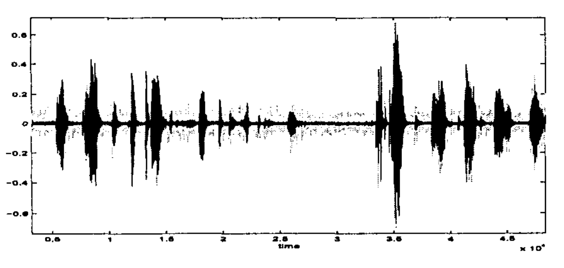

Fig. 8. Time-domain waveform of original noisy signal (gray) and noise-suppressed signal (dark) from our functional circuit simulation.

4. CONCLUSION

We have presented a continuous-time audio noise suppression algorithm for implementation in extremely low-power, real-time analog circuits. For quality and practical issues, we proposed a normalized SNR, non-linear gain function and asymmetric continuous time constants for estimating the signal and noise respectively.

Even though we use simple, low-complexity building blocks such as the low-complexity noise estimator and simple gain function, the simulation results show significant noise suppression while generating few artifacts. We believe that the choice of filters, noise estimator, and gain function yield a robust system for a variety of noise levels and conditions. Also, since each of the functional modules of proposed algorithm was developed for continuous-time implementation, we expect to build a programmable noise suppression IC with low cost directly from this algorithm.

5. REFERENCES

[1] P. Hasler, and T. S. Lande, "Overview of floating-gate devices, circuits, and systems", *IEEE Trans. on Circuits and Systems II:Analog and Digital Signal Processing.*, Vol. 48, No. 1, pp. 1-3, January 2001.

[2] P. Hasler, and D. V. Anderson, "Cooperative analog and digital signal processing", in *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, Orlando, Florida, 2002.

[3] R. Ellis, H. Yoo, D. Graham, P. Hasler, and D. V. Anderson, "A continuous-time speech enhancement front-end for microphone inputs", in *Proceedings of the IEEE International Symposium on Circuits and Systems* , Phoenix, Arizona, 2002.

[4] M. Kucic, A. Low, P. Hasler, and J. Neff, "A programmable continuous-time analog Fourier processor", *IEEE Trans. on Circuits and Systems II:Analog and Digital Signal Processing.*, Vol. 48, No. 1, pp. 90-99, January 2001.

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

[6] S. L. Gay, and J. Benesty, *Acoustic Signal Processing for Telecommunication*, Kluwer Academic Publishers, 2000.

[7] E. J. Diethorn, "A subband noise-reduction method for enhancing speech in telephony & teleconferencing", in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk Mountain House, New Paltz, NY, Oct. 19-22, 1997.

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

[9] Levent Arslan, Alan McCree, and Vishu Viswanathan, "New methods for adaptive noise suppression," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1995, vol. 1, pp. 812–815.