MULTICHANNEL FILTERING FOR OPTIMUM NOISE REDUCTION IN MICROPHONE ARRAYS

Dinei A. Florêncio and Henrique S. Malvar

Microsoft Research, One Microsoft Way, Redmond, WA, 98052

ABSTRACT

This paper introduces a new optimization criterion for the design of microphone arrays, and derives an optimum filter based on this criterion. The algorithm computes two separate correlation matrices for the signal: one for when only background noise is present, and one for when both noise and signal are present. A filter is then computed based on these matrices, optimizing the proposed weighted mean-square error criterion. A block-recursive version of the algorithm is presented, using LMS-like adaptation of the multichannel filters, with a computational complexity under 40 MIPS for a typical application with four microphones. Simulation results with typical office noise show improvements of up to 20 dB in signal-to-noise ratio, even in low-noise environments.

1. INTRODUCTION

Using signal processing to improve the quality of speech acquired by a microphone has been a long-standing interest in the DSP community, with some of the most promising technologies being based on microphone arrays. The microphone array literature is particularly populated with algorithms based on the Generalized Sidelobe Canceller (GSC) [1], but performance degrades quickly with reverberation [2]. A few other algorithms are based on optimum filtering concepts, or signal subspace projection [3]-[5]. A completely different approach comes from Blind Source Separation (BSS) [6]. Curiously, while BSS techniques tend to be overly sensitive to ambient conditions (e.g., room reverberation), it performs extremely well in some environments.

We analyzed some of the situations in which traditional microphone arrays are outperformed by BSS. We noted that BSS techniques generally focus in making the recovered signals statistically independent, putting essentially no penalty on signal distortion. While ignoring signal distortion altogether may not be a good idea, paying extra attention to (statically independent) noise seem to be highly justified by our subjective perception of speech quality. Based on this reasoning, and on a few subjective test experiments, we concluded that a new optimization criterion is needed, which accounts for noise differently from signal distortion. In this paper, after formalizing this new criterion, we derive a new algorithm, which produces an optimum filter under this new error criterion. The algorithm has some resemblance to the SVD-based algorithm proposed by Doclo and Moonen [5], but does not involve SVD computations, and optimizes the proposed error measure, instead of mean square error (MSE). We also present a less computationally intensive, LMS-based, version of our algorithm. This version of the algorithm has some similarities with the algorithm proposed by Nordholm at all [7], but does not rely on pre-stored calibration signals. Results of using the proposed algorithms in real-world signals are presented, which show noise suppression of up 20 dB. This is around 8 dB higher than that of an optimum filter based on traditional criteria, while the extra signal distortion is essentially unnoticeable.

2. A WEIGHTED ERROR CRITERIA FOR MULTICHANNEL WIENER FILTER

We now introduce our notation. For simplicity, we replicate each sample of the input as many times as filter taps that will use that sample, and form an input vector **x**(n), which contains samples from all input channels, and from current and past (or "future") sample of each of those channels. So, for example, if we denote one microphone signal as $x_1(n)$, and another microphone signal as $x_2(n)$, an input vector $\mathbf{x}(n)$ for a 3-tap per channel array could be composed as:

$$\mathbf{x}(n) = [x_1(n-1)\ x_1(n)\ x_1(n+1)\ x_2(n-1)\ x_2(n)\ x_2(n+1)]. \quad (1)$$

Therefore, at each time instant n, $\mathbf{x}(n)$ is a $T \times 1$ vector, where T is the number of total taps in the filter (generally the number of channels multiplied by the number of taps used for each channel). Furthermore, from now on we will drop the time index n, and write simply **x** to denote the input vector. We use a similar notation for all other vectors and variables.

We assume that noise is linearly added to the desired signal. In other words, we can write:

$$\mathbf{x} = \mathbf{s} + \mathbf{n}, \quad (2)$$

where **s** is the speech component of the signal and **n** is the additive ambient or interfering noise. Furthermore, we assume that the noise is statistically independent from the desired signal, although it might be correlated between different microphones.

The basic hypothesis is that the desired signal is essentially the same on all channels, possibly with the exception of a delay, or maybe different room-transfer functions. We want to compute a filter **w**, which will produce a single-channel output y, given by:

$$y = \mathbf{w}.\mathbf{x}, \quad (3)$$

where **w** is the $1 \times T$ filter vector, and which minimizes an appropriate error measure between y and a desired signal d.

2.1. A weighted error criteria

We want to choose the filter **w** in (3) such that the output signal y is as close as possible to desired signal d. We could simply use the overall mean square error (MSE):

$$\varepsilon' = \mathrm{E}\{(y-d)^2\} = \mathrm{E}\{(\mathbf{w}.(\mathbf{s}+\mathbf{n})-d)^2\}. \quad (4)$$

Where E{·} denotes expected value. However, the MSE is not appropriate, as discussed in the Introduction. It gives the same weight to any *distortion* introduced in the desired signal and any remaining *noise* left in the output. In contrast, subjective tests clearly favor a distorted signal when compared to a signal with the equivalent noise level in terms of MSE. We therefore want to give different weights to the residual noise and to the error due to distortion in the desired signal. We introduce a parameter β to denote this extra weight, and so we can write our new weighted error measure ε as:

$$\varepsilon = \mathrm{E}\{(\mathbf{w}.\mathbf{s} - d)^2 + \beta(\mathbf{w}.\mathbf{n})^2\}. \qquad (5)$$

Note that the first term in (5) is due to the signal distortion, while the second term reflects only the effects of the noise. Since we assume **n** and d are statically independent, ε and ε' become the same for $\beta = 1$. By using a higher value for β, we can put more weight on the independent noise component, which is the criterion we discussed above.

2.2. The optimum filter for stationary signals

In general, even continuous speech is characterized by periods of silence between utterances. Therefore, as commonly done in many other speech enhancement algorithms, we use a speech activity detector to classify the signal into "speech" and "silence" periods. Assuming that the noise is stationary, we can use these "silence" periods to obtain estimates about the statistical properties of the noise, as commonly done in spectral subtraction and Wiener filtering.

We can compute the optimum filter that minimizes the weighted error given in (5). Using a derivation similar to that for the Wiener filter, this optimum filter can be shown to be:

$$\mathbf{w}_{opt} = (R_{xx} + \rho R_{nn})^{-1}\left(\mathrm{E}\{d\mathbf{x}\}\right), \qquad (6)$$

where R_{xx} is the autocorrelation matrix for the input vector **x** (which includes both signal and noise), R_{nn} is the correlation matrix for the noise component **n**, and $\rho = \beta - 1$. Note that for $\rho = 0$, this is simply the traditional Wiener filter, as we would expect.

Implicit in the derivation is the assumption that the signal and the noise are stationary. We can therefore compute the statistics for **n** during "silence" periods, and the statistics for **x** during speech activity periods.

To circumvent the need for the desired signal d in (6), we select one of the channels (and implicitly the room impulse responses within that channel) as our "primary" channel, and use that (reverberated) speech signal as our desired signal. We can then write:

$$\mathbf{w}_{opt} = (R_{xx} + \rho R_{nn})^{-1}\left(E\{x_o\mathbf{x}\} - E\{n_o\mathbf{n}\}\right), \qquad (7)$$

where x_0 and n_0 represent the signal and noise from the selected reference channel, respectively.

Figure 1 shows a block diagram illustrating this basic algorithm. The statistics for noise and for signal+noise are computed/updated periodically. The block "spatial filter" is the optimum filter based on these accumulated statistics, and is computed (periodically) based on Equation (7). Note also that

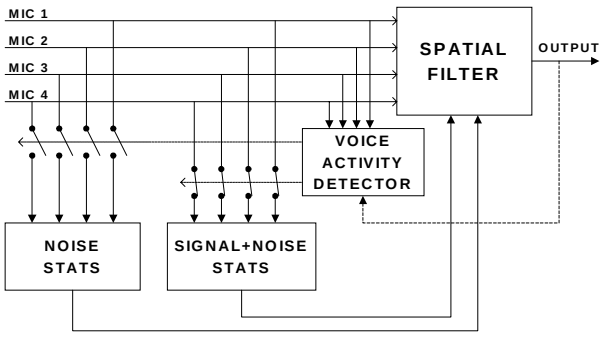

Figure 1 – High-level block diagram of the optimum filter.

$E\{x_0\mathbf{x}\}$ and $E\{n_0\mathbf{n}\}$ are in fact just columns of R_{xx} and R_{nn} respectively.

2.3. Some considerations about stationarity

Although in most situations the noise is approximately stationary (e.g. typical office noise from PCs and air conditioning), the signal certainly is not. By using relatively short processing frames, we can consider the speech as slowly varying. Thus, we can still use the noise statistics computed during the silence periods, and interactively compute the signal+noise statistics when the signal is present. The filter obtained in this manner would be time-varying, and would be close to optimum at all time instants, as long as the estimate for the signals statistics are accurate. This scenario is the typical case of speech in a stationary noise background.

Finally, we note that the term "stationary" here refers to both time/frequency and space domains. In other words, a fixed-location noise source will excite always the same correlation patterns between microphones. So, even for a completely non-stationary source (in terms of time behavior), the algorithm would be able to cancel the noise if the source is spatially fixed, albeit it may not be optimum anymore when compared to a time-varying filter tracking the behavior of the noise source.

3. TRANSFORM DOMAIN

Applying the approach derived in Section 2 directly to the input signal would imply in using long filters, and therefore, would require manipulating (and inverting) large matrices. Furthermore, estimating noise in one frequency range would interfere with the performance of the filter in other bands, which would be a problem because of the highly colored nature of typical speech signals. In order to reduce computational complexity, and improve the overall performance of the filter, we first apply a transform to each input signal (i.e., each microphone signal). We then apply the filtering process previously described to each frequency bin.

We use a modulated complex lapped transform (MCLT) [8], but other filter banks with perfect reconstruction could also be used. The MCLT helps reducing the uncancelled aliasing components, which appear in subband processing without the use of cross filters [8]. Figure 2 shows a block diagram of the MCLT version of the algorithm. Note also that a single signal presence variable is computed, but it is based on information from all frequency bands.

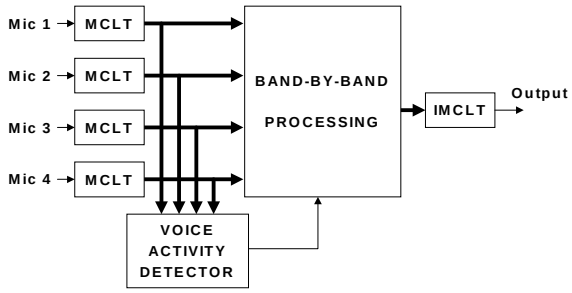

Figure 2 – High-level diagram of the MCLT version.

The band-by-band processing indicated in Figure 2 may be done in similar fashion to the one described before, or can use the LMS-based version, discussed in Section 4. Of course, the filters now will be complex filters, and have to be adjusted appropriately.

4. AN LMS-BASED VERSION

The algorithm described in the previous sections works based on the differences between the statistics of the signal and noise. These statistics are computed in a two-phase process ("noise only" and "signal+noise"), and stored as internal states in the system, represented by the two matrices (one for each phase). The adaptation is therefore based on using the incoming signal to update one of these matrices at a time, according to the presence (or absence) of the desired signal. In contrast, an LMS-based filter doesn't have the same two separate internal states matrices. It usually incorporates input data directly into the filter coefficients, and therefore does not allow for this two-phase process.

To circumvent this problem, we first note that the data contained in the statistics matrices is essentially a subset of the information contained in the corresponding signals, from which the matrices were computed. So, instead of storing the two statistics matrices, we propose to store the data itself in two separate circular buffers, and use them to directly adapt an LMS-filter. More precisely, the incoming data is classified as either "signal+noise" or "noise," and stored in the appropriate buffer for later usage. We then generate a synthetic input signal **z** and its associated desired signal *d* by adding data from the circular buffers to the input data. This synthetic signal is used to adapt an LMS filter. The filter coefficients are continuously copied to a separate filter, which directly process the input signal. This approach is similar to that used by Nordholm at all [7], but without the inconvenient of using calibration signals, thus making the overall system more robust to changes in the environment, the speaker, the noise, or the microphones. Also, the careful choice of synthetic signals – as described below – avoids the need to acquire a "clean" signal, as required in [7]. Figure 3 shows a block diagram of the proposed algorithm, as it applies to a single band.

The key in achieving the desired results is, of course, how to compose the signals that are used to adapt the LMS filter. We design our composed signals based on the optimization criteria discussed before, and assuming the circular buffers are short enough so that the signals contained in each are representative of the two classes. We propose using a two-phase composition: if speech is detected in the incoming signal **x**, we add more noise (from the circular noise buffer), to guarantee the desired extra

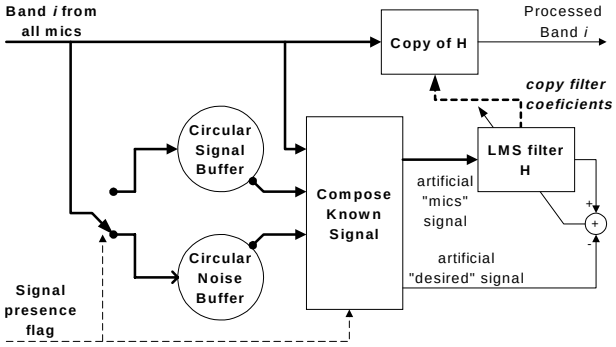

Figure 3. Processing in each frequency band.

noise attenuation. In other words, the input signal **z** to the adaptive filter is computed as:

$$\mathbf{z} = \mathbf{x} + \rho\,\mathbf{n}\,, \tag{8}$$

and the desired signal is set to:

$$d = x_0 - \frac{1}{\rho} n_0. \tag{9}$$

Note the negative term added to the desired noise; it is designed to guarantee that the filter will not try to preserve the small amount of noise present in x_0, but will converge to an unbiased estimate of the filter described in (7) instead. On the other hand, when no speech is detected in the incoming signal, we add a little bit of signal, to avoid converging to a signal-canceling filter:

$$\mathbf{z} = \rho\,\mathbf{x} + \mathbf{s}\,, \tag{10}$$

and set the associated desired signal to:

$$d = -\frac{1}{\rho} x_0 + s_0\,. \tag{11}$$

Note again the negative term in the desired signal, which has the same purpose as described before. Note also that we scale the input signal in such a way that the energy at the input of the filter does not vary significantly between speech and silence periods. Finally, while the algorithm adds different signals – depending on the Speech Activity Detector (SAD) – this is not actually critical. An eventual misclassification will not have significant consequences. On the other hand, including any parts of the desired signal in the noise buffer may lead to signal cancellation. To alleviate this problem, and since the SAD is not in the direct signal path, we use a long-delay SAD to decide in which buffer to store the incoming signal. This SAD has a "not-sure" region, where the signal is not stored in either buffer.

5. RESULTS

We have implemented frequency-domain versions of both the direct and the LMS forms of the proposed algorithm. In both cases we use a 64-band MCLT. While we have performed a number of experiments, we report here the results of using a 4-microphone array, in a typical 4m × 3m office. We compare the results with

those from using an optimum multichannel Wiener filter (computed with the aid of a close-talking microphone), and with a delay-and-sum beamformer. The signal-to-noise ratio (SNR) was computed by considering the ratio between the average energy during speech and average energy during silence periods, after convergence of the filters.

The SNR for the 4 microphones in the arrays was 13.3 dB, 12.6 dB, 11.4 dB, and 11.1 dB, while the close talking microphone presented a SNR of 29.8 dB. Using a delay-and-sum approach yielded an SNR of 12.9 dB, which is actually marginally worse than the best microphone signal. A reference LMS filter, based on traditional LMS error criteria, and using the close-talking signal as reference, improved the SNR to 25.4 dB. Using our algorithm we achieved a 33.0 dB SNR for the direct implementation, and 30.2 for the LMS-version. This is up to 7.6 dB better then the reference LMS, and more than 20 dB better than delay-and-sum. Furthermore, note that this reference LMS could not be implemented in practice, since it makes uses of the close talking microphone signal.

Finally, we observe that, even though, the extra noise attenuation is obtained at the expense of increased signal distortion, no significant degradation in the speech signal was observed. Particularly useful to notice is that the processing in the signal path has an intrinsic delay of less than 64 samples, and is essentially linear, therefore avoiding artifacts like musical noise, which are common in spectral subtraction and other nonlinear noise-reduction techniques. Samples of the input and processed signals are shown in Figure 4, and the signals are available for listening in [10].

6. CONSIDERATIONS ABOUT REVERBERATION

By replacing the desired signal in (6) with the desired signal as received in one of the microphones, as in (7), we intrinsically accept whatever level of reverberation was present at that microphone. In applications where the speaker may be seating more than a few feet away from the microphone array, and/or in highly reverberating rooms, the resulting reverberation may be unacceptable. There are several ways in which this problem can be alleviated. In particular, one may decide to simply use a reverberation reduction technique as the one proposed in [9], which can be readily cascaded with the algorithm presented herein by using the output of the reverberation reduction algorithm as the reference channel for the noise reduction array.

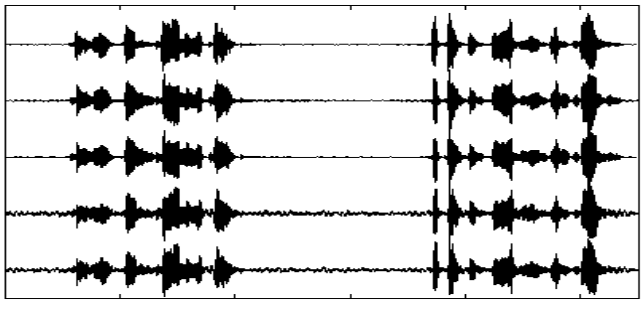

Figure 4. Sample waveforms. From top to bottom: processed (33.0dB SNR), reference-LMS (25.4dB), close-talking mic (29.8 dB), delay-and-sum (12.9 dB), and best mic (13.3 dB).

7. CONCLUSIONS

We have proposed a new optimization criterion for computing filters to enhance a signal in presence of noise. Based on this criterion, we have presented two algorithms. One of the key points underlying the proposed algorithms is the possibility of giving higher importance to independent noise, when compared to signal distortion. This has shown to significantly improve noise attenuation and overall subjective results. The LMS-like algorithm preserves the noise-reduction properties, while requiring a lower computational load. Matlab simulations have shown good results at a complexity below 40 MIPS, when processing a four-microphone signal sampled at 16KHz, using a 64-band MCLT, and 10 taps per frequency band. This computational complexity is certainly within today's computer typical processing power. Noise reduction of up to 20dB in low noise situations has been observed. In lower SNR environments, an even higher noise reduction can be obtained, as long as the presence of the noise does not affect the Voice Activity Detector.

8. REFERENCES

[1] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas Propagat.*, vol. AP-30, pp. 27–34, Jan. 1982.

[2] J. Bitzer, K Simmer, and K. Kammeyer, "Theoritical noise reduction limits of the generalized sidelobe canceller (GSC) for speech enhancement," *Proc. ICASSP*, Phoenix, AZ, pp. 2965–2968, 1999.

[3] F. Asano, S. Hayamizu, T. Yamada, and S. Nakamura, "Speech enhancement based on the subspace method," *IEEE Trans. on Speech and Audio Proc.*, vol. 8, pp. 497–507, Sep. 2000.

[4] S. Affes and Y. Grainier, "A signal subspace tracking algorithm for microphone array processing of speech," *IEEE Trans. on Speech and Audio Proc.*, vol. 5, pp. 425-437, Sep. 1997.

[5] S. Doclo and M. Moonen, "Robustness of SVD-based optimal filtering for noise reduction in multi-microphone speech signals," *Proc. IWAENC*, Pocono Manor, PA, pp. 80–83, Sep. 1999.

[6] L. Parra and C. Spence, "Convolutive blind separation of non-stationary sources," *IEEE Trans. on Speech and Audio Proc.*, vol. 8, pp. 320–327, May 2000.

[7] S. Nordholm, I. Claesson, and M. Dahl, "Adaptive microphone array emplying calibration signals: an analytical evaluation," *IEEE Trans. on Speech and Audio Proc.*, vol. 7, pp. 241–252, May 1999.

[8] H. Malvar, "A modulated complex lapped transform and its applications to audio processing," *Proc. ICASSP*, Phoenix, AZ, pp. 1421–1424, 1999.

[9] B. W. Gillespie, H. S. Malvar, and D. A. Florencio, "Speech dereverberation via maximum-kurtosis subband adaptive filtering," *Proc. ICASSP*, Salt Lake City, UT, 2001.

[10] http://research.microsoft.com/signal/micarray/icassp01/ma.