REAL-TIME AUDIO SOURCE SEPARATION BY DELAY AND ATTENUATION COMPENSATION IN THE TIME DOMAIN

Justinian Rosca, NingPing Fan, Radu Balan

Siemens Corporate Research
Princeton, NJ 08550
{justinian.rosca,ningping.fan,radu.balan}@scr.siemens.com

ABSTRACT

There is increased interest in using microphone arrays in a variety of audio source separation and consequently speech processing applications. In particular, small arrays of two to four microphones are presently under focus in the research literature, especially with regard to real-time source separation and speech enhancement capability. In this paper we focus on a real-time implementation of the delay and attenuation compensation (DAC) algorithm. Although the algorithm is designed for anechoic environments, its complexity and performance on real data represent a basis for designing more complex approaches to deal with reverberant environments. We highlight real-time issues and analyze the algorithm's real-time performance on a database of more than 1500 mixtures of real voice recordings ranging from an anechoic to a strongly echoic office with reverberation time of 500 msec.

1. INTRODUCTION

Speech enhancement and audio source separation applications are fertile ground for blind source separation (BSS) techniques using small arrays of microphones. Such solutions present an economic potential due to their low cost and improvement capability over single microphone solutions [1]. BSS approaches have been applied here but with limited success so far. Challenges highlighted two years ago in [2] still plague current approaches to blind separation of audio signals. Paramount among these challenges are methods for separating audio signals in real (echoic) environments and real-time performance on such data. These issues are increasingly being dealt with in the recent literature [3, 4, 5, 6] and are also dealt with here.

In this paper, we describe a real-time implementation of an algorithm for separating two audio signals recorded using a small array of two microphones. The algorithms relies on an anechoic mixture model, however it is shown to offer a basis for a fast implementation and offers insights when dealing with real data. Our main goals herein are to highlight such critical insights: determine complexity and performance factors on one side and measure limitations of the simple anechoic approach in real environments on the other. We expect that more complex solutions can be build upon these results.

Although our approach is derived from blind source separation principles, we use an anechoic propagation model in order to reduce the complexity of the problem and make it possible to effectively identify and invert the mixing process using second order statistics. For sources far away from the microphone array (in practice this means one meter at least, compared to the several centimeter distance between sensors), the model is simplified to depend on just a few parameters: relative *delays* of arrival of the wave front and *attenuations* at the microphones. The algorithm estimates the parameters in order to *compensate* for their true values, hence the name of the source separation algorithm is delay and attenuation compensation (DAC). An early description and results with DAC have been reported in [7].

Its real-time implementation has presented a number of challenges. We managed to simplify the estimation problem and also account for the fact that microphones are neither identical nor calibrated. The evaluation of our system is done using the instantaneous SNR measure for over 1500 mixtures of real data collected in both anechoic and echoic environments. The average instantaneous SNR gain was 13dB for anechoic environments and 3dB in echoic environments, where the better voice was separated at an average of 6 dB. In all cases processing introduces no artefacts.

The paper is organized as follows. Section 2 defines the parametric model used in our approach. Section 3 presents the real time algorithm used in this work. Section 3.3 discusses implementation challenges. Section 4 presents instantaneous SNR separation results obtained on anechoic and office data. Section 5 constrasts this work with related work published recently. Finally, Section 6 concludes and highlights challenges to be overcome in the near future.

2. PARAMETRIC MODEL

A general convolutive model for the mixing of two source signals at two sensors is:

$$\begin{array}{l} x_1(t) = h_1 \otimes s_1(t) + h_2 \otimes s_2(t) \\ x_2(t) = s_1(t) + s_2(t) \end{array} \tag{1}$$

where h_i represent unknown relative transfer functions of the first sensor versus the second and $\otimes$ represents convolution. We will use the assumptions that sources are decorrelated al all lags.

With a low complexity source separation algorithm in mind, we first simplify our treatment of the mixing problem by considering only direct path signal components, rather than use the general convolutive propagation model above. The direct-path component from one source arrives at two closely spaced sensors with a fractional delay between sensors. By fractional delay, we mean that delay between sensors is not generally an integer multiple of the sampling period [8]. The delay and attenuation compensation (DAC) mixing model in the time domain corresponds to the following equations [7]:

$$\begin{array}{l} x_1(t) = s_1(t - \delta_1) + c_1 \cdot s_2(t - \delta_2) \\ x_2(t) = c_2 \cdot s_1(t + \delta_1) + s_2(t + \delta_2) \end{array} \tag{2}$$

where: the delays δ_i are functions of the directions of arrival θ_j. They are defined with respect to the midpoint between sensors and depend also on the distance between sensors d and the speed of sound c: $\delta_i = \frac{d}{2c} \cos\theta_i, i = 1, 2$. We denote by Δ the maximal possible delay between sensors, $\Delta = \frac{d}{c}$. c_1, c_2 are two positive real numbers that account for the ratio of attenuations of the paths between sources and sensors for non-calibrated microphones and for deviations from the far-field assumption. Equation 2 describes the mixing matrix for the model in the time domain, in terms of four parameters, δ_1,δ_2, c_1, c_2.

The DAC solution to source separation in a nondegenerate case (number of sensors equals number of sources) is to invert this mixing matrix. This is obvious to perform in the frequency domain, and results into the following time domain solution:

$$\begin{array}{c} y_1(t) = h \otimes (x_1(t + \delta_2) - c_1 x_2(t - \delta_2)) \\ y_2(t) = h \otimes (-c_2 x_1(t + \delta_1) + x_2(t - \delta_1)) \end{array} \tag{3}$$

where the convolutive filter $h = h(t, \delta_1, \delta_2, c_1, c_2)$ accounts for the division with the determinant of the mixing matrix.

In practice we simplified the criterion above to a sufficient condition: decorrelation between fractionally delayed sensor recordings:

$$\begin{array}{l} y_1(t) = x_1(t + d_1) - c_1 x_2(t) \\ y_2(t) = c_2 x_1(t + d_2) + x_2(t) \end{array} \tag{4}$$

This is possible due to the freedom to shift signals under the assumption of decorrelation at any lag.

The DAC algorithm performs source separation by compensating for the true fractional delays and attenuations in the time domain with values determined by minimizing the output decorrelation objective:

$$R_{y_1 y_2}(\tau) = \langle y_1(t), y_2(t - \tau) \rangle = 0, \ \forall \tau \tag{5}$$

as a function of two unknown delays d_1 and d_2 and unknown scalar coefficients c_1 and c_2. $\langle \cdot, \cdot \rangle$ is the sampled cross-correlation between differences of fractionally delayed measurements. This is equivalent to the following optimization problem:

$$\left\{ \hat{d_1}, \hat{d_2}, \hat{c_1}, \hat{c_2} \right\} = \text{argmin} \sum_{\tau} |R_{y_1 y_2}(\tau)| \tag{6}$$

Note that the objective above relies on the anechoic model. A generalization of the solution in the reverberant case follows similar arguments, but introduces additional parameters to account for secondary propagation paths [9]. A classical approach to signal enhancement that implicitly accounts for convolutional effects is to consider all Wiener-like linear filtering combinations of $X_1 = X_1(\omega, t)$ and $X_2 = X_2(\omega, t)$, the windowed Fourier transforms of the measurements, of the form:

$$Y_i = G_{i1} X_1 + G_{i2} X_2 \tag{7}$$

If Y_i is an estimate of source X_i, then the minimum variance criterion

$$\mathrm{argmin}_{\mathrm{G_{i1}, G_{i2}}} \mathrm{Var}(\mathrm{Y_i} - \mathrm{S_i})$$

results in a paradoxically simple solution whose implementation necessitates the estimation of complex filters H_1 and H_2 defining the mixing model in 1 of a desired complexity:

$$Y(\omega) = \frac{1}{H_1 - H_2} \cdot \begin{bmatrix} 1 & -H_2 \\ 1 & H_1 \end{bmatrix} \cdot X \tag{8}$$

In the simple anechoic case, this reduces to our solution (4). In the case when the variances of the sources can be estimated, the solution can be extended to the degenerate case of more sources that sensors.

In the rest of the paper we focus on a real-time implementation of separation for the anechoic solution described by Equation 6 for reasons listed before. Complexity and performance characteristics of the simple algorithm particularly on real data from typical environments would influence decisions for the choice of a model to deal with reverberant conditions, which still results in an effective real-time implementation.

3. ON-LINE ALGORITHM

The algorithm used in the present implementation simplifies the estimation problem by dealing with attenuations in a calibration phase and evaluating output decorrelation based on the covariance of the mixtures. Calibration is performed online and also accounts for the fact that either microphones are not identical or that there is a deviation from the far-field assumption.

3.1. Online Calibration

Ideally, $c_1 = c_2 = 1$ under the far-field assumption, and microphones have identical gain characteristics. In practice however, it is hard to impose the latter condition. In the following we consider an *online* calibration criterion for making gain levels comensurate on the two channels. The criterion is to equalize variances of the channels on the current data frame. Below, assume the upper index represents the data frame index, m is the frame size, and N is total number of samples in a finite horizon before reading the current block of data. We recursively express the equalized means and variances of the mixtures after k frames using the present frame means and variances $\overline{x}_j$ and $Var(x_j)$, and then normalize the second channel to bring it to a similar variance level with the first channel as follows:

$$\begin{array}{l} \overline{x}_j^{(k)} = \frac{N}{N+m}\overline{x}_j^{(k-1)} + \frac{m}{N+m}\overline{x}_j \\ Var(x_j)^{(k)} = \frac{N-1}{N+m-1}Var(x_j)^{(k-1)} + \frac{m-1}{N+m-1}Var(x_j) \\ x_2 \leftarrow \sqrt{\frac{Var(x_1)^{(k)}}{Var(x_2)^{(k)}}} \cdot x_2 \end{array} \tag{9}$$

The recursive formulae above have direct online implementation. This allows us to drop the attenuation parameters in equation (4) and simplify the estimation of delays.

3.2. Delay Estimation

The decorrelation criterion (4) can be further simplified by expressing the cross-covariance of y_1 and y_2, $R_{y_1y_2}(\tau)$, as:

$$\begin{array}{l} E[(x_1(t+d_1) - x_2(t))\,(x_1(t+d_2-\tau) - x_2(t-\tau))] \\ \quad = R_{x_1}(d_1 - d_2 + \tau) - R_{x_1x_2}(d_2 - \tau) - \\ \quad - R_{x_1x_2}(d_1 + \tau) + R_{x_2}(\tau) \end{array} \tag{10}$$

Delay parameters are estimated by minimizing this expression. Note that in order to compute subunit delayed versions of crosscorelations, correlations have to be computed for a number of lags L.

3.3. Implementation

The real-time application is implemented as a multi-threaded Windows task on a Pentium III PC. The algorithm inputs come from the auxiliary input of the standard PC sound card, while outputs are continously streamed to the headphones. One thread performs the I/O of audio data in real time. The second thread is responsible with the analysis, calibration, delay estimation and syhthesis of the demixed signals.

Calibrated data are fed into the delay parameter estimation module, which uses the Amoeba optimization method [10] to find a locally optimal solution. We can ensure that the solution is also global by constraining the delay values based on d. Optimization uses the cost function (10). It starts with an initial simplex of three pairs of delays. The initial simplex is centered at the delays of last data block $(d_1 + 0.05, d_2 + 0.05)$, $(d_1 - 0.05, d_2 - 0.05)$, and $(d_1 + 0.05, d_2 - 0.05)$ (in samples). Solutions (d_1^*, d_2^*) of the optimization are smoothed using a learning rate α:

$$d_j = d_j^k = (1-\alpha) \cdot d_j^{k-1} + \alpha \cdot d_j^*, \quad j = 1, 2. \tag{11}$$

Delays are sorted in order to insure stability to the permutation problem [7] and are then directly used to generate the separated outputs.

4. EXPERIMENTAL RESULTS

4.1. Tests and evaluation

The algorithm was extensively evaluated on real data recorded at 16kHz in an anechoic room and an echoic office environment. Male, female (TIMIT database voices) and noises (e.g. printing, vacuuming, abd typing) were played from a loudspeaker placed at positions which were multiple of 30 degrees. Pairwise mixtures were created for all possible voice and angle combinations (excuding same voice or angle mixtures). The process resulted in more than one thousand test files. In the case of the office environment, the measured impulse revealed a reverberation time of about 500 msec. Figure 1 shows the time required for the energy to decrease by 60dB, which defines room reverberation.

We evaluated the online DAC method by computing SNR measures for each frame as follows. Denote the energy contribution of source j on output channel k by P_{jk}^o. Similarly, a superscript i denotes contribution at the microphone. Then a conservative measure of the SNR gain is:

$$\begin{array}{rl} SNR_1 = & max\{10log_{10}\frac{P_{11}^o}{P_{21}^o}, 10log_{10}\frac{P_{12}^o}{P_{22}^o}\} \\ & -max\{10log_{10}\frac{P_{11}^i}{P_{21}^i}, 10log_{10}\frac{P_{12}^i}{P_{22}^i}\} \\ SNR_2 = & -min\{10log_{10}\frac{P_{11}^o}{P_{21}^o}, 10log_{10}\frac{P_{12}^o}{P_{22}^o}\} \\ & +min\{10log_{10}\frac{P_{11}^i}{P_{21}^i}, 10log_{10}\frac{P_{12}^i}{P_{22}^i}\} \end{array}$$

The SNR results averaged the contribution of each frame except the first half a second of data, discarded to let the process converge. The other parameters used in the results below were L, m, $\alpha = 0.5$, $P = 0$.

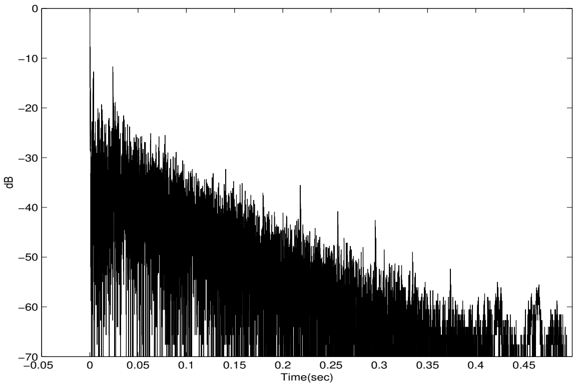

Fig. 1. Impulse response for echoic data set.

	m=512	m=1024.	m=4096
L=8	990 ms/s	500 ms/s	200 ms/s
L=10	1050 ms/s	600 ms/s	205 ms/s
L=20	1500 ms/s	750 ms/s	260 ms/s

Table 1. Real-time performance on a Pentium III 600 Mhz for various values of L (number of lags) and window size.

4.2. Performance

A first important characteristic of the DAC approach in general and the present real-time implementation is the artefact-free nature of the outputs. Table 4.2 presents performance measurements with this implementation. Overall, for $m = 4096$ and $L = 8$ the algorithm used 200msec CPU time for every second of real-time. The average instantaneous SNR gain varied somehow with angle, and achieved 10-14dB in anechoic environment and 3dB in echoic environments, where the better voice was separated at an average of 6dB. The performance was confirmed by audibility tests.

Figures 2,3 and 4,5 present the average of SNR values obtained on a frame basis during online processing for the anechoic and echoic data respectively.

The delay estimation algorithm converged close to the true delay values provided voice was present after processing only about 150-200 miliseconds of anechoic data (or about 2500 samples at 16kHz sampling frequency). Figures 6 and 7 exemplify the convergence and variation in the delay estimates and the instantaneous SNR as the online algorithm progresses as a function of the number of data frames processed.

5. RELATED WORK

[11] introduced a delay approach similar to the one used here. Their model is justified on logical grounds rather than

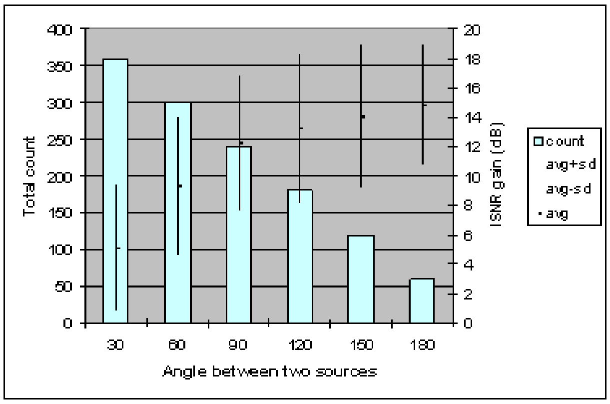

Fig. 2. SNR separation results as a function of the difference in angles of arrival for anechoic data set.

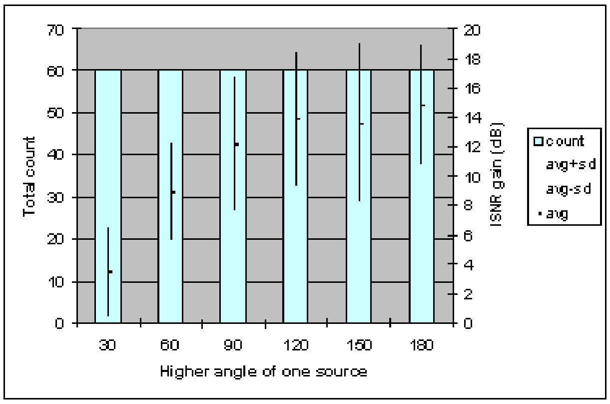

Fig. 3. SNR separation results as a function of the higher angle of one of the two sources for anechoic data set.

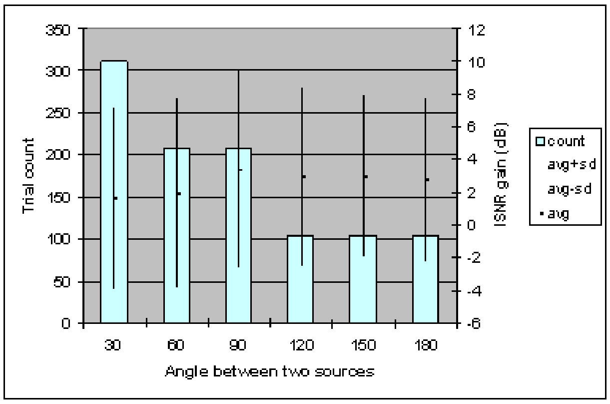

Fig. 4. SNR separation results as a function of the difference in angles of arrival for echoic data set.

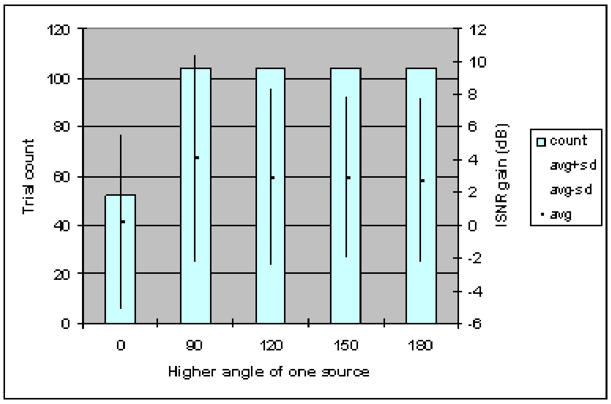

Fig. 5. SNR separation results as a function of the higher angle of one of the two sources for echoic data set.

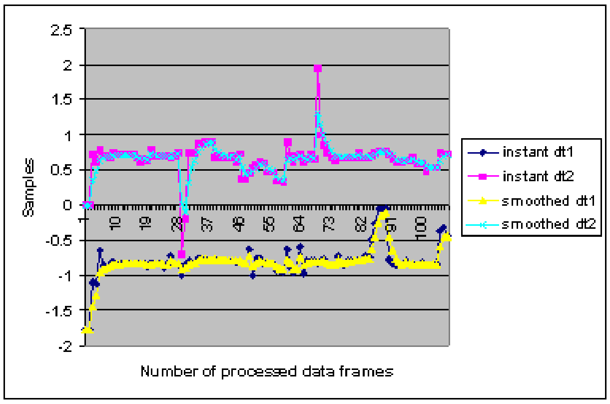

Fig. 6. Evolution of absolute and smoothed delay parameters (in samples) as a function of the number of frames processed for an anechoic example.

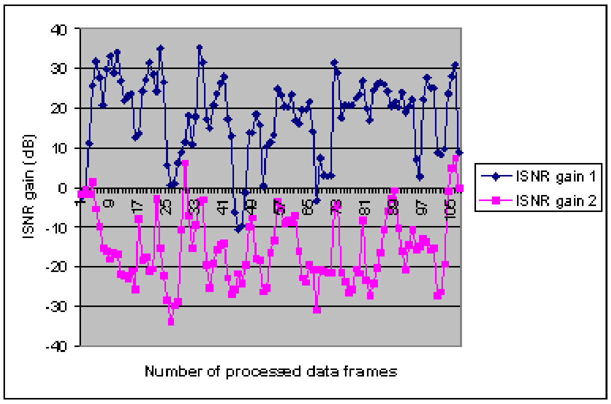

Fig. 7. Evolution of the instantaneous SNR for the example in Figure 6.

being derived from first principles. One difference with this work in the compensation for attenuations due to differences in the calibration of the microphones. In [11], this is taken care by a deconvolution stage using filters of of about a thousand taps. This is expensive computationally and may be hard to implement in real-time at the present computational load. Another difference, the direct path delay estimation is solved by a double filtering, which may be less efficint computationally. The work has a good discussion about the role of non-stationarity in the sources for achieving separation. This is an important observation for the way computation is to be decomposed for an effective implementation.

[12] uses a mixing model similar to that in equation 2. However, the decorrelation criterion it uses is computed for integer delays, therefore the technique assumes a large distance between the microphones. Under such conditions the assumption about sources being far-field may not hold well, and the model may not be a good approximation. Therefore, [12] further develops the model to include higher order tap coefficients. The overall model in that case suggests being contrained to a particular physical situation. No extensive results were presented to prove the contrary.

Another set of related spatial filtering techniques are antenna array processing techniques [1, 13]. Such techniques assume as given information about the microphone array layout. The DAC separation approach does not necessarily make this assumtion, however weaker information such as the distance or a bound on the distance bewteen sensors can help during the parameter estimation phase. Of particular interest for comparison are robust beamforming techniques [14]. Adaptive beamformers assume a known direction of arrival, while the present source separation technique estimates them. Separation is then performed in a manner similar to the way a Griffiths-Jim beamformer's blocking matrix obtains estimates of the noise by exploiting the additional channels available in a microphone array. Directional notches are placed in the direction of sources of interference.

Improvements from the beamforming literature, as discussed in [14], could be applied as well in order to deconvolve source estimates. Recent source separation approaches attempt to combine independent component analysis (ICA) or blind source separation (BSS) and elements of a beamformer in order to improve the performance of ICA/BSS techniques (see for example [4]).

6. CONCLUSIONS

This paper presented a real-time implementation of delay and attenuation based source separation model. The implementation has been tested on more than one thousand mixtures of voices recorded in real anechoic and echoic environments. The performance of the system is good on anechoic data. Although the algorithm is designed for anechoic environments, its complexity and performance on real data represent a basis for designing more complex approaches, as suggested in Section 2 to deal with reverberant environments.

One limitation of the approach is that it can only deal with a number of sources equal to the number of microphones. Recent work exploiting both time and frequency distributions can also deal with more sources than the number of sensors [15]. The filtering approach mentioned here applies more generally but has the drawback of requiring source variances.

Future work will address extensions of this work for echoic environments, along the lines of the general solution presented in Section 2. There is evidence that low order parameterizations of the model can buy slightly improved performance [9], but how this can be effectively implemented is still open. More complex parameterization, on the other side, appear ineffective because even slight deviations from true parameters results in significant performance degradation.

7. REFERENCES

[1] M. Brandstein and D. Ward, Eds., *Microphone Arrays*, Springer, 2001.

[2] K. Torkolla, "Blind separation for audio signals: Are we there yet?," in *First International Workshop on Independent component analysis and blind source separation*, Aussois, France, Jan. 1999, pp. 239–244.

[3] F. Asano and S. Ikeda, "Evaluation and real-time implementation of blind source separation system using time-delayed decorrelation," in *Proceedings of the Second International Workshop on ICA and BSS*, P. Pajunen and J. Karhunen, Eds. 2000, Otamedia.

[4] S. Kurita, H. Saruwatari, S. Kajita, K. Takeda, and F. Itakura, "Evaluation of signal separation method using directivity pattern under reverberant conditions," in *Proceedings ICASSP*. 2000, IEEE Press.

[5] H. Saruwatari, S. Kurita, and K. Takeda, "Blind source separation combining frequency domain ica and beamforming," in *Proceedings ICASSP*. 2001, IEEE Press.

[6] Jrn Anemller and Birger Kollmeier, "Amplitude modulation decorrelation for convolutive blind source separation," in *Proceedings of the second international workshop on independent component analysis and blind signal separation*, Petteri Pajunen and Juha Karhunen, Eds., Helsinki, Finland, June 19–22 2000, pp. 215–220.

[7] Justinian Rosca, Joseph Ó Ruanaidh, Alexander Jourjine, and Scott Rickard, "Broadband direction-of-arrival estimation based on second order statistics," in *Advances in Neural Information Processing Systems 12*, S.A. Solla, T.K. Leen, and K.-R. Müller, Eds. 2000, pp. 775–781, MIT Press.

[8] Tim Laakso, Vesa Valimaki, Matti Karjalainen, and Unto Laine, "Splitting the unit delay," *IEEE Signal Processing Magazine*, pp. 30–60, 1996.

[9] R. Balan, J. Rosca, and S. Rickard, "Robustness of parametric source demixing in echoic environments," in *submitted to ICA 2001*, 2001.

[10] W.H. Press and al., *Numerical Recipes in C*, Cambridge University Press, 1988.

[11] T. J. Ngo and N.A. Bhadkamkar, "Adaptive blind separation of audio sources by a physically compact device using second order statistics," in *First International Workshop on ICA and BSS*, Aussois, France, Jan. 1999, pp. 257–260.

[12] Y. Xiang, Y. Hua, S. An, and A. Acero, "Experimental investigation of delayed instantaneous demixer for speech enhancement," in *Proceedings ICASSP*. 2001, IEEE Press.

[13] V. Van Veen and Kevin M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, 1988.

[14] O. Hoshuyama and A. Sugiyama, *Microphone Arrays*, chapter Robust Adaptive Beamforming, pp. 87–110, Springer, 2001.

[15] A. Jourjine, S. Rickard, and O. Yilmaz, "Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2000, IEEE Press, June 5-9, 2000, Istanbul, Turkey.