

# Adaptive Blind Audio Signal Separation on a DSP

J. van de Laar, E.A.P. Habets, J.D.P.A. Peters, P.A.M. Lokkart  
Technische Universiteit Eindhoven, Dept. of Electrical Engineering,  
EH 3.27, P.O. Box 513, 5600 MB Eindhoven, The Netherlands  
Email: j.v.d.laar@tue.nl

**Abstract**—Blind Source Separation (BSS) deals with the problem of separating independent sources from their observed mixtures only while both the mixing process and original sources are unknown. Examples of BSS algorithms employed in acoustical applications can be found among others in audio conferencing systems. This paper describes the main ideas and implementation of an Adaptive Blind Signal Separation algorithm. In order to make the real-time implementation feasible, the BSS algorithm is based on a simplified mixing model (SMM). The input signals are reconstructed by assuming that they are statistically uncorrelated and imposing this constraint on the signal estimates. The nonstationarity of the input signals is used to restrict the set of solutions. The system is realized on a TI TMS320C6701 DSP and is capable of separating two independent simultaneously occurring audio signals in an ordinary acoustic environment in real-time and in an adaptive way. Finally, the separation performance of the algorithm is evaluated using benchmarks downloaded from the web and own real-world recordings.

**Keywords**—Blind Signal Separation, teleconferencing.

## I. INTRODUCTION

A growing number of researchers have been attracted to the Blind Signal Separation problem mainly concentrating on theory and simulations. In many cases one of the most important steps, i.e. the evaluation by experiments in real-world situations, is not carried out. BSS deals with the problem of separating independent sources from their observed linear convolutive mixtures only, while both the mixing acoustical transfer functions and the original sources are unknown. The complexity of the entire system is related to the room acoustics. The acoustic model of a room is very complex. In addition, due to the continuously changing nature of the acoustics, the (un)mixing system is time varying and therefore adaptive filters are required. This fact, together with the inherent computational complexity of BSS algorithms makes that most BSS algorithms are not suitable for real-time realization. Therefore, an adaptive BSS algorithm that is based on efficiency and simplicity is used. In this paper, the realization of an adaptive blind audio signal separation algorithm on a Texas Instruments TMS320C6701 DSP is described and the results of real-time experiments are given. Efficiency has been ob-

tained by using an appropriate way of block processing in the frequency domain. Simplicity has been obtained on the one hand by using a simplified mixing model. On the other hand, simplicity has been achieved by a defining a separation cost function based on second order statistics (SOS) and by using a relatively simple adaptation rule. Due to the non-stationarity of the source signals, only second order statistics are sufficient [4]. In this paper, we consider specifically the two-channel case.

The rest of this paper is organized as follows. First, the adaptive blind signal separation algorithm is described in section II. Section II-A explains the simplified mixing model. Section II-B describes the frequency domain translation of the problem and section II-C explains the estimation of the so-called Difference Room Impulse responses used in the SMM. Next, some implementation details are given in section III. After that, the experimental results are described in section IV and finally we give a brief summary and concluding remarks in section V.

## II. ADAPTIVE BLIND SIGNAL SEPARATION ALGORITHM

### A. BSS using a Simplified Mixing Model

The mixing process that takes place in an acoustic environment from two sources to two microphones contains four different acoustic impulse responses  $h_{ij}$ , where  $h_{ij}$  is the impulse response from source  $j$  to microphone  $i$ . The discrete-time two-channel case with inputs  $s_1$  and  $s_2$  and observed outputs  $x_1$  and  $x_2$  is depicted at the left hand side of Fig. 1. The relationship between the sources and the mixtures is described by:

$$\begin{pmatrix} x_1[n] \\ x_2[n] \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} * \begin{pmatrix} s_1[n] \\ s_2[n] \end{pmatrix} \quad (1)$$

Demixing of the observed signals by means of a BSS algorithm requires the inversion of a matrix containing four different acoustic impulse responses  $h_{ij}$ , ( $i, j = 1, 2$ ), each of which has to be estimated. The result of this demixing process is an estimation of the original sources  $s_1$  and  $s_2$ . The complexity of the system shown at the left hand side of Fig. 1 can be reduced by using a simplified mixing model (SMM), which is developed below. The above described

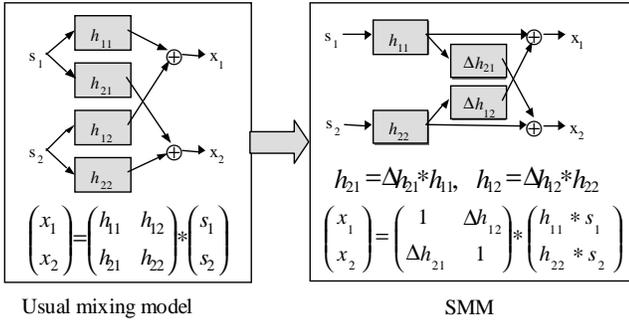


Fig. 1. The conventional BSS mixing model (left) and the simplified mixing model (right)

system is mathematically completely equivalent with the system depicted at the right hand side of Fig. 1.

The so-called difference room impulse responses (DRIR's)  $\Delta h_{ij}[n]$ 's are defined as  $(h_{jj}^{-1} * h_{ij})[n]$  for  $i, j = 1, 2$ . These  $\Delta h_{ij}$ 's often have to be non-causal double sided filters due to the inversion of the direct path room impulse responses (RIR's), which are typically non-minimum phase. In order to be realizable in a practical system, the DRIR's are shifted in time by  $\tau$  samples and then truncated. By doing so, the estimate

$$\Delta h_{ij} = \mathbf{Tr}\{(h_{jj}^{-1} * h_{ij})[n - \tau]\} \approx (h_{jj}^{-1} * h_{ij})[n - \tau] \quad (2)$$

becomes causal. The simplified mixing system can now be described as:

$$\begin{pmatrix} x_1[n - \tau] \\ x_2[n - \tau] \end{pmatrix} = \begin{pmatrix} \delta[n - \tau] & \Delta h_{12} \\ \Delta h_{21} & \delta[n - \tau] \end{pmatrix} * \begin{pmatrix} (h_{11} * s_1)[n] \\ (h_{22} * s_2)[n] \end{pmatrix} \quad (3)$$

The proper choice of the delay  $\tau$  depends on several factors, e.g., the wall reflection and the distance between sources and microphones. In particular, if the reverberation is weak and the audio source is located not too far from its microphone, the delay may be chosen close to zero since  $h_{ij}$  in this case shows minimum phase characteristics. The detailed experimental results can be found in [2]. For our experiments, the delay was set to zero. For simplicity reasons all experimental results in this paper are obtained for a delay of zero.

When the sensors are closely spaced, the acoustic transfer functions from a single source to two closely spaced microphones are very similar and therefore the DRIR's require only a relatively small number of coefficients ( $\approx 500$ ). It is important to note that we do not try to obtain the clean sources themselves, but the sources filtered by their correspondig direct paths (which can include reverberation, etc.) only. Compared to the usual mixing model, the resulting output signals of this simplified approach are

estimates of the source signals as they sound in the neighborhood of the microphones, i.e.  $\tilde{s}_1[n] = (s_1 * h_{11})[n]$  and  $\tilde{s}_2[n] = (s_2 * h_{22})[n]$  respectively. This can be an advantage since the recovered signals sound more natural because each of them reflects the sound field at the position of a microphone if only one source would have been present. Now, the demixing of the observed signals by means of a BSS algorithm requires the inversion of a matrix containing only two DRIR's that have to be estimated. This is done by an adaptive algorithm, as is described in the next sections.

## B. Frequency domain transformation

It is well known from literature that a filtering or convolution operation in time domain can be performed efficiently in frequency domain by a complex multiplication. In order to meet this efficiency an appropriate way of block processing and the FFT are used. This results in the overlap-save method [3]. The 'infinitely' long input signal is cut into blocks, each of which is convolved with the impulse response, resulting in blocks of the output signal. The convolution, which is a linear operation, is performed in the frequency domain. The input signal blocks are transformed to the frequency domain by FFT's, which are cyclic operations. In order to perform a linear convolution with a cyclic one, the input blocks need an overlap and the impulse responses have to be padded with zeros. The last part of the output block (equal to the number of overlapped samples) is a clean linear convolution. When applying this technique to adaptive filters, the update will take place in the frequency domain and an extra constraint is necessary to set the last part of the time domain impulse response vector to zero.

Since the frequency bins are more or less independent from each other, from now on we focus on one frequency bin for simplicity. Assuming zero delay, the frequency domain counterpart of Eq. 3 for one input data block (block index is left out for simplicity) is given by:

$$\begin{pmatrix} X_1(\omega) \\ X_2(\omega) \end{pmatrix} = \begin{pmatrix} 1 & \Delta H_{12} \\ \Delta H_{21} & 1 \end{pmatrix} \cdot \begin{pmatrix} \tilde{S}_1(\omega) \\ \tilde{S}_2(\omega) \end{pmatrix} \quad (4)$$

where

$$\begin{pmatrix} \tilde{S}_1(\omega) \\ \tilde{S}_2(\omega) \end{pmatrix} = \begin{pmatrix} (H_{11}(\omega) \cdot S_1(\omega)) \\ (H_{22}(\omega) \cdot S_2(\omega)) \end{pmatrix} \quad (5)$$

Here,  $\omega = 0, \dots, \frac{(N-1)}{N}2\pi$  denotes the frequency and  $X_i(\omega)$  represents the  $N$ -point DFT of the considered input block of the  $i$ th microphone signal. Thus, once estimates  $\Delta \hat{H}_{ij}$ 's of the DRIR's have been obtained, the observed

signals can be demixed for each frequency bin as follows:

$$\begin{aligned} \begin{pmatrix} Y_1(\omega, p) \\ Y_2(\omega, p) \end{pmatrix} &= \begin{pmatrix} 1 & \Delta\hat{H}_{12} \\ \Delta\hat{H}_{21} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} X_1(\omega) \\ X_2(\omega) \end{pmatrix} \\ &= \Lambda^{-1} \cdot \begin{pmatrix} 1 & -\Delta\hat{H}_{12} \\ -\Delta\hat{H}_{21} & 1 \end{pmatrix} \begin{pmatrix} X_1(\omega) \\ X_2(\omega) \end{pmatrix} \end{aligned} \quad (6)$$

where  $Y_i(\omega) = \hat{S}_i$  and  $\Lambda = 1 - \Delta\hat{H}_{12} \cdot \Delta\hat{H}_{21}$ . For convenience, the mixing and demixing models of equations 4 and 6, are written in the more concise matrix-vector notation as follows:

$$\underline{X} = \mathbf{H} \cdot \underline{\tilde{S}} \Rightarrow \underline{Y} = \hat{\mathbf{H}}^{-1} \cdot \underline{X} \quad (7)$$

### C. Estimation of the DRIR's

The task of the BSS algorithm now boils down to identifying the DRIR's and then demixing the observed signals. The reconstruction of the input signals is based on the assumption that they are mutually statistically uncorrelated (because the input signals are mutually statistically independent). Therefore, a cost function  $J(\omega)$  that is a function of the cross-correlations between the output signals, is defined for each frequency bin as follows:

$$\begin{aligned} J(\omega) &= \sum_{i \neq j} | \langle (\underline{Y}(\omega) \cdot \underline{Y}^h(\omega))_{ij} \rangle |^2 \\ &= \sum_{i \neq j} | \langle (\hat{\mathbf{H}}^{-1}(\omega) \cdot \underline{X}(\omega) \cdot \underline{X}^h(\omega) \cdot \hat{\mathbf{H}}^{-h}(\omega))_{ij} \rangle |^2 \\ &= \sum_{i \neq j} | \langle (\hat{\mathbf{H}}^{-1}(\omega) \cdot \mathbf{R}_X(\omega) \cdot \hat{\mathbf{H}}^{-h}(\omega))_{ij} \rangle |^2 \end{aligned} \quad (8)$$

Here

$$\mathbf{R}_X(\omega) = \begin{pmatrix} \langle X_1(\omega)X_1^*(\omega) \rangle & \langle X_1(\omega)X_2^*(\omega) \rangle \\ \langle X_2(\omega)X_1^*(\omega) \rangle & \langle X_2(\omega)X_2^*(\omega) \rangle \end{pmatrix} \quad (9)$$

and  $\langle \cdot \rangle$  denotes the expectation operator. The summation is running over the non-diagonal elements  $i \neq j$  of the 2x2 (squared) cross output power matrix. The 2x2 matrix  $\mathbf{R}_X(\omega)$  is the cross power spectra matrix of the microphone signals. The (DRIR) filter coefficients are obtained by minimizing this cost-function. However, for fixed time no unique solution exists that minimizes the cost function, because statistical uncorrelatedness of the signals does not imply statistical independence. Second order statistics are not sufficient to find the true unique solution for stationary sources. However, in practical non-stationary situations second order statistics can solve this problem [5]. The power matrix  $\mathbf{R}_X(\omega)$  is now time-varying and therefore there is a different set of solutions for each time instant. The true solution is included in the intersection of all sets

at different times. Whether the true solution can be determined as one point or not depends on the non-stationarity of the signals. By iteratively minimizing  $J$  we expect the algorithm to converge to the true solution. Due to the continuously changing nature of the mixing model, and therefore also the unmixing model, an adaptive algorithm is required in order to track the variations in the mixing model. Here, a gradient descent adaptive algorithm is used for each frequency bin  $\omega$  separately:

$$\Delta\hat{H}_{ij}(\omega) := \gamma \cdot \Delta\hat{H}_{ij}(\omega) - \mu_{ij}(\omega) \cdot \left( \frac{\delta J(\omega)}{\delta \Delta\hat{H}_{ij}(\omega)} \right)^* \quad (10)$$

for  $i, j = 1, 2$  and  $i \neq j$ , where  $\gamma$  is a leakage factor ( $0 < \gamma < 1$ ) that improves the stability and robustness of the algorithm, and the  $\mu_{ij}(\omega)$ 's are the adaptation constants. The leakage factor ensures that the adaptive weights of one of the DRIR's go to zero when the corresponding source is not present. In [8] it is shown that the derivative of the cost function with respect to the  $\Delta\hat{H}_{ij}$ 's is as follows:

$$\begin{aligned} \left( \frac{\delta J(\omega)}{\delta \Delta\hat{H}_{ij}(\omega)} \right)^* &= - \sum_{p \neq q} \{ (\hat{\mathbf{H}}^{-1}(\omega) \cdot E_{ij} \cdot \mathbf{R}_Y(\omega))_{qp}^* \cdot (\mathbf{R}_Y)_{pq}^*(\omega) \\ &\quad + (\hat{\mathbf{H}}^{-1}(\omega) \cdot E_{ij} \cdot \mathbf{R}_Y(\omega))_{qp}^* \cdot (\mathbf{R}_Y)_{pq}(\omega) \} \end{aligned} \quad (11)$$

with  $\mathbf{R}_Y(\omega) = \hat{\mathbf{H}}^{-1}(\omega) \cdot \mathbf{R}_X(\omega) \cdot \hat{\mathbf{H}}^{-h}(\omega)$ . In this equation the 2x2 selection matrix  $E_{ij}$  is a matrix whose  $i, j$ -th element is one and the others are zero.

The time-varying power matrix  $\mathbf{R}_X$  is updated by means of exponential averaging as follows:

$$\mathbf{R}_X(\omega) := \alpha \cdot \mathbf{R}_X(\omega) + (1 - \alpha) \cdot \underline{X}(\omega) \cdot \underline{X}^H(\omega) \quad (12)$$

where  $\alpha$  is a forgetting factor that is usually chosen close to 1. The algorithm assumes stationary input signals during the time when the cross power spectra of the input signals are estimated. It is obvious that a speech signal is a non-stationary signal. However, a speech signal can be regarded as a stationary stochastic process if the time segment is small enough ( $\approx 20$  ms).

As indicated earlier, in order to make the filters perform linear convolutions instead of cyclic ones, part of the coefficient vectors have to be set to zero. By putting the gradients for all frequencies in a vector, this constraint can be implemented as a constraint on the gradient update as follows:

$$\frac{\delta J}{\delta \Delta\hat{H}_{ij}} := \mathcal{F} \cdot \mathcal{Z} \cdot \mathcal{F}^{-1} \cdot \frac{\delta J}{\delta \Delta\hat{H}_{ij}} \quad (13)$$

with  $\mathcal{F}$  the Fourier matrix and  $\mathcal{Z}$  a diagonal matrix with  $(\mathcal{Z})_{ii} = 0$  for that part of the weight vector that has to

be constrained to zero, the other diagonal elements are 1. Finally it is noted that the permutation problem (different frequency bins of different estimates of the sources can be hustled) is solved by pushing this extra constraint on the adaptive weights for every iteration [5].

### III. ALGORITHM IMPLEMENTATION

The described algorithm has been implemented on a TMS320C6701 Evaluation Module board from Texas Instruments [6]. This EVM has been installed in a PCI slot of a host PC running Windows NT 4.0 and contains all the required hardware. It is equipped with TMS320C6701 floating-point DSP running at a processor speed of 133 MHz. The CS4231A [1] 16-bit stereo audio codec on the EVM is used to sample the two input signals and to convert the two separated output signals from digital to analogue. The sampling frequency of the system is 8 kHz. The codec is connected to the 'C6701 by means of a serial interface. A graphical user interface (GUI) has been made. This GUI runs on the host PC and is used for starting, stopping or re-setting the algorithm, setting the parameters, choosing the sound signal the user wants to listen to, etc. In order to realize this GUI, the RTDX (Real Time Data eXchange) protocol from TI has been used. The length of the input data blocks equals 1024 samples, while the filter lengths equal 512, hence 512 new output samples are computed for each block and the overlap is 50 %. Fig. 2 shows a flow chart of the algorithm. The algorithm has been programmed mainly in C. The variable  $P$  in the flow chart of Fig. 2 indicates the number of the block being processed. The first 5 input blocks are used to make an initial estimation of the power matrix. The next blocks are used to update the power matrix using Eq. 12.

### IV. EXPERIMENTAL RESULTS

Experiments have been performed in order to evaluate the real-time adaptive blind source separation system. Here, the results of real-world experiments are described. Both real-world benchmark recordings that have been downloaded from the web and own real-world recording have been evaluated. Since the system is blind and no reference signals are available, it is not straightforward to define a performance measure. Since the cost function  $J$  defined in Eq. 8 is related to the cross-correlations between the outputs, it is used as a separation performance measure. In Fig. 3 the cost function ( $10 \cdot \log J$ ) is plotted as a function of time. This is done for three benchmarks which can be downloaded from the locations listed in [7]. It can be seen that our BSS algorithm decreases the cost function with approximately 25 dB. The peaks occurring in the figures, are due to the non-stationarity of the input signals. It

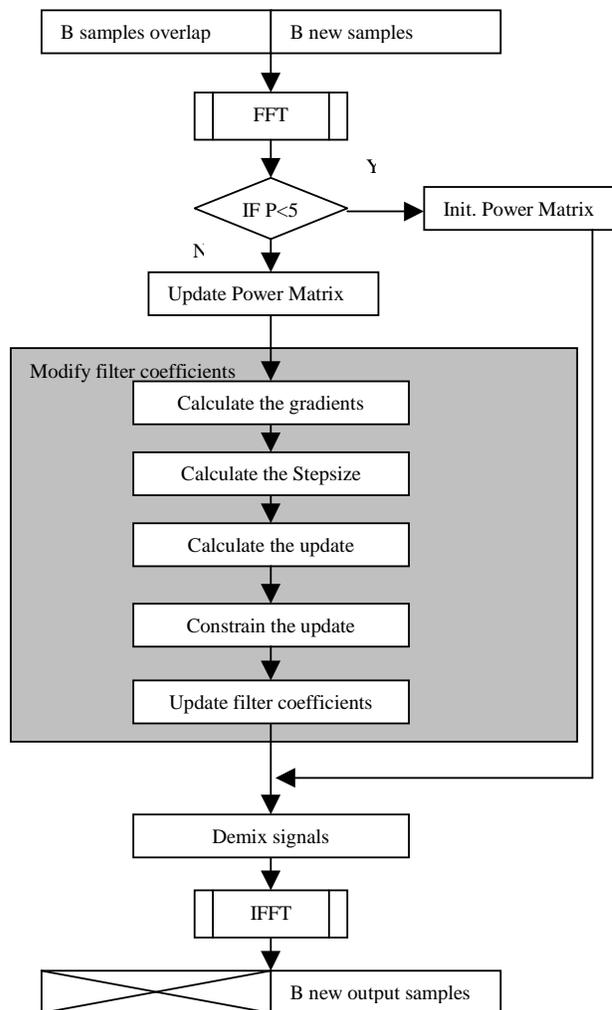


Fig. 2. Flow chart of adaptive BSS algorithm

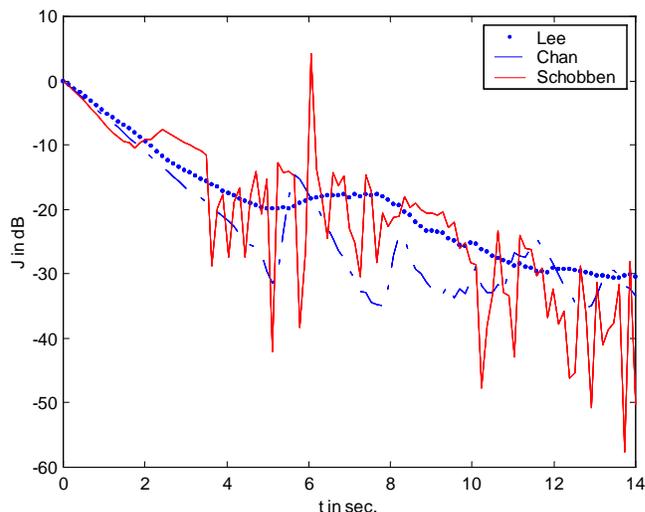


Fig. 3. Decrease in cost function for benchmarks available on the web

can be seen that the cost function has decreased 20 dB after approximately 3 seconds. The perceptual separation is also a very important measure for signal separation. From informal listening tests it can be concluded that the intelligibility of the desired source is improved. We also evaluated the algorithm in a room with dimensions 5.2m by 3.8m by 3.4m (length  $\times$  width  $\times$  height). The acoustical behavior of the room can be changed by the use of wall panels with different reflection properties and the use of curtains in front of the windows. Two omni-directional microphones are used, which are placed at a distance of 10 cm from each other. The distance between the sources and the microphones is chosen to be 60 cm. The distance between the loudspeakers is 50 cm. Music was played by one speaker and speech by the other. The decrease in cost function for this situation is plotted in Fig. 4. Listening tests re-

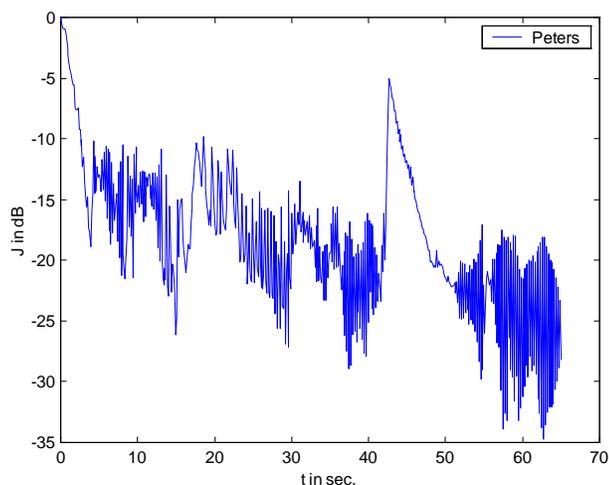


Fig. 4. Decrease in cost function for real-world recordings

veal that the separation is not very good in this case, even though Fig. 4 shows a 20 dB decrease in the cost function. The main problem is probably that the experimental setup in the audio laboratory produces rather complex acoustical mixtures with too much reverberation. Other experiments revealed that in this case the algorithm also has problems in separating two speech signals.

## V. CONCLUSION

An adaptive blind signal separation algorithm that is based on a simplified mixing model (SMM) using only second order statistics has been realized on a TI TMS320C6701 DSP. The main advantages of the algorithm are:

- Less parameters have to be estimated because less filters have to be identified and they have a relatively small number of coefficients

- The demixed signals sound more natural because not the original sources are recovered, but the sources as they sound in the neighborhood of the microphones
- The algorithm can be implemented in real-time

The separation performance of the algorithm has been evaluated using benchmarks downloaded from the web and self-made real-world recordings. The experiments show that the algorithms works well when applied to the benchmarks which are claimed to be real-world recordings. However, its performance is worse for the self-made real-world recordings. Probably there is too much reverberation in the room. All experiments have been carried out with zero delay in the difference impulse response filters. However, the acoustical transfer functions become more and more non-minimum phase as the amount of reverberation increases. Therefore, experiments with delay larger than zero must be performed in the future and are likely to give better separation results.

## REFERENCES

- [1] B.G. Carlson, "TMS320C6000 McBSP Interface to the CS4231A Multimedia Audio Codec", Texas Instruments Incorporated, SPRA477, Dec. 1998.
- [2] He, P., Sommen, P.C.W. and Yin, B. "A realtime DSP blind signal separation experimental system based on a new simplified mixing model", *Proc. of EUROCON'2001*, Bratislava, Slovak Republic, July, 2001.
- [3] A.V. Oppenheim and W. Schaffer, "Digital Signal Processing" New Jersey, Prentice-Hall, 1975.
- [4] L. Parra and C.Spence, "Convolutional blind source separation based on multiple decorrelation", *Proc. of NNSP98*, Cambridge, UK, September, 2000.
- [5] L. Parra and C.Spence, "Convolutional Blind Separation of Non-Stationary Sources", *IEEE Transactions on speech and audio processing*, 2000. Vol. 8, No. 3, p. 320-327.
- [6] Texas Instruments, "TMS320C6201/6701 Evaluation Module Technical Reference", Texas Instruments Incorporated, SPRU305, Dec. 1998.
- [7]
  - T. Lee, <http://www.cnl.salk.edu/~tewon/>
  - D. Chan, <http://www-sigproc.eng.cam.ac.uk/oldusers/dcbcl/research/demo.html>
  - D. Schobben, <http://www.esp.ele.tue.nl/onderzoek/daniels/BSS.html>
- [8] Yin, B. and Sommen, P.C.W., "A new convolutional blind signal separation algorithm based on second order statistics using a simplified mixing model", *Proc. Eusipco 2000, European Signal Processing Conference*, Tampere, Finland, 2000.