Audio Engineering Society

Convention e-Brief

Presented at the 147th Convention
2019 October 16 – 19, New York

FAST TIME DOMAIN STEREO AUDIO SOURCE SEPARATION USING FRACTIONAL DELAY FILTERS

Oleg Golokolenko[1] and Gerald Schuller[1]

[1] *Technische Universität Ilmenau, Department of Media Technology, 98693 Ilmenau, Germany*

Correspondence should be addressed to Oleg Golokolenko (`Oleg.Golokolenko@tu-ilmenau.de`)

ABSTRACT

Our goal is a system for the separation of two speakers during teleconferencing or for hearing aids. To be useful in real time, we want it to work online with as low delay as possible. Proposed approach works in time domain, using attenuation factors and fractional delays between microphone signals to minimize cross-talk, the principle of a fractional delay and sum beamformer. Compared to other approaches this has the advantage that we have lower computational complexity, no system delay and no musical noise like in frequency domain algorithms. We evaluate our approach on convolutive mixtures generated from speech signals taken from the TIMIT data-set using a room impulse response simulator.

1 Introduction, Previous Approaches

Our system is for applications where we have two microphones and want to separate two audio sources. This could be for instance a teleconferencing scenario with a stereo webcam in an office and two speakers around it, or for hearing aids, where low computational complexity is important.

Previous approaches: An early previous approach is Independent Components Analysis (ICA). It can unmix a mix of signals with no delay in the mixture. It finds the coefficients of the unmixing matrix by maximizing non-gaussianity or maximizing the Kullback-Leibler Divergence [1]. But for audio signals and a stereo microphone pair we always have propagation delay, in general convolutions with the room impulse responses,in the mix. Approaches to deal with it often apply the Short Time Fourier Transform (STFT) to the signals, e.g., AuxIVA [2] and ILRMA [3]. This converts the signal delay into a complex valued factors in the STFT subbands, and a (complex valued) ICA can be applied in the resulting subbands (e.g. [4]).

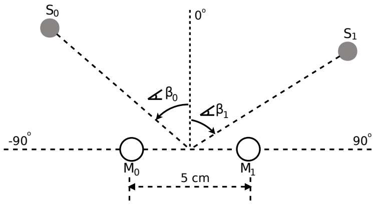


Fig. 1: Setup of loudspeakers and microphones in the simulation.

Problem: A problem that occurs here is a permutation in the subbands, the separated sources can appear in different orders in different subbands; and the gain for different sources in different subbands might be different, leading to a modified spectral shape, a spectral flattening. Besides, we have a signal delay resulting from applying an STFT. It needs the assembly of the

signal into blocks, which needs a system delay corresponding to the block size.
Time domain approaches, like TRINICON [5], or approaches that use the STFT with short blocks and more microphones [6], have the advantage that they don't have a large blocking delay of the STFT, but they usually have a higher computational complexity, which makes them hard to use on small devices.

2 Proposed Approach

In the proposed approach, we focus on the case of teleconferencing applications, for the separation of two speakers, or a speaker and a noise source, in a small office environment, where the speakers are not too far from a stereo microphone pair, as in available stereo webcams. To avoid the processing delays associated with frequency domain approaches, we use a time domain approach. Instead of using FIR filters, we use IIR filters, which are implemented as fractional delay allpass filters [7], with an attenuation factor, the principle of a fractional delay and sum or adaptive beamformer [8]. To achieve small algorithmic delay, we don't do a dereverberation either we focus on the crosstalk minimization instead. In effect we model the Relative Transfer Function between the two microphones by an attenuation and a pure fractional delay [9].
We assume a mixture recording from 2 sound sources (S_0 and S_1) made with 2 microphones (M_0 and M_1). The sound sources are assumed to be in fixed positions as shown in Figure 1. In order to avoid the need for modeling of non-causal impulse responses the sound sources have to be in different half-planes of the microphone pair (left-right).
Instead of the commonly used STFT, we use the z-transform for the mathematical derivation, because it does not need the decomposition of the signal into blocks, with its associated delay. This makes it suitable for a time domain implementation with no signal delay. We use capital letter to denote z-transform domain signals.
Let us define $s_0(n)$ and $s_1(n)$ as our two time domain sound signals, and their z-transforms as $S_0(z)$ and $S_1(z)$. The two microphone signals are $m_0(n)$ and $m_1(n)$, and their z-transforms are $M_0(z)$ and $M_1(z)$ (Figure 2).
The Room Impulse Responses (RIRs) from the i's source to the j's microphone are $h_{i,j}(n)$, and their z-transform $H_{i,j}(z)$. Thus, our convolutive mixing system can be described in the z-domain as

$$\begin{bmatrix} M_0(z) \\ M_1(z) \end{bmatrix} = \begin{bmatrix} H_{0,0}(z) & H_{1,0}(z) \\ H_{0,1}(z) & H_{1,1}(z) \end{bmatrix} \cdot \begin{bmatrix} S_0(z) \\ S_1(z) \end{bmatrix} \tag{1}$$

In simplified matrix multiplication we can rewrite Equation (1) as

$$\boldsymbol{M}(z) = \boldsymbol{H}(z) \cdot \boldsymbol{S}(z) \tag{2}$$

For an ideal sound source separation we would need to invert the mixing matrix $\boldsymbol{H}(z)$. Hence, our sound sources could be calculated as

$$\boldsymbol{S}(z) = \boldsymbol{H}^{-1}(z) \cdot \boldsymbol{M}(z) \Rightarrow$$

$$\begin{bmatrix} S_0(z) \\ S_1(z) \end{bmatrix} = \begin{bmatrix} H_{1,1}(z) & -H_{1,0}(z) \\ -H_{0,1}(z) & H_{0,0}(z) \end{bmatrix} \cdot \frac{1}{\det(\boldsymbol{H}(z))} \cdot \begin{bmatrix} M_0(z) \\ M_1(z) \end{bmatrix} \tag{3}$$

Since $\det(\boldsymbol{H}(z))$ and diagonal elements of the inverse matrix are linear filters which do not contribute to the unmixing, we can neglect them for the separation, and bring them to the left side of eq. (3). This results in

$$\begin{bmatrix} H_{1,1}^{-1}(z) & 0 \\ 0 & H_{0,0}^{-1}(z) \end{bmatrix} \cdot \begin{bmatrix} S_0(z) \\ S_1(z) \end{bmatrix} \cdot \det(\boldsymbol{H}(z)) =$$

$$= \begin{bmatrix} 1 & -H_{1,1}^{-1}(z) \cdot H_{1,0}(z) \\ -H_{0,0}^{-1}(z) \cdot H_{0,1}(z) & 1 \end{bmatrix} \cdot \begin{bmatrix} M_0(z) \\ M_1(z) \end{bmatrix} \tag{4}$$

where $H_{1,1}^{-1}(z) \cdot H_{1,0}(z)$ and $H_{0,0}^{-1}(z) \cdot H_{0,1}(z)$ are now Relative Transfer Functions.
Next we approximate these Relative Transfer Functions by fractional delays by d_i samples and attenuation factors a_i,

$$H_{i,i}^{-1}(z) \cdot H_{i,j}(z) \approx a_i \cdot z^{-d_i} \tag{5}$$

where $i, j \in 0, 1$.
For the fractional delays by d_i samples we use the fractional delay filter in the next section (2.1). Note that for simplicity we keep the linear filter resulting from the determinant and from the matrix diagonal $H_{i,i}(z)$ on the left hand side, meaning there is no dereverberation. The signal flowchart of convolutive mixing and demixing process can be seen in Figure 2.

2.1 The fractional delay allpass filter

In order to implement Relative Transfer Functions (eq. 5), we use the fractional delay allpass filter. As a result we obtain IIR filter out of a single fractional delay coefficient, needed for good cross-talk cancellation.

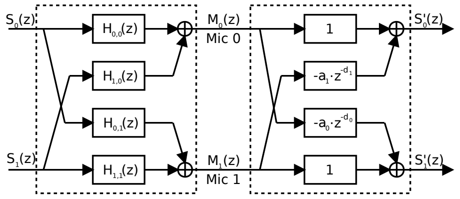


Fig. 2: Signal block diagram of convolutive mixing and demixing process.

In [7], a practical method for designing fractional delay allpass filters based on IIR filters with maximally flat group delay is described.

We use the following equations to obtain the coefficients for our fractional delay allpass filter, for a fractional delay $\tau = d_i$. Its transfer function in the z-domain is $A(z)$, with

$$A(z) = \frac{z^{-L} D(\frac{1}{z})}{D(z)}, \text{ where } D(z) \text{ is of order } L = \lceil \tau \rceil,$$

defined as follows:

$$D(z) = 1 + \sum_{n=1}^{L} d(n) z^{-n}$$

The filter $d(n)$ is generated as:

$$d(0) = 1, \quad d(n+1) = d(n) \cdot \frac{(L-n)(L-n-\tau)}{(n+1)(n+1+\tau)}$$

for $0 \leq n \leq (L-1)$.

2.2 Objective function

As objective function, we use a function $D(P_0, P_1)$ which is derived from the Kullback-Leibler Divergence (KLD),

$$D_{KL}(K||Q) = \sum_n K(n) \log \left(\frac{K(n)}{Q(n)} \right) \tag{6}$$

where $K(n)$ and $Q(n)$ are probability distributions of microphones signals, and n runs over the discrete distributions. In order to make the computation faster we avoid computing histograms. Instead of the histogram we use the normalized magnitude of the time domain signal itself,

$$P_i(n) = \frac{|\boldsymbol{s}'_i(n)|}{|||\boldsymbol{s}'_i|||_1} \tag{7}$$

where s'_i is the unmixed time domain signal and n now is the sample index. Notice, that $P_i(n)$ has similar properties with that of a probability, namely:

1. $P_i(n) \geq 0, \forall n$.
2. $\sum_{n=0}^{\infty} P_i(n) = 1$.

with $i = 0, 1$. Instead of using the KL-Divergence directly, we turn our objective function into a symmetric function by using the sum $D_{KL}(K||Q) + D_{KL}(Q||K)$, since this makes separation more stable between the two channels. Hence, our resulting objective function $D(P_0, P_1)$ is:

$$D(P_0, P_1) = -\sum_n \left[P_0(n) \log \left(\frac{P_0(n)}{P_1(n)} \right) + \right. \\ \left. + P_1(n) \log \left(\frac{P_1(n)}{P_0(n)} \right) \right] \tag{8}$$

In order to apply minimization instead of maximization, the negative value of $D(P_0, P_1)$ has to be taken.

Algorithm 1 Optimization algorithm

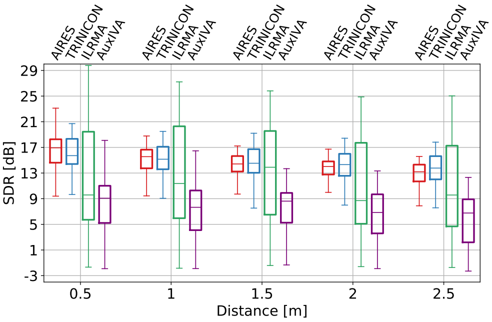
```
INITIALIZATION
X                                   ← convolutive mixture
init_coeffs = [1.0, 1.0, 1.0, 1.0]            ←
        initial guess for separation coefficients
coeffweights          ← weights for random search
coeffs = init_coeffs        ← separation coefficients
negabskl_0 = negabskl(coeffs, X)              ←
calculation of KLD
OPTIMIZATION ROUTINE
loop:
coeffvariation=(random(4)*coeffweights)       ←
        random variation of separation coefficients
negabskl_1 = negabskl(coeffs+coeffvariation, X) ←
                        calculation of new KLD
if negabskl_1 < negabskl_0 then
    negabskl_0 = negabskl_1
    coeffs = coeffs+coeffvariation            ←
                update separation coefficients

```

2.3 Optimization

A widespread optimization method for BSS is Gradient Descent. This has the advantage that it finds the "steepest" way to an optimum, but it requires the computation of gradients, and gets easily stuck in local minima or is slowed down by "narrow valleys" of the objective function. Hence, for the optimization of our coefficients

we use a novel optimization of "random directions", similar to "Differential Evolution" [10].
Instead of differences of coefficient vectors for the update, we use a weight vector to model the expected variance distribution of our coefficients. This leads to a very simple yet very fast optimization algorithm, which can also be easily applied to real time processing, which is important for real time communications applications. The algorithm starts with a fixed starting point [1.0, 1.0, 1.0, 1.0], which we found to lead to robust convergence behaviour. Then it perturbs the current point with a vector of uniformly distributed random numbers between -0.5 and +0.5 (the random direction), element-wise multiplied with our weight vector. If this perturbed point has a lower objective function value, we choose it as our next current point, and so on (Algorithm 1).



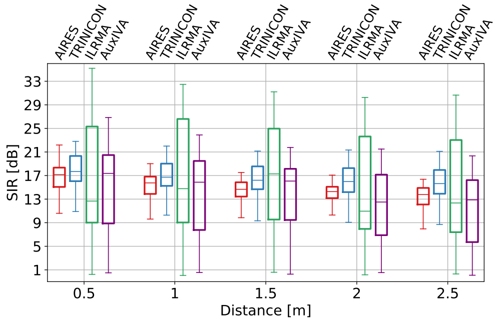
(a) $RT_{60} = 0.1s$

Fig. 3: Box-plots for the SDR obtained by AIRES and previous BSS approaches applied to simulated data.

3 EXPERIMENTAL RESULTS

In this section, we evaluate the proposed time domain separation method, which we call **AIRES** (time domAIn fRactional dElay Separation), by using simulated room impulse responses and different speech signals from the TIMIT data-set [11]. In order to evaluate the performance of **AIRES**, comparison with State-of-the-Art BSS algorithms has been done, namely, time domain TRINICON [5], frequency domain AuxIVA [2] and ILRMA [3]. An implementation of the TRINICON BSS has been received from its authors. Implementations of AuxIVA and ILRMA BSS were taken from [12] and [13], respectively. The experiment has been performed using MATLAB R2017a on a laptop with CPU Core i7 8-th Gen. and 16Gb of RAM.

The room impulse response simulator based on the image model technique [14] was used to generate room impulse responses. For the simulation setup the room size have been chosen to be $7m \times 5m \times 3m$. The microphones were positioned in the middle of the room at $[3.475, 2.0, 1.5]m$ and $[3.525, 2.0, 1.5]m$, and the sampling frequency was 16kHz. Ten pairs of speech signals were randomly chosen from the whole TIMIT data-set and convolved with the simulated RIRs. For each pair of signals, the simulation has been performed at 35 random angle positions of the sound sources relatively to microphones, for 5 different distances and 3 reverberation times (RT_{60}).
The visualization of the setup can be seen in Figure 1. The evaluation of the separation performance was done objectively by computing the Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR) measures and the computation time. The results are shown in Figures 3 and 4.



(a) $RT_{60} = 0.1s$

Fig. 4: Box-plots for the SIR obtained by AIRES and previous BSS approaches applied to simulated data.

Table 1: Comparison of average computation time $\tilde{t}$.

BSS	**AIRES**	TRINICON	ILRMA	AuxIVA
$\tilde{t}$	0.07s	14.55s	1.98s	1.33s

The obtained results show a good performance of our approach for reverberation times smaller than 0.2s. For

RT_{60} = 0.05s the average SDR measure over all distances is 18.62dB and for RT_{60} = 0.1s it is 14.4dB. For a reverberation time RT_{60} = 0.2s **AIRES** is on the second place after TRINICON with SDR = 9.22dB.
The average computation time over all simulations can be seen in Table 1. Thus, one can conclude that **AIRES** is comparable with frequency domain State-of-the-Art BSS in terms of separation performance and outperforms all of them in terms of computation time. Moreover, from Figure 4 it can be seen that **AIRES** shows more stable behaviour as TRINICON in contrast to ILRMA and AuxIVA.
By listening to the results we found that an SIR of about 8dB results in good speech intelligibility, and our approach indeed features no unnatural sounding artifacts.

4 CONCLUSIONS

In this paper, we presented a fast time domain blind source separation technique based on the estimation of IIR fractional delay filters to minimize crosstalk between two audio channels. We have shown that estimation of the fractional delays and attenuation factors results in a fast and effective separation of the source signals from stereo convolutive mixtures. We evaluated and compared our system with other State-of-the-Art methods on simulated data. Results show that our system demonstrates comparable separation performance while having lower computational complexity and no system delay. These properties make **AIRES** well suited for real time applications on small devices. A test program of **AIRES** BSS and full evaluation results are available on our GitHub [15].

References

[1] A. Hyvärinen, J. Karhunen, and E. Oja, "Independent component analysis." John Wiley & Sons, 2001.

[2] J. Janský, Z. Koldovský, and N. Ono, "A computationally cheaper method for blind speech separation based on auxiva and incomplete demixing transform," in *IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, Xi'an, China, 2016.

[3] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization," in *IEEE/ACM Trans. ASLP*, vol. 24, no. 9, 2016, pp. 1626–1641.

[4] H.-C. Wu and J. C. Principe, "Simultaneous diagonalization in the frequency domain for source separation," in *Proc. ICA*, 1999, pp. 245–250.

[5] H. Buchner, R. Aichner, and W. Kellermann, "Trinicon: A versatile framework for multichannel blind signal processing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Que., Canada, 2004.

[6] J. Chua, G. Wang, and W. B. Kleijn, "Convolutive blind source separation with low latency," in *Acoustic Signal Enhancement(IWAENC), IEEE International Workshop*, 2016, pp. 1–5.

[7] I. Senesnick, "Low-pass filters realizable as all-pass sums: design via a new flat delay filter," in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, 1999.

[8] M. Brandstein and D. Ward, "Microphone arrays, signal processing techniques and applications," in *Springer*, 2001.

[9] J. Benesty, M. M. Sondhi, and Y. Huang, Eds., *Springer Handbook of Speech Processing*, ser. Springer Handbooks. Berlin: Springer, 2008.

[10] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," in *IEEE Trans. on Evolutionary Computation*, Feb. 2011, vol. 15, no. 1, pp. 4–31.

[11] J. Garofolo *et al.*, "Timit acoustic-phonetic continuous speech corpus," 1993.

[12] "Microphone array speech processing," https://github.com/ZitengWang/MASP, accessed: 2019-07-29.

[13] "Ilrma," https://github.com/d-kitamura/ILRMA, accessed: 2019-07-29.

[14] R. B. Stephens and A. E. Bate, *Acoustics and Vibrational Physics*, London, U.K., 1966.

[15] "Comparison of blind source separation techniques," https://github.com/TUIlmenauAMS/Comparison-of-Blind-Source-Separation-techniques, accessed: 2019-07-29.