THE METHOD OF RANDOM DIRECTIONS OPTIMIZATION FOR STEREO AUDIO SOURCE SEPARATION

Oleg Golokolenko, Gerald Schuller

Technische Universität Ilmenau
Department of Media Technology
98693 Ilmenau, Germany

ABSTRACT

In this paper, a novel fast time domain audio source separation technique based on fractional delay filters with low computational complexity and small algorithmic delay is being presented and evaluated in experiments. Our goal is a Blind Source Separation (BSS) technique which can be applicable for low cost and less powerful devises where processing is done in real-time, e.g. hearing aids or teleconferencing setups. The proposed approach optimizes fractional delays implemented as IIR filters and attenuation factors 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 it has a lower computational time and there is no system delay like in frequency domain BSS. The proposed algorithm is validated via numerical experiments. Its separation performance is competitive with State-of-the-Art approaches but has lower computational complexity and absence of system delay.

Index Terms— Blind source separation, time domain, binaural room impulse responses, optimization

1. INTRODUCTION

With a rapid deployment of more sophisticated Internet Of Things (IoT), personal and medical portable gadgets, such as teleconference systems and modern hearing aids, the need of novel fast and robust techniques for BSS algorithms is increasing. Thus, proposed approach aims to perform stereo sound source separation with low computational complexity, with no system delay [1, 2] and no musical noise [3].

Previous BSS approaches mostly apply the Short Time Fourier Transform (STFT) to the signals [4], e.g., AuxIVA [5] and ILRMA [6, 7]. This converts the signal delay into a complex valued factors in the STFT subbands.

Despite good separation of sound sources using frequency domain BSS approaches, they have several disadvantages. Namely, the permutation problem and the gains in the subbands might be different, leading to a modified spectral shape. Moreover, there is 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 [1, 2].

On the other hand, time domain approaches, like TRINICON [8, 9], or approaches that use the STFT with short blocks and more microphones [10, 11], 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.

Moreover, TRINICON and [12] are meant to do dereverberation. Which is based on the estimation of coefficients for multiple FIR filters, which causes an increase in computation time. Even though [12] is meant to be the time domain BSS algorithm, unmixing FIR filters here are estimated based on the AuxIVA [5] approach in the frequency domain. Hence, separation performance depends on the speed of the unmixing matrix update, which can lead to utilization of outdated information.

To avoid the problem of system delays associated with frequency domain approaches, a low-latency time domain stereo source separation scheme is proposed in this paper. Moreover, to speed up the crosstalk minimization process, we are not focused on the dereverberation.

2. PROPOSED APPROACH

2.1. Proposed Time Domain BSS

In the proposed approach, 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 [13, 14], with an attenuation factor, the principle of a fractional delay and sum or adaptive beamformer [15]. 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 [16].

We assume a mixture recording from 2 sound sources (S_0 and S_1) made with 2 microphones (M_0 and M_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 1).
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 (2.2). 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 Fig. 1.

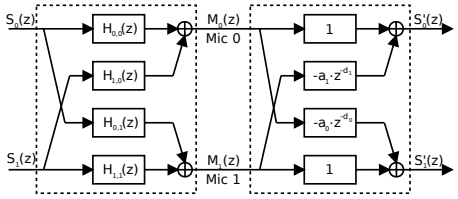

Fig. 1. Signal block diagram of convolutive mixing and demixing process.

2.2. 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.
In [13], 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 $0 \leq n \leq (L-1)$.

2.3. 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, i - the channel number 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 KLD 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

procedure OPTIMIZE FILTERS COEFFICIENTS
Input: **X** *# Signal to be separated*
Input: $alpha = 0.8$ *# Smoothness factor*
Input: $num\_iter = 30$ *# Number of optimization iterations*
Output: coeffs *# Filters coefficients*
    INITIALIZATION
    # Initial guess for separation coefficients
    coeffs = [1.0, 1.0, 1.0, 1.0]
    # Calculate objective value
    negabskl_0 = negabskl(coeffs, **X**)
    # Weights for random search
    coeffweights = [0.1, 0.1, 1.0, 1.0]*alpha
    OPTIMIZATION ROUTINE
    for i in range(num_iter):
        # Random variation of separation coefficients
        coeffvariation=(random_vector*coeffweights)
        tmp_coeffs = coeffs+coeffvariation
        # Calculate new objective value
        negabskl_1 = negabskl(tmp_coeffs, **X**)
        if negabskl_1 < negabskl_0 **then**
            negabskl_0 = negabskl_1
            coeffs = tmp_coeffs

2.4. 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" [17]. 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).

2.5. Real-Time Adaptation of AIRES

The most important change that has to be added to the offline version to turn it into real-time version is that it operates on a running window of past samples. We assume that the sound sources are moving continuously without significant jumps in space. This is done by saving of N past input signal blocks and corresponding unmixing filter states (since we have IIR filters). The current signal block is concatenated to the N stored past input signal blocks and the unmixing coefficients for the current block are calculated based on this set of blocks. Here we assume that the sound sources do not change their positions significantly, hence the unmixing coefficients should have only small changes. Moreover, the use of the memory helps to overcome the permutation problem and works as interpolation of unmixing coefficients Fig. 2.

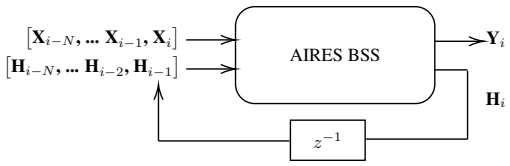

Fig. 2. Memory for online AIRES.
($\boldsymbol{X}_i$ - block of current input signal, $\boldsymbol{H}_i$ - block of current unmixing filter states and $\boldsymbol{Y}_i$ - block of current unmixed signal)

3. NUMERICAL EXPERIMENTS

In this section, we evaluate and compare the performance of the proposed **AIRES** (time domAIn fRactional dElay Separation) to that of TRINICON [8, 9] via numerical experiments.

3.1. Setup

For the simulations, the room impulse response simulator based on the image model technique [18] was used to generate room impulse responses. The room size have been chosen to be $7m \times 5m \times 3m$. The microphones were positioned in the

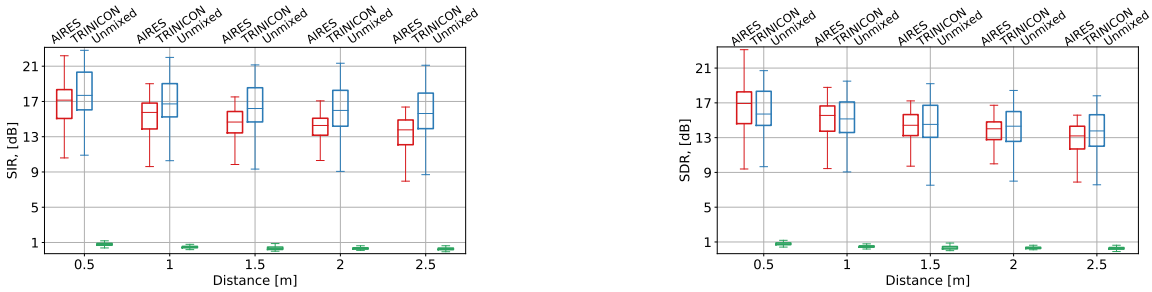

Fig. 3. Box-plots for the Signal-to-Interference Ratio (SIR, left) and Signal-to-Distortion Ratio (SDR, right) of the **offline** AIRES and TRINICON BSS approaches applied to simulated data ($RT_{60} = 0.1$).

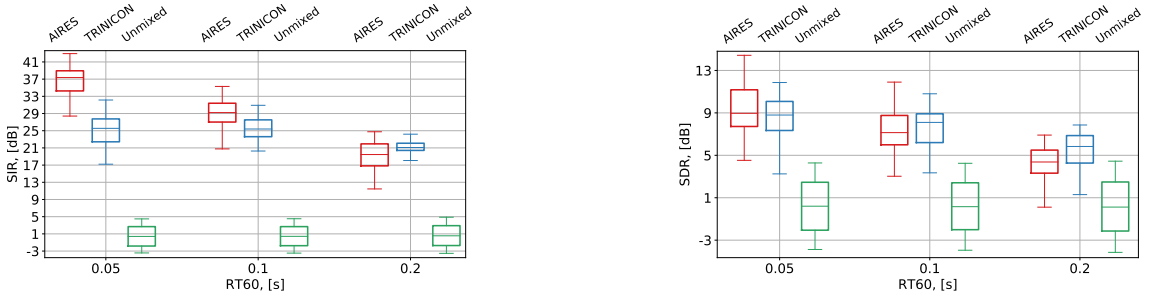

Fig. 4. Box-plots for the Signal-to-Interference Ratio (SIR, left) and Signal-to-Distortion Ratio (SDR, right) of the **online** AIRES and TRINICON BSS approaches applied to simulated data ($RT_{60} = 0.1$).

middle of the room with displacement of $0.05m$. In the simulations we used speech signals from the TIMIT data-set [19] with sampling frequency of 16kHz.

1. Offline setup: 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 and 5 different distances of the sound sources relatively to microphones, and for 3 reverberation times (RT_{60}).

2. Online setup: Here, the sound sources were moving randomly with movement each 512 samples and the mean moving speed was $0.2m/s$. Moreover, for relative comparison, in our experiments we used the same number of iterations per data block (2 iterations) for each BSS algorithm.

3.2. Results

We evaluate the separated signals in terms of Signal-to-Distortion ratio (SDR) and Signal-to-Interference ratio (SIR) as defined in [20]. These metrics are computed using the mir_eval toolbox [21].

The box-plots of SDR and SIR of the separated signals for offline mode is illustrated in Fig. 3. The obtained results show that **AIRES** demonstrates comparable separation performance while having lower computational complexity (Table 1) and no system delay.

As can be observed in Fig. 4, online **AIRES** outperforms TRINICON in separation performance in terms of SIR, but at RT60 = 0.2s **AIRES** works slightly worse. Moreover, one can see that TRINICON suppresses better artificial noises in the separated channels, which can be due to the fact that **AIRES** does not perform dereverberation.

	AIRES	TRINICON	Signal length
Offline	0.07s	14.55s	120s
Online	0.001s	0.002s	512 samples

Table 1. Comparison of average computation time.

4. CONCLUSION

We presented a novel approach for stereo audio source separation in time domain. Our method of random directions is successfully separating sources, in an offline and online ways, with low complexity (for embedded devices), and with fast convergence which is important for moving sources.

A test program of **AIRES** BSS and full evaluation results are available on our GitHub [22].

5. REFERENCES

[1] H. Sawada, N. Ono, H. Kameoka, and D. Kitamura, "Blind audio source separation on tensor representation," in *ICASSP*, Apr. 2018.

[2] J. Harris, Syed Mohsen Naqvi, J. A. Chambers, and Ch.

Jutten, "Real-time independent vector analysis with student's t source prior for convolutive speech mixtures," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, Apr. 2015.

[3] T. Esch and P. Vary, "Efficient musical noise suppression for speech enhancement system," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 4409–4412.

[4] J. Benesty, J. Chen, and Emanuel A.P. Habets, "Speech enhancement in the stft domain," in *Springer*, 2012.

[5] 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.

[6] 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*, 2016, vol. 24, pp. 1626–1641.

[7] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation with independent low-rank matrix analysis," in *Springer*, 2018, p. 31.

[8] 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.

[9] Robert Aichner, Herbert Buchner, Fei Yan, and Walter Kellermann, "A real-time blind source separation scheme and its application to reverberant and noisy acoustic environments," *Signal Processing*, vol. 86, no. 6, pp. 1260 – 1277, 2006, Applied Speech and Audio Processing.

[10] 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.

[11] W.B. Kleijn and K. Chua, "Non-iterative impulse response shortening method for system latency reduction," in *Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 581–585.

[12] M. Sunohara, C. Haruta, and N. Ono, "Low-latency real-time blind source separation for hearing aids based on time-domain implementation of online independent vector analysis with truncation of non-causal components," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 216–220.

[13] 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*, 1999, vol. 46.

[14] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay," in *IEEE Signal Processing Magazine*, Jan. 1996.

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

[16] Jacob Benesty, M. Mohan Sondhi, and Yiteng Huang, Eds., *Springer Handbook of Speech Processing*, Springer Handbooks. Springer, Berlin, 2008.

[17] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," in *IEEE Trans. on Evolutionary Computation*, vol. 15, pp. 4–31. Feb. 2011.

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

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

[20] R. Gribonval E. Vincent and C. Fevotte, "Performance measurement in blind audio source separation," in *IEEE Trans. Audio, Speech, Lang. Process.*, Jun. 2006, vol. 41, pp. 1–24.

[21] C. Fevotte, R. Gribonval, and E. Vincent, "Bss eval toolbox user guide," in *Tech. Rep. 1706, IRISA Technical Report 1706*, Rennes, France, 2005.

[22] "Comparison of blind source separation techniques," `https://github.com/TUIlmenauAMS/`, Accessed: 2019-10-21.