INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

The Method of Random Directions Optimization for Stereo Audio Source Separation

Oleg Golokolenko, Gerald Schuller

Ilmenau University of Technology, Ilmenau, Germany
`oleg.golokolenko@tu-ilmenau.de, gerald.schuller@tu-ilmenau.de`

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 presented and evaluated in experiments. Our goal is a Blind Source Separation (BSS) technique, which can be applicable for the low cost and low power devices 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 crosstalk, the principle of a fractional delay and sum beamformer. The experiments have been carried out for offline separation with stationary sound sources and for real-time with randomly moving sound sources. Experimental results show that separation performance of the proposed time domain BSS technique is competitive with State-of-the-Art (SoA) approaches but has lower computational complexity and no system delay like in frequency domain BSS.

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 devices, such as teleconference systems and modern hearing aids, the need of novel fast and robust techniques for BSS algorithms is increasing. Thus, the proposed approach aims to perform stereo sound source separation with low computational complexity, with neither system delay [1, 2] and nor 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 (FD) BSS approaches, there are several disadvantages. Namely, the permutation problem and the gains in the subbands might be different, leading to a modified spectral shape - musical noise. Moreover, there is a signal delay resulting from applying an STFT. It needs the assembly of the signal into blocks, which requires a system delay corresponding to the block size [1, 2].

On the other hand, time domain (TD) 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. However, they usually have a higher computational complexity, which makes them hard to use on small devices with less powerful processors in real-time.

Moreover, TRINICON and [12] are meant to do dereverberation. This 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.

Conventional BSS algorithms are based on estimation of the FIR filter coefficients for sound source separation using only integer signal delay. As a result, without data pre-processing, commonly used Gradient Descend optimization might have gradient equals to zero or infinity resulting in degradation of separation performance. Hence, to speed up our separation algorithm as much as possible to be able to implement it on a low power hardware, we are not focused on dereverberation and data pre-processing. Moreover, to avoid the problems associated with frequency domain approaches, such as system delays and musical noise, we use a time domain stereo source separation scheme. Thus, assigned constraints require exploration of a new separation algorithm together with cost function and optimization method. The proposed low-latency time domain BSS method formulation together with evaluation are presented in the following sections.

2. Proposed approach

2.1. Formulation of the proposed Time Domain BSS

In the proposed approach, instead of using FIR filters, we employ IIR filters, which are implemented as fractional delay allpass filters [13, 14], with attenuation factors. This can be seen as a sum or adaptive beamformer [15, 16]. The IIR delay filter implementation has the advantage that each such filter has only two coefficients to be optimized, the fractional delay and the attenuation. As written above, 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 microphones by an attenuation and a pure fractional delay [17].

In this paper, we assume a mixture recording from two sound sources (S_0 and S_1) made with two 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.

Instead of the commonly used STFT, we use the z-transform for the mathematical derivation, because it does not need a decomposition of the signal into blocks, with its associated delay. This makes the mathematical derivation applicable for a time domain implementation with no signal delay. Thus, 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

Copyright © 2020 ISCA

http://dx.doi.org/10.21437/Interspeech.2020-1409

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 IIR filters using fractional delays d_i 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 allpass filter described in the next section (2.2).

The signal flowchart of convolutive mixing and demixing process can be seen in Fig. 1.

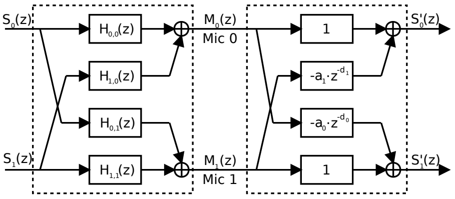

Figure 1: *Signal block diagram of convolutive mixing and demixing process.*

Please note, since we are not focused on the dereverberation, we keep the linear filter resulting from the determinant and from the matrix diagonal $H_{i,i}(z)$ on the left hand side of eq. (4).

2.2. The fractional delay allpass filter

In order to implement Relative Transfer Functions (eq. 5), we use the IIR fractional delay allpass filter [13] with a maximally flat group delay response. As a result we obtain the filter out of a single fractional delay coefficient, needed for sufficient cross-talk cancellation.

We use 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 $0 \leq n \leq (L-1)$.

As a next step, we propose a fast analogue of mutual information calculation as an objective function.

2.3. Objective function

The most used and conventional objective functions in BSS theory are Mutual Information and the Kullback-Leibler Divergence (KLD). The biggest drawback of these approaches is the computational complexity. Thereby, there is a need to calculate signals probability distributions and joint entropy, which are computationally complex and time costly. Thus, in this research, we propose a new fast objective function, which is derived from the Kullback-Leibler Divergence.

The conventional KLD is defined as follows,

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

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.

A comparative study of the proposed cost function together with conventional ones has shown the similarities of the separation performance except the computational time. Where our proposed cost function works remarkably faster. However, this comparison is out of the scope of this paper.

Algorithm 1 Optimization algorithm

```
 1: 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
 2:     INITIALIZATION
 3:     # Weights for random search
 4:     coeffweights = [0.1, 0.1, 1.0, 1.0]*alpha
 5:     # Initial guess for separation coefficients
 6:     coeffs = [1.0, 1.0, 1.0, 1.0]
 7:     # Calculate objective value
 8:     negabskl_0 = negabskl(coeffs, X)
 9:     OPTIMIZATION ROUTINE
10:     for i in range(num_iter):
11:         # Random variation of separation coefficients
12:         coeffvariation=(random_vector*coeffweights)
13:         tmp_coeffs = coeffs+coeffvariation
14:         # Calculate new objective value
15:         negabskl_1 = negabskl(tmp_coeffs, X)
16:         if negabskl_1 $<$ negabskl_0 then
17:             negabskl_0 = negabskl_1
18:             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. Especially this is the case for non-convex functions as in our scenario with not pre-processed signals. Better results can be achieved using Differential Evolution (DE) [18, 19] or Genetic Algorithm Optimization (GAO) [20, 21]. Unfortunately, DE optimization has huge processing time, while GAO does not give good sound source separation results without an extensive enhancement.

Hence, we came up with a modified version and combination of DE and GAO. In order to make it even more faster we use only one solution per population and initialized our population with reasonable values instead of random values.

Unlike pure GAO, for the population update, we use a weight vector to model the expected variance distribution of our coefficients. Since attenuation factors change slower than delays, they get smaller variances (Algorithm 1, line 4). 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 communication applications.

The algorithm starts with a fixed starting point (population) [1.0, 1.0, 1.0, 1.0], which we found to lead to a robust convergence behaviour (Algorithm 1, line 6). Then, it perturbs the current point with a vector of uniformly distributed random numbers between -0.5 and +0.5 (the random direction), elementwise 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. The simplified pseudocode of the optimizer is shown in 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 is operation on a running window of past samples. Here, we assume that the sound sources are moving continuously without significant jumps in space (with maximum speed of $0.3m/s$). This is done by saving of N past input signal blocks as overlapping windows. The current signal block is concatenated to the stored past input signal blocks, and the oldest is dropped. Since we assume that the sound sources do not change their positions significantly, the unmixing coefficients should have only small changes. Moreover, the use of overlapping windows helps to overcome the permutation problem and works as interpolation of unmixing coefficients.

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 for **offline** and **real-time** scenarios. We omit a comparison with BSS algorithms based on deep learning, since such systems are trained for specific cases. Whereas, **AIRES** is adaptive BSS algorithm. Moreover, in order to be consistent, TD and FD BSS algorithms are not compared, because of the unfavorable properties of the FD algorithms (see Sec. 1).

3.1. Setup

For the simulations, the room impulse response simulator based on the image model technique [22] was used to generate room impulse responses. The room size has been chosen to be $[7 \times 5 \times 3]m$. The microphones were positioned in the middle of the room with displacement of $0.05m$. In the simulations, speech signals from the TIMIT data-set [23] (male and female) with sampling frequency of $16kHz$ have been used. In each single simulation, one pair of speech signals was randomly chosen from the whole TIMIT data-set and convolved with the simulated RIRs. In both testing scenarios, the positions of the sound sources and the RT_{60} are variable and simulations are preformed for ten pairs of speech signals. In the real-time scenario block processing has been performed to blocks of size 512 samples and a window length of 3 blocks.

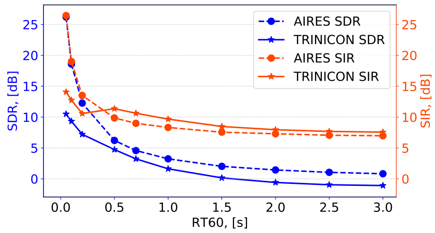

Figure 2: *Signal-to-Interference Ratio (SIR) and Signal-to-Distortion Ratio (SDR) of the* ***offline*** *AIRES and TRINICON BSS approaches applied to simulated data (Distance $= 1.5[m]$).*

1. Offline setup: For each pair of signals, the simulation has been performed at 50 random angle positions and 8 different distances of the sound sources relatively to microphones (polar coordinate system), and for 9 reverberation times (RT_{60}). Thus, in total - 4000 simulations per RT_{60}.

2. Online setup: Here, the sound sources movement is implemented randomly with a delay of 512 samples, and the mean moving speed - $0.2m/s$. The simulations have been repeated 50 times per speech pair. Thus, in total - 500 simulations per

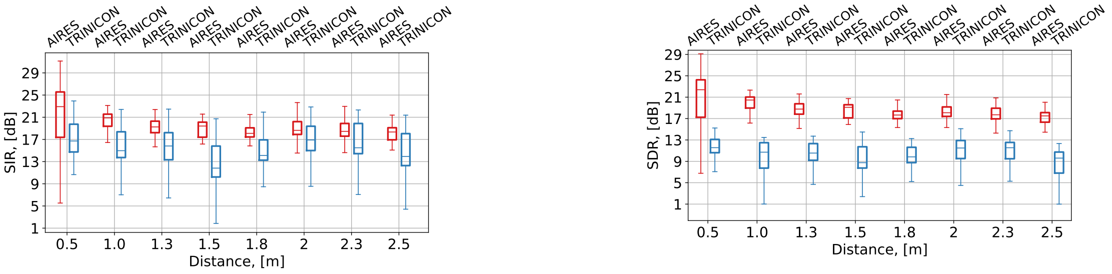

Figure 3: *Box-plots for the Signal-to-Interference Ratio (SIR, left) and Signal-to-Distortion Ratio (SDR, right) of the* ***offline*** *AIRES and TRINICON BSS approaches applied to simulated data ($RT_{60} = 0.1[s]$).*

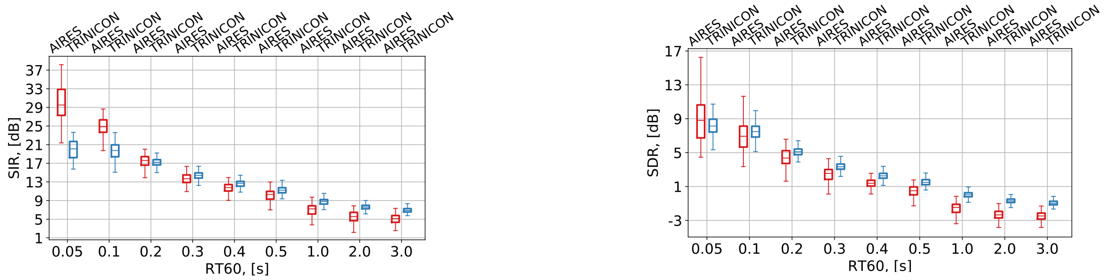

Figure 4: *Box-plots for the Signal-to-Interference Ratio (SIR, left) and Signal-to-Distortion Ratio (SDR, right) of the* ***online*** *AIRES and TRINICON BSS approaches applied to simulated data.*

Table 1: *Comparison of average computation time.*

	AIRES	TRINICON	Signal length
Offline	0.07s	14.55s	120s
Online	0.00097s	0.00249s	512 samples

$RT_{60} \in [0.05, ..., 3]s$. Moreover, for a relative comparison, in our experiments we used the same number of separation 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 [24]. These metrics are computed using the mir_eval toolbox [25].

1. Offline setup: Here, one may assume the distance to sound sources does not exceed $\approx 1.5m$. This is a common scenario for the proposed applications. Thus, the results over RT_{60} for the assumed maximum distance are shown in Fig. 2. The results for $RT_{60} = 0.1s$ (as was presented for TRINICON in [8]) for a cross comparison are shown in Fig. 3. The obtained results show that **AIRES** outperforms TRINICON at $RT_{60} = 0.1s$ (Fig. 3), while it is slightly behind at $RT_{60} > 0.6s$ (Fig. 2).

2. Online setup: As can be observed in Fig. 4, online **AIRES** outperforms online TRINICON in separation performance at $RT60 < 0.2s$. Moreover, one can see that TRINICON has a better SDR measure at $RT60 > 0.2s$, which can be due to the fact that **AIRES** does not perform dereverberation.

Examination of computational complexity (Table 1) shows superiority of **AIRES**. Thus, in the offline scenario **AIRES** works 207 times faster then TRINICON, while in the online scenario - 2.5 times faster. This is crucial, since we are focused on low cost and low power devices where computational complexity is constrained.

Besides this, hearing tests of separated sound sources have shown that an SIR of about 8dB results in a good speech intelligibility. Which means that for the offline scenario both BSS algorithms fail in separation at $RT60 > 1.5s$, and for online scenario at $RT60 > 1s$.

4. Conclusions and future work

We presented a novel approach for stereo audio source separation in the time domain. The proposed **AIRES** BSS technique successfully separates reverberated sources, in offline and online scenarios, with low complexity, and with fast convergence, which is important for moving sources. A small separation difference of about $1.5dB$ in SIR measure at high $RT60$ is the price which we have to pay in order to have faster processing, which is crucial for embedded devices with low processing power.

As the future work we are going to implement and evaluate proposed **AIRES** BSS technique on smartphone devices and FPGA board. Besides this, the computational complexity in terms of number of operations will be investigated.

A test program of **AIRES** BSS and evaluation results are available at [26].

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, S. M. Naqvi, J. A. Chambers, and C. 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 E. A. 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*, vol. 24, no. 9, 2016, pp. 1626–1641.

[7] ——, "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] R. Aichner, H. Buchner, F. Yan, and W. 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168405003270

[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. 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. Selesnick, "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.

[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] "Beamforming," http://www.labbookpages.co.uk/audio/beamforming/delaySum.html, accessed: 2019-04-21.

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

[18] 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.

[19] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," in *Journal of Global Optimization. 11 (4)*, 1997, pp. 341–359.

[20] A. P. D. Charles C. Peck, "Genetic algorithms as global random search methods: An alternative perspective," in *Evolutionary Computation, Volume 3 Issue 1, MIT Press.*, March 1995.

[21] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," in *Addison Wesley Publishing Company, Ind. USA*, 1989.

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

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

[24] R. G. E. Vincent and C. Fevotte, "Performance measurement in blind audio source separation," in *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 41, no. 1-4, Jun. 2006, pp. 1–24.

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

[26] "Comparison of blind source separation techniques," https://github.com/TUIlmenauAMS/, accessed: 2020-05-02.