# An Approach for Solving the Permutation Problem of Convolutive Blind Source Separation Based on Statistical Signal Models

Radoslaw Mazur and Alfred Mertins, *Senior Member, IEEE*

***Abstract*—In this paper, we present a new algorithm for solving the permutation ambiguity in convolutive blind source separation. Transformed to the frequency domain, existing algorithms can efficiently solve the reduction of the source separation problem into independent instantaneous separation in each frequency bin. However, this independency leads to the problem of correctly aligning these single bins. The new algorithm models the frequency-domain separated signals by means of the generalized Gaussian distribution and employs the small deviation of the parameters between neighboring bins for the detection of correct permutations. The performance of the algorithm will be demonstrated on synthetic and real-world data.**

***Index Terms*—Blind source separation (BSS), convolutive mixture, frequency-domain ICA, permutation problem.**

## I. Introduction

BLIND source separation (BSS) is a method to recover signals from observed mixtures of different sources without knowledge of the sources or the mixing system. A number of efficient approaches have been proposed for the case of linear instantaneous mixtures [1]–[3].

The situation becomes more difficult when we apply these approaches to real-world mixtures of human speech. In a reverberant environment, the signals arrive multiple times with different time lags. Thus, the mixing process is convolutive and must be modeled by applying room transfer functions. Modeling the room transfer functions, however, requires filters with thousands of taps in realistic scenarios. BSS includes estimation of an inverse system of filters which usually have similar or even greater lengths.

One possibility to solve the convolutive blind separation problem is to calculate the unmixing filters directly in the time domain [4], [5]. However, this approach results in high computational cost and often shows difficulties with convergence, because the algorithm can get trapped in one of the many local minima of the objective function. Another method is to transform the signals to the frequency domain, so that convolution becomes a multiplication [6]. However, if all frequency bins are separated independently, the discrete bins usually have different scalings, and they can be arbitrarily permuted. One method to avoid such problems is to use a frequency-domain separation criterion, but to restrict the time-domain impulse responses of the unmixing filters to a certain maximum length [7], which means that the coefficients for all frequency bins have to be modified jointly. Similar to the direct time-domain approaches, the objective function shows many local minima in which the algorithm can get trapped [8], and a good initialization is often essential to achieve good performance.

Another class of frequency-domain methods solves the blind separation problems independently in the various frequency bins and then deals explicitly with the scaling and permutation ambiguities in a subsequent processing step. In these algorithms, the separation step in the individual frequency bins often converges very fast with good bin-wise separation performance, and the main task is to scale and group the components that stem from the same source. The scaling problem can be satisfactorily addressed using the postfilter method proposed in [9]. Using inverse postfilters allows us to recover the signals as they have been recorded at the microphones. This method accepts the filtering done by the mixing system while ensuring that the unmixing system will not add any further distortions. There are different methods to deal with the permutation problem. One possibility is based on the assumption that neighboring bins have a similar time structure [10]. Correlation coefficients for signals in neighboring bins thus yield a criterion for correct permutation. In [11], the authors used the amplitude modulation correlation for getting a separation criterion which avoids the permutation problem. Other approaches employ the unmixing matrices as beamformers [12] or look at the general directivity patterns [13], [14]. For example, it has been shown in [12] that most of the bins can be aligned properly after the directions of arrival have been computed. However, computation becomes a difficult matter with more than two sensors in a nonuniform array. In [15], the authors of [12] extended their approach to three dimensions. By using near-field and far-field models, a separation of multiple sources was possible [16].

The method proposed in this paper belongs to the class of algorithms in which bin-wise separation is followed by explicit depermutation. In particular, we propose a new way to explicitly solve the permutation problem based on the statistics of the signals. Our algorithm models the discrete frequency bins using the generalized Gaussian distribution (GGD) and employs the tiny differences of the parameters of the GGD between neighboring bins for aligning permutations. We draw on our previous work in

Manuscript received December 12, 2007; revised July 17, 2008. Current version published December 11, 2008. This work was supported by the German Research Foundation under Grant ME 1170/1. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Gael Richard.

The authors are with the Institute for Signal Processing, University of Lübeck, D-23538 Lübeck, Germany (e-mail: mazur@isip.uni-luebeck.de; mertins@isip.uni-luebeck.de).

Digital Object Identifier 10.1109/TASL.2008.2005349

[17], where we proposed to use the $\beta$-parameter of the GGD and extend this to the use of more than one parameter. While the algorithm of [17] showed very promising results on synthetic data, it often failed on the lower frequencies of real-world recorded speech examples. The 2-D extension proposed in this paper performs much better and is able to tackle most drawbacks of the previous method. The new clustering method is able to cluster more bins and double the average cluster size. The overall performance is much improved and especially the problematic low frequencies are handled much better.

The information exploited in our approach is entirely different from the directivity information used in [12]–[15]. Thus, in future works it may be possible to combine the proposed signal-statistics-based method with the beamforming paradigm and further improve the performance of depermutation algorithms.

## II. Model and Methods

### A. BSS for Instantaneous Mixtures

In the instantaneous case, the mixing process of $N$ sources into $N$ observations can be modeled by an $N \times N$ matrix $\mathbf{A}$. Given the source vector $\mathbf{s}(n) = [s_1(n), \ldots, s_N(n)]^T$ and assuming negligible measurement noise, the vector of observation signals can be described as $\mathbf{x}(n) = \mathbf{A} \cdot \mathbf{s}(n)$. The separation can be written as a multiplication with an $N \times N$ matrix $\mathbf{B}$, resulting in a vector $\mathbf{y}(n) = \mathbf{B} \cdot \mathbf{x}(n)$ of unmixed signals. The aim of BSS is to find $\mathbf{B}$ from the observed process $\mathbf{x}(n)$ so that $\mathbf{BA} = \mathbf{D\Pi}$, where $\mathbf{\Pi}$ is a permutation matrix and $\mathbf{D}$ an arbitrary diagonal matrix. These matrices represent the two ambiguities of BSS: 1) the separated signals appear in arbitrary order and 2) they are scaled versions of the sources.

We here consider the well known gradient-based update rule [1] $\mathbf{B}_{k+1} = \mathbf{B}_k + \Delta\mathbf{B}_k$ with

$$\Delta\mathbf{B}_k = \mu_k(\boldsymbol{I} - E\{\boldsymbol{g}(\mathbf{y})\mathbf{y}^T\})\mathbf{B}_k \tag{1}$$

and $\boldsymbol{g}(\mathbf{y}) = (g_i(y_i), \ldots, g_n(y_n))$ being a component-wise vector function of nonlinear score functions $g_i(s_i) = -p_i'(s_i)/p_i(s_i)$, where $p_i(s_i)$ are the assumed source probability densities. These should be known or at least well approximated in order to achieve good separation performance [18].

### B. Statistical Source Models and Estimators

Speech signals usually contain many small and few large values, and their amplitude statistics in voiced activity intervals can be reasonably well approximated by a Laplacian probability density functions (pdf) [19]. For a Laplacian pdf, the nonlinear function $g(\cdot)$ reduces to

$$g(y) = \frac{\operatorname{sgn}(y)}{\sigma}. \tag{2}$$

However, when speech pauses are also considered, the observation of Laplacian pdf no longer holds, and a sufficient approximation can, for example, be achieved by the GGD [19]

$$p_y(y) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|y|/\alpha)^\beta} \tag{3}$$

with $\alpha, \beta > 0$. In (3), $\Gamma(\cdot)$ is the Gamma function given by

$$\Gamma(y) = \int_0^\infty x^{y-1} e^{-x} dx. \tag{4}$$

The $\beta$-parameter of the GGD describes the sparsity of the distribution. With $\beta = 2$ the GGD reduces to a standard Gaussian distribution and with $\beta = 1$ to a Laplacian distribution. The parameter $\alpha$ is the generalized measure of the variance and is also called dispersion or scale parameter. For the Gaussian distribution, $\alpha$ reduces to the known standard deviation $\sigma$.

Using the GGD, the nonlinear function $g_i(\cdot)$ becomes

$$g_i(x_i) = |x_i|^{\beta-1} \operatorname{sgn}(x_i) \tag{5}$$

and with $\operatorname{sgn}(x) = (x/|x|)$, the expression reduces to

$$g_i(x_i) = \frac{x_i}{|x_i|^{2-\beta}}. \tag{6}$$

By applying this nonlinear function, even mixtures of sub- and super-Gaussian signals can be separated, as [18] shows. The approach of [18] has been further extended in [20], where an adaptive algorithm for determining $\beta$ from the statistics of the separated signals was proposed. The authors used the method of moments [21] to estimate $\beta$ after each iteration of (1) and used this new value for the next step. This approach leads to improved performance in terms of both separation and convergence.

Besides the method of moments, further estimators for the parameters of the GGD have been studied in [22]. These include a generalized entropy matching estimator and a maximum likelihood estimator. It was found that the maximum likelihood estimator performs best for small sample size and small $\beta$, whereas the generalized entropy matching estimator is best for large $\beta$. For large sample size, all estimators performed equally well.

### C. Convolutive Mixtures

The mixing channels in acoustic real-world scenarios can be modeled using finite-impulse response (FIR) filters of length $L$, where $L$ can exceed 2000, depending on the reverberation time and sampling rate. The convolutive mixing model reads $\mathbf{x}(n) = \sum_{l=0}^{L-1} \mathbf{H}(l)\mathbf{s}(n-l)$, where $\mathbf{H}(n)$ is a sequence of $N \times N$ matrices containing the impulse responses of the mixing channels. For the separation, one can use FIR filters of length $M \geq (N-1)(L-1)+1$ [23] and obtains $\mathbf{y}(n) = \sum_{l=0}^{M-1} \mathbf{W}(l)\mathbf{x}(n-l)$ with $\mathbf{W}(n)$ containing the unmixing coefficients. There exist approaches to estimate $\mathbf{W}(n)$ in the time domain [4], [5], but the results are often unsatisfactory due to distortions that result from the unmixing system.

In frequency domain approaches, the convolution is turned into multiplication by means of the blockwise short-time Fourier transform (STFT) [6]. For frequency $\omega_k$, one can write

$$\boldsymbol{Y}(\omega_k, \tau) = \boldsymbol{W}(\omega_k)\boldsymbol{X}(\omega_k, \tau) \tag{7}$$

where $\boldsymbol{X}(\omega_k, \tau)$ and $\boldsymbol{Y}(\omega_k, \tau)$ are the STFTs of $\mathbf{x}(n)$ and $\mathbf{y}(n)$ in frame $\tau$, respectively, and $\boldsymbol{W}(\omega)$ is the discrete-time Fourier transform of the sequence $\mathbf{W}(n)$. Rather than estimating all coefficients at once, it is now possible to separate the sources in each frequency bin independently. However, a major drawback

of this method is that in each frequency bin $\omega_k$ only scaled and permuted versions of the signals can be estimated

$$\boldsymbol{Y}(\omega_k,\tau) = \boldsymbol{W}(\omega_k)\boldsymbol{X}(\omega_k,\tau) = \boldsymbol{D}(\omega_k)\boldsymbol{\Pi}(\omega_k)\boldsymbol{S}(\omega_k,\tau) \tag{8}$$

where $\boldsymbol{\Pi}(\omega)$ is a frequency-dependent permutation matrix and $\boldsymbol{D}(\omega)$ an arbitrary diagonal scaling matrix. Therefore, it is necessary to correct the amplitudes and solve the permutation before transforming the signals back to the time domain.

If the scaling in the different bins is not corrected then the restored signals are only a filtered version of the sources. As these filters are quite arbitrary, usually this means added reverberation and thus reduced intelligibility of the speech [24]. In [9], Ikeda and Murata proposed a method for minimizing the scaling ambiguity. The main notion is to retrieve the signals the way they have been recorded by the sensors. This has been accomplished by applying postfilters on the single separated signals which were the inverses of the unmixing filters. In [25], the same technique was used, and it was demonstrated that this approach minimizes $E\{|y(t) - x(t)|^2\}$. Their so-called Minimal Distortion Principle uses the following unmixing matrix:

$$\boldsymbol{W}'(\omega) = \operatorname{diag}(\boldsymbol{W}^{-1}(\omega)) \cdot \boldsymbol{W}(\omega) \tag{9}$$

with $\operatorname{diag}(\cdot)$ returning the argument with all off-diagonal elements set to zero.

The correction of the permutation ambiguity is even more important. If every bin is perfectly separated, but different permutations occur at different frequencies, the sources will still not be separated by the unmixing system. Some approaches to overcome this obstacle have been proposed, but their performance is often unsatisfactory, and it remains a challenging task to solve the permutation ambiguity. The methods can be divided into two large groups. On one side there are algorithms exploiting the statistics of the signals and on the other side algorithms using the properties of the unmixing system. In the following, we give a brief overview of existing techniques.

### D. Methods for Resolving the Permutation Ambiguity

One method to resolve the permutation ambiguity is to demand continuity of the frequency responses of the unmixing filters [26]. Unfortunately, in real-world scenarios, the approach often fails when the signals are not completely separable in a certain bin and the corresponding unmixing matrix thus differs significantly from its neighbors.

The algorithm presented in [27] transforms the smoothness of the unmixing filters into the smoothness of the time-frequency representations, which the authors call profiles. Utilizing the smoothness of the unmixing filters, the permutation can be resolved up to some frequency jumps, which means that there are blocks of correctly sorted bins. However, if there are several jumps in close proximity, it is not possible to uncover them, and especially single or small blocks of permuted bins cannot be detected.

Another successful idea of utilizing the information of the unmixing system is to interpret the unmixing system as a beamformer that forms spatial zeros to the different sources [12]. By estimating the directions of arrival, the frequency components that stem from the same spatial direction are aligned to form an output signal. This algorithm has some drawbacks because not all frequencies can be sorted in general. At high frequencies, spatial aliasing occurs and not all frequencies can be aligned correctly. A solution to this problem was proposed in [28], but it is still limited to a maximum of three sources and it works only in low-reverberant rooms. At low frequencies, where the phase differences at the microphones are small, the direction of arrival cannot always be estimated exactly. Moreover, if there are more than two sources and two sensors the estimation of direction can be very difficult. Solutions involve, for example, pairwise analyses [15].

The methods in [9], [10] show attempts to solve the permutation problem by aligning the time structure of the separated signals. The central notion is that the envelopes of all bins belonging to the same source are highly correlated. With $\boldsymbol{V}(\omega,\tau) = |\boldsymbol{Y}(\omega,\tau)|$ the correlation between two bins $k,l$ is defined as

$$\rho_{qp}(\omega_k,\omega_l) = \frac{\sum_{\tau=0}^{\mathcal{T}-1} V_q(\omega_k,\tau)V_p(\omega_l,\tau)}{\sqrt{\sum_{\tau=0}^{\mathcal{T}-1} {V_q}^2(\omega_k,\tau)}\sqrt{\sum_{\tau=0}^{\mathcal{T}-1} {V_p}^2(\omega_l,\tau)}} \tag{10}$$

where $p, q$ are the indices of the separated signals, $V_q(\omega_k,\tau)$ is the $q$th element of $\boldsymbol{V}_q(\omega_k,\tau)$, and $\mathcal{T}$ is the number of frames. To decide whether two bins are permuted or not the value of

$$r = \frac{\rho_{pp}(\omega_k,\omega_l) + \rho_{qq}(\omega_k,\omega_l)}{\rho_{pq}(\omega_k,\omega_l) + \rho_{qp}(\omega_k,\omega_l)} \tag{11}$$

can be used. If $r > 1$, the bins are sorted correctly. Otherwise, a permutation has occurred. With more than two sources, the value of $r$ has to be estimated for all pairs, which means that $(N(N-1)/2)$ calculations have to be performed, resulting in a significant computational cost. For speech signals, it is commonly not possible to sort all bins with respect to $r$ for all $p$ and $q$, because the key assumptions of this method are usually not satisfied for all frequencies. This is demonstrated in Fig. 1. The two neighboring bins 100 and 101 have peaks at almost the same position while bin 300 significantly differs. In such cases the value of $r$ is often smaller than one and thus leads to wrong permutation. Another problem is the not always perfect separation in the frequency domain. If the bins are not well separated then the envelopes of the signals do not differ enough. One idea to resolve this problem is the dyadic sorting algorithm [10], which starts with pairwise correlation of two neighboring bins and then takes pairs of grouped bins and groups them again, in the hope that a few falsely aligned bins within larger groups will not preponderate. However, if many false permutations occur in close proximity at an early stage, the entire separation process can fail.

## III. Proposed Method

In this paper, we propose to use the smoothness of the parameters $\alpha$ and $\beta$ of the GGD for solving the permutation problem. The statistics of the magnitude of every bin $|Y(\omega)|$ are approximated by the generalized Gaussian distribution as in [29]

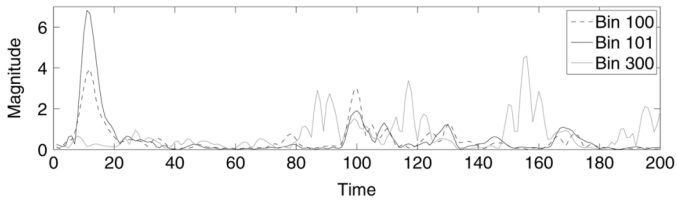


Fig. 1. Magnitudes of three frequencies of one signal.

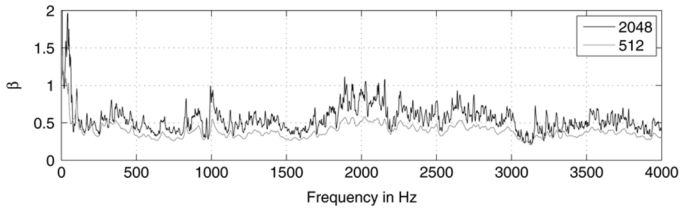


Fig. 2. Comparison of $\beta$ values over frequency for different window lengths.

$$p_{|Y(\omega)|}(Y(\omega)) = \frac{\beta(\omega)}{\alpha(\omega)\Gamma(1/\beta(\omega))} e^{-(|Y(\omega)|/\alpha(\omega))^{\beta(\omega)}}. \tag{12}$$

For obtaining the parameters $\alpha$ and $\beta$ from samples, we used the method of moments because of its low computational complexity. The maximum-likelihood estimator was studied as well, but not differences could be observed with regard to the final separation performance.

The author of [29] studied the properties of $\beta$ depending on the frequency and window length. He showed that, on average, $\beta$ stays constant over frequency and increases with the window length. He also showed that, although $\beta$ is constant on average, its deviation is quite high. This results from the fact that, for a given signal, $\beta$ usually varies significantly over the frequency. In Fig. 2, this situation is illustrated. One can also observe that, with higher window length, $\beta$ gets larger and varies more.

For a typical BSS scenario, in which the unmixing filters are several thousand taps long, the values of $\alpha$ and $\beta$ vary in a large range, but in neighboring bins the differences are usually small. Even more important is the fact that in most bins the values are distinct enough for creating a criterion for determining correct permutations. In Fig. 3, a typical situation for the $\beta$ parameter is shown. For the bins around 3920, no differentiation is possible as the values are almost identical. However, in the range 3800 to 3840 the values are clearly different enough for a correct identification of the permutations. For speech signals, this situation is quite common, so a large portion of bins can be assigned to clusters. This way the permutation problem can be reduced from several thousand single bins to about a hundred clusters. These clusters still have to be depermuted using another method, for example using the correlation approach.

The work in this paper is an extension of our previous conference contribution [17] from one to more dimensions. For the method in [17] it was found that many bins could be sorted correctly, but that there were still quite a few frequency ranges where the depermutation had failed. The reason for this is the fact that the correlation method failed for the lower frequencies where less bins could been correctly clustered. The larger gaps between the clusters hindered the effectiveness of the correlation method.

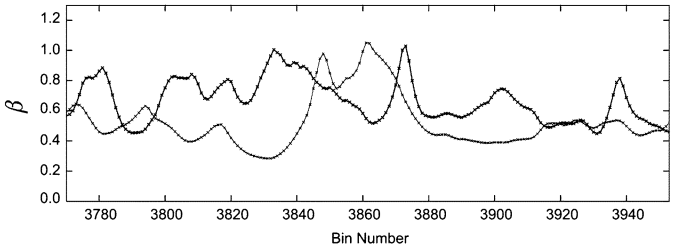


Fig. 3. Beta values of two signals.

In the present paper, we base the clustering on both parameters $\alpha$ and $\beta$ of the GGD. This enables us to find the correct permutations when at least one of the two parameters shows distinct values for the separated components. Thus, crossings like the ones of the $\beta$ value at several positions in Fig. 3 do not pose a problem if the $\alpha$ parameters do not cross at the same bins. Because $\alpha$ utilizes other statistical properties of the signals than $\beta$, there is, in general, a good chance that at least one of the two parameters allows proper clustering. As a consequence, in comparison to the original method, more bins can be assigned to less clusters. This is a much better starting point for the correlation approach.

The proposed method consists of five parts: 1) estimation of the boundaries of the clusters using $\beta$, 2) estimation of the boundaries of the clusters using $\alpha$, 3) joining both cluster types, 4) calculation of the permutation between the clusters, and 5) aligning the remaining bins.

### A. Calculation of the Cluster Boundaries Using Parameter $\beta$

The starting point for the clustering procedure is the estimation of $\beta(\omega)$ for all bins. For the reason of simplicity, we first describe the algorithm for $N = 2$ sources. An extension to multiple signals will be given later. When using the algorithm from Section II, then the values are already given. It is also possible to use any other BSS algorithm, as the values for $\beta(\omega)$ can be estimated after separation.

The second step is to make a simple grouping. The bins are compared pairwise and the ones with higher value of $\beta(\omega)$ are assigned to one and the ones with lower values to the other source, as shown in Fig. 4(b). The corresponding high and low $\beta(\omega)$ values are given by

$$\beta_H(\omega) = \max[\beta(\omega,1), \beta(\omega,2)] \tag{13}$$
$$\beta_L(\omega) = \min[\beta(\omega,1), \beta(\omega,2)] \tag{14}$$

The third step is to determine the actual clusters. The idea for a simple and fast method is the following: Take an existing cluster and find out if the neighboring bin can be added to it. The decision is based on the assumption of the values of $\beta$ being distinct and smooth.

The actual implementation is as follows.

1) Start at bin $l = 1$.

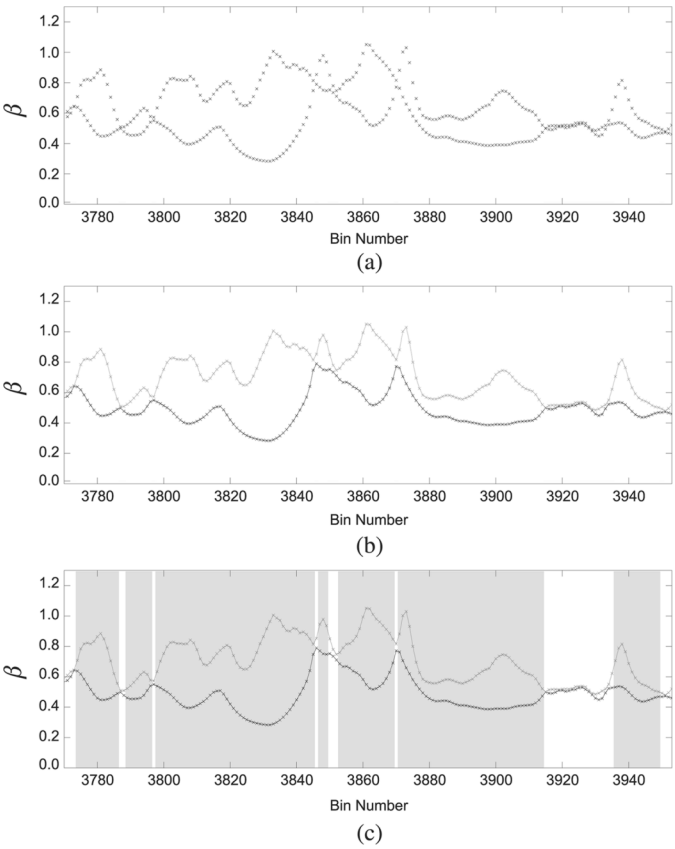


Fig. 4. Clustering procedure using $\beta$. (a) The estimated values of $\beta$. (b) Sorting. (c) Detected clusters.

2) Test if the next bin $l+1$ can be added (appropriate criteria will be discussed below).
3) If yes, then add this bin to the cluster, increase $l$ and go to Step 2.
4) If not, then the end of the cluster has been found. If the number of the bins in the cluster is greater than a threshold, add this cluster to the database. Increase $l$ by one, mark $l$ as the beginning of a new cluster, and go to Step 2.

The test of whether the next bin can be added is based on the assumption of small differences in neighboring bins and large differences between the signals. With $\beta > 0$ a typical clustering test for two neighboring bins has the following form: when

$$\beta_H(\omega_l) > k_1 \cdot \beta_L(\omega_l) \tag{15}$$
$$\beta_H(\omega_{l+1}) > k_1 \cdot \beta_L(\omega_{l+1}) \tag{16}$$

then these two bins can be clustered. If the cluster already contains more bins a more sophisticated set of rules can be used. In Appendix A, we present such a set.

### B. Calculation of the Cluster Boundaries Using Parameter $\alpha$

The basic idea of clustering can also be used for building the clusters using $\alpha$. However, the statistics of $\alpha(\omega)$ are different and the approach has to be modified. As $\alpha(\omega)$ can be compared to the energy, the values at higher frequencies become very low and are not distinguishable. By taking the logarithm of $\alpha$, a sufficient discrimination can be achieved

$$\alpha_H(\omega) = \max[\log(\alpha(\omega,1)), \log(\alpha(\omega,2))] \tag{17}$$
$$\alpha_L(\omega) = \min[\log(\alpha(\omega,1)), \log(\alpha(\omega,2))]. \tag{18}$$

The variances of sequencing $\alpha_H(\omega)$ and $\alpha_L(\omega)$ do not depend on the absolute value. The differences are more important

$$\tilde{\alpha}(\omega) = |\alpha_H(\omega) - \alpha_L(\omega)|. \tag{19}$$

A typical clustering test has then the following form:

$$\tilde{\alpha}(\omega_l) > k_1' \tag{20}$$
$$\tilde{\alpha}(\omega_{l+1}) > k_1'. \tag{21}$$

An extended set of testing rules is given in Appendix B.

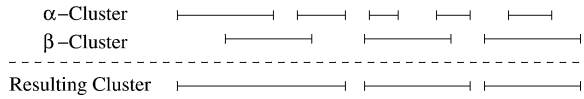


Fig. 5. Cluster joining. Overlapping parts of both types are used for merging. The results are fewer and larger clusters.

### C. Extension to Multiple Sources

When there are more than two sources the algorithm can be easily extended. For this the algorithm is applied multiple times. At first the two largest values of $\beta$ or respectively $\alpha$ are processed. Then the value is removed and the procedure is applied again until all sources have been processed. For an increased performance a analogous procedure can be applied to the bottommost values.

### D. Calculation of Cluster Correlations

When both sets of clusters have been calculated based on the $\alpha$ and $\beta$ parameters, a simple merge is done as follows. If clusters formed by the $\alpha$ and $\beta$ methods overlap, these clusters can be joined on the basis of the overlapping parts. Fig. 5 shows an example of such a merging process. The resulting clusters are bigger and usually occupy most of the bins, so the gaps between them are small. Therefore, an assumption of highly correlated envelopes of bins of neighboring clusters can be made.

For aligning the clusters we follow the idea of dyadic sorting [10]. This algorithm shows the best performance when there are already ranges of correctly de-permuted bins, which is exactly the case when doing the $\alpha\beta$ clustering. So we only need to calculate the value of $r$ according to (11) for all bins of two neighboring clusters, and the highest value of $r$ gives the correct permutation. If the clusters are very large, this leads to a comparison of distinct bins within the clusters for which the assumption of highly correlated envelopes may not be true. So, for a better performance, the comparison can be restricted to smaller parts of the clusters in the vicinity of the corresponding cluster boundaries.

The final step is to calculate the permutation of the remaining bins. By using the clustered bins as a reference frame the permutation is calculated again using (11).

### E. Analysis and Synthesis Filterbank

For the transformation from the time to the time-frequency domain, the blockwise STFT can be used [30]. By using an appropriate window, like the Hann or Hamming window, perfect reconstruction can be achieved, provided that sufficient oversampling is performed. An oversampled version is also advisable for better performance of the ICA algorithm in every bin. This means, a window shift of $(K/d)$ is performed with $d$ being a power of $2, d > 2$, and $K$ being the window length.

The proposed algorithm for solving the permutation problem models every bin using the generalized Gaussian distribution and utilizes the small variance of the parameters in neighboring bins. However, there are some cases in which this variance is too big to correctly identify the permutation. There exists an easy but very powerful extension to the analysis filterbank which enables us to overcome this handicap. The idea is to use additional bins, so the differences of the parameters become smaller. This can be achieved by using an FFT-length that is longer than the Hann or Hamming window length (i. e., using zero padding). A feasible value for the frequency oversampling is a factor of 2 or even 4. Higher values than 4 usually do not yield any further improvements. This approach leads to more bins to be separated and de-permuted which seems to make the problem more difficult. However, the properties of the subband signals actually make the clustering procedure much more reliable.

When just increasing the window length the number of independent bins grows and therefore the permutation problem gets worse. By increasing only the FFT-length, the number of bins grows also, but as they have overlapping spectra they are not independent. As the neighboring bins share parts of their signals the values of the GGD-parameters cannot vary much. Therefore the clustering procedure can use much stricter constraints and calculate the cluster more precisely.

The biggest problem that comes along with the extended FFT length is the extended length of the unmixing filter impulse responses. When the number of separated bins is doubled, the length of the unmixing filter's impulse responses gets doubled too. The situation gets even worse, as the independent multiplication in every bin results in circular convolution instead of the desired linear convolution. Such long filters then may result in clearly audible reverberation. As a remedy, this issue can be resolved using a two-stage approach. The reduction of the filter length can be performed by throwing away the unessential bins. For example, using an FFT length that is four times the window length, it is sufficient to use just every fourth bin to reconstruct the signal correctly. This means the additional bins, which are used for identifying the correct permutations, can be skipped for the synthesis of the time signal. The still remaining problem of circular convolution can be resolved using the method proposed in [31]. There, the authors proposed a technique for spectral smoothing which is able to reduce these artifacts.

TABLE I
COMPARISON OF CLUSTER SIZES FOR THE ARTIFICIAL DATA

| | $\alpha$-Cluster | $\beta$-Cluster | Result Cluster |
|---|---|---|---|
| Number | 114 | 98 | 61 |
| Clustered bins | 3121 | 3338 | 3618 |
| Avg. Cluster Size | 27.38 | 34.06 | 59.31 |

## IV. SIMULATIONS

### A. Artificial Data

The first tests of the algorithm were made on perfectly separated signals. For this, two test signals were transformed into the time-frequency domain. No ICA algorithm was used in this experiment, but an arbitrary permutation in every bin was performed. As the permutation was known, the new algorithm could be tested using this ground truth.

For different setups, the algorithm was able to de-permute all bins correctly. In the following, one exemplary setup is discussed closely. The test signals were obtained from [32]. This data set consists of eight seconds long speech recordings sampled at 8 kHz. The chosen parameters were a Hanning window of length 2048, a window shift of 256, and an FFT-length of 8192. As the signals are real valued, 4097 bins have to be separated and de-permuted. Using this setup, one bin represents a frequency range of approximately 1 Hz and has 228 data points.

In Table I, a summary of the clustering stage is shown. The algorithm from [17] using just the exponent $\beta$ is able to cluster 3338 bins into 98 clusters. Using just the $\alpha$ parameter the algorithm performs slightly worse. It clusters less bins, and the average cluster size is also smaller.

When combing the GGD parameters $\alpha$ and $\beta$ as proposed in Section III, the real power of the algorithm can be recognized. Both methods are able to cluster different bins and therefore the number of total clustered bins rises. The even more important fact is that the clusters of both sets are partially overlapping, so that they can be joined together. The result is an almost doubled average cluster size.

In conclusion, although the clustering procedure using the parameter $\alpha$ alone does not perform as well as the one based on $\beta$, the benefits of using a combination of both clustering methods can be clearly seen. The next stage of calculating permutations between the clusters can be done with much more certainty as the clusters are on average double the size. Furthermore, as the number of clusters is reduced, the number of possible wrong block permutation is reduced.

In this setup, the subsequent stage of calculation of block correlations was performed correctly in all cases. Using the de-permuted clusters as a reference, the remaining single bins could also be correctly assigned.

### B. Real-World Data

For the tests with real-world signals, again the data from [32] was used. The recordings were separated in the frequency bins using 400 iterations of the gradient-based update rule (1). Other

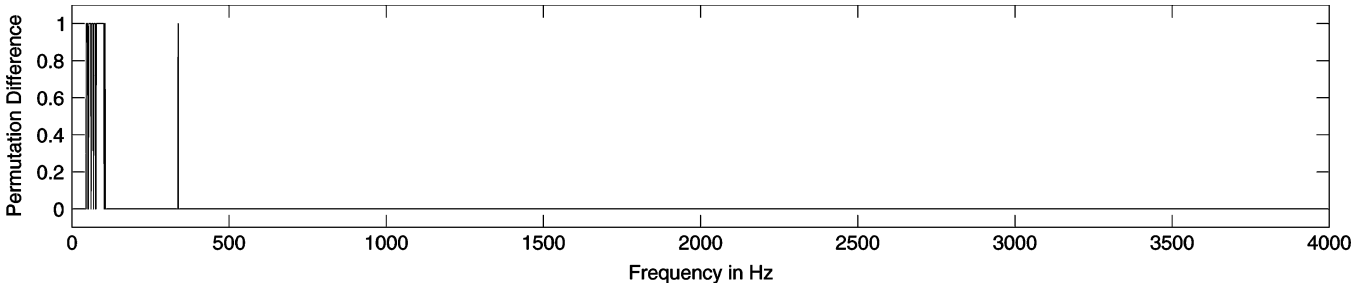


Fig. 6. Differences between the non-blind and $\alpha\beta$ algorithm. A value of 1 indicates a different assignment.

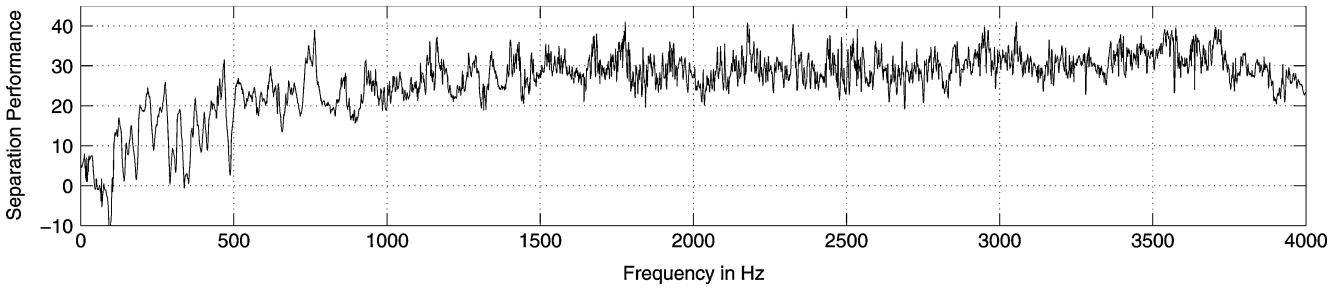


Fig. 7. Separation ratio for single bins. Bins over 500 are separated well, but quite few of the lower frequencies are not. Below 100, a small range has negative separation, which indicates a wrong permutation.

TABLE II
COMPARISON OF CLUSTER SIZES FOR THE REAL-WORLD DATA

| | $\alpha$-Cluster | $\beta$-Cluster | Result Cluster |
|---|---|---|---|
| Number | 114 | 107 | 64 |
| Clustered bins | 3079 | 3306 | 3519 |
| Avg. Cluster Size | 27.00 | 30.90 | 54.98 |

TABLE III
COMPARISON OF SEPARATION PERFORMANCE

| | Left Channel | Right Channel | Overall |
|---|---|---|---|
| $\alpha\beta$ Algorithm | 18.4 | 18.4 | 18.4 |
| Non blind | 18.8 | 18.1 | 18.4 |

parameters were the same as for the synthetic data. After separation the new clustering procedure was applied, and the time signals were restored.

With real-world signals, the separation in the single bins is not always perfect and therefore a reduced performance of the clustering algorithm can be expected. Table II shows the resulting cluster sizes for the real-world data. As the comparison of Tables I and II reveals, under real-world conditions, either the number of clusters is increased or the number of clustered bins is reduced.

The separation performance can be computed if the single contributions of the signals to the microphones are known. For this the signal-to-interference ratio

$$\mathrm{SIR}_{y_i} = 10\log_{10}\frac{E[(g_{ii}(n) * s_i(n))^2]}{E\left[\left(\sum_{j=1,j\neq i}^{N} g_{ij}(n) * s_j(n)\right)^2\right]} \tag{22}$$

with $g_{ij}(n) = w_i(n) * h_j(n)$ can be used. Using the described setup, an overall separation of 18.4 dB has been achieved. See Table III for details.

As the original signals are available in this experiment, the de-permutation can alternatively be done nonblindly by comparing the separated bins with the original sources. Such de-permutation can be seen as an upper bound for the achievable separation performance. As shown in Table III, the separation in the different channels varies minimally but this has no influence on the overall performance. The comparison with the nonblind de-permutation approach clearly shows that the new algorithm performs very well as it is able to achieve the same overall separation ratio. Further improvements can only be made in the separation stage.

The bins that were assigned differently by the nonblind and the $\alpha\beta$ algorithm are shown in Fig. 6. Affected are few bins below 100 Hz and one bin at 348 Hz. In Fig. 7, the separation ratio in every bin is shown. When looking closely at the data, it becomes clear that the single bin is a case where the separation failed and therefore both permutations are equally bad. The permutation of this bin has no influence on the overall performance. Furthermore this special bin has not been assigned to a cluster and the calculation in the last stage failed.

The bins below 100 Hz are also partially wrongly permuted. As the signals are speech signals, this range has just little energy and is not relevant for speech intelligibility. The wrong permutation is just measurable in terms of separation performance in the single channels. The overall performance stays unaffected.

The proposed algorithm is able to de-permute all relevant bins. In the frequency ranges where the separation in the single bins is high enough the improved clustering procedure is able to cluster wide ranges of bins. These clusters are big enough for the correlation approach to be successful. In the lower frequencies where some bins are poorly separated, the clustering is still performing well. The aligning of the remaining single bins suffers the known issues. However, as these bins are poorly separated, the false permutations do not harm the overall performance.

TABLE IV
COMPARISON OF SEPARATION PERFORMANCE FOR MULTIPLE SOURCES

| | 2 channels | 3 channels | 4 channels |
|---|---|---|---|
| $\beta$ Algorithm | 13.1 | 5.5 | 1.9 |
| $\alpha\beta$ Algorithm | 18.4 | 12.3 | 8.3 |
| Method from [9] | 3.1 | 1.4 | 0 |
| DOA - Algorithm | 17.3 | 12.4 | 9.2 |
| Non blind | 18.4 | 13.6 | 9.9 |

Using the dataset from [32] it was possible to test the new algorithm also on multiple sources. In Table IV, the results are shown. One can clearly see the improvements over the method in [17] obtained through the use of the additional parameter. While the algorithm using $\beta$ is able to separate two sources reasonably well, it fails for multiple sources. The new algorithm performs better in all cases and is able to separate even multiple sources. The separation performance is quite well and is comparable to other state of the art de-permutation algorithms like the one in [12]. The method from [9], which only looks at adjacent bins, could hardly find the correct permutations.

While the method in [12] needs the a priori information of the sensor array setup, our method is completely blind and does not rely on any prior knowledge. Since the proposed algorithm and the one from [12] use entirely different information for de-permutation, they may even be combined in order to further improve the performance.

## V. SUMMARY

In this paper, we presented a new way to solve the permutation problem in convolutive blind source separation. In our method, every bin is modeled using the generalized Gaussian distribution. The small variance of the parameters between the bins are used to calculate the correct permutations. As the differences are not high enough for all bins, a clustering procedure for large portions of the data has been proposed. This presorted data is then de-permuted using other known algorithms. The performance of the algorithm has been tested on synthetic and real-world data. The comparison with a non-blind de-permutation shows a very good performance. The synthetic data was perfectly de-permutated, and for the real-world data, all significant parts have been de-permuted. The statistical information exploited in our method is complementary to the directivity information used in other techniques, and future works will be directed toward combining both approaches.

## APPENDIX A
## RULE SET FOR CLUSTERING USING $\beta$

The sequencing bin can be added if one of the conditions (23), (24), (29), or (30) is satisfied

$$\begin{aligned} &\beta_H(\omega_l) > k_1 \cdot \beta_L(\omega_l) \\ \text{and}\quad &\beta_H(\omega_{l+1}) > k_1 \cdot \beta_L(\omega_{l+1}) \end{aligned} \tag{23}$$

or

$$\begin{aligned} &\beta_H(\omega_l) > k_2 \cdot \beta_L(\omega_l) \\ \text{and}\quad &\beta_H(\omega_{l+1}) > k_2 \cdot \beta_L(\omega_{l+1}) \\ \text{and}\quad &\beta_H(\omega_l) < \beta_H(\omega_{l+1}) \\ \text{and}\quad &\beta_L(\omega_l) > \beta_L(\omega_{l+1}) \end{aligned} \tag{24}$$

with some $k_1 > k_2 > 1$.

TABLE V
STABILITY ANALYSIS FOR CLUSTERING PARAMETERS USING $\beta$. C STANDS FOR CORRECT CLUSTERING, WHILE F MARKS CASES WHERE SOME BINS HAVE BIN FALSELY CLUSTERED

| | | | | |
|---|---|---|---|---|
| $k_1$: | 1.40 | *1.30* | 1.25 | 1.20 |
| Clustered Bins (C/F): | 3292 C | 3307 C | 3334 C | 3364 F |
| $k_2$: | 1.15 | *1.10* | 1.07 | 1.05 |
| Clustered Bins (C/F): | 3203 C | 3307 C | 3398 C | 3455 F |
| $k_3$: | 1.25 | *1.20* | 1.18 | 1.15 |
| Clustered Bins (C/F): | 3287 C | 3307 C | 3326 C | 3228 F |
| $k_4$: | 0.050 | 0.045 | *0.040* | 0.035 |
| Clustered Bins (C/F): | 3338 F | 3323 C | 3307 C | 3278 C |
| $k_5$: | 1.07 | *1.06* | 1.04 | 1.03 |
| Clustered Bins (C/F): | 3292 C | 3307 C | 3335 C | 3363 F |

When a cluster already contains several bins, a test for flatness can be performed. With

$$D_H^1 = \beta_H(\omega_l) - \beta_H(\omega_{l-1}) \tag{25}$$
$$D_H^2 = \beta_H(\omega_{l+1}) - \beta_H(\omega_l) \tag{26}$$
$$D_L^1 = \beta_L(\omega_l) - \beta_L(\omega_{l-1}) \tag{27}$$
$$D_L^2 = \beta_L(\omega_{l+1}) - \beta_L(\omega_l) \tag{28}$$

the test is one of the following:

$$\begin{aligned} &\min(\beta_H(\omega_{l-1}), \beta_H(\omega_l), \beta_H(\omega_{l+1})) \\ &< k_3 \max(\beta_L(\omega_{l-1}), \beta_L(\omega_l), \beta_L(\omega_{l+1})) \end{aligned} \tag{29}$$

or

$$\begin{aligned} &\left|D_H^2 - D_H^1\right| < k_4 \\ \text{and}\quad &\left|D_L^2 - D_L^1\right| < k_4 \\ \text{and}\quad &-\beta_H(\omega_{l+1}) > k_5 \cdot \beta_L(\omega_{l+1}) \end{aligned} \tag{30}$$

with $k_3 > 1$, $k_5 > 1$, and $k_4 > 0$.

Table V shows the number of clustered frequency bins and the correctness of all cluster assignments for different choices for $k_1 - k_5$, using the same signals as in Section IV. Essentially, the larger $k_1$, $k_2$, $k_3$, and $k_5$ are, the safer the clustering is. However, with increasing parameters the cluster size decreases. Regarding $k_4$, the cluster assignments become safer for smaller values. The following parameters have been found to be a good compromise of cluster size and correctness of the clustering and were used in the experiments: $k_1 = 1.3$, $k_2 = 1.1$, $k_3 = 1.2$, $k_4 = 0.05$, and $k_5 = 1.06$.

## APPENDIX B
## RULE SET FOR CLUSTERING USING $\alpha$

For clustering according to the parameter $\alpha$, the following rules can be used. One of the conditions (31), (32), (36), and (37) has to be satisfied

$$\begin{aligned} &\tilde{\alpha}(\omega_l) > k_1' \\ \text{and}\quad &\tilde{\alpha}(\omega_{l+1}) > k_1' \end{aligned} \tag{31}$$

TABLE VI
STABILITY ANALYSIS FOR CLUSTERING PARAMETERS USING $\alpha$. C STANDS FOR CORRECT CLUSTERING, WHILE F MARKS CASES WHERE SOME BINS HAVE BIN FALSELY CLUSTERED

| $k_1'$: | 0.5 | *0.4* | 0.3 | 0.2 |
|---|---|---|---|---|
| Clustered Bins (C/F): | 3109 C | 3179 C | 3179 C | 3296 F |
| $k_2'$: | 0.30 | *0.25* | 0.20 | 0.15 |
| Clustered Bins (C/F): | 3174 C | 3179 C | 3208 C | 3248 F |
| $k_3'$: | 0.25 | *0.2* | 0.15 | 0.09 |
| Clustered Bins (C/F) | 3131 C | 3179 C | 3242 C | 3385 F |
| $k_4'$: | 0.03 | *0.05* | 0.09 | 0.12 |
| Clustered Bins (C/F) | 3159 C | 3179 C | 3222 C | 3255 F |

or

$$\begin{aligned} &\tilde{\alpha}(\omega_l) < \tilde{\alpha}(\omega_{l+1}) \\ \text{and} \quad &\tilde{\alpha}(\omega_l) > k_2' \end{aligned} \tag{32}$$

with $k_1' > k_2' > 0$.

Using

$$\tilde{D}^1 = \tilde{\alpha}(\omega_l) - \tilde{\alpha}(\omega_{l+1}) \tag{33}$$

$$\tilde{D}^2 = \tilde{\alpha}(\omega_{l-1}) - \tilde{\alpha}(\omega_l) \tag{34}$$

$$\tilde{D}^3 = \tilde{\alpha}(\omega_{l-2}) - \tilde{\alpha}(\omega_{l-1}) \tag{35}$$

bigger clusters can be tested

$$\min(\tilde{\alpha}(\omega_{l-1}), \tilde{\alpha}(\omega_l), \tilde{\alpha}(\omega_{l+1})) > k_3' \tag{36}$$

or

$$\max(|\tilde{D}^1|, |\tilde{D}^2|, |\tilde{D}^3|) < k_4' \tag{37}$$

with $k_3' > 0$ and $k_4' > 0$.

Results for different parameter choices for $k_1'$, $k_2'$, $k_3'$, $k_4'$ are given in Table VI. For $k_1'$, $k_2'$, and $k_3'$, the clustering becomes safer for increasing values. For $k_4'$ the opposite is the case. The following values were selected for the experiments, as they give a good compromise of cluster size and correctness: $k_1' = 0.4$, $k_2' = 0.25$, $k_3' = 0.2$, and $k_4' = 0.05$.

## REFERENCES


[1] S. Amari, A. Cichocki, and H. H. Yang, , D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds., "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1996, vol. 8, pp. 757–763.

[2] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 9, pp. 1483–1492, 1997.

[3] J.-F. Cardoso and A. Soulomiac, "Blind beamforming for non-Gaussian signals," *Proc. Inst. Elect. Eng. F*, vol. 140, no. 6, pp. 362–370, Dec. 1993.

[4] S. C. Douglas, H. Sawada, and S. Makino, "Natural gradient multichannel blind deconvolution and speech separation using causal FIR filters," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1, pp. 92–104, Jan. 2005.

[5] R. Aichner, H. Buchner, S. Araki, and S. Makino, "Online time-domain blind source separation of nonstationary convolved signals," in *Proc. 4th Int. Symp. Ind. Compon. Anal. Blind Signal Separation (ICA2003)*, Nara, Japan, Apr. 2003, pp. 987–992.

[6] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, no. l–3, pp. 21–34, 1998.

[7] T. Mei, J. Xi, F. Yin, A. Mertins, and J. F. Chicharo, "Blind source separation based on time-domain optimizations of a frequency-domain independence criterion," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 6, pp. 2075–2085, Nov. 2006.

[8] M. Z. Ikram and D. R. Morgan, "Exploring permutation inconsistency in blind separation of speech signals in a reverberant environment," in *Proc. ICASSP'00*, 2000, vol. II, pp. 1041–1044.

[9] S. Ikeda and N. Murata, "A method of blind separation based on temporal structure of signals," in *Proc. Int. Conf. Neural Inf. Proc.*, 1998, pp. 737–742.

[10] K. Rahbar and J. P. Reilly, "A frequency domain method for blind source separation of convolutive audio mixtures," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 832–844, Sep. 2005.

[11] J. Anemüller and B. Kollmeier, "Amplitude modulation decorrelation for convolutive blind source separation," in *Proc. 2nd Int. Workshop Ind. Compon. Anal. Blind Signal Separation*, 2000, pp. 215–220.

[12] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 5, pp. 530–538, Sep. 2004.

[13] W. Wang, J. A. Chambers, and S. Sanei, "A novel hybrid approach to the permutation problem of frequency domain blind source separation," in *Lecture Notes in Computer Science*. New York: Springer, 2004, vol. 3195, pp. 532–539.

[14] M. Z. Ikram and D. R. Morgan, "Permutation inconsistency in blind speech separation: Investigation and solutions," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1, pp. 1–13, Jan. 2005.

[15] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Blind source separation of 3-d located many speech signals," in *Proc. 2005 IEEE Workshop Applicat. Signal Process. Audio Acoust.*, Oct. 2005, pp. 9–12.

[16] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Frequency-domain blind source separation of many speech signals using near-field and far-field models," *EURASIP J. Appl. Signal Process.*, p. 13, Article ID 83683.

[17] R. Mazur and A. Mertins, , M. E. Davies, Ch. J. James, S. A. Abdallah, and M. D. Plumbley, Eds., "Solving the permutation problem in convolutive blind source separation," in *Independent Component Analysis and Signal Separation*. New York: Springer, 2007, vol. 4666, pp. 512–519.

[18] S. Choi, A. Cichocki, and S. Amari, T. Constantinides, S. Y. Kung, M. Niranjan, and E. Wilson, Eds., "Flexible independent component analysis," *Neural Netw. Signal Process. VIII*, pp. 83–92, 1998.

[19] S. Gazor and W. Zhang, "Speech probability distribution," *IEEE Signal Process. Lett.*, vol. 10, no. 7, pp. 204–207, Jul. 2003.

[20] K. Kokkinakis and A. K. Nandi, *Multichannel Speech Separation Using Adaptive Parameterization of Source PDFs*. New York: Springer, 2004, vol. 3195, Lecture Notes in Computer Science.

[21] M. K. Varanasi and B. Aazhang, "Parametric generalized gaussian density estimation," *Acoust. Soc. Amer. J.*, vol. 86, no. 4, pp. 1404–1415, Oct. 1989.

[22] K. Kokkinakis and A. K. Nandi, "Exponent parameter estimation for generalized Gaussian probability density functions with application to speech modeling," *Signal Process.*, vol. 85, no. 9, pp. 1851–1858, Sep. 2005.

[23] K. Rahbar and J. P. Reilly, "Blind source separation of convolved sources by joint approximate diagonalization of cross-spectral density matrices," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 7–11, 2001, vol. 5, pp. 2745–2748.

[24] R. Mazur and A. Mertins, "Reducing reverberation effects in convolutive blind source separation," in *Proc. Eur. Signal Process. Conf.*, Florence, Italy, Sep. 2006, CD-ROM.

[25] K. Matsuoka, "Minimal distortion principle for blind source separation," in *Proc. 41st SICE Annu. Conf.*, Aug. 5–7, 2002, vol. 4, pp. 2138–2143.

[26] D.-T. Pham, Ch. Serviere, and H. Boumaraf, "Blind separation of convolutive audio mixtures using nonstationarity," in *Proc. 4th Int. Conf. Ind. Compon. Anal. Blind Signal Separation (ICA'03)*, Nara, Japan, Apr. 2003, pp. 975–980.

[27] Ch. Serviere and D. T. Pham, "Permutation correction in the frequency domain in blind separation of speech mixtures," *EURASIP J. Appl. Signal Process.*, no. 1, p. 16, 2006, Article ID 75206, 2006.

[28] H. Sawada, S. Araki, R. Mukai, and S. Makino, "Grouping separated frequency components with estimating propagation model parameters in frequency-domain blind source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 5, pp. 1592–1604, Jul. 2007.
[29] E. Vincent, , M. E. Davies, Ch. J. James, S. A. Abdallah, and M. D. Plumbley, Eds., "Complex nonconvex lp norm minimization for underdetermined source separation," in *Independent Component Analysis and Signal Separation*. New York: Springer, 2007, vol. 4666, pp. 430–437, LNCS.
[30] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, ser. Signal Proc. Series. Englewood Cliffs: Prentice-Hall, 1978.
[31] H. Sawada, R. Mukai, S. de la Kethulle, S. Araki, and S. Makino, "Spectral smoothing for frequency-domain blind source separation," in *Proc. Int. Workshop Acoust. Echo Noise Control (IWAENC'03)*, Sep. 2003, pp. 311–314.
[32] [Online]. Available: http://www.kecl.ntt.co.jp/icl/signal/sawada/demo/bss2to4/index.html.

**Radoslaw Mazur** was born in Wroclaw, Poland, in 1976. He received the Diplominformatiker degree from the University of Oldenburg, Oldenburg, Germany, in 2004. He is currently pursuing the Ph.D. degree at the Institute for Signal Processing, University of Lübeck, Lübeck, Germany.

He was an Assistant Researcher at the Department of Physics, University of Oldenburg, from 2004 to 2006, and then joined the University of Lübeck. His current research interests are digital signal and audio processing, with a special focus on blind source separation.

**Alfred Mertins** (M'96–SM'08) received the Dipl.-Ing. degree from the University of Paderborn, Paderborn, Germany, in 1984, the Dr.-Ing. degree in electrical engineering and the Dr.-Ing. habil. degree in telecommunications from the Hamburg University of Technology, Hamburg, Germany, in 1991 and 1994, respectively.

From 1986 to 1991, he was a Research Assistant at the Hamburg University of Technology, and from 1991 to 1995, he was a Senior Scientist at the Microelectronics Applications Center Hamburg. From 1996 to 1997, he was with the University of Kiel, Kiel, Germany, and from 1997 to 1998 with the University of Western Australia. In 1998, he joined the University of Wollongong, where he was an Associate Professor of Electrical Engineering. From 2003 to 2006, he was a Professor in the Faculty of Mathematics and Science at the University of Oldenburg, Oldenburg, Germany. In November 2006, he joined the University of Lübeck, Lübeck, Germany, where he is a Professor and Director of the Institute for Signal Processing. His research interests include speech, audio, and image processing, wavelets and filter banks, pattern recognition, and digital communications.