Underdetermined Anechoic Blind Source Separation via ℓ^q-Basis-Pursuit with $q < 1$

Rayan Saab, Özgür Yılmaz, Martin J. McKeown, Rafeef Abugharbieh

***Abstract*—In this paper, we address the problem of underdetermined Blind Source Separation (BSS) of anechoic speech mixtures. We propose a demixing algorithm that exploits the sparsity of certain time-frequency expansions of speech signals. Our algorithm merges ℓ^q-basis-pursuit with ideas based on the degenerate unmixing estimation technique (DUET) [1]. There are two main novel components to our approach: (1) Our algorithm makes use of *all* available mixtures in the anechoic scenario where both attenuations and arrival delays between sensors are considered, *without* imposing any structure on the microphone positions. (2) We illustrate experimentally that the separation performance is improved when one uses l^q-basis-pursuit with $q < 1$ compared to the $q = 1$ case. Moreover, we provide a probabilistic interpretation of the proposed algorithm that explains why a choice of $0.1 \leq q \leq 0.4$ is appropriate in the case of speech. Experimental results on both simulated and real data demonstrate significant gains in separation performance when compared to other state-of-the-art BSS algorithms reported in the literature. A preliminary version of this work can be found in [2].**

***Index Terms*—blind source separation, BSS, sparse signal representation, DUET, time-frequency representations, Gabor expansions, underdetermined signal un-mixing, over-complete representations, basis pursuit**

I. Introduction

Blind source separation (BSS) is the term used to describe the process of extracting some underlying original source signals from a number of observable mixture signals, where the mixing model is either unknown or the knowledge about the mixing process is limited. Numerous methods for solving the BSS problem in various contexts and under various assumptions and conditions were proposed in the literature. Early approaches concentrated on tackling even-determined and over-determined demixing. Independent component analysis (ICA), first presented by Jutten and Herault [3] [4], and later developed in an information maximization framework by Bell and Sejnowski [5] pioneered those early approaches. Other early papers on source separation include, for example, [6] and [7]. For an extensive overview of ICA see [8].

ICA extracts n sources from n recorded mixtures under the crucial assumption that the underlying source signals are independent. Lewicki et al. [9] and Lee et al. [10] generalized this technique to the underdetermined instantaneous BSS case, where the number of available recorded mixtures are less than the underlying sources, by proposing a method for learning the over-complete basis using a probabilistic model of the observed data. This technique, however, is still constrained to the instantaneous mixing model of time-domain signals. Other methods were proposed that cater to anechoic demixing, where both signal attenuations and arrival delays between sensors are considered. Anemuller [11] used a complex ICA technique to extract an equal number of sources from mixtures in various separate spectral bands to solve the BSS problem for electro-encephalographic (EEG) data. This approach, however, is complicated by the need to either identify whether any sources extracted from different spectral bands correspond to one another and therefore solve a scaling and permutation problem, or to assume that the underlying sources are spectrally disjoint, i.e. confined to localized spectral bands. Other BSS approaches based on source sparsity assumptions in some transform domain were recently proposed ([12], [13], [14]) and have come to be known as 'sparse methods'. The assumption of these methods is that the sources have a sparse representation in some given basis. These approaches typically employ an instantaneous mixing model to solve the BSS problem in the underdetermined case by adopting l^1-minimization approaches that maximize sparsity. Vielva et al. [15] considered the case of underdetermined instantaneous BSS where source densities are parameterized by a sparsity factor, and presented a maximum a posteriori method for separation, and [16] focused on the estimation of the mixing matrix for underdetermined BSS under the assumption of sparsity. Yılmaz and Rickard [1], see also [17], developed a practical algorithm for underdetermined anechoic mixing scenarios called the degenerate unmixing estimation technique (DUET), which exploits sparsity in the short-time Fourier transform (STFT) domain, and uses masking to extract several sources from two mixtures. This approach is, however, restricted to using two mixtures only. Bofill [18] deals with the anechoic underdetermined scenario as well and estimates the attenuation coefficients by using a scatter plot technique and the delays by maximizing a kernel function. To extract the sources, [18] solves a complex constrained l^1-minimization problem via second order cone programming. This algorithm, like DUET, uses only two of the available mixtures. More recently, Melia and Rickard [19] presented a technique which extends DUET and is able to utilize multiple microphone readings obtained from a uniform linear array of sensors. This allows the use of all available mixtures at the expense of

Rayan Saab is with the Department of Electrical and Computer Engineering, The University of British Columbia. Support provided by the Whitaker Foundation (RG-00-0357) (RS & MJM)

Özgür Yılmaz is with the Department of Mathematics, The University of British Columbia. He was partially supported by the Natural Sciences and Engineering Research Council of Canada.

Rafeef Abugharbieh is with the Department of Electrical and Computer Engineering, The University of British Columbia. She was partially supported by the Natural Sciences and Engineering Research Council of Canada.

Martin J. McKeown is with Department of Medicine (Neurology), Pacific Parkinson's Research Centre, The University of British Columbia

imposing structure on the sensor array.

A comprehensive survey of sparse and non-sparse methods in source separation can be found in [20].

In this paper, we employ a two-step demixing approach for BSS problems for the general case of anechoic mixing in the underdetermined case. Such a two step approach was adopted in, e.g., [12], [1] and [14] and formalized by Theis and Lang [21]. The two step approach is comprised of blind recovery of the mixing model followed by recovery of the sources. The novel aspects of our approach can be summarized as follows:

- We generate feature vectors that incorporate both attenuation and delay parameters for a large, in principle arbitrary, number of mixtures in the underdetermined BSS case. Thus, unlike DUET, our algorithm makes use of all available mixtures, both in the mixing model recovery and sources extraction stages. Moreover, unlike [19], we do not need to impose a pre-determined structure on the sensor array.
- We compare the performance of source extraction algorithms based on ℓ^q minimization and ℓ^q-basis-pursuit for values $0 \leq q \leq 1$ in STFT domain, and illustrate that the best separation performance is obtained for $0.1 \leq q \leq 0.4$.
- Generalizing the approach of [22] to the complex-valued case, we provide an interpretation based on the empirical probability distribution of speech in the STFT domain, which justifies the use of ℓ^q minimization with $0.1 \leq q \leq 0.4$.
- Clearly ℓ^q-basis-pursuit, where $0 \leq q < 1$ is a combinatorial problem. On the other hand, the size of the problems in BSS of speech signals is typically small. In the experiments we conducted, we observed that solving ℓ^q-basis-pursuit combinatorially and solving ℓ^1-basis-pursuit via convex programming spend comparable computation time. Thus, our approach is computationally tractable.

II. Mixing Model

In the anechoic mixing model, one has n sources $s_1(t), \dots, s_n(t)$ and m mixtures $x_1(t), \dots, x_m(t)$ such that

$$x_i(t) = \sum_{j=1}^{n} a_{ij} s_j(t - \delta_{ij}),\ i = 1, 2, \dots, m \tag{1}$$

where a_{ij} and δ_{ij} are scalar attenuation coefficients and time delays, respectively, associated with the path from the j^{th} source to the i^{th} receiver. Without loss of generality one can set $\delta_{1j} = 0$ and scale the source functions s_j such that

$$\sum_{i=1}^{m} |a_{ij}|^2 = 1 \tag{2}$$

for $j = 1, \dots, n$. Throughout the paper, we assume $n > m$, i.e. the number of the sources to be separated exceeds the number of available mixtures, and thus the mixing is *underdetermined*.

The short time Fourier transform (STFT) of a function s with respect to a fixed *window function* W is defined as:

$$F^W[s](\tau, \omega) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} W(t - \tau) s_j(t) e^{-i\omega t} dt \tag{3}$$

which will henceforth be referred to as $\hat{s}(\tau, \omega)$. Practically,

$$\begin{aligned} F^W[s_j(\cdot - \delta)](\tau, \omega) &= \exp(-i\omega\delta) F^W[s_j](\tau - \delta, \omega) \\ &\approx \exp(-i\omega\delta) F^W[s_j](\tau, \omega). \end{aligned} \tag{4}$$

is a realistic assumption as long as the window function W is chosen appropriately. For a detailed discussion on this, see [23].

Now given $x_1, x_2, \dots, x_m$, the problem to be solved is basically one of estimating $s_1, \dots, s_n$ in the general case where $n \geq m$ and n is unknown . Taking the STFT of $x_1,\dots,x_m$ with an appropriate window function W and using (4), the mixing model (1) reduces to

$$\hat{\mathbf{x}}(\tau, \omega) = A(\omega) \hat{\mathbf{s}}(\tau, \omega), \tag{5}$$

where

$$\hat{\mathbf{x}} = [\hat{x}_1 \dots \hat{x}_m]^T,\ \hat{\mathbf{s}} = [\hat{s}_1 \dots \hat{s}_n]^T, \tag{6}$$

and

$$A(\omega) = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} e^{-i\omega\delta_{21}} & \dots & a_{2n} e^{-i\omega\delta_{2n}} \\ \vdots & \vdots & \vdots \\ a_{m1} e^{-i\omega\delta_{m1}} & \dots & a_{mn} e^{-i\omega\delta_{mn}} \end{bmatrix}. \tag{7}$$

Note that via (2) the column vectors of A have unit norms.

The equivalent discrete counterpart is henceforth used to replace the continuous STFT, i.e. the samples of the STFT of s are evaluated on a lattice in the time-frequency plane given by

$$\hat{s}_j[k, l] = \hat{s}_j(k\tau_0, l\omega_0) \tag{8}$$

where τ_0 and ω_0 are the time-frequency lattice parameters. The equivalence is nontrivial and only true if the family $\{e^{il\omega_0 t} W(t - k\tau_0)\ :\ k, l \in \mathbb{Z}\}$ constitutes a *Gabor frame* for the signal space of interest. For this, one needs an appropriately chosen window function W with sufficiently small τ_0 and ω_0. Extensive discussions on Gabor frames can be found, e.g., in [24], [25]. Note that, in the discrete framework, the mixing model can be written as

$$\hat{\mathbf{x}}[k, l] = A(l\omega_0) \hat{\mathbf{s}}[k, l]. \tag{9}$$

with $\hat{\mathbf{x}}, \hat{\mathbf{s}}$ as in (6), and A as in (7). $\hat{\mathbf{s}}[k, l]$ is the Gabor or STFT coefficient of $\mathbf{s}$ at the time-frequency (T-F) point $[k, l]$.

In this paper, we shall use the STFT as the preferred T-F representation. Some of the reasons for this choice are as follows: (i) STFT is linear (unlike, e.g., Wigner-Ville disributions), (ii) STFT converts delays in the time domain to phase shifts in the T-F domain (unlike, e.g., wavelet transforms), (iii) STFT is easy to implement and invert, and (iv) STFT of speech signals are sparse (more so then wavelet transforms [26]), as we shall discuss in section III-A. Note we need the properties (i) and (ii) to write (1) as a matrix equation. Property (ii) is also critical for our blind mixing model recovery algorithm, presented in Section III. Property (iii) is important for computational efficiency and speed. Finally, property (iv) facilitates both the blind mixing model recovery algorithm of Section III and the source extraction algorithm of Section IV.

III. Blind Mixing Model Recovery

A. STFT Sparsity in Speech Signals

In order to estimate the mixing parameters of our model, we shall utilize the observation that time-frequency representations of speech signals are sparse, and thus only a few Gabor coefficients capture most of the signal power, cf., [1], [20], [27]. This has been empirically verified in [1]. Moreover [26] investigates sparsity of speech in the STFT domain as well as in the wavelet domain, with the conclusion that the STFT provides slightly higher sparsity with the proper window choice. Nevertheless, in this paper we present additional experiments to further demonstrate sparsity of speech in the STFT domain.

Fifty speech sources from the TIMIT data base, each consisting of 50,000 samples sampled at 16 kHz are used for this experiment. The speech signals are transformed into the STFT domain using Hamming windows of three different sizes (32 ms, 64 ms, and 128 ms), with an overlap factor of 50%. Figure 1 shows the average cumulative power of the sorted STFT coefficients along with the average cumulative power of both the time domain sources and their Fourier (DFT) coefficients. Table I shows the percentage of coefficients needed to represent 90%, 95% and 98% of the total signal power using the STFT (with varying window sizes), the time domain signal, and the Fourier transformed signals. The results indicate that the STFT with a 64ms window-size demonstrates superior performance in terms of sparsity, capturing 98% of the total signal power with ca 9% of the coefficients only. Figures 2a and 2b further illustrate the sparsity of speech signals in the STFT domain by showing the normalized histograms of the sample magnitudes as well as the magnitudes of the Gabor coefficients, respectively. Figure 2 again illustrates that the STFT transform domain exhibits a much sparser speech signal representation, where the magnitudes of most of the STFT coefficients are very close to zero.

TABLE I
Percentage of Coefficients Needed to Represent Various Percentages of the Total Signal Power

Percentage of the Total Power	Percentage of Points Needed				
	STFT: 32ms	STFT: 64ms	STFT: 128ms	Fourier Domain	Time Domain
90.000	2.6781	**2.2656**	2.6999	16.954	14.098
95.000	5.0167	**4.5710**	5.5212	24.604	23.756
98.000	9.4100	**9.1439**	11.032	34.560	38.970

B. Feature Vector Extraction and Mixing Parameter Estimation

Motivated by the sparsity of the STFT coefficients of speech signals, we shall represent our mixtures $x_1, \ldots, x_m$ in the T-F domain via (9), and propose an algorithm to estimate the matrix $A(l\omega_0)$ for every l. To that end, we first construct a $2m-1$ dimensional feature space where the first m coordinates correspond to the normalized attenuations of the mixtures $x_1, \ldots, x_m$ while the remaining $m-1$ dimensions are the delays of $x_2, \ldots, x_m$ relative to x_1. More precisely, let $\hat{\mathbf{x}}$ be as in (6). First, at each point $[k,l]$ on the T-F lattice, we define the normalized attenuation vector

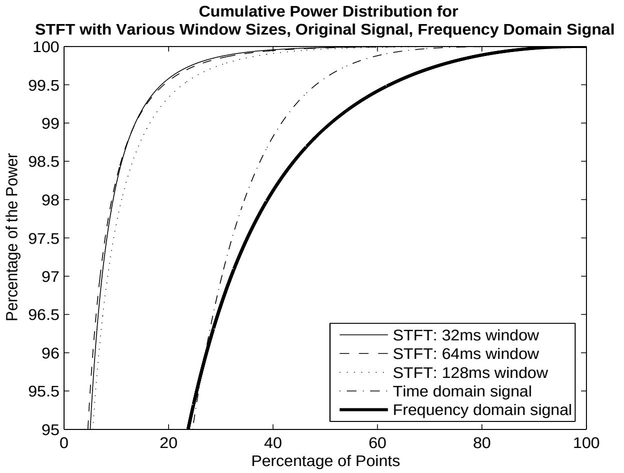


Fig. 1. Average cumulative power of the time domain signals, frequency (Fourier) domain signals and STFT of speech for window sizes of 32ms, 64ms and 128ms. The STFT with 32ms and 64ms window length exhibit a sparser representation of the data (more power is concentrated in fewer coefficients) compared to the original time domain representation and the frequency domain representation.

$$\hat{\mathbf{x}}_{\text{at}}[k,l] := \frac{1}{\|\hat{\mathbf{x}}[k,l]\|} \begin{bmatrix} |\hat{x}_1| & \ldots & |\hat{x}_m| \end{bmatrix} [k,l]. \tag{10}$$

Here $\|\cdot\|$ denotes the Euclidean norm. Note that the resulting $\hat{\mathbf{x}}_{\text{at}}[k,l]$ correspond to points on the unit sphere of $\mathbb{R}^m$. N ext, we calculate the complex phases $-l\omega_0\tilde{\Delta}_{i1}[k,l]$ of mixtures $\hat{x}_i,\ j = 2, \ldots, m$ relative to the mixture $\hat{x}_1$ at each T-F point $[k,l]$, as in [1]. This yields

$$\tilde{\Delta}_{i1}[k,l] := -\frac{1}{l\omega_0} \angle \frac{\hat{x}_i[k,l]}{\hat{x}_1[k,l]}. \tag{11}$$

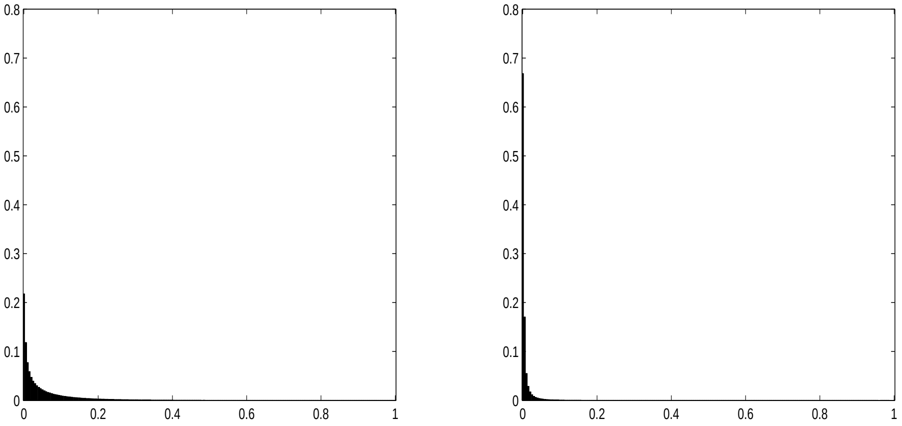
Finally, we append the m-dimensional feature vector $\hat{\mathbf{x}}_{\text{at}}$, defined as in (10), to obtain the $2m-1$ dimensional feature vector $\mathbf{F}[k,l]$ given by

$$\begin{aligned} &\mathbf{F}[k,l] \\ &:= \begin{bmatrix} \left|\frac{\hat{x}_1[k,l]}{\|\hat{\mathbf{x}}[k,l]\|}\right| & \cdots & \left|\frac{\hat{x}_m[k,l]}{\|\hat{\mathbf{x}}[k,l]\|}\right| & \cdots & \tilde{\Delta}_{21}[k,l] & \cdots & \tilde{\Delta}_{m1}[k,l] \end{bmatrix}. \end{aligned} \tag{12}$$

Note that if only one source, say s_J, is active at a certain T-F point $[k,l]$, i.e., $\hat{s}_J[k,l] \neq 0$ and $\hat{s}_j[k,l] = 0$ for $j \neq J$, the feature vector will then reduce to

$$\begin{aligned} \mathbf{F}[k,l] &= \mathbf{F}_J \\ &:= \begin{bmatrix} a_{1J} & \cdots & a_{mJ} & \cdots & \delta_{2J} & \cdots & \delta_{mJ} \end{bmatrix} \end{aligned} \tag{13}$$

where we used the fact that the columns of the mixing matrix A are normalized. Furthermore, we assume that the attenuation coefficients are *positive* real numbers. In this case $\mathbf{F}_J$ does not depend on $[k,l]$, rather it is completely determined by the mixing parameters in the Jth column of the mixing matrix, given in (7). Moreover, the converse is also true, i.e., given $\mathbf{F}_J$, one can extract the mixing parameters. Indeed, the first m coordinates of $\mathbf{F}_J$ yield $a_{1J}, \ldots, a_{mJ}$. On the other hand, $\delta_{1J} = 0$ and the rest of the delay parameters $\delta_{2J}, \ldots, \delta_{mJ}$ are directly given by the last $m-1$ components of $\mathbf{F}_J$. Therefore, if the sources have disjoint T-F representations, i.e. if at any T-F point only one source has a non-zero STFT coefficient



(a)

(b)

Fig. 2. Normalized histograms of (a) time-domain sample magnitudes of 50 speech signals, and (b) STFT coefficient magnitudes of the same 50 sources (with a window length of 64ms and 50% overlap). Note that the STFT coefficients provide a significantly more sparse representation. In both (a) and (b), the values have been normalized to the range [0 1]

at the most, then the feature vector $\mathbf{F}[k,l]$, corresponding to any T-F point $[k,l]$ at which at least one source is active, will be in the set $\{\mathbf{F}_1,\ldots,\mathbf{F}_n\}$. Once this set is obtained, one can compute the mixing parameters using (13) as described above.

In practice, it is *not* realistic to expect that the sources have disjoint T-F representations. However, as discussed in Section III-A as well as in [1], speech signals have sparse Gabor expansions. Therefore, it is highly likely that there will be an abundance of T-F points at which one source is dominant. In other words, there will be several points in the T-F plane where one source has a “high contribution”, i.e., a relatively large STFT coefficient, while other sources have near-zero contributions. (See Assumption 4 of Section III-C for a more detailed discussion about this observation.) Thus, in the feature space, points $\mathbf{F}[k,l]$ will tend to cluster around $\mathbf{F}_j, j=1,\ldots,n$. Based on this, after constructing a feature vector $\mathbf{F}[k,l]$ at every T-F point where the mixtures $|\hat{x}_j|$ are not smaller than a threshold, we perform K-means clustering to obtain the cluster centers. Other authors also use K-means [28], as well as various different techniques for clustering-based methods in various BSS problems. For example [29] uses a Fuzzy C-means approach, while [28] detects attenuations by locating horizontal lines using an elaborate technique based on a relaxed sparsity assumption. At this point, it is important to note that we do *not* claim that K-means is an optimal method for detecting the attenuations and delays. We simply propose K-means as a generic method that can be used to estimate the mixing parameters as the proposed feature vectors tend to cluster and the cluster centers identify the mixing parameters.

The cluster centers, obtained as discussed above, yield estimates $\tilde{a}_{ij}$ and $\tilde{\delta}_{ij}$ of the mixing parameters a_{ij} and δ_{ij} which are computed again using (13). At this point we assumed that n, the number of sources, is *known* a priori. This issue will be revisited in Assumption 3 of Section III-C where the effects of incorrect estimation of the number of sources is further discussed.

The proposed **Parameter Estimation Algorithm** can be summarized as follows:

1) Compute the mixture vector $\hat{\mathbf{x}}[k,l]$, as in (6) at every T-F point $[k,l]$.
2) At every T-F point $[k,l]$, compute the corresponding feature vector $\mathbf{F}[k,l]$, as in (12).
3) Perform some clustering algorithm (e.g. K-means) to find the n cluster centers in the feature space. The cluster centers will yield preliminary estimates $\bar{a}_{ij}$ and $\bar{\delta}_{ij}$ of the mixing parameters a_{ij} and δ_{ij}, respectively, via (13).
4) Normalize the attenuation coefficients to obtain the *final attenuation parameter estimates* $\tilde{a}_{ij}$, i.e.

$$\tilde{a}_{ij} := \bar{a}_{ij}/(\sum_{i=1}^{m}\bar{a}_{ij}^2)^{1/2}.$$

The *final delay parameter estimates* are $\tilde{\delta}_{ij} := \bar{\delta}_{ij}$.

Note that the algorithm proposed above simply extends clustering based approaches for the estimation of the mixing parameters to accommodate the $m\times n$ anechoic mixing model.

C. Method Assumptions and Limitations

The parameter estimation algorithm described above will yield a meaningful estimate of the mixing parameters only if certain assumptions hold.

Assumption 1: Due to the periodicity of the complex exponential and to avoid phase indeterminacy, we assume that

$$|\omega\delta_{ij}| < \pi \tag{14}$$

for all i,j and every ω. This is equivalent to assuming that

$$|\delta_{max}| < \pi/\omega_{max} \tag{15}$$

where δ_{max} is the largest delay in the system and ω_{max}, is the maximum frequency component present in the sources. If $\omega_{max} = \omega_s/2$, where ω_s is the sampling frequency, then the algorithm will yield accurate estimates of the delay parameters δ_{ij} as long as each of these delays is not larger than *one sample*. This entails that the spacing between any two microphones should be limited to $d < 2\pi c/\omega_s$, where c is the speed of sound, see [1]. Note that one does not need to know the actual spacing between the microphones- only that it is within the bound.

Assumption 2: We assume that all the attenuation parameters a_{ij} are positive. This is again due to the problem of phase indeterminacy. More precisely, the equality

$$ae^{-i\omega\delta} = -ae^{-i(\omega\delta+(2k+1)\pi)} = -ae^{-i\omega(\delta+(2k+1)\pi/\omega)} \quad (16)$$

for every $k \in \mathbb{Z}$ leads to *two* possible attenuation coefficients (and infinitely many delay parameters corresponding to each attenuation coefficient) for every entry in the feature vectors given by (13). To avoid this problem, we assume that delays are limited to one sample at most, i.e., *Assumption 1 holds*, and that attenuation parameters are positive. Note that Assumption 2 holds for anechoic audio mixtures.

Assumption 3: We assume that the number of sources n is known prior to running the clustering algorithm. In practice, this is rarely the case. However, our experiments indicate that the proposed algorithm is robust with respect to changes in the number of assumed sources, particularly if the number of sources is *overestimated*. Figure 3 shows a three dimensional view of the feature vectors obtained using the correct (real) parameter values as well as those obtained using the extracted cluster centers by applying the described parameter estimation algorithm on $m = 3$ simulated mixtures of $n = 5$ sources. In this example, the "user" overestimated the number of sources and the algorithm thus extracted $\hat{n} = 6$ sources. However, it is clear that five of the extracted cluster centers can be used to estimate the correct mixing parameters. The sixth center, on the other hand, produces a "bogus" column in the mixing matrix. Because of the sparsity of the STFT expansions of speech signals, according to our experiments (see section VI), this does not seem to cause any serious problems with demixing. For an extensive study of the effect of "overestimating" the number of sources on the separation performance see [14].

Assumption 4: We assume that

(4.1) there is an abundance of T-F points at which only one source is active, i.e., only one source has a large coefficient, and

(4.2) at any T-F point, no more than m (the number of mixtures) sources are active.

A similar, yet stronger, assumption was introduced and thoroughly investigated in [1]. The so-called W-disjoint orthogonality of [1] is equivalent to the (4.1) and a stronger version of (4.2) obtained by replacing m above with 1. In [1], the authors provide mathematical measures that can be used to quantify the extent to which the W-disjoint orthogonality assumption is satisfied by speech signals, and present results from experiments conducted on a large number of mixtures. They conclude, for example, that speech signals are 96.3% "disjoint" in mixtures of 2 sources, and 64% "disjoint" in mixtures of 10 sources. These observations in [1] show that our assumptions (4.1) and (4.2) are satisfied for speech signals to an even greater extent as (4.1) and (4.2) are weaker versions of the W-disjoint orthogonality assumption.

Note that in Section IV-D, we investigate the same BSS problem in a probabilistic setting. In this case, we assume that at each T-F point, the magnitudes of the STFT of speech signals are *independent*, identically distributed (i.i.d.), with a distribution that is *concentrated at the origin*. Roughly speaking, such an assumption is the probabilistic version of (4.1) and (4.2) in that it ensures that (4.1) and (4.2) are satisfied with high probability.

At this point we note that there are two different "sparsity" notions that are of concern:

1) Sparsity of a particular source signal in the transform domain.
2) The number of sources simultaneously active at a given T-F point.

Both in the deterministic and the probabilistic settings, we use the first notion only as a means to arrive at the second; the methods proposed in this paper are all arrived at via the second sparsity notion. In the deterministic setting, sparsity of each individual source is used only heuristically to explain why assumptions (4.1) and (4.2) are observed to hold. In the probabilistic setting, discussed in IV-D, both notions are interconnected under the assumption that the sources are i.i.d. in the T-F domain.

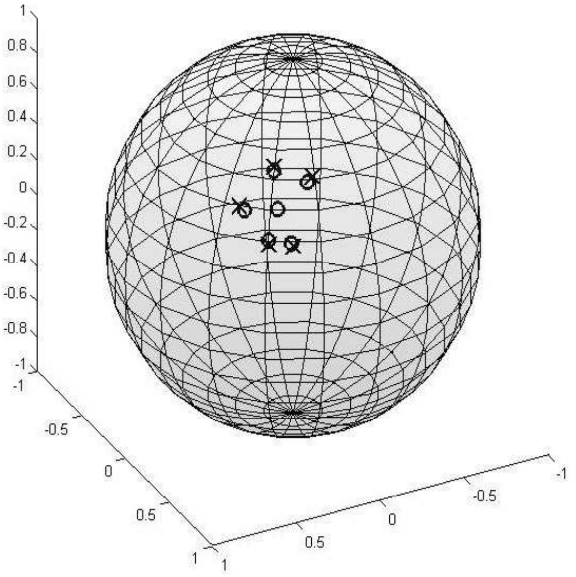


Fig. 3. 3-D view of real (crosses) and estimated (circles) parameters as recovered from the K-means clustering stage. The algorithm was run on 3 simulated mixtures of 5 sources (m=3, n=5), with the user solving for 6 sources. Note the proximity of the real to the estimated parameters. Also note the estimated source parameter that does not correspond to any real source. Displayed are the 3-dimensional normalized attenuation parameters. The delay parameters have not been included.

IV. Source Extraction

This section describes the method proposed for extracting the original sources based on the estimated parameters obtained as in the previous section.

First we construct the estimated mixing matrix $\tilde{A}[l]$ as

$$\tilde{A}[l] = \begin{bmatrix} \tilde{a}_{11}e^{-il\omega_0\tilde{\delta}_{11}} & \dots & \tilde{a}_{1n}e^{-il\omega_0\tilde{\delta}_{1n}} \\ \tilde{a}_{21}e^{-il\omega_0\tilde{\delta}_{21}} & \dots & \tilde{a}_{2n}e^{-il\omega_0\tilde{\delta}_{2n}} \\ \vdots & \vdots & \vdots \\ \tilde{a}_{m1}e^{-il\omega_0\tilde{\delta}_{m1}} & \dots & \tilde{a}_{mn}e^{-il\omega_0\tilde{\delta}_{mn}} \end{bmatrix}. \tag{17}$$

Here, $\tilde{a}_{ij}$ are the estimated attenuation parameters and $\tilde{\delta}_{ij}$ are the estimated delay parameters, computed as discussed earlier. Note that each column of $\tilde{A}[l]$ is a unit vector in $\mathbb{C}^m$.

The goal now is to compute "good" estimates $s_1^e, s_2^e, ..., s_n^e$ of the original sources $s_1, s_2, ..., s_n$. These estimates must satisfy

$$\tilde{A}[l]\hat{\mathbf{s}}^{\mathbf{e}}[k,l] = \hat{\mathbf{x}}[k,l], \tag{18}$$

where $\hat{\mathbf{s}}^e = [\hat{s}_1^e, ...\hat{s}_n^e]^T$ is the vector of source estimates in the T-F domain. At each T-F point $[k,l]$, (18) provides m equations (corresponding to the m available mixtures) with $n > m$ unknowns $(\hat{s}_1^e, ...\hat{s}_n^e)$. Assuming that this system of equations is consistent, it has infinitely many solutions. To choose a reasonable estimate among these solutions, we shall exploit the sparsity of the source vector in the T-F domain, in the sense of assumptions (4.1) and (4.2).

A. Sparsity and ℓ^q Minimization

At this stage, we wish to find the "sparsest" $\hat{\mathbf{s}}^{\mathbf{e}}$ that solves (18) at each T-F point. This problem can be formally stated as

$$\min_{\hat{\mathbf{s}}^e} \|\hat{\mathbf{s}}^e\|_{\text{sparse}} \quad \text{subject to} \quad \tilde{A}\hat{\mathbf{s}}^{\mathbf{e}} = \hat{\mathbf{x}}, \tag{19}$$

where $\|\mathbf{u}\|_{\text{sparse}}$ denotes some measure of sparsity of a vector $\mathbf{u}$.

Given a vector $\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{R}^n$, one measure of its sparsity is simply the number of the non-zero components of $\mathbf{u}$, commonly denoted by $\|\mathbf{u}\|_0$. Replacing $\|\mathbf{u}\|_{\text{sparse}}$ in (19) with $\|\mathbf{u}\|_0$, one gets the so-called P_0 problem, e.g. [30]. Solving P_0 is, in general, combinatorial with the solution being very sensitive to noise. More importantly, the sparsity of the Gabor coefficients of speech signals essentially suggests that most of the coefficients are very small, though not identically zero. In this case, P_0 fails miserably. Alternatively, one can consider

$$\|\mathbf{u}\|_q := (\sum_i |u_i|^q)^{1/q}$$

where $0 < q \le 1$ as a measure of sparsity. Here, smaller q signifies increased importance of the sparsity of $\mathbf{u}$, e.g. [31].

Motivated by this, we propose to compute the vector of source estimates $\hat{\mathbf{s}}^e$ by solving the P_q problem at each T-F point $[k,l]$, *if the mixing matrix is real*. The P_q problem is defined by replacing $\|\mathbf{u}\|_{\text{sparse}}$ in (19) with $\|\mathbf{u}\|_q$ to obtain the following optimization problem

$$P_q : \quad \min_{\hat{\mathbf{s}}^e} \|\hat{\mathbf{s}}^e\|_{\text{q}} \quad \text{subject to} \quad \tilde{A}\hat{\mathbf{s}}^{\mathbf{e}} = \hat{\mathbf{x}}. \tag{20}$$

Note that this approach, with $q = 1$, was proposed before, e.g., [12], [13], [14]. It is a standard result that if $\tilde{A}$ is real, P_1 is equivalent to ℓ^1-basis-pursuit (L1BP), given by

$$\text{L1BP:} \quad \min_{\hat{\mathbf{s}}^e} \|\hat{\mathbf{s}}^e\|_1 \quad \text{subject to} \quad \tilde{A}\hat{\mathbf{s}}^{\mathbf{e}} = \hat{\mathbf{x}} \quad \text{and} \quad \|\hat{\mathbf{s}}^e\|_0 \le m. \tag{21}$$

In Theorem 1, below, we prove that such an equivalence holds for any $0 < q < 1$ as well, provided $\tilde{A}$ and $\mathbf{x}$ are real. More precisely, in this case the solution of P_q is identical to the solution of the ℓ^q-basis-pursuit (LQBP) problem, given by

$$\text{LQBP:} \quad \min_{\hat{\mathbf{s}}^e} \|\hat{\mathbf{s}}^e\|_{\text{q}} \quad \text{subject to} \quad \tilde{A}\hat{\mathbf{s}}^{\mathbf{e}} = \hat{\mathbf{x}} \quad \text{and} \quad \|\hat{\mathbf{s}}^e\|_0 \le m. \tag{22}$$

Note that to solve the LQBP problem, one needs to find the "best" basis for the column space of $\tilde{A}$ that minimizes the ℓ^q norm of the solution vector.

In the next section, we shall investigate solution strategies for P_q and LQBP, and discuss how to handle the case when the matrix $\tilde{A}$ is complex.

B. Solving P_q and ℓ^q-basis-pursuit

The optimization problem P_q is not convex for $0 < q < 1$, thus computationally challenging. Under certain conditions on the mixing matrix A and on the sparsity of $\mathbf{x}$, it can be shown that a near minimizer can be obtained by solving the convex P_1 problem [30], [32], [33], which if A and $\mathbf{x}$ are real, can be reformulated as a linear program. This is, in fact, one of the main motivations of the ℓ^1-based approaches in the literature. On the other hand, we do not want to impose any a priori conditions on the mixing matrix A (consequently on the estimated mixing matrix $\tilde{A}$). In fact, the experimental results presented in Section VI indicate that the mixing matrices that correspond to anechoic mixing scenarios do not satisfy these a priori conditions, and therefore, we cannot approximate the solution of P_q by the solution of P_1. Without such conditions, only local optimization algorithms for solving P_q are available in the literature, e.g., [33]. Below, we prove that the P_q problem with $0 < q < 1$ can be solved in combinatorial time whenever the mixing matrix A is real.

Theorem 1: Let $A = [\mathbf{a}_1|\mathbf{a}_2|\dots|\mathbf{a}_n]$ be an $m \times n$ matrix with $n > m$, $A_{ij} \in \mathbb{R}$, and suppose that A is full rank. For $0 < q < 1$, the P_q problem

$$\min_{\mathbf{s}} \|\mathbf{s}\|_q \quad \text{subject to} \quad A\mathbf{s} = \mathbf{x}$$

where $\mathbf{x} \in \mathbb{R}^n$, has a solution $\mathbf{s}^* = (s_1^*, ...s_n^*)$ which has $k \le m$ non-zero components. Moreover, if the non-zero components of $\mathbf{s}^*$ are $s_{i(j)}^*$, $j = 1, \dots, k$, then the corresponding column vectors $\{\mathbf{a}_{i(j)} : \ j = 1, \dots, k\}$ of A are linearly independent.

The proof of this theorem is provided in the Appendix.

Given an $m \times n$ real mixing matrix A with $m < n$, Theorem 1 shows that the solution of the corresponding P_q problem will have at most m non-zero entries, and therefore will automatically satisfy the additional constraint of LQBP (compare (20) and (22)). Thus, if the matrix A is real, the solution of LQBP and the solution of the P_q problem are identical. As such, by solving LQBP, i.e., by finding all the subsets of the set of columns of A that form a basis and choosing the one that offers a solution with the minimum ℓ^q-norm, we can solve the P_q problem. In other words, P_q is computationally tractable whenever the mixing matrix A and $\mathbf{x}$ are real, and can be solved via the following straight-forward combinatorial LQBP algorithm.

LQBP Algorithm: Let $\mathcal{A}$ be the set of all $m \times m$ invertible sub-matrices of A ($\mathcal{A}$ is non-empty as A is full rank). The solution of ℓ^q-basis-pursuit (and thus, by Theorem 1, the solution of P_q in the real valued case) is given by the solution of

$$\min \|B^{-1}\mathbf{x}_B\|_q \quad \text{where} \quad B \in \mathcal{A}. \tag{23}$$

Here, for $B = [\mathbf{a}_{i(1)}|\cdots|\mathbf{a}_{i(m)}]$, $\mathbf{x}_B := [x_{i(1)} \cdots x_{i(m)}]$. Note that $\#\mathcal{A} \leq \binom{n}{m}$, thus (23) is a combinatorial problem.

Theorem 1 does not hold when the matrix A is complex-valued; a counter example and discussion can be found in [34]. Note, however, that the goal of finding the solution with the smallest ℓ^q (quasi-) norm is to impose sparsity. Thus if the statement of the above theorem does not hold, i.e., the l^0 "norm" of the minimizer of P_q is larger than the rank of A, then one would not, in fact, wish to use that solution. For this reason, in the case of anechoic mixtures, thus complex A, we propose to extract the sources using the *ℓ^q-basis-pursuit* approach, i.e., by finding the best basis composed by a subset of columns of A that minimizes the ℓ^q norm of the solution vector. Theorem 1 shows that this is equivalent to solving the P_q problem in the real-valued case.

C. The Separation Algorithm

Based on the discussion above, the proposed **separation algorithm** can be summarized as follows. At each T-F point $[k, l]$:

1) Construct the estimated mixing matrix $\tilde{A}[l]$ as in (17).
2) Find the estimated source vector $\hat{\mathbf{s}}^e[k, l]$ by solving the ℓ^q-basis-pursuit problem with $A = \tilde{A}[l]$ as described above for some $0 < q < 1$ (as demonstrated in Section VI, a choice of $0.1 \leq q \leq 0.4$ is appropriate).
3) After repeating steps 1 and 2 for all T-F points, reconstruct $\mathbf{s}^{\mathbf{e}}(t)$, the time domain estimate of the sources from the estimated Gabor coefficients.

Remark. In the literature the main focus has been to use P_1 or ℓ^1-basis-pursuit for solving the source extraction problem, e.g., [12], [13], [14], [20]. The main motivation for this as discussed above, is that ℓ_1-basis-pursuit can be formulated as a convex program, and thus is preferable from a computational point of view. Therefore, the attempt to consider ℓ_q-basis-pursuit with $0 < q < 1$ might sound counter-intuitive at first. However, in the case of BSS of speech signals, the size of each ℓ_q-basis-pursuit problem to be solved is quite small (A is an $m \times n$ matrix where m is the number of microphones and n is the number speakers). Thus the combinatorial algorithm proposed above is in fact of comparable complexity with a convex program. A similar observation was also made in [34]. See section V for a more detailed discussion.

D. Probabilistic Interpretation

This section provides an interpretation of the presented source separation algorithm from a Bayesian point of view by generalizing the approach of Delgado et al. [22] to the complex-valued case. Recall that at a given T-F point the algorithm attempts to extract n sources from m mixtures with $m < n$ using an estimate $\tilde{A}$ of the mixing matrix A. In other words, one needs to find T-F estimates of the sources so that (18) is satisfied. Since the system of equations defined by (18) is underdetermined, it has an infinite number of solutions. If we now assume that the STFT coefficient magnitudes of the sources at all T-F points are i.i.d. random variables and all the coefficient phases are i.i.d random variables that are independent from the magnitudes, we can adopt the Bayesian approach and choose the solution that is given by the corresponding maximum a posteriori estimator. That is, at each T-F point $[k, l]$ the extracted source vector $\hat{\mathbf{s}}^{\mathbf{e}}[k, l]$ must satisfy

$$\begin{aligned}
\hat{\mathbf{s}}^{\mathbf{e}}[k,l] &= \arg\max_{\hat{\mathbf{s}}^{\mathbf{e}}[k,l]} P(\hat{\mathbf{s}}^{\mathbf{e}}[k,l]|\tilde{A}, \hat{\mathbf{x}}[k,l]) \\
&= \arg\max_{\hat{\mathbf{s}}^{\mathbf{e}}[k,l]} P(\hat{\mathbf{x}}[k,l]|\tilde{A}, \hat{\mathbf{s}}^{\mathbf{e}}[k,l]) P(\hat{\mathbf{s}}^{\mathbf{e}}[k,l]) \\
&= \arg\max_{\hat{\mathbf{s}}^{\mathbf{e}}[k,l]} P(\hat{\mathbf{s}}^{\mathbf{e}}[k,l]) \qquad (24) \\
&= \arg\max_{\hat{\mathbf{s}}^{\mathbf{e}}[k,l]} P(|\hat{\mathbf{s}}^{\mathbf{e}}[k,l]|, \angle\hat{\mathbf{s}}^{\mathbf{e}}[k,l]) \qquad (25) \\
&= \arg\max_{\hat{\mathbf{s}}^{\mathbf{e}}[k,l]} P(|\hat{\mathbf{s}}^{\mathbf{e}}[k,l]|) P(\angle\hat{\mathbf{s}}^{\mathbf{e}}[k,l]) \qquad (26)
\end{aligned}$$

Note that in the third equality we use the fact that $\tilde{A}[l]\hat{\mathbf{s}}^{\mathbf{e}}[k, l] = \hat{\mathbf{x}}[k, l]$. Now, assuming that

$$P(|\hat{s_i}^e[k,l]|) = \frac{\mu^{1/p} p e^{-\mu|\hat{s_i}^e[k,l]|^p}}{\Gamma\left(p^{-1}\right)} \tag{27}$$

i.e., that the magnitudes of the sources are independent and identically distributed following a Box-Tiao distribution [35] (equivalently, a generalized Gaussian distribution) for some $\mu > 0$ and $p < 1$, and that

$$P(\angle\hat{s_i}^e[k,l]) = \frac{1}{2\pi}, \tag{28}$$

i.e., that the phases are uniformly distributed, we obtain

$$\begin{aligned}
\hat{\mathbf{s}}^{\mathbf{e}}[k,l] &= \arg\max_{\hat{\mathbf{s}}^{\mathbf{e}}[k,l]} e^{-\sum_{i=1}^{n} |\hat{s}^e[k,l]|^p} \\
&= \arg\min_{\hat{\mathbf{s}}^{\mathbf{e}}[k,l]} \sum_{i=1}^{n} |\hat{s}_i^e[k,l]|^p.
\end{aligned}$$

Reintroducing the constraint set by (18), the problem then becomes

$$\min_{\hat{\mathbf{s}}^{\mathbf{e}}[k,l]} \|\hat{\mathbf{s}}^{\mathbf{e}}[k,l]\|_p, \text{ subject to } \tilde{A}\hat{\mathbf{s}}^{\mathbf{e}}[k,l] = \hat{\mathbf{x}}[k,l],$$

which is identical to the P_q problem defined as in (20) with $q = p$. In other words, by solving P_q of Section IV-A with $q \in (0, 1]$, we intrinsically compute the maximum a posteriori (MAP) estimate if the magnitudes of the sources in the T-F domain were distributed according to the Box-Tiao distribution with $p = q$, and if the phases were uniformly distributed. Thus, one would expect to obtain best separation results using P_q (or LQBP) if the underlying sources are in fact Box-Tiao distributed with parameter $p = q$. Although we do not claim that this family of distributions provide the best model for the STFT-magnitudes of speech signals, we expect that, among the family of algorithms given by P_q (or ℓ^q-basis-pursuit) with $q \in (0, 1]$, the best separation will be observed for the value of q

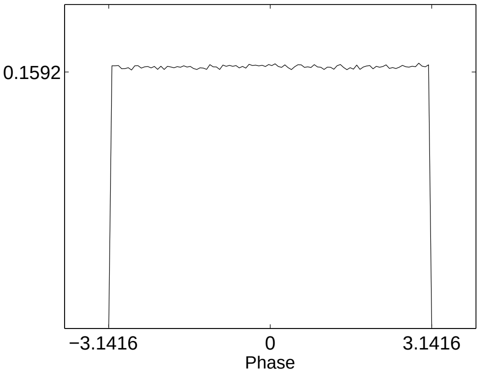


Fig. 4. The empirically calculated probability density function of the phase of speech STFT coefficients. One can see that the phases are uniformly distributed between $-\pi$ and π.

that optimizes the fit between the empirical distribution of the magnitudes of the STFT of speech signals and the distributions given by (27)

We computed the STFTs of 300 three-second-long speech signals from the TIMIT database (sampled at 16kHz) using a window length of 1024 and an overlap of 50%. We calculated the empirical probability density of the phases, plotted in Figure 4, which clearly shows the validity of the assumption of a uniform distribution. We then used the Nelder-Mead simplex search algorithm [36] to find the maximum likelihood estimate of the value of p, for the magnitudes in each case. This yields estimates for p with *sample mean* 0.2737 and *sample standard deviation* 0.0313. Note that, as expected, this matches with the results presented in Section VI, where best separation performance is obtained with $0.1 \leq q \leq 0.4$.

The fact that we obtain a small sample standard deviation further indicates that the value of q that provides the best fit is appropriate for speech signals in the STFT domain, with a window size of approximately $64ms$. Note that there is no guarantee that such a choice would be optimal for each individual speech signal. If for example, we had prior knowledge that the signals we are dealing with are not very sparse, then a larger value of q would be justified. On the other hand, since we are dealing with *blind* source separation, our goal is to work with a fixed value q that is suited to the signals at hand. The above discussion suggests that $q \approx 0.27$ is a good choice. Finally, we note that the question of how to incorporate additional information, such as some sources being sparser than others, remains an interesting open problem.

E. Interference Suppression and Distortion Reduction

The algorithm proposed in Section IV-C separates n sources from m mixtures. The task is accomplished by extracting at most m sources at each T-F point that minimize via ℓ^q-basis-pursuit, as discussed above. The following assumptions are required to ensure an accurate recovery of the sources:

i. No more than m sources are active at that T-F point.
ii. The columns of the mixing matrix were accurately extracted in the mixing model recovery stage.
iii. The mixing matrix is full rank.
iv. The noise affecting the mixtures is negligible.

If these assumptions hold, then the decomposition of the mixtures into their source contributions will be successful. We shall not address here the problem of having more than m active sources at certain T-F points as this would violate our basic sparsity assumption and render the use of ℓ^q-basis-pursuit inappropriate. A more important issue is that of the mixing model recovery stage not yielding the perfect columns in the mixing matrix, as this would negatively affect the source estimates. Under the sparsity assumption, it is also very likely for the number of active sources to be less than m at many T-F points. In that case, errors in the estimation of the mixing directions, and possible existence of noise might lead to false assignments of some contributions to sources that are in fact silent. These contributions, which the algorithm would record as source activity, could be due to projections of contributions from other sources or due to noise.

To avert these problems, we introduce a power ratio parameter ρ, where $0 < \rho < 1$, which the user may adjust based on the noise level or expected difficulty of separation. Accordingly, after resolving the contribution of each source via ℓ^q-basis-pursuit, we inspect each source's contribution to the total power (of all sources at that T-F point). We then preserve the r highest sources where $1 < r < m$, which contribute, collectively, to at least $100\rho\%$ of the total power and set the rest to zero. The motivation behind this is that if a source is inactive, noise will still project on the source's direction giving a contribution, albeit a small one, hence the need to introduce the parameter ρ to get rid of these unwanted small contributions. To summarize:

Interference Suppression Algorithm: At each T-F point $[k, l]$:

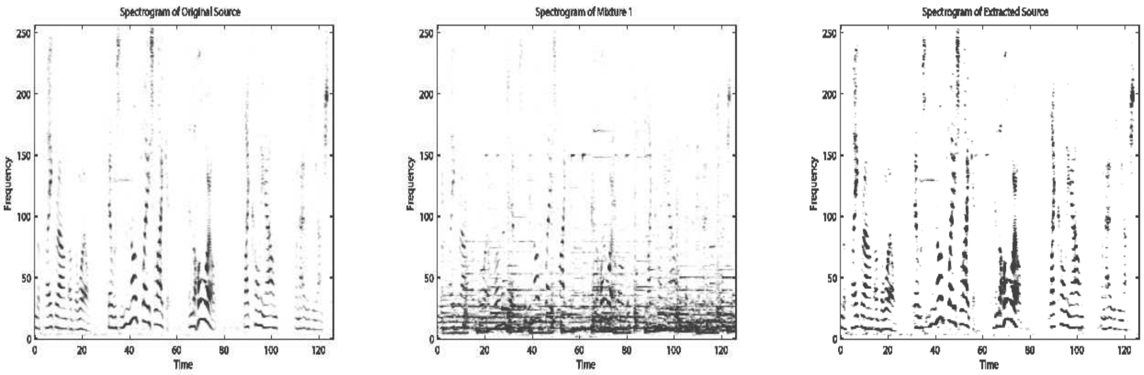
1) Sort the source coefficient estimates in decreasing order.
2) Preserve the first (highest) k sources that contribute to at least $100\rho\%$ of the total power.
3) Set the remaining estimates to zero.

See Section VI for implementation of this algorithm with various values of ρ.

V. Computational Complexity

In order to get an idea about the computational complexity of the proposed ℓ^q-basis-pursuit algorithm, we conducted a series of tests comparing the proposed technique to second order cone programming (SOCP) [37], an interior point method used to solve l_1 minimization problems in the complex domain. We utilized the SeDuMi package [38] as a numerical solver for the SOCP problem. Note that a SOCP approach can only be utilized for the P_1 problem and not for the general P_q problem with $q < 1$.

The experiments involved varying the number of mixtures m from 2 to 5 and the number of sources n from m to 15. The results reported in Figure 7 clearly indicate that given a number of sources ($n < 12$) and a relatively small number of mixtures ($m \leq 5$), LQBP outperforms SOCP in terms of computational speed. A similar conclusion was previously reported in [34] where a combinatorial approach was proposed for the P_1 problem in the complex domain. One could see from the figure that the computational complexity of SOCP is high

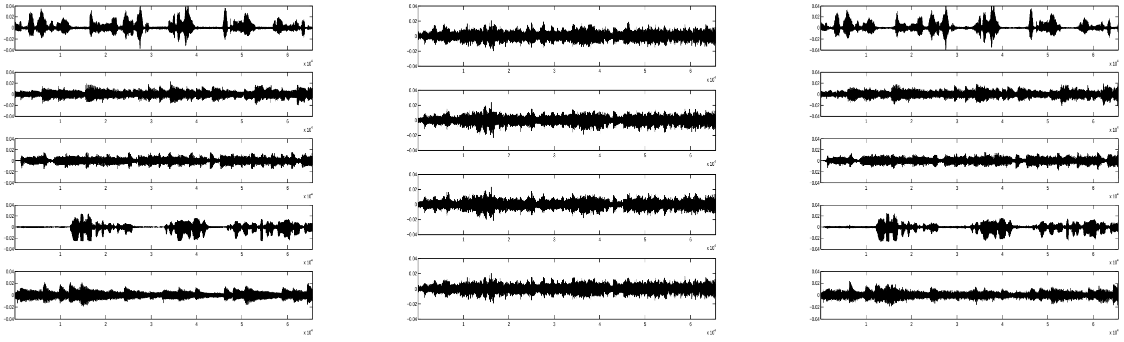


(a) Original Source Spectrogram

(b) Mixture 1 Spectrogram

(c) Extracted Source Spectrogram

Fig. 5. The spectrogram of (a) one of the original sources, (b) one of the mixtures, and (c) the corresponding extracted sources from 4 mixtures of 5 underlying sources when the user estimates the existence of 6 sources.



(a) Original Sources

(b) Mixtures

(c) Extracted Sources

Fig. 6. The (a) original sources, (b)mixtures, and (c) extracted sources from 4 mixtures of 5 underlying sources when the user estimates the existence of 6 sources.

initially but grows very slowly with the number of sources and mixtures. On the other hand, LQBP has a much lower complexity for a small number of mixtures and sources, but that complexity tends to grow quickly as m and n increase. Thus, for the range of m and n that we are dealing with in this paper, the combinatorial approach is computationally tractable and more favorable than SOCP.

Another advantage of the combinatorial approach as indicated by [34] is in the reusability of results. In other words, given a certain frequency, all the matrix inversions need only be done once and the results can be stored and used as needed. On the other hand, the SOCP algorithm needs to be rerun for every single T-F point. This observation was not used to generate the results reported in Figure 7 where the matrix inversions were repeated for LQBP.

VI. Experiments and Results

The performance of the proposed algorithm is evaluated in this section using experiments with both simulated and real mixtures. To assess the quality of the separation, the performance measures suggested in [39] are used, namely the Source to Artifact Ratio (SAR), the Source to Interference Ratio (SIR) and the Source to Distortion Ratio (SDR). SAR measures the distortion due to algorithmic or numerical artifacts such as ”forced zeros” in the STFT. SIR measures the interference due to sources other than the one being extracted and that have residual components in the extracted source. SDR, on the other hand, is a measure of all types of distortion, whether artifacts, interference or noise. In [39] it was observed that informal listening tests correlated well with the nature of the perceived distortion as quantified by the SIR and SAR measures. Our own informal listening tests confirm this observation.

In order to thoroughly test the proposed methods we conducted experiments under a variety of conditions and we report the results here. First, we highlight the importance of using all the available mixtures by demixing 5 sources while decreasing the number of mixtures used from 5 to 2. Next, we test the algorithm in a difficult scenario where we have 10 sources and 5 mixtures and show that it performs favorably. We then present average results of a large number of experiments conducted using a model of an anechoic room and compare our results, obtained for various values of q, where $q < 1$, with those of DUET, both in cases where we have two mixtures, as well as in cases when more than two mixtures are available. Finally, we present the results of our algorithm when applied to a real world echoic mixing scenario with 2 mixtures and

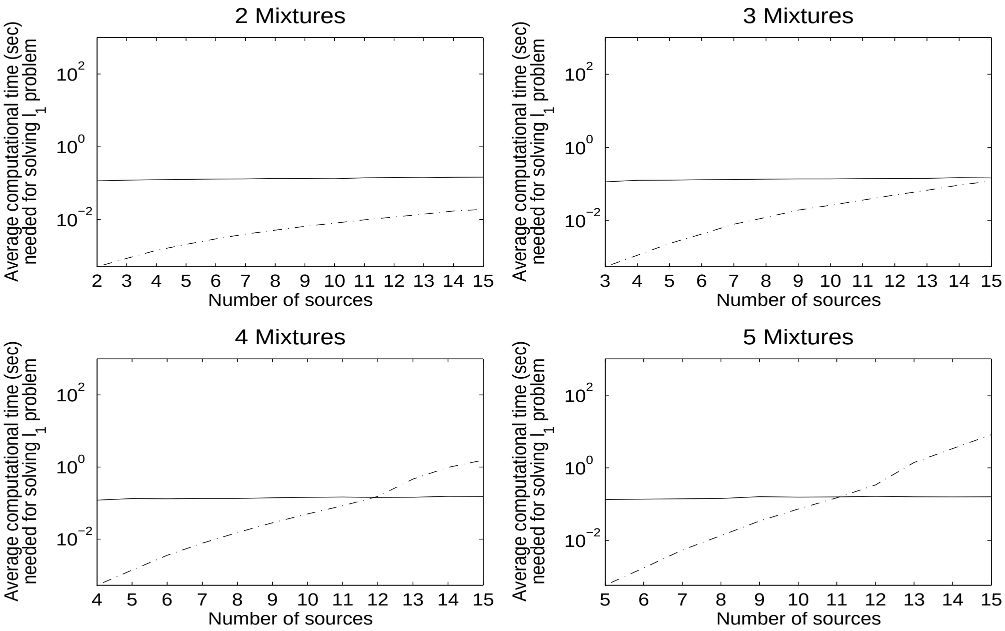


Fig. 7. Average time (log scale) taken to solve LQBP (dashed line) and P_1 via SOCP (solid line) as a function of the number of sources, with varying number of available mixtures.

show that it performs well here also.

A. Simulated Mixtures With Random Mixing Parameters

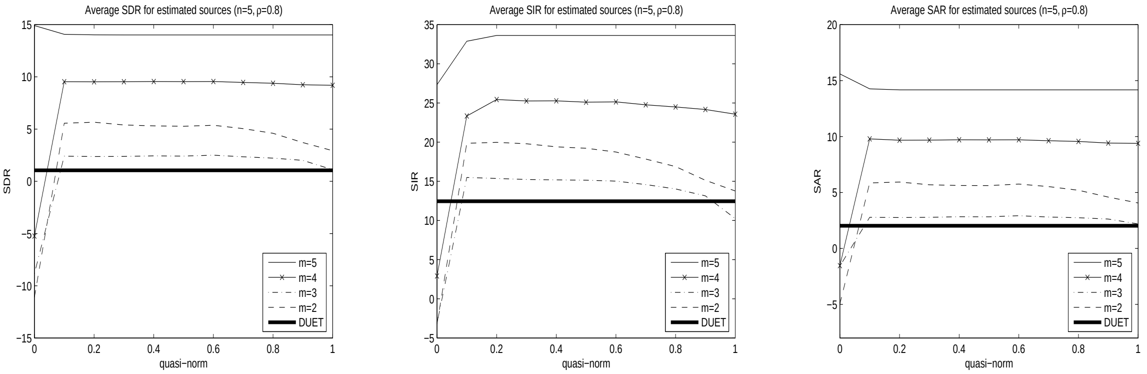
The algorithm is first tested on simulated mixtures of 5 sources, 2 of which are speech and 3 are music. The 5 mixtures are generated as delayed and attenuated versions of the sources with the mixing parameters as shown in Table II

TABLE II
RANDOM MIXING PARAMETERS USED TO SIMULATE MIXTURES WITH 5 SOURCES.

	s_1	s_2	s_3	s_4	s_5
a_{1i}	0.61	0.71	0.73	0.82	0.87
a_{2i}	0.94	0.65	0.83	0.99	0.72
a_{3i}	0.85	0.76	0.72	0.93	0.60
a_{4i}	0.80	0.76	0.56	0.64	0.92
a_{5i}	0.68	0.79	0.71	0.67	0.65
δ_{2i}	0.01	-0.62	0.08	0.72	0.80
δ_{3i}	0.42	-0.61	-0.70	0.71	0.64
δ_{4i}	-0.14	0.36	0.40	0.19	0.29
δ_{5i}	-0.39	-0.39	-0.24	-0.01	0.64

Figure 5 shows the T-F decomposition of one of the sources, one of the mixtures, and the corresponding extracted source, when using 4 mixtures for separation and setting q to 0.3. Figure 6 shows all the original sources, mixtures, and extracted sources when using 4 mixtures to perform the separation. Tables III, IV and V show the demixing performance based on the SIR, SAR, and SDR, respectively. All results were obtained by running the algorithm with the user parameters $\hat{n} = 6$ and $\rho = 0.8$ and by setting $q = 0.3$. Each column in these tables indicates the performance before demixing as well as when using either 5, 4, 3, or 2 of the available mixtures for demixing; the last column reports the results obtained when using DUET. Table III shows a mean gain in SIR of 19dB when demixing 5 sources from just 3 mixtures. The mean gain reaches over 32dB when utilizing all five available mixtures, which highlights the importance of using the available mixtures. Similarly, Table V which reports the SDRs of the extracted sources shows a gain ranging from 2dB to over 14dB when using 2 to 5 mixtures respectively. Interestingly, the SAR degrades upon separation when using less than 5 mixtures. This may be attributed to the fact that our algorithm is non-linear in nature, and acts on the STFT transform of the mixtures, extracting the sources in that domain. This may introduce numerical artifacts, such as forced zeros or non-smooth transitions in the STFT of the sources, which are then reflected by the SAR values reported. Note that when comparing the results of SDR, SIR and SAR obtained by applying the proposed algorithm on 2 mixtures only vs. the results obtained by applying DUET, the presented algorithm outperforms, on average, DUET in all criteria (last 2 columns of Tables III,IV, and V). Finally, Figure 8, shows the SDR, SIR and SAR resulting from separation using 5, 4, 3 or 2 mixtures, for values of q, $0 \leq q \leq 1$, in steps of 0.1.

To test the performance of the algorithm when the number of sources is large, we next used it on 5 mixtures of 10 sources, using random delays and attenuations with the same 5 sources used in the previous experiment augmented with 5 more speech sources. The results are shown in Figure 9. The average gains in SIR, SAR and SDR of approximately 18.8, 4.9 and 12.3 dB, respectively, confirm that the proposed algorithm performs remarkably well even in such a scenario where there is a much higher number of sources than mixtures. In fact, all but the very last extracted source are recovered



(a) SDR (b) SIR (c) SAR

Fig. 8. Average SDR, SIR and SAR (over the five sources) obtained from demixing various number of *simulated anechoic mixtures of 5 sources as a function of q with a preserved power parameter of 0.8. The horizontal line represents the results obtained using DUET. Across all results, the user estimates the existence of 6 sources.*

TABLE III
DEMIXING PERFORMANCE EXAMPLE WITH THE RANDOM MIXING PARAMETERS OF TABLE II ON MIXTURES OF 5 SOURCES, $\rho = 0.8$, $\hat{n} = 6$: SIR

Source	SIR (dB) before demixing	SIR (dB) after demixing with 5 mixtures	SIR (dB) after demixing with 4 mixtures	SIR (dB) after demixing with 3 mixtures	SIR (dB) after demixing with 2 mixtures	SIR (dB) DUET with 2 mixtures
s_1	-4.46	30.23	23.38	18.09	19.61	21.76
s_2	-4.86	34.30	21.20	15.84	14.40	12.94
s_3	-5.46	19.05	19.63	22.51	10.34	6.99
s_4	-4.06	40.90	26.25	22.28	16.15	10.33
s_5	-3.59	39.92	26.22	20.53	16.96	14.85
$mean$	-4.49	32.88	23.33	19.85	15.49	13.37

TABLE IV
DEMIXING PERFORMANCE EXAMPLE WITH THE RANDOM MIXING PARAMETERS OF TABLE II ON MIXTURES OF 5 SOURCES, $\rho = 0.8$, $\hat{n} = 6$: SAR

Source	SAR(dB) before demixing	SAR (dB) after demixing with 5 mixtures	SAR (dB) after demixing with 4 mixtures	SAR (dB) after demixing with 3 mixtures	SAR (dB) after demixing with 2 mixtures	SAR (dB) DUET with 2 mixtures
s_1	12.40	13.51	11.89	6.20	5.44	5.15
s_2	13.64	15.48	11.38	7.74	2.76	1.50
s_3	13.64	11.80	7.79	4.69	-0.62	-0.59
s_4	10.16	14.89	8.30	3.63	2.05	1.98
s_5	17.86	15.62	9.58	7.01	4.30	4.12
$mean$	13.54	14.26	9.79	5.85	2.79	2.43

successfully and the speakers' sentences could be discerned without difficulty. It is worth noting that there is an improvement even in the SAR values. A possible explanation for this observation is that, due to the high number of sources, the number of points in the T-F plane that the algorithm sets to zero is reduced thus reducing artifacts in the extracted sources.

B. Simulated Mixtures, Anechoic Room Mixing Parameters

In addition to the experiments with simulated data described in the previous section, additional experiments where the mixing parameters were derived from an anechoic room model [40] were conducted. The model simulates multi-microphone multi-source scenarios in an anechoic room. Tests for extracting 3, 4, and 5 sources from 2 or 3 mixtures were each conducted and repeated 60 times with various anechoic room mixing parameters and sources. The results were compared to those obtained using DUET and are illustrated in Figures 10, 11, and 12 as functions of q, as well as in Table VII. Note that the reported results indicate that the best performance occurs when using $0.1 \leq q \leq 0.4$, which agrees with the probabilistic interpretation and results provided in section IV-D. Note that the case of demixing 3 mixtures of 3 sources with the user estimating 3 sources, is an even-determined scenario; therefore all q values will yield the same results.

C. Real Mixtures

Next, to provide an example on real mixtures, we test the algorithm using the mixtures posted on [41], which have 2 sources and 2 microphones. The microphones are placed 35cm apart, and the sources are placed 60^o degrees to the left of the microphones and 2m on the mid-perpendicular of the microphones respectively [41], [42]. Table VIII shows that the proposed algorithm outperforms that of [42] for which the audio separation results can be found at [41].

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel blind source separation algorithm for the *underdetermined anechoic* case which is ca-

TABLE V

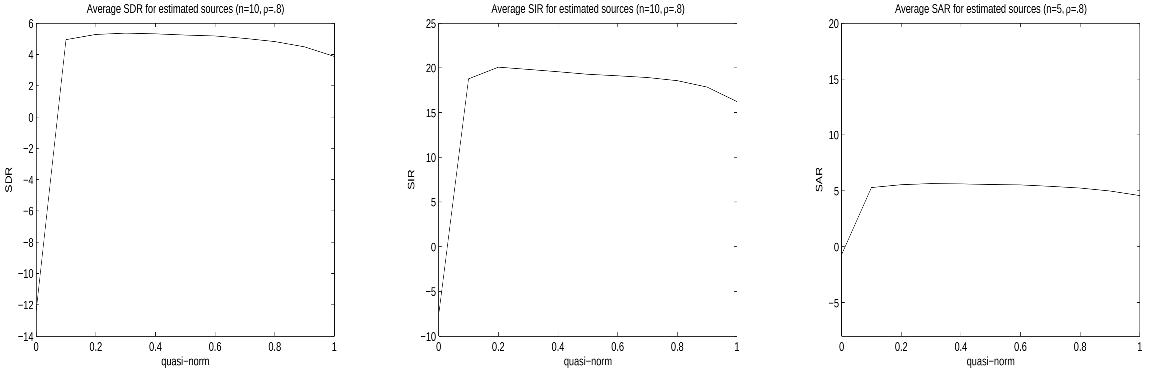
DEMIXING PERFORMANCE EXAMPLE WITH THE RANDOM MIXING PARAMETERS OF TABLE II ON MIXTURES OF 5 SOURCES, $\rho = 0.8$, $\hat{n} = 6$: SDR

Source	SDR(dB) before demixing	SDR (dB) after demixing with 5 mixtures	SDR (dB) after demixing with 4 mixtures	SDR (dB) after demixing with 3 mixtures	SDR (dB) after demixing with 2 mixtures	SDR (dB) DUET with 2 mixtures
s_1	-4.79	13.41	11.58	5.86	5.23	5.01
s_2	-5.10	15.43	10.92	7.02	2.32	0.61
s_3	-5.70	11.01	7.47	4.59	-1.31	-4.14
s_4	-4.61	14.88	8.22	3.54	1.78	0.00
s_5	-3.69	15.60	9.48	6.78	3.99	3.79
$mean$	-4.78	14.07	9.53	5.56	2.40	1.06

TABLE VI

DEMIXING PERFORMANCE EXAMPLE WITH 5 MIXTURES OF 10 SOURCES (RANDOM MIXING PARAMETERS), $\rho = 0.8$, $\hat{n} = 12$

SIR (dB)	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	**mean**
before	-7.46	-6.55	-5.34	-10.43	-5.96	-6.30	-3.78	-10.15	-9.48	-4.31	**-6.98**
after	24.12	13.04	25.65	16.34	18.68	20.24	20.47	21.94	18.98	18.80	**19.83**
gain	31.58	19.59	30.99	26.78	24.64	26.54	24.24	32.09	28.46	23.12	**26.80**
SAR (dB)	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	**mean**
before	2.63	4.08	2.63	4.08	3.27	-4.14	-0.89	-0.89	-3.16	-3.16	0.446
after	1.72	5.03	2.79	3.76	8.69	6.69	9.24	7.03	6.51	1.87	5.33
gain	-0.91	0.946	0.163	-0.327	5.42	10.8	10.1	7.92	9.67	5.03	**4.89**
SDR(dB)	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	**mean**
before	-7.46	-6.84	-5.65	-10.43	-6.77	-6.60	-4.59	-10.15	-10.40	-5.43	**-7.43**
after	7.67	4.48	0.91	3.05	6.20	3.06	9.06	9.24	5.47	4.47	**5.36**
gain	15.12	11.32	6.55	13.48	12.97	9.66	13.66	19.40	15.87	9.90	**12.79**

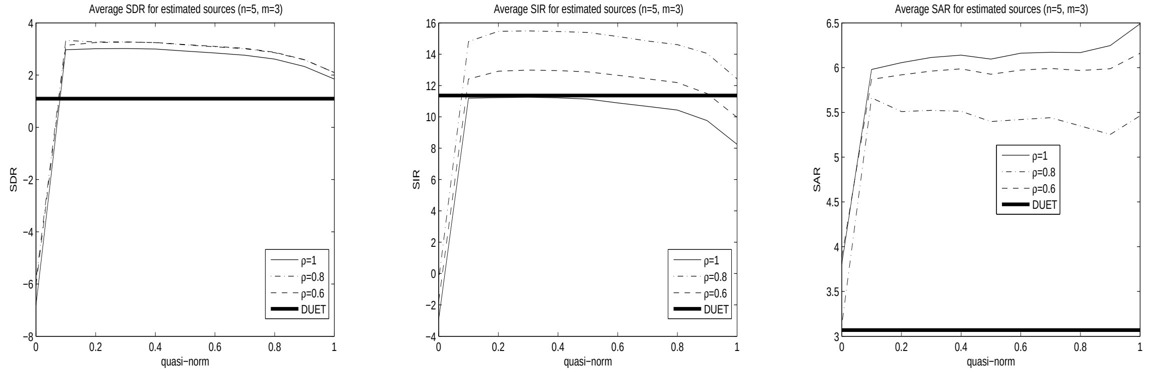


(a) SDR

(b) SIR

(c) SAR

Fig. 9. Average SDR, SIR and SAR over the 10 sources obtained from demixing 5 mixtures as a function of the q. The user estimates the existence of 12 sources.

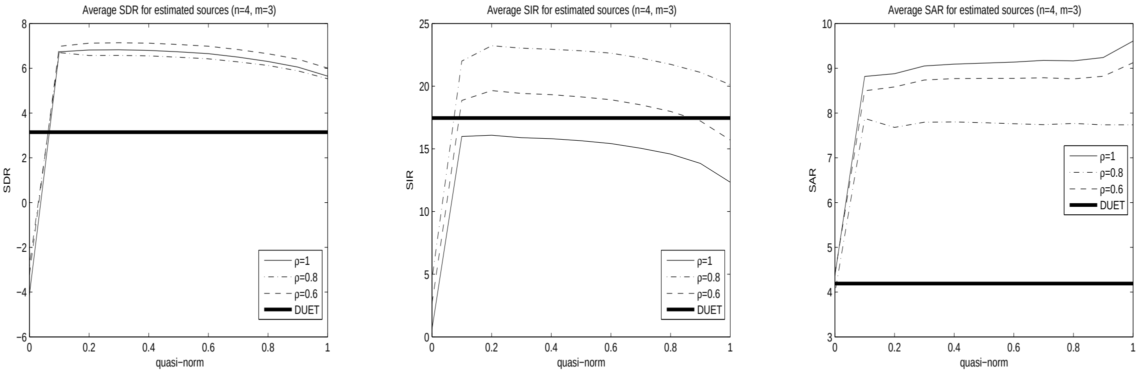


(a) SDR

(b) SIR

(c) SAR

Fig. 10. *Average* SDR, SIR and SAR (over 5 sources in 60 experiments) obtained from demixing 3 mixtures when the user estimates the existence of 6 sources. Results are plotted as a function of the q for varying preserved power parameter. The horizontal line represents the results obtained using DUET.

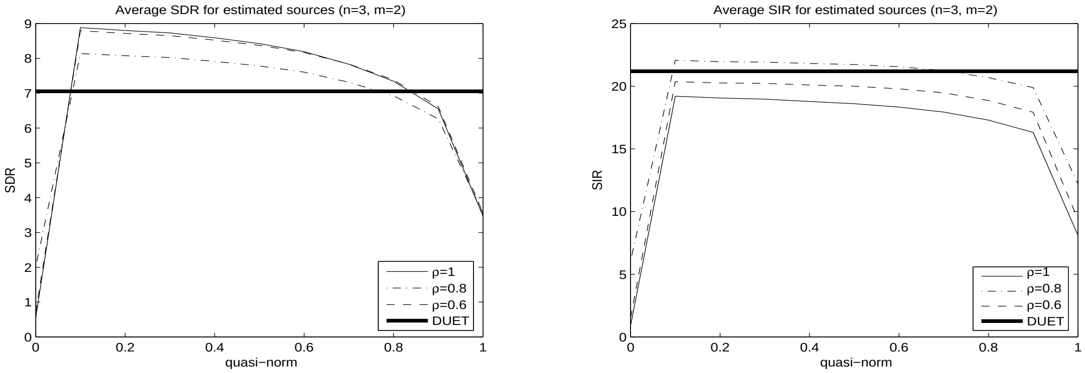


(a) SDR

(b) SIR

(c) SAR

Fig. 11. *Average* SDR, SIR and SAR (over 4 sources in 60 experiments) obtained from demixing 3 mixtures when the user estimates the existence of 5 sources. Results are plotted as a function of q for varying preserved power parameter. The horizontal line represents results obtained using DUET.



(a) SDR

(b) SIR

Fig. 12. *Average* SDR, SIR and SAR (over 3 sources in 60 experiments) obtained from demixing 2 mixtures when the user estimates the existence of 3 sources. Results are plotted as a function of q for varying preserved power parameter. The horizontal line represents the results obtained using DUET.

TABLE VII
AVERAGE DEMIXING PERFORMANCE (60 VARIOUS EXPERIMENTS) WITH 3 ANECHOIC SIMULATED MIXTURES OF 3 SOURCES, $\hat{n} = 3$

	$\rho = 1$	$\rho = 0.8$	$\rho = 0.6$	$DUET$
SIR (dB)	21.713	29.861	$\mathbf{34.495}$	19.688
SAR (dB)	$\mathbf{11.051}$	10.171	9.4898	8.4862
SDR (dB)	$\mathbf{9.6528}$	9.2686	8.7186	6.3667

pable of using *all* available mixtures in the anechoic scenario, where both attenuations as well as arrival delays between sensors are considered. The proposed technique improves the separation performance by incorporating ℓ^q-basis-pursuit with $q < 1$. In the first stage, certain feature vectors are extracted that are in turn used to extract the parameters of the mixing model via, for example, a clustering approach. This *blind mixing model recovery stage* is followed by a *blind source extraction stage*, which is based on ℓ^q-basis-pursuit, where the demixing is performed separately at every significant T-F point because the mixing matrix is frequency dependent. Further enhancement of the discussed algorithm was also proposed which was based on preservation of certain percentages of the signal power in order to reduce the effects of noise and clustering errors. We also provided a standard probabilistic interpretation of the proposed algorithm and showed that among a class of distributions parametrized by p, the distribution of the STFT of speech is best fit using $p \approx 0.27$. Solving the ℓ^q minimization problem corresponds to assuming an underlying source distribution with $p = q$, which agrees with the observation that the separation performance is best when using $0.1 \leq q \leq 0.4$.

Experimental results were presented for both simulated as well as real mixtures in anechoic underdetermined environments. The results demonstrated the robustness of the presented algorithm to *user-set parameters* and to the *lack of a priori knowledge* of the actual number of sources. The algorithm performance was measured based on the SDR, SIR and SAR. Results consistently demonstrated the method's superior performance with respect to all criteria. The use of the preserved power ratio parameter enabled the user to balance the type of distortions to incur ranging from artifacts

TABLE VIII
DEMIXING PERFORMANCE (IN dB) WITH 2 REAL MIXTURES OF 2 SOURCES, $\rho = 0.7$, $\hat{n} = 2$

	SIR [42]	SIR (our algorithm)	SAR [42]	SAR (our algorithm)	SDR [42]	SDR (our algorithm)
s_1	26.232	40.7632	4.5363	7.4011	4.4967	7.3987
s_2	55.410	43.4322	5.6433	10.4101	5.6433	10.4077
$mean$	40.821	**42.0977**	5.0898	**8.9056**	5.0700	**8.9032**

and interference. The optimal choice of this parameter, and its relationship to the estimated number of sources used for demixing and the actual number of sources remains a topic for further research.

In this paper, we have not explicitly considered noise as a part of our mixing model. By virtue of the STFT, a denoising stage via hard thresholding in the spirit of [43] is already incorporated to our algorithm after the actual separation stage during interference suppression, as discussed in Section IV-E. On the other hand, one could include a similar thresholding stage prior to the separation, or even explicitly model for the noise when formulating the optimization problem. This latter approach would lead to a difficult optimization problem, however, that can currently be solved by using computationally expensive methods that are guaranteed to provide only local minima as solutions, cf., [33]. As a natural extension of this work, we plan to investigate the anechoic blind source separation problem in the presence of noise.

VIII. ACKNOWLEDGMENTS

The authors would like to thank Scott Rickard for the anechoic room simulation software. The authors would also like to thank the reviewers for their constructive comments and suggestions which improved the paper significantly.

APPENDIX
PROOF OF THEOREM 1

For the sake of completeness, let us first restate Theorem 1.

Theorem 1: Let $A = [\mathbf{a}_1|\mathbf{a}_2|\dots|\mathbf{a}_n]$ be an $m \times n$ matrix with $n > m$, $A_{ij} \in \mathbb{R}$, and suppose that A is full rank. For $0 < q < 1$, the P_q problem

$$\min_{\mathbf{s}} \|\mathbf{s}\|_q \quad \text{subject to} \quad A\mathbf{s} = \mathbf{x}$$

where $\mathbf{x} \in \mathbb{R}^n$, has a solution $\mathbf{s}^* = (s_1^*, ...s_n^*)$ which has $k \le m$ non-zero components. Moreover, if the non-zero components of $\mathbf{s}^*$ are $s_{i(j)}^*$, $j = 1, \dots, k$, then the corresponding column vectors $\{\mathbf{a}_{i(j)} : \ j = 1, \dots, k\}$ of A are linearly independent.

We shall use the following lemma to prove this theorem.

Lemma 1: Let $\mathbf{s} = [s_1 \dots s_n]^T \in \mathbb{R}^n$ be such that $A\mathbf{s} = \mathbf{x}$, where A and $\mathbf{x}$ are as above. Suppose the column vectors of A in $\{\mathbf{a}_j : \ j \in \text{supp } \mathbf{s}\}$ are linearly dependent. Then there exists $\mathbf{s}^*$ with the following properties:

i. $A\mathbf{s}^* = \mathbf{x}$,
ii. $\|\mathbf{s}^*\|_q \le \|\mathbf{s}\|_q$, and
iii. $\#\text{supp } \mathbf{s}^* \le \#\text{supp } \mathbf{s} - 1$,

where supp $\mathbf{s} := \{j : \ s_j \neq 0\}$, and $\#U$ denotes the cardinality of a set U.

Proof: For simplicity, set $\Lambda :=$ supp $\mathbf{s}$. Then the vectors in $\{\mathbf{a}_j : \ j \in \Lambda\}$ are linearly dependent, i.e., there exist c_j, not all zero, such that $\sum_{j \in \Lambda} c_j \mathbf{a}_j = 0$. Define now the vector $\mathbf{c}$ by setting $\mathbf{c}(j) = c_j$ if $j \in \Lambda$ and $\mathbf{c}(j) = 0$ otherwise. Note that supp $\mathbf{c} \neq \emptyset$ and supp $\mathbf{c} \subseteq \Lambda$. Then, we have

$$A\mathbf{s}_\lambda = \mathbf{x}, \quad \forall \ \lambda \in \mathbb{R},$$

where $\mathbf{s}_\lambda := \mathbf{s} + \lambda\mathbf{c}$. Next, we shall show that $\mathbf{s}^* = \mathbf{s}_{\lambda^*}$ where λ^* is the solution of

$$\min_\lambda \|s_\lambda\|_q$$

satisfies $\#\text{supp } \mathbf{s}^* \le \#\text{supp } \mathbf{s} - 1$, which will complete the proof of the lemma. To that end, consider the equivalent minimization problem

$$\min_\lambda \|s_\lambda\|_q^q.$$

We want to minimize

$$f(\lambda) := \|\mathbf{s}_\lambda\|_q^q = \sum_{j \in \Lambda} \eta(s_j + \lambda c_j),$$

where $\eta(u) := |u|^q$. Noting that $\eta''(u) < 0$ for $u \neq 0$, and $\lim_{|u| \to \infty} \eta(u) = \infty$, we observe that

i. $\lim\limits_{|\lambda| \to \infty} f(\lambda) = \infty$, and
ii. $f''(\lambda) < 0$ for $\lambda \notin \{\lambda_j : \lambda_j = -s_j/c_j, \ j \in \text{supp } \mathbf{c}\}$ (recall that $\#\text{supp } \mathbf{c} \ge 1$).

Thus, the global minimum of f must be at one of its critical points λ_j, $j \in$ supp $\mathbf{c}$, where f is not differentiable; say it is at λ_{j^*}. Then, after setting $\mathbf{s}^* = \mathbf{s}_{\lambda_{j^*}}$, we have

i. $A\mathbf{s}^* = \mathbf{x}$,
ii. $\|\mathbf{s}^*\|_q \le \|\mathbf{s}\|_q$, and
iii. supp $\mathbf{s}^* \subseteq$ supp $\mathbf{s} \setminus \{j^*\}$ which implies $\#\text{supp } \mathbf{s}^* \le \#\text{supp } \mathbf{s} - 1$.

■

Proof of Theorem 1: Let $A, \mathbf{s}$, and $\mathbf{x}$ be as in the statement of Theorem 1. As A is full rank and $n > m$, the equation $A\mathbf{s} = \mathbf{x}$ has infinitely many solutions. Suppose now that $\mathbf{s}^*$ is the solution of the P_q problem. Then, by Lemma 1, the column vectors $\{\mathbf{a}_j : \ j \in \text{supp } \mathbf{s}^*\}$ are necessarily linearly independent, and as a consequence $\#\text{supp } \mathbf{s}^* \le m$. ■

REFERENCES

[1] Ö. Yılmaz and S. Rickard, “Blind source separation of speech mixtures via time-frequency masking,” *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, July 2004.

[2] R. Saab, Ö. Yılmaz, M. J. McKeown, and R. Abugharbieh, “Underdetermined sparse blind source separation with delays,” in *Workshop on Signal Processing with Adaptative Sparse Structured Representation (SPARS05)*, November 2005.

[3] C. Jutten and J. Herault, “Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture,” *Signal Processing*, vol. 24, pp. 1–10, 1991.

[4] C. Jutten, J. Herault, P. Comon, and E.Sorouchiary, “Blind separation of sources, parts i,ii and iii,” *Signal Processing*, vol. 24, pp. 1–29, 1991.

[5] A. Bell and T. Sejnowski, “An information-maximization approach to blind separation and blind deconvolution,” *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[6] H. Attias and C. E. Schreiner, “Blind source separation and deconvolution: the dynamic component analysis algorithm,” *Neural Computation*, vol. 10, no. 6, pp. 1373–1424, 1998. [Online]. Available: citeseer.ist.psu.edu/attias98blind.html

[7] S. T. Roweis, “One microphone source separation,” in *NIPS*, 2000, pp. 793–799. [Online]. Available: citeseer.ist.psu.edu/roweis00one.html

[8] A. Hyvarinen and E. Oja, “Indpendent component anlysis: Algorithms and applications,” *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[9] M. Lewicki and T. Sejnowski, “Learning overcomplete representations,” in *Neural Computation*, 2000, pp. 12:337–365.

[10] T.-W. Lee, M. Lewicki, M. Girolami, and T. Sejnowski, “Blind source separation of more sources than mixtures using overcomplete representations,” *IEEE Signal Proc. Letters*, vol. 6, no. 4, pp. 87–90, April 1999.

[11] J. Anemuller, T. Sejnowski, and S. Makeig, “Complex independent component analysis of frequency domain electroencephalographic data,” in *Neural Networks*, 2003, pp. 16:1311–1323.

[12] P. Bofill and M. Zibulevsky, “Blind separation of more sources than mixtures using sparsity of their short-time Fourier transform,” in *International Workshop on Independent Component Analysis and Blind Signal Separation (ICA)*, Helsinki, Finland, June 19–22 2000, pp. 87–92.

[13] Y. Li, A. Cichocki, and S. Amari, “Sparse component analysis for blind source separation with less sensors than sources,” in *Proceedings of 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, Riken. Kyoto, Japan: ICA, Apr. 2003, pp. 89–94. [Online]. Available: http://www.bsp.brain.riken.jp/publications/2003/ICA03LiAmariCich.pdf

[14] Y. Li, A. Cichocki, S. Amari, S. Shishkin, J. Cao, and F. Gu, “Sparse representation and its applications in blind source separation,” in *Seventeenth Annual Conference on Neural Information Processing Systems (NIPS-2003)*, Vancouver, Dec. 2003. [Online]. Available: http://www.bsp.brain.riken.jp/publications/2003/NIPS03LiCiAmShiCaoGu.pdf

[15] L. Vielva, D. Erdogmus, and J. Principe, “Underdetermined blind source separation using a probabilistic source sparsity model,” in *2nd International Workshop on Independent Component Analysis and Blind Signal Separation*, June 2000.

[16] D. Luengo, I. Santamaria, L. Vielva, and C. Pantaleh, “Underdetermined blind separation of sparse sources with instantaneous and convolutive mixtures,” in *IEEE XIII Workshop on Neural Networks for Signal Processing*, 2003.

[17] A. Jourjine, S. Rickard, and O. Yılmaz, “Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures,” in *Proc. ICASSP2000, June 5-9, 2000, Istanbul, Turkey*, June 2000.

[18] P. Bofill, “Underdetermined blind separation of delayed sound sources in the frequency domain.” *Neurocomputing*, vol. 55, no. 3-4, pp. 627–641, 2003.

[19] T. Melia and S. Rickard, “Extending the duet blind source separation technique,” in *Workshop on Signal Processing with Adaptative Sparse Structured Representation (SPARS05)*, November 2005.

[20] P. O’Grady, B. Pearlmutter, and S. Rickard, “Survey of sparse and non-sparse methods in source separation,” *International Journal of Imaging Systems and Technology*, vol. 15, no. 1, 2005.

[21] F. Theis and E. Lang, “Formalization of the two-step approach to overcomplete bss,” in *Proc. 4th Intern. Conf. on Signal and Image Processing (SIP’02) (Hawaii)*, N. Younan, Ed., 2002.

[22] K. Delgado, J. Murry, B. Rao, K. Engan, T. Lee, and T. Sejnowski, “Dictionary learning algorithms for sparse representation,” *Neural Computation*, vol. 15, pp. 349–396, 2003.

[23] R. Balan, J. Rosca, S. Rickard, and J. O’Ruanaidh, “The influence of windowing on time delay estimates,” in *Conf. on Info. Sciences and Systems (CISS)*, vol. 1, Princeton, NJ, March 2000, pp. WP1–(15–17).

[24] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.

[25] K. Gröchenig, *Foundations of Time-Frequency Analysis*. Boston: Birkhäuser, 2001.

[26] S. Rickard and M. Fallon, “The gini index of speech,” in *Conference on Information Sciences and Systems*, March 2004.

[27] S. Rickard and O. Yılmaz, “On the approximate W-disjoint orthogonality of speech,” in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, Orlando, Florida, May 13–17 2002, pp. 529–532.

[28] Y. Li, S.-I. Amari, A. Cichocki, D. Ho, and S. Xie, “Underdetermined blind source separation based on sparse representation,” *IEEE Transactions on Signal Processing*, vol. 54, 2006.

[29] M. Zibulevsky and B. A. Pearlmutter, “Blind source separation by sparse decomposition in a signal dictionary,” *Neural Computation*, vol. 13, no. 4, pp. 863–882, 2001.

[30] D. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization,” in *Proc. Natl. Acad. Sci. USA 100 (2003), 2197-2202*.

[31] D. Donoho, “Sparse components of images and optimal atomic decompositions,” *Constructive Approximation*, vol. 17, pp. 352–382, 2001.

[32] ——, “Compressed Sensing,” *Preprint*. [Online]. Available: http://www-stat.stanford.edu/∼donoho/Reports/2004/CompressedSensing091604.pdf

[33] D. Malioutov, “A sparse signal reconstruction perspective for source localization with sensor arrays,” Master’s thesis, MIT, 2003.

[34] S. Winter, H. Sawada, and S. Makino, “On real and complex valued l1-norm minimization for overcomplete blind source separation,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 2005.

[35] G. Box and G. Tiao, “A further look at robustness via bayes theorem,” *Biometrika*, vol. 49, 1962.

[36] J. Nelder and R. Mead, “A simplex method for function minimization,” *computer journal*, vol. 7, pp. 308–313.

[37] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Philadelphia, PA: SIAM, 2001.

[38] J. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999, special issue on Interior Point Methods (CD supplement with software). [Online]. Available: citeseer.ist.psu.edu/sturm99using.html

[39] R. Gribonval, L. Benaroya, E. Vincent, and C. Fevotte, “Proposal for performance measurement in source separation,” in *Proceedings of 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, april 2003, pp. 763–768.

[40] S. Rickard, “Personal communication,” 2005.

[41] [Online]. Available: http://medi.uni-oldenburg.de/demo/demo_separation.html

[42] J. Anemuller and B. Kollmeier, “Adaptive separation of acoustic sources for anechoic conditions: a constrained frequency domain approach,” *Speech Commun.*, vol. 39, no. 1-2, pp. 79–95, 2003.

[43] D. L. Donoho and I. M. Johnstone, “Ideal denoising in an orthonormal basis chosen from a library of bases,” Tech. Rep., 1994. [Online]. Available: citeseer.ist.psu.edu/7496.html