ONLINE BLIND SOURCE SEPARATION BASED ON TIME-FREQUENCY SPARSENESS

Benedikt Loesch and Bin Yang

Chair of System Theory and Signal Processing, University of Stuttgart
Email: {benedikt.loesch, bin.yang}@LSS.uni-stuttgart.de

ABSTRACT

Recently, blind source separation (BSS) has been proposed to separate signals recorded by a microphone array in a reverberant environment. This paper deals with BSS of a *time-varying number* of *moving* sources, which often occurs in practical situations. We develop two online algorithms based on time-frequency (TF) sparseness that are able to deal with moving sources: A block-online algorithm that estimates the number of sources and a gradient-based online algorithm with prespecified maximum number of sources. Both algorithms are evaluated in simulations and real-world scenarios and show good separation performance.

Index Terms— blind source separation, adaptive beamforming, real-time separation, moving sources, time-frequency sparseness

1. INTRODUCTION

The task of blind source separation is to separate M (possibly) convolutive mixtures $x_m[i], m = 1, \ldots, M$ into N different source signals. Mathematically, we write the sensor signals $x_m[i]$ as a sum of convolved source signals

$$x_m[i] = \sum_{n=1}^{N} h_{mn}[i] * s_n[i], \quad m = 1 \ldots M \tag{1}$$

Our goal is to find signals $y_n[i]$, $n = 1 \ldots N$ such that, after solving the permutation ambiguity, $y_n[i] \approx s_n[i]$ or a filtered version of $s_n[i]$. In the case of moving sources, the impulse responses $h_{mn}[i]$ are time-varying. Several approaches to blind separation of moving sources have been proposed: [1, 2] use a blockwise ICA algorithm in the time-frequency domain. They can deal with (over)determined cases ($N \leq M$) only. [3] proposes maximum SNR beamforming in the TF domain together with a voice activity detection to find suitable segments for direction-of-arrival (DOA) estimation using GCC-PHAT.

Our online algorithms are based on the observation vector clustering (OVC) algorithm detailed in [4]. They use the so-called normalized observation vectors $\bar{\mathbf{X}}[k, l]$ as feature vectors where k is the frequency index and l is the time frame index, respectively. Each cluster with centroid $\mathbf{c}_n$ corresponds to a different source. By defining different cost functions of cluster centroids $\mathbf{c}_n$, which we are looking for,

$$\begin{aligned} \text{online: } \mathcal{J}_l &= \sum_k \min_n \|\bar{\mathbf{X}}[k, l] - \mathbf{c}_n\|^2, \\ \text{block-online: } \mathcal{J}_{l_1}^{l_2} &= \sum_{l=l_1}^{l_2} \mathcal{J}_l, \\ \text{offline: } \mathcal{J} &= \sum_l \mathcal{J}_l, \end{aligned} \tag{2}$$

we can derive different versions of the clustering algorithm. The separation algorithms presented in [4, 5] use k-means clustering and operate in offline mode since the cost function $\mathcal{J}$ is defined over the complete observation time interval. Hence they are limited in two ways: They can only deal with stationary sources and need to know the number of sources. As will be shown in section 6, the offline algorithm provides good separation for stationary sources, but fails when sources are moving. In this paper, we propose two modified algorithms to overcome these limitations: A block-online separation algorithm using the number of sources estimation technique (NOSET) [6] and a gradient-based online algorithm.

2. OBSERVATION VECTOR CLUSTERING

First, we briefly summarize the OVC algorithm. After a short time Fourier transform (STFT), we can approximate the convolutive mixtures in the time-domain as instantaneous mixtures at each frequency bin k:

$$\mathbf{X}[k, l] \approx \sum_{n=1}^{N} \mathbf{H}_n[k] S_n[k, l] \tag{3}$$

$\mathbf{X} = [X_1, \ldots, X_M]^T$ is called an observation vector and $\mathbf{H}_n = [H_{1n}, \ldots, H_{Mn}]^T$ is the vector of frequency responses from source n to all sensors. We assume that the microphone array is placed in the near-field of the sources. This implies that we can assume a strong direct-path and weak multipath components. The OVC algorithm consists of three steps: normalization, clustering, and reconstruction of the separated signals.

Normalization: All observation vectors $\mathbf{X}[k, l]$ are phase-normalized with respect to a reference sensor and normalized to unit length. They form clusters each of which corresponds to an individual source.

Clustering: The next step is to find clusters $C_1, \ldots, C_N$ of $\bar{\mathbf{X}}[k, l]$ with centroids $\mathbf{c}_n$. This can be done with the k-means clustering algorithm [4] applied to the cost function $\mathcal{J}$. However, k-means clustering has several drawbacks as discussed in [6]. Hence, we will use the NOSET algorithm from [6] together with its one step clustering procedure.

Reconstruction: To reconstruct the separated signals, we can design a binary TF mask $M_n[k, l]$ that extracts the TF points in each cluster or we can perform blind beamforming as discussed in [5]. This approach has the advantage of reducing or completely removing musical noise artifacts which are common in binary TF mask based separation.

3. OFFLINE SEPARATION USING NOSET

Our offline separation algorithm minimizing $\mathcal{J}$ in (2) consists of two steps: Source number and DOA estimation and separation using a beamformer array.

Source Number & DOA Estimation: We first perform a source number and DOA estimation using our algorithm NOSET [6]:

1. Select reliable TF points $\mathcal{I}$, i.e. TF points $[k, l]$ where we have a large enough phase difference among sensors and a high signal

power. The first criterion corresponds to a frequency selection $k > \tilde{k}$ above a certain threshold where phase estimates are reliable. The second criterion selects TF points that are characterized by one dominant source only.

2. Estimate the bearing $\hat{\theta}$ from the phase information of the normalized observation vectors and sensor positions for all TF points in $\mathcal{I}$ using a least-squares approach.

3. Form a histogram $R[\nu]$ of all $\hat{\theta}$. $R[\nu]$ is the number of bearing estimates that fall into bin number ν. The estimated number of sources $\hat{N}$ is determined by the number of relevant peaks in the histogram.

Separation: After we have estimated the number of sources and the corresponding DOAs, we use them as initial centroids in the normalized observation vector space and perform a single k-means iteration as proposed in [6]. After finding the cluster membership for all TF points, we perform blind beamforming using a beamformer array as discussed in [5]. The reason for using a beamformer array and not just a single beamformer is that a single beamformer can suppress at most $M-1$ interferers. Hence it cannot suppress all interferers if $M < N$. We first pre-separate the signals using binary TF masks. Then we design a beamformer array, consisting of beamformers dedicated to suppress different sets of $\min(M-1, N-1)$ interferers. Following the derivation in [5], we design several beamformers d to extract each source n using the following beamformer weights:

$$\mathbf{w}_{nd}[k] = \frac{\mathbf{R}_{nd}^{-1}[k]\mathbf{a}_n[k]}{\mathbf{a}_n^H[k]\mathbf{R}_{nd}^{-1}[k]\mathbf{a}_n[k]}. \quad (4)$$

$\mathbf{R}_{nd}[k]$ is the corresponding sensor correlation matrix of noise-plus-interference only. $\mathbf{a}_n[k]$ is the vector of transfer functions (steering vector) of source n to all sensors, estimated using a Wiener-Filter

$$\mathbf{a}_n[k] = \frac{\sum_l \mathbf{X}[k,l]\hat{y}_{nJ}^*[k,l]}{\sum_l |\hat{y}_{nJ}[k,l]|^2}, \quad (5)$$

where $\hat{y}_{nJ}[k,l] = M_n[k,l]X_J[k,l]$ and $M_n[k,l]$ is the mask of the desired signal n. $X_J[k,l]$ is the microphone signal at reference sensor J. Each beamformer is designed to suppress a different set of interferers and has a different input signal. To obtain the separated signals, we combine the outputs of different beamformers. This process is summarized in Fig. 1 for $M = 3, N = 4, n = 1$, where each of the three beamformers $\mathbf{w}_{1d} (1 \le d \le 3)$ tries to suppress $M - 1 = 2$ interferers.

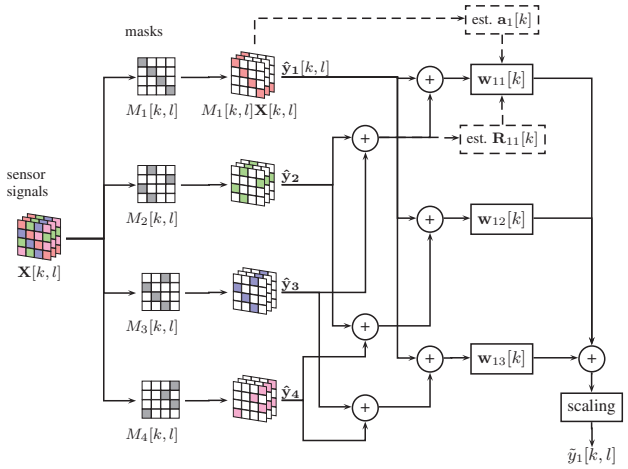

Fig. 1. Beamformer array for extracting source 1

After the beamforming step, additional optional binary TF masks can be used to further suppress the interference. Then we convert the separated signals back to the time domain using an inverse STFT.

4. BLOCK-ONLINE SEPARATION ALGORITHM

The block-online separation algorithm makes use of the NOSET algorithm described above and hence does not need to know the number of sources. It uses the cost function $\mathcal{J}_{l_1}^{l_2}$ from (2) with a block-size of $P = l_2 - l_1 + 1$ STFT frames. The algorithm consists of the following steps: source number and DOA estimation, source association/permutation and source separation.

Source Number & DOA Estimation: For each block of data, we perform the source number and DOA estimation using the NOSET algorithm as in section 3.

Source Association/Permutation: In order to ensure continuous signals, we need to associate the demixed signals of the current block p with those from the previous block $p-1$. This is done by finding the best match between the DOA estimates of the current block $\hat{\theta}[p]$ and of the previous block $\hat{\theta}[p-1]$. In order to deal with a changing source number (pauses in speech etc.), we perform the matching in the following way:

1. For each $\hat{\theta}_n[p]$, calculate the distances $d[p,j] \in [0^\circ, 180^\circ]$ between $\hat{\theta}_n[p]$ and all estimates of the previous block $\hat{\theta}_j[p-1]$:
$$d[p,j] = \begin{cases} \tilde{d}_{pj} & \text{if } \tilde{d}_{pj} < 180^\circ \\ 360^\circ - \tilde{d}_{pj} & \text{else} \end{cases}$$
with $\tilde{d}_{pj} = |\hat{\theta}_n[p] - \hat{\theta}_j[p-1]|$.
2. Find the minimal distance $d_{\min} = \min_j d[p,j]$ and the corresponding index $m = \arg\min_j d[p,j]$.
   (a) If $d_{\min} < t$, where t is a maximal allowable deviation, then associate $\hat{\theta}_n[p]$ with $\hat{\theta}_m[p-1]$.
   (b) If $d_{\min} \ge t$, then increment the number of tracked sources and associate $\hat{\theta}_n[p]$ with the new source.
3. If $\hat{N}[p] < \hat{N}[p-1]$, some DOA estimates $\hat{\theta}[p-1]$ of the previous block have not been associated with any DOA estimate $\hat{\theta}[p]$ of the current block. In this case, the corresponding source currently exhibits a speech pause and we are not able to estimate its DOA. We keep the previous DOA estimate in the set of tracked sources. DOA estimates that have not been active for a long period (e.g. 5 s) are removed.

Independently of our research, [7] proposes a similar procedure for source association.

Source Separation: After the source association step, we use the reordered DOA estimates $\hat{\theta}[p]$ as cluster centroids in the observation vector space. Separation for the block-online algorithm is performed for each block of data in the same way as for the offline algorithm.

5. ONLINE SEPARATION ALGORITHM

The online algorithm operates on a single frame basis. [8] developed a two-sensor gradient based online BSS algorithm based on binary TF masking. We extend this principle to OVC and blind beamforming. We use a gradient search for the cluster centroids $\mathbf{c}_n$ with prespecified maximum number of sources N. Inactive sources result in empty clusters. The cost function $\mathcal{J}_l$ for the l-th STFT frame is

$$\mathcal{J}_l = \sum_k \min_n \|\bar{\mathbf{X}}[k,l] - \mathbf{c}_n\|^2 = \sum_k \min(d_1, \ldots, d_N) \quad (6)$$

with $d_n = \|\bar{\mathbf{X}}[k,l] - \mathbf{c}_n\|^2$. In order to calculate the gradient of $\mathcal{J}_l$ with respect to $\mathbf{c}_n$, we use the approximation from [8]

$$\min(d_1, \ldots, d_N) = \frac{-1}{\lambda} \ln(e^{-\lambda d_1} + \cdots + e^{-\lambda d_N}) \quad (7)$$

where $\lambda > 0$ is a parameter to control the degree of the approximation. The cost function $\mathcal{J}_l$ is then approximated as

$$\mathcal{J}_l = \frac{-1}{\lambda} \sum_k \ln \sum_n e^{-\lambda \|\bar{\mathbf{X}}[k,l] - \mathbf{c}_n\|^2}. \quad (8)$$

Its gradient vector with respect to $\mathbf{c}_n$ is

$$\frac{\partial \mathcal{J}_l}{\partial \mathbf{c}_n} = -\sum_k \frac{2(\bar{\mathbf{X}}[k,l]-\mathbf{c}_n)e^{-\lambda\|\bar{\mathbf{X}}[k,l]-\mathbf{c}_n\|^2}}{\sum_n e^{-\lambda\|\bar{\mathbf{X}}[k,l]-\mathbf{c}_n\|^2}}. \quad (9)$$

The update rule for the cluster centroid $\mathbf{c}_n$ is

$$\mathbf{c}_n[l] = \mathbf{c}_n[l-1] - \beta\alpha_n[l]\frac{\partial \mathcal{J}_l}{\partial \mathbf{c}_n} \quad (10)$$

where $\beta > 0$ is a constant learning rate and $\alpha_n[l] > 0$ is a time-variant learning rate. Similar to [8], we select $\alpha_n[l]$ as a function of the amount of TF points associated with source n, that is

$$\alpha_n[l] = \frac{u_n[l]}{\sum_{l'=0}^{l}\gamma^{l-l'}u_n[l']},\ u_n[l] = \sum_k \frac{e^{-\lambda\|\bar{\mathbf{X}}[k,l]-\mathbf{c}_n\|^2}}{\sum_n e^{-\lambda\|\bar{\mathbf{X}}[k,l]-\mathbf{c}_n\|^2}} \quad (11)$$

with the forgetting factor $0 < \gamma \leq 1$.

The other separation steps such as masking, blind beamforming, and post-processing are similar to the block-online algorithm. We define a binary TF mask for each source n:

$$M_n[k,l] = \begin{cases} 1 & \text{if } d_n[k,l] < d_j[k,l] \quad \forall j \neq n \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

with $d_n[k,l] = \|\bar{\mathbf{X}}[k,l]-\mathbf{c}_n\|^2$. A computationally efficient version of the blind beamforming algorithm can be obtained by using a recursive update of the steering vectors $\mathbf{a}_n[k,l]$ and the correlation matrices $\mathbf{R}_{nd}[k]$

$$\mathbf{R}_{nd}[k,l] = \delta\mathbf{R}_{nd}[k,l-1] + \mathbf{n}_{nd}[k,l]\mathbf{n}_{nd}^H[k,l],$$
$$\mathbf{a}_n[k,l] = \delta\mathbf{a}_n[k,l-1] + \frac{\mathbf{X}[k,l]\hat{y}_{nJ}^*[k,l]}{|\hat{y}_{nJ}[k,l]|^2} \quad (13)$$

with the forgetting factor $0 < \delta \leq 1$. $\mathbf{n}_{nd}$ is the noise-plus-interference at the input of the beamformer $\mathbf{w}_{nd}$. The new weight vector of the beamformer can be calculated as

$$\mathbf{w}_{nd}[k,l] = \frac{\mathbf{R}_{nd}^{-1}[k,l]\mathbf{a}_n[k,l]}{\mathbf{a}_n^H[k,l]\mathbf{R}_{nd}^{-1}[k,l]\mathbf{a}_n[k,l]}. \quad (14)$$

By applying the matrix inversion lemma, we can also update the inverse correlation matrix $\mathbf{R}_{nd}^{-1}[k,l]$ recursively.

6. EXPERIMENTAL EVALUATION

For the experimental evaluation, we used a sampling frequency of $f_s = 8\,\text{kHz}$, a STFT with frame length 512 and 75% overlap, and a cross-array (⁘) with $M = 5$ microphones. The microphone spacing is $d = 4\,\text{cm} < c/f_s$, where $c = 343\,\text{m/s}$ is the propagation speed. All sources had equal power and were placed $0.8 \ldots 1.0\,\text{m}$ from the center of the array. The block size of the block-online algorithm was $P = 64$ STFT frames (roughly 1 second). The online algorithm used parameters $\lambda = 10, \beta = 0.006, \gamma = 0.95, \delta = 0.95$. The average real-time factors (= CPU-time / signal length) of our MATLAB implementations on a single core of an Intel Xeon E5440@2.83GHz for $N = 2, 4$ are: $0.21, 0.42$ (block-online algorithm) and $0.21, 0.40$ (online algorithm).

6.1. Stationary Sources

In the case of stationary sources, we evaluated the separation algorithms using samples from the TIMIT database [9]. They are played by loudspeakers and recorded in a real office room with $T_{60} = 520\,\text{ms}$. We consider source angular separations of $60^\circ, 36^\circ$. The average signal-to-noise ratio (SNR) of the microphone signals was between 20 and 30 dB. Generally, the block-online and online separation algorithms will perform worse than the offline algorithm, if we have a fixed number of stationary sources. Table 1 shows the separation performance in terms of the signal-to-interference-ratio (SIR) gains [4] for different number of sources N. SIR gains are calculated globally over the complete observation time interval and averaged over all sources. We see that the block-online algorithm performs only slightly worse than the offline algorithm in most cases as expected. The performance loss of the online algorithm is larger since it needs some time to learn the correlation matrices for beamforming and the used sound files are only a few seconds long[1].

ang. sep.	algorithm	$N=2$	$N=3$	$N=4$	$N=5$	$N=6$
60°	offline	14.82	15.69	15.55	15.36	14.27
	block-online	14.02	14.85	15.35	15.58	15.92
	online	12.41	13.22	13.43	13.46	13.80
36°	offline	13.39	13.64	12.82	12.52	12.69
	block-online	12.73	13.15	13.46	13.57	13.86
	online	10.06	10.69	11.01	11.33	11.85

Table 1. Global SIR gain in dB for stationary sources

6.2. Moving Sources

In order to have a precise reference for moving source positions, we performed simulations using the MATLAB ISM RoomSim toolbox [10]. The considered room was of size $3\,\text{m} \times 4\,\text{m} \times 2.5\,\text{m}$ and we chose reverberation times of $T_{60} = 100, 300\,\text{ms}$. SNR was 30 dB.

Fixed Number of Sources: We have $N = 2$ sources that move along a circle with radius 1.0 m. Source 1 moves from $\theta_1 = 30^\circ$ to $\theta_1 = 180^\circ$ and back and source 2 moves from $\theta_2 = 180^\circ$ to $\theta_2 = 330^\circ$ and back. The total simulation time is 24 s. We selected one male and one female speaker from the CHAINS corpus [11]. In order to have natural speech pauses, we used the short stories from the CHAINS corpus as source signals.

Fig. 2 shows the estimated angles $\hat{\theta}_{\text{blk}}[l]$ using our block-online algorithm and $\hat{\theta}_{\text{onl}}[l]$ using our online algorithm as well as the reference angles $\theta_{\text{true}}[l]$. The online-algorithm was initialized with the true angles $\theta_{\text{true}}[0]$, whereas the block-online algorithm does not need an explicit initialization. As we see, both algorithms accurately track the sources. During speech pauses, angle estimates are not updated.

The separation performance of the block-online, online and offline algorithm is summarized in Table 2. As expected, the offline algorithm fails to separate the source signals while our block-online and gradient-based online algorithm achieve good results. The reason for the failure of the offline algorithm is that it averages the DOAs of the moving sources. As a consequence, the separation performance drops significantly when the sources start moving as shown in Fig. 3. It shows the local SIR gains which are calculated over non-overlapping segments of 1 second and averaged over the two sources. Note that the simulations assumed omnidirectional

T_{60}	algorithm	source 1	source 2	average
100 ms	offline	1.67	3.19	2.43
	block-online	24.19	21.72	22.96
	online	30.26	26.01	28.14
300 ms	offline	3.95	3.96	3.96
	block-online	8.59	7.91	8.25
	online	11.58	10.58	11.08

Table 2. Global SIR gain in dB for moving sources

sources which results in a much lower direct-to-reverberant ratio than in a real world scenario with directional sources. This is also why the SIR gains for $T_{60} = 300\,\text{ms}$ in Table 2 are lower than those for the real world scenario with $T_{60} = 520\,\text{ms}$ in Table 1.

[1] Increasing the number of sources N results in almost the same SIR gains due to two competing effects: lower input SIR (due to more interfering sources) results in higher (possible) SIR gain, more interfering sources results in more overlap in the TF domain and hence a perfomance reduction.

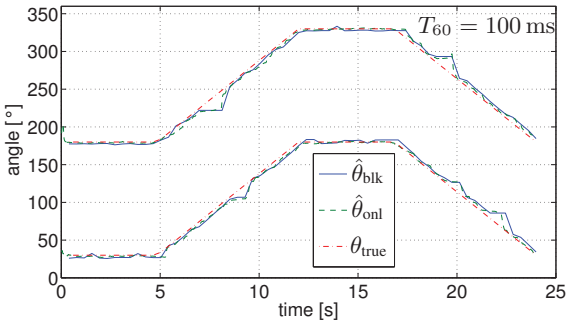

Fig. 2. DOA tracking, fixed number of moving sources

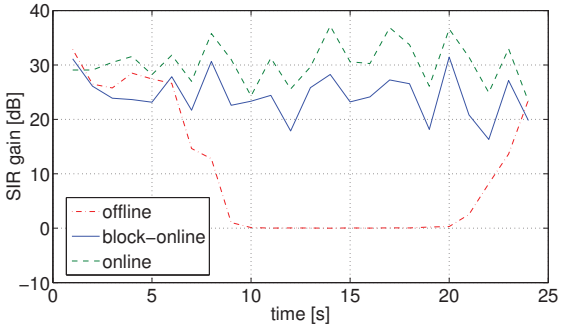

Fig. 3. Local SIR gain, fixed number of moving sources

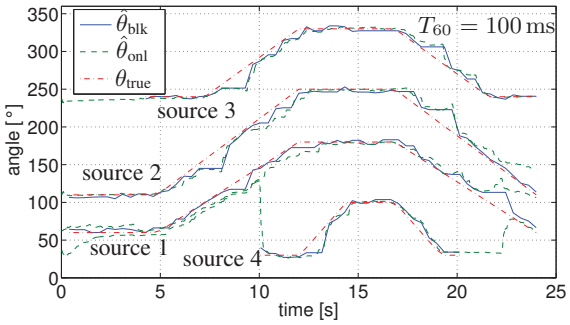

Fig. 4. DOA tracking, varying number of moving sources

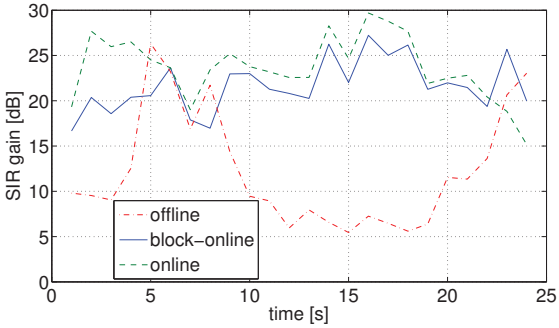

Fig. 5. Local SIR gain, varying number of moving sources

Varying Number of Sources: Fig. 4 and Fig. 5 show the DOA tracking perfomance and local SIR gains of the following scenario: We have a total of $N = 4$ sources, two of them are active all the time and two sources are only partially active. Source 3 is active in the time interval $[5\,\text{s}, 24\,\text{s}]$ whereas source 4 is active in $[10\,\text{s}, 20\,\text{s}]$. We again initialize the online algorithm with the true DOAs $\theta_{\text{true}}[0]$. The block-online algorithm has an advantage over the online algorithm since it is able to estimate the number of active sources within a block. For source 4, the online algorithm performs worse than the block-online algorithm since during the inactivity period of source 4, its DOA θ_4 follows the DOA of source 1. As soon as source 4 becomes active, the online algorithm moves θ_4 to the correct position. Global SIR gains are summarized in Table 3. Clearly, the block-online algorithm performs the best on average and especially for source 4.

	source 1	source 2	source 3	source 4	average
offline	7.12	5.97	10.24	3.19	6.63
block-online	19.87	20.57	23.89	25.35	22.42
online	17.83	25.92	25.64	12.24	20.41

Table 3. Global SIR gain in dB for varying number of moving sources, $T_{60} = 100\,\text{ms}$

7. CONCLUSION

In this paper we have presented two blind source separation algorithms based on TF sparseness that are able to deal with a time-varying number of moving sources. Experimental results have shown that both algorithms can accurately track and separate moving sources in mildly reverberant environments in real-time. The performance for stationary sources is almost the same as for the offline algorithm. The block-online algorithm needs neither the number of sources nor an initialization of DOAs. The online algorithm needs to know the maximum number of sources and at least an initialization close to the true DOAs. The block-online algorithm outperforms the online one when the number of sources is time-varying.

8. REFERENCES

[1] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Blind source separation for moving speech signals using blockwise ICA and residual crosstalk subtraction," *IEICE Transactions Fundamentals*, vol. E87-A, no. 8, pp. 1941–1948, August 2004.

[2] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Real-time blind source separation and DOA estimation using small 3-D microphone array," *Proc. IWAENC*, pp. 45–48, 2005.

[3] S. Araki, H. Sawada, and S. Makino, "Blind speech separation in a meeting situation with maximum SNR beamformers," *Proc. ICASSP*, 2007.

[4] S. Araki, H. Sawada, R. Mukay, and S. Makino, "Underdetermined blind sparse source separation of arbitrarily arranged multiple sensors," *Signal Processing*, vol. 87, no. 8, pp. 1833–1847, 2007.

[5] J. Cermak, S. Araki, H. Sawada, and S. Makino, "Blind source separation based on a beamformer array and time frequency binary masking," *Proc. ICASSP*, 2007.

[6] B. Loesch and B. Yang, "Source number estimation and clustering for underdetermined blind source separation," *Proc. IWAENC*, 2008.

[7] N. Madhu and R. Martin, "A scalable framework for multiple speaker localisation and tracking," *Proc. IWAENC*, 2008.

[8] S. Rickard, R. Balan, and J. Rosca, "Real-time time-frequency based blind source separation," *Proc. Int. Conf. on Independent Component Analysis and Signal Separation*, 2001.

[9] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallet, and N. S. Dahlgren, "The DARPA TIMIT Acoustic-Phonetic Continous Speech Corpus CDROM," `www.ldc.upenn.edu/lol/docs/TIMIT.html`, 1986.

[10] E. Lehmann, "Image-source method for room impulse response simulation (room acoustics)," `http://www.watri.org.au/~ericl/ism_code.html`, 2008.

[11] F. Cummins, M. Grimaldi, T. Leonard, and J. Simko, "The CHAINS corpus (characterizing individual speakers)," `http://chains.ucd.ie/`, 2006.