Adaptive Segmentation and Separation of Determined Convolutive Mixtures under Dynamic Conditions

Benedikt Loesch and Bin Yang

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

Abstract. In this paper, we propose a method for blind source separation (BSS) of convolutive audio recordings with short blocks of stationary sources, i.e. dynamically changing source activity but no source movements.It consists of a time-frequency sparseness based localization step to identify segments with stationary sources whose number is equal to the number of microphones. We then use a frequency domain independent component analysis (ICA) algorithm that is robust to short data segments to separate each identified segment. In each segment we solve the permutation problem using the state coherence transform (SCT). Experimental results using real room impulse responses show a good separation performance.

Key words: blind source separation, dynamic mixing conditions

1 Introduction

The task of convolutive blind source separation is to separate M convolutive mixtures into N different source signals. In this paper we consider dynamically changing source activity, i.e. active sources can change at any time during the recording but the sources cannot move.

With stationary mixing conditions we can apply frequency domain ICA with permutation correction to the complete recording (batch processing). However, the performance will be poor if the source positions change during the recording. To overcome this problem we can apply a frame-by-frame or block adaptive processing but performance will be limited by the convergence time and the limited amount of considered data. A better separation can be achieved if we run batch processing on each segment of $N = M$ stationary sources. This is why we propose to first find segments of $N = M$ stationary sources using a TF sparseness based localization step. This is done using source positions and pauses as segmentation cues. Once we have identified the segments, we apply a frequency domain ICA algorithm to each segment that can cope with short data segments. The permutation problem is solved using the state coherence transform (SCT) [1, 2] which is also robust to short data lengths.

Some recent works for dynamically changing source activity are [3, 4]. [3] models source activity with a hidden Markov model and switches off learning of

the demixing parameters for inactive sources. However, the computation complexity increases exponentially with the number of sources since all possible combinations of source activity need to be modelled. [4] proposes an online Bayesian learning procedure for instantaneous mixtures to incrementally estimate the mixing matrix and source signals in each time frame. This approach greatly reduces the computational complexity. However, it is not the purpose of this paper to compare the different approaches for dynamically changing mixing conditions. Instead we want to propose a simple but effective algorithm to find and separate segments of $N = M$ active sources.

2 Proposed Segmentation Algorithm

After a short-time Fourier transform (STFT), we can approximate the convolutive mixtures in the time-domain as instantaneous mixtures at each time-frequency (TF) point $[k, l]$:

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

$k = 1, \cdots, K$ is the frequency bin index, $l = 1, \ldots, L$ is the time frame index. $\mathbf{X} = [X_1, \ldots, X_M]^T$ is called an observation vector, $\mathbf{H}_n = [H_{1n}, \ldots, H_{Mn}]^T$ is the vector of frequency responses from source n to all sensors. $\tilde{N}$ in (1) reflects the total number of sources of which only up to $N = M$ sources are assumed to be active in each time frame l, i.e. the other source signals $S_n[k, l]$ are zero.

We assume that the direct path is stronger than the multipath components. This allows us to exploit the DOA information for segmentation. The proposed algorithm consists of two steps: normalization and segmentation.

2.1 Normalization

From the observation vectors $\mathbf{X}[k, l]$, we derive normalized phase vectors $\bar{\mathbf{X}}[k, l]$ which contain only the phase differences of the elements of $\mathbf{X}[k, l]$ with respect to a reference microphone J:

$$\bar{\mathbf{X}}[k,l] = \left[e^{j\cdot\arg(X_m[k,l]/X_J[k,l])}\right], \quad m = 1, \cdots, M \tag{2}$$

For a single active source, the phase of the ratio of two elements of $\mathbf{X}[k, l]$ is a linear function of the frequency index k (modulo 2π). We use a distance metric that includes mod 2π to estimate the direction-of-arrival (DOA) θ_n of the sources:

$$\|\bar{\mathbf{X}}[k,l] - \mathbf{c}[k,\theta]\|^2 = 2M - 2\cdot\sum\nolimits_{m=1}^{M} \cos\left(\arg\left[\frac{X_m[k,l]}{X_J[k,l]}\right] - 2\pi\Delta f k\tau_m(\theta)\right) \tag{3}$$

Δf is the frequency bin width. $\mathbf{c}[k, \theta] = [c_m]_{1\le m\le M} = \left[e^{j2\pi\Delta f k\tau_m(\theta)}\right]_{1\le m\le M}$ is a state vector which contains the expected phase differences between the microphones $m = 1, \cdots, M$ and the reference one J for a potential source at DOA θ. Using this distance metric and TF sparseness, we can localize the active sources. For more details, please refer to [5].

2.2 Segmentation Algorithm

After the normalization we calculate the function

$$\mathcal{J}_l(\theta) = \sum_k \rho(\|\bar{\mathbf{X}}[k,l] - \mathbf{c}[k,\theta]\|^2) \tag{4}$$

where $\rho(\cdot)$ is a monotonously decreasing nonlinear function which reduces the influence of outliers and increases DOA resolution. Inspired from [1], we propose to use $\rho(t) = 1 - \tanh(\alpha\sqrt{t})$ in (4). Independently of our research, [6] proposed a similar cost function $\mathcal{J}_l(\theta)$ for only two microphones. In the ideal case, the function $\mathcal{J}_l(\theta)$ in (4) shows maxima at the true source DOAs θ for frame l and a small value for other DOA values.

We want to use this two-dimensional function $\mathcal{J}_l(\theta)$ to detect source position changes and to find segments with $N = M$ stationary sources by looking for the cumulative source activity in the time interval $[l_{\text{start}}, l_{\text{end}}]$. By this we mean how many sources have been active in total during this time interval. For this purpose we define $\mathcal{J}(\theta) = f\left(\mathcal{J}_{l_{\text{start}}}(\theta), \cdots, \mathcal{J}_{l_{\text{end}}}(\theta)\right)$, where the generic function $f(\cdot)$ could be $\underset{l}{\text{mean}}(\cdot)$, $\underset{l}{\text{median}}(\cdot)$, $\underset{l}{\max}(\cdot)$, or $\underset{l}{\max_q}(\cdot)$. The operation $\underset{l}{\max_q}(\cdot)$ selects the q-th largest value from its arguments. The mean and median operation have the disadvantage of a long memory, i.e. they detect a new source too late. The max operation detects a new source very fast, but it is not robust to single spikes of $\mathcal{J}_l(\theta)$. In comparison, the $\max_q$ operation is more robust since $\mathcal{J}_l(\theta)$ should have a large value in at least q frames for a fixed θ before $\mathcal{J}(\theta)$ confirms the source activity with a large value as well for the same θ. However, the $\max_q$ operation detects a new source too late. Hence, we use a combination of the max and $\max_q$ approaches (Algorithm 1):

Algorithm 1 Search for segments with $N = M$ stationary sources

$l_{\text{start}} := 1, l_{\text{end}} := l_{\text{start}} + l_{\min}$, marker:=[ ], $l_{\text{prev}} := 1$, $l_{\text{b}} := 1$
while $l_{\text{end}} < L$ **do**
  Determine $\hat{N}$ using Algorithm 2 with $\mathcal{J}(\theta) := \underset{l}{\max}(\mathcal{J}_{l_{\text{start}}}(\theta), \cdots, \mathcal{J}_{l_{\text{end}}}(\theta))$
  if $\hat{N} \leq M$ **then**
    $l_{\text{end}} := l_{\text{end}} + 1$
  else
    Determine $\hat{N}_2$ using Algorithm 2 with $\tilde{\mathcal{J}}(\theta) = \underset{l}{\max_q}(\mathcal{J}_{l_{\text{prev}}}(\theta), \cdots, \mathcal{J}_{l_{\text{end}}}(\theta))$
    if $\hat{N}_2 > M$ **then**
      Append l_{b} to the list of segment boundaries: marker:=[marker l_{b}], $l_{\text{prev}} := l_{\text{b}}$
    end if
    Start a new segment: $l_{\text{b}} := l_{\text{end}}$, $l_{\text{start}} := l_{\text{b}}, l_{\text{end}} := l_{\text{start}} + l_{\min}$
  end if
end while

The proposed algorithm starts with a short segment of length $l_{\min}$ frames and increases the size of the current segment until $\hat{N} > M$ active sources are

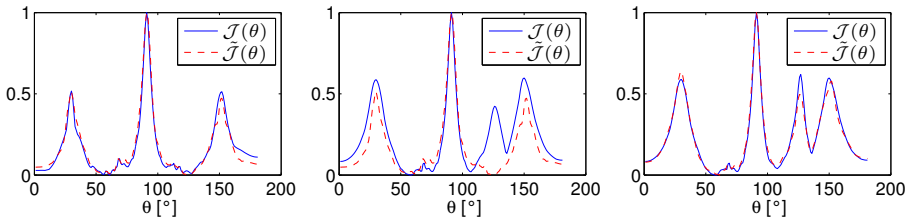
detected by $\mathcal{J}(\theta) = \max_l \mathcal{J}_l(\theta)$. We store the current frame as a potential segment boundary in l_b. We then start a new segment and increase this segment until we detect the next potential segment boundary by $\mathcal{J}(\theta) = \max_l \mathcal{J}_l(\theta)$. Now we verify the previously detected segment boundary l_b by checking if $\tilde{\mathcal{J}}(\theta) = \max_{l,q} \mathcal{J}_l(\theta)$ shows $\hat{N}_2 > M$ maxima for the combined segment $[l_\mathrm{prev}, l_\mathrm{end}]$ containing the previous and current segment. l_prev contains the last but one segment boundary. This process is repeated until the end of the recording. The number of sources $\hat{N}$ for the current segment is determined using Algorithm 2 by looking for the number of significant and distinct maxima of $\mathcal{J}(\theta)$ or $\tilde{\mathcal{J}}(\theta)$.

Algorithm 2 Source number estimation

```
Find all extrema of J(θ)
Find the distance h in height between each maximum and its neighbouring minima
Discard maxima with small h
Sort remaining maxima θ_n in descending order of J(θ_n)
n:=1, max_list:=[ ]
while J(θ_n) > t_1 do
  if (min |max_list − θ_n|) > t_2 then
    max_list:=[max_list θ_n]
  end if
  n:=n+1
end while
N̂ := length(max_list)
```

Fig. 1 illustrates $\mathcal{J}(\theta)$ and $\tilde{\mathcal{J}}(\theta)$ for three segments starting at 0 s and ending at 1.5 s, 3 s and 3.9 s. We want to identify segments with $N = M = 3$ sources. 3 sources at $\theta_{1,2,3} = 30°, 90°, 150°$ are active before 3 s. At the time instant 3 s, a new fourth source at $\theta_4 = 126°$ appears while the third source at $\theta_3 = 150°$ disappears. Clearly $\mathcal{J}(\theta)$ detects the fourth source as soon as it becomes active (at 3 s) since $\mathcal{J}(\theta)$ shows four distinct maxima in Fig. 1(b). $\tilde{\mathcal{J}}(\theta)$ takes additional 0.9 s to verify that it is truely a new source and not a spurious spike since $\tilde{\mathcal{J}}(\theta)$ shows three distinct maxima in Fig. 1(b) and four distinct maxima in Fig. 1(c).

Algorithm 1 works quite well, but sometimes it still detects a segment boundary too late if the newly active source does not start with a frame with high phase coherence, i.e. $\mathcal{J}(\theta)$ is not large enough. This can happen if the newly active source has a smaller power or there is no frame at the beginning of the segment where it is the single dominant source. However, we can use additional information based on pauses in the segmentation process: We first detect pauses of more than T frames by counting the number of consecutive frames where $\max_\theta \mathcal{J}_l(\theta)$ is small. This corresponds to frames with no source activity. We detect a pause end if the maximum of $\mathcal{J}(\theta)$ gets larger than a predefined threshold t_3, i.e. the coherence of the observed phase becomes large. This corresponds to one or multiple active sources. This procedure is summarized in Algorithm 3. Using the



(a) $0 - 1.5\,\mathrm{s}$ (b) $0 - 3\,\mathrm{s}$ (c) $0 - 3.9\,\mathrm{s}$

Fig. 1. Segmentation process using $\mathcal{J}(\theta)$ and $\tilde{\mathcal{J}}(\theta)$, new source becomes active at 3 s

detected pauses, we perform a segment verification step: If Algorithm 1 detects a segment boundary shortly after a pause we move the segment boundary to the end of the pause if this yields a segmentation with $N = M$ sources in the previous and current segment.

Algorithm 3 Pause detection

```
count:=0
for all l = 1 to L do
  if max_θ J_l(θ) < t_3 then
    count:=count+1
  else
    count:=0
  end if
  pause_count[l] := count
end for
pause_end:={l ∈ [1, ··· , L] : pause_count[l] = 0 ∨ pause_count[l − 1] > T}
```

3 Separation

After we have identified the segments containing $N = M$ active sources, we perform frequency domain ICA to separate the sources in each segment. We have to deal with the following two issues:

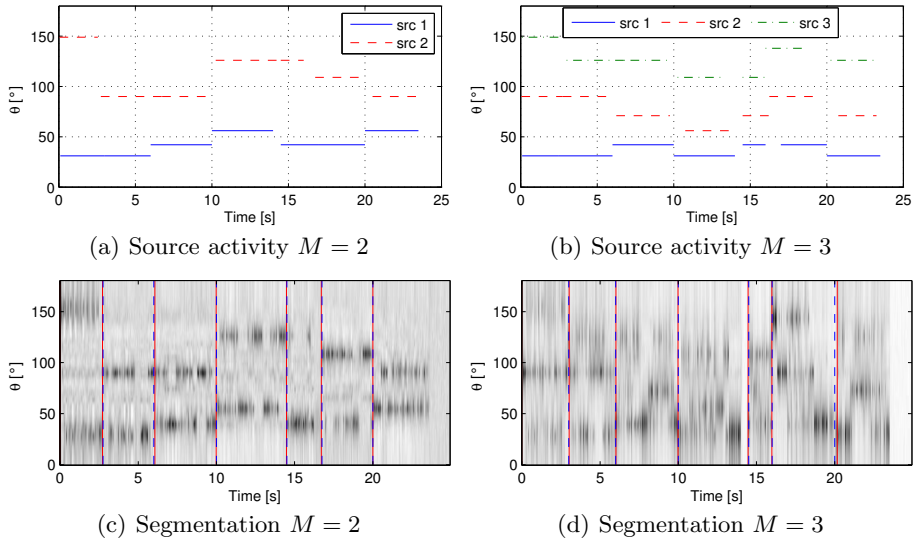
- **Choice of the ICA algorithm for short data segments.** It is well known that the performance of most ICA algorithms degrades if only a small amount of data is available. Since we are considering dynamically changing mixing conditions, we have to use an ICA algorithm that can deal with short amounts of data. [7] showed that a recursive initialization of the demixing matrices across frequencies improves the robustness of the scaled Infomax algorithm for short data segments. We use this separation algorithm below.
- **Permutation problem.** Since we are applying ICA to each frequency bin individually, the permutation problem has to be solved. For this task many

approaches have been proposed. They can be classified into a family using properties of the separated signals (e.g. correlation across frequency) and another one based on propagation model parameters or smoothness of the demixing matrices across frequency. Correlation based methods work well if the observable data length is sufficiently long. However, when the data length is short, performance decreases. We have shown in [2] that the multi-dimensional SCT is a robust way to solve the permutation problem even for short data lengths. Hence, we will use it to solve the permutation problem in the given context of short data segments with stationary sources. For more details please refer to [1, 2].

4 Experimental Results

4.1 Results using RWCP Database

We consider two scenarios using impulse responses from the E2A room ($T_{60} = 300\,\text{ms}$) of the RWCP database [8]: We use an uniform linear array (ULA) with $M = 2$ or $M = 3$ sensors with a total aperture of $d = 11\,\text{cm}$ and segments from the short stories of the CHAINS database [9]. The source activity and the corresponding source DOAs for the two scenarios are depicted in Fig. 2(a) and (b). Each scenario has 7 segments with different lengths and source DOAs.



(a) Source activity $M = 2$ (b) Source activity $M = 3$

(c) Segmentation $M = 2$ (d) Segmentation $M = 3$

Fig. 2. Source activity and resulting segmentation using our algorithm

Fig. 2(c) and (d) show $\mathcal{J}_l(\theta)$ from (4) as gray value and the detected segment boundaries (red solid lines) together with the true ones (blue dashed lines).

Clearly our algorithm detects the segments with $N = M$ sources very well since the estimated segment boundaries match the true boundaries.

For each segment found by our proposed segmentation algorithm, we run frequency domain ICA with the SCT for permutation correction. We used an STFT frame size of 4096 with 75% overlap. Evaluation of the separation quality is done using the BSS_EVAL toolbox [10] for each segment where there are $N = M$ active sources. We use the signal-to-interference ratio (SIR), signal-to-distortion ratio (SDR) and signal-to-artifact ratio (SAR) defined in [10] as separation performance measures. As proposed in the SISEC2010 task "Determined Convolutive Mixtures under Dynamic Conditions", we use an A-weighting filter before the evaluation of the performance measures to model the frequency characteristic of the human ear. The separation results for $M = 2$ and $M = 3$ are summarized in Table 1. Clearly, the proposed algorithm is able to separate the sources very well. Separation quality is influenced by the duration of the segments, the amount of activity for each source and the angular spacing between the sources. The more difficult case of $N = M = 3$ shows a slightly lower separation quality than $N = M = 2$.

Table 1. Separation performance for each segment in dB with A-weighting

	segment	1	2	3	4	5	6	7	mean
$N = M = 2$	SIR	19.4	21.1	21.1	17.5	18.9	16.9	16.9	18.9
	SDR	5.9	11.7	8.0	4.3	7.6	10.2	4.3	7.4
	SAR	6.2	12.3	8.3	4.6	8.0	11.4	4.6	7.9
$N = M = 3$	SIR	18.2	20.0	18.9	13.2	11.8	20.1	17.8	17.2
	SDR	6.2	8.8	6.6	4.0	3.4	10.8	7.5	6.8
	SAR	6.6	9.3	6.9	4.9	4.7	11.5	8.0	7.4

4.2 SISEC2010 Data

We have submitted our algorithm for the task "Determined Convolutive Mixtures under Dynamic Conditions" of the SISEC2010 campaign. The task uses impulse responses from a very reverberant room with $T_{60} = 700\,\text{ms}$ and different datasets for a microphone array with $M = 2$ microphones and spacing $d = 2, 6, 10\,\text{cm}$. Here we show the results for the example dataset for a microphone spacing of $d = 6\,\text{cm}$. The separation performance for the complete recording using an STFT frame size of 8192 with 75% overlap is summarized in Table 2 where we give the mean values of SIR, SAR and SDR with and without A-weighting and the corresponding standard deviations.

On the test dataset (`http://irisa.fr/metiss/SiSEC10/dynamic/dynamic_task2_all.html`), our algorithm outperforms the other approaches except for the case of $d = 2\,\text{cm}$. A possible explanation is that localization accuracy for $d = 2\,\text{cm}$ is insufficent to yield an accurate segmentation.

Table 2. Mean and standard deviation of separation performance for SISEC2010 example dataset in dB

without A-weighting			with A-weighting		
SIR	SDR	SAR	SIR	SDR	SAR
9.13 ± 2.99	3.21 ± 2.86	6.42 ± 1.54	12.13 ± 3.78	4.47 ± 3.71	6.47 ± 2.36

5 Conclusion

In this paper we have presented a method to separate recordings of short blocks of stationary sources. It is based on a segmentation of the recording into blocks of $N = M$ active sources through a time-frequency sparseness based DOA estimation for each time frame. Through a sliding time window, the change points are detected and the recordings are divided into blocks of $N = M$ active sources. We then use a frequency domain ICA algorithm suited for short data segments [7] together with permutation correction using the state coherence transform [1, 2]. Experimental results show that our approach achieves good separation performance even when the source activity changes frequently.

References

1. F. Nesta, M. Omologo, and P. Svaizer, "A novel robust solution to the permutation problem based on a joint multiple TDOA estimation," *Proc. International Workshop for Acoustic Echo and Noise Control (IWAENC)*, 2008.
2. B. Loesch, F. Nesta, and B. Yang, "On the robustness of the multidimensional state coherence transform for solving the permutation problem of frequency-domain ICA," *Proc. ICASSP*, 2010.
3. A. Masnadi-Shirazi, W. Zhang, and B.D. Rao, "Glimpsing indepdendent vector analysis: Separation more sources than sensors using active and inactive states," *Proc. ICASSP*, 2010.
4. H.-L. Hsieh and J.-T. Chien, "Online bayesian learning for dynamic source separation," *Proc. ICASSP*, 2010.
5. B. Loesch and B. Yang, "Blind source separation based on time-frequency sparseness in the presence of spatial aliasing," *submitted to LVA/ICA*, 2010.
6. Z.E. Chami, A. Guerin, A. Pham, and C. Serviere, "A phase-based dual microphone method to count and locate audio sources in reverberant rooms," *Proc. IEEE Workshop on Applications of Signal processing to Audio and Acoustics (WASPAA)*, 2009.
7. F. Nesta, P. Svaizer, and M. Omologo, "Separating short signals in highly reverberant environment by a recursive frequency domain BSS," *Proc. Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, May 2008.
8. Real World Computing Partnership, "RWCP Sound Scene Database in Real Acoustic Environment," `http://tosa.mri.co.jp/sounddb/indexe.htm`, 2001.
9. F. Cummins, M. Grimaldi, T. Leonard, and J. Simko, "The CHAINS corpus (characterizing individual speakers)," `http://chains.ucd.ie/`, 2006.
10. E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 4, 2006.