Adaptive Time-Domain Blind Separation of Speech Signals*

Jiří Málek[1], Zbyněk Koldovský[1,2], and Petr Tichavský[2]

[1] Faculty of Mechatronic and Interdisciplinary Studies
Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic
jiri.malek@tul.cz
[2] Institute of Information Theory and Automation, Pod vodárenskou věží 4,
P.O. Box 18, 182 08 Praha 8, Czech Republic

Abstract. We present an adaptive algorithm for blind audio source separation (BASS) of moving sources via Independent Component Analysis (ICA) in time-domain. The method is shown to achieve good separation quality even with a short demixing filter length (L = 30). Our experiments show that the proposed adaptive algorithm can outperform the off-line version of the method (in terms of the average output SIR), even in the case in which the sources do not move, because it is capable of better adaptation to the nonstationarity of the speech.

1 Introduction

The task considered in this paper is the blind separation of d unknown audio sources (BASS) from m recordings, where the unknown mixing process is convolutive and potentially dynamic, e.g., due to moving sources. It is assumed that the system changes slowly and may be considered being static in short time intervals. Therefore, within interval of the length P, the classical convolutive mixing problem is considered, which is described by

$$x_i(n) = \sum_{j=1}^{d} \sum_{\tau=0}^{M_{ij}} h_{ij}(\tau) s_j(n-\tau), \quad i = 1, \ldots, m. \tag{1}$$

Here, $x_1(n), \ldots, x_m(n)$ are the observed signals on microphones, $s_1(n), \ldots, s_d(n)$ are the unknown source signals, and h_{ij} are unknown impulse responses of length M_{ij}. The original sources are assumed to be independent, which allows the basis of the separation to be the Independent Component Analysis (ICA) [1]. For simplicity, we will assume that the number of sources d remains the same throughout the whole recording.

The separation of dynamic mixtures is usually done with block-by-block application of a method intended for stationary mixtures. The method may be

* This work was partly supported by Ministry of Education, Youth and Sports of the Czech Republic through the project 1M0572 and partly by Grant Agency of the Czech Republic through the projects 102/09/1278 and 102/08/0707.

V. Vigneron et al. (Eds.): LVA/ICA 2010, LNCS 6365, pp. 9–16, 2010.

modified more or less to respect the continuity of the (de)mixing process, and the outputting signals are synthesized from the separated signals on blocks. We call such methods *on-line*.

An on-line method applying ICA in the frequency domain was proposed by Mukai et al in [2]. An on-line method working in time-domain based on second-order statistics cost function was proposed by Buchner et al in [3]. Sparseness based on-line algorithm working in frequency domain was presented by Loesch and Yang in [4].

In this paper, we propose an online algorithm that comes from the BASS method from [5]. This original method applies an ICA algorithm to the mixed signals in time-domain to obtain independent components that correspond to randomly filtered versions of the original signals. The components are then grouped into clusters so that components in a cluster correspond to the same source. Finally, components of a cluster are used to reconstruct separated responses (spatial images) of the corresponding source on microphones. In the proposed on-line method, this process is modified so that the ICA and clustering algorithms adapt their internal parameters by performing one iteration in each block only. A new clustering criterion for the similarity of components, which is computationally more effective than the one in [6], is proposed. The speed of adaptivity can be driven by learning parameters and could be made very fast, due to fast convergence of ICA that is based on BGSEP from [7].

The following Section 2 describes all necessary details of the proposed on-line method. Section 3 demonstrates its performance in experiments with real-world recordings and Section 4 concludes the paper.

2 The Proposed Algorithm

The input signals are divided into overlapping blocks of length P, with the shift of T samples such that $R = P/T$ is an integer. The length of overlap of two consecutive blocks is thus $P - T$. The Ith block of the jth input signal will be denoted by

$$x_j^I(n) = x_j((I-1)\cdot T + n), \qquad n = 1, \ldots, T. \tag{2}$$

The uppercase superscript I will be used to denote data and quantities related to the Ith block. A separation procedure described below is successively applied to blocks of input signals and outputs blocks of separated microphone responses (spatial images) of the source signals.

Like the off-line method, the on-line procedure forms delayed copies of the microphone signals, (I) applies a simplified BGSEP algorithm to decompose the data matrix into its independent components and (II) uses a special fuzzy clustering method to group the independent components to form independent subspaces that represent the separated sources. The third step (III) consists in the reconstruction of the separated signals in each block and averaging the signals in the overlapping windows.

2.1 Step I: ICA (Simplified BGSEP Algorithm)

Let $\mathbf{X}^I$ be the data matrix from the Ith block of input signals defined as

$$\mathbf{X}^I = \begin{bmatrix} x_1^I(1) & x_1^I(2) & \dots & \dots & x_1^I(P) \\ x_1^I(2) & x_1^I(3) & \dots & \dots & x_1^I(P+1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^I(L) & x_1^I(L+1) & \dots & \dots & x_1^I(P+L) \\ x_2^I(1) & x_2^I(2) & \dots & \dots & x_2^I(P) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m^I(L) & x_m^I(L+1) & \dots & \dots & x_m^I(P+L) \end{bmatrix}, \tag{3}$$

where L is a free parameter corresponding to the length of the demixing MIMO filter.

The goal of the step is to find a demixing matrix $\mathbf{W}^I$ so that rows of $\mathbf{C}^I = \mathbf{W}^I\mathbf{X}^I$ are as independent as possible, thus, correspond to "independent" components (ICs) of $\mathbf{X}^I$.

The matrix $\mathbf{X}^I$ can be partitioned in a vertical way in M blocks of equal size, $(mL) \times (P/M)$,

$$\mathbf{X}^I = [\mathbf{X}^{I,1}, \dots, \mathbf{X}^{I,M}]. \tag{4}$$

The simplified BGSEP algorithm estimates $\mathbf{W}^I$ by a joint approximate diagonalization of a set of the covariance matrices

$$\mathbf{R}^{I,k} = \frac{M}{P}\mathbf{X}^{I,k}(\mathbf{X}^{I,k})^T, \qquad k = 1, \dots, M. \tag{5}$$

For convenience and computation savings we assume that the number of the matrices M is equal to the parameter R that appears in the division of the signal to overlapping blocks. Then, in the transition $\{\mathbf{R}^{I-1,k}\}_{k=1}^M \rightarrow \{\mathbf{R}^{I,k}\}_{k=1}^M$, the set of matrices remains unchanged, except for the removed matrix $\mathbf{R}^{I-1,1}$ and the added matrix $\mathbf{R}^{I,M}$.

The diagonalization proceeds by performing one iteration of the WEDGE algorithm - Weighted Exhaustive Diagonalization with Gauss itErations [7], with the weight matrices that are diagonal, optimized for the case when the signals obey the piecewise stationary model. The algorithm uses the estimate of demixing matrix from the previous segment $\mathbf{W}^{I-1}$ to partially diagonalize the matrices in (5)

$$\mathbf{P}^{I,k} = \mathbf{W}^{I-1}\mathbf{R}^{I,k}(\mathbf{W}^{I-1})^T \qquad k = 1, \dots, M. \tag{6}$$

As in [7], the demixing matrix $\mathbf{W}^I$ is obtained by updating $\mathbf{W}^{I-1}$ as

$$\mathbf{W}^I = (\mathbf{A}^I)^{-1}\mathbf{W}^{I-1} \tag{7}$$

where $\mathbf{A}^I$ has ones on its main diagonal, and the off-diagonal elements are obtained by solving the 2×2 systems

$$\begin{bmatrix} \mathbf{A}_{kl}^I \\ \mathbf{A}_{lk}^I \end{bmatrix} = \beta^I \begin{bmatrix} \mathbf{r}_{ll}^T\mathbf{Z}_{kl}\mathbf{r}_{ll} & \mathbf{r}_{kk}^T\mathbf{Z}_{kl}\mathbf{r}_{ll} \\ \mathbf{r}_{kk}^T\mathbf{Z}_{kl}\mathbf{r}_{ll} & \mathbf{r}_{kk}^T\mathbf{Z}_{kl}\tilde{\mathbf{r}}_{kk} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_{ll}^T\mathbf{Z}_{kl}\mathbf{r}_{kl} \\ \mathbf{r}_{kk}^T\mathbf{Z}_{kl}\mathbf{r}_{kl} \end{bmatrix}, \tag{8}$$

with

$$\mathbf{r}_{kl} = [(\mathbf{P}^{I,1})_{kl}, \ldots, (\mathbf{P}^{I,M})_{kl}]^T, \tag{9}$$

and

$$\mathbf{Z}_{kl} = \mathrm{diag}\left(\frac{1}{(\mathbf{P}^{I,1})_{kk}(\mathbf{P}^{I,1})_{ll}}, \ldots, \frac{1}{(\mathbf{P}^{I,M})_{kk}(\mathbf{P}^{I,M})_{ll}}\right) \tag{10}$$

for $k, l = 1, \ldots, mL$, $k > l$. The variable β^I in (8) does not exist in the original WEDGE algorithm: it is added here to control the speed of algorithm's convergence. The choice of β^I will be discussed later in Section 2.4.

2.2 Step II: Clustering of Independent Components

Similarity of ICs. Due to the indeterminacy of ICA, the ICs of $\mathbf{X}^I$ are arbitrarily filtered versions of the original signals. To recognize whether two components correspond to the same source, we compute their generalized cross-correlation coefficients known as GCC-PHAT [8]. The coefficients are invariant to the magnitude spectra of the signals and depend on their phase spectra only, which makes them appropriate for the similarity evaluation.

Let $C_i^I(k)$ and $C_j^I(k)$ denote the Fourier transform of the ith and jth component, respectively, $i, j = 1, \ldots, mL$, and k denotes the frequency index. The GCC-PHAT coefficients of the components, denoted by $g_{ij}^I(n)$, are equal to the inverse Fourier transform of

$$G_{ij}^I(k) = \frac{C_i^I(k) \cdot C_j^I(k)^*}{|C_i^I(k)| \cdot |C_j^I(k)|}, \tag{11}$$

where * denotes the complex conjugation. Fast computation of $g_{ij}^I(n)$ can be done by means of the FFT.

If the components correspond exactly to the same source, i.e. without any residual interference, $g_{ij}^I(n)$ is equal to delayed unit impulse function, where the delay cannot be greater than L. Hence, the similarity between the ith and jth component can be measured by $\sum_{n=-L}^{L} \left|g_{ij}^I(n)\right|$ and the matrix of mutual similarity $\mathbf{D}^I$ can be computed according to

$$\mathbf{D}_{ij}^I = \sum_{n=-L}^{L} \left|g_{ij}^I(n)\right| + \beta_2 \cdot \mathbf{D}_{ij}^{I-1}, \qquad i, j = 1, \ldots, mL,\ i \neq j, \tag{12}$$

where β_2 is a learning parameter, $0 \leq \beta_2 \leq 1$. The diagonal elements of $\mathbf{D}^I$ have no importance for the clustering and are all set to 1.

Clustering Algorithm. For simplicity, we assume that the number of sources d is known and does not change in time. The goal is thus to find d clusters of components according to their mutual similarity given by $\mathbf{D}^I$. We propose to use the Relational Fuzzy C-Means algorithm (RFCM) from [9], which allows tracking of continual changes of the clusters.

The affiliation of a component to a cluster is expressed by a value from $[0, 1]$ where 0 means that the component does not belong to the cluster and vice versa. Let $\boldsymbol{\Lambda}^I_{kj}$ be the kjth element of a $d \times mL$ partition matrix $\boldsymbol{\Lambda}^I$ and represents the affiliation of the jth component to the kth cluster. By definition it holds that $\sum_{k=0}^{d} \boldsymbol{\Lambda}^I_{kj} = 1$.

Now, let $\mathbf{B}$ denotes the dissimilarity matrix whose elements are $\mathbf{B}^I_{ij} = 1/\mathbf{D}^I_{ij}$ for $i \neq j$ and $\mathbf{B}^I_{ii} = 0$. Let $\boldsymbol{\mu}^{I,f}_k$ be a $mL \times 1$ vector defined as $\boldsymbol{\mu}^{I,f}_k = [(\boldsymbol{\Lambda}^I_{k1})^f, \ldots, (\boldsymbol{\Lambda}^I_{k(mL)})^f]^T / \sum_{j=1}^{mL} (\boldsymbol{\Lambda}^I_{kj})^f$ called the prototype of the kth cluster associated with a "fuzzyfication" parameter f, $f > 1$. (We use the experimentally verified value $f = 1.5$). The transition of $\boldsymbol{\Lambda}^{I-1}$ to $\boldsymbol{\Lambda}^I$ is given as one iteration of RFCM as

$$\boldsymbol{\Lambda}^I_{kj} = \Big(\sum_{i=1}^{d} (\mathbf{V}_{kj} / \mathbf{V}_{ij})^{1/(f-1)} \Big)^{-1}, \tag{13}$$

where

$$\mathbf{V}_{kj} = \big(\mathbf{B}^I \boldsymbol{\mu}^{(I-1),f}_k \big)_j - \frac{1}{2} \big(\boldsymbol{\mu}^{(I-1),f}_k \big)^T \mathbf{B}^I \, \boldsymbol{\mu}^{(I-1),f}_k \tag{14}$$

is the distance of the jth component to the prototype $\boldsymbol{\mu}^{I,f}_k$ (for details see [9]).

2.3 Step III: Reconstruction

The contribution of ICs of the kth cluster to $\mathbf{X}^I$ is given by matrix

$$\widehat{\mathbf{S}}^I_k = (\mathbf{W}^I)^{-1} \mathrm{diag}\big[(\boldsymbol{\Lambda}^I_{k1})^\alpha, \ldots, (\boldsymbol{\Lambda}^I_{k,mL})^\alpha \big] \, \mathbf{C}^I, \tag{15}$$

where α is an adjustable positive parameter. This matrix has analogous structure as $\mathbf{X}^I$ in (3). In the ideal case the rows of $\widehat{\mathbf{S}}^I_k$ contain delayed microphone responses of the kth estimated source only. The response of the kth source at the ith microphone is therefore estimated by summing these rows as

$$\hat{s}^{i,I}_k(n) = \frac{1}{L} \sum_{q=1}^{L} \big(\widehat{\mathbf{S}}^I_k \big)_{(i-1)L+q, n+q-1}, \tag{16}$$

where $\big(\widehat{\mathbf{S}}^I_k \big)_{\alpha,\beta}$ is the $\alpha\beta$th element of the matrix $\widehat{\mathbf{S}}^I_k$.

Finally, the overall outputs of the on-line algorithm are synthesized by putting together the estimated blocks of separated signals. The overlapping parts are averaged using a windowing function, for example, the Hann window.

2.4 Implementation Details

The speed of convergence of the ICA can be driven through the parameter β^I in (8). We found it helpful to increase the speed when the clusters of ICs did not seem well separated in the previous block of data. Otherwise, β^I can be close to zero to maintain the continuity. Therefore, we take

$$\beta^I = (1 - \gamma^{I-1})/2. \tag{17}$$

γ^I is the Silhouette index [10] of the hard clustering which is derived from the fuzzy clustering. Let K_k be the set of indices of the components for which $\boldsymbol{\Lambda}^I_{kj} = \max_\ell \boldsymbol{\Lambda}^I_{\ell j}$ (the kth cluster is the closest one to them). The Silhouette index is defined through $\gamma^I = \frac{1}{mL}\sum_{i=1}^{mL}\gamma_i^I$, where

$$\gamma_i^I = \frac{\min_{j\notin K_k}(\mathbf{B}^I_{ij}) - \frac{1}{|K_k-1|}\sum_{j\in K_k, i\neq j}\mathbf{B}^I_{ij}}{\max\{\min_{j\notin K_k}(\mathbf{B}^I_{ij}), \frac{1}{|K_k-1|}\sum_{j\in K_k, i\neq j}\mathbf{B}^I_{ij}\}}. \tag{18}$$

The Silhouette index reflects the separateness of clusters as it takes values from [-1,1], where negative values mean poor separateness and vice versa.

The whole algorithm can be initialized so that $\mathbf{W}^0$ is the outcome of the BGSEP algorithm applied to $\mathbf{X}^1$ and the components $\mathbf{W}^0\mathbf{X}^1$ are grouped by the full RFCM algorithm.

3 Experiments

We present two experiments evaluated by means of the BSS_EVAL toolbox [11] using the true sources. The results are presented in the form of three criteria: (i) Signal-to-Interference Ratio (SIR), (ii) Signal-to-Distortion Ratio (SDR), and (iii) Signal-to-Artifact Ratio (SAR).

3.1 Fixed Source Positions

In this experiment, we examine the online algorithm in separating stationary mixtures of speech signals. Positions of the sources and the microphones were fixed. We compare it with the results of the original method from [5]. Hereinafter, the proposed on-line method will be referred to as *on-line T-ABCD*, while the original method will be named *off-line T-ABCD*[1].

To this end, we use data from the publicly available sites of Hiroshi Sawada[2]. The recordings of *four sources using four microphones* are considered. The original signals are utterances of the length 7 s sampled by 8 kHz. The reverberation time of the room is 130 ms. Omnidirectional microphones were used.

The on-line and off-line T-ABCD were both applied with $L = 30$. The other parameters of the on-line method were set to $P = 6144$, $T = 512$, $\beta_2 = 0.95$ and $\alpha = 3$. The separation results are evaluated block-by-block of the same size as in the on-line method. Table 1 summarizes the results averaged over all blocks, separated microphone responses, and sources.

On-line T-ABCD achieves better results in terms of SIR and SDR than the off-line algorithm. It points out to the fact that the on-line method is able to adapt the separating filters throughout the recordings respecting the nonstationarity of sources. On the other hand, the time-invariant separation done by off-line T-ABCD produces less artifacts as indicated by SAR.

[1] The acronym "T-ABCD" comes from the original method as it is does Time-domain Audio source Blind separation based on the Complete Decomposition of the observation space.

[2] `http://www.kecl.ntt.co.jp/icl/signal/sawada/`

Table 1. Results of separation of sources at fixed positions

	SIR[dB]	SDR[dB]	SAR[dB]
on-line T-ABCD	8.43	1.58	4.41
off-line T-ABCD	6.25	1.09	5.38

Table 2. Results of separation of data simulating dynamic conditions

2 cm	SIR[dB]	SDR[dB]	SAR[dB]
o. T-ABCD	10.39	3.87	6.16
Nesta	11.21	4.59	6.54

6 cm	SIR[dB]	SDR[dB]	SAR[dB]
o. T-ABCD	8.77	1.44	4.45
Nesta	7.60	1.37	5.77

3.2 Moving Sources

In this experiment, we consider data given in the task "Determined convolutive mixtures under dynamic conditions" (Audio Signal Separation) in the SiSEC 2010 evaluation campaign organized at this conference[3]. The data simulate dynamic conditions so that *the maximum of two of six speakers* located at fixed positions *around a stereo microphone array* were active at a time. The separating algorithm applied to the data thus needs to adapt to active speakers. The distances of microphones were 2 and 6 cm, and the sampling rate was 16 kHz.

We compare the proposed on-line T-ABCD with the frequency-domain BSS method by Francesco Nesta et al [12,13]. The online method was applied with $L = 30$, $P = 6144$, $T = 512$, $\beta_2 = 0.95$ and $\alpha = 4$. The Nesta's method uses FFT of the length 4096 samples with 75% overlap. As the method works off-line, it was applied independently on disjoint blocks of 1 second where the maximum of two sources were active.

The proposed method appears to be slightly inferior to the frequency-domain method if the distance of the microphones is 2cm, but it achieves better results if the distance is 6cm. We conclude that the on-line T-ABCD seems to outperform the frequency-domain algorithm in cases of larger microphone distances, where the spatial aliasing occurs.

3.3 Computational Demands

The experiments mentioned above were performed on a computer with single core 2.6 Ghz processor with 2 GB RAM. The on-line T-ABCD was implemented in Matlab environment. The computational demands of the algorithm depend on the demixing filter length L. The mixture signals in Section 3.2 were 3 minutes, 29 seconds long, sampled by 16kHz. The on-line T-ABCD separation lasted 14 minutes, 36 seconds ($L = 30$). Although the implementation in Matlab may be considered as rather slow and inefficient, this separation task can be performed in real time when $L = 10$. Mixtures of two sources sampled by 8 kHz can be separated in real-time when $L = 18$.

[3] `http://sisec.wiki.irisa.fr/`

4 Conclusions

We have proposed a method for blind separation of moving audio sources. The algorithm applies the fast converging BGSEP algorithm and fuzzy clustering RFCM algorithm. It is shown that presented time-domain method (with rather short separating filters) is able to achieve results that are comparable to the frequency domain BSS algorithm. The experiment with fixed sources suggests the ability of the proposed method to adapt the separation to the nonstationarity of the data as well.

References

1. Comon, P.: Independent component analysis: a new concept? Signal Processing. 36, 287–314 (1994)
2. Mukai, R., Sawada, H., Araki, S., Makino, S.: Blind Source Separation for Moving Speech Signals Using Blockwise ICA and Residual Crosstalk Subtraction. IEICE Transactions Fundamentals E87-A(8), 1941–1948 (2004)
3. Buchner, H., Aichner, R., Kellermann, W.: A Generalization of Blind Source Separation Algorithms for Convolutive Mixtures Based on Second-Order Statistics. IEEE Trans. on Speech and Audio Proc. 13(1), 120–134 (2005)
4. Loesch, B., Yang, B.: Online blind source separation based on time-frequency sparseness. In: ICASSP 2009, Taipei, Taiwan (2009)
5. Koldovský, Z., Tichavský, P.: Time-domain blind audio source separation using advanced component clustering and reconstruction. In: HSCMA 2008, Trento, Italy, pp. 216–219 (2008)
6. Koldovský, Z., Tichavský, P.: Time-domain blind audio source separation using advanced ICA methods. In: Interspeech 2007, Antwerp, Belgium (2007)
7. Tichavský, P., Yeredor, A.: Fast Approximate Joint Diagonalization Incorporating Weight Matrices. IEEE Transactions of Signal Processing 57(3), 878–891 (2009)
8. Knapp, C.-H., Carter, G.-C.: The Generalized Correlation Method for Estimation of Time Delay. IEEE Transactions on Signal Processing 24(4), 320–327 (1976)
9. Hathaway, R.-J., Bezdek, J.-C., Davenport, J.-W.: On relational data versions of c-means algorithm. Pattern Recognition Letters (17), 607–612 (1996)
10. Rousseeuw, J.P.: Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics 20, 53–65 (1987)
11. Févotte, C., Gribonval, R., Vincent, E.: BSS EVAL toolbox user guide. IRISA, Rennes, France, Tech. Rep. 1706 (2005), `http://www.irisa.fr/metiss/bsseval/`
12. Nesta, F., Svaizer, P., Omologo, M.: A BSS method for short utterances by a recursive solution to the permutation problem. In: SAM 2008, Darmstadt, Germany (2008)
13. Nesta, F., Svaizer, P., Omologo, M.: A novel robust solution to the permutation problem based on a joint multiple TDOA estimation. In: IWAENC 2008, Seattle, USA (2008)