Technical University of Liberec

Faculty of Mechatronics,
Informatics and Interdisciplinary
Studies

Doctoral Thesis

December 2010 **Jiří Málek**

Technical University of Liberec

Faculty of Mechatronics, Informatics and Interdisciplinary Studies

Institute of Information Technology and Electronics

Blind audio source separation via Independent Component Analysis

Ing. Jiří Málek

Doctoral Thesis

Supervisor: Ing .Zbyněk Koldovský Ph.D.

Ph.D. Program: Technical Cybernetics

Branch of Study: Electronics and Informatics

Declaration

I declare herewith that this Ph.D. thesis is my own work and that all used resources are listed in the Bibliography section.

Jiří Málek
Liberec, December 10, 2010

Acknowledgements

Herewith I would like to thank my supervisor Zbyněk Koldovský for his most valuable advice and for the time he devoted to my research.

I would also like to thank professor Jan Nouza for his help and support during the first year of my Ph.D. studies.

I am very grateful to all collaborators for their efforts and patience, namely I want to thank Petr Tichavský, who helped me a great deal with the correction of my papers. Thanks to Yannick Deville and Shahram Hosseini for the pleasant and enlightening internship in the Astrophysics Lab, Toulouse.

The thesis could not be completed without the support of the Department of Information Technology and Electronics (ITE) at the Faculty of Mechatronics, Informatics and Interdisciplinary Studies. Financial support for the work was provided through the Student Grant Scheme (SGS) at the Technical University of Liberec and through the project P103/11/1947 of the Grant Agency of the Czech Republic.

Abstract

This thesis deals with the general problem of Blind Source Separation (BSS), the main focus being on audio sources. In an appendix, a research outside the topic of BSS is disscused: The automatic classification of biomedical signals.

The general goal of the blind source separation is to estimate unknown source signals from their measured mixtures, while there is no information about the mixing process available. This task is too general and cannot be solved without additional assumptions about the unknown sources. One such assumption is the mutual independence of the source signals. Methods performing the separation based on the independence assumption are denoted as Independent Component Analysis (ICA). Most algorithms described in this thesis are based on ICA.

The thesis consists of three main parts. The first part deals with the separation of non-stationary instantaneous mixtures. The author contributed to the proposal of an ICA algorithm called Block EFICA. This algorithm is proved to be asymptotically efficient (i.e. gives the most accurate estimates of the sources possible) provided that the source signals are block-wise stationary and have constant variances.

The second and third part concerns the separation of convolutive mixtures of audio sources. This problem can be solved via ICA in the time domain, for example by an existing method known as T-ABCD, which was proposed Koldovský and Tichavský. Modifications for this method are presented. These modifications improve the reconstruction of the sources in the case of imperfect ICA separation. Furthermore, an "online" version of the T-ABCD algorithm is proposed. It is able to adapt to the changes in the mixing system and to the non-stationarity of the source signals.

The appendix deals with automatic classification of medical signals which originate in the screening examination of lower limbs arteries. This examination, performed via inexpensive ultrasound units, aims at an early diagnostics of Peripheral Arterial Disease. Automatic classifiers for the data measured by mentioned ultrasound units are designed and trained. These classifiers are able to assign the signals to predefined classes, which reflect the degree of arterial occlusion.

Contents

List of Figures

List of Tables

List of symbols

General notation

$\mathbf{Y}$	Matrix $\mathbf{Y}$
$\mathbf{y}_i$ or $(\mathbf{Y})_i$	The ith row of $\mathbf{Y}$
$\mathbf{Y}(j)$ or $(\mathbf{Y})_{*j}$	The jth column of $\mathbf{Y}$
$y_i(j)$ or $\mathbf{Y}_{ij}$	The ij-th entry of $\mathbf{Y}$

Most important symbols

d	Number of sources
m	Number of sensors/microphones
	Within this thesis, determined case is assumed, i.e. $d = m$
$\mathbf{S}$	$m \times N$ matrix of the original signals
$\widehat{\mathbf{S}}$	Estimate of matrix $\mathbf{S}$
$\mathbf{X}$	$m \times N$ matrix of the mixed signals
N	Number of available samples in $\mathbf{S}, \mathbf{X}$
$\mathbf{A}$	$m \times m$ mixing matrix
$\mathbf{Z}$	$m \times N$ matrix of whitened mixtures
$\mathbf{W}$	$m \times m$ demixing matrix
L	Desired demixing filter length (Convolutive mixtures)
$\tilde{\mathbf{X}}$	Observation matrix of size $mL \times N$ (Convolutive mixtures)
$\tilde{\mathbf{S}}$	Source matrix of size $mL \times N$ (Convolutive mixtures)
$\mathbf{C}$	Matrix of independent components of $\tilde{\mathbf{X}}$
$\mathbf{I}$	The identity matrix
$\mathbf{1}_N$	$N \times 1$ vector of ones
E	The expected value operator
$\hat{\mathrm{E}}$	The sample mean operator

List of abbreviations

AJD	Approximate Joint Diagonalization
ASR	Automatic Speech Recognition
BARBI	Block AutoRegressive Blind Identification
BASS	Blind Audio Source Separation
BC	Bayesian Classifier
BPI	Brachial Pressure Index
BGL	Block Gaussian Likelihood
BGSEP	Block Gaussian SEParation
BN	Broadcast News
BPSK	Binary Phase Shift Keying
CRLB	Cramér Rao Lower Bound
EEG	Electro-EncephaloGram
EFICA	Efficient FastICA
FCM	Fuzzy C-Means
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FOBI	Fourth Order Blind Identification
GCC	Generalized Correlation Coefficients
GGD	Generalized Gaussian Distribution
ICA	Independent Component Analysis
IC	Independent Component
iid/i.i.d.	Independent and Identically Distributed
IIR	Infinite Impulse Response
ISR	Interference to Signal Ratio
JADE	Joint Approximate Diagonalization of Eigenmatrices
LBG	Linde-Buzo-Gray algorithm
MDC	Minimum Distance Classifier
MEG	Magneto-EncephaloGram
MLE	Maximum Likelihood Estimation
NMF	Non-negative Matrix Factorization
PAD	Peripheral Arterial Disease
PCA	Principal Component Analysis
PDF	Probability Density Function
PI	Pulsation Index
RFCM	Relational Fuzzy C-Means
RFCMdd	Relational Fuzzy C-Medoids
RPCM	Relational Possibilistic C-Means
RI	Resistance Index
RP	Resistance Parameter (of an artery)
SAHN	Standard Agglomerative Hierarchical Non-overlapping clustering

SAR	Signal to Artifact Ratio
SCA	Sparse Component Analysis
SDR	Signal to Distortion Ratio
SFS	Sequential Forward Search
SIR	Signal to Interference Ratio
STFT	Short Time Fourier Transform
SOBI	Second Order Blind Identification
SOS	Second Order Statistics
T-ABCD	Time-domain Audio source Blind separation based on Complete Decomposition
VTI	Velocity Time Index
WASOBI	Weights Adjusted SOBI
WEDGE	Weighted Exhaustive Diagonalization with Gauss itErations
WSS	Wide Sense Stationary

Chapter 1

Introduction

1.1 Blind source separation

The general goal of Blind Source Separation (BSS, [1]) is to estimate unknown sources from a set of observed mixtures. The estimation is performed with no prior information about either the sources or the mixing process. In this general context, the task cannot be solved. Therefore, specific restrictions are placed on the mixing model and the source signals in order to limit the generality.

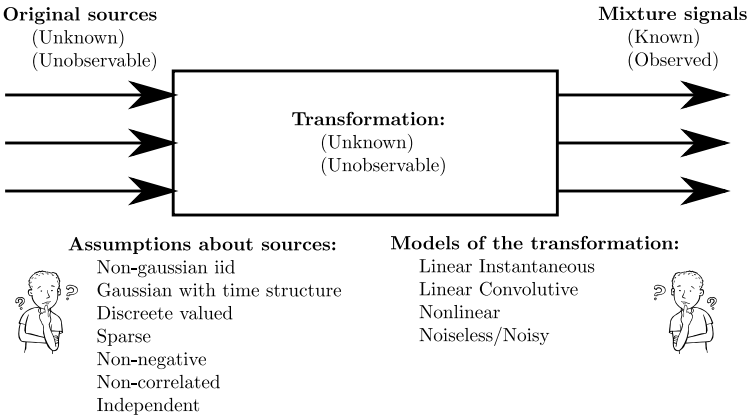


Figure 1.1: The general mixing transformation with weak assumptions on signals and mixing models

1.1.1 Models of the mixing process

There are two basic **noiseless models** which describe the mixing transformation. The **linear instantaneous model** defines observed mixtures as a weighted sum of unknown source signals. Such transformation is then described as

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{S}, \tag{1.1}$$

where $\mathbf{X}$ is the known $m \times N$ matrix of mixtures, m is the number of mixtures obtained by measurement on sensors and N is the number of available samples. Matrix $\mathbf{S}$ contains the samples of the original unknown sources and is of size $d \times N$, d is the number of unknown sources. A single row of $\mathbf{X}$ or $\mathbf{S}$ will be further denoted as $\mathbf{x}_i$ or $\mathbf{s}_i$, a single column as $\mathbf{X}(n)$ or $\mathbf{S}(n)$ and a single element as $x_i(n)$ or $s_i(n)$,

respectively. The samples stored in $\mathbf{S}$ may be real valued or complex valued. The unknown $m \times d$ matrix $\mathbf{A}$ represents parameters of the mixing model.

The linear instantaneous model is encountered in situations where the sampling frequency is small in comparison to the propagation speed of the signals. Here, all the signals reach the sensors at the same time. Moreover, this model takes into consideration no delays or echoes which may originate due to the multi-path propagation of the signals in the real environment. Methods based on (1.1) can be applied in biomedical signal analysis, where e.g. the signals measured by electro-encephalogram (EEG) or magneto-cardiogram (MEG) conform well to this model.

The **linear convolutive mixing model** is a generalization of (1.1), because the observed mixtures contain *delayed* original sources and their echoes. This problem is often encountered in acoustics, where it is called the *cocktail party problem*. It can be described by

$$x_i(n) = \sum_{k=1}^{d} \sum_{\tau=0}^{M_{ik}} a_{ik}(\tau) \cdot s_k(n-\tau), \tag{1.2}$$

where $x_1(n) \dots x_m(n)$ are the signals observed on microphones , $s_1(n) \dots s_m(n)$ are the original unknown sources. The unknown parameters $a_{ik}(\tau)$ represent the source-sensor impulse responses, i.e. impulse responses expressing the propagation of sound from the location of each source to each microphone. The convolutive model coincides with the instantaneous model, when $M_{ik} = 0$.

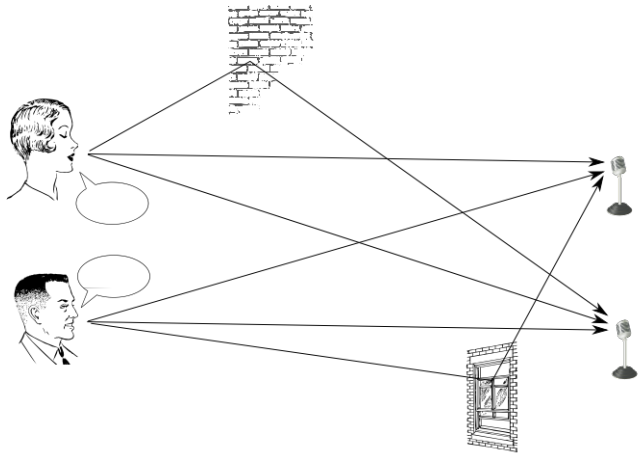


Figure 1.2: An example of speech signal propagation in the real environment. The observed mixtures should be modeled by convolutive model (1.2) due to presence of delays and echoes.

Both above mentioned models can be extended to **noisy models**. For example, the **linear instantaneous model with additive noise** is then given by

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{S} + \mathbf{N}, \tag{1.3}$$

where the $m \times N$ matrix $\mathbf{N}$ denotes the matrix of additive noise. The noise is usually modeled as a Gauss vector random variable with zero mean, which elements are mutually uncorrelated.

All methods described in this work are based on the noiseless models (both instantaneous and convolutive). However, scenarios with background noise are among presented experiments.

The linear models described above are approximative models, which are used in many BSS methods for their theoretical simplicity. In most of the real world situations this approximation is valid, but there are various situations where it does not hold. For instance, when the sensors used to observe the mixtures are nonlinear or the sensor saturation is encountered during the measurement. Here, the general model of nonlinear mixtures needs to be considered.

The **nonlinear mixing model** is given by

$$\mathbf{X} = \mathcal{A}(\mathbf{S}) + \mathbf{N}, \tag{1.4}$$

where $\mathcal{A}$ is a nonlinear transform. It can be either instantaneous, where input $\mathbf{S}(n)$ outputs $\mathbf{X}(n)$ or global, operating on the whole sequence $\mathbf{S}$. The BSS task based on this model cannot be solved without additional constraints. Most of the methods utilizing the nonlinear model assume mutual statistical independence of the sources in $\mathbf{S}$, but even with this assumption the problem does not have a unique solution. It can be achieved by incorporating prior knowledge about the source statistics or by constraining the mixing transform $\mathcal{A}$. As an example of a nonlinear transform, the so called *post-nonlinear mixtures* may be named. Here, the transform $\mathcal{A}$ consists of linear instantaneous or convolutive transform followed by some nonlinear transform, which operates component-wise.

The mixing transform may be constant during the observation of the mixtures or may change in time. This will be further denoted as **stationary mixtures** or **dynamic mixtures**, respectively. The separation of dynamic mixtures may be done with the block-by-block application of a method intended for stationary mixtures (further denoted as *Batch method*). The method may be modified more or less to respect the continuity of the (de)mixing process, and the outputting signals are synthesized from the separated signal blocks. In this thesis, such methods are referred to as *Online methods*.

1.1.2 Source models and BSS methods

In order to allow the separation, general assumptions are placed on the unknown sources. These assumptions determine techniques which can be exploited for source estimation. Nowadays, following source separation methods are used most frequently: Independent Component Analysis (ICA), Principal Component Analysis (PCA), Sparse Component Analysis (SCA) and Non-negative Matrix Factorization (NMF).

These methods are usually designed for the linear instantaneous model in (1.1). If the problem to be solved is convolutive (or nonlinear), it needs to be transformed

prior to the application of these methods. Such transformation will be described in Chapter 4.

In general, the methods seek to decompose a known matrix $\mathbf{X}$ in (1.1) into two new matrices. These matrices have a different meaning and properties, according to the technique used. This task is called the *inverse problem*.

Solving of the inverse problem consists in solving the under-determined set of equations in (1.1) because the whole right side of the equations is unknown. To overcome this indeterminateness, the methods utilized for decomposition introduce additional assumptions about $\mathbf{A}$ and $\mathbf{S}$.

Independent Component Analysis [2] assumes that the unknown sources are statistically independent random processes. The ICA techniques model the sources according to one of the two following principles.

The sources can be considered as *non-gaussian and temporally independent and identically distributed* random variables (iid). This leads to the utilization of higher (mostly 4^{th}) order statistics.

The sources can be modeled as *gaussian with certain temporal structure* (temporally dependent). This leads to the utilization of second order statistics.

ICA is used as the separator within BSS methods introduced in this thesis and will be discussed in more detail in Chapter 2.

Principal Component Analysis [3] is the simplest of the techniques mentioned above. It assumes a regular *orthonormal* matrix $\mathbf{A}$. The signals $\mathbf{s}_i$ are assumed to be uncorrelated and are called *Principal Components*. Moreover, the principal components have the highest variance among all orthogonal transforms of the mixtures $\mathbf{X}$. The columns of $\mathbf{A}$ are computed as eigenvectors of the covariance matrix of $\mathbf{X}$.

The uncorrelatedness is a necessary condition of independence, therefore the PCA is often used as preprocessing step prior to the utilization of ICA.

Sparse Component Analysis [4, 1] assumes that the unknown signals in $\mathbf{S}$ are *sparse* in a specifically chosen domain, i.e. contain only a few significant non-zero elements. An example of a sparse signal is speech, which is considered sparse in the time-frequency domain.

Most of the SCA methods assume that at each sampling instant n, a single source is significantly more active than the others. Let $T_i \subset \{1 \dots N\}$ denotes a set of sample indices where the ith source is the most active, then for all $n \in T_i$ holds $|s_i(n)| >> |s_j(n)|$ for $i \neq j$ and therefore

$$\mathbf{X}(n) \approx s_i(n)\mathbf{A}(i), \tag{1.5}$$

where $\mathbf{A}(i)$ is the unknown ith column of the mixing matrix.

The equation (1.5) suggests that the points $\mathbf{X}(n), n \in T_i$ are aligned along the straight line passing through the origin with the direction given by $\mathbf{A}(i)$. The mixing matrix estimate $\widehat{\mathbf{A}}$ can be determined e.g. by a clustering algorithm, where the mixture samples in $\mathbf{X}$ are grouped into d clusters and the direction of each is determined.

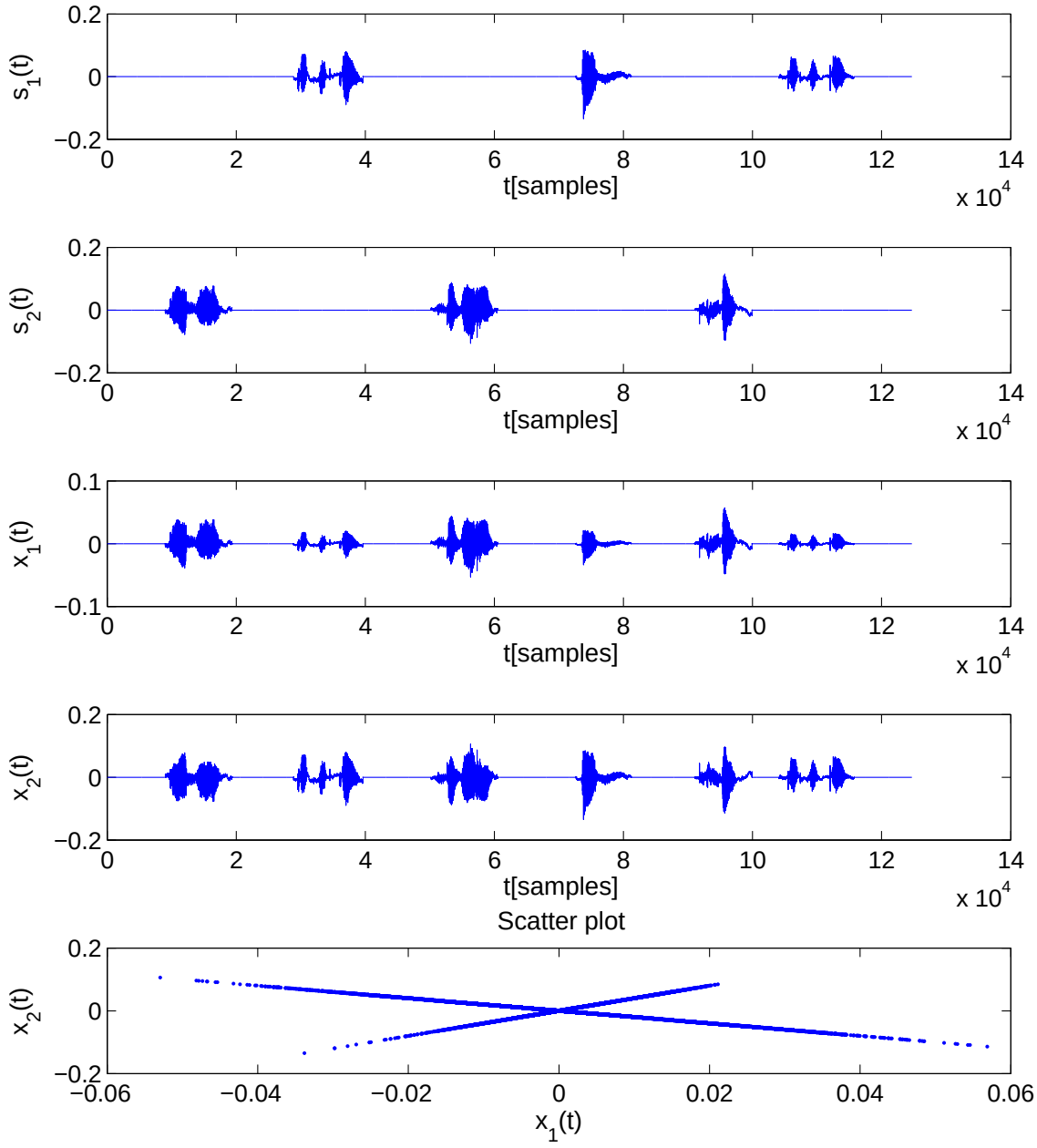


Figure 1.3: Geometric principle of sparse component analysis. Two speech signals with disjoint temporal support and the alignment of their mixtures along the columns of the mixing matrix.

Subsequently, the source estimates are computed. In the case that $d \geq m$, this is done via (pseudo)inverse of $\widehat{\mathbf{A}}$. However, SCA is able to perform the separation even in the case when $d < m$, i.e. the number of sources exceeds the number of the observed mixtures. This is one of the most appealing properties of SCA. Here, various binary masking techniques are often used.

Many of the real world sources are not sparse in the temporal domain. These sources need to be, prior to the estimation, represented in some sparse domain. The most often used representations are computed via the Short Time Fourier Transform, wavelet transform or techniques known as joint sparse decompositions [1].

Most SCA methods, therefore, start with the application of some sparse transform, which makes the supports of the sources as disjoint as possible. Subsequently, the mixing matrix $\widehat{\mathbf{A}}$ is estimated (e.g. via clustering) and the source estimates are determined in the sparse domain. Finally, an inverse transform is applied to these signals to obtain the sources in the original domain.

Non-negative Matrix Factorization [5, 1] assumes that the matrices in (1.1) contain *non-negative entries* only. Non-negative signals arise e.g. in image or speech processing, where the signals represent brightness values or magnitude spectra.

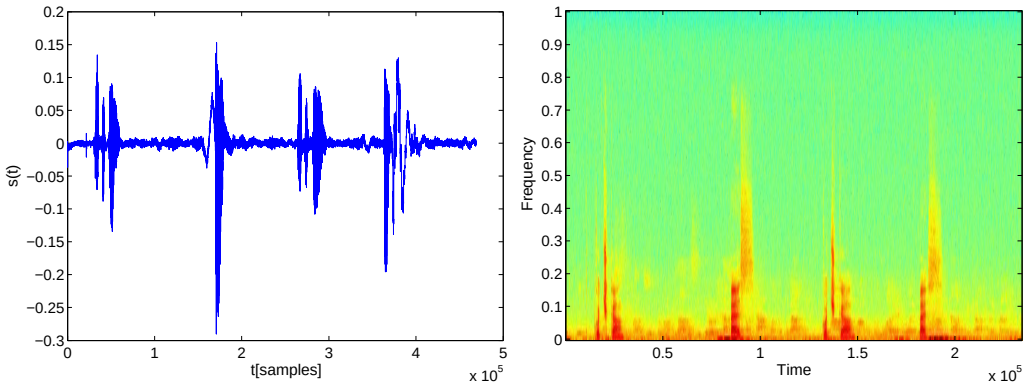


Figure 1.4: Speech signal and its spectrogram - a nonnegative signal

However, unlike the ICA factorization, the existence of NMF is not guaranteed in general. Often, an approximate factorization is searched, given by $\mathbf{X} \approx \mathbf{AS}$, which minimizes certain criterion. Such a criterion can be e.g. given by

$$J(\mathbf{A}, \mathbf{S}) = \|\mathbf{X} - \mathbf{AS}\|_F^2 , \tag{1.6}$$

The properties of factorized matrices depend strongly on the criterion used. Therefore, there are many different methods exploiting NMF, which are designed for various specific applications.

In practice, other signal properties are encountered, which can be used as prior information for the design of the separation algorithm. This leads to specific algorithms, which are simple and robust. In digital communications, signals are often *discrete valued* or *cyclostationary*. Methods for these types of signals were proposed in [6] and [7], respectively. Furthermore, the properties of signals with *distributions with bounded support* were exploited for the algorithm design in [8].

1.2 Motivation and goals of the thesis

The previous section briefly summarizes the objectives and methods for BASS. This section points out to some of the open questions in the discussed field and presents the motivation and the goals for the research contained in this thesis.

Firstly, the thesis deals with the separation of instantaneous mixtures via ICA. Although this problem has been studied since the early 1990s and is now well explored, there are still areas which deserve more research attention. Most of the ICA methods assume that the sources are either iid random variables or Gaussian random processes with a certain temporal structure. Although these models cover many types of signals, there exist very important sources which do not follow these assumptions. For example, speech is often modeled by Laplace distribution. Moreover, speech is stationary only in short intervals of 20-25 ms. This fact motivates the proposal of an ICA method, which is able to efficiently separate the non-stationary non-Gaussian sources. The algorithm design is discussed in Chapter 3.

Subsequently, the thesis deals with the blind separation of real-world audio sources. Nowadays, this task is widely studied and has considerable practical utilization. The main challenge here poses the separation in highly reverberant environments or the separation in presence of the background noise. Thanks to the plausible assumption of the independence of the audio sources, the problem can be solved via ICA. Recently, a method denoted as T-ABCD [9] has been proposed by Koldovský and Tichavský. It solves the separation task via ICA in the time-domain. The method exhibits an advantageous modular structure. This allows simple implementation of modifications which improve the performance of the method. The original T-ABCD assumes a perfect separation performed by ICA and reconstructs the sources accordingly. This is usually not true in practice. Chapter 5 discusses the modification, which aims at improvement of the reconstruction of the sources in the case of imperfect ICA separation.

The original T-ABCD assumes that the mixing system is fixed and does not change its inner parameters in time. This fact may not be entirely true in practice, e.g. due to the changing positions of the sources. This motivates the proposal of a new version of T-ABCD, further denoted as online, which is able to adapt to the changes in parameters of the mixing system. This subject is presented in Chapter 6.

Chapter 2

Linear Independent Component Analysis

2.1 The goal of ICA

The linear ICA is one of the methods solving the general BSS task. It is based on the instantaneous mixture model (1.1). ICA can be used for the separation of convolutive mixtures as well, provided that the convolutive model is transformed prior to the application of ICA (for details see Chapter 4).

ICA seeks to decompose the observed signals in $\mathbf{X}$ into components which are statistically independent. Based on the number of microphones m and the number of sources d, there are three distinct ICA cases. However, the true number of sources d is rarely known in practice and must be estimated.

The case when $d = m$ is denoted as the **determined mixtures**. Here, the matrix $\mathbf{A}$ is square and regular. The goal of ICA is to estimate the demixing matrix $\mathbf{W} = \mathbf{A}^{-1}$ so that the estimated signals

$$\widehat{\mathbf{S}} = \widehat{\mathbf{W}} \cdot \mathbf{X}, \tag{2.1}$$

are as independent as possible. The matrices $\widehat{\mathbf{S}}$ and $\widehat{\mathbf{W}}$ denote the estimates of $\mathbf{S}$ and $\mathbf{W}$, respectively. The rows of $\widehat{\mathbf{S}}$ are called *Independent Components* (ICs). In the case of determined mixtures, the identification of matrices $\mathbf{A}$ and $\mathbf{S}$ stands as equivalent tasks.

The case when $d > m$ is referred to as the **under-determined mixtures**. In this scenario, the identification of transform $\mathbf{A}$ and the estimation of independent components $\mathbf{S}$ are two distinct tasks. The demixing matrix $\mathbf{W}$ cannot be obtained by the inversion of $\mathbf{A}$. Hypothetically, if matrix $\mathbf{A}$ was available, there would still exist an infinite number of solutions. Another prior information is necessary in order to guarantee the uniqueness of the components. Such information could be e.g. the sparseness of the sources or that the samples in $\mathbf{S}$ are integer-valued. The most basic form of $\mathbf{S}$ computation is the pseudoinverse,

$$\widehat{\mathbf{S}} = \widehat{\mathbf{A}}^T(\widehat{\mathbf{A}}\widehat{\mathbf{A}}^T)^{-1}\mathbf{X}. \tag{2.2}$$

The case when $d < m$ is called the **over-determined mixtures**. In this case, the mixtures are usually reduced to a determined case in order to save the computational burden and to allow the application of algorithms designed for determined mixtures. The reduction may be performed e.g. via PCA.

The ICs $\widehat{\mathbf{S}}$ exhibit several ambiguities compared to true original sources $\mathbf{S}$. It is not possible to determine the variances of the components, since multiplying a component by an arbitrary positive number does not affect the mutual independence. In practice, a unit variance is often assumed, i.e. $\mathrm{E}\{\mathbf{s}_i^2\} = 1$.

Even with the assumption of a unit variance, the components are estimated up to a multiplicative constant ± 1 (real sources) or with an unknown phase (complex sources). Furthermore, the original order in $\mathbf{S}$ cannot be determined since any permutation of $\widehat{\mathbf{S}}$ still results in independent random variables.

Within this thesis, only the determined cases are considered.

2.2 Design of an ICA algorithm

In most cases, the ICA methods exhibit a similar structure and can be divided into three distinct parts.

Prior to the application of ICA on observed mixtures $\mathbf{X}$, some *preprocessing* technique may be exploited in order to ensure the ICA assumptions about the data. This includes e.g. the removal of the sample mean. Some methods assume that the mixtures are uncorrelated, then the orthogonalization transform called *Whitening* is performed. For details on whitening see Section 2.3.

The ICA algorithm itself often consists of the formulation of a certain *objective (contrast) function*, which is subsequently optimized via an *optimization technique*. The contrast function is used to quantify the mutual dependence of components and its global extrema matches the solution of the separation problem.

The properties of an ICA method as a whole depend on both elements mentioned above. Statistical properties, like asymptotical accuracy or robustness, depend on the choice of the contrast function. The algorithmical properties, like the speed of convergence or numerical stability, depend on the chosen optimization technique.

The ICA methods differ in the manner the components are estimated. Algorithms can estimate the components one after another, which is called *One-Unit*. The other way is to estimate the components all at once, which is called *Multi-Unit*. Methods can differ in the way they analyze input data. The mixtures can be available upon the execution of the method, such algorithms are called *Batch*. The other possibility is that the algorithm acquires the data in successive steps during the computation and the separating matrix $\widehat{\mathbf{W}}$ is continuously updated. These methods are denoted as *Online*.

2.3 Preprocessing

The preprocessing techniques ensure that the observed data $\mathbf{X}$ satisfy the assumptions of an ICA method about the input data.

Most of the ICA algorithms assume that the sources have a zero mean, i.e. $\mathrm{E}\{\mathbf{s}_i\} = 0$. Therefore, the mean value $\overline{\mathbf{X}}$ is subtracted from the data.

The mutual uncorelatedness of $\mathbf{x}_i$ is a necessary condition of the independence and the variance of the sources cannot be retrieved. Therefore, many ICA methods apply a technique called **Whitening** on $\mathbf{X}$. It substitutes the signals $\mathbf{X} - \overline{\mathbf{X}}$ by an orthogonal system of signals $\mathbf{Z}$. The signals $\mathbf{Z}$ span the same space as mixtures $\mathbf{X}$ and are mutually uncorrelated. Moreover, the orthogonality constraint simplifies the formulation of the separation criterion, as will be discussed further.

When the sources have unit variance, their covariance matrix is (due to their mutual independence) equal to the identity matrix, i.e. $\mathrm{E}[\mathbf{S}\mathbf{S}^T] = \mathbf{I}$. The signals $\mathbf{S}$ are said to be *spatially white*. Let $\mathbf{M}$ be a whitening transform matrix such that $\mathbf{Z} \triangleq \mathbf{M}\mathbf{X}$ is spatially white. Then the transformation $\mathbf{M}\mathbf{A}$ is a *rotation* because it relates two white vectors $\mathbf{S}$ and $\mathbf{Z} = \mathbf{M}\mathbf{A}\mathbf{S}$. This means that a separating matrix $\mathbf{W}$ can be factorized into $\mathbf{W} = \mathbf{W}_R\mathbf{M}$, where $\mathbf{M}$ is the decorrelation transform and $\mathbf{W}_R$ is a rotation matrix.

The whitening transform $\mathbf{M}$ is not uniquely given because when multiplied by any orthogonal matrix, the matrix product will still be a whitening matrix. The matrix $\mathbf{M}$ can be determined for example by PCA.

Spatial whiteness imposes $d(d+1)/2$ constraints and leaves $d(d-1)/2$ unknown rotation parameters to be determined by ICA. The orthogonality constraints are, therefore, very favorable from the algorithmic point of view, but may restrain the statistical efficiency of the algorithm.

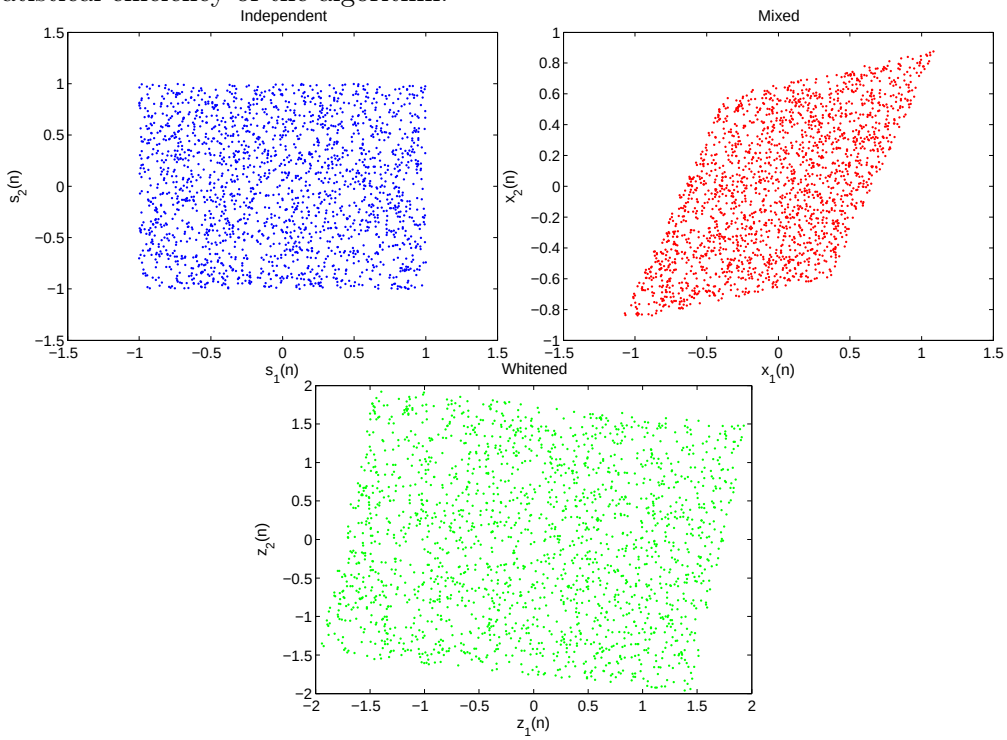


Figure 2.1: Scatter plot (joint distribution) of independent uniformly distributed signals $s_1(n)$ and $s_2(n)$, mixed signals $\mathbf{X} = \mathbf{A}\mathbf{S}$ and whitened signals $\mathbf{Z} = \mathbf{M}\mathbf{X}$.

2.4 Statistical principles and contrast functions

Many various ICA algorithms have been proposed since the topic was first introduced in the early 1990s. This section discusses the basic common underlying principles of ICA methods and the relations among them. A detailed overview of the discussed topic is covered e.g. [1, 2, 10].

Many ICA algorithms assume the unknown sources in a form of i.i.d. random variables. Here, the separation is carried out through the optimization of a criterion, which is based on one of the following, closely related, principles: Maximum Likelihood Estimation, Mutual Information or Entropy. These methods rely on higher order statistics.

Other methods assume a certain time-structure of the unknown sources. Most often, the components are modeled as Gauss random variables with a changing variance or wide sense stationary random processes with distinct spectra. These methods exploit second order statistics.

Throughout this section the determined noiseless mixtures are considered.

2.4.1 ICA methods assuming unknown sources in the form of iid random variables

This section describes principles of the ICA methods based on the assumption of the iid model of the source signals. The methods also assume that not more than one of the sources has Gaussian distribution. It stems from the fact that any orthogonal transform applied on two independent Gaussian distributions results in two independent Gaussian variables. Only Gaussian distribution exhibits this property. Hence, the orthogonal rotation matrix $\mathbf{W}_R$ (see Section 2.3) cannot be uniquely estimated for Gaussian variables.

A very popular approach for estimating the ICs is the **Maximum Likelihood Estimation** (MLE). This approach has a significant advantage when the pdfs of the unknown sources are known. Then the methods based on this principle are asymptotically efficient, i.e. they give asymptotically the best estimate which an unbiased estimator can provide.

The likelihood function for the ICA noise-free model is derived in the following manner. The probability density $p_{\mathbf{X}}$ of the mixture vector can be formulated as

$$p_{\mathbf{X}}(\mathbf{X}) = \left|\det \mathbf{A}^{-1}\right| p_{\mathbf{S}}(\mathbf{A}^{-1}\mathbf{X}) = \left|\det \mathbf{A}^{-1}\right| \prod_i p_{\mathbf{s}_i}((\mathbf{A}^{-1}\mathbf{X})_i). \tag{2.3}$$

Assuming that N independent observations of $\mathbf{X}$ are available, the likelihood function $L(\widehat{\mathbf{S}}|\widehat{\mathbf{A}})$ can be formulated for the estimated sources $\widehat{\mathbf{S}} = \widehat{\mathbf{A}}^{-1}\mathbf{X}$ according to

$$L_{\widehat{\mathbf{A}}^{-1}\mathbf{X}}(\widehat{\mathbf{S}}|\widehat{\mathbf{A}}) = \prod_{n=1}^{N} \left|\det \mathbf{A}^{-1}\widehat{\mathbf{A}}\right| \prod_{i=1}^{d} p_{\mathbf{s}_i}((\mathbf{A}^{-1}\widehat{\mathbf{A}})_i \widehat{\mathbf{S}}(n)). \tag{2.4}$$

Finally, the logarithm of the likelihood function is given as

$$\frac{1}{N}\log L_{\widehat{\mathbf{A}}^{-1}\mathbf{X}}(\widehat{\mathbf{S}}|\widehat{\mathbf{A}}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{d}\log p_{\mathbf{s}_i}((\mathbf{A}^{-1}\widehat{\mathbf{A}})_i\widehat{\mathbf{S}}(n)) + \log\left|\det \mathbf{A}^{-1}\widehat{\mathbf{A}}\right|. \quad (2.5)$$

To simplify the notation, the likelihood function can be written as

$$\frac{1}{N}\log L_{\widehat{\mathbf{A}}^{-1}\mathbf{X}}(\widehat{\mathbf{S}}|\widehat{\mathbf{A}}) = \widehat{\mathrm{E}}\{\sum_{i=1}^{d}\log p_{\mathbf{s}_i}((\mathbf{A}^{-1}\widehat{\mathbf{A}})_i\widehat{\mathbf{S}}(n))\} + \log\left|\det \mathbf{A}^{-1}\widehat{\mathbf{A}}\right|, \quad (2.6)$$

where $\widehat{\mathrm{E}}$ represents the sample mean operator. A global maxima of this function with respect to the elements of $\widehat{\mathbf{A}}$ gives the ML estimate of the ICs.

The MLE is closely related to the Kullback-Leibler divergence, which is for two probability density functions $p(\mathbf{y})$ a $q(\mathbf{y})$ on $\mathbb{R}^d$ defined as

$$K(p|q) \stackrel{\triangle}{=} \int_{\mathbf{y}} p(\mathbf{y})\log\frac{p(\mathbf{y})}{q(\mathbf{y})}d\mathbf{y}. \quad (2.7)$$

The divergence between densities of two random vectors $\mathbf{y}$ and $\mathbf{z}$ is denoted as $K(\mathbf{y}|\mathbf{z})$. An important property of $K(\mathbf{y}|\mathbf{z})$ is that $K(\mathbf{y}|\mathbf{z}) \geq 0$. The divergence is zero if and only if the two random vectors have the same distribution. The divergence Kullback-Leibler is not symmetric, though $(K(\mathbf{y}|\mathbf{z}) \neq K(\mathbf{z}|\mathbf{y}))$.

Lets assume that the number of available samples tends to infinity, i.e. $N \to \infty$, then (2.6) equals

$$\frac{1}{N}\log L_{\widehat{\mathbf{A}}^{-1}\mathbf{X}}(\widehat{\mathbf{S}}|\widehat{\mathbf{A}}) \stackrel{N\to\infty}{\to} -K(\widehat{\mathbf{A}}^{-1}\mathbf{X}|\mathbf{S}) + const. \quad (2.8)$$

This shows that (up to a constant) the log-likelihood $\log L(\widehat{\mathbf{S}}|\widehat{\mathbf{A}})$ is an estimate of $-K(\widehat{\mathbf{A}}^{-1}\mathbf{X}|\mathbf{S})$ and that it minimizes the divergence between the original signals $\mathbf{S}$ and the estimated signals $\widehat{\mathbf{A}}^{-1}\mathbf{X}$. The MLE is, therefore, associated with the *contrast function*

$$J_{ML}(\widehat{\mathbf{S}}) = K(\widehat{\mathbf{S}}|\mathbf{S}), \quad (2.9)$$

which should be minimized with respect to the model parameters $\widehat{\mathbf{A}}$.

Another popular principle serving as a basis for ICA contrast functions is the **Mutual Information** of entries in $\widehat{\mathbf{A}}^{-1}\mathbf{X}$.

The MLE approach is based on the fixed hypothesis about the distribution of the sources. Here, a problem may arise when the hypothesized distribution differ greatly from the true one. Therefore, the divergence $K(\widehat{\mathbf{S}}|\mathbf{S})$ should be minimized with respect to $\widehat{\mathbf{A}}$ (via the distribution of $\widehat{\mathbf{A}}^{-1}\mathbf{X}$) as well as with respect to the distribution of $\mathbf{S}$ [1].

Let us denote as $\underline{\mathbf{S}}$ a random vector with independent entries which are distributed in the same way as entries of $\widehat{\mathbf{S}}$. Then for any matrix $\mathbf{S}$ with independent entries it holds that

$$K(\widehat{\mathbf{S}}|\mathbf{S}) = K(\widehat{\mathbf{S}}|\underline{\mathbf{S}}) + K(\underline{\mathbf{S}}|\mathbf{S}). \quad (2.10)$$

The equation (2.10) shows that $K(\widehat{\mathbf{S}}|\mathbf{S})$ is minimized with respect to $\mathbf{S}$ by minimizing $K(\underline{\mathbf{S}}|\mathbf{S})$. This can be achieved by assuming $(\mathbf{S} = \underline{\mathbf{S}}) \Rightarrow K(\underline{\mathbf{S}}|\mathbf{S}) = 0$ since $K(\widehat{\mathbf{S}}|\underline{\mathbf{S}})$ does not depend on $\mathbf{S}$.

Now the minimization of the divergence $K(\widehat{\mathbf{S}}|\mathbf{S})$ with respect to $\widehat{\mathbf{S}}$ leads to the minimization of the contrast function

$$J_{MI}(\widehat{\mathbf{S}}) \stackrel{\triangle}{=} K(\widehat{\mathbf{S}}|\underline{\mathbf{S}}). \tag{2.11}$$

The Kullback-Leibler divergence $K(\widehat{\mathbf{S}}|\underline{\mathbf{S}})$ between a distribution and the closest distribution with independent entries is called the *mutual information* between entries of $\widehat{\mathbf{S}}$. It is always non-negative and is zero if and only if $\widehat{\mathbf{S}}$ is distributed as $\underline{\mathbf{S}}$. By definition of $\underline{\mathbf{S}}$, this requires the mutual independence of entries in $\widehat{\mathbf{S}}$. The function $J_{MI}(\widehat{\mathbf{S}})$, therefore, measures the independence between the entries of $\mathbf{A}^{-1}\mathbf{X}$ and is thus a valid ICA contrast function.

The contrasts J_{ML} in (2.9) and J_{MI} in (2.11) are related via

$$J_{ML}(\widehat{\mathbf{S}}) = J_{MI}(\widehat{\mathbf{S}}) + \sum_{i=1}^{d} K(\widehat{\mathbf{s}}_i|\mathbf{s}_i), \tag{2.12}$$

where $K(\underline{\mathbf{S}}|\mathbf{S}) = \sum_{i=1}^{d} K(\widehat{\mathbf{s}}_i|\mathbf{s}_i)$ because of the independence of entries in $\underline{\mathbf{S}}$ and $\mathbf{S}$. This shows that the maximization of $J_{ML}(\widehat{\mathbf{S}})$ with fixed assumptions about the sources consists of two terms: the first one is the mutual information as a measure of independence and the other one measures divergence of the marginal distribution of the estimates from the assumed distributions of the sources.

The contrast J_{MI} can be simplified under the **Orthogonality Constraint** $\mathrm{E}\widehat{\mathbf{S}}\widehat{\mathbf{S}} = I$ discussed in Section 2.3. It holds that J_{MI} is equal up to a constant term to the sum of *Shanon entropies* of each output [11]. Therefore, the minimization of the mutual information between elements of $\widehat{\mathbf{S}}$ is equivalent to the minimization of the sum of entropies of the elements, i.e.

$$J_{ENT}(\widehat{\mathbf{S}}) \stackrel{\triangle}{=} \sum_{i} H(\widehat{\mathbf{s}}_i), \tag{2.13}$$

where $H(y)$ is a Shanon entropy of a random variable y with pdf $p(y)$ given as

$$H(y) = -\int p(x) \ln p(x) dx. \tag{2.14}$$

The mixing of elements of $\mathbf{S}$ increases their entropies. The original sources, therefore, may be found as signals with minimal marginal entropies.

Moreover, $-H(\widehat{\mathbf{s}}_i)$ is (up to a constant term) the Kullback-Leibler divergence between the distribution of $\widehat{\mathbf{s}}_i$ and the zero-mean unit variance Gaussian distribution [10]. This can be explained by the *Central Limit Theorem*, which states that a sum of random variables will have distribution closer to Gaussian than any of the original random variables. In order to separate the random variables, the separation technique should drive the distribution away from normality. The ICA methods built upon this principle are often called *non-gaussianity based*.

All the contrasts $J_{MI}(\widehat{\mathbf{S}})$, $J_{ML}(\widehat{\mathbf{S}})$ and $J_{ENT}(\widehat{\mathbf{S}})$ require the knowledge of the distributions of the unknown sources. These distributions may be known a priori or at least some of their features may be known, like moments or bounded support. In many cases though, the densities are completely unknown and must be estimated in a certain manner.

The estimation of a probability distribution is essentially a nonparametric task. An ICA method using a nonparametric density estimation was proposed e.g. by Boscolo et al. in [12], where the authors use a kernel technique [13] to estimate the pdfs and their derivatives.

The nonparametric density estimation is known to be a difficult and computationally demanding problem, however. Many methods avoid it by the approximation of the unknown distribution by a family of densities that are characterized by a limited number of parameters. An algorithm known as Infomax, based on maximum likelihood approach and using parametric density approximation, was proposed by Bell and Sejnowski in [14]. A fixed-point algorithm called FastICA, based on approximation of entropy, was introduced by Hyvärinen and Oja in [15].

Another approach how to deal with unknown source distributions represents the utilization of higher order statistics. It can be used for the definition of contrast functions, which are approximations to those derived from the maximum likelihood approach. High order information is most simply expressed by cumulants (see [2]). The minimization of mutual information based on the approximation via cumulants was proposed by Common in [11]. The properties of cumulants and cumulant based tensors were utilized for the contrast function formulation by Cardoso, who proposed the algorithms FOBI (Fourth Order Blind Identification) [16] and well known JADE (Joint Approximate Diagonalization of Eigenmatrices) [17].

2.4.2 ICA methods assuming sources with a time structure

The methods discussed in the previous section assume unknown sources to be independent and identically distributed random variables. The key assumption in this scenario is that the distributions of the sources are non-gaussian. Methods based on the second order statistics (SOS) cannot separate such sources. Here, the SOS can be exploited in preprocessing, but it cannot determine the orthogonal rotation matrix $\mathbf{W}_R$.

In practice, many sources exhibit some temporal structure. The separation of such sources can be done by the techniques presented in the previous section. However, then the temporal information is not used directly.

When the second order characteristics of the sources exhibit sufficient diversity, the separation can be based on SOS alone, ignoring the higher order characteristics. Moreover, the SOS based ICA techniques are able to separate Gaussian sources, which is not possible for the methods described in Section 2.4.1.

The SOS based methods can be basically divided, based on model of the source signals, into two groups. One group consists of methods assuming the sources in the form of wide-sense stationary processes, the other group consists of methods

assuming the sources to be Gaussian random variables with a changing variance.

In this section, the ICA model in (1.1) becomes

$$\mathbf{X}(t) = \mathbf{A}\mathbf{S}(t), \tag{2.15}$$

where $t = 1 \dots N$ has the meaning of time. The equation (2.15) emphasizes the time index because the time structure of the sources is important for SOS based methods. Unlike the iid case, where an arbitrary permutation can be applied to the samples (identical for all mixtures) before the application of an ICA method.

Spectral diversity based methods assume sources to be Gaussian wide-sense stationary processes (WSS). The Gaussian WSS process is fully characterized by its correlation function, which depends only on the time difference between two sampling instants, not on the sampling instant itself.

In the case of an existing time structure, the correlation function is generally non zero for all time instants $\tau \geq 0$. In the case of iid sources, the only non zero value is for $\tau = 0$. Correlations between two and more different signals can be grouped together in time-lagged covariance matrix $\mathbf{R}_{\mathbf{X}}[\tau]$ given by

$$\mathbf{R}_{\mathbf{X}}[\tau] = \mathrm{E}\{\mathbf{X}(t)\mathbf{X}(t-\tau)^T\}, \tag{2.16}$$

where $\mathbf{X}(t-\tau)$ denotes mixtures, which are delayed by τ samples.

The sources $\mathbf{S}$ are mutually independent and all their lagged covariance matrices $\mathbf{R}_{\mathbf{S}}[\tau]$ are diagonal. Therefore, the ICA methods based on SOS often take on the form of the simultaneous diagonalization of multiple matrices $\mathbf{R}_{\mathbf{X}}[\tau]$. Diagonalization of a single matrix $\mathbf{R}_{\mathbf{X}}[\tau]$ is not sufficient because it makes the mixtures uncorrelated but not independent. Moreover, the separation is possible if and only if the sources have distinct spectra (for details see e.g. [1, 18]).

The simultaneous diagonalization of square matrices is performed using a technique called the *approximate joint diagonalization* (AJD), which will be described in more detail in Section 2.5.

The non-stationarity based methods assume the unknown sources in the form of Gaussian random variable with a changing variance. The signals are now described by the covariance matrix at different time instants τ, i.e.

$$\mathbf{R}_{\mathbf{XX}}[\tau] = \mathrm{E}\{\mathbf{X}(t-\tau)\mathbf{X}(t-\tau)^T\}. \tag{2.17}$$

In practice, the mixtures are segmented into blocks and the estimates of $\mathbf{R}_{\mathbf{XX}}[\tau]$ are computed in each block separately.

The non-stationarity based methods estimate the demixing matrix $\mathbf{W}$ by forcing the mixtures to be uncorrelated at every moment. Such a way of separation is possible because (due to nonstationarity) the $\mathbf{R}_{\mathbf{XX}}[\tau]$ are time dependent. Hence, the noncorrelatedness at each time instant is a stronger condition than simple whitening. The separation is achievable if and only if the sources have distinct variances in at least one time instant τ (see [2]).

The following ICA algorithms are based on the principles described in this section. A simple AJD based ICA algorithm called AMUSE, jointly diagonalizing two matrices $\mathbf{R}_{\mathbf{X}}[\tau]$, was proposed by Tong et al. in [19].

An extension of AMUSE, an algorithm called SOBI (Second Order Blind Identification) based on the diagonalization of several matrices $\mathbf{R_X}[\tau]$, was introduced by Belouchrani et al. in [20].

The AJD of several matrices $\mathbf{R_{XX}}[\tau]$ was used for design of the BGL algorithm (Block Gaussian Likelihood) by Pham and Cardoso in [21].

2.4.3 Hybrid mixtures and generalized source models

Recently, separating methods for instantaneous mixtures based on the **combination of source models** have been proposed. These techniques are based on the fact that in real-world mixtures it may happen that no single model offers correct representation of all sources.

For example, in biomedicine both spectral-diversity-based and non-gaussianity-based methods are studied. The combination of these two models was demonstrated on EEG data by Gorodnitsky and Belouchrani in [22].

A technique which separates mixtures of iid sources and Gaussian AR processes, was proposed by Tichavsky et al. in [23]. The introduced technique combines the algorithms EFICA [24] and WASOBI [25].

An ad-hoc algorithm unifying all three models was proposed by Hyvärinen in paper [26].

Other methods generalize the basic models in some way, for example by adding the **non-stationarity of the data**. These techniques deal with situations, when the sources are well described as iid random variables or Gaussian AR processes, however the parameters describing the signals gradually change. The separation such sources usually proceeds via segmentation into blocks, where the signal characteristics can be considered as constant.

An asymptotically efficient algorithm called BARBI (Block AutoRegressive Blind Identification), which is designed for sources modeled as block-wise stationary Gaussian AR processes, was proposed by Tichavsky et al. in [27].

An asymptotically efficient method separating block-wise stationary non-gaussian sources called Block EFICA was proposed by Koldovský and the author of the thesis in [28].

2.5 Optimization techniques

In general, the estimation of the separation matrix $\mathbf{W}$ cannot be solved in the closed form, i.e. written as a function of the sample set which outputs the entries of $\mathbf{W}$. Instead, the solution is based on the numerical optimization of the cost functions introduced in Section 2.4.1 whose extrema provide the solutions. This section briefly introduces some typical optimization algorithms and their properties. At the end of the section, the Approximate Joint Diagonalization (AJD) of regular matrices is discussed. This technique is often used in ICA techniques based on SOS.

The Gradient Method is the basic numerical technique for the minimization

of objective function $J(\cdot)$. It is given by

$$\Delta\mathbf{w} = -\alpha\frac{\partial J(\mathbf{w})}{\partial\mathbf{w}}, \tag{2.18}$$

where $\mathbf{w}$ is the vector of searched parameters (a row of the demixing matrix from (1.1)). The gradient method has the advantage of low computational demands. Its disadvantage consists in the choice of parameter α, which influences the length of the minimization step. When the selected step length is unsuitable, the algorithm converges slowly or diverges. The gradient method has a linear convergence rate.

The Newton's method is a well known method for finding the roots of a set of equations. It can be used to find local minima or maxima of a function. In this case, it solves the equation where the derivative of a function is put equal to zero. The Newton's method is given by

$$\Delta\mathbf{w} = -\left(\frac{\partial^2 J(\mathbf{w})}{\partial\mathbf{w}^2}\right)^{-1}\cdot\frac{\partial J(\mathbf{w})}{\partial\mathbf{w}}. \tag{2.19}$$

This method is one of the most effective numerical techniques for the minimization of the given criterion, which provides a fast quadratic convergence rate. Its disadvantage consists in high computational demands as it requires the computation of Hessian matrix inverse in each iteration.

In the context of SOS based ICA methods, **the Approximate Joint Diagonalization** (AJD) is often used for the estimation of the demixing matrix $\mathbf{W}$. AJD attempts to fit a set of real symmetric square matrices $\mathbf{R_X}[r], r = 0\ldots R-1$ of the size $d\times d$ by structured matrices of the form

$$\mathbf{R_X}[r] = \mathbf{W}\mathbf{R_S}[r]\mathbf{W}^T, \tag{2.20}$$

where $\mathbf{R_S}[r]$ represents the unknown diagonal matrices. Usually, $\mathbf{W}$ cannot be expressed exactly according to (2.20). Instead, it is estimated approximately through minimization of a certain criterion, e.g.

$$J_{LS}(\mathbf{W}) = \sum_{i=0}^{T-1}\left\|\mathrm{off}(\mathbf{W}^{-1}\mathbf{R_X}[r]\mathbf{W}^{-1\,T})\right\|_F^2, \tag{2.21}$$

where off is an operator which sets the diagonal elements of a matrix to zero and $\|.\|_F$ denotes the Frobenius matrix norm.

2.6 Objective evaluation of ICA algorithms

The objective evaluation of ICA algorithms and their mutual comparison is performed via Signal to Interference Ratio (SIR) or Interference to Signal Ratio (ISR). Both criteria are defined through the elements of *Gain matrix* given as

$$\mathbf{G} = \widehat{\mathbf{W}}\mathbf{A}, \tag{2.22}$$

which can be computed if the mixing matrix $\mathbf{A}$ is known. If the components are estimated exactly and source signals have a unit variance, then the matrix $\mathbf{G}$ is a matrix of zeros, with an exactly single non-zero element in every row and column, which is equal to ± 1. SIR and ISR criteria are defined, provided that the sources have a unit variance $\mathrm{E}\{\mathbf{s}_i^2\} = 1$ and the order of the estimates is correct, as follows

$$\mathrm{SIR}_k = \frac{\mathbf{G}_{kk}^2}{\sum_{i=1, i\neq k}^{d} \mathbf{G}_{ki}^2} \tag{2.23}$$

and

$$\mathrm{ISR}_k = \frac{\sum_{i=1, i\neq k}^{d} \mathbf{G}_{ki}^2}{\mathbf{G}_{kk}^2}. \tag{2.24}$$

The values of SIR and ISR are usually given in the logarithmic scale. In the case of perfect separation, the value of SIR $= +\infty = +\infty$dB and the value of ISR $= 0 = -\infty$dB.

2.7 Efficiency of ICA algorithms

This section addresses the statistical concept of an efficient estimator and subsequently, the efficiency of ICA algorithms is discussed.

Suppose we have a parametric model with unknown parameters $\boldsymbol{\theta} = \theta_1 \dots \theta_M$ and data sampled from this model $\mathbf{x} = x(1) \dots x(N)$. Let $T(\mathbf{x})$ be an estimator for the unknown parameters $\boldsymbol{\theta}$. The estimator is said to be *unbiased* if $\mathrm{E}\{\widehat{\boldsymbol{\theta}}\} = \boldsymbol{\theta}$, where $T(\mathbf{x}) = \widehat{\boldsymbol{\theta}}$.

Let $\mathbf{C}_{\widehat{\boldsymbol{\theta}}}$ be the covariance matrix of errors of the estimator $T(\mathbf{x})$ given by $\mathbf{C}_{\widehat{\boldsymbol{\theta}}} = \mathrm{E}\{(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})^{\mathbf{T}}\}$. Let $T(\mathbf{x})$ be an unbiased estimator. The Cramér-Rao inequality states that the entries of the matrix $\mathbf{C}_{\widehat{\boldsymbol{\theta}}}$ are bounded from bellow as

$$\mathbf{C}_{\widehat{\boldsymbol{\theta}}} \geq (\mathbf{F}_{\boldsymbol{\theta}})^{-\mathbf{1}}, \tag{2.25}$$

where $\mathbf{F}_{\boldsymbol{\theta}}$ is the Fisher information matrix of the model with parameters $\boldsymbol{\theta}$ given by

$$\mathbf{F}_{\boldsymbol{\theta}} = \mathrm{E}\left[\left(\frac{\partial}{\partial \boldsymbol{\theta}} \ln p(\mathbf{x}|\boldsymbol{\theta})\right)\left(\frac{\partial}{\partial \boldsymbol{\theta}} \ln p(\mathbf{x}|\boldsymbol{\theta})\right)^{T}\right]. \tag{2.26}$$

The term $p(\mathbf{x}|\boldsymbol{\theta})$ denotes the joint distribution of the measurements $\mathbf{x}$ with the parameter vector $\boldsymbol{\theta}$. The matrix inequality in (2.25) means that the matrix $\mathbf{C}_{\widehat{\boldsymbol{\theta}}} - (\mathbf{F}_{\boldsymbol{\theta}})^{-\mathbf{1}}$ is positive semidefinite.

An unbiased estimator $T(\mathbf{x})$ is said to be a *finite sample efficient estimator* if it achieves the lower bound in the Cramér-Rao inequality (2.25) for all $\theta_i \in \boldsymbol{\theta}$ and a finite sample number N. The estimator is called *asymptotically efficient* if it achieves the equality with $N \to \infty$.

In the context of linear instantaneous ICA with iid sources, the error covariance matrix can be expressed for the terms of the gain matrix $\mathbf{G}$ given as $\mathbf{G} = \widehat{\mathbf{W}}\mathbf{A}$. It

reflects the mean value of residual interference among estimated sources $\widehat{\mathbf{S}}$. In [29] it is proven that

$$\mathrm{var}(\mathbf{G}_{ij}) \geq \mathrm{CRB}(\mathbf{G}_{ij}) = \frac{1}{N}\frac{\kappa_j}{\kappa_i\kappa_j - 1} \tag{2.27}$$

where $i, j = 1 \ldots d$, $\mathrm{var}(\mathbf{G}_{ij})$ is the variance of the entries of $\mathbf{G}$, CRB denotes the Cramér-Rao Bound and κ_i is given by definition as

$$\kappa_i \stackrel{\triangle}{=} \mathrm{E}[\psi_i^2(\cdot)]. \tag{2.28}$$

The function ψ_i denotes the *score function* of the ith source, i.e. $\psi_i(x) = -\frac{p_i'(x)}{p_i(x)}$.

An asymptotically efficient version of the famous algorithm FastICA [15] was proposed by Koldovský et al. in [24].

The CRB for the case when the sources are modeled as Gaussian AR processes was derived for the variance of estimation errors $(\widehat{\mathbf{W}} - \mathbf{W})$ by Dégerine et al. in [30]. It was extended and reformulated as a function of the gain matrix $\mathbf{G}$ by Doron et al. in [31].

An asymptotically efficient version of the SOBI algorithm [20] called WASOBI (Weights Adjusted SOBI) was proposed by Yeredor in [25]. The original WASOBI technique was very computationally demanding for a higher number of sources (beyond five). Tichavský et al. in [32] proposed a computationally affordable scheme that can be employed in WASOBI, reducing its computational burden.

Other efficient ICA algorithms designed for the separation of non-stationary sources were mentioned already in Section 2.4.3. The algorithm called BARBI separating block-wise stationary AR processes was proposed by Tichavsky et al. in [27]. A method for separation of non-stationary non-gaussian sources called Block EFICA was proposed by Koldovský and the author of this thesis in [28].

2.8 The FastICA algorithm

The FastICA algorithm is one of the most popular ICA techniques, mainly due to its speed, accuracy and relatively simple implementation. It was proposed by Appo Hyvärinen and Erkki Oja in [15] and later extended in [33]. This section covers a brief description of FastICA, because it is the basis of the algorithms EFICA [24] and Block EFICA [28], which are discussed in later chapters.

FastICA starts with a preprocessing step called whitening (see Section 2.3). It consists in the removal of the sample mean and in the decorrelation of $\mathbf{X}$ in (1.1). It is given by

$$\mathbf{Z} = \widehat{\mathbf{R}}^{-1/2}(\mathbf{X} - \bar{\mathbf{X}}), \tag{2.29}$$

where $\widehat{\mathbf{R}}$ is the estimate of the mixture covariance matrix, such that $\widehat{\mathbf{R}} = (\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})^T/N$, $\bar{\mathbf{X}}$ is a sample mean of $\mathbf{X}$ and $\mathbf{Z}$ is the matrix of transformed orthogonal mixtures. The whitening matrix $\widehat{\mathbf{R}}^{-1/2}$ is computed via eigenvalue decomposition $\widehat{\mathbf{R}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is a unitary matrix of eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues. It is given as

$$\widehat{\mathbf{R}}^{-1/2} = \mathbf{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T. \tag{2.30}$$

FastICA's main part is iterative. A fixed-point iteration scheme is used to estimate the rows of the demixing matrix $\mathbf{W}$. It was introduced in two variants: *one-unit* or *symmetric* technique. In the **one-unit approach**, the components are estimated sequentially via finding the extrema of a contrast function given by

$$J(\mathbf{w}) = \mathrm{E}\{G(\mathbf{w}^T\mathbf{Z})\}, \quad \|\mathbf{w}\| = 1, \tag{2.31}$$

where E is the expected value operator and $\mathbf{w}$ is one row of the demixing matrix $\mathbf{W}$. The function $G(\cdot)$ represents an approximation of entropy (2.13) and is applied element-wise on $\mathbf{w}^T\mathbf{Z}$.

The common choices of $G(\cdot)$ include functions [1]

$$\begin{aligned} G_1(x) &= x^4 \\ G_2(x) &= \ln\cosh x \\ G_3(x) &= -\exp(-x^2/2) \end{aligned} \tag{2.32}$$

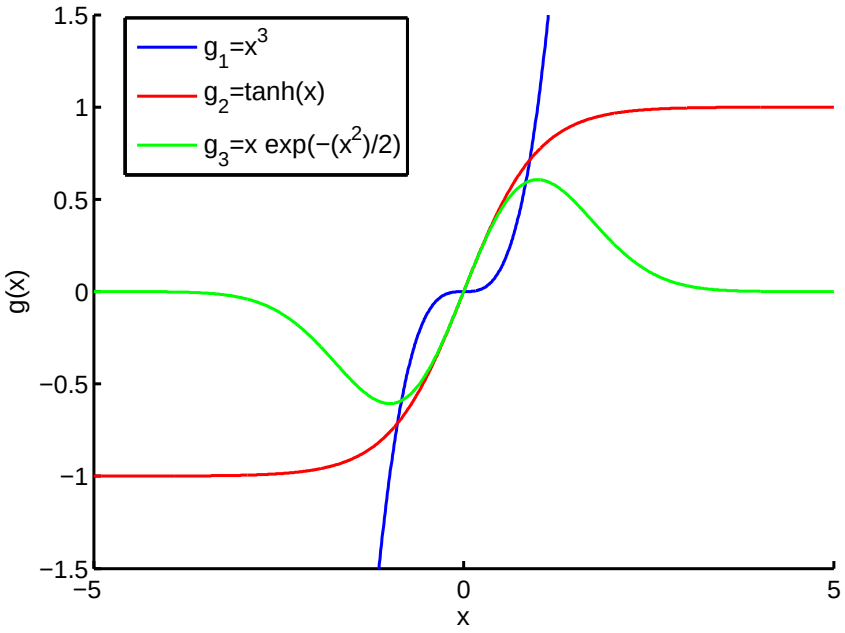


Figure 2.2: Derivatives of FastICA contrast functions

The one-unit iteration process starts with initialization by a random vector $\mathbf{w}$ and continues according to

$$\begin{aligned} \mathbf{w} &\leftarrow \hat{\mathrm{E}}\{\mathbf{Z}g(\mathbf{w}^T\mathbf{Z})\} - \hat{\mathrm{E}}\{g'(\mathbf{w}^T\mathbf{Z})\mathbf{w}\} \\ \mathbf{w} &\leftarrow \mathbf{w}/\left\|\mathbf{w}\right\|, \end{aligned} \tag{2.33}$$

[1] If the densities of all sources $p_i(x)$ were known, the function $G^{opt}(x) = -\log(p_i(x))$ would be optimal, because $-\mathrm{E}\{\log(p_i(x))\}$ is the entropy itself. Therefore, the log-densities of some known important probability distributions are often utilized as $G(x)$.

until convergence is achieved. The functions $g(\cdot), g'(\cdot)$ denote the first and the second derivative of $G(\cdot)$ respectively. The convergence is reached when

$$\mathbf{w}_{(k-1)}^T \mathbf{w}_{(k)} \approx \pm 1, \tag{2.34}$$

where $\mathbf{w}_{(k)}$ denotes the estimate of $\mathbf{w}$ computed in the kth one-unit iteration. Vectors $\mathbf{w}_{(k-1)}, \mathbf{w}_{(k)}$ need not point in the same direction, because $\mathbf{w}$ and $-\mathbf{w}$ define the separation of the same component multiplied by -1.

During the one-unit iterations, it is not known in advance which one of the components is being estimated. This depends largely on the initialization. A Gaussian signal cannot be retrieved by one-unit iterations. In practice, multiple convergence into a single solution is prevented by enforcing the orthogonality on the estimated components.

One manner of introducing the orthogonality constraint represents **the deflation approach**. The components are estimated in sequence and the currently estimated signal is enforced to be orthogonal to all previously estimated ones. This is performed via the one-unit iteration followed by e.g. the Gramm-Smidt orthogonalization [2]. Here, the orthogonal projections of all previously estimated vectors $\mathbf{w}$ are subtracted from the currently estimated vector $\mathbf{w}$. The order of the components still depends on the initialization. A major drawback of the deflation approach consists in the cumulation of estimation errors from previously estimated components.

The symmetric approach estimates all the rows of $\mathbf{W}$ in parallel. One-unit iteration is performed on each row and subsequently symmetric orthogonalization of the rows is performed. The symmetric iterations can be written as

$$\begin{aligned} \mathbf{W} &\leftarrow g(\mathbf{WZ})\mathbf{Z}^T - \mathrm{diag}\{g'(\mathbf{WZ})\mathbf{1}_N\}\mathbf{W} \\ \mathbf{W} &\leftarrow (\mathbf{WW}^T)^{-1/2}\mathbf{W}, \end{aligned} \tag{2.35}$$

where $\mathbf{1}_N$ has the meaning of $N \times 1$ vector of ones.

The symmetric approach allows the estimation of the possible Gaussian signal and may improve the estimation of some components that are not easily estimated. On the other hand, signals which would be accurately estimated by the one-unit approach may be degraded by the effect of symmetric orthogonalization.

2.9 The EFICA algorithm

The EFICA algorithm is an extension of FastICA, which was introduced in [24]. It is proven to be asymptotically efficient under the assumption that the probability distribution of the independent components belongs to the class of *Generalized Gaussian Distribution* (GGD) with the parameter $\alpha < 2$. This section focuses on a brief description of EFICA, because it is the basis of the Block EFICA ([28], Chapter 3) algorithm, proposed partially by the author of this thesis.

First, let us introduce **the Generalized Gaussian Distribution**, which plays an important role in EFICA. GGD is a family of parametric continuous distributions

on the real line. It is given by a single positive parameter α that controls the exponential decay rate of the distribution.

A random variable y, $\mathrm{E}\{y\} = 0, \mathrm{E}\{y^2\} = 1$, belongs to the GGD family if its probability density function is given by

$$f(y, \alpha) = a(\alpha) \cdot \exp\{-[b(\alpha) \cdot |y|]^\alpha\}, \tag{2.36}$$

where $a(\alpha) = \alpha \cdot b(\alpha)/2\Gamma(1/\alpha)$, $b(\alpha) = \sqrt{\Gamma(3/\alpha)/\Gamma(1/\alpha)}$ and $\Gamma(\cdot)$ is a Gamma function.

The Gaussian distribution is given as $f(y, 2)$. Distributions with $\alpha < 2$ are called *Supergaussian*, with kurtosis higher than Gaussian distribution, e.g. the Laplace distribution with $\alpha = 1$. The distributions with $\alpha > 2$ are *Subgaussian*, e.g. the uniform distribution ($\alpha \to \infty$).

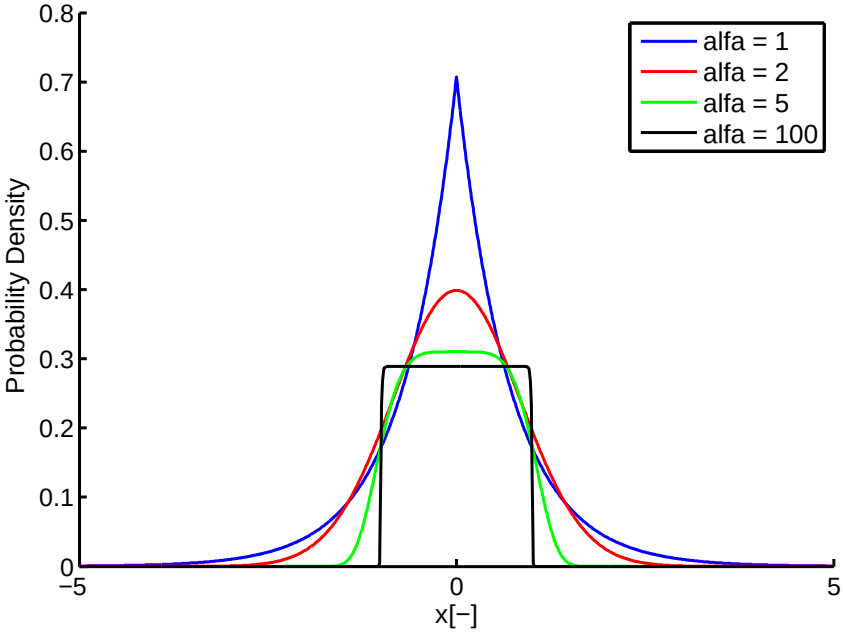


Figure 2.3: Examples of pdfs which belong to the GGD family

EFICA measures the independence by the entropy approximations in the form of suitable non-linear functions $G(\cdot)$, i.e. in the same manner as the original FastICA. EFICA stems from the idea that FastICA can be run with distinct nonlinearity $G_i(\cdot)$ for each of the sources. An optimal choice on nonlinearity $G_i^{opt}(\cdot)$ is the integral of the estimated source's score function (see footnote in Section 2.8), i.e. $G_i(x) = \int \psi_i(x)dx = -\log p_i(x)$. The source distributions are unknown, but EFICA models all sources as if they have GGD with an appropriate parameter α. The function $g(\cdot)$ used further in this section denotes the first derivative of $G(\cdot)$.

Moreover, during the symmetric orthogonalization, it is possible to compute auxiliary constants which minimize mean square error in one row of the estimated matrix $\widehat{\mathbf{W}}$. It is possible to compute these constants in parallel for each of the rows,

thus obtaining $\widehat{\mathbf{W}}$ that asymptotically achieves CRB. This holds when the true score functions are used as nonlinearities in the iterations.

The algorithm consists of three steps. In the first step, the sources are estimated via **(I) the symmetric FastICA** with standard nonlinearity $g(x) = \tanh(x)$. Here the *Test of Saddle Points* [34] is performed in order to get reliable source estimates. The test detects cases when the FastICA gets stuck in the local minima of the contrast function and corrects the incorrectly estimated sources.

The second step resides in **(II) the adaptive choice of nonlinearity** $G_i(x)$. The nonlinearity is selected according to the estimates of the score function of the signals obtained in step (I). It is assumed that the sources are distributed according to GGD. The free parameter α is estimated by equaling the sample fourth-order moment of the source estimates by the theoretical fourth-order moment of the GGD.

The GGD score function is proportional to

$$\psi_{GGD}(x, \alpha) = \text{sign}(x) \cdot |x|^{(\alpha-1)} . \tag{2.37}$$

A problem with the function $\psi_{GGD}(x, \alpha)$ is that it is not continuous for $\alpha \leq 1$, and thus is not a valid FastICA nonlinearity. Here, EFICA distinguishes two cases.

In the case of subgaussian sources ($\alpha > 2$), it selects the optimal nonlinearity as $g_i(x) = \psi_{GGD}(x, \widehat{\alpha}_i)$ where $\widehat{\alpha}_i$ denotes the estimated value of α for the ith source. In the case of supergaussian ($\alpha < 2$) sources, the nonlinearity is given as an ad-hoc function

$$g_i(x) = x \cdot \exp -\eta\,|x|, \tag{2.38}$$

where $\eta = 3.348$.

The final step of the algorithm is called **(III) the refinement**. Here, several one-unit FastICA iterations (2.33) are performed for each of the source estimates from step (I) with the nonlinearity selected in step (II). These iterations are called *fine-tuning*.

Finally, auxiliary constants $c_{k\ell}; k, \ell = 1 \cdots d$ are computed which minimize the mean square error of the rows of $\widehat{\mathbf{W}}$ so that the algorithm attains the CRB (2.27). The computation proceeds as follows, for details see [24][2].

For each $k, \ell = 1 \ldots d$ compute

$$c_{k\ell} = \begin{cases} \frac{V^{1U}_{k\ell}}{V^{1U}_{\ell k}+1}, & k \neq \ell \\ 1, & k = \ell, \end{cases} \tag{2.39}$$

where $V^{1\text{U}}_{k\ell}$ reflects the theoretical asymptotic performance of one-unit FastICA iterations. It is given (see Section 2.7 for details) in the terms of the variance of non-diagonal elements of the gain matrix $\mathbf{G} = \widehat{\mathbf{W}}^{1\text{U}}\mathbf{A}$ as

$$\text{var}[\mathbf{G}^{1\text{U}}_{k\ell}] \approx \frac{1}{N} V^{1\text{U}}_{k\ell}, \quad k \neq \ell \tag{2.40}$$

[2] In fact, the definition of weights in (2.39) slightly differs from [24]. It originates from [28], where it was introduced, because it is more suitable for the proposal of the Block EFICA algorithm in Chapter 3.

More specifically, the terms $V_{k\ell}^{1\mathrm{U}}$ are given by

$$V_{k\ell}^{1\mathrm{U}} = \frac{\gamma_k}{\tau_k^2}, \qquad k \neq \ell, \tag{2.41}$$

with

$$\begin{aligned} \mu_k &= \mathrm{E}[\mathbf{s}_k g_k(\mathbf{s}_k)] & \gamma_k &= \beta_k - \mu_k^2 \\ \nu_k &= \mathrm{E}[g_k'(\mathbf{s}_k)] & \tau_k &= \nu_k - \mu_k \\ \beta_k &= \mathrm{E}[g_k^2(\widehat{\mathbf{s}}_k)], \end{aligned} \tag{2.42}$$

In practice, when a finite number of data samples is available, the expected values in formulas (2.42) are replaced by respective sample means.

The constants $c_{k\ell}$ are used to form the matrix

$$\mathbf{W}_k^+ = [c_{k1}\mathbf{w}_1/\|\mathbf{w}_1\|, \ldots, c_{kd}\mathbf{w}_d/\|\mathbf{w}_d\|]^T, \tag{2.43}$$

where $\mathbf{w}_i, i = 1 \ldots d$ are the rows of the demixing matrix $\widehat{\mathbf{W}}$, estimated by the fine-tuning iterations. The k-th row of the matrix $(\mathbf{W}_k^+ \mathbf{W}_k^{+T})^{-1/2}\mathbf{W}_k^+$ is the kth row of the final demixing matrix $\widehat{\mathbf{W}}$ estimated by EFICA.

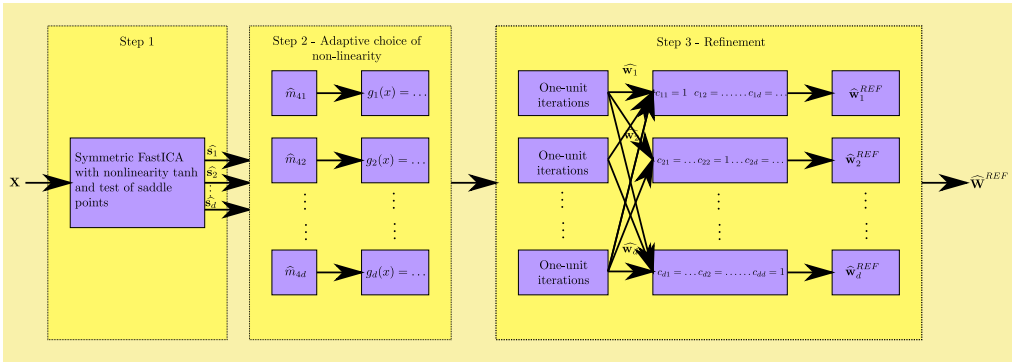


Figure 2.4: The flowchart of the EFICA algorithm. $\widehat{m}_{4i}$ denotes the sample estimate of the 4^{th} order moment of the ith source estimate.

2.10 The BGSEP algorithm

BGSEP (Block Gaussian SEParation) is an ICA algorithm which uses AJD to separate the sources modeled as block-wise stationary iid Gaussian random variables. This asymptotically efficient and computationally undemanding technique was proposed by Tichavský and Yeredor in [32]. This section deals with a brief description of BGSEP because it is used as a separator within the T-ABCD algorithm (the batch version as well as the online version).

BGSEP jointly diagonalizes the sample estimates of covariance matrices $\mathbf{R}_{\mathbf{XX}}[\tau]$ from (2.17). To do so, BGSEP utilizes low complexity diagonalization scheme

called WEDGE (Weighted Exhaustive Diagonalization with Gauss itErations). The WEDGE technique is completely general and its use on different regular matrices results in different ICA algorithms. Two other ICA techniques are proposed by Tichavský in [32]: a computationally effective alternative for WASOBI [25] intended for the separation of stationary AR processes and the BARBI [27] algorithm separating block-stationary AR sources.

There are two forms of the WEDGE iteration: an unweighted (or uniform weighted) alternative U-WEDGE and complete WEDGE introducing arbitrary positive definite weighting matrices $\mathbf{K}$. BGSEP performs several U-WEDGE iterations first in order to obtain a consistent estimate of the sources. Subsequently, the (re)estimation of optimal weight matrices (computed from the source estimates) alternates with single WEDGE iteration refining the estimates, until convergence is achieved.

Let us denote by $\widehat{\mathbf{R}}_{\mathbf{XX}}[r], r = 0 \dots R-1$ the set of sample covariance matrices to be diagonalized and by the upper index I the sequence index of iteration. The BGSEP iterations proceed as follows: first, the estimated demixing matrix is initialized by an initial guess, e.g. $\widehat{\mathbf{W}}^0 = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix. Then during the Ith iteration the target matrices $\widehat{\mathbf{R}}_{\mathbf{XX}}$ are partially diagonalized as

$$\widehat{\mathbf{R}}_{\mathbf{S}}^I[r] = \widehat{\mathbf{W}}^{I-1}\widehat{\mathbf{R}}_{\mathbf{XX}}[r](\widehat{\mathbf{W}}^{I-1})^T, \;\; r = 0 \dots R-1 \tag{2.44}$$

Subsequently, the demixing matrix is updated via

$$\widehat{\mathbf{W}}^I = (\widehat{\mathbf{A}}^I)^{-1}\widehat{\mathbf{W}}^{I-1}, \tag{2.45}$$

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

$$\begin{bmatrix} \widehat{\mathbf{A}}_{kl}^I \\ \widehat{\mathbf{A}}_{lk}^I \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{ll}^T\mathbf{K}[kl]^I\mathbf{r}_{ll} & \mathbf{r}_{kk}^T\mathbf{K}[kl]^I\mathbf{r}_{ll} \\ \mathbf{r}_{kk}^T\mathbf{K}[kl]^I\mathbf{r}_{ll} & \mathbf{r}_{kk}^T\mathbf{K}[kl]^I\mathbf{r}_{kk} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_{ll}^T\mathbf{K}[kl]^I\mathbf{r}_{kl} \\ \mathbf{r}_{kk}^T\mathbf{K}[kl]^I\mathbf{r}_{kl} \end{bmatrix}, \tag{2.46}$$

where $\mathbf{r}_{kl}$ is a $R \times 1$ vector given by

$$\mathbf{r}_{kl} = [(\widehat{\mathbf{R}}_{\mathbf{S}}^I[0])_{kl}, \dots, (\widehat{\mathbf{R}}_{\mathbf{S}}^I[R-1])_{kl}]^T \tag{2.47}$$

for $k, l = 1, \dots, d$, $k > l$ and $\mathbf{K}[kl]$ denotes square $R \times R$ matrix of weights. For the initial U-WEDGE iterations, the weight matrix $\mathbf{K}[kl] = \mathbf{I}$. The weighted iterations introduce diagonal weight $\mathbf{K}[kl]$ in the form

$$\mathbf{K}[kl] = \mathrm{diag}\left(\frac{1}{(\widehat{\mathbf{R}}_{\mathbf{S}}^I[0])_{kk}(\widehat{\mathbf{R}}_{\mathbf{S}}^I[0])_{ll}}, \dots, \frac{1}{(\widehat{\mathbf{R}}_{\mathbf{S}}^I[R-1])_{kk}(\widehat{\mathbf{R}}_{\mathbf{S}}^I[R-1])_{ll}} \right) \tag{2.48}$$

This iterating continues until convergence is achieved.

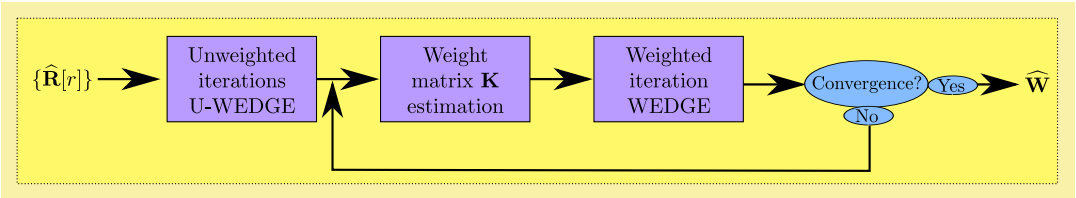


Figure 2.5: The flowchart of the BGSEP algorithm

2.11 Applications of ICA

The ICA methods can be applied to a wide variety of practical problems. Let us name some technical areas, where ICA can be usefully exploited.

The ICA methods can be used for solving of the **cocktail party problem** mentioned earlier. However, the instantaneous ICA model (1.1) does not take the basic properties of sound into account. The mixing of acoustic signals is better modeled by the convolutive model (1.2). Prior to the utilization of ICA on sound mixtures, the convolutive mixing needs to be transformed into instantaneous one. This process will be explained later in Chapter 4.

ICA may be applied to recordings originated from non-invasive brain mapping techniques, like EEG and MEG. For these signals the instantaneous noiseless ICA model holds well. The evoked potentials measured via EEG are truly mutually independent. Moreover, the speed of propagation of the electrical brain activity is high enough, for the mixtures recorded on electrodes to be considered instantaneous. The mixing process is stationary due to the mutual fixed position of the brain centers. On the other hand, there is one theoretical violation of the ICA model: The EEG and MEG signals are not stationary. This is solved by the block-wise application of the ICA methods on analyzed data, where the data in each block are considered stationary.

In the context of EEG, ICA is usually used for **artifact removal**. Here, the artifact is a signal which is not produced by the brain activity. It is created via some external influence, like muscular activity, eye movement or heart activity. The artifact may originate even from some external electro-magnetical phenomenons, e.g. the power supply voltage of 50 Hz may influence the measured EEG. The artifacts and brain signals are truly mutually independent. The artifacts can be located even when their energy is much lower than the energy of brain signals. More details can be found in [35].

ICA can be utilized for **image coding and noise reduction**. The image can be expressed as a linear superposition of several basis images (functions) as

$$I(x,y) = \sum_{i=1}^{k} \iota_i(x,y) a_i, \tag{2.49}$$

where $I(x,y)$ is the brightness of image I on coordinates x, y, a_i are some coefficients (which vary for different images) and $\iota_i(x,y)$ are the basis images. The most suitable

basis images can be learned directly from pictures encoded via ICA.

For practical computation, the target image is fragmented into small areas. The values of brightness of each such area stored in the columns of $\mathbf{X}$, the basis images are represented by a mixing matrix $\mathbf{A}$ and the columns of $\mathbf{S}$ contain the unknown weighting coefficients for all fragments.

The noise reduction consists in the fact that the basis functions with lower coefficients a_i usually represent noise. Therefore, the image is reconstructed from basis images with a_i higher than a selected threshold only. For detail information to this topic see [36].

ICA may be used for **financial data analysis**. There are situations where parallel financial time series are available, e.g. the stock portfolio values over time or the currency exchange rates. The independent components in this scenario represent seasonal/annual variations or factors which have a sudden effect on customers, such as changes in commodity prizes. The components may also have a specific meaning in the context of the analyzed time series, e.g. the current situation in the country using the currency. Nevertheless, the meaning of specific components needs to be determined. This poses a considerable complication for the ICA application on financial data.

An example of the ICA application on financial data may be found in [37]. Here, the cash flow of forty stores that belong to the same retail chain is analyzed. The analysis aims at finding common factors which affect cash flow. ICA was used to predict the exchange rates in [38].

Chapter 3

The Block EFICA algorithm

This chapter centers on a thorough description of the Block EFICA algorithm, which is designed for the separation of non-stationary and non-gaussian sources. The algorithm is proven to be asymptotically efficient, provided that the variances of unknown sources are constant.

The algorithm is based on the EFICA algorithm ([24], Section 2.9). There are two major differences between EFICA and Block EFICA algorithms.

The first modification consists in the implementation of the parametric score function estimator. The estimated score function is used as a contrast in the one-unit iterations within the fine-tuning part of the method. This approach is described in Section 3.1 and paper [39].

The second realized modification consists in the introduction of non-stationarity into the EFICA's model. The Block EFICA assumes the sources to be block-wise stationary. This approach is described in detail in Section 3.2 and papers [28] and [40].

3.1 Parametric score function estimator

This section introduces the utilization of a parametric score function estimator within (Block) EFICA. The term score function has already been discussed in Section 2.7. For the sake of clarity, let us restate it first.

The optimal contrast function $G_k^{opt}(x)$ for the kth source in the FastICA criterion $J(\mathbf{w})$ (2.31) is obtained by taking $G_k^{opt}(x) = -\ln(p_k(x))$. Then $J(\mathbf{w})$ expresses directly the entropy.

The function $G_k^{opt}(x)$ itself is not present in the FastICA/EFICA iteration. The iteration contains its derivative denoted as $g_k^{opt}(x)$. The function $g_k^{opt}(x) = \psi_k(x) = [-\ln(p_k)(x)]' = -p_k'(x)/p_k(x)$ is called the *score function* of probability density $p_k(x)$ and can be estimated directly from available realizations of the random variable.

The original EFICA implements the score function estimator for the GGD sources. This section considers more general score function estimator which is not restricted for GGD sources and thus can be more accurate for a wider range of distributions. It is suitable for FastICA based algorithms because it provides continuous and differentiable functions. Moreover, the proposed estimator is computationally undemanding, which is advantageous for the implementation within Block EFICA. Here, the score function may be different in each analyzed block.

The proposed modification is following. Within the adaptive second step of EFICA, the function $g_k(x)$ is selected as a score function estimate for each of the sources separately. This is done prior to the first one-unit iteration and the non-linearity is re-estimated after each iteration.

Several parametric and non-parametric score function estimators/probability density estimators have been considered for the utilization within the modified EFICA algorithm. More specifically, following approaches have been investigated: the non-parametric density estimator with the Gauss kernel [12], the parametric density estimator for the Generalized Gaussian Density family [41], the parametric density estimator for the generalized gamma density [42], the score-function estimation through the Pearson parametric system of rational functions [43].

Finally, the parametric score function estimator based on the minimization of mean squared error [44] was selected as the most suitable estimator for the utilization in the modified EFICA. Its main advantages include stability and computational simplicity.

This estimator minimizes the criterion given as the mean squared error between the true unknown score function $\Psi(y)$ and its searched estimate $\mathbf{h}(y|\theta)$, i.e.

$$\mathrm{E}[(\Psi(y) - \mathbf{h}(y|\theta))^2], \tag{3.1}$$

where θ is the vector of parameters which is subject to minimization and E stands for the expected value operator.

The square of (3.1) gives

$$\mathrm{E}[\Psi(y)^2] - 2\mathrm{E}[\mathbf{h}(y|\theta)\Psi(y)] + \mathrm{E}[\mathbf{h}(y|\theta)^2]. \tag{3.2}$$

Equation (3.2) can be expressed without dependence on the unknown score function using the following theorem from [44] and [45]

$$\mathrm{E}[\mathbf{h}(y|\theta)\Psi(y)] = E[\mathbf{h}'(y|\theta)]. \tag{3.3}$$

From (3.2) by using (3.3) and by leaving the term independent on θ, the unknown vector Θ is expressed in the form

$$\Theta = \arg\min_{\theta} \mathrm{E}[\mathbf{h}(y|\theta)^2] - 2\mathrm{E}[\mathbf{h}'(y|\theta)] \tag{3.4}$$

To simplify the optimization problem, the function $\mathbf{h}(y|\theta)$ is considered to be a linear combination of functions $h_1(y), ..., h_k(y)$, i.e.

$$\mathbf{h}(y|\theta) = \sum_{i=1}^{k} \theta_i \cdot h_i(y), \tag{3.5}$$

This allows to express (3.4) in the form

$$\Theta = \mathrm{E}[\mathbf{h}^T(y) \cdot \mathbf{h}(y)]^{-1} \cdot \mathrm{E}[\mathbf{h}'(y)], \tag{3.6}$$

where $\mathbf{h}(y) = [h_1(y), ..., h_k(y)]^T$ and $\mathbf{h}'(y) = [h_1'(y), ..., h_k'(y)]^T$.

The functions $h_i(y)$ are called basis functions and their selection is an important issue. It is possible to choose $h_i(y) = y^i$. In this case, the function $\mathbf{h}(y|\theta)$ is a polynomial, which was used for example in [45]. Another possible approach is to select the functions $h_i(y)$ as score functions of some important distributions. This approach is suggested in [46]. A suitable choice for the utilization in the Block EFICA iteration is the set of functions which were used as contrast functions in the FastICA algorithm. This approach is described in [39]. Based on this observation, following functions are chosen as basis functions for the separation of super-gaussian and sub-gaussian components, respectively:

$$h_1(y) = x/(1 + 6\,|x|)^2, h_2(y) = x^3. \tag{3.7}$$

3.2 Separation of block-wise stationary sources

In the following text, the term block-wise stationary model is discussed. Subsequently, the BlockEFICA algorithm, which is based on the model, is proposed.

3.2.1 Block-wise stationary instantaneous model of ICA

In the case of non-stationary source signals, the samples of $\mathbf{s}_i$ are not identically distributed. One way to model the non-stationarity is to assume that the distributions of the sources differ in each time instant or have fixed distributions with varying parameters, e.g. variances.

To allow practical estimation, it is expected that there are Q blocks in $\mathbf{S}$ of the same integer length. In these blocks, the distribution of the signals does not change. In this section, the upper index (I) is used to denote random variables or functions which are related to the Ith such block.

The source signals $\mathbf{s}_i$ are in blocks i.i.d. random variables. Thus, the instantaneous model in (1.1) holds in each considered block, i.e.

$$\mathbf{X}^{(I)} = \mathbf{A} \cdot \mathbf{S}^{(I)}, I = 1 \ldots Q. \tag{3.8}$$

3.2.2 Description of Block EFICA

The Block EFICA consist of three steps, which are similar to those in the original algorithm (see Section 2.9). The flow of the algorithm is illustrated in Figure 3.1.

(I) The symmetric FastICA [15] with the test of saddle points([34]) is exploited to obtain the pre-estimate of the demixing matrix $\widehat{\mathbf{W}}$.

(II) The fine-tuning of the rows of $\widehat{\mathbf{W}}$ is performed. The computation proceeds via modified FastICA one-unit iterations.

In order to reflect the non-stationary of the model (3.8), the Block EFICA optimizes a different contrast function than EFICA. The Block EFICA contrast function is given by

$$J(\mathbf{w}_k) = \lambda_k^{(1)} \mathrm{E}\{G_k^{(1)}(\mathbf{w}_k^T \mathbf{Z}^{(1)})\} + \ldots + \lambda_k^{(Q)} \mathrm{E}\{G_k^{(Q)}(\mathbf{w}_k^T \mathbf{Z}^{(Q)}))\}, \tag{3.9}$$

where $k = 1 \dots d$, $\lambda_k^{(I)}$ are some weights and $G_k^{(I)}$ are suitable nonlinear functions. Note that this contrast function is not a simple linear combination of $G_k(x)^{(1)}, \dots, G_k(x)^{(Q)}$, because the expectations in (3.9) depend on different distributions from corresponding blocks of the whitened mixtures.

The extrema of (3.9) are computed through a fixed-point iteration given by

$$\mathbf{w_k} \leftarrow \lambda_k^{(1)}(\mathrm{E}[\mathbf{Z}^{(1)} g_k^{(1)}(\mathbf{w}_k^T \mathbf{Z}^{(1)})] - \mathbf{w}_k \mathrm{E}[g_k^{(1)'}(\mathbf{w}_k^T \mathbf{Z}^{(1)})]) + \dots \\ \dots + \lambda_k^{(Q)}(\mathrm{E}[\mathbf{Z}^{(Q)} g_k^{(Q)}(\mathbf{w}_k{}^T \mathbf{Z}^{(Q)})] - \mathbf{w}_k \mathrm{E}[g_k^{(Q)'}(\mathbf{w}_k{}^T \mathbf{Z}^{(Q)})]. \quad (3.10)$$

In practice, when the number of samples is finite, the expectations are replaced by sample means. Throughout the text, equation (3.10) is further referred to as *block one-unit FastICA iteration*. The non-linearities $g_k^{(I)}(\cdot)$ are computed using the parametric score function estimator described in Section 3.1 and are re-estimated after each one-unit iteration. Simultaneously, the weights $\lambda_k^{(I)}$ are updated. The selection of the weights $\lambda_k^{(I)}$ is based on the theoretical performance analysis and is discussed in the next section.

(III) The refinement consists in the computation of the auxiliary constants $c_{k\ell}, k, \ell = 1 \dots d$, which minimize the mean square error of rows of $\widehat{\mathbf{W}}$. In this manner, the asymptotical efficiency of the algorithm is attained. The constants are exploited to form the matrix

$$\mathbf{W}_k^+ = [c_{k1}\mathbf{w}_1/\|\mathbf{w}_1\|, \dots, c_{kd}\mathbf{w}_d/\|\mathbf{w}_d\|]^T \quad (3.11)$$

Then, the k-th row of the matrix $(\mathbf{W}_k^+ \mathbf{W}_k^{+T})^{-1/2}\mathbf{W}_k^+$ is the kth row of the final demixing matrix $\widehat{\mathbf{W}}^{\mathrm{BEF}}$ estimated by Block EFICA. The selection of weights is discussed in detail in the next section.

The number of blocks Q is usually unknown in practice. When a too high value of Q is chosen, the overparametrization may occur and cause a higher estimation error. The simulations in Figure 3.5.1 show that Block EFICA is not highly sensitive to this parameter. Neither the overestimation nor the underestimation decreases the performance significantly.

The selection of Q should reflect the characteristics of the signals to be separated. For instance, when separating speech, the length of blocks should correspond to 20-25 ms where the speech is considered stationary. Blind estimation of number of blocks is possible as well, based on the estimation of residual inter-signal interference (signal-to-interference ratio - SIR) using analytical expressions from Section 3.3, namely (3.18). Corresponding statistics are estimated from separated signals. In the beginning, SIR usually improves with growing Q, but for larger Q the growth is slower and slower. Q should be selected as a value where the increase of SIR becomes slow; see Figure 3.8. Another possible approach for an automated choice of Q can be found in [47].

At the end of this section let us mention one important detail concerning the implementation of the **parametric score function estimator** in Block EFICA.

In Section (3.1), two basis functions were proposed for the computation in the estimator: One for super-gaussian sources and the other for sub-gaussian sources.

For the utilization in Block EFICA, it is reasonable to take into account the identity function $h_3(x) = x$ as the third basis function. The identity function does not have any relevance in the context of the original FastICA/EFICA. There, for the stationary whitened data, it holds that $\mathrm{E}[\mathbf{Z}\mathbf{Z}^T] = \mathbf{I}$. Thus, the effect of identity in (2.33) is zeroed in all cases. On the other hand, in the context of BlockEFICA, the block-wise stationary sphered data $\mathbf{Z}$ may not necessarily be sphered within each considered block.

Inclusion of the identity function into the score function estimator, in fact, conveys the direct utilization of second-order statistics of signals. The consideration is worthwhile especially when separating signals with a changing variance.

3.3 Performance analysis of the Block EFICA algorithm

The performance analysis of the Block EFICA algorithm determines the selection of weights $\lambda_k^{(I)}$ and auxiliary constants $c_{k\ell}$. Suitably selected values of these constants allow Block EFICA to optimize its performance, and under some conditions, to achieve the asymptotical efficiency. For details on the performance analysis see [40].

Throughout this section it is assumed that the variances of the sources are equal to 1 in each block. This simplification is introduced, because the formulae for weights $\lambda_k^{(I)}$ become otherwise overparametrized and the weights are thus prone to estimation errors. Although this fact restricts the theoretical performance, the simulations prove that the performance of the algorithm is not distinctively deteriorated when the variances of the signals are changing.

First, let us state the Cramér-Rao lower bound for the piecewise stationary instantaneous ICA model with unit variance signals. As discussed in Section 2.7, the performance of ICA algorithms is measured through variance of the non-diagonal elements of the gain matrix $\mathbf{G} = \widehat{\mathbf{W}}\mathbf{A}$. The Cramér-Rao lower bound of this variance is given via

$$\mathrm{CRLB}[\mathbf{G}_{k\ell}] = \frac{1}{N}\frac{\overline{\kappa}_\ell}{\overline{\kappa}_k\,\overline{\kappa}_\ell - 1} \qquad k \neq \ell, \tag{3.12}$$

where $\overline{\kappa}_k \overset{\text{def.}}{=} \frac{1}{Q}\sum_{I=1}^{Q} \kappa_k^{(I)}$ and $\kappa_k = \mathrm{E}[\psi_k^2(x)]$.

The variance of the gain matrix elements obtained from the block one-unit FastICA iteration is approximately given by

$$\mathrm{var}[\mathbf{G}_{k\ell}^{\mathrm{B1U}}] \approx \frac{1}{N} V_{k\ell}^{\mathrm{B1U}}, \qquad k \neq \ell, \tag{3.13}$$

where $V_{k\ell}^{\mathrm{B1U}}$ is given by

$$V_{k\ell}^{\mathrm{B1U}} = \frac{\overline{\beta}_k - \overline{\mu}_k^2}{\overline{\tau}_k^2}, \qquad k \neq \ell, \tag{3.14}$$

with

$$\begin{aligned}
\overline{\beta}_k =& \frac{1}{Q}\sum_{I=1}^{Q}(\lambda_k^{(I)})^2\beta_k^{(I)} & \overline{\mu}_k =& \frac{1}{Q}\sum_{I=1}^{Q}\lambda_k^{(I)}\mu_k^{(I)} \\
\overline{\tau}_k =& \overline{\nu}_k - \overline{\mu}_k & \overline{\nu}_k =& \frac{1}{Q}\sum_{I=1}^{Q}\lambda_k^{(I)}\nu_k^{(I)} \\
\mu_k^{(I)} =& \mathrm{E}[s_k^{(I)}\, g_k^{(I)}(s_k^{(I)})] & \nu_k^{(I)} =& \mathrm{E}[g_k^{(I)'}(s_k^{(I)})] \\
\beta_k^{(I)} =& \mathrm{E}[g^2{}_k^{(I)}(s_k^{(I)})]
\end{aligned} \tag{3.15}$$

To achieve the best performance, the term $V_{k\ell}^{\mathrm{B1U}}$ must be minimized with respect to the free parameters $\lambda_k^{(I)}$. The optimal weights are then given by

$$\lambda_k^{(J)} = \frac{1}{Q}\left(\frac{\tau_k^{(J)}}{\beta_k^{(J)}} + A_k B_k \frac{\mu_k^{(J)}}{\beta_k^{(J)}}\right), \qquad J = 1, \ldots, Q, \tag{3.16}$$

where

$$A_k = \left(\sum_{I=1}^{Q}\frac{\gamma_k^{(I)}}{\beta_k^{(I)}}\right)^{-1}$$

and

$$B_k = \sum_{I=1}^{Q}\frac{\mu_k^{(I)}\tau_k^{(I)}}{\beta_k^{(I)}}.$$

The final performance of Block EFICA is given after the refinement step III. Here, the optimum constants $c_{k\ell}$ are exploited according to (3.11) such that the Cramér-Rao lower bound for the performance is achieved. The constants $c_{k\ell}$ are given by

$$c_{k\ell} = \begin{cases} \frac{V_{k\ell}^{\mathrm{B1U}}}{V_{\ell k}^{\mathrm{B1U}}+1}, & k \neq \ell \\ 1, & k = \ell \end{cases}, \tag{3.17}$$

The final performance of Block EFICA, given in the terms of the gain matrix $\mathbf{G}^{\mathrm{BEF}} = \widehat{\mathbf{W}}^{\mathrm{BEF}}\mathbf{A}$, is expressed as

$$\mathrm{var}[\mathbf{G}_{k\ell}^{\mathrm{BEF}}] \approx \frac{1}{N}\frac{V_{k\ell}^{\mathrm{B1U}}(V_{\ell k}^{\mathrm{B1U}}+1)}{V_{k\ell}^{\mathrm{B1U}}+V_{\ell k}^{\mathrm{B1U}}+1}, \qquad k \neq \ell. \tag{3.18}$$

The optimum performance of Block EFICA occurs when the nonlinearities estimated in step II equal the true score functions in respective block, i.e. $g_k(x)^{(I)} = \psi_k^{(I)}(x)$ for $k = 1 \ldots d, I = 1 \ldots Q$. Then it holds that $\beta_k^{(I)} = \nu_k^{(I)} = \kappa_k^{(I)}$, $\mu_k^{(I)} = 1$, and $\tau_k^{(I)} = \gamma_k^{(I)} = \kappa_k^{(I)} - 1$. Next, the formula for λs (3.16) simplifies to a constant, namely, $\lambda_k^{(J)} = 1/Q$, however another constant value may be considered. Then $\overline{\beta}_k = \overline{\nu}_k = \overline{\kappa}_k$ and $\overline{\mu}_k = 1$. Now, the performance (3.14) becomes equal to

$$V_{k\ell}^{\mathrm{B1U}} = \frac{1}{\overline{\kappa}_k - 1}. \tag{3.19}$$

Inserting (3.19) into (3.18) gives

$$\mathrm{var}[\mathbf{G}_{k\ell}^{\mathrm{BEF}}] \approx \frac{1}{N} \frac{\overline{\kappa}_\ell}{\overline{\kappa}_k \, \overline{\kappa}_\ell - 1}, \qquad k \neq \ell. \tag{3.20}$$

When comparing (3.20) and (3.12), it can be seen that Block EFICA is asymptotically efficient for piecewise stationary sources with constant variance, provided that the score functions are estimated correctly.

The uniformity of the weights $\lambda_k^{(I)}$ for the case $g_k(x)^{(I)} = \psi_k^{(I)}(x)$ results in a simplified version of the algorithm called **Uniform Block EFICA**. Here, all the weights $\lambda_k^{(I)}$ are equal one. This reduces the number of parameters which need to be estimated by dQ.

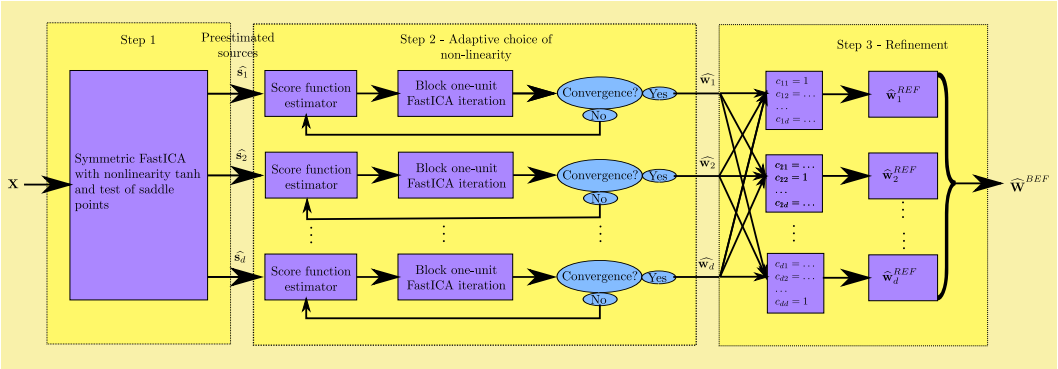


Figure 3.1: Flowchart of the Block EFICA algorithm.

3.4 Experiments: The modified EFICA with the parametric score function estimator

This section presents a series of experiments, which test the modified EFICA algorithm with the parametric score function estimator. The experiments verify the functionality of the estimator within the algorithm and suggest that the estimator can be applied within the Block EFICA algorithm.

In the experiments, the performance of the modified EFICA is compared to other well known ICA algorithms based on non-gaussianity. Namely, the FastICA with the contrast function "tanh" [33], JADE [17] and the original EFICA algorithm [24] were selected.

3.4.1 Separation of the GGD sources

This experiment deals with the separation of sources with pdfs belonging to GGD. It proves that the modified EFICA algorithm with the score function estimator separates these signals as accurately as the original version, which is designed specially

for these sources. The original EFICA is proven to be statistically efficient for the GGD sources with $\alpha > 2$.

The experiment proceeds as follows: five random signals of length 5000 samples were generated for a fixed value α in the interval $0,3 < \alpha < 20$. A random matrix $\mathbf{A}$ was generated and the data were mixed according to (1.1). Four competing algorithms performed the separation. Subsequently, the values of ISR were computed and averaged over all sources.

Figure 3.2 shows the results of this experiment averaged over 100 Monte-Carlo trials with respect to specific values α. The modified version of EFICA with the score function estimator is denoted as "EFICA mod" in the legend, the original version is labeled as "EFICA orig".

It can be seen that the performance of the original and of the modified algorithm are very similar as expected. The modified algorithm has the advantage of the general selection of a contrast function, while the original method assumes that the sources belong to the GGD family of distributions. The experiment validates that the selected score function estimator is suitable for the computation of a contrast function in the EFICA/Block EFICA method.

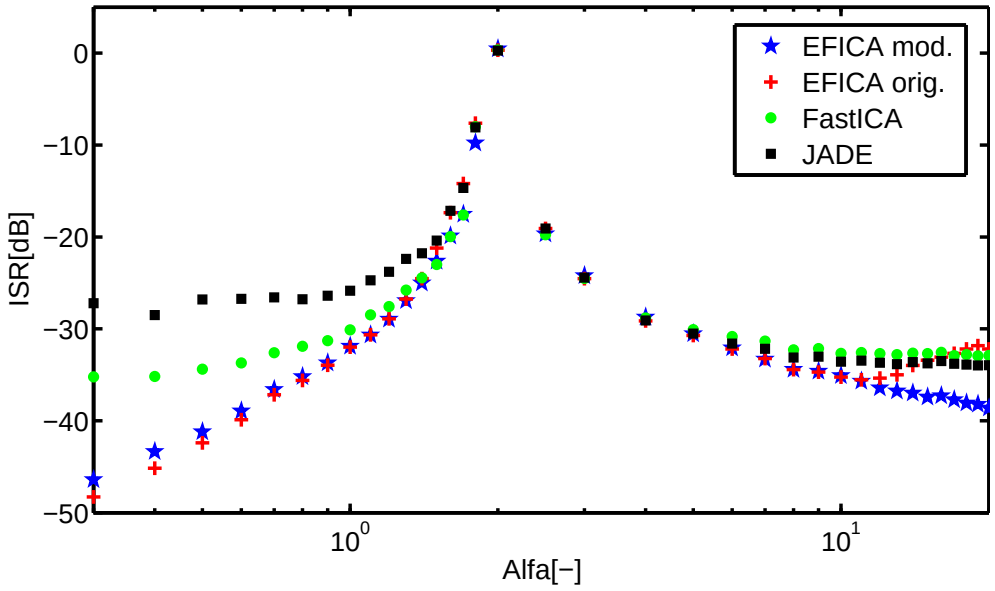


Figure 3.2: Separation of GGD sources

3.4.2 Separation of GGD sources with absorbed Gaussian noise

This experiment deals with the separation of sources which do not belong to the GGD family. The example aims at showing the advantage of the general selection of the score function without any assumptions about the distributions of the sources.

The experiment was designed in the following manner: five randomly generated signals with a distribution from the GGD family with fixed α were summed with

Gaussian noise whose variance was 0.2 times the variance of these signals. The length of the signals was $N = 2500$ samples. The sources were mixed according to (1.1) via a random square mixing matrix $\mathbf{A}$ and subsequently separated via four compared algorithms. The evaluation is given in the terms of ISR averaged over all sources.

The probability distribution of the sources separated in this example does not belong to the GGD family. It stems from the fact that the sum of two random variables has the probability density which is given as the convolution of the two original distributions.

Figure 3.3 summarizes the results of the experiment. The mean ISR is averaged over 100 Monte-Carlo trials. The modified EFICA outperforms the other algorithms for $\alpha < 2$. For $\alpha > 2$, the results are comparable to the original EFICA algorithm.

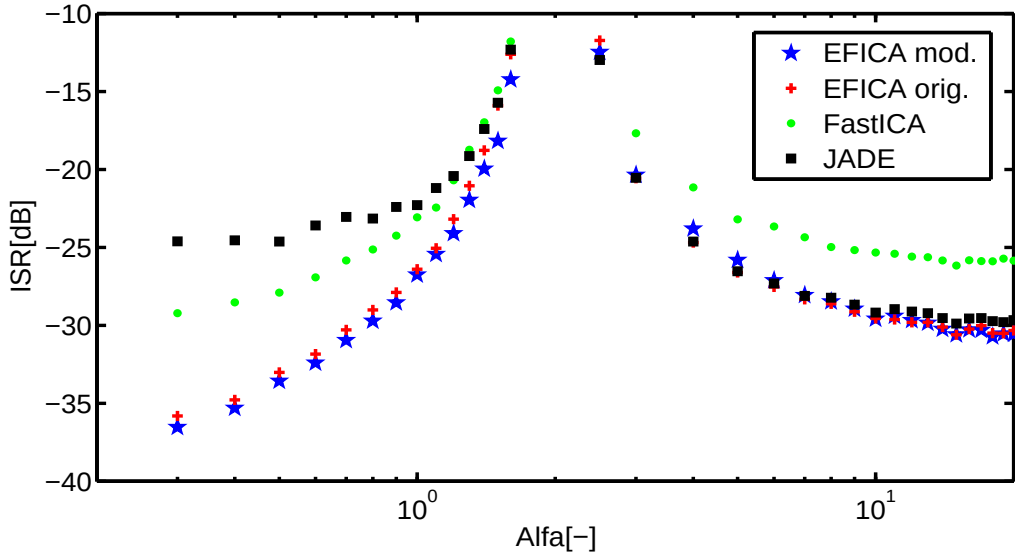


Figure 3.3: Separation of GGD sources with absorbed Gaussian noise

3.4.3 Separation of binary phase shift keying (BPSK) signals with absorbed Gaussian noise

This example shows the ability of the parametric score function estimator to deal with sources which have bimodal probability distributions.

The sources employed in this experiment are distributed according to $\sqrt{1-\varepsilon^2}b + \varepsilon \cdot n$, where b is a binary random variable equal to 1 or -1 with equal probabilities, n is a standard Gaussian variable and $\varepsilon \in [0, 1]$ is a free parameter, which determines the shape of the distribution. The sources are binary random variables for $\varepsilon = 0$. With growing epsilon, the probability density changes from a strongly bimodal to unimodal Gaussian distribution when $\varepsilon = 1$. The pdfs for distinct values of ε are shown in Figure 3.4.

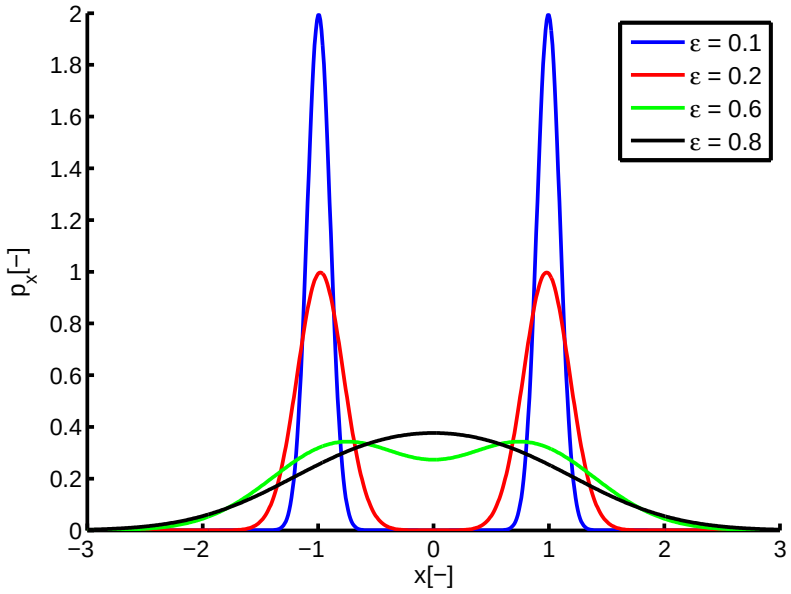


Figure 3.4: Probability density functions of BPSK signals with absorbed Gaussian noise for $\varepsilon = 0.1, 0.2, 0.6, 0.8$.

Five random signals of length $N = 2500$ samples were generated according to the distribution described above and mixed with a random matrix $\mathbf{A}$. Subsequently, four competing algorithms were exploited for separating the sources back. The output mean ISR values averaged all sources and 100 Monte-Carlo trials are presented in Figure 3.5. It can be seen that the modified EFICA algorithm outperforms the other algorithms, especially for $\varepsilon \approx 0, 2$, where the separated signals have a strongly bi-modal distribution.

3.4.4 Conclusion

All the implemented experiments show the advantage of general selection of the score function via the parametric estimator. The performance of the modified EFICA algorithm suggests that the estimator is suitable for the utilization in the second adaptive step of the EFICA algorithm and can be exploited within the proposed Block EFICA.

3.5 Experiments with the Block EFICA algorithm

Several various experiments were performed to verify the accuracy and stability of the Block EFICA algorithm as a whole. The performance of the proposed algorithm is compared to the most competitive algorithms for the given scenario. The FastICA [15] with the nonlinearity $g(\cdot) = \tanh(\cdot)$, the EFICA [24] algorithm and the Extended Infomax [48] are considered as representatives of the non-gaussianity

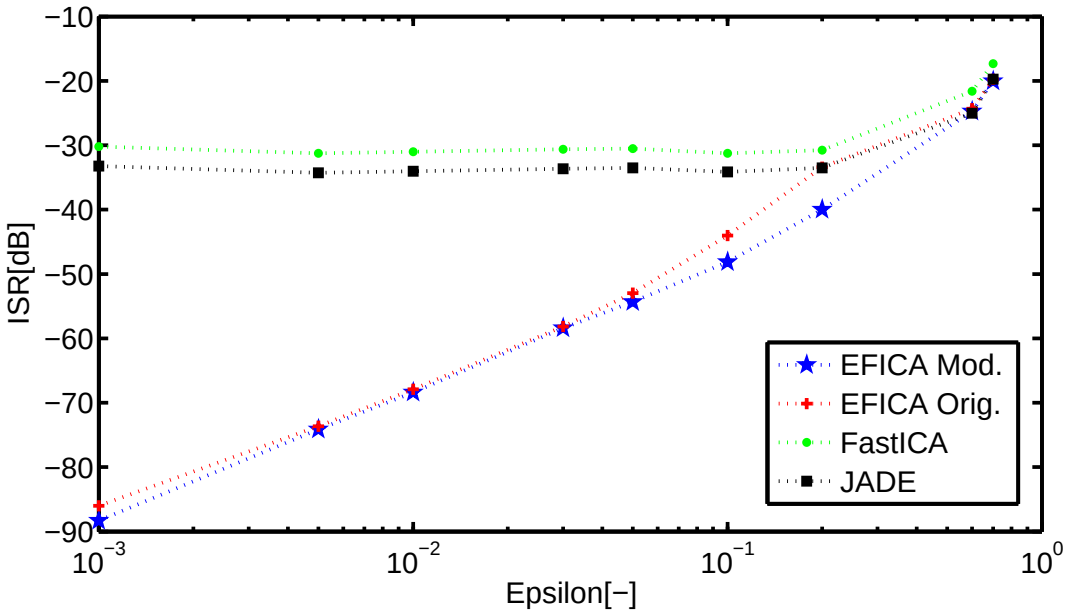


Figure 3.5: Separation of BPSK sources with absorbed Gaussian noise

based approaches. Second order statistics based BGL [21] algorithm, designed for separation of block-wise stationary Gaussian sources, is considered in several examples.

The NSNG algorithm proposed by Pham [49], which exploits both non-stationarity and non-Gaussianity, stands here for a method which belongs to the same class of algorithms as Block EFICA. NSNG yields excellent theoretical performance. However, during the experiments, cases of instability were encountered. NSNG seems to work well in scenarios where a small number of sources is considered and their properties perfectly fit the given model. On the other hand, in experiments with real-world signals, the method fails with a non-negligible probability. The performance of NSNG is shown only in experiments where the method yields meaningful results.

For several experiments, the computational demands of the algorithms are stated in the legends of the corresponding figures. All simulations were performed in Matlab^{TM} on a PC with single core 3 GHz processor and 2 GB of RAM.

Several experiments presented in this section deal with the separation of artificial sources with constant variance which correspond to the theoretical model of Block EFICA. These experiments verify the performance analysis of Block EFICA presented in Section 3.3. The other experiments verify the robustness of Block EFICA in scenarios where the respective model does not hold. This concerns the experiments with artificial sources with a varying variance as well as real world speech data. Moreover, the examples with real world data prove the practical contribution of the algorithm.

3.5.1 Separation of block-wise stationary constant variance signals

The experiments demonstrated in this section deal with signals which correspond to the theoretical model of Block EFICA (3.8).

The first example considers the separation of twenty artificial block-wise stationary sources. The length of the signals is $N = 10000$ samples. Each signal consists of four blocks of the same length $N/4$. The first and the third blocks have the Gaussian distribution and the second and the fourth blocks have the GGD(α) distribution. The parameter α is fixed for each of the 20 signals, where its values are uniformly chosen from $[0.1, 10]$. The variance of all the distributions is one. Thus, the signals have constant variance. For the purposes of the experiment, these signals are mixed by a random matrix $\mathbf{A}$. The results shown in Figure 3.6 are presented in the terms of the mean interference-to-signal ratio and are averaged over 100 Monte Carlo trials.

Theoretical performance, marked in figures by "theory" in the legend, was estimated from separated signals using (3.14) and (3.18). Results of this experiment corroborate the validity of the analysis due to the proximity of the theoretical results with the empirical ones. They also demonstrate the improved performance of the proposed method compared to EFICA, as different distributions on the four blocks of signals were considered. The performance of the NSNG algorithm is not demonstrated here, because its original implementation is designed for sub-gaussian signals only, and the method fails to converge in this experiment.

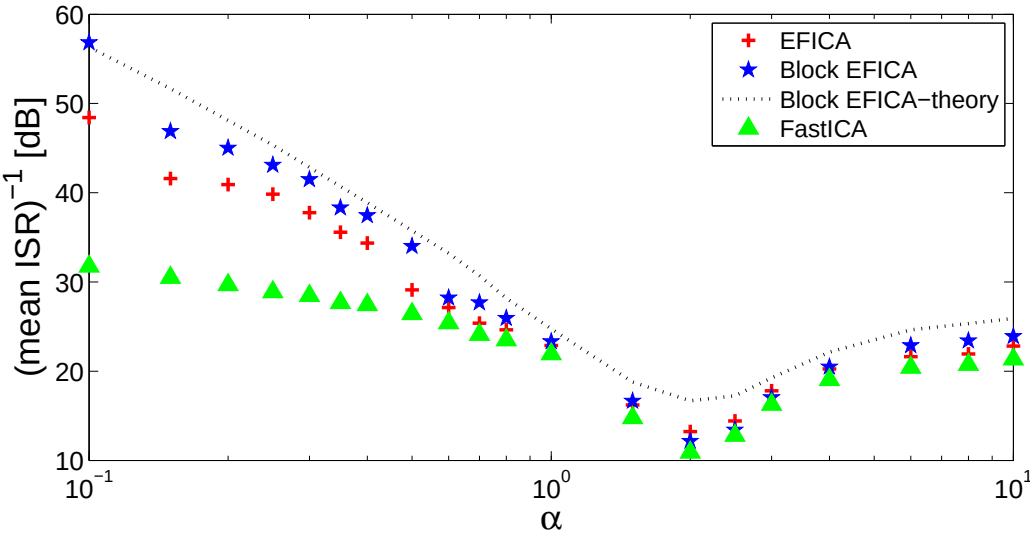


Figure 3.6: Mean interference-to-signal ratio of the separated signals in the experiment with 20 block-wise stationary sources, computed over 100 Monte Carlo trials.

A scenario with sub-gaussian signals is presented in Figure 3.7. Here, the performance achieved through the separation of 10 sources, composed of $Q = 10$ blocks, is shown. The kth signal, $k = 1, \cdots, 10$, is uniformly distributed (with variance one) in the first k blocks and Gaussian elsewhere. Again, the length of the signals

is $N = 10000$ samples and the signals were mixed with a random mixing matrix $\mathbf{A}$. In Figure 3.7 the mean interference-to-signal ratio averaged over 1000 Monte Carlo trials is shown.

Similarly to the previous experiment, this example demonstrates the strongest point of Block EFICA, which consists in its ability to adapt to varying signal distribution. The same performance was achieved by the NSNG algorithm, and it performed even better when a smaller length of data was considered, which is likely owing to the lower number of parameters compared to Block EFICA. However, also in this scenario, NSNG failed to converge in a few trials. To allow the presentation of its performance, the trials where the divergence occurred had to be skipped.

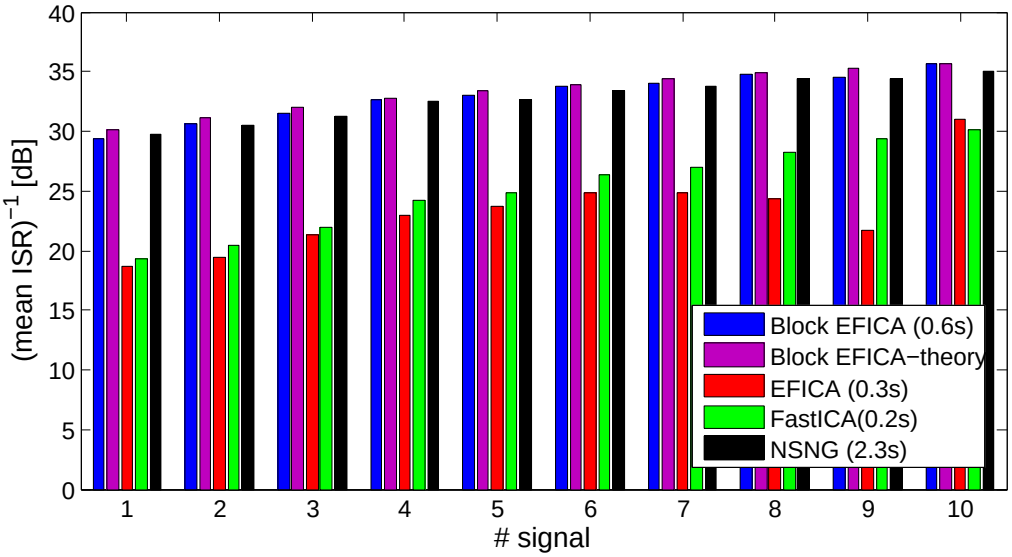


Figure 3.7: Mean interference-to-signal ratio for the separation of 10 sub-Gaussian signals averaged over 1000 Monte Carlo trials.

The same setup as in the previous example was utilized in the experiment which examines the influence of the input parameter Q (the assumed number of blocks within sources) on the performance of Block EFICA. The parameter Q takes values from 1 to 40. The overall performance averaged over all sources is shown in Figure 3.8.

Although the performance is optimal for the correct value of $Q = 10$, the deterioration of the performance due to over- or underestimation of Q is not high. For Q close to 1 the performance of Block EFICA approaches that of EFICA, which is as expected. Certain local maxima can be observed for Q being a multiple of 10, which occurs owing to fitting the boundaries of blocks exactly to the instants where the distributions of signals are switched. Nevertheless, the negligible improvement demonstrates the lower importance of the correct fitting.

The theoretical performance computed using (3.18) monotonically grows with Q. Therefore, it becomes slightly overoptimistic for higher values of Q, because it

does not take the practical effect of overparametrization into account. Nevertheless, it may be used for the selection of an effective value of Q.

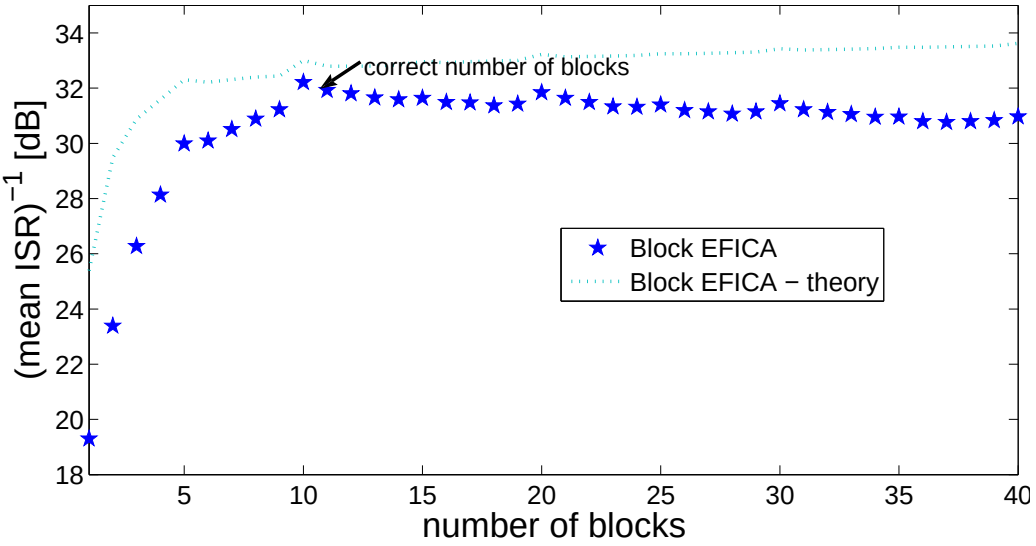


Figure 3.8: Average interference-to-signal ratio for the separation of 10 sub-Gaussian signals achieved by Block EFICA when changing the number of blocks Q considered by the algorithm.

3.5.2 Separation of block-wise stationary sources with changing variance

The experiments shown in this section deal with artificial non-stationary sources with a changing variance. This type of signals does not correspond to the source model of Block EFICA, which assumes a unit variance of signals within all blocks.

The experiments present a scenario where a signal having a variable variance is separated from another signal that is stationary. The first (nonstationary) signal has variances, respectively, equal to 1, σ, and σ^2 in the three consecutive blocks of the same length, and the second signal is Gaussian having a constant variance equal to one. An example of the signals for a particular value of the parameter σ, considered on interval $(0, 1]$, are shown in Figure 3.9.

Two distinct situations which differ in the selected distribution of the first non-stationary signal are considered. In the first setup, the distribution is Gaussian in all blocks. Then, for σ close to one, where the two signals are almost stationary, the mixture cannot be separated due to the Gaussianity of the signals. In the second setup, the distribution is Laplacian, which makes the mixture separable even for σ close to one. The signals can be separated for both cases when σ is close to zero. Then, the first signal is strongly nonstationary and has a different variance-envelope than the second signal, which is the general requirement of the BGL algorithm.

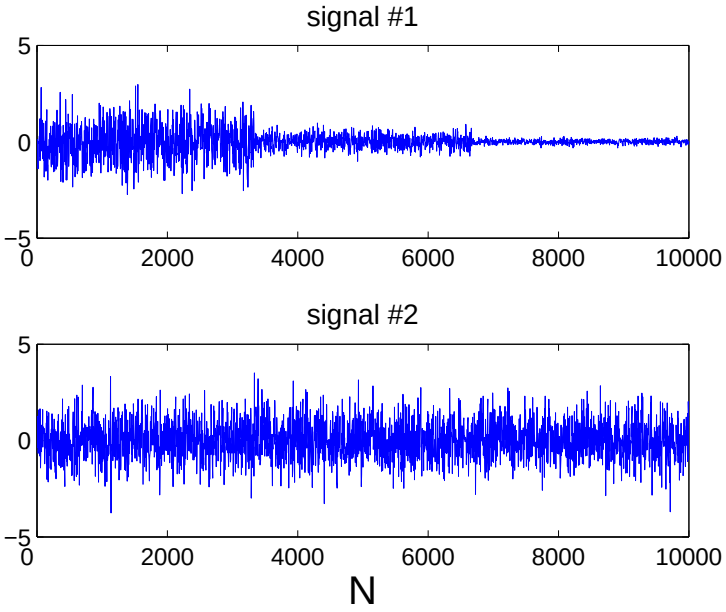


Figure 3.9: Illustration of the (non-)stationary Gaussian signals of length $N = 10^4$ when the parameter σ that controls the nonstationarity of the first signal equals 0.1.

Figures 3.10 and 3.11 show the results obtained for both settings of the experiment. Performance is evaluated for each considered value of the parameter σ separately. It is presented in the form of the mean Interference-to-Signal ratio averaged over 1000 Monte Carlo trials. The line denoted as CRLB represents the Cramér-Rao bound for separation performance.

The first scenario with Gaussian signals fits the Block Gaussian model, where the performance of BGL should be optimal. This fact is confirmed by the results shown in Figure 3.10. Similar performance is achieved by the NSNG algorithm without yielding any instability, which reveals its excellent ability to utilize the nonstationarity of signals in simple examples such as the two-dimensional one considered here.

The proposed Block EFICA algorithm achieves comparable results up to $\sigma \in [0.7, 1]$ where the Gaussian signals are almost stationary, which makes them hardly distinguishable for non-gaussianity-based methods. Hence, the drop in performance is caused by failures of initialization brought on by the Symmetric FastICA in the Step 1 of Block EFICA.

The plots marked by "Block EFICA (identity)" demonstrate further improvement of Block EFICA achieved through including the identity function in the score function estimator. The better performance shows that the option allows a more effective exploitation of the nonstationarity of signals.

The second scenario simulates the case when the original signals exhibit both the non-gaussianity and the non-stationarity since the distribution of the first signal is Laplacian. Here, Block EFICA yields performance that is superior to the other methods. The BGL algorithm suffers from stationarity of the signals as σ is

approaching one. Conversely, the original EFICA does not utilize effectively their nonstationarity for σ close to zero. The implementation of the NSNG algorithm lacks the ability to accurately estimate the score function of the Laplacian distribution. It has significantly lower performance than EFICA and Block EFICA. Nevertheless, its ability to profit both from non-stationarity and non-gaussianity is confirmed.

3.5.3 Separation of real-world speech signals mixed linearly

The proposed example demonstrates a practical contribution of the method, which is shown to be robust when separating real world sources. These sources do not follow the source model of Block EFICA (3.8).

Twenty speech signals of length 5000 samples are mixed linearly with a random mixing matrix $\mathbf{A}$ and subsequently separated by the competing non-gaussianity-based algorithms. The speech signals were selected randomly from a database of utterances, which is along with the source code of the method available online [50].

The above described experiment was performed 1000 times and the results were averaged over all trials and signals. The results are depicted in Figure 3.12 and are quantified in the means of three distinct criteria. The proposed Block EFICA shows improved separation results compared to the original EFICA, specifically 1dB ISR improvement and 2,5dB SIR improvement.

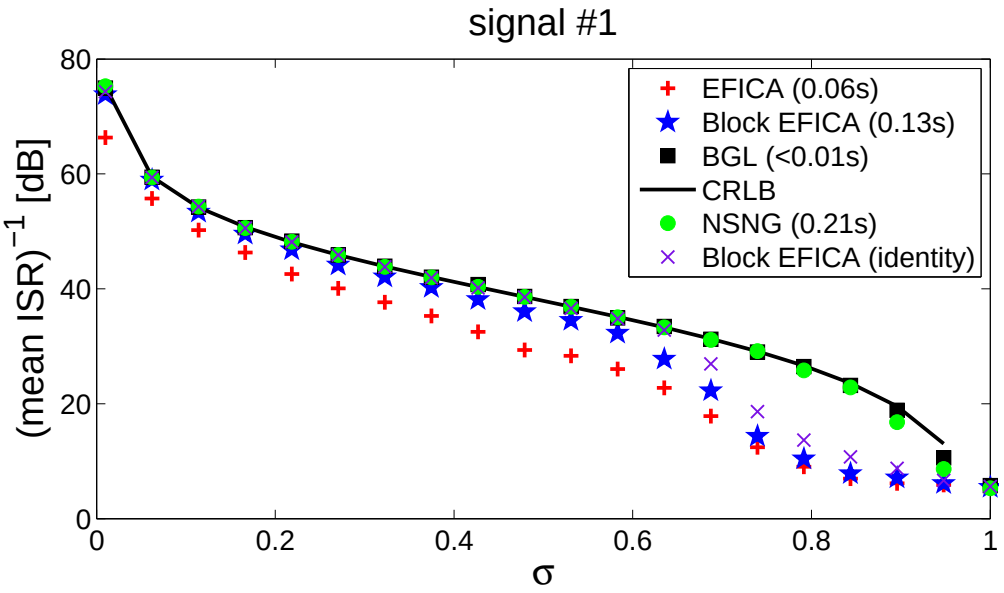
3.5.4 Separation of real-world mixtures of speech signals

In the previous example, real-world data were mixed artificially and subsequently separated. To demonstrate the strength of Block EFICA on true real-world mixtures which are convolutive in nature, an example is presented where a convolutive mixture of two speech signals recorded by two microphones is separated. The experiment consists in the utilization of the Block EFICA algorithm as the ICA separator in the T-ABCD [9] algorithm.

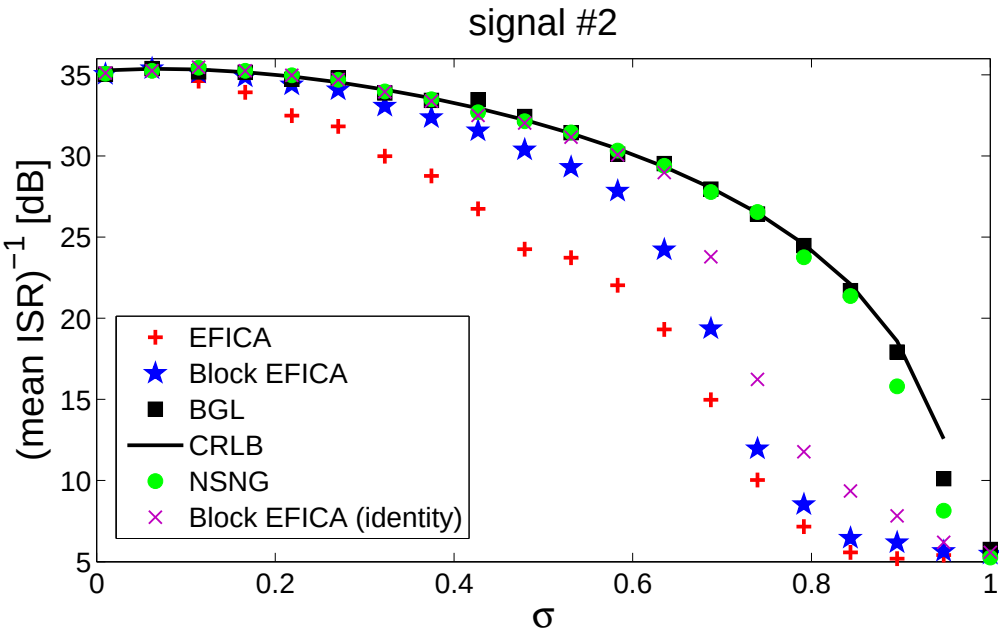
However, the evaluation of an ICA algorithm constructed in this manner is not entirely accurate. The limitations stem from the fact that the ICA performance within T-ABCD is restricted by the desired demixing filter length L and the optimal solution is not unique due to the fact that the estimated signals are retrieved up to an unknown filtering. Despite of these limitations, the experiment reflects well the ability of ICA methods to deal with real-world signals/mixtures.

Figure 3.13 (a) shows data [1] containing real recordings of two speakers (played over loudspeakers) simultaneously saying the digits from one to ten in English and in Spanish, respectively. The loudspeakers were placed close to the microphones (60 cm), so direct-path signals and possibly early reflections from the closest

[1] This data are available online on the web page of T-W. Lee `http://www.cnl.salk.edu/~tewon/ Blind/blind_audio.html`.

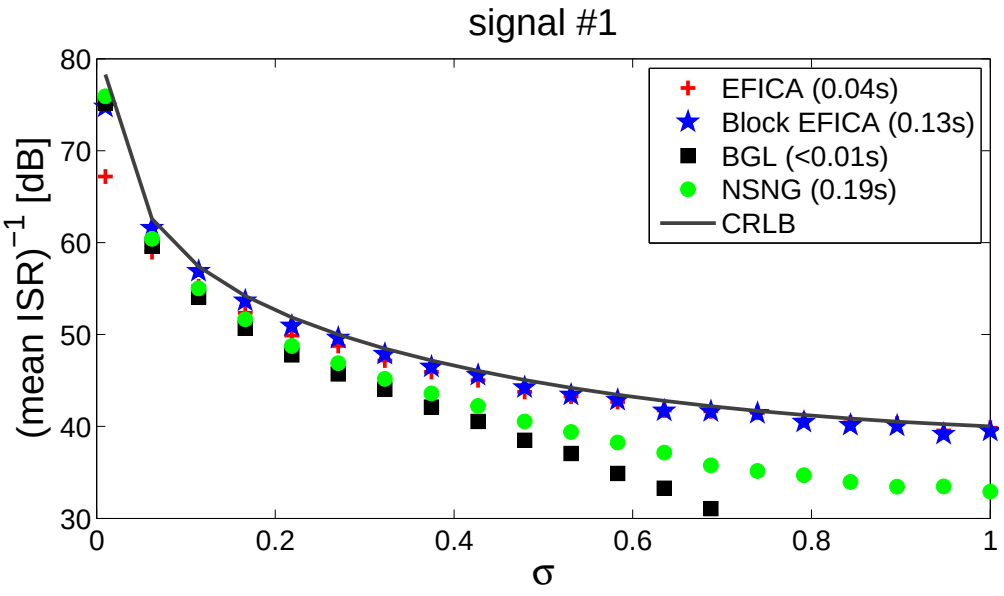


(a)

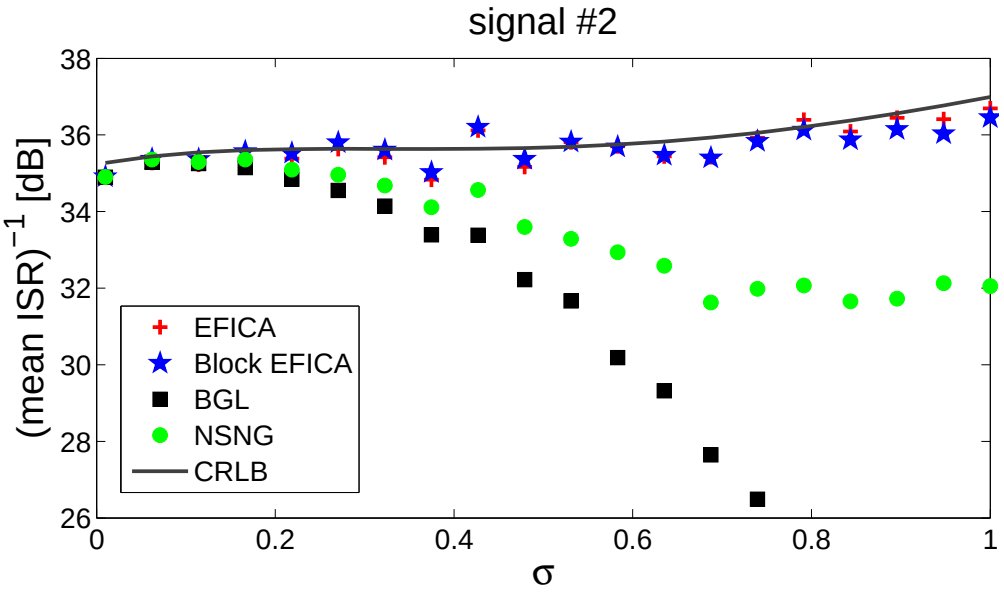


(b)

Figure 3.10: Results of the experiment with nonstationary signals with a changing variance. In this scenario the signal #1 was Gaussian.



(a)



(b)

Figure 3.11: Results of the experiment with nonstationary signals with changing variance. In this scenario the signal #1 was Laplacian.

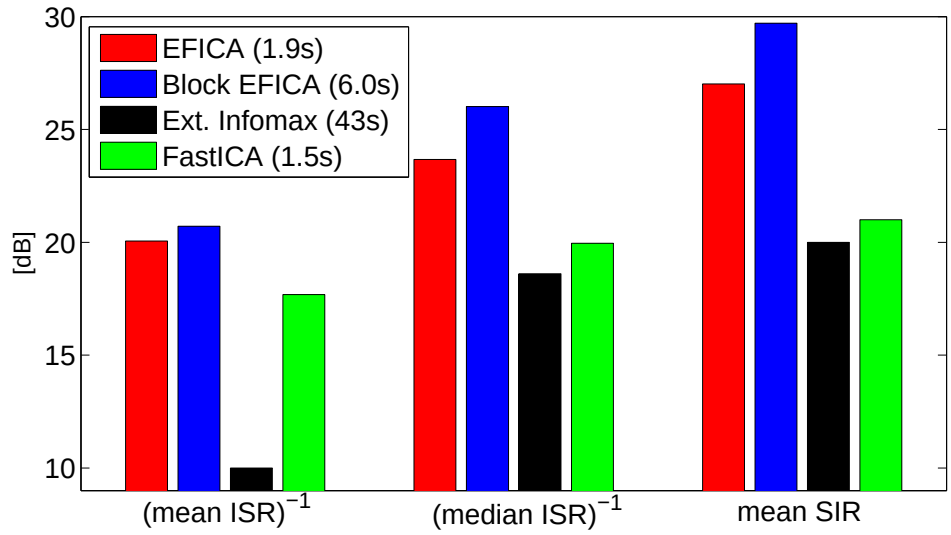


Figure 3.12: Separation of real-world speech signals mixed linearly

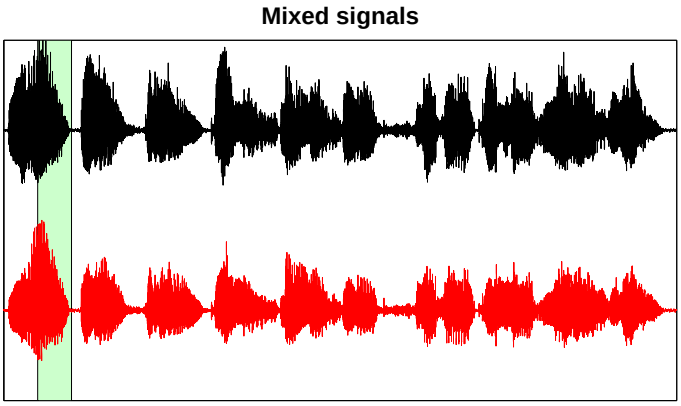
objects are much stronger than other reverberations in the recorded convolutive mixture.

Since the rhythms of the speech signals are similar and synchronized, there occur many short segments (say of length 6000 samples - the sampling frequency is 16kHz) where the dynamics of the speech signals are very close. Owing to possible changing mixing conditions (e.g. moving sources), the aim is to separate as short segments of signals as possible. However, the similar dynamics of sources in short segments cause malfunctioning of nonstationarity-based methods. From this point of view, the methods that use not only the nonstationarity but also the non-gaussianity of speech are more flexible because they do not fail in such situations.

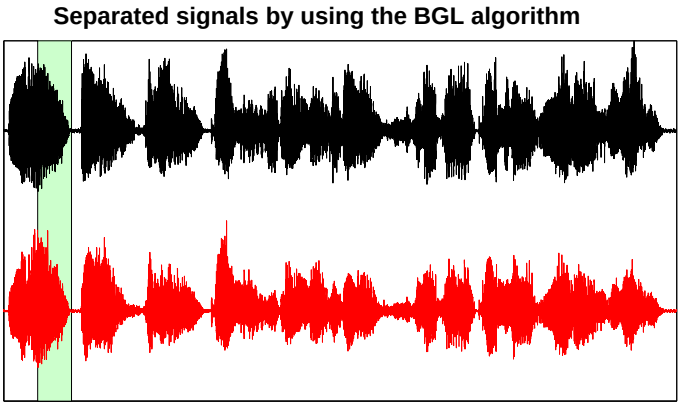
To demonstrate this, Figures 3.13 (b) and 3.13 (c) show the results of separation with $L = 20$ via non-stationarity based BGSEP [32] and Block EFICA, respectively, when only using a short segment of data for the mixture identification (learning data). Then, the resulting separating filters are applied to the whole signals. Since the mixture is stationary here (loudspeakers and microphones remain in their positions during the whole recording), the separated signals reveal the ability of the ICA methods to separate them using data from the given data segment only.

Since the dynamics of signals are too similar in the chosen segment, the nonstationarity-based BGL algorithm yields poorly separated components of $\tilde{\mathbf{X}}$ so that average SIR of the finally separated sources is 3.3dB [2], while the original SIR of the mixed signals is 3.4dB. By contrast, the Block EFICA algorithm succeeded to separate the signals yielding average SIR of 12.2dB, which means a "good" result in this convolutive audio source separation task.

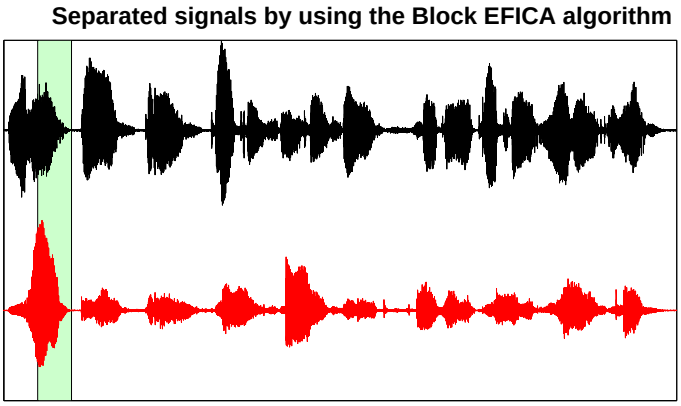
[2]The Signal-to-Interference ratio was evaluated by means of the BSS_EVAL toolbox from [51]. Lee's separated signals were used as the reference true sources.



(a)



(b)



(c)

Figure 3.13: Results of the separation of real-world convolutive mixture of two speech signals recorded by two microphones. Respective ICA methods were applied to the subspace generated by a selected data segment of 6000 samples. The segment is delimited by vertical lines in the graphs.

Chapter 4

Separation of Convolutive Mixtures

The linear convolutive mixture model (1.2) is well suited for audio applications because it takes the propagation of sound in the real environment into consideration. Sound does not reach all microphones at the same time because its propagation speed is limited. This introduces delays into the mixing. Moreover, sound reflects on walls and other obstacles, which introduces multiple echoes of a single source into the mixing process.

For the sake of clarity, let us restate that the convolutive mixture model is given by

$$x_i(n) = \sum_{k=1}^{d} \sum_{\tau=0}^{M_{ik}} a_{ik}(\tau) \cdot s_k(n-\tau), \tag{4.1}$$

where d is the number of estimated sources, m is the number of microphones, $x_1(n) \dots x_m(n)$ are known signals recorded on microphones, $s_1(n) \dots s_m(n)$ are the original unknown sources. The unknown $a_{ik}(\tau)$ represent the source-sensor impulse responses, i.e. impulse responses expressing the propagation of sound from the location of each source to each microphone. The convolutive model is thus a generalization of the instantaneous model, which is obtained by setting $M_{ik} = 0$.

The separation usually proceeds via filtering of the mixtures with some estimated MIMO filter via

$$\hat{s}_k(n) = \sum_{i=1}^{m} \sum_{\tau=0}^{L-1} w_{ki}(\tau) \cdot x_i(n-\tau), \tag{4.2}$$

where L is the length of the demixing MIMO filters w_{ki}.

The estimated sources are determined up to several ambiguities, similar to the ambiguities of ICA. It is not possible to determine the ordering and amplitude of the estimated sources. Moreover, the sources are determined up to an arbitrary filtering caused by the propagation in the environment because the source-sensor impulse responses are unknown. Therefore, the sources are estimated as spatial images (responses) of the sources on microphones, rather then the sources themselves.

The response of the kth source on the ith microphone is given by

$$s_k^i(n) = \sum_{\tau=0}^{M_{ik}} a_{ik}(\tau) \cdot s_k(n-\tau). \tag{4.3}$$

To allow practical separation, a general prerequisite of the mutual independence of unknown sources is often assumed. This allows the utilization of an ICA method as

a separator. Prior to the utilization of ICA, the convolutive mixing model needs to be transformed into an instantaneous one. There exist two basic approaches for this transformation. It can be performed either in the time-domain or in the frequency domain.

4.1 Separation via ICA in the time-domain

The transformation in the time-domain expresses discrete convolution via matrix multiplication. Two matrices $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{X}}$, which have (usually, not necessarily) the block-Toeplitz structure, are introduced. The rows of these matrices represent delayed source signals or mixtures, respectively. The matrices have the size $mL \times (N - L + 1)$, where L is the desired length of the demixing filter. The linear space spanned by the rows of $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{X}}$ is called source space or observation space, respectively. The matrices have following structure:

$$\tilde{\mathbf{S}} = \begin{bmatrix} s_1(L) & \dots & \dots & s_1(N) \\ s_1(L-1) & \dots & \dots & s_1(N-1) \\ \vdots & \vdots & \vdots & \vdots \\ s_1(1) & \dots & \dots & s_1(N-L+1) \\ s_2(L) & \dots & \dots & s_2(N) \\ s_2(L-1) & \dots & \dots & s_2(N-1) \\ \vdots & \vdots & \vdots & \vdots \\ s_d(1) & \dots & \dots & s_d(N-L+1) \end{bmatrix}, \tag{4.4}$$

and

$$\tilde{\mathbf{X}} = \begin{bmatrix} x_1(L) & \dots & \dots & x_1(N) \\ x_1(L-1) & \dots & \dots & x_1(N-1) \\ \vdots & \vdots & \vdots & \vdots \\ x_1(1) & \dots & \dots & x_1(N-L+1) \\ x_2(L) & \dots & \dots & x_2(N) \\ x_2(L-1) & \dots & \dots & x_2(N-1) \\ \vdots & \vdots & \vdots & \vdots \\ x_m(1) & \dots & \dots & x_m(N-L+1) \end{bmatrix}. \tag{4.5}$$

Using this notation, the convolutive mixing model in (4.1) may be approximated via

$$\tilde{\mathbf{X}} = \tilde{\mathbf{A}}\tilde{\mathbf{S}}, \tag{4.6}$$

where $\tilde{\mathbf{A}}$ is defined like

$$\tilde{\mathbf{A}} = \begin{bmatrix} a_{11}(0) & a_{11}(1) & \dots & a_{11}(L-1) & a_{12}(0) & \dots & \dots & a_{1d}(L-1) \\ 0 & a_{11}(0) & a_{11}(1) & \dots & a_{11}(L-1) & a_{12}(0) & \dots & a_{1d}(L-2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{11}(0) & 0 & 0 & \dots & a_{1d}(0) \\ a_{21}(0) & a_{21}(1) & \dots & a_{21}(L-1) & a_{12}(0) & \dots & \dots & a_{2d}(L-1) \\ 0 & a_{21}(0) & a_{21}(1) & \dots & a_{21}(L-1) & a_{22}(0) & \dots & a_{2d}(L-2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{m1}(0) & 0 & 0 & \dots & a_{md}(0) \end{bmatrix}. \tag{4.7}$$

The approximation gets more accurate with L approaching to M_{ik}. However, even for high values of L, (4.6) is not exact. The reason is the truncation of impulse responses located near the bottom rows of blocks in $\tilde{\mathbf{A}}$. The equality can be achieved for rectangular $\tilde{\mathbf{A}}$ only (see [52]).

With this approximation, it is possible to estimate the demixing filter coefficients contained in matrix $\widehat{\mathbf{W}}$ via an ICA method described in Chapter 2. The independent components $\mathbf{C}$, obtained via $\mathbf{C} = \widehat{\mathbf{W}}\tilde{\mathbf{X}}$, correspond to the original sources up to an unknown order and filtering.

The time-domain generalization (4.6) has a major drawback: a rapid growth of dimension when expanding mixtures $x_i(n), i = 1 \dots d$ into $\tilde{\mathbf{X}}$. The suitable filter length L differs according to application, but can reach hundreds of samples. The ICA problem dimension grows in the same rate, i.e. by factor mL.

Various methods performing the separation of convolutive mixtures via ICA in the time domain have been proposed in the recent years. The methods can be classified based on the fact whether they perform *complete* or *partial* decomposition of the observation space $\tilde{\mathbf{X}}$ into subspaces corresponding to the estimated sources.

The partial decomposition consists in the estimation of one dimensional subspaces (components). These components are subsequently utilized for the reconstruction of the source responses on the microphones. The partial decomposition approach makes the problem of rapid dimension growth of $\tilde{\mathbf{X}}$ easier. On the other hand, the performance of the methods doing it depends largely on the initialization of the applied convergence scheme. These methods may find two components related with a single source and skip another source completely. From this point of view, the complete decomposition is a more reliable choice at the cost of a higher computational burden.

A natural gradient based method performing partial decomposition was proposed by Amari et al. in [53]. A more robust extension of this algorithm was proposed by Douglas et al. in [54].

The complete decomposition with constraints lowers the computational demands of the methods by introducing some simplifying assumptions about the decomposing transform or the observation space. One of such constraints is the block-Toeplitz or block-Sylvester structure of the matrix $\tilde{\mathbf{A}}$ in (4.7).

A generalization of the ICA algorithm SOBI for convolutive mixtures, performing complete constrained decomposition, was proposed by Belouchrani et al. in [55].

A time-domain algorithm, using both non-stationarity and spectral diversity of the sources mixed with a convolutive transform, was proposed in [56] by Buchner et al.

The utilization the Joint Block Diagonalization (for details see [55]) on the observation space with an applied orthogonal constraint was proposed by Févotte et al. in [52].

The complete constrained decomposition assumes that all estimated subspaces have the same dimension, which may be a possible drawback. **The complete unconstrained decomposition** need not assume this and therefore provides a way to utilize the available data as effectively as possible. In the general form stated in this section, the complete decomposition may be considered to be computationally too demanding, however.

A modular algorithm performing complete the unconstrained decomposition was proposed by Koldovský and Tichavský in [9] and subsequent extensions. In the papers, the authors propose several improvements to the common decomposition scheme, which allow to circumvent partially its computational burden. The algorithm is described in Section 4.3.

4.2 Separation via ICA in the frequency-domain

Another effective possibility for the separation of convolutive mixtures via ICA constitutes the separation in the frequency-domain. It consists in the application of the Fourier Transform (FT) on the mixtures. This procedure exploits the basic property of FT; the convolution in the time-domain is mapped to the multiplication in the frequency domain. The application of FT on both sides of (4.1) gives

$$X_i(\omega) = \sum_{j=1}^{d} A_{ij}(\omega) \cdot S_i(\omega), \tag{4.8}$$

where $X_i(\omega)$, $A_{ij}(\omega)$ and $S_i(\omega)$ are the Fourier transforms of $x_i(n)$, $a_{ij}(n)$ and $s_i(n)$. The application of FT in the time-domain transforms the convolutive mixtures into instantaneous mixtures in the frequency-domain. The mixing matrix $\mathbf{A}$ is a function of ω in (4.8), unlike the instantaneous model (1.1), where it is constant.

In practice, the computation of FT is performed through the Short Time Fourier Transform (STFT). The frames of the mixtures are weighted by a window function (like Hamming or Gauss Window) and the Fast Fourier Transform (FFT) is applied on each frame. Subsequently, the computed spectra $X_i(\omega)$ are split according to the frequencies ω into segments and ICA is applied on each such spectral segment (for each ω) separately. The separation is complicated by the fact, that the mixtures $X_i(\omega)$ and the demixing matrix $\mathbf{W}(\omega)$ are complex valued.

The separation in the frequency domain has a major drawback called the *Permutation Problem*. The ambiguities of ICA (the order and signs of rows in $\mathbf{W}$) still

apply for the estimated matrices $\mathbf{W}(\omega)$. In each frequency bin, the order may differ and it is necessary to re-order the rows of $\mathbf{W}(\omega)$ so that the respective independent components have the same order in all the spectral segments, which is vital for the reconstruction in the time-domain.

An advantage of frequency based methods lies in the ability of computing long separating filters, which is very favorable in audio applications. On the other hand, the computation of such filters requires long recordings, in order to generate sufficient amount of data for each frequency bin (for details see [57]).

The utilization of various instantaneous ICA techniques for the separation of convolutive mixtures in the frequency domain was proposed e.g. by Smaragdis in [58] and by Mitianoudis et al. in [59].

A robust way how to deal with the permutation problem was proposed e.g. by Sawada et al. in [60] or Nesta et al. in [61].

4.3 Time-domain Audio Source Blind Separation Based on the Complete Decomposition

This section introduces the algorithm designed for separation of convolutive mixtures, which was proposed by Koldovský and Tichavský in [9] and later extended in [62]. The modifications of this method proposed by the author are described separately in Chapter 5. The algorithm will be denoted as T-ABCD further in the text, which stands for Time-domain Audio source Blind separation based on the Complete Decomposition. The T-ABCD is based on unconstrained separation via ICA in the time-domain according to (4.6).

The algorithm consists of five consecutive steps. In the first step, the convolutive model from (4.1) is transformed into an instantaneous one (4.6). The matrix $\tilde{\mathbf{X}}$ (or its generalized form) is constructed from the known mixtures $x_i(n)$. The second step consists in the application of an arbitrary ICA method on $\tilde{\mathbf{X}}$ and its decomposition into independent components (ICs). The goal of the third step is to compute a measure of similarity among ICs. This measure forms the basis for the clustering of ICs into groups corresponding to original sources. The clustering is performed in the fourth step of the algorithm. Finally, the responses of the sources on microphones are reconstructed, based on the determined clusters. A mono channel estimate of each source can be formed if desired. Each part will now be described in details.

4.3.1 Construction of observation subspace

The T-ABCD algorithm starts with the construction of the observation subspace $\tilde{\mathbf{X}}$ and the transformation of the convolutive model (4.1) into instantaneous one (4.6). Two distinct ways of the construction were proposed for the utilization within T-ABCD. The T-ABCD algorithm assumes that the number of sources d is equal to the number of sensors m.

The common construction of matrix $\tilde{\mathbf{X}}$, similar to most time-domain BSS

methods, is done according to (4.5). This approach has a serious limitation tied with the computational demands of the algorithm. In order to increase the separating filter length by one, m rows need to be added into $\tilde{\mathbf{X}}$. Therefore, the dimension of $\tilde{\mathbf{W}}$ grows linearly with L.

The generalized construction of $\tilde{\mathbf{X}}$ was proposed in the paper [63]. It allows (in theory) to compute separating filters of infinite length L, without increasing the dimension of $\tilde{\mathbf{X}}$. A generalized observation subspace is for a given set of invertible filters $f_{i,\ell}$ defined as

$$\tilde{\mathbf{X}} = \begin{bmatrix} \{f_{1,1} \star x_1\}(1) & \dots & \dots & \{f_{1,1} \star x_1\}(N) \\ \{f_{1,2} \star x_1\}(1) & \dots & \dots & \{f_{1,2} \star x_1\}(N) \\ \vdots & \vdots & \vdots & \vdots \\ \{f_{1,L} \star x_1\}(1) & \dots & \dots & \{f_{1,L} \star x_1\}(N) \\ \{f_{2,1} \star x_2\}(1) & \dots & \dots & \{f_{2,1} \star x_2\}(N) \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \{f_{m,L} \star x_m\}(1) & \dots & \dots & \{f_{m,L} \star x_m\}(N) \end{bmatrix}, \tag{4.9}$$

where $\star$ denotes the convolution operator.

The linear combinations of rows of $\tilde{\mathbf{X}}$ defined in this way correspond to outputs of MIMO filters with a generalized feed-forward structure, which were introduced in [64]. The filters $f_{i,\ell}$ are referred to as *eigenmodes*. The generalized observation subspace defined in (4.9) coincides with the one in (4.5) when $f_{i,\ell} = \delta(n - \ell + 1)$, where $\delta(n)$ stands for the unit impulse function. In this case $f_{i,\ell}$ realizes a backward time-shift by $\ell - 1$ samples. The multiplication of $\tilde{\mathbf{X}}$ defined via (4.9) with $\tilde{\mathbf{W}}$ allows T-ABCD to apply a long separating filter (even IIR) without increasing L.

In practice, the authors of T-ABCD propose to select the eigenmodes as the Laguerre filters, which have the feed-forward structure (see [64] for details). The Laguerre filters $f_{i,\ell}$ are for all microphones (omitting the lower index i) defined recursively through their transfer functions F_ℓ as

$$F_1(z) = 1, \tag{4.10}$$

$$F_2(z) = \frac{\mu z^{-1}}{1 - (1 - \mu) z^{-1}}, \tag{4.11}$$

$$F_n(z) = F_{n-1}(z) G(z), \quad n = 3, \dots, L, \tag{4.12}$$

where

$$G(z) = \frac{(\mu - 1) + z^{-1}}{1 - (1 - \mu) z^{-1}}, \tag{4.13}$$

and μ is a free parameter which takes values from $(0, 2)$.

The construction of $\tilde{\mathbf{X}}$ through Laguerre eigenmodes includes (4.5) as a special case for $\mu = 1$ (because $F_2(z) = G(z) = z^{-1}$ and $f_2(n) = g(n) = \delta(n - 1)$). This is

the only case where the Laguerre filters are FIR of the length L. For $\mu \neq 1$, these filters are always IIR.

The effective length of the Laguerre filters L_* is defined as the minimum length needed to capture 90% of the total energy contained in the impulse response. For the Laguerre filters, according to [65], it is approximately

$$L_* = (1 + 0.4|\mu - 1| \log_{10} L)L/\mu. \tag{4.14}$$

It can be seen that $L_* > L$ for $\mu < 1$ and vice versa.

4.3.2 Decomposition of the observation subspace via ICA

The independent component analysis can be performed via an arbitrary algorithm described in Section 2.4. Two algorithms were implemented as the ICA separators within T-ABCD, the EFICA algorithm ([24], Section 2.9) and the BGSEP algorithm ([32], Section 2.10).

The result of ICA is the estimate of a demixing matrix $\tilde{\mathbf{W}}$ denoted as $\widehat{\mathbf{W}}$ and the independent components given by $\mathbf{C} = \widehat{\mathbf{W}}\tilde{\mathbf{X}}$. The ith row of $\mathbf{C}$ is denoted as $\mathbf{c}_i$ and its nth element as $c_i(n)$.

In the case of perfect separation and when the rows of $\tilde{\mathbf{S}}$ are truly independent, the independent components represent the attenuated and delayed versions of the original sources. However, the rows of $\tilde{\mathbf{S}}$ in (4.4) are not independent, due to the temporal structure of the audio/speech sources. In this case, the components become arbitrary filtered copies of the unknown sources, i.e. the sources estimated from components exhibit random coloration.

The number of components is greater than the number of sources, due to dimension of $\tilde{\mathbf{X}}$. Therefore, it is possible to find groups of components where all the components are related to a single original source. These groups are found via clustering, which is based on the mutual similarity of components, and are exploited for the final reconstruction of the sources.

4.3.3 Mutual similarity of components

Two different similarity measures were proposed for utilization within the T-ABCD algorithm. The first measure consists in the projections of component $\mathbf{c}_i$ on the subspace spanned by the delayed component $\mathbf{c}_j$ (see [9]). The other measure is based on generalized correlation coefficients known as GCC-PHAT (see [66]).

The projection approach is based on the following similarity measure, where the similarity $\mathbf{D}_{ij}$ between components $\mathbf{c}_i$ and $\mathbf{c}_j$ is defined as

$$\mathbf{D}_{ij} = \hat{\mathrm{E}}\{\mathbf{P}_i\mathbf{c}_j\}^2 + \hat{\mathrm{E}}\{\mathbf{P}_j\mathbf{c}_i\}^2, \tag{4.15}$$

where $\hat{\mathrm{E}}$ denotes the sample mean operator and $\mathbf{P}_i$ is a projection on subspace $\mathbf{C_i}$ spanned by the delays of component $\mathbf{c}_i$, i.e.

$$\mathbf{C_i} = \begin{bmatrix} c_i(1) & \dots & c_i(2*L+1) \\ \vdots & \vdots & \vdots \\ c_i(N-2*L) & \dots & c_i(N) \end{bmatrix}. \tag{4.16}$$

The operator of projection on subspace $\mathbf{C_i}$ is given by

$$\mathbf{P}_i = \mathbf{I} - \mathbf{C}_i(\mathbf{C}_i^T\mathbf{C}_i)^{-1}\mathbf{C}_i^T. \tag{4.17}$$

The generalized correlation approach consists in the computation of generalized-correlation coefficients known as GCC-PHAT [66]. These coefficients are invariant to the magnitude spectra of the signals and depend on their phase spectra only, which makes them appropriate for similarity evaluation since the ICs have random magnitude spectra.

Let $C_i(k)$ and $C_j(k)$ 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 $g_{ij}(n)$, are equal to the inverse Fourier transform of

$$G_{ij}(k) = \frac{C_i(k) \cdot C_j(k)^*}{|C_i(k)| \cdot |C_j(k)|}, \tag{4.18}$$

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

If the components correspond exactly to the same source, i.e. without any residual interference, $g_{ij}(n)$ is equal to the 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

$$\mathbf{D}_{ij} = \sum_{n=-L}^{L} |g_{ij}(n)| \qquad i, j = 1, \ldots, mL,\ i \neq j, \tag{4.19}$$

The diagonal elements of $\mathbf{D}$ have no importance for the clustering and are all set to one.

4.3.4 Clustering of independent components

Within the original prototype of T-ABCD from [9], the clustering of independent components is done via Standard Agglomerative Hierarchical Non-overlapping algorithm (SAHN) with average linking strategy [67].

The SAHN algorithm produces so-called **Hard Partition**; it consists in computation of the affiliation matrix $\mathbf{U}$ of size $d \times mL$, which describes the affiliation of each component to each cluster. Each element of $\mathbf{U}$ takes values 0 or 1 depending on the fact whether the component belongs to the cluster or not. Indeed, in the case of perfect separation, each component is completely affiliated with one of the clusters (sources) only.

In the case of imperfect separation though, each component contains the interference of all sources to a certain degree. This fact is better expressed by a fuzzy clustering technique where the elements of $\mathbf{U}$ take values from the close range [0,1]. This type of affiliation is called the **Fuzzy Partition**. The utilization of a fuzzy clustering technique within T-ABCD was proposed by the author and is described in details in Chapter 5.

4.3.5 The reconstruction of source responses on microphones

The reconstruction aims at transforming the clustered components into source responses on microphones.

First, **the observation subspace is reconstructed** according to the affiliation of the components to the kth cluster via

$$\widehat{\mathbf{S}}_k = \widehat{\mathbf{W}}^{-1} diag\{\mathbf{\Lambda}_{k1}, \dots \mathbf{\Lambda}_{k(mL)}\} \widehat{\mathbf{W}} \tilde{\mathbf{X}}, \tag{4.20}$$

where the coefficients $\mathbf{\Lambda}_{kj}$ are suitable positive weights which are computed from the values contained in the hard partition matrix $\mathbf{U}$.

When a perfect separation is assumed [9], then the **binary weights** are set to

$$\mathbf{\Lambda}_{kj} = \mathbf{U}_{kj}, \tag{4.21}$$

i.e. the weights of components affiliated with the source are equal to 1 and the weights of the non-affiliated components are equal to 0.

In practice, the separation is rarely perfect, e.g. due to an insufficient demixing filter length L. Here, the idea of **fuzzy weights** [62] appears to be suitable. Each component is affiliated with each of the sources to a certain degree. The clusters are still determined based on the SAHN algorithm, but the non-negative weights are determined via

$$\mathbf{\Lambda}_{kj} = \left(\frac{\sum_{i \in K_k, i \neq j} \mathbf{D}_{ij}}{\sum_{i \notin K_k, i \neq j} \mathbf{D}_{ij}} \right)^{\alpha}, \tag{4.22}$$

where K_k contains the indices of components affiliated with the kth cluster and α is a free parameter which determines the "fuzziness" of the weights. With growing parameter α the fuzziness of the weights diminishes.

Subsequently, the estimates of source **responses on microphones are computed**.

In the case of **the common construction** of $\tilde{\mathbf{X}}$ according to (4.5), the estimate of the response of the kth source on the ith microphone is acquired by

$$\hat{s}_k^i(n) = \frac{1}{L} \sum_{\ell=1}^{L} (\widehat{\mathbf{S}}_k)_{(i-1)L+\ell, n+\ell-1}, \tag{4.23}$$

where $\widehat{\mathbf{S}}_{\alpha,\beta}$ is the $\alpha\beta$th coefficient of $\widehat{\mathbf{S}}$.

In the case of **the generalized construction** of $\tilde{\mathbf{X}}$ according to (4.9) the estimation proceeds in the following manner: let $f_{i,\ell}^{-1}$ be the inverse of filter $f_{i,\ell}, \ell = 1 \dots L$. Then the estimate of the response of the kth source on the ith microphone is acquired by

$$\hat{s}_k^i(n) = \frac{1}{L} \sum_{\ell=1}^{L} \{ f_{i,\ell}^{-1} \star (\widehat{\mathbf{S}}_k)_{(i-1)L+\ell,*} \}(n), \tag{4.24}$$

where $\widehat{\mathbf{S}}_{\alpha,*}$ is the αth row of $\widehat{\mathbf{S}}$. The reconstruction via (4.24) coincides with (4.23) if $f_{i,\ell}(n) = \delta(n - \ell + 1)$.

Responses $\widehat{s}_k^i(n)$ can be exploited for the computation of a single channel estimate $\widehat{s}_k(n)$. This is done via the **Delay and Sum Beamformer**. It chooses a single reference response of the kth source. The cross-correlation functions of the reference response with the other ones are computed. The maxima of these functions correspond to the mutual delay of the responses. The single channel estimate is formed by the summation of the respectively delayed responses.

Chapter 5

Author's Modifications of the T-ABCD Algorithm

This chapter describes several modifications of the algorithm for the blind audio source separation called T-ABCD (see Section 4.3) that were derived and implemented by the author of this thesis. The results of this chapter were published in paper [68].

The proposed changes consist in the utilization of a fuzzy clustering technique for the clustering of ICs (T-ABCD, Step 4). Since the separation performed by ICA is not perfect, the components cannot be simply clustered into groups corresponding to the original sources. It is more plausible to assume that the components are affiliated to a certain extent with all sources. These degrees of affiliation can be established through fuzzy clustering and utilized in the process of source reconstruction.

First, a comparison of several fuzzy clustering methods is presented. The most suitable method for utilization within T-ABCD is selected based on experimental results. Subsequently, the separation performance of the algorithm modified by fuzzy clustering is compared with the original separation technique with hard clustering. It is shown that fuzzy clustering improves the performance of T-ABCD. Finally, the modified algorithm is tested in an experiment which deals with the speech enhancement. The results are verified through an automatic speech recognition system developed for the Czech language on the Institute of Information Technology and Electronics by the Technical University of Liberec.

5.1 Fuzzy clustering of the components

As stated in the introduction of the chapter, the separation performed by ICA is not perfect. The independent components contain some residual interference. This causes that each component is (to a certain extent) affiliated with more than one source.

The clustering of ICs within the original T-ABCD neglects this fact. It computes the hard partition (see Section 4.3.4), which is subsequently transformed into fuzzy weights $\mathbf{\Lambda}$ via an ad-hoc formula (4.22).

The approach proposed in this section applies the idea of component affiliation to multiple sources/clusters directly to the clustering step. It replaces the SAHN [67] clustering algorithm, computing the *hard partition* by a method which produces the *fuzzy partition*.

Let a fuzzy partition be described by the affiliation matrix $\mathbf{U}$ which is given by the set of equations

$$\begin{aligned} &\mathbf{U}_{kj} \in [0,1] \\ &\textstyle\sum_{k=1}^{c} \mathbf{U}_{kj} = 1 \\ &\textstyle\sum_{j=1}^{K} \mathbf{U}_{kj} > 0, \end{aligned} \tag{5.1}$$

where K is the number of clustered objects/components and c is the number of clusters.

In general, the clustering techniques can be divided according to the way they represent the clustered objects. There are two basic possibilities [69]: the first one is referred to as *the object description*. Here, each object is described by a feature vector of an arbitrary length. This vector contains information which describes the object unambiguously. The other possibility is called *the relational description*. Here, the individual properties of objects are not available. Instead, the data are described as a whole, by a matrix of mutual similarities (or distances) of size $K \times K$.

In T-ABCD, a suitable measure for similarity of ICs is defined via projections (4.15) or GCC-PHAT coefficients (4.19). This approach leads to a matrix of pairwise similarities $\mathbf{D}$, i.e. to a relational description.

The following relational clustering methods were implemented for utilization in the T-ABCD algorithm: RFCMdd - Relational fuzzy c-medoids [70], RPCM - Relational possibilistic c-means [71] and RFCM - Relational fuzzy c-means [71]. RFCM was found experimentally (see Section 5.4.1) to be the most suitable method.

5.2 Relational Fuzzy C-Means Algorithm

The RFCM algorithm is a relational version of the well known clustering method Fuzzy c-means (FCM) [72]. The input of the FCM algorithm consists of K objects to be clustered into a known number of clusters c. The objects are described by feature vectors $\mathbf{y}_1 \dots \mathbf{y}_K$. The FCM seeks the optimum matrix $\mathbf{U}$ of object affiliations via minimization of the objective function

$$J_f(\mathbf{U}, \mathbf{G}) = \sum_{k=1}^{c} \sum_{j=1}^{K} (\mathbf{U}_{kj})^f \mathbf{V}_{kj}, \tag{5.2}$$

where c is the number of clusters and $f > 1$ is a *fuzzyfication* parameter which determines to what extent is a clustered object affiliated with more than one cluster. Further, $\mathbf{G} = [\mathbf{g}_1 \dots \mathbf{g}_c]$ is a matrix of cluster prototypes which are computed as average feature vectors according to

$$\mathbf{g}_k = \sum_{j=1}^{K} (\mathbf{U}_{kj})^f \mathbf{y}_j / \sum_{j=1}^{K} (\mathbf{U}_{kj})^f \quad k = 1 \dots c \tag{5.3}$$

and $\mathbf{V}$ is a matrix of Euclidean distances between prototypes and clustered objects, i.e.

$$\mathbf{V}_{kj} = \|\mathbf{g}_k - \mathbf{y}_j\|^2 . \tag{5.4}$$

The algorithm is iterative. First, it is initialized by a random fuzzy partition $\mathbf{U}$ satisfying (5.1), number of clusters c and the fuzzyfication parameter f. Subsequently, it alternates the prototypes updates (5.3) and the recomputation of affiliation matrix $\mathbf{U}$ via

$$\mathbf{U}_{kj} = \Big(\sum_{i=1}^{c} (\mathbf{V}_{kj}/\mathbf{V}_{ij})^{1/(f-1)}\Big)^{-1}, \tag{5.5}$$

until convergence is achieved.

The RFCM algorithm allows the utilization of FCM on relational data where the feature vectors $\mathbf{y}_1 \dots \mathbf{y}_K$ are not known and the cluster prototypes cannot be explicitly expressed by (5.3). Here, the data are described by matrix $\mathbf{B}$ of pair-wise distances (dissimilarities) between the clustered objects. RFCM assumes the distances in $\mathbf{B}$ to be Euclidean. The RFCM expresses the object-prototype distances $\mathbf{V}_{kj}$ as functions of matrices $\mathbf{U}$ and $\mathbf{B}$ according to

$$\mathbf{V}_{kj} = (\mathbf{B}\boldsymbol{\mu}_k)_j - \frac{1}{2}\boldsymbol{\mu}_k^T \mathbf{B}\boldsymbol{\mu}_k, \tag{5.6}$$

where

$$\boldsymbol{\mu}_k = [(\mathbf{U}_{k1})^f, \dots, (\mathbf{U}_{kK})^f]^T / \sum_{j=1}^{K} (\mathbf{U}_{kj})^f. \tag{5.7}$$

In the paper [73] is proven that if the dissimilarities in $\mathbf{B}$ are computed via $\mathbf{B}_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|^2$ then the distances $\mathbf{V}_{kj}$ computed via (5.4) and (5.6) are equal. Moreover, the sequences of partitions $\mathbf{U}$ produced by FCM and RFCM using (5.5) are identical.

5.3 RFCM applied in T-ABCD

The following text describes the details of RFCM implementation in T-ABCD and the utilization of the output affiliation matrix $\mathbf{U}$ for the reconstruction of the sources.

5.3.1 Similarity/dissimilarity transforms

The application of RFCM in T-ABCD starts with the computation of the matrix of pair-wise distances (dissimilarities) between components $\mathbf{B}$. The distances are calculated from the matrix of similarities $\mathbf{D}$ given by (4.15) or (4.19). Note that there are $K = mL$ objects/components to be clustered.

The elements of $\mathbf{D}$ are positive with range $[0, \infty]$. In this general case, it is possible to compute $\mathbf{B}$ according to a **reciprocal transform**

$$\mathbf{B}_{ij} = \begin{cases} 1/\mathbf{D}_{ij} & i \neq j \\ 0 & i = j. \end{cases} \tag{5.8}$$

Another possibility for the dissimilarity computation gives the normalization of the entries of $\mathbf{D}$ to range $[0, 1]$ by $\mathbf{D} = \mathbf{D}/\max\{\mathbf{D}_{ij}\}$ and the computation of weights

according to a **subtractive transform**

$$\mathbf{B}_{ij} = \begin{cases} 1 - \mathbf{D}_{ij} & i \neq j \\ 0 & i = j. \end{cases} \tag{5.9}$$

The experiments presented in Section 5.4.2 show that the performance of RFCM is invariant to the choice of the similarity/dissimilarity transform. Either of these transforms are suitable for utilization in T-ABCD.

5.3.2 Spreading transformation

The RFCM algorithm requires the dissimilarities $\mathbf{B}$ to be Euclidean. However, the measures of similarity (4.15) or (4.19) do not possess this property. Thus, the object-prototype distances $\mathbf{V}$ from (5.6) might be negative. In order to avoid this and to guarantee the convergence of RFCM, the *Spreading Transform* [71] is applied to $\mathbf{V}$ and $\mathbf{B}$.

The spreading transform consists in the addition of a positive number φ to off-diagonal elements of $\mathbf{B}$ in the case that $\mathbf{V}_{kj} < 0$ for any $k = 1 \dots c, j = 1 \dots mL$, i.e.

$$\mathbf{B}_{ij} \leftarrow \begin{cases} \mathbf{B}_{ij} + \varphi & i \neq j \\ 0 & i = j. \end{cases} \tag{5.10}$$

The exact computation of φ involves a computationally expensive eigenvalue problem. On the other hand, the usage of a gross overestimate of φ may result in the distortion of $\mathbf{B}$ and a possible loss of cluster information. The paper [71] proposes a sufficiently accurate approximation of φ. It uses the by-products of the calculation of $\mathbf{V}$, which lowers the computational demands. The computation proceeds as follows.

If $\mathbf{V}_{kj} < 0$ for any $k = 1 \dots c, j = 1 \dots mL$ then

$$\varphi = \max\{-2(\mathbf{V}_{kj})/(\|\boldsymbol{\mu}_k - \mathbf{e}_j\|^2)\}, \tag{5.11}$$

where $\mathbf{e}_j$ is a $mL \times 1$ vector of zeros with the jth element equal one. Subsequently, the substitution

$$\mathbf{V}_{kj} \leftarrow \mathbf{V}_{kj} + \varphi \|\boldsymbol{\mu}_k - \mathbf{e}_j\|^2 \tag{5.12}$$

is performed for $k = 1 \dots c, j = 1 \dots mL$ along with (5.10).

5.3.3 Reconstruction of sources via the affiliations of ICs

The output matrix of fuzzy affiliations $\mathbf{U}$ is used in the reconstruction step within the modified T-ABCD algorithm. The reconstruction is still performed via (4.20), however the weights $\boldsymbol{\Lambda}$ are determined according to one of the following approaches.

The indirect application: The fuzzy partition $\mathbf{U}$ is transformed into a hard affiliation by assigning each component to the cluster with the highest affiliation degree. The weights $\boldsymbol{\Lambda}$ are computed (as in original T-ABCD) via (4.22).

This approach exploits fuzzy clustering for the determination of the clusters only. The weights are computed according to a formula from the original version of T-ABCD, which is known to be suitable for the reconstruction of sources.

The modified application: The weights $\mathbf{\Lambda}$ are defined by

$$\mathbf{\Lambda}_{kj} = \left(\frac{\mathbf{U}_{kj}}{1 - \mathbf{U}_{kj}} \right)^{\alpha}, \tag{5.13}$$

where α denotes a positive adjustable parameter. We select $\alpha = 2$.

This approach fully exploits the information contained in the fuzzy partition $\mathbf{U}$. It is designed to emphasize the ICs with the highest affiliation to the cluster.

The direct application: The fuzzy affiliations $\mathbf{U}_{ij}$ are applied directly as weights to the components, i.e.

$$\mathbf{\Lambda}_{kj} = (\mathbf{U}_{kj})^{\alpha}. \tag{5.14}$$

The experiments presented in Section 5.4.3 reveal that the direct application of the fuzzy affiliations $\mathbf{U}$ leads to the best separation results from all considered possibilities.

5.4 Experiments

The following experiments deal with the comparison of the considered fuzzy clustering techniques. The RFCM algorithm [71] is shown to achieve the best results among the competing methods.

The ICA separation is performed in all experiments in this section via the BGSEP algorithm [32].

In this section, similarities $\mathbf{D}$ are computed via the GCC-PHAT coefficients (4.18). The calculation of generalized correlations is computationally less demanding than the computation of projections. Moreover, (4.18) can be easily evaluated blockwise, which is exploited in the online version of T-ABCD.

Our choice of the fuzzyfication parameter is $f = 1.5$.

5.4.1 Clustering of independent components

The following experiment compares the clustering results of relational fuzzy clustering techniques with the results of the SAHN algorithm. The algorithms were exploited for the clustering of ICs in T-ABCD. The similarities $\mathbf{D}$ were transformed into distances $\mathbf{B}$ via the reciprocal transform (5.8).

For the purposes of this experiment, an ideal hard clustering method was designed, which serves as a reference for the comparison. The reference clustering is based on the idea that the component is assigned to a cluster/source where it achieves the highest SIR value. These SIR values can be computed if the true responses of the sources on microphones are known. Let $\widehat{\mathbf{w}}_i$ be the ith row of matrix

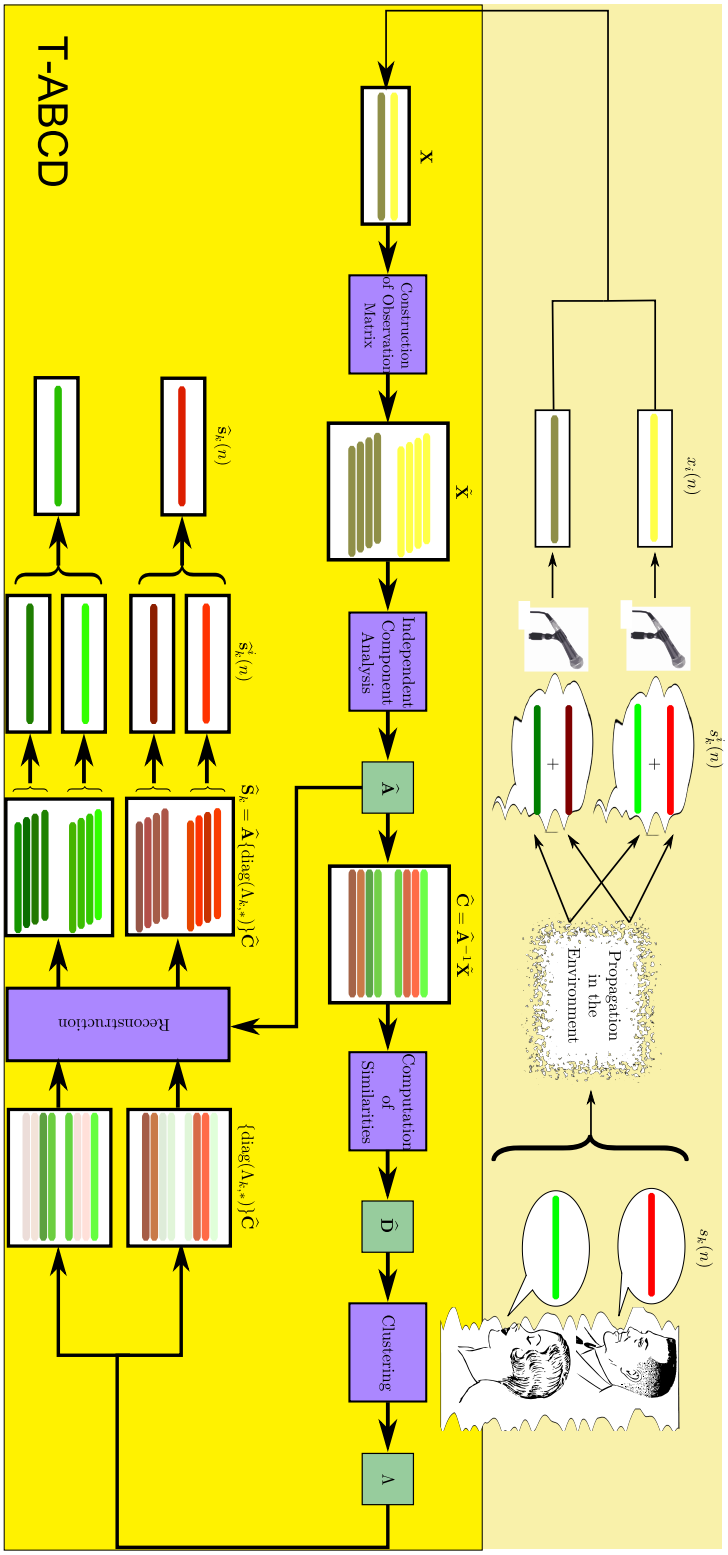


Figure 5.1: The mixing of the utterances in the environment and the flowchart of the T-ABCD algorithm with implemented fuzzy clustering.

$\widehat{\mathbf{W}}$, then SIR of this component subject to the kth source is equal to

$$SIR_i^k = \frac{(\widehat{\mathbf{w}}_i\tilde{\mathbf{S}}_k)(\widehat{\mathbf{w}}_i\tilde{\mathbf{S}}_k)^T}{(\widehat{\mathbf{w}}_i(\tilde{\mathbf{X}}-\tilde{\mathbf{S}}_k))(\widehat{\mathbf{w}}_i(\tilde{\mathbf{X}}-\tilde{\mathbf{S}}_k))^T}, \tag{5.15}$$

i.e. as energy of the kth source in the ith component divided by the energy of other sources in this component. The matrix $\tilde{\mathbf{S}}_k$ exhibits a similar structure to $\tilde{\mathbf{X}}$. It is constructed from the responses of the kth source on microphones when the other sources are silent, i.e.

$$\tilde{\mathbf{S}}_k = \begin{bmatrix} s_k^1(L) & \dots & \dots & s_k^1(N) \\ s_k^1(L-1) & \dots & \dots & s_k^1(N-1) \\ \vdots & \vdots & \vdots & \vdots \\ s_k^1(1) & \dots & \dots & s_k^1(N-L+1) \\ s_k^2(L) & \dots & \dots & s_k^2(N) \\ s_k^2(L-1) & \dots & \dots & s_k^2(N-1) \\ \vdots & \vdots & \vdots & \vdots \\ s_k^m(1) & \dots & \dots & s_k^m(N-L+1) \end{bmatrix}, \tag{5.16}$$

The experimental data consisted of two distinct sounds (replayed on loudspeakers) recorded on eight microphones in a room depicted in Figure 5.2. Five times two sources were mixed together. There were six distinct sources available, including two male voices, two female voices, a typewriter sound and a Gaussian noise.

For the experiment, two of the eight microphone recordings were selected in order to obtain the determined mixtures. There are $7 \cdot 8/2 = 28$ possibilities how to do so. When all five recorded source combinations are considered, it gives $5 \cdot 7 \cdot 8/2 = 140$ different separation scenarios, which means $140 \cdot 2L$ clustering decisions.

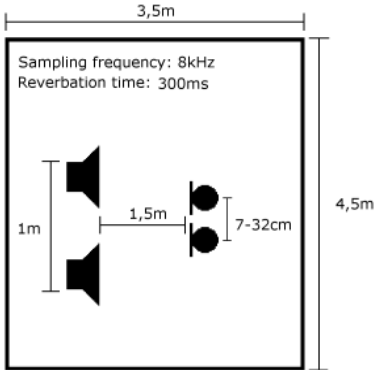


Figure 5.2: Scheme of the room where the mixtures were recorded.

Assuming that the reference clustering described above gives correct decisions, these decisions were compared to the ones obtained by the SAHN method. To allow the comparison with the results of fuzzy techniques, the fuzzy partition was reduced

into a hard one. Each component was assigned to a cluster/source where it achieved the highest fuzzy affiliation.

Table 5.1 shows the number of incorrect decisions obtained by the methods for various separating filter lengths L. The experiment shows that RFCM achieves a lower number of incorrect assignments compared to all other algorithms for all available filter lengths L.

	L=16	L=21	L=26	L=31
SAHN	610	782	988	1260
RFCM	599	770	958	1232
RFCMdd	763	1087	1458	2108
RPCM	1530	1971	2520	3048
Total decisions	4480	5880	7280	8680

Table 5.1: The number of incorrect component assignments compared to ideal clustering

During the experiment, the computational burden of implemented techniques was measured in terms of the time necessary to complete all 140 clustering tasks. The experiment was performed in Matlab 7.9 on a PC with a double-core 2,66GHz processor and 2GB RAM. The results are shown in Table 5.2. As can be seen, the iterative fuzzy algorithms are almost five times faster than SAHN.

	L=16	L=21	L=26	L=31
SAHN	13.34	13.01	13.42	14.53
RFCM	2.77	2.85	2.91	2.97
RFCMdd	2.36	2.36	2.38	2.55
RPCM	3.70	3.74	3.77	3.85

Table 5.2: The time (in seconds) necessary to accomplish all 140 clustering tasks

5.4.2 Comparison of similarity/dissimilarity transformations

This section investigates how the computation of dissimilarity influences the results of the clustering. In Section 5.3, two possible similarity into dissimilarity transformations were described, namely the reciprocal transform (5.8) and the subtractive transform (5.9).

To determine which transform is better suited for clustering of components, the experiment from Section 5.4.1 and the ideal clustering defined there were exploited. For each of the 140 clustering scenarios, the similarities among components $\mathbf{D}$ were computed via generalized correlations (4.19). Subsequently, two distance matrices $\mathbf{B}$ were computed via both mentioned transforms. Finally, the components were

clustered via implemented fuzzy algorithms and the results were compared with the ideal clustering. The filter length was set to $L = 26$.

The numbers of incorrect decisions are summarized in Table 5.3. The experiment indicates that the partition determined by RFCM and RFCMdd is rather invariant to the choice of the dissimilarity transform. On the other hand, the number of incorrect assignments computed by RPCM is lower by 22% when the chosen transformation is subtractive.

(Total 7820)	RFCM	RFCMdd	RPCM
Subtractive	942	1482	1986
Reciprocal	958	1458	2520

Table 5.3: The number of incorrect clustering decisions. Comparison of two formulas for the dissimilarity matrix $\mathbf{B}$ computation.

5.4.3 Separation results of T-ABCD with hard/fuzzy clustering

The experiment compares the separation results of T-ABCD with the hierarchical clustering and the results of modified T-ABCD with RFCM. Again, the data from Section 5.4.1 were separated. All available 140 combinations of sources and microphone distances were considered. The data were sampled at 16kHz. The similarities $\mathbf{D}$ were transformed into distance through the subtractive transform (5.9). The inner parameters of T-ABCD were as follows: $L = 26, N = 6000, \alpha = 2, f = 1.5$. The reconstruction weights were within the original T-ABCD with SAHN computed via (4.22).

The separation quality was evaluated via the BSS_EVAL toolbox [51] 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).

The experiment compares various possibilities for the computation of the reconstruction weights $\mathbf{\Lambda}$. In Section 5.3 following variants were proposed: The indirect utilization of the fuzzy partition $\mathbf{U}$, the modified utilization via (5.13) and the direct application. The results, averaged over all scenarios and sources, are shown in Table 5.4.

The experiment suggests that the separation results of T-ABCD with hierarchical clustering and T-ABCD with indirectly determined weights are nearly identical. This means both clustering techniques are similarly successful when clustering the ICs, because they determine the demixing weights in the same manner.

Slightly better results are obtained by the direct application of the affiliations $\mathbf{U}$ to reconstruction weights.

The modified computation of the reconstruction weights outputs estimates which are characterized by high values of SIR and low values of SAR. The direct listening to

	SIR[dB]	SDR[dB]	SAR[dB]
SAHN	7.56	5.74	12.69
RFCM - Indirect	7.53	5.72	12.71
RFCM - Direct	7.73	5.98	13.12
RFCM - Modified	11.92	5.35	7.11

Table 5.4: The separation results of the original T-ABCD with hard clustering and the modified version with implemented fuzzy clustering. Three formulas for the computation of the reconstruction weights $\mathbf{\Lambda}$ from the fuzzy partition $\mathbf{U}$ are considered.

the estimated signals uncovers that the signals are fairly well separated but heavily distorted.

It can be concluded that the utilization of RFCM within T-ABCD and the direct application of affiliations $\mathbf{U}$ as reconstruction weights result in a slight improvement of the separation results. Moreover, the RFCM algorithm is less computationally demanding than SAHN (provided that the number of clusters/sources is known) and has a favorable iterative computation scheme. This feature is advantageous when non-stationary mixing is considered and the necessity of demixing filter updates in time arises. This topic is discussed further in the chapter which describes the online version of T-ABCD.

5.4.4 Utilization of modified T-ABCD for speech enhancement

A large series of experiments is disscused that prove the ability of the modified T-ABCD to separate speech from interfering noise. Speech signals employed in the experiment were taken from the European database of recorded broadcast news that was collected in the COST278 action in 2003 [74] and later also in 2005. The database contains complete recordings of TV news in 10 European languages. The Czech part of the database was exploited in order to be able to perform speech recognition by means of the Czech Automatic Speech Recognition (ASR) system. The exploited ASR system [75] was developed by the Institute of Information Technology and Electronics, Technical University in Liberec. The test set included 653 utterances taken from 9 Czech broadcast news shows. They represented a large variety of spoken data, from clear studio speech of professional speakers to spontaneous utterances recorded in very noisy conditions. The total number of words in the test set was 10,322. When transcribed by the ASR from [75], the overall recognition rate for this set was 81.02%.

Two types of interference were simulated: 1) a test utterance mixed with a Gaussian noise and 2) a test utterance mixed with another utterance (from the same data set).

Each mixture obeys the convolutive mixing model (4.1). The mixing of two sources recorded on two microphones is considered $d = 2, m = 2$. The convolving filters were randomly generated for each mixture so that $a_{ij}(\tau)$ has the Gaussian

distribution with a zero mean and variance τ^{-2}. The length of the filters was 2000 taps, which corresponds to 125 ms of reverberation time at a 16kHz sampling rate. Each mixture was separated by a filter of length $L = 21$ that was estimated by the BSS algorithm using only segment of 6000 samples where both the speech and the interfering signal were active. The remaining inner parameters of T-ABCD were $\alpha = 2$ and $f = 1.5$. The similarities between ICs were computed via projections (4.15) and transformed into distances via (5.8).

The mixed utterances as well as the separated ones were sent to the recognizer and the recognition performance was evaluated in terms of accuracy defined as

$$Acc = 100(C - D - I - S)/C \tag{5.17}$$

which is computed through a comparison of a reference text with the recognized one. Here, C is the number of words in the reference text, D is the number of deletions, I is the number of insertions, and S denotes the number of substitutions.

In the two series of experiments mentioned above, different mixing conditions with respect to SNR were simulated. In the first one, a Gaussian noise of varying power was added to each test utterance. In the other one, interfering speech with a varying gain was added. For each value of SNR, a recognition accuracy value was obtained from the ASR system. After that, the proposed BSS method was applied to separate the utterances back. They were sent to the ASR system again to evaluate the accuracy achieved for the enhanced signals. The level of enhancement received for selected SNR values can be observed from the diagrams in Figure 5.3 and Figure 5.4.

Let us discuss, for example, the 10 dB SNR case. It can be seen that the addition of a Gaussian noise reduced the recognition accuracy (from the original value 81.02 %) to 35.16 %, but after applying the BSS method, the score was increased to 63.13 %. In case of added speech and the same 10 dB SNR situation, the method helped to enhance recognition accuracy from 44.57 % to 52.52 %.

In general, the results confirm a significant improvement of the recognition rate after applying the separation algorithm, especially, in cases of low SNR ($\approx$ 0dB). A slight decline of accuracy was observed for very high SNR (30dB). This is caused by some distortion of the separated signal due to a higher value of the parameter α in (5.14). For instance, by taking $\alpha = 1$ instead of the default value $\alpha = 2$, accuracy grows from 76.13% to 77.73%.

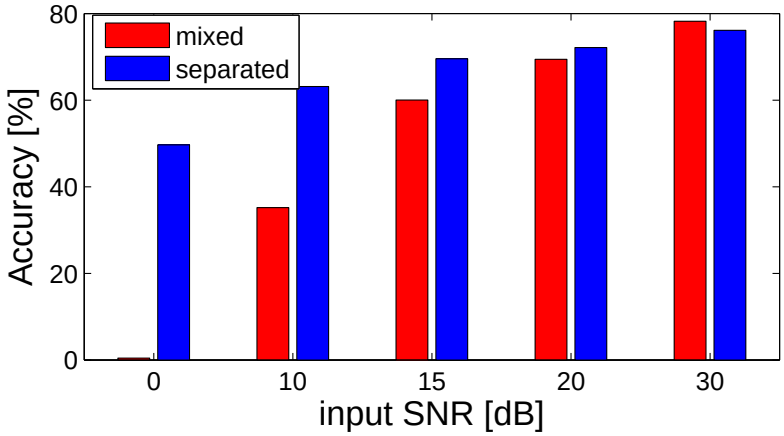


Figure 5.3: Accuracy of the ASR system for speech mixed with Gaussian noise

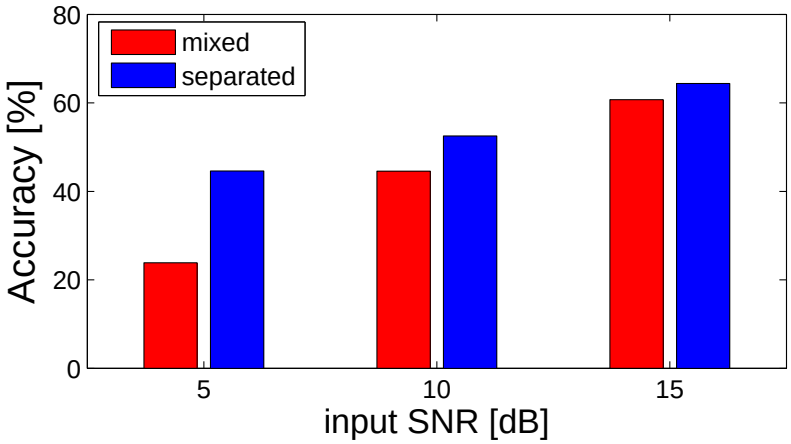


Figure 5.4: Accuracy of the ASR system for speech mixed with other speech

Chapter 6

The Online T-ABCD algorithm

This chapter is concerned with an adaptive algorithm for the blind separation of audio sources via ICA in the time-domain. The proposed algorithm is based on the T-ABCD method described in Section 4.3 and will be further denoted as *online T-ABCD*. The algorithm was published in [76].

The motivation for designing an online version of T-ABCD is the ability of the batch T-ABCD to estimate the demixing filters using short data segments only. Moreover, many improvements proposed for the batch method can be exploited in the online version as well, e.g. the generalized construction of the observation subspace from (4.9).

Although the mixing process considered by online T-ABCD is potentially dynamic, e.g. due to moving sources, it is assumed that it changes slowly and may be considered stationary within short data segments. Therefore, during a mixture interval of length P, the classical convolutive mixing problem, which is described by (4.1), is considered. The original sources are assumed to be independent, which allows ICA to be the basis of the separation. More specifically, the fast converging ICA algorithm BGSEP is utilized for the decomposition. The online T-ABCD algorithm assumes that the number of sources d remains the same throughout the whole recording.

The basic idea of online T-ABCD consists in the block-wise utilization of its batch counterpart on the stationary segments of the mixtures. The batch variant is modified in such a way that it respects the continuity of the mixing process by introducing memory into the algorithm. The demixing filters are updated via a single iteration of ICA and clustering in each segment. The adaptivity of the algorithm is controlled by learning parameters, which control e.g. the convergence speed of the BGSEP method.

The algorithm is shown to achieve good separation results with relatively short demixing filters ($L \approx 30$). The online version outperforms its batch counterpart (in the terms of average output SIR) even when the source positions are fixed, because it is capable of better adaptation to the non-stationarity of the audio signals.

6.1 Description of the algorithm

The input of online T-ABCD consists of m mixture signals, each of length N. The proposed algorithm starts with the segmentation of these signals into overlapping blocks of length P, with the shift of T samples such that $R = P/T$ is an integer. The overlap length of two consecutive blocks is thus $P - T$.

The Ith block of the jth input signal will be denoted by

$$x_j^I(n) = x_j((I-1)\cdot T + n), \qquad n = 1, \ldots, T. \tag{6.1}$$

The uppercase superscript I will be used to denote the 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 batch variant, the online procedure first constructs some form of the observation matrix. For the sake of simplicity, let us assume that the traditional form (4.5) is utilized. Subsequently, it (I) applies a simplified BGSEP algorithm to decompose the data matrix into its independent components and (II) uses a relational 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.

6.1.1 Step I: Independent Component Analysis via Simplified BGSEP algorithm

A simplified version of BGSEP is utilized as a separator within the online T-ABCD algorithm. The original BGSEP ([32], Section 2.10) is a second order statistics based ICA algorithm. It utilizes a general AJD scheme, incorporating arbitrary weight matrices, called WEDGE (Weighted Exhaustive Diagonalization with Gauss itErations). Based on the choice of the weight matrices, a specific algorithm is formed. BGSEP and its weight matrices are optimized for sources which have a Gaussian distribution and are piecewise stationary.

Unlike Section 2.10, the superscript I denotes here the index of data segment, not the BGSEP iteration index. In fact, only a single iteration of the simplified BGSEP is performed in each segment of the mixed data.

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

$$\mathbf{X}^I = \begin{bmatrix} x_1^I(1) & x_1^I(2) & \ldots & \ldots & x_1^I(P) \\ x_1^I(2) & x_1^I(3) & \ldots & \ldots & x_1^I(P+1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^I(L) & x_1^I(L+1) & \ldots & \ldots & x_1^I(P+L) \\ x_2^I(1) & x_2^I(2) & \ldots & \ldots & x_2^I(P) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m^I(L) & x_m^I(L+1) & \ldots & \ldots & x_m^I(P+L) \end{bmatrix}, \tag{6.2}$$

where L is a free integer parameter, corresponding to the desired demixing filter length.

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

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

$$\mathbf{X}^I = [\mathbf{X}^{I,1}, \ldots, \mathbf{X}^{I,M}]. \tag{6.3}$$

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

$$\mathbf{R}^{I,k} = \frac{M}{P}\mathbf{X}^{I,k}(\mathbf{X}^{I,k})^T, \qquad k = 1, \ldots, M. \tag{6.4}$$

For the sake of convenience and computational savings, it is assumed that the number of matrices M is equal to 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 the removed covariance matrix $\mathbf{R}^{I-1,1}$ and added matrix $\mathbf{R}^{I,M}$.

The diagonalization proceeds by performing a single WEDGE iteration [32] with the diagonal weight matrices that are optimized for the case when the signals obey the gaussian piecewise stationary model. The algorithm uses the latest available demixing matrix $\mathbf{W}^{I-1}$ to partially diagonalize the matrices in (6.4)

$$\mathbf{P}^{I,k} = \mathbf{W}^{I-1}\mathbf{R}^{I,k}(\mathbf{W}^{I-1})^T \qquad k = 1, \ldots, M. \tag{6.5}$$

The demixing matrix $\mathbf{W}^I$ is obtained by updating $\mathbf{W}^{I-1}$ as

$$\mathbf{W}^I = (\mathbf{A}^I)^{-1}\mathbf{W}^{I-1}, \tag{6.6}$$

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

$$\begin{bmatrix} \mathbf{A}^I_{kl} \\ \mathbf{A}^I_{lk} \end{bmatrix} = \beta_1^I \begin{bmatrix} \mathbf{r}_{ll}^T\mathbf{K}[kl]\mathbf{r}_{ll} & \mathbf{r}_{kk}^T\mathbf{K}[kl]\mathbf{r}_{ll} \\ \mathbf{r}_{kk}^T\mathbf{K}[kl]\mathbf{r}_{ll} & \mathbf{r}_{kk}^T\mathbf{K}[kl]\tilde{\mathbf{r}}_{kk} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_{ll}^T\mathbf{K}[kl]\mathbf{r}_{kl} \\ \mathbf{r}_{kk}^T\mathbf{K}[kl]\mathbf{r}_{kl} \end{bmatrix}, \tag{6.7}$$

where $\mathbf{r}_{kl}$ is a $R \times 1$ vector given by

$$\mathbf{r}_{kl} = [(\mathbf{P}^{I,1})_{kl}, \ldots, (\mathbf{P}^{I,M})_{kl}]^T, \tag{6.8}$$

and $\mathbf{K}[kl]$ is a $R \times R$ matrix given by

$$\mathbf{K}[kl] = \operatorname{diag}\left(\frac{1}{(\mathbf{P}^{I,1})_{kk}(\mathbf{P}^{I,1})_{ll}}, \ldots, \frac{1}{(\mathbf{P}^{I,M})_{kk}(\mathbf{P}^{I,M})_{ll}}\right) \tag{6.9}$$

for $k, l = 1, \ldots, mL$, $k > l$. The variable β_1^I in (6.7) does not exist in the original BGSEP algorithm. It is added here to control the convergence speed of the algorithm. The choice of β_1^I will be discussed later.

6.1.2 Step II: Clustering of independent components

Due to the indeterminacy of ICA, the ICs of $\mathbf{X}^I$ are arbitrarily filtered versions of the original signals. There are groups of components which correspond to one of the sources and form thus independent subspaces within $\mathbf{C}^I$. These groups are determined via clustering of the ICs. Prior to clustering, a similarity measure among the components needs to be established, which becomes the basis for the clustering.

Similarity of ICs is quantified via the generalized cross-correlation coefficients known as GCC-PHAT [66]. These 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, since the magnitude spectra of the ICs are random and contain no useful information.

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 let k denote the frequency index. The GCC-PHAT coefficients of the components, denoted by $g_{ij}^I(n)$, are equal to the inverse Fourier transform of

$$G_{ij}^I(k) = \frac{C_i^I(k) \cdot C_j^I(k)^*}{|C_i^I(k)| \cdot |C_j^I(k)|}, \tag{6.10}$$

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 the 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 $\mathbf{D}$ can be computed according to

$$\mathbf{D}_{ij}^I = \sum_{n=-L}^{L} \left| g_{ij}^I(n) \right| + \beta_2 \cdot \mathbf{D}_{ij}^{I-1}, \qquad i, j = 1, \ldots, mL,\ i \neq j, \tag{6.11}$$

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.

The clustering algorithm utilized in online T-ABCD is the Relational Fuzzy C-Means algorithm (RFCM) from [71], which allows tracking of continual changes of the clusters. For the sake of 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 $\mathbf{D}^I$.

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 $\mathbf{\Lambda}_{kj}^I$ be the kjth element of a $d \times mL$ partition matrix $\mathbf{\Lambda}^I$ and represents the affiliation of the jth component to the kth cluster. By definition it holds that $\sum_{k=0}^{d} \mathbf{\Lambda}_{kj}^I = 1$.

Now, let $\mathbf{B}$ denotes the dissimilarity matrix, whose elements are $\mathbf{B}_{ij}^I = 1/\mathbf{D}_{ij}^I$ for $i \neq j$ and $\mathbf{B}_{ii}^I = 0$. Let $\boldsymbol{\mu}_k^{I,f}$ be a $mL \times 1$ vector defined as $\boldsymbol{\mu}_k^{I,f} =$

$[(\mathbf{\Lambda}^I_{k1})^f, \ldots, (\mathbf{\Lambda}^I_{k(mL)})^f]^T / \sum_{j=1}^{mL} (\mathbf{\Lambda}^I_{kj})^f$ called the prototype of the kth cluster associated with a "fuzzyfication" parameter f, $f > 1$. (The experimentally determined value $f = 1.5$ is used). The transition of $\mathbf{\Lambda}^{I-1}$ to $\mathbf{\Lambda}^I$ is given by one iteration of RFCM given by

$$\mathbf{\Lambda}^I_{kj} = \Big(\sum_{i=1}^{d} (\mathbf{V}_{kj} / \mathbf{V}_{ij})^{1/(f-1)} \Big)^{-1}, \tag{6.12}$$

where

$$\mathbf{V}_{kj} = \big(\mathbf{B}^I \boldsymbol{\mu}_k^{(I-1),f} \big)_j - \frac{1}{2} \big(\boldsymbol{\mu}_k^{(I-1),f} \big)^T \mathbf{B}^I \, \boldsymbol{\mu}_k^{(I-1),f} \tag{6.13}$$

is the distance of the jth component to the prototype $\boldsymbol{\mu}_k^{I,f}$ (for details see [71]).

6.1.3 Step III: Reconstruction

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

$$\widehat{\mathbf{S}}^I_k = (\mathbf{W}^I)^{-1} \mathrm{diag}\big[(\mathbf{\Lambda}^I_{k1})^\alpha, \ldots, (\mathbf{\Lambda}^I_{k,mL})^\alpha \big] \, \mathbf{C}^I, \tag{6.14}$$

where α is an adjustable positive parameter. This matrix has an analogous structure as $\mathbf{X}^I$ in (6.2). In an ideal case $\widehat{\mathbf{S}}^I_k$ contains delayed microphone responses of the kth estimated source only. The response of the kth source at the ith microphone is therefore estimated by

$$\hat{s}^{i,I}_k(n) = \frac{1}{L} \sum_{q=1}^{L} \big(\widehat{\mathbf{S}}^I_k \big)_{(i-1)L+q, n+q-1}, \tag{6.15}$$

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 online 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.

6.1.4 Implementation details

The ICA convergence speed can be driven through the parameter β^I_1 in (6.7). 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_1 can be close to zero to maintain continuity. Therefore, we take

$$\beta^I_1 = (1 - \gamma^{I-1})/2. \tag{6.16}$$

γ^I is the Silhouette index [77] of the hard clustering, which is derived from the fuzzy clustering. Let K_k be the set of indices of the components for which $\mathbf{\Lambda}^I_{kj} = \max_\ell \mathbf{\Lambda}^I_{\ell j}$ (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

$$\gamma^I_i = \frac{\min_{j \notin K_k} (\mathbf{B}^I_{ij}) - \frac{1}{|K_k - 1|} \sum_{j \in K_k, i \neq j} \mathbf{B}^I_{ij}}{\max\{ \min_{j \notin K_k} (\mathbf{B}^I_{ij}), \frac{1}{|K_k - 1|} \sum_{j \in K_k, i \neq j} \mathbf{B}^I_{ij} \}}. \tag{6.17}$$

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.

6.2 Experiments

This section presents various experiments which either explain the selection of various design aspects of online T-ABCD or evaluate the separation performance of the proposed algorithm.

All of the experiments in this section were evaluated via the BSS_EVAL toolbox [51] (excluding the example with pseudoconvolutive mixtures).

6.2.1 Comparison of online T-ABCD with block-wise applied batch T-ABCD

The experiment compares online T-ABCD and its batch counterpart applied block-wise on the analyzed mixtures. It aims at proving that the proposed changes, which are designed to respect the continuity of the data, improve the average performance of the algorithm.

The example is designed as follows. Real-world mixtures of speech signals are separated. The data are publicly available on Hiroshi Sawada's web page.[1] The recordings consist of *two simultaneously active sources recorded by two microphones.* The mixtures are 7s long, sampled by 8kHz, recorded in a room with an impulse response of 130 ms using omni-directional microphones.

Online T-ABCD was applied to the mixtures and estimated the source responses on microphones $\hat{s}_k^i(n)$. Batch T-ABCD was applied on blocks of the data segmented in the same manner as would be done by its online counterpart and estimated the source spatial images in each such block. The inner parameters of both T-ABCD versions were: $P = 6144$, $T = 512$, $L = 30$, $\alpha = 3$ and $f = 1.5$.

The performance was evaluated block-wise and was averaged over both sources. The results of the experiment as a function of a segment number are shown in Figures 6.1, 6.2. The average performance is presented in Table 6.1.

Online T-ABCD is designed to respect the continuity of the sources. It uses the information learned in previous data segments to separate the signals in the current one. This can be done even when the separation is not possible in the current segment, e.g. when the sources do not follow the assumption of non-stationarity necessary for BGSEP. The performance of the original batch may deteriorate when the method is applied on such a data block, whereas the online T-ABCD is able to perform the separation. Due to this fact, the online method outperforms the batch version applied block-wise on average.

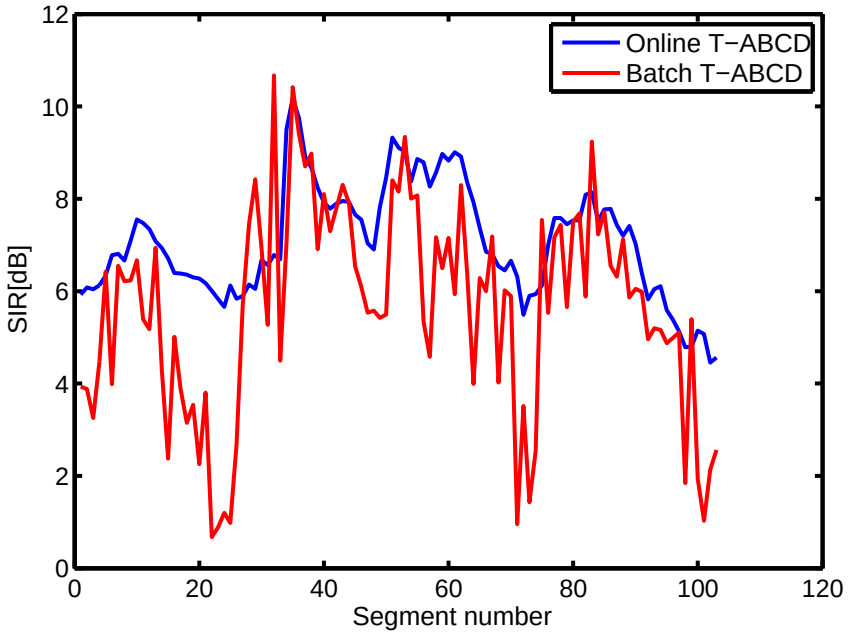


Figure 6.1: Comparison of the online method with its batch counterpart applied block-wise. The results are presented in the terms of the SIR averaged over all sources and microphones

Table 6.1: Comparison of the online T-ABCD with its batch counterpart applied block-wise. The results are presented in the terms of the SIR/SDR, averaged over all segments, sources and microphones

	SIR[dB]	SDR[dB]	SAR[dB]
online T-ABCD	7.07	5.75	13.96
block-wise batch T-ABCD	5.64	4.57	15.05

6.2.2 Selection of the beta parameter within simplified BGSEP

The selection of the parameter β_1 in (6.7) is investigated. The experiment aims at proving that the dynamic choice of β_1 improves the average performance of Step I (ICA) of online T-ABCD. The section is structured as follows. At first, the nature of the experiment is introduced and the objective criterion for the ICA performance evaluation within T-ABCD [2] is proposed. Subsequently, the setup of the experiment is showed and the achieved results are presented.

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

[2] The proposed criterion is general and can be used for any time-domain BASS method based on ICA

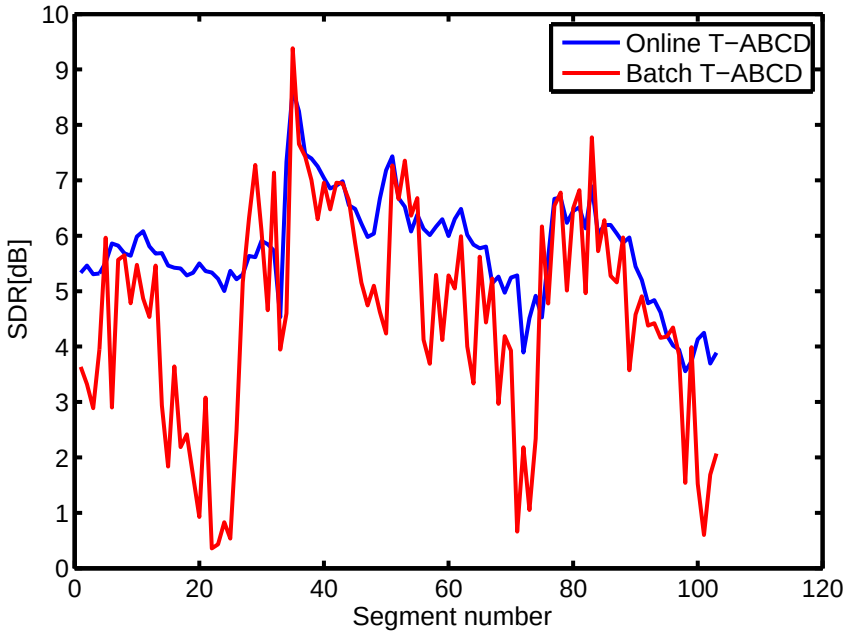


Figure 6.2: Comparison of the online method with its batch counterpart applied block-wise. The results are presented in the terms of the SDR averaged over all sources and microphones

The performance evaluation of the ICA algorithm in time-domain BASS method

Often, the evaluation of the performance of BASS algorithms (i.e. algorithms based on the convolutive mixture model (4.1)) is performed via BSS_EVAL toolbox. However, when evaluating the performance of the ICA algorithm in a BASS method, this approach has serious drawbacks which limit the accuracy of such a measurement. More specifically, the performance of the ICA algorithm is limited by the demixing filter length L and the optimal solution is not unique due to the fact that the estimated signals are retrieved up to an unknown filtering.

To avoid these limitations, the presented experiment deals with the separation of *pseudo-convolutive mixtures* as was proposed by Koldovský in [78]. This approach consists in the construction of the source matrix $\tilde{\mathbf{S}}$, similar as in (4.4), from known sources and the subsequent computation of the mixtures via instantaneous mixing $\tilde{\mathbf{X}} = \tilde{\mathbf{A}}\tilde{\mathbf{S}}$. The mixing process is thus *not* exactly convolutive and can (at least in theory) be inverted perfectly via a demixing matrix $\tilde{\mathbf{W}} = \tilde{\mathbf{A}}^{-1}$.

However, since the corresponding rows of $\tilde{\mathbf{S}}$ are not independent (due to the temporal structure of the sources), ICA applied to $\tilde{\mathbf{X}}$ tend to estimate arbitrarily filtered versions of original sources, not the sources themselves. With respect to this fact, the measure for the evaluation of the ICA performance within the time-domain BASS method should reflect the separateness of the sources and should be independent of random coloration.

As stated earlier, the estimated ICs are affiliated with one of the sources and thus form independent subspaces with minimized inter-source interference. To find these subspaces, the clustering algorithm is used. However, a real clustering method may introduce additional errors into the experiment by the incorrect assignment of ICs to sources. To eliminate this effect, an optimum grouping of the components is defined, based on SIR.

Consider the SIR_j^i of the jth IC with respect to the ith source. Let $\mathbf{C} = \widehat{\mathbf{W}}\tilde{\mathbf{A}}\tilde{\mathbf{S}} \stackrel{def}{=} \mathbf{G}\tilde{\mathbf{S}}$ be a $dL \times N$ matrix of ICs with $\mathbf{G}$ being the so called gain matrix. Then the jth IC can be written as a linear combination of $s_i(n)$ and its delays plus the interference caused by all other sources and their delayed versions. The SIR is then defined as a ratio of energies of these two sums via

$$\mathrm{SIR}_j^i = \frac{\hat{\mathrm{E}}[\sum_{\ell=1}^{L} \mathbf{G}_{j,(i-1)L+\ell} s_i(n-\ell+1)]^2}{\hat{\mathrm{E}}[c_j(n) - \sum_{\ell=1}^{L} \mathbf{G}_{j,(i-1)L+\ell} s_i(n-\ell+1)]^2} \tag{6.18}$$

The optimum clustering using SIR (6.18) affiliates the jth IC with the ith cluster, where it has achieved the largest SIR_j^i.

The SIR defined in (6.18) reflects the separateness of the original sources but is independent of the unknown filtering of the respective ICs. This fact makes it a good criterion for the comparison of the ICA methods' performance (but not for the overall separation quality of $s_i(n)$).

The ideal clustering gives the set of the indices J_i of the components which belong to the ith cluster. The estimated ith source delayed by ℓ samples, i.e. the estimate of the ℓth row of $\tilde{\mathbf{S}}$ denoted by $s_i^\ell(n)$, is obtained as

$$\hat{s}_i^\ell(n) = \sum_{j \in J_i} (\mathbf{G})^{-1}_{(i-1)L+\ell,j} c_j(n), \quad \ell = 0, \ldots L-1 \tag{6.19}$$

The computation through the inverse of $\mathbf{G}$ avoids unknown source permutations given by ambiguities of ICA. The final estimate of the ith source is given by

$$\hat{s}_i(n) = \frac{1}{L} \sum_{\ell=1}^{L} \hat{s}_i^\ell(n+\ell). \tag{6.20}$$

To compute the SIR of the jth estimate with respect to the ith source, the respective components of the jth estimate need to be written as the signal plus interference as suggested in (6.18). The ratio of their energies gives the desired ICA performance criterion.

The setting of the experiment

The experiment was proposed as follows. Four different couples of real-world audio sources of the same length $N_\ell = 100000$ samples were exploited for the construction of the source matrices $\tilde{\mathbf{S}}_\ell, \ell = 1 \ldots 4$. The number of delays $L = 10$. Subsequently, these source matrices were concatenated into matrix $\tilde{\mathbf{S}} = [\tilde{\mathbf{S}}_1, \tilde{\mathbf{S}}_2, \tilde{\mathbf{S}}_3, \tilde{\mathbf{S}}_4]$ of size

$(2L) \times 4N_1$. The sources included two male utterances, one female utterance and two music recordings.

Four random mixing matrices $\tilde{\mathbf{A}}_\ell$ of size $2L \times 2L$ were generated. These matrices were exploited for the construction of a mixture matrix $\tilde{\mathbf{X}} = [\tilde{\mathbf{A}}_1\tilde{\mathbf{S}}_1, \tilde{\mathbf{A}}_2\tilde{\mathbf{S}}_2, \tilde{\mathbf{A}}_3\tilde{\mathbf{S}}_3, \tilde{\mathbf{A}}_4\tilde{\mathbf{S}}_4]$. In this manner, a sudden change of the mixing system is simulated in order to test the ability of the algorithms to adapt the demixing matrix $\tilde{\mathbf{W}}$.

Subsequently, the mixture matrix $\tilde{\mathbf{X}}$ was segmented into overlapping blocks of length $P = 8000$ samples, with a shift $T = 200$ samples. The competing algorithms were applied on these blocks and the SIRs (6.18) were computed for both sources in each block.

The following table summarizes five versions of the BGSEP algorithm which were involved in the experiment. All online versions were in all segments (except for the first one) initialized by $\widehat{\mathbf{W}}$ estimated in the previous segment.

Table 6.2: Algorithms involved in the experiment with pseudo-convolutive mixtures.

Algorithm	Modification
Original BGSEP [32]	None
Online BGSEP	No U-WEDGE, single WEDGE, $\beta_1 = 1$
Constrained BGSEP 0.3	No U-WEDGE, single WEDGE, $\beta_1 = 0.3$
Constrained BGSEP 0.6	No U-WEDGE, single WEDGE, $\beta_1 = 0.6$
Dynamic BGSEP	No U-WEDGE, single WEDGE, β_1 via (6.16),(6.17)

Table 6.3 summarizes the average SIR of the competing algorithms along with its standard deviation given in brackets. Dynamic BGSEP outperformed all other algorithms in the case of the first signal (the improvement of 1dB to the second algorithm), but achieved worse results in the case of the second signal (about 0.4 dB worse than the best online algorithm).

It should be stressed that the Dynamic BGSEP achieved the lowest SIR variance of all the online algorithms. This points out to the fact that the Dynamic BGSEP provides the most stable performance.

It should also be noted that the experiment slightly favors the Original BGSEP, which does not need to adapt to the sudden changes of the mixing matrix. It converges directly into a new solution, thus achieving better results in the terms of average SIR and a lower standard deviation.

An example of the SIR course with respect to the time position is shown in Figure 6.3. The figure presents the separation of the first source. The results are shown in the time interval when the mixing matrix $\tilde{\mathbf{A}}$ is suddenly changed.

It can be seen that the Dynamic BGSEP adapts quickly to the change of the demixing system, e.g. more quickly then the Constrained BGSEP 0.3 does. On the other hand, in the situations, when the demixing system does not change, the Dynamic BGSEP tends to change $\widehat{\mathbf{W}}$ slowly, thus keeping stable results. Again, in

Table 6.3: The SIRs achieved by separation of pseudo-convolutive mixtures averaged over all data blocks. The standard deviations are given in brackets.

Algorithm	SIR_1[dB]	SIR_2[dB]
Original BGSEP	45.71 (8.76)	46.50 (10.45)
Online BGSEP	45.35 (9.82)	46.37 (10.76)
Constrained BGSEP 0.3	45.36 (10.94)	46.15 (11.17)
Constrained BGSEP 0.6	45.42 (9.90)	46.40 (10.86)
Dynamic BGSEP	46.59 (9.56)	46.01 (8.90)

comparison with the Constrained BGSEP 0.3, the Dynamic BGSEP achieves better separation results on average.

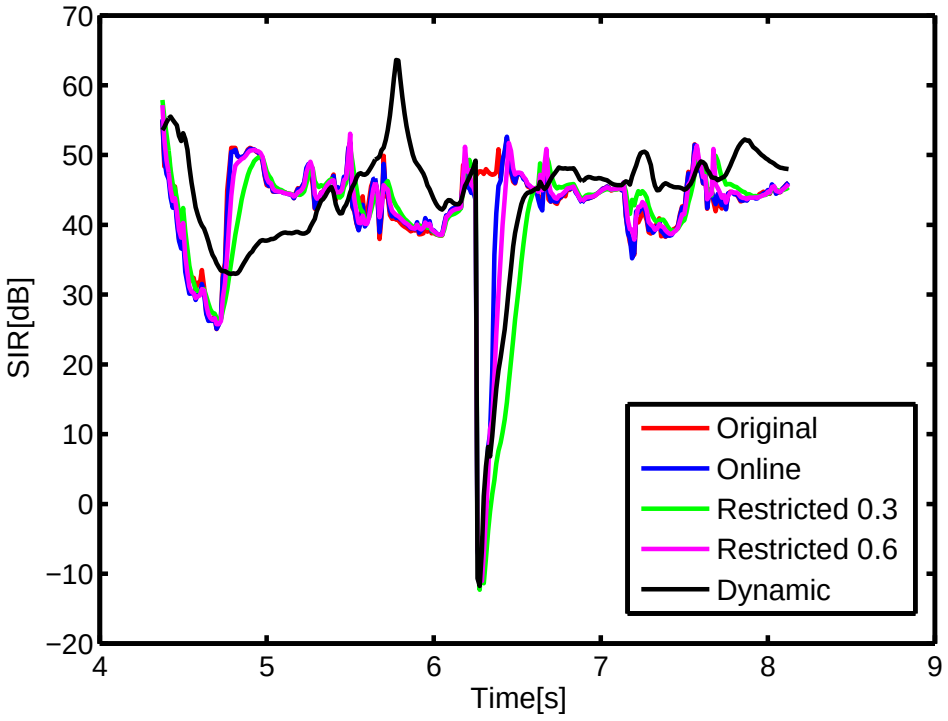


Figure 6.3: The course of the SIR during the separation of the first source within the time interval of the first change of the demixing matrix $\tilde{\mathbf{A}}$ (6.25s).

6.2.3 Separation of sources with fixed positions

In this experiment, the online algorithm separates real-world stationary mixtures of speech signals. The positions of the sources and the microphones are fixed. The results of online T-ABCD are compared to the performance of its batch counterpart.

The experiment aims at showing that the online T-ABCD is able to adapt to the non-stationarity of the speech signals. This improves its average performance

even when the mixing system is stationary. The experiment uses publicly available data from Hiroshi Sawada's websites.[3]

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. Omni-directional microphones are used to record the mixtures.

The online and batch T-ABCD were both applied with $L = 30$. The other parameters of the online method were set to $P = 6144$, $T = 512$, $\beta_2 = 0.95$ and $\alpha = 3$. The separation results are evaluated block-by-block. The blocks are of the same size as in the online method .

Unlike Section 6.2.1, the batch method is *not* applied block-wise. It estimates the demixing transform using a single block of the data only. This should be sufficient, since the mixing process is stationary. Table 6.4 summarizes the results averaged over all blocks, microphones and sources.

Table 6.4: Results for separation of sources at fixed positions.

	SIR[dB]	SDR[dB]	SAR[dB]
online T-ABCD	8.43	1.58	4.41
batch T-ABCD	5.01	0.08	5.79

Online T-ABCD achieves better results in terms of SIR and SDR than the batch algorithm. This verifies the fact that the online method is able to adapt the separating filters throughout the recordings respecting the nonstationarity of sources (*not* the changes in the mixing process which is stationary in this experiment). On the other hand, the time-invariant separation done by batch T-ABCD produces less artifacts as indicated by SAR.

6.2.4 Separation of moving sources

This experiment shows the ability of online T-ABCD to separate the mixtures of moving sources. To this end, we consider data given in the task "Determined convolutive mixtures under dynamic conditions" (Audio Signal Separation) in the SiSEC 2010 [4] evaluation campaign. The campaign was organized as part of the LVA/ICA 2010 [5] conference .

The data simulate dynamic conditions so that *the maximum of two of six speakers* located at fixed positions *around a stereo microphone array* are active at a time. The separating algorithm applied to the data thus needs to adapt to active speakers. The distances between microphones were 2 or 6 cm and the sampling rate was 16 kHz.

We compare the proposed online T-ABCD with the frequency-domain BSS method by Francesco Nesta et al [61, 79]. The online method was applied with

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

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

[5] `http://lva2010.inria.fr/`

$L = 30$, $P = 6144$, $T = 512$, $\beta_2 = 0.95$ and $\alpha = 4$. The Nesta's method uses FFT of the length 4096 samples with 75% overlap. As the method works offline, it was applied independently on disjoint blocks of 1 second where the maximum of two sources were active.

Table 6.5: Results for the 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	9.25	1.94	4.43
Nesta	7.90	1.85	5.37

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. It can be concluded that online T-ABCD seems to outperform the frequency-domain algorithm in cases of larger microphone distances, where the spatial aliasing occurs.[6]

Further experiments with non-stationary mixtures can be found in the SISEC2010 evaluation campaign, which results are available online. More specifically, it concerns Task 2 in scenario "Determined convolutive mixtures under dynamic conditions", which the author participated in. The results presented there verify the ability of online T-ABCD to separate the non-stationary mixtures. However, in the competition with the presented frequency domain approaches, the online T-ABCD was outperformed in some cases.

6.2.5 Computational demands

The presented experiments were performed on a computer with single core 2.6 Ghz processor with 2 GB RAM. Online T-ABCD was implemented in Matlab environment. The computational demands of the algorithm depend on the demixing filter length L and the sampling frequency of the mixtures. The mixture signals in Section 6.2.4 were 3 minutes, 29 seconds long, sampled at 16kHz. Online T-ABCD separation lasted 14 minutes, 36 seconds ($L = 30$). This separation task can be performed in real time, when the desired filter length $L = 10$. Mixtures of two sources sampled at lower sampling frequency 8 kHz can be separated in real-time when $L = 18$.

[6] The values presented in Table 6.5 slightly differ (for the case 6cm) from values published in [76]. This is due to correction of an error which was encountered during computation of the evaluating criteria in [76]. However, the recalculation of the values does not change the results (the meaning) of the experiment.

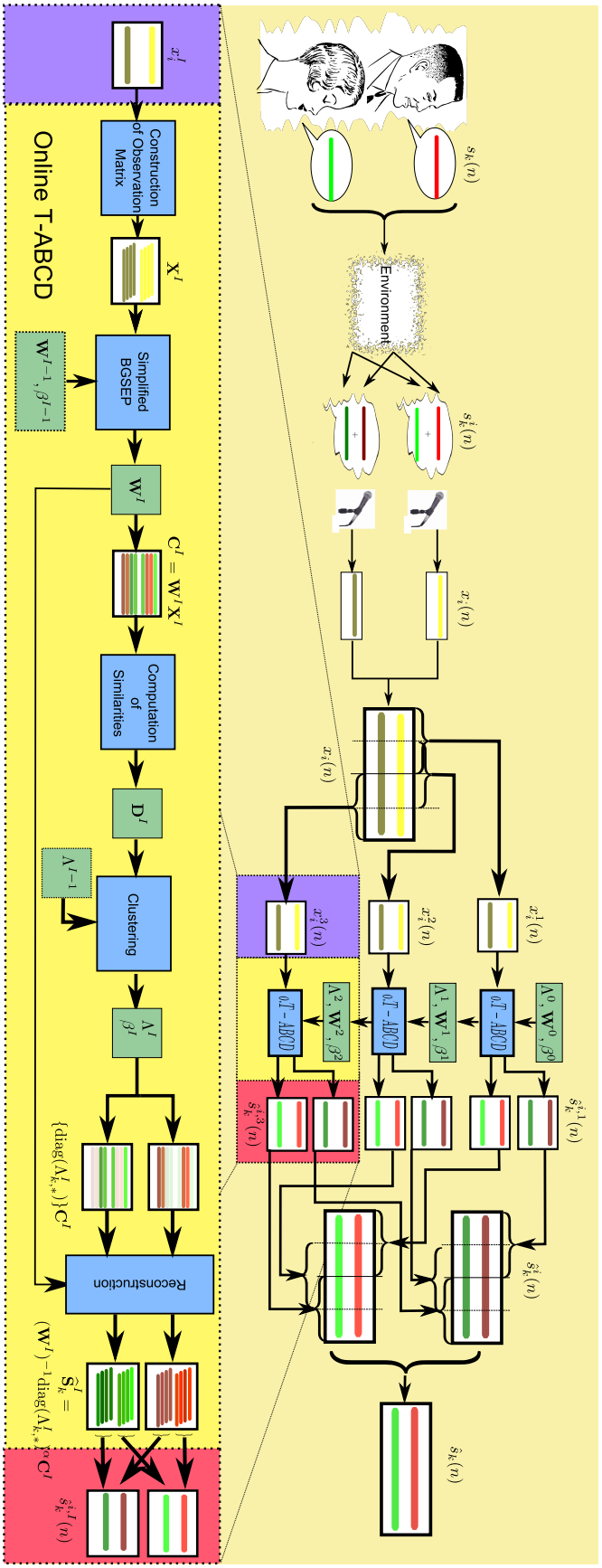


Figure 6.4: Flowchart of the online T-ABCD algorithm

Chapter 7

Conclusions and Future Work

This chapter summarizes the research presented in the thesis and points out to future work ideas.

7.1 Separation of non-stationary non-gaussian sources

The author contributed to the development of the Block EFICA algorithm, which is designed for the separation of non-stationary and non-gaussian sources.

The Block EFICA algorithm is based on the famous algorithm FastICA [15] and its efficient variant EFICA [24]. It consists of three consecutive steps. In the first step, the sources are pre-estimated via FastICA. Within the second step, the score functions of the sources are estimated. Subsequently, the obtained functions are used as non-linearities in the one-unit Block EFICA iterations, which refine the estimates of the sources. The algorithm is finished by the computation of constants which minimize the residual mean square error of the rows of the demixing matrix. The method is proven to be asymptotically efficient provided that the sources are block-wise stationary with a constant variance.

Extensive experiments show the ability of the proposed method to accurately estimate all types of sources, including real world signals which do not obey the model of the algorithm. Moreover, it is shown that the theoretical assumption of the constant variance of the sources does not deteriorate the method's practical performance, when signals with varying energy are separated.

7.2 Modifications of the T-ABCD algorithm

The blind separation of audio sources can be performed by methods based on ICA in the time-domain. An example of such a method is T-ABCD [9] proposed by Koldovský and Tichavský. The algorithm exhibits an advantageous modular structure which allows simple implementation of new features.

The separation performed by ICA within T-ABCD is usually not perfect and a single IC often contains interference from other sources. Therefore, in the clustering step, the component cannot be uniquely assigned to a single source, but should be affiliated with multiple sources to a certain degree. To reflect this ambiguity, the author proposed the utilization of the Relational Fuzzy C-Means algorithm [71] for clustering of the ICs.

The experiments suggest that RFCM slightly improves the separation performance of T-ABCD compared to the original hard clustering method and is less

computationally demanding. Moreover, RFCM features a favorable iterative structure, which enables it for exploitation within the online version of T-ABCD.

7.3 Adaptive separation of audio signals

The original batch T-ABCD assumes that that the mixing process does not change its inner parameters in time. In practice though, the mixing may be non-stationary, e.g. due to the movement of the sources.

The author proposed an online variant of the T-ABCD method which profits from the fact that the original T-ABCD is able to estimate the demixing filters using short data segments only. The method is based on the block-wise application of the original T-ABCD, which is modified to respect the continuous changes in the parameters of the mixing system. The adaptation is performed via a single iteration of ICA/clustering algorithms in each of the blocks. The speed of adaptivity is controlled by learning parameters.

The experiments suggest that the algorithm is able to adapt to the changing position of the sources. Moreover, it is shown that the algorithm can adapt to the non-stationarity of speech signals as well. Regarding the computational burden; the separation can be performed in real time provided that the desired separation filter is reasonably long.

7.4 Future work

The intended future research is focused on the online or batch T-ABCD algorithms, since the remaining projects are concluded.

Online T-ABCD was subject to numerous tests. However, detailed testing regarding the influence of some of its inner parameters on the separation performance has not yet been performed. This concerns especially the length and the shift of the ICA segment along with the forgetting factor in (6.11). These parameter values were selected in an ad-hoc manner. Moreover, cases of instability, which deteriorate the performance of the algorithm, occasionally occur, especially when a shorter length of the ICA segment is used. These cases need to be revealed through testing and corrected. The intended testing will include the objective criteria such as SIR/SDR/SAR as well as listening tests.

Further development of the algorithm is directed towards simplification and acceleration. The most computationally demanding part of the method is the computation of similarities among ICs. In the current version, the similarities are based on the GCC-PHAT [66] coefficients. These can be replaced, for example, by spectral coherence. It is a standard tool in the analysis of time series, which measures the dependence between two time series. The computation of coherence is simpler than the computation of GCC-PHAT. The computational burden could be further lowered by the computation of the similarities directly from the ICA demixing matrix $\widehat{\mathbf{W}}$, if it is possible and usable.

The algorithm can be simplified by focusing on specific settings and scenarios. Recently, a paper [80] concerning the implementation of online T-ABCD in C++ has been submitted. The paper focuses on the case of two sources recorded by two microphones. Here, some algebraic simplifications can be exploited. For example, the explicit formula for matrix inversion can be used. Furthermore, the affiliations of ICs to clusters need to be computed for the first cluster only, the second is just complement to one.

For the general case when multiple microphones are used, the algebraic simplifications can be found as well. For example, the matrix inversion could be in certain cases replaced by its Taylor expansion approximation. However, this approach needs to be further examined: the application of simplifications may lead to the deterioration of the performance and stability of the algorithm.

Online T-ABCD is influenced by the ambiguity of the ordering within its ICA step. In most cases, its consequence is suppressed, due to a short shift of the ICA window and the continuous changes of the demixing matrix. However, if there appears an interval in the data where the separation via ICA cannot be performed and thus the continuity is disturbed, the estimated sources may switch channels. This problem arises for example, when the source positions are located behind each other and subsequently leave the cover. This situation could be probably solved by the speaker diarization process. It detects various speakers in the data and groups together the utterances spoken by the same speaker.

Appendix

Automatic Classifiers for Medical Data from Doppler Unit

The appendix discusses research which does not concern blind audio source separation. This research was conducted by the author of the thesis during the first year of his Ph.D. studies and was published in papers [81, 82]. It deals with the automatic classification of medical data: signals originated in ultrasound measurement of blood flow in the arteries of the lower limbs are analyzed. The measurement is part of the screening examination for atherosclerosis.

The presented work is based on the results published in the author's master thesis. Unlike the original results introduced there, the author proposes a new set of features which includes attributes used for the diagnostics by human experts. The practical computation of the features is refined. The classifiers are re-trained and their extensive testing is performed. The re-design of the classifiers leads to an improvement of the recognition score by 5%. Moreover, the evaluation of the classifiers in the means of the sensitivity and specificity is performed. These measures are often used for the evaluation of medical tests.

8.1 Introduction and motivation

Atherosclerosis and diseases of the cardiovascular system pose a serious threat for the modern population. Typical risk factors are smoking, diabetes mellitus, hypertension and the lack of movement. The manifestation of these diseases in human extremities is called the Peripheral Arterial Disease [83]. The illness has four progress stages. In the early stage, the patient feels no subjective troubles. The last stage is distinguished by tissue necrotization and gangrene, which is very dangerous for patient's extremities and possibly life threatening. The lower limb arteries are afflicted more often than those in the upper limbs. The well-timed diagnostic of the disease is important because it can considerably simplify the treatment.

Nowadays, there are many methods how to detect obstructions within arteries. The simplest method is the physical examination of the limb: a medical doctor inspects the color and the temperature of the extremity. The disadvantage of this approach is that pathological changes are often detectable only after the disease enters its final stages. The CT angiography is a very accurate state-of-the-art examination method, but there is some danger connected with its invasiveness. The best noninvasive method appears to be the ultrasonic duplex scan, which is able to visualize the profile of the artery along with the dynamic representation of blood

flow within. Both above mentioned methods require expensive equipment, so these examinations are used in clinical medicine, not in general medicine practice.

For fast noninvasive screening of PAD in diabetological and cardiological ambulances, ultrasonic Doppler devices have been used for a long time. These inexpensive units measure average blood flow velocity along with distant blood pressures on several typical places of the lower limb. The resulting signal is usually denoted as a *Doppler velocity waveform*. From the shapes of the waveforms (or the sound emitted by the device into headphones) the expert can detect PAD. Doppler units are notably cheaper than duplex scanners and are affordable in general medicine practice.

The following sections present a set of automatic classifiers which analyze the measured waveforms and affiliate them with classes which reflect the progress of PAD. These classifiers, along with the cheap Doppler probe, could help to identify the first phases of the PAD in general practice surgeries. The automated examination performed by general practitioners could help to further improve the well-timed diagnostics of PAD, because the traditional analysis is partly subjective and depends to some extent on the experience of the expert.

8.2 Automatic Classifier Design

The proposed classifiers are designed according to the principle of supervised learning. In order to train them, a large database of real-world Doppler waveforms had to be collected. In our case, the data come from the Regional Hospital in Liberec, where it has been acquired during the last few years. Prior to their use for research purposes, the data were made anonymous, all personal information of the patients was deleted.

8.2.1 Measurement of Doppler waveforms

The waveforms were measured by the hand-held ultrasonic unit Multi Dopplex II and sent to a PC for storing via the RS232 interface. They represent the mean velocity of blood flow in the artery in a short time period. By default, the signals are measured along with blood pressures. Five standard locations on each leg are examined, i.e. there are 10 waveforms from one patient per one examination. The standard positions examined are following: 1) artery femoralis, 2) a. poplitea, 3) a. tibialis posterior, 4) a. tibialis anterior and 5) a. dorsalis pedis. These positions are depicted in Figure 8.1.

Multi Dopplex II is a bi-directional device; the wave-forms could be displayed as forward and backward flow or as a difference of these two signals in a combined waveform. Examples of Doppler waveforms can be seen in Figure 8.2.

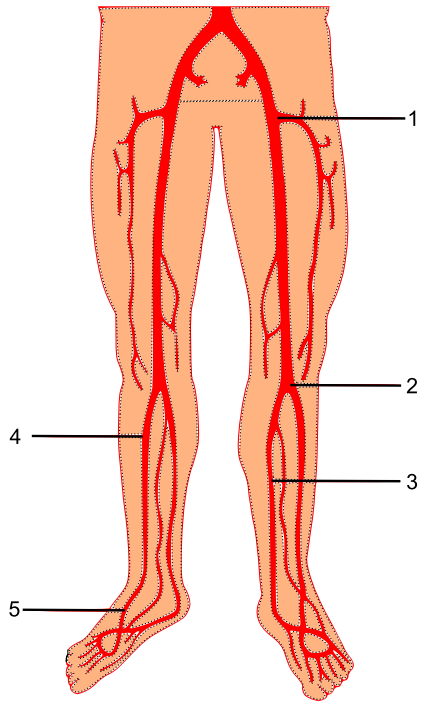


Figure 8.1: Standard positions for examinations with the Doppler unit (picture taken from utility software distributed along with the Doppler Unit - Dopplex Reporter)

8.2.2 Classes

In accord with literature [84] and with the expert's opinion, four classes are chosen for the classification. Three of the classes reflect various degrees of artery occlusion. The remaining class contains waveforms which are deteriorated by some error encountered during measurement.

The classes are proposed as follows:

- **Normal course** – Waveforms acquired by the examination of arteries without PAD, e.g. Figure 8.2 (a).
- **Stenotic course** – Signals measured in arteries with a stenotic diameter, e.g. Figure 8.2 (b).
- **Occlusion**– Signals measured in arteries with a total arterial obstruction, e.g. Figure 8.2 (c).
- **Incorrect course** – This class covers four kinds of measurement errors which are commonly encountered during examination, i.e.

  1. The setting of the amplification factor is too high. The resulting waveform is clipped.
  2. The signal is under strong influence of near veins, the dicrotic notch usually present in the normal triphasic waveform is lost in the noise.

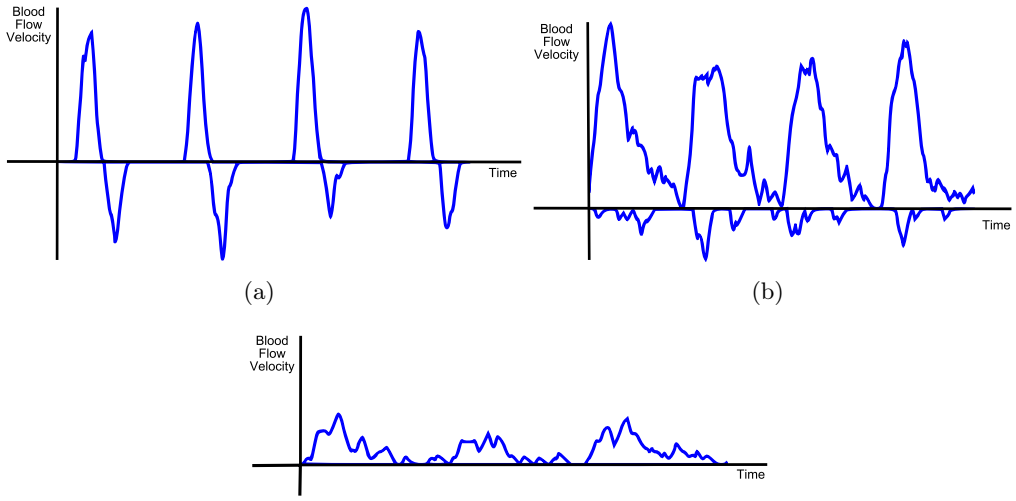


Figure 8.2: Representative waveforms of classes chosen for the automatic classification. Directional waveforms acquired in (a) artery with no PAD, (b) artery with stenotic diameter, (c) artery with total arterial occlusion.

3. Measured forward and backward velocities are echoes of each other, after the calculation of the difference the combined signal is almost zero.
4. The signal was not measured at all. This could happen by the wrong placing of the probe, but it could also mean the total obstruction of the artery.

The expert of angiology classified part of the available database manually into designed classes before training of the classifier. This prior knowledge is used in the training process and also in the testing phase, when expert's opinion is compared with the results of the classifier.

8.2.3 Features

During the design process, 18 features were considered as potentially useful for the classifier. These features describe the quality of measured signals in time domain, frequency domain or have a special medical meaning.

Obtaining the exact values of velocities can be difficult with a simple Doppler unit because the application angle of the probe strongly influences the amplitude of the measured data. The standard angle ranges from 45° to 60°. This fact is proved true in experiment 8.3.1, where the most informative features are selected. The features depending on exact values of velocities were not identified as useful for the classifier. The most useful features found were those which use ratios of velocities. These features reflect rather the shape of the waveform than their amplitudes.

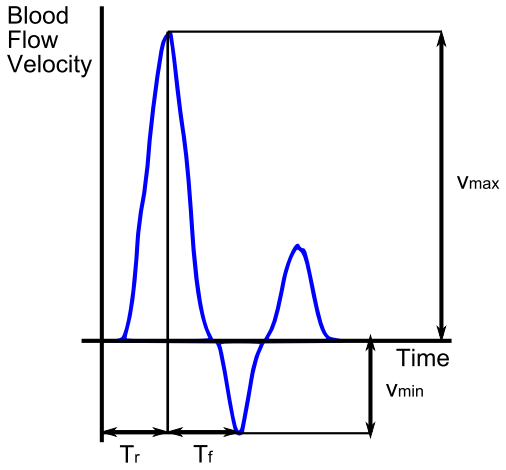


Figure 8.3: Proposed features of the Doppler waveforms

Based on the literature [84, 85, 86] and the expert's information, the following list of potential features was compiled.

- **Brachial pressure index** (BPI) – The ratio of the patient's system blood pressure (measured on a. brachialis) and distal pressure in the examined position on the lower limb.
- **Pulsation index** (PI)
$$\mathrm{PI} = \frac{v_{max} - v_{min}}{v_{avg}}, \tag{8.1}$$
where v_{avg} is the average velocity during one pulse duration.
- **Resistance (Pourcelot) index** (RI)
$$\mathrm{RI} = \frac{v_{max} - v_{min}}{v_{max}}, \tag{8.2}$$
- **Maximum velocity** v_{max} - Maximum velocity during pulse.
- **Minimum velocity** v_{min} - Minimum velocity during pulse.
- **Acceleration** (A)
$$\mathrm{A} = \frac{v_{max}}{T_r}, \tag{8.3}$$
- **Deceleration** (D)
$$\mathrm{D} = \frac{v_{max} - v_{min}}{T_f}, \tag{8.4}$$
- **Velocity-time index** (VTI)
$$\mathrm{VTI} = \frac{v_{max} - v_{min}}{T_r + T_f}, \tag{8.5}$$

- **Artery Resistance Parameter** (RP)

$$\mathrm{RP} = \frac{v_{max}}{v_{min}}, \tag{8.6}$$

- **Log Energy**

$$\mathrm{Energy} = \ln(\sum_{\ell=1}^{N} s(\ell)^2), \;\; \ell = 1 \ldots N, \tag{8.7}$$

  where $s(\ell)$ is a sample of the Doppler waveform; N samples are available.

- **A set of 8 frequency features** – The standard duration of measuring at one position is 5 seconds using a 100 Hz sampling frequency. The spectrum is calculated from the entire discrete signal via Fast Fourier Transform (FFT). The most energy in the spectra is concentrated up to one quarter of the sampling frequency. These spectral coefficients are multiplied by eight triangle windows with a half overlap in order to get 8 frequency features F1 to F8.

Before computing the features, the signal is preprocessed by low pass filtering. This suppresses high frequency noise, but keeps the shape of the waveform.

The Sequential Forward Search (SFS) algorithm [87] is used to determine the most significant features. Its advantage consists in the fact that it utilizes the target classifier. The iterative SFS algorithm works in the following manner.

In the first step, it identifies (via recognition with the target classifier) the feature with the best score. In the nth step, the set of previously selected $n-1$ features is extended by adding that feature from the remaining ones which makes the best classification with the n-feature set. The algorithm is terminated if the score in the current step is lower than in the previous one or if the number of steps (and already selected features) reaches a predefined limit. In this way we get the set of the M most informative features for the target classifier.

8.2.4 Detection of waveform periods

All above mentioned features are calculated automatically in real time, without human intervention. Most of them require that a single pulse is extracted and its shape and size is analyzed.

The detection in the time domain is quite complicated. With growing stenosis in the artery, the waveforms lose their shape and become non-pulsative. Also the presence of the vein signal (mostly in signals from a. femoralis, a. poplitea) complicates this task.

The autocorrelation function is used for the detection of pulses in the signal. Maxima in the waveform are traced to detect the beginning and end of one pulse. The sample difference is used for identifying the extremes.

8.2.5 Classifier types

Two basic types of classifiers were implemented during the design process: *The minimal distance classifier* and the *Bayesian classifier*. The classifiers analyze waveforms represented by a column feature vector of length M.

The minimal distance classifier (MDC) expresses each class by its best representative called prototype, which is a sample with a minimum distance to all others within given class. The implemented MDC classifier uses the Mahalonobis metrics, where the distance between two feature vectors $\mathbf{x}, \mathbf{y}$ is given by

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})' \Sigma^{-1} (\mathbf{x} - \mathbf{y})}, \tag{8.8}$$

where Σ is the sample covariance matrix of features within each class.

The Bayesian classifier (BC) represents each class $C_i,\ i = 1 \ldots 4$ by a Gaussian probability density function (pdf) in the M-dimensional feature space. Its two parameters are the means and the variances. Since the classes have different occurrence rates, the class prior probabilities are also taken into account. The pdf is defined as follows:

$$P(\mathbf{x}|C_i) = \frac{1}{\sqrt{(2\pi)^M \cdot \det(\Sigma)}} \cdot \exp[-0.5(\mathbf{x} - \bar{\mathbf{x}})' \Sigma^{-1} (\mathbf{x} - \bar{\mathbf{x}})] \tag{8.9}$$

8.2.6 Training and testing of the classifiers

For better modeling of the distributions of the feature vectors in the M-dimensional space, it is useful to split data in each class into clusters and represent each of them by a separate prototype or a separate pdf. In our case, the clusters are identified via the well known K-Means algorithm in combination with the Linde-Buzo-Gray algorithm (LBG) [88].

During the training phase, each diagnostic class is represented by one or more clusters, where each one is described by its parameters, i.e. mean vectors, covariance matrices and occurrence counts.

In the testing phase, the minimal distance classifier assigns the measured data represented by a feature vector to the nearest prototype and decides to which class the unknown data belong. The Bayesian classifier assigns the class whose posterior probability is the highest one. The errors of incorrect course are detected before the classification stage. If the signal is identified as incorrectly measured, the classification is denied.

In order to train the classifiers and to make extensive tests, a large database of real-world Doppler waveforms was prepared by an expert. He manually classified data from 900 examinations. These were measured at 10 standard positions (5 on each leg), i.e. there were 9,000 sample waveforms available. Approximately 15% of all these signals were found incorrect. The reason lay mostly in setting amplification too high on a. femoralis so that the signal was clipped. From the correctly measured ones, 47% were assigned to the class Normal Course, 32% to the class Stenosis and the rest 6% into the class Occlusion. In the experiments, data from 720 randomly

chosen examinations were used for training the classifier; the remaining data (of 180 subjects) were left for testing. In each individual test, the result of the classifier was compared to the expert's decision. This was done for all test data and then the recognition score was calculated as a ratio of the correctly assigned samples to all available testing ones. The scores were calculated for each measured position and later averaged over all positions. To make the results more significant, the random database splitting into the training and testing part was repeated 5 times and the final scores were calculated as the means from the 5 tests. In other words, all the scores mentioned in the following section are averaged results from 9000 individual classifications (180 subjects x 10 positions x 5 repetitions).

8.3 Experiments

8.3.1 Selection of features by the SFS algorithm

The results obtained by the application of the SFS algorithm on the set of 18 proposed features are illustrated in Figure 8.4 and Table 8.1. It can be observed that the best classification is acquired with 6 features, while adding more yields lower, and then even higher degradation of performance. The SFS algorithm identified the following best 6 features: BPI, deceleration, resistance (Pourcelot) index , velocity-time index and second and third frequency feature. As most of the energy of waveform spectra is centered in low frequencies, the higher frequency features did not bring any additional improvement.

If we compare these 6 most informative features with those used by human experts in vascular labs, we can see that the PI feature often used by experts was not selected by the SFS algorithm. This may be caused by the fact that the average velocity in (8.1) can be influenced by less accurately detected borders of the pulse when compared with a manual measurement.

Table 8.1: Detailed results from the SFS algorithm's first 9 steps. The scores and added features are shown.

M	Score [%]	Added Feature
1	81,75	VTI
2	87,5	BPI
3	88,78	F2
4	89,08	RI
5	89,18	D
6	89,19	F3
7	88,88	A
8	88,64	1st freq.
9	88,39	Energy

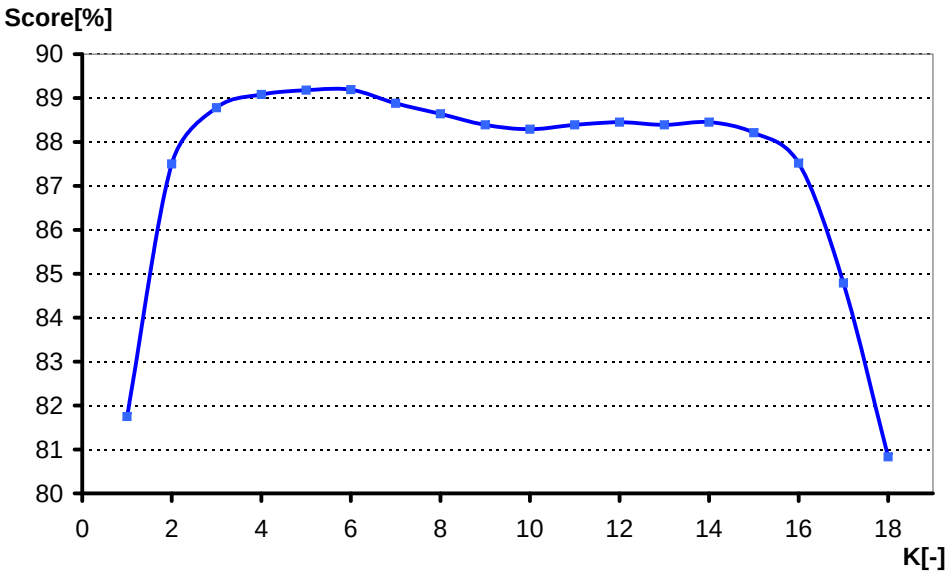


Figure 8.4: Recognition score as a function of feature vector length M. Results obtained by the SFS algorithm.

8.3.2 Testing of the classifiers

In Tables 8.2 and 8.3, the comparison of the results from the two classifiers and their various settings are shown. The scores are based on correct decisions that include a) the classification into a correct diagnostic class and b) a correctly detected measurement error. It is evident that the best results were achieved by the Bayesian classifier with multi-modal pdfs and prior probabilities. The best score was 89%, i.e. the classifier and the expert agreed in 89% of the cases.

In medicine, the results are often indicated as sensitivity and specificity rates. *Sensitivity* is defined as

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{8.10}$$

and *specificity* is defined as:

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \tag{8.11}$$

Because the rates are applicable for binary classifiers only, we had to identify as positive such waveforms that contained pathological attributes (classes "Stenosis" and "Occlusion"). The detection of measurement errors is not implicated in these values. Human expert's opinion acts here as the golden standard.

The strictest view on performance evaluation is given by the recognition rate of the classifier that has to decide between 4 classes. In our experiments, the best

Table 8.2: Recognition scores for different classifier types and settings

Classifier	Setting	Score[%]
MDC	Mahalonobis - 1 cluster	81.98
MDC	Mahalonobis - Multiple clusters	83.65
BC	1-modal pdf without prior prob.	80.17
BC	1-modal pdf with prior prob.	84.46
BC	Multi-modal pdf without prior prob.	86.96
BC	Multi-modal pdf with prior prob.	89.19

Table 8.3: Sensitivity and specificity for different classifier types

Classifier	Setting	Sensitivity[%]	Specificity[%]
MDC	Mahal. - 1 cluster	87.83	81.93
MDC	Mahal. - Multiple clusters	90.95	85.77
BC	1-modal pdf without p.p.	93.05	75.38
BC	1-modal pdf with p.p	90.23	78.14
BC	Multi-modal pdf without p.p.	90.15	87.90
BC	Multi-modal pdf with p.p.	87.73	90.54

results were achieved by using the multi-modal BC with prior probability. The 89% agreement can be considered fairly high if we realize that the boundaries between the classes can be questionable in some cases, even for a human expert.

The evaluation of the classifier by means of the sensitivity and specificity rates forces the classifier to accept only two classes. Here, the specificity of the MDC and the one-modal BC is lower since these classifier settings often tend to erroneously assign border cases of the class "Normal" (Negative) to the class "Stenosis" (Positive).

8.3.3 Discussion

During the testing of the classifiers, it was found that one of the most critical issues is the detection of pulses, their description and measurement. Skilled experts can do it easily, but for a fully automated system, it may pose a problem.

Further improvement may be achieved by the suppression of vein signals. The author believes that this could be partly accomplished by a properly designed high pass filter. The vein signal has a rate of 15 to 20 pulses per minute and appears as a slowly changing trend in the data.

The extension of the feature set could be also useful, especially by adding features that could be calculated even when the waveform is non-pulsative or distorted in some way. Recently, the frequency attributes fulfill this role. It also seems useful to extend the number of classes for a more precise classification of signals. The class Stenosis could be split into two subclasses: the mild and the severe Stenosis.

8.4 Conclusions

Two types of automatic classifiers were designed for the Doppler waveforms. A large database of anonymous waveforms was compiled and subsequently used for training and testing of the classifiers. Based on the literature and the expert's opinion, the author introduced four diagnostic classes which reflect the progress of arterial occlusion. Eighteen features were proposed to describe the waveforms; the experiments revealed six most informative ones. The testing of the classifiers led to 89% agreement between the automatic classifier and the human expert.

Bibliography

List of cited papers

[1] P. Comon, Ch. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, ISBN: 978-0123747266, 2010.

[2] A. Hyvärinen, J. Karhunen, E. Oja, *Independent component analysis*, Wiley-Interscience, New York, 2001.

[3] I. T. Jolliffe, *Principal Component Analysis*, Springer, 2002.

[4] P. Georgiev, F. Theis, A. Cichocki, Sparse component analysis and blind source separation of underdetermined mixtures, *IEEE Transactions on Neural Networks*, Vol. 16, No. 4, pp. 992-996, 2005.

[5] A. Cichocki, R. Zdunek, A. H. Phan, S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation* , Wiley-Interscience, ISBN: 978-0-470-74666-0, 2009.

[6] O. Grellier, P. Comon, Blind separation of discrete sources, *IEEE Signal Processing Letters*, No. 5, pp. 212–214, 1998.

[7] P. Jallon, A. Chevreuil, P. Loubaton, and P. Chevalier, Separation of convolutive mixtures of cyclostationary sources: a contrast function based approach, *Proc. ICA2004*, Granada, Spain, September 2004.

[8] D.-T. Pham, F. Vrins, Discriminacy of the minimum range approach to blind separation of bounded sources, *Proc. ESANN 2006*, Bruges, Belgium, April 2006.

[9] Z. Koldovský and P. Tichavský, Time-Domain Blind Audio Source Separation Using Advanced ICA Methods, *Proc. Interspeech 2007*, pp. 846-849, August 2007.

[10] J. F. Cardoso, Blind signal separation : Statistical principles, *Proceedings of the IEEE*, Vol. 86, No. 10, pp. 2009-2024, 1998.

[11] P. Comon, Independent component analysis - a new concept?, *Signal Processing*, Vol. 36, No. 3, pp. 287-314, 1994.

[12] R. Boscolo, H. Pan, and V. P. Roychowdhury, Independent Component Analysis Based on Nonparametric Density Estimation, *IEEE Trans. on Neural Networks*, Vol. 15, No. 1, pp. 55-65, 2004.

[13] W. Silverman, *Density Estimation for Statistics and Data Analysis*, New York: Chapman and Hall, 1985.

[14] A. J. Bell, T.J.Sejnowski, An information-maximization approach to blind separation and blind deconvolution, *Neural Computation*, Vol. 7, pp. 1129-1159, 1995.

[15] A. Hyvärinen, E. Oja, A fast fixed-point algorithm for independent component analysis, *Neural Computation*, Vol. 9, No. 7, pp. 1483-1492, 1997.

[16] J. F. Cardoso, Source separation using higher order moments, *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP'89)* , pp. 2109-2112, Glasgow, UK, 1989.

[17] J. F. Cardoso and A. Soulomiac, Blind Beamforming for non Gaussian signals, *IEEE Procedings-F*, Vol. 140, No. 6, pp. 362-370, 1993.

[18] K. Abed-Meraim, Y. Xiang, and Y. Hua, Generalized second order identifiability condition and relevant testing technique, *Proc. ICASSP'00*, pp. 2989–2992, 2000.

[19] L. Tong, R.-W. Liu, V.C.Soon, Y.-F. Huang, Indeterminacy and Identifiability of Blind Identification, *IEEE Transactions on Circuits and Systems*, Vol. 38, pp. 499-509, 1991.

[20] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E.Moulines, "A blind source separation technique using secondorder statistics", *IEEE Transactions on Signal Processing*, Vol. 45, pp. 434-444, 1997.

[21] D.-T. Pham, J. F. Cardoso, Blind separation of instantaneous mixtures of non stationary sources, *IEEE Transactions on Signal Processing*, Vol. 49, No. 9, pp. 1837-1848, 2001.

[22] I.F. Gorodnitsky and A. Belouchrani, Joint Cumulant and Correlation Based Signal Separation with Application to EEG Data, *Proc. ICA 2001*, San Diego, 2001.

[23] P. Tichavský, Z. Koldovský, A. Yeredor, G. G. Herrero, and E. Doron, A Hybrid Technique for Blind Separation of Non-Gaussian and Time-Correlated Sources Using a Multicomponent Approach, *IEEE Trans. on Neural Networks*, Vol. 19, No. 3, pp. 421-430, ISSN: 1045-9227, March 2008.

[24] Z. Koldovský, P.Tichavský, E. Oja, Efficient Variant of Algorithm FastICA for Independent Component Analysis Attaining the Cramér-Rao Lower Bound, *IEEE Transactions on Neural Networks*, Vol. 17, No. 5, pp. 1265-1277, 2006.

[25] A. Yeredor, Blind separation of gaussian sources via second-order statistics with asymptotically optimal weighting, *IEEE Signal Processing Letters*, No. 7, pp. 197–200, 2000.

[26] A. Hyvärinen, A unifying model for blind separation of independent sources, *Signal Processing*, Vol. 85, No. 7, pp.1419-1427, 2005.

[27] P. Tichavský, A. Yeredor, and Z. Koldovský, A fast asymptotically efficient algorithm for blind separation of a linear mixture of block-wise stationary autoregressive processes, *Proc. ICASSP'09*, 2009.

[28] Z. Koldovský, J. Málek, P. Tichavský, Y. Deville, and S. Hosseini, Extension of EFICA Algorithm for Blind Separation of Piecewise Stationary Non Gaussian Sources, *Proc. ICASSP 2008*, pp. 1913-1916, 2008.

[29] Z. Koldovský, P. Tichavský and E. Oja, Cramér-Rao Lower Bound for Linear Independent Component Analysis, *Proc. ICASSP 2005*, Philadelphia, Vol. 3, pp. 581 - 584, March 2005.

[30] S. Dégerine and A. Zaidi, Separation of an instantaneous mixture of gaussian autoregressive sources by the exact maximum likelihood approach, *IEEE Trans. on Signal Processing*, Vol. 52, pp. 1492–1512, 2004.

[31] E. Doron, A. Yeredor, and P. Tichavský, Cramér-Rao lower bound for blind separation of stationary parametric gaussian sources, *IEEE Signal Processing Letters*, Vol. 14, pp. 417–420, 2007.

[32] P. Tichavský and A. Yeredor, Fast Approximate Joint Diagonalization Incorporating Weight Matrices, *IEEE Transactions of Signal Processing*, Vol. 57, No. 3, pp. 878-891, 2009.

[33] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, *IEEE Transactions on Neural Networks*, Vol. 10, No. 3, pp. 626-634, 1999.

[34] P. Tichavský, Z. Koldovský, and E. Oja, Asymptotic Performance Analysis of the Fixed-Point Algorithm (FastICA) for Independent Component Analysis, *Proc. IEEE Statistical Signal Processing Workshop*, Bordeaux, France, July 2005.

[35] T.P. Jung, C. Humphries, T.W. Lee, S. Makeig, Extended ICA removes artifacts from electroencephalographic recordings, *Advances in Neural Information Processing Systems*, Vol. 10, MIT Press, 1998.

[36] A. Hyvärinen, J. Hurri, P.O. Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*, Springer, ISBN: 978-1-84882-490-4, 2009.

[37] K. Kiviluoto, E. Oja, Independent Component Analysis For Parallel Financial Time Series, *Proc. ICONIP '98*, Vol. 2, pp. 895-898, Tokyo, Japan, 1998.

[38] S. Malaroiu, K. Kiviluoto, E. Oja, Time Series Prediction with Independent Component Analysis, *Proc. International Conference on Advanced Investment Technology*, Gold Coast, Australia, 2000.

[39] J. Málek, Z. Koldovský, S. Hosseini, Y. Deville, A variant of algorithm EFICA with adaptive parametric score function estimator, *Proc. ECMS 2007*, 2007.

[40] Z. Koldovský, J. Málek, P. Tichavský, Y. Deville, and S. Hosseini, Blind Separation of Piecewise Stationary NonGaussian Sources, *Signal Processing*, Volume 89, Issue 12, pp. 2570-2584, ISSN 0165-1684, 2009.

[41] K.Kokinakis, A. K. Nandi, Exponent parameter estimation for generalized Gaussian probability density functions with application to speech modeling, *Elsevier Signal Processing*, pp. 1852-1858, 2005.

[42] K.Kokinakis, A. K. Nandi, Flexible score functions for blind separation of speech signals based on generalized gamma probability density functions, *Proc. ICASSP 2006*, pp. 1217-1221, 2006.

[43] J. Karvanen, V. Koivunen, Blind separation methods based on Pearson system and its extensions, *Elsevier Signal Processing*, pp. 663-673, 2002.

[44] D.-T. Pham, P.Garrat, C. Jutten, Separation of mixture of independent sources through a maximum likelihood approach. *Proc. EUSIPCO 1992*, pp. 771-774, 1992.

[45] S. Hosseini, R. Guidara, Y. Deville, Ch. Jutten, Markovian Blind Image Separation, *ICA 2006*, Springer-Verlag, pp.106-114, 2006.

[46] Ch. L. Brown, R. F. Brcich, A. Taleb, Suboptimal robust estimation using rank score functions, *Proc. ICASSP 2003*, pp. 753-756, 2003.

[47] Y. Deville, M. Benali, F. Abrard, Differential source separation for underdetermined instantaneous or convolutive mixtures: concept and algorithms, *Signal Processing*, Vol. 84, Issue 10, pp. 1759-1776, 2004.

[48] T. W. Lee, M. Girolami, T. J. Sejnowski, Independent Component Analysis using an Extended Infomax Algorithm for Mixed Sub-Gaussian and Super-Gaussian Sources, *Neural Computation*, Vol. 11, No. 2, pp. 417-441, 1999.

[49] D.-T. Pham, Blind Separation of Non Stationary Non Gaussian Sources, *Proc. EUSIPCO 2002*, Toulouse, France, 2002.

[50] [Online] The Matlab*TM* implementation of the Extended EFICA algorithm: `http://itakura.ite.tul.cz/zbynek/eefica.htm`.

[51] C. Févotte, R. Gribonval, and E. Vincent, BSS EVAL toolbox user guide, IRISA, Rennes, France, Tech. Rep. 1706, 2005, [Online], `http://www.irisa.fr/metiss/bss eval/`

[52] C. Févotte, A. Debiolles, and C. Doncarli. Blind separation of FIR convolutive mixtures: application to speech signals, *Proc. 1st ISCA Workshop on Non-Linear Speech Processing*, 2003.

[53] S. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang, Multichannel Blind Deconvolution and Equalization Using the Natural Gradient, *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications*, Paris, France, pp. 101-104, April 1997.

[54] S. C. Douglas, H. Sawada, and S. Makino, Natural gradient multichannel blind deconvolution and speech separation using causal FIR filters, *IEEE Trans. Speech Audio Processing*, Vol. 13, No. 1, pp. 92-104, Jan. 2005.

[55] H. Bousbia-Salah, A. Belouchrani, and K. Abed-Meraim, Jacobi-like algorithm for blind signal separation of convolutive mixtures, *IEE Elec. Letters*, Vol. 37, No. 16, pp. 1049-1050, Aug. 2001.

[56] H. Buchner, R. Aichner, and W. Kellermann, A Generalization of Blind Source Separation Algorithms for Convolutive Mixtures Based on Second-Order Statistics, *IEEE Trans. on Speech and Audio Processing*, Vol. 13, No. 1, pp. 120-134, Jan. 2005.

[57] E. Vincent, R. Gribonval, and M.D. Plumbley, Oracle estimators for the benchmarking of source separation algorithms, *Signal Processing*, Vol. 87, No. 8, pp. 1933-1950, 2007.

[58] P. Smaragdis, Blind separation of convolved mixtures in the frequency domain, *Neurocomputing*, Vol. 22, pp. 21-34, 1998.

[59] N. Mitianoudis and M. E. Davies, Audio Source Separation of Convolutive Mixtures, *IEEE Trans. on Speech Audio Process.*, Vol. 11, No. 5, pp. 489-497, Sep. 2003.

[60] H. Sawada, R. Mukai, S. Araki, and S. Makino, A robust and precise method for solving the permutation problem of frequency-domain blind source separation, *IEEE Trans. Speech Audio Processing*, Vol. 12, No. 5, pp. 530-538, 2004.

[61] F. Nesta, P. Svaizer, M. Omologo, "A novel robust solution to the permutation problem based on a joint multiple TDOA estimation", *Proc. IWAENC 2008*, Seattle, USA, 2008.

[62] Z. Koldovský, P. Tichavský, Time-domain Blind Audio Source Separation Using Advanced Component Clustering and Reconstruction, *Proc. HSCMA 2008*, pp. 216-219, 2008.

[63] Z. Koldovský and P. Tichavský, Time-Domain Blind Separation of Audio Sources on the basis of a Complete ICA Decomposition of an Observation

Space, accepted for publication in *IEEE Trans. on Speech, Audio and Language Processing*, April 2010.

[64] J. C. Principe, B. de Vries, Guedes de Oliveira, "Generalized feedforward structures: a new class of adaptive filters", *Proc. ICASSP 1992*, Vol. 4, pp. 245-248, March 1992.

[65] Hild II, K. E., Erdogmuz, D., Principe, J. C.: Experimental upper bound for the performance of convolutive source separation methods, *IEEE Trans. on Signal Processing*, Vol. 54, No. 2, pp. 627-635, 2006.

[66] Ch. H. Knapp, G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Transactions on Signal Processing*, Vol. 24, No. 4, 1976.

[67] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning (2nd ed.)*, New York: Springer. pp. 520–528, ISBN 0-387-84857-6, 2009.

[68] J. Málek, Z. Koldovský, J. Žďánský, J. Nouza, Enhancement of Noisy Speech Recordings using Blind Source Separation, *Proc. Interspeech 2008*, Brisbane, Austrálie, 2008.

[69] J. V. Oliveira , W. Pedrycz, *Advances in Fuzzy Clustering and its Applications*, Wiley-Interscience, 2007.

[70] R. Krishnapuram, A. Joshi, O. Nasraoui, L. Yi, Low-Complexity Fuzzy Relational Clustering Algorithms for Web Mining, *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 4, pp. 595 - 607, 2001.

[71] R. J. Hathaway, J. C. Bezdek, J. W. Davenport, On relational data versions of c-means algorithm, *Elsevier - Pattern recognition letters*, No. 17, pp. 607-612, 1996.

[72] J. C. Bezdek, Fuzzy Mathematics in Pattern Classification, *PhD Thesis at Cornell University*, Ithaca, USA, 1973.

[73] R. J. Hathaway, J. W. Davenport, Relational duals of the c-means clustering algorithms, *Pattern Recognition*, Vol. 22, pp. 205–212, 1989

[74] A. Vandecatseye, et al., The COST278 pan-European Broadcast News Database, *Proceedings of 4th International Conference on Language Resources and Evaluation*, Portugal, 2004.

[75] J. Žďánsky, P. Červa, J. Silovský, J. Nouza, Acoustic Model Management Strategies for Improved Automatic Transcription of Broadcast Programs, *Proc. SPECOM'2007*, Russia, pp. 503-508, 2007.

[76] J. Málek, Z. Koldovský, and P. Tichavský, "Adaptive Time-Domain Blind Separation of Speech Signals," in Latent Variable Analysis and Signal Separation, *Lecture Notes in Computer Science*, Volume 6365, pp. 9-16, ISBN: 978-3-642-15994-7, Springer, 2010.

[77] J. P. Rousseeuw, Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis, *Computational and Applied Mathematics*, Vol. 20, pp. 53-65, 1987.

[78] Z. Koldovský and P. Tichavský, A Comparison of Independent Component and Independent Subspace Analysis Algorithms, *EUSIPCO 2009* , pp. 1447-1451, Glasgow, Scotland, 2009.

[79] F. Nesta, P. Svaizer, M. Omologo, A BSS method for short utterances by a recursive solution to the permutation problem, *Proc. SAM 2008*, Darmstadt, Germany, 2008.

[80] O. Hnilička, J. Málek, K. Paleček and Z. Koldovský, "A Fast C++ Implementation of Time-domain Blind Speech Separation Algorithm", to be published in *Proc. 20th Czech-German Workshop on Speech Processing*, Prague, 2010

[81] J. Málek: Bayesian classifier for medical data from doppler unit, *Acta Polytechnica*, Vol.46, No. 4, pp. 21-22, 2006.

[82] J. Málek, J.Nouza, T.Klimovič: Automatic classifiers for medical data from doppler unit, *Radioengineering*, Vol.16, No.2, pp. 62-66, 2007.

[83] Transatlantic inter-society consensus (TASC), Management of Peripheral Arterial Disease (PAD). *International Angiology*, Suppl. 1, 2000.

[84] Rozman, J. et al., Diagnostics of ischemic disease of lower limbs via Doppler measuring of blood flow velocity (in Czech), *Lékař a technika*, pp.76-78, pp.100-102, pp. 120-123, Praha, 1995.

[85] Rose, S. C., Non-invasive Vascular Laboratory for Evaluation of Peripheral Arterial Occlusive Disease, *Journal of Vascular and Interventional Radiology*, pp.1107-1114, 2000.

[86] Lee, H.-G., Yum, M.-K., Fourier Transformation of Arterial Doppler Waveforms of the Lower Extremity, *Journal of clinical ultrasound*, Vol. 32, No. 6, pp. 277-285, 2004.

[87] Devier, P. A., Kittler, J., Pattern Recognition: A Statistical Approach, Prentice-Hall, 1982.

[88] Linde, Y., Buzo, A., Gray, R. M., An Algorithm for Vector Quantizer Design, *IEEE Transactions on Communications*, pp. 702-710, 1980.

[89] [Online] Free clipart galery, `http://www.openclipart.org/`

List of author's papers

[v1] J. Málek: Bayesian classifier for medical data from doppler unit, *Acta Polytechnica*, Vol.46, No. 4, pp. 21-22, 2006.

[v2] J. Málek, J.Nouza, T.Klimovič: Automatic classifiers for medical data from doppler unit, *Radioengineering*, Vol.16, No.2, pp. 62-66, 2007.

[v3] J. Málek, Z. Koldovský, S. Hosseini, Y. Deville, A variant of algorithm EFICA with adaptive parametric score function estimator, *Proc. ECMS 2007*, Liberec, 2007.

[v4] Z. Koldovský, J. Málek, P. Tichavský, Y. Deville, and S. Hosseini, "Extension of EFICA Algorithm for Blind Separation of Piecewise Stationary Non Gaussian Sources", *Proc. ICASSP 2008*, pp. 1913-1916, ISBN: 1-4244-1484-9, 2008.

[v5] J. Málek, Z. Koldovský, J. Žďánský, J. Nouza, Enhancement of Noisy Speech Recordings using Blind Source Separation, *Proc. Interspeech 2008*, Brisbane, Austrálie, 2008.

[v6] Z. Koldovský, J. Málek, P. Tichavský, Y. Deville, and S. Hosseini, "Blind Separation of Piecewise Stationary NonGaussian Sources", *Signal Processing*, Volume 89, Issue 12, pp. 2570-2584, ISSN 0165-1684, 2009.

[v7] J. Málek, Z. Koldovský, and P. Tichavský, "Adaptive Time-Domain Blind Separation of Speech Signals," in Latent Variable Analysis and Signal Separation, *Lecture Notes in Computer Science*, Volume 6365, pp. 9-16, ISBN: 978-3-642-15994-7, Springer, 2010.

[v8] Z. Koldovský, P. Tichavský, and J. Málek, "Subband Blind Audio Source Separation Using a Time-Domain Algorithm and Tree-Structured QMF Filter Bank," in Latent Variable Analysis and Signal Separation, *Lecture Notes in Computer Science*, Volume 6365, pp. 25-32, ISBN: 978-3-642-15994-7, Springer, 2010.

[v9] Z. Koldovský, P. Tichavský, and J. Málek, "Time-Domain Blind Audio Source Separation Method Producing Separating Filters of Generalized Feedforward Structure," in Latent Variable Analysis and Signal Separation, *Lecture Notes in Computer Science*, Volume 6365, pp. 17-24, ISBN: 978-3-642-15994-7, Springer, 2010.

[v10] O. Hnilička, J. Málek, K. Paleček and Z. Koldovský, "A Fast C++ Implementation of Time-domain Blind Speech Separation Algorithm", to be published in *Proc. 20th Czech-German Workshop on Speech Processing*, Prague, 2010.