Blind Source Separation for Speaker Recognition Systems

Michael Unverdorben, Martin Rothbucher, Klaus Diepold

Technical Report

Blind Source Separation for Speaker Recognition Systems

Michael Unverdorben, Martin Rothbucher, Klaus Diepold

March 25, 2014

Lehrstuhl für Datenverarbeitung
Technische Universität München

Michael Unverdorben, Martin Rothbucher, Klaus Diepold. Blind Source Separation for Speaker Recognition Systems. . Technische Universität München, LDV. March 25, 2014.

Abstract

In this thesis, a combined blind source separation (BSS) and speaker recognition approach for teleconferences is studied. By using a microphone array, consisting of eight microphones, different methods to perform overdetermined independent vector analysis (IVA) are compared. One method is to select a subset of microphones or all available microphones to perform IVA. The second method, the so called subspace method, that utilizes a principal component analysis (PCA) for dimensionality reduction, is applied prior to IVA.

For the evaluation of IVA, the BSS Eval toolbox is used to calculate the source to distortion ratio (SDR), the source to interferences ratio (SIR) and the source to artifacts ratio (SAR), that indicate the quality of the separation.

The speaker recognition system is based on Gaussian mixture models (GMMs), that are trained on the mel frequency cepstral coefficients (MFCCs) of each speaker. The performance of the speaker recognition is measured by the diarization error rate (DER).

The evaluation results of the speaker recognition show, that a combined BSS and speaker recognition can increase the performance of the speaker recognition system. For the case of two simultaneously active speakers, the rate of detecting both speakers correctly could be improved from 0% without separation to 66% with separation in an anechoic room. For an echoic office room 57% could be achieved.

Contents

1. Introduction **7**
- 1.1. Motivation . . . 7
- 1.2. Objectives . . . 8
- 1.3. Previous Work . . . 9
- 1.4. Related Work . . . 9
- 1.5. Outlook . . . 10

2. Background **11**
- 2.1. Source Separation . . . 11
  - 2.1.1. Overview . . . 11
  - 2.1.2. Blind Source Separation . . . 12
  - 2.1.3. Independent Vector Analysis . . . 16
  - 2.1.4. Overdetermined Blind Source Separation . . . 19
  - 2.1.5. Subspace Method . . . 21
- 2.2. Speaker Recognition . . . 22
  - 2.2.1. Fundamentals of Speaker Recognition . . . 22
  - 2.2.2. Preprocessing . . . 23
  - 2.2.3. Feature Extraction . . . 24
  - 2.2.4. Classification . . . 24
- 2.3. Evaluation . . . 25
  - 2.3.1. Evaluation Criteria for Source Separation . . . 25
  - 2.3.2. Evaluation Criteria for Speaker Recognition . . . 27

3. Overdetermined Independent Vector Analysis **29**
- 3.1. Microphone Array . . . 29
- 3.2. Basic IVA Implementation . . . 30
- 3.3. PCA Subspace Method Implementation . . . 31
- 3.4. Evaluation Data Set for IVA . . . 34
- 3.5. Graphical User Interface . . . 38
- 3.6. Evaluation Results for the Anechoic Room Recordings . . . 38
  - 3.6.1. Evaluation Results for IVA with Two Microphones . . . 40
  - 3.6.2. Evaluation Results for IVA with More Than Two Microphones . . . 44
  - 3.6.3. Evaluation Results for IVA with PCA Subspace Method . . . 47
- 3.7. Evaluation Results for the Echoic Office Room Recordings . . . 48
  - 3.7.1. Evaluation Results for the Basic IVA Implementation . . . 48

3.7.2. Evaluation Results for IVA with PCA Subspace Method 48
3.8. Summary of the Evaluation Results 51

4. Joint Source Separation and Speaker Recognition **53**
4.1. The Speaker Recognition System 53
4.1.1. Model Training 53
4.1.2. Speaker Recognition 55
4.1.3. Application of the Speaker Recognition to the Separated Signals . . 55
4.2. Evaluation Data Set for Speaker Recognition 56
4.3. Evaluation of the Joint Source Separation and Speaker Recognition 60
4.3.1. Evaluation for One Active Speaker 60
4.3.2. Evaluation for Two Active Speakers 62

5. Concluding Remarks **67**
5.1. Conclusion 67
5.2. Future Work 69

A. Appendix **71**
A.1. DVD Content 71
A.2. List of the Evaluated Microphone Combinations 71
A.3. List of all Functions and Scripts 73
A.4. SDR, SIR, SAR Values for 2 Microphones for the Anechoic Recordings . . . 75

Bibliography **85**

1. Introduction

In times of global networking, teleconferencing gets more and more important. Using teleconferencing systems saves a lot of time and traveling expenses. Today's teleconferencing systems can provide a high quality of the transmitted sound, recorded in conferencing rooms, but high quality alone is not sufficient for the requirements of future teleconferencing systems.

A major problem in teleconferencing is, when more people in one room are talking at the same time. In real situations a listener can easily distinguish between two simultaneously talking speakers and bring the speaker of interest into his focus. This ability of humans is called the *cocktail party effect* [4]. In the scenario of a teleconference, where the utterances of several speakers are mixed together in one audio channel, it is no longer as easy as before to distinguish between two speakers, because the listener has no geometrical information about the positions of the speakers. In long-lasting conferences this can be very annoying and destructive for the flow of a conversation. So it would be great to have for every speaker a separate channel, which contains only parts belonging to his voice. For this purpose, source separation can be used.

When we have obtained separated signals, containing only utterances of one speaker in each channel, we can apply these signals to a speaker recognition, to find out at what time which speaker was active.

1.1. Motivation

There are many possible ways to solve the problem of source separation and speaker recognition [4]. Most of the algorithms, solving this problem, perform well and are also able to work online (that means in real time). This is very important for a teleconferencing system, because too long delays due to long computation times decrease the performance of the system. Thus, in most cases there has to be made a compromise between quality and computational complexity.

Although a low computational complexity is important, in this thesis we focus mainly on the quality and assume that there is enough computer performance to perform the calculations in real time. The reason for this approach is that we want to study, what separation results and speaker recognition results can be achieved, if there are no constraints regarding the computation time. This can be used for the future, when more powerful computers are available, or if we want want to analyze a recording of a meeting offline, where the

computation time is not an issue. It would be very interesting to see, how the performance of a speaker recognition can be influenced by increased separation results.

Furthermore, in most cases of the source separation, there are only as much microphones used as there are speakers. This is called the *determined* case. However, in this work a microphone array is utilized, that consists of eight microphones, which are arranged circularly. So in this case we have more microphones than speakers, because it is very unlikely that eight persons at a conference are talking at the same time. This case is called the *overdetermined* case. By using more microphones for the source separation as needed, we get some redundancy. With this redundancy we might be able to improve the separation results, if we find a suitable way to use this redundancy.

1.2. Objectives

The objective of this thesis is to perform a blind source separation and apply a speaker recognition to the separated signals. In the following, an overview of the objectives of this thesis is given.

Since here, the scenario of a conference is assumed, a circular microphone array, containing eight microphones is used to record the participants of the conference.

For the case that multiple speakers are talking at the same time, blind source separation (BSS) is applied to separate the utterances of the different speakers. For blind source separation, the method of independent vector analysis (IVA) [17] is utilized.

To the separated signals, a speaker recognition is applied, in order to identify the current speakers and to assign each separated channel to a speaker.

Figure 1.1 shows an overview of the system, that is intended for this thesis. The system can be divided into three components. These three components are the microphone array, the BSS and the speaker recognition.

The aim of this thesis is, to find out, how these components can be connected, to obtain good source separation results as well as good speaker recognition results.

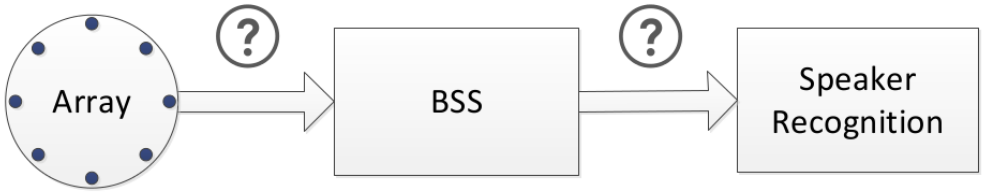


Figure 1.1.: System overview

Here, eight microphones are used, for recording conferences, which is much more than needed. We spend some redundancy, in order to yield a good source separation. For this

overdetermined case, a solution has to be found, how the best separation results can be achieved.

It also has to be determined, how to connect the blind source separation with the speaker recognition, in order to get a good recognition rate.

Recordings in different acoustical environments have to be made for the evaluation of the source separation and for the evaluation of the speaker recognition. Here, recordings in an anechoic room and an echoic office room have to be made.

All components of this system are implemented in Matlab. There are already implementations available for performing IVA [8] and speaker recognition [14], that are used as basis for this thesis.

1.3. Previous Work

At the Institute for Data Processing, there has already been done a lot of work on topics, that are relevant for this thesis. Matlab implementations exist for source separation and speaker recognition. Also a microphone array, containing eight microphones, already exists. Therefore, a lot of these things can be used in this thesis and do not have to be developed completely. This makes it possible to cover both blind source separation and speaker recognition in one thesis.

The following theses are relevant for this thesis:

- **Christian Denk, *Robotic sound source separation using independent vector analysis*** [8]: In this work a BSS algorithm, called independent vector analysis (IVA) has been implemented, which will be used in this thesis for performing source separation.
- **Christoph Kozielski, *Online speaker recognition for teleconferencing systems*** [14]: In this thesis a speaker recognition system has been implemented. This implementation will be our basis for performing speaker recognition in this thesis.
- **Johannes Feldmaier, *Sound localization and separation for teleconferencing systems*** [9]: In this thesis a source localization and separation system, using beamforming and geometric source separation (GSS) has been developed. A microphone array, containing eight microphones, has also been used. In this thesis, the same microphone array will be used for recording speech.

1.4. Related Work

Only very little work on speaker recognition or diarization systems for overlapping speech can be found in the literature, especially for the case of a combined blind source separation and speaker recognition. Most state-of-the-art speaker recognition systems assign only

one speaker to each speech segment. But for conferences, where two speakers may talk at the same time, these overlaps also have to be detected by the recognition system.

In [1], several possibilities are shown to perform speaker recognition in conference scenarios. The simplest case is to place a table-top microphone in front of each speaker or use close talking microphones. So, each speaker has one individual channel and single channel speaker recognition can be applied to each microphone channel. The advantage of close talking microphones is that the recorded speech signals have a high signal-to-noise ratio. For table-top microphones, the performance can be increased by noise reduction or echo cancellation techniques. One drawback of these two methods is, that cross-talk from one speaker to another speaker's microphone can occur and decrease the performance of the recognition system. The third method is the use of a microphone array and the application of beamforming techniques. This makes it possible to focus on the sources of interest and enhance its signals by filtering and combining the different microphone signals.

A beamforming approach for the detection of multiple speakers during a conference is also proposed in [19]. In this approach, a steered response power - phase transform (SRP-PHAT) localization is combined with a particle filter and a geometric source separation (GSS). The particle filter increases the stability of the localization. The signals, separated by the geometric source separation, are then fed to a speaker recognition.

One approach for detecting overlapping speech, without applying a source separation, is shown in [5]. For the detection of overlapping speech, an overlap detection system is used, that utilized a HMM-based segmenter. The segmenter distinguishes the three classes nonspeech, speech and overlapping speech. When a speech segment has been detected as overlapping speech, the segment is associated with the two most probable speakers. This system can detect maximal two speakers at one time.

In [10], an approach that combines standard acoustic features with compressed-domain video features is proposed to improve the performance of the speaker recognition.

An approach, combining blind source separation with a speaker recognition, like the one in this thesis, was not found during my research. The main difference to the approaches that utilize beamforming is, that by applying blind source separation to the mixtures, recorded by a microphone array, no source localization is needed. Also no knowledge about the microphone positions is needed for the separation.

1.5. Outlook

This thesis consists of five chapters. In Chapter 2 the theoretical background, that is necessary to understand BSS and speaker recognition, is presented. Chapter 3 deals with the application and optimization of independent vector analysis for the overdetermined case. In Chapter 4 the source separation is connected with the speaker recognition. And finally, chapter 5 will summarize all important facts, that have been obtained in this thesis and give some suggestions for future work.

2. Background

In this chapter the theoretical background which is essential for source separation and speaker recognition is introduced. This should give a short overview of the problems and show how to solve them.

2.1. Source Separation

2.1.1. Overview

Source separation deals with the problem of separating sources out of a mixture of sources. In our case the sources are audio signals recorded by microphones. These audio signals in general are utterances of speakers. When more people are speaking simultaneously in an acoustic environment, a mixture of all speakers' signals and noise from other sources arrive at the microphone. For humans, it is no problem to distinguish between different speakers if they listen to a person who is standing next to them, although the environment is very loud and many people are talking at the same time. We are able to focus on the person, we want to listen to, and mask out other speakers. This phenomenon is called the *cocktail party effect* [4]. If we listen to a mono signal, recorded in such a situation, it is not as easy as before for humans to understand the speaker of interest. So we can call the problem of separating sound sources also a *cocktail party problem* in this case. For solving this problem, microphone arrays in combination with source separation methods can be used. There are different approaches to recover the original source signals:

- **Beamforming:** *Beamforming* can be seen as a multidimensional filter in space and time that uses multiple microphones. The microphone signals are delayed and filtered in order to enhance the signals arriving from the source position. This can be seen as a virtual microphone or a beam that is focused on the source. Thus it is called beamforming. More details about beamforming can be found in [9].
- **Blind source separation:** *Blind source separation (BSS)* exploits only the statistical characteristics of the signals which have to be separated. In the case of speech signals we can also use the expression *Blind Speech Separation*.
- **Geometric source separation:** *Geometric source separation (GSS)* combines beamforming with blind source separation (BSS) in order to exploit the advantages of both methods. This method is also explained in detail in [9].

Of course there are much more approaches to perform source separation. But these three methods are the most common methods when using microphone arrays for source separation and these methods have also been investigated at the Institute for Data Processing in previous theses (see [9, 8]). Since in this thesis only blind source separation is treated, only BSS methods are explained in more detail in the following sections.

2.1.2. Blind Source Separation

The task of *blind source separation (BSS)* is the recovering of source signals out of a mixture of different sources without having any prior information about the source signals and the mixing process [17, 4]. For the separation only the mixtures recorded by microphones are available. The sources and the mixing process are assumed to be unknown. Thus the separation is called "blind".

It is assumed that speech signals originating from different talkers at different spatial locations are statistically independent. Thus BSS algorithms try to maximize the statistical independence of the output signals [17].

First of all, let's make some definitions about the source signals, the microphone signals and the noise signals. We assume that there are N different *source signals* $s_i(t)$ with source index $i = 1,\ldots,N$ and time index t. We can write this source signals as a vector

$$\mathbf{s}(t) = (s_1(t), s_2(t), \ldots, s_N(t))^T. \tag{2.1}$$

The observed *microphone signals* $x_j(t)$ with microphone index $j = 1,\ldots,M$, where M is the number of microphones, can be written in vector notation as

$$\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_M(t))^T. \tag{2.2}$$

The *noise signals* $n_j(t)$ can be formulated as vector

$$\mathbf{n}(t) = (n_1(t), n_2(t), \ldots, n_M(t))^T. \tag{2.3}$$

When different sources in a room are active simultaneously, the signals arriving at each microphone are a mixture of the sources. There are different ways how the signals can be mixed together. In general we can distinguish between two main mixture models, the *instantaneous mixture model* and the *convolutive mixture model*.

Instantaneous Mixture Model: This is the simplest case of a mixing process. In this case we have a linear time-invariant mixing system where all signals arrive at the microphones at the same time, weighted by a factor a_{ji} plus some additive noise $\mathbf{n}(t)$ [17]. Thus each observed microphone signal $x_j(t)$ is generated by

$$x_j(t) = \sum_{i=1}^{N} a_{ji} \cdot s_i(t) + n_j(t). \tag{2.4}$$

In matrix notation we can express equation (2.4) in the following way:

$$\begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_M(t) \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{pmatrix} \cdot \begin{pmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{pmatrix} + \begin{pmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_M(t) \end{pmatrix}. \tag{2.5}$$

The factors a_{ji} can be summarized to a *mixing matrix* $\mathbf{A}$ with dimension $M \times N$:

$$\mathbf{x}(t) = \mathbf{A} \cdot \mathbf{s}(t) + \mathbf{n}(t). \tag{2.6}$$

Due to reflections and differences in the propagation time of the sound waves, the instantaneous mixture model cannot be used for real acoustic environments [17]. Thus, for describing the mixing process, we need to use a model which also takes time delays into account. For this purpose we can use the convolutive mixture model.

Convolutive Mixture Model: Due to propagation time and reflections, many delayed and differently weighted versions of the original source signal $\mathbf{s}(t)$ arrive at the microphones. Thus an instantaneous mixture model does not hold for acoustic mixtures.

For acoustic mixtures, the mixing can be described by [17, 7]

$$x_j(t) = \sum_{l=-\infty}^{\infty} \sum_{i=1}^{N} a_{ji}(l) \cdot s_i(t-l) + n(t) = \tag{2.7}$$

$$= \sum_{i=1}^{N} a_{ji}(t) * s_i(t) + n_j(t), \tag{2.8}$$

where l is the delay. This mixture model is called *convolutive mixture model* [18]. In matrix notation we can write equation (2.8) as

$$\begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_M(t) \end{pmatrix} = \begin{pmatrix} a_{11}(t) & a_{12}(t) & \dots & a_{1N}(t) \\ a_{21}(t) & a_{22}(t) & \dots & a_{2N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1}(t) & a_{M2}(t) & \dots & a_{MN}(t) \end{pmatrix} * \begin{pmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{pmatrix} + \begin{pmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_M(t) \end{pmatrix}. \tag{2.9}$$

So we get the equation

$$\mathbf{x}(t) = \mathbf{A}(t) * \mathbf{s}(t) + \mathbf{n}(t). \tag{2.10}$$

It should be noted, that the noise signal $\mathbf{n}(t)$ is omitted in many separation algorithms, but for the sake of completeness, it is mentioned here.

Independent Component Analysis (ICA): An often used method to perform blind source separation is *independent component analysis (ICA)*. As the name suggests, ICA tries to separate the sources by finding independent output signals. It is assumed that the different sources s_i are statistically independent so that [13]

$$p(s_1, s_2, \ldots, s_N) = p_1(s_1) \cdot p_2(s_2) \cdot \ldots \cdot p_N(s_N), \tag{2.11}$$

where $p(\cdot)$ is the probability density function (PDF).

Therefore ICA tries to estimate a *separation matrix* $\mathbf{W}$ that makes the output signals as independent as possible [16]. In the ideal case the separation matrix $\mathbf{W}$ is the inverse of the mixing matrix $\mathbf{A}$, which has been used to describe the instantaneous mixing process in Equation (2.6). For the case, that the number of sources equals the number of microphones[1], this can be written as

$$\mathbf{W} = \mathbf{A}^{-1}. \tag{2.12}$$

So the estimated source signals $\hat{\mathbf{s}}(t) = (\hat{s}_1(t), \ldots, \hat{s}_N(t))^T$ can be calculated by

$$\hat{\mathbf{s}}(t) = \mathbf{W} \cdot \mathbf{x}(t) = \mathbf{A}^{-1} \cdot \mathbf{x}(t). \tag{2.13}$$

Since ICA was designed for instantaneous mixtures, ICA cannot be deployed directly to separate audio mixtures, which are described by the convolutive mixture model. But there is a good solution to circumvent this problem. If we transform the recorded signals from time to frequency domain, the mixture becomes instantaneous, because a convolution in the time domain becomes a multiplication in the frequency domain [4]. If we apply a Fourier transform to the convolutive mixture model, as defined in Equation (2.10), we get

$$\mathbf{X}(\omega) = \mathbf{A}(\omega) \cdot \mathbf{S}(\omega) + \mathbf{N}(\omega). \tag{2.14}$$

Now the convolutive mixture has become an instantaneous mixture in frequency domain and we can apply ICA. This can be seen by comparing it to Equation (2.6). We can now estimate the source signals by finding a separation matrix $\mathbf{W}(\omega)$, for every frequency $\omega = 2\pi f$, that is the inverse of the mixing matrix $\mathbf{A}(\omega)$ in the frequency domain:

$$\mathbf{W}(\omega) = \mathbf{A}^{-1}(\omega). \tag{2.15}$$

Under the assumption that there is no noise, we obtain the estimated source signals $\hat{\mathbf{S}}(\omega) = (\hat{S}_1(\omega), \ldots, \hat{S}_N(\omega))^T$ by the following equation:

$$\hat{\mathbf{S}}(\omega) = \mathbf{W}(\omega) \cdot \mathbf{X}(\omega) = \mathbf{A}^{-1}(\omega) \cdot \mathbf{X}(\omega). \tag{2.16}$$

As speech is non-stationary, a *short-time Fourier transform (STFT)* should be applied under the assumption that the signals are stationary in short blocks [8]. In [8], it is suggested to weight a signal x(n), where n is the number of the current sample, by a *cosine window*

$$w(n) = \begin{cases} 0 & |n| > L \\ \cos(n) & |n| \leq L \end{cases} \tag{2.17}$$

[1] Otherwise, instead of the inverse, the pseudo inverse has to be caluculated

with a window length of L samples. The windowed signals $x_{w,i}(n)$, with window index i, can then be calculated by

$$x_{w,i}(n) = x(n) \cdot w(n - i \cdot S), \tag{2.18}$$

where $S < L$ is the overlap of two neighboring windows.

The windowed blocks can now be transformed into the frequency domain by applying a *discrete Fourier transform (DFT)*. So we get a time-frequency representation $X(f,i)$, where f is the index of the frequency bin and i is a time index denoting the i-th block.

Since in this thesis we mainly use vector notations, the mixtures $X(f,i)$ after the STFT for each frequency bin f are described by

$$\mathbf{x}^f = (x_1^f, x_2^f, \ldots, x_M^f)^T. \tag{2.19}$$

So the separation process of one STFT block for each frequency bin f can be written as

$$\hat{\mathbf{s}}^f = \mathbf{W}^f \cdot \mathbf{x}^f = (\mathbf{A}^f)^{-1} \cdot \mathbf{x}^f. \tag{2.20}$$

The Permutation Problem: Applying a STFT to the mixtures to perform ICA in the frequency domain, as described in Equation (2.20), can be a solution to separate a convolutive mixture. But there is one problem. Since in BSS problems we do not know the true sources $\mathbf{s}^f$ and the mixing matrices $\mathbf{A}^f$, ICA cannot recover the source signals exactly due to some ambiguities. There are two main kinds of ambiguities, the permutation ambiguity and the scaling indeterminacy [4].

Permutation ambiguity means, that when applying BSS to a mixture, we do not know, to which channels the components of the different sources are assigned. In time domain, this permutation would be no problem, because just the channels are permuted. But in frequency domain BSS, for each frequency bin one separation problem is solved and the assignment to the channels can be different for every frequency bin. This means, when transforming the separated signals back to time domain, at each channel the components of different sources can be mixed. Most frequency domain BSS methods try to correct this permutations by a postprocessing step. In Chapter 2.1.3 a frequency domain BSS method is shown, that can prevent the occurrence of permutations and thus needs no additional postprocessing.

The second significant ambiguity of frequency domain BSS is the *scaling indeterminacy*. This indeterminacy occurs because the true scaling of the original sources cannot be estimated by ICA. When the separation is executed for every frequency bin independently, the separated signals may have a different spectrum than the original source signals, even if the separation works perfectly [4]. So, after the separation, a spectral compensation has to be performed in order to recover the true scaling of the frequency components as well as possible.

2.1.3. Independent Vector Analysis

One approach for solving convolutive mixture problems in the frequency domain is called *Independent Vector Analysis (IVA)*, which is promising and seems to be very robust [22]. Here, an overview about the most important features of IVA is given. For more details I refer to [17, 15, 8].

IVA prevents permutation ambiguities from occurring, so that no additional postprocessing for correcting permutations is needed [11]. IVA also solves one ICA problem for each frequency bin, but there is one difference to other methods, that perform a frequency domain ICA. It assumes that the frequency components of each source are dependent among all frequency bins [17]. So the following assumptions are exploit, when performing IVA:

- The components of different sources within one frequency bin are mutually independent.
- The components of one source over all frequency bins are dependent.

Thus the sources can be summarized as a multivariate vector source $\mathbf{s}_i = (s_i^1, s_i^2, \ldots, s_i^F)^T$ for all sources i, where the components s_i^f within each vector source are dependent and the vector sources $\mathbf{s}_i$ of different sources are mutually independent. This multivariate mixture model of IVA [17] is depicted in Figure 2.1 for the case of a 2×2 mixture, containing 2 microphones and 2 sources.

Before the separation is done, whitening can be performed after the STFT to simplify the separation problem [8]. By the whitening process the mixtures become uncorrelated and their variance is equal to 1. In [13] it was shown that for whitened signals the mixing matrix is orthogonal, which reduces the complexity of ICA because we only have to look for orthogonal demixing matrices.

As shown above, the mixing process for a frequency bin f is defined as

$$x^f = \mathbf{A}^f \mathbf{s}^f. \tag{2.21}$$

For the whitening, as shown in [13], a whitening matrix

$$\mathbf{Q}^f = (E\{\mathbf{x}^f \mathbf{x}^{fH}\})^{-\frac{1}{2}} \tag{2.22}$$

can be applied to the mixtures $\mathbf{x}^f$ [13, 8], where $E\{\cdot\}$ is the expectation and $E\{\mathbf{x}^f \mathbf{x}^{fH}\}$ is the correlation matrix of $\mathbf{x}^f$. The whitened mixtures can then be calculated by

$$\mathbf{x}_0^f = \mathbf{Q}^f \mathbf{x}^f. \tag{2.23}$$

In order to assign the frequency components to the right source, the speech sources have to be modeled by a *probability density function (PDF)* p, which is also called *source prior*. Since speech can be modeled as supergaussian, in [8] *spherically symmetric Laplacian distribution (SSL)* and *spherically symmetric exponential norm distribution (SEND)* were used to model the speech sources for IVA.

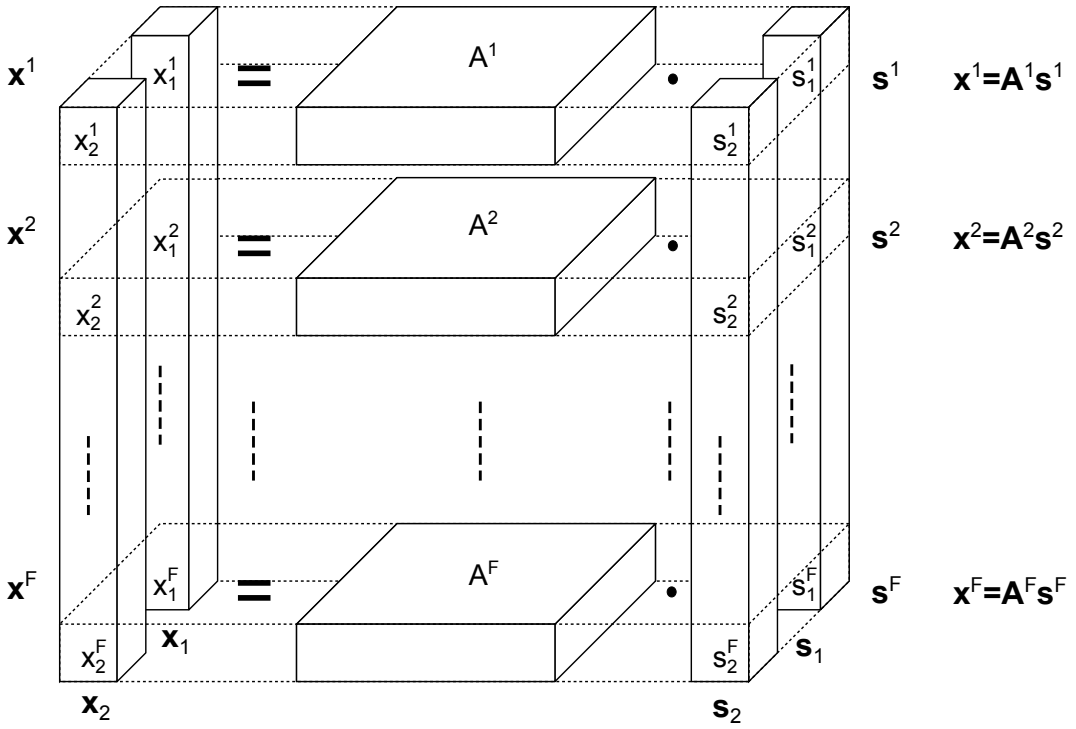


Figure 2.1.: The IVA mixture model [17]. For each frequency bin $f(=1,2,\ldots,F)$ one instantaneous mixture is defined. The components over all frequency bins belonging to one source are assumed to be dependent and are summarized to a vector source, indicated by the vertical pillars.

The *SEND distribution* for a source $\mathbf{s}(t)$ is defined as [8, 15]

$$p_{SEND}(\mathbf{s}(t)) = c\frac{e^{-\sqrt{(2/F)}\|\mathbf{s}\|_2}}{\|\mathbf{s}\|_2^{2F-1}} \quad \forall t, \tag{2.24}$$

where F is the number of discrete frequencies, c is a normalization factor and $\|\mathbf{s}\|_2$ is the L2-norm of $\mathbf{s}$.

The *SSL distribution* is defined as [8, 15]

$$p_{SSL}(\mathbf{s}(t)) = c \cdot e^{-2\cdot\|\mathbf{s}\|_2} \quad \forall t. \tag{2.25}$$

The goal of IVA is to find a set of demixing matrices $\mathbf{W}^1, \ldots, \mathbf{W}^F$ which separate the mixtures according to the distribution of the source prior. If we formulate the demixing process as

$$\hat{\mathbf{y}}^f = \mathbf{W}^f \mathbf{x}_0^f, \tag{2.26}$$

where $\mathbf{x}_0^f$ are the whitened mixtures, a likelihood approach to measure the likelihood of the estimates $\hat{\mathbf{y}}_i$ to the source distributions can be utilized. The *likelihood* C_i of a separated source $\hat{\mathbf{y}}_i$ can be calculated by

$$C_i(\mathbf{W}^1, \ldots, \mathbf{W}^F) = \prod_{n=1}^{T} p(\hat{\mathbf{y}}_n). \tag{2.27}$$

The likelihood of all sources is then

$$C(\mathbf{W}^1, \ldots, \mathbf{W}^F) = \prod_{i=1}^{N} C_i(\ldots) = \prod_{i=1}^{N}\prod_{n=1}^{T} p_i(\hat{\mathbf{y}}_n). \tag{2.28}$$

Because the SSL and SEND distributions are both exponential, it is easier to use the *log-likelihood* instead of the likelihood, which is defined as

$$L(\mathbf{W}^1, \ldots, \mathbf{W}^F) = \ln(C) = \sum_{i=1}^{N}\sum_{n=1}^{T} \ln(p_i(\hat{\mathbf{y}}_n)). \tag{2.29}$$

This log-likelihood function gives us a measure for the "quality" of the used separation matrices [8]. Thus the goal is to maximize this log-likelihood in order to get the optimal separation matrices. This is called the *optimization problem* and can be formulated as

$$\underset{\mathbf{W}^1,\ldots,\mathbf{W}^F}{\arg\max}\, L(\mathbf{W}^1, \ldots, \mathbf{W}^F) \quad \text{s.t.} \quad \mathbf{W}^f \mathbf{W}^{fH} = \mathbf{I} \quad \forall f. \tag{2.30}$$

How exactly the separation matrices $\mathbf{W}^f$ are estimated is shown in [15]. Because this calculation is very complex and is also not part of this theses, let us assume, that an algorithm is given that gives us an estimation of the separation matrices. This algorithm iteratively updates and refines the separation matrices $\mathbf{W}^f$ until the log-likelihood no longer

increases [8]. When the log-likelihood no longer increases, the maximum is found and the algorithm stops. Then the actual matrices $\mathbf{W}^f$, estimated in the last estimation step, are used as separation matrices. Now the signals can be separated by applying Equation (2.26).

Before the separation the input mixtures were whitened to yield uncorrelated mixtures with variance 1, now the effect of the whitening has to be reverted. Otherwise the separated signals would not sound like human voice. This step is also called *spectral compensation*. For more details I refer to [8].

As last step by applying an *inverse STFT* the separated signals can be transformed back to time domain and we yield separated sources.

2.1.4. Overdetermined Blind Source Separation

Most BSS algorithms assume, that the number of sources *N* is equal to the number of mixtures *M*. This case is called *determined case* [17]. But in a realistic scenario it is not reasonable to assume, that there is a fixed number of sources that does not change. Thus we assume that there are more microphones than sources ($M > N$). We call this *overdetermined case*, which is illustrated in Figure 2.2. There are N sources $s_1, \ldots, s_N$, which are in our scenario different speakers, talking anywhere in a room. The N speakers are recorded by M microphones $m_1, \ldots, m_M$. The arrows indicate that at every microphone a mixture of all sources is recorded. For simplicity, influences from noise sources or due to reflections, are omitted here.

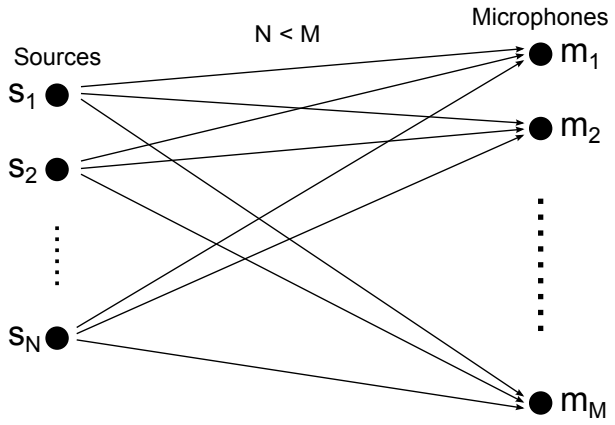


Figure 2.2.: Overdetermined mixture

By using more microphones than sources we obtain some redundancy which might be used to improve the performance of the source separation. For this purpose we have to find a way how to use this redundancy efficiently for the separation. There is also a third case, the *underdetermined case*, where M is less than N. Due to the fact that this thesis

only covers determined and overdetermined mixtures, the underdetermined case is not further discussed.

There are the following possibilities to solve the overdetermined separation problem:

1. **Separation with all available microphone signals:** This is the easiest way to perform overdetermined BSS. But it is also computationally expensive, because the separation algorithm tries to find M independent components, although there are only N real sources. Since we do not care about execution time in this thesis, this method could be a possible solution.

2. **Change of the separation model:** Standard BSS algorithms based on ICA assume, that the mixing matrix $\mathbf{A}$ and the unmixing matrix $\mathbf{W}$ are square [13]. This means that the number of mixtures is equal to the number of sources and the number of output signals, calculated by the BSS algorithm, is equal to the number of input signals. This assumption makes the computation of the independent components more easy, because the unmixing matrix $\mathbf{W}$ is the inverse of the mixing matrix $\mathbf{A}$ [13]. Thus, changing the mixture model from square matrices with dimension $M \times M$ to non square matrices with dimension $M \times N$ would make the separation more difficult. For the mixture model of IVA this would mean, that the mixing matrix $\mathbf{A}^f$ of each frequency bin f changes from a $M \times M$ to a $M \times N$ matrix.

3. **Selection of a subset of microphones:** Another possibility is to use only as much microphone signals as needed for the separation. Theoretically we only need N microphones to separate N speakers, so it would be sufficient to choose only N or $N+1$ microphones out of $M > N$ available microphones. One problem of this approach is to find the microphone combinations which yield the best separation results, because we don't know them a priori. Depending on the number of speakers and the positions of the speakers in a room, different microphone combinations could achieve varying results. Therefore, if we want to apply this method, we have to know the number and the positions of the speakers and also determine, which microphone combinations are suitable for every particular situation.

4. **Dimension Reduction:** As overdetermined mixtures contain redundancy, it is possible to find a smaller set of variables which describe the recorded mixtures with less redundancy and less dimensions than with the complete recording [13]. In [2, 24, 12, 3] a method, called *subspace method*, based on a *principal components analysis (PCA)* is proposed to reduce the dimension of the input mixtures without losing much information. This method can be used as preprocessing step for the separation with IVA. This approach utilizes an eigenvalue decomposition of the mixed signals under the assumption, that the energy of the N directional source signals is concentrated on the N dominant eigenvalues [2]. So the signal can be divided into a signal subspace, spanned by the eigenvectors belonging to the N largest eigenvalues, and a noise subspace spanned by the eigenvectors belonging to the $M-N$

smallest eigenvalues. Therefore, the dimension of the mixtures can be reduced by removing the noise subspace. This method looks promising since through dimension reduction the complexity of IVA is reduced and the influence of noise can also be reduced. More details to the subspace method are explained in Chapter 2.1.5.

2.1.5. Subspace Method

As mentioned above, the *subspace method* is a promising preprocessing step for IVA, as it can reduce the dimension of the separation problem from an overdetermined problem to a determined problem without losing much information. So it seems to be a good and efficient solution for the overdetermined BSS problem. Also its ability to suppress the influence of noise, if the number of microphones M is larger than the number of sources N, is beneficial [12]. Another advantage of the subspace method, that utilizes a principal components analysis (PCA), is that in the whitening step of IVA also PCA is used. So the subspace method can easily be integrated into the whitening process of IVA.

The first step of the subspace method is to perform a PCA, which uses the *spatial correlation matrix* $\mathbf{R}^f$ of the mixtures $\mathbf{x}^f$ for each frequency bin f. Since we have convolutive mixtures, a PCA is performed in the time-frequency domain after applying a STFT for each frequency bin f. In [12], the spatial correlation matrix is calculated by

$$\mathbf{R}^f = \mathrm{E}\{\mathbf{x}^f \mathbf{x}^{fH}\}. \tag{2.31}$$

After applying an eigenvalue decomposition, there are M eigenvalues $\lambda_1^f, \lambda_2^f, \ldots, \lambda_M^f$ that are sorted by decreasing energy, which can be written as

$$\lambda_1^f \geq \lambda_2^f \geq \cdots \geq \lambda_M^f \tag{2.32}$$

with the corresponding eigenvalues $\mathbf{e}_1^f, \mathbf{e}_2^f, \ldots, \mathbf{e}_M^f$.

When there are N active sources, it is assumed that there are also N dominant eigenvalues [2, 12], which can be described by

$$\lambda_1^f, \ldots, \lambda_N^f \quad \gg \quad \lambda_{N+1}^f, \ldots, \lambda_M^f. \tag{2.33}$$

The N eigenvectors $\mathbf{e}_1^f, \ldots, \mathbf{e}_N^f$ are the basis vectors that span the signal subspace [12]. The remaining eigenvectors $\mathbf{e}_{N+1}^f, \ldots, \mathbf{e}_M^f$ span the noise subspace. So, when removing the noise subspace, the dimension of the signal can be reduced without losing information about the signal of interest. With an eigenvector matrix $\mathbf{E}^f = [\mathbf{e}_1^f, \ldots, \mathbf{e}_N^f]$, containing only the first N eigenvectors and an eigenvalue matrix $\Lambda^f = \mathrm{diag}(\lambda_1^f, \ldots, \lambda_N^f)$, a PCA matrix

$$\mathbf{W}_{PCA}^f = (\Lambda^f)^{-1/2} \mathbf{E}^{fH} \tag{2.34}$$

can be calculated, that filters the mixtures $\mathbf{x}^f$ in every frequency bin f by

$$\mathbf{x}_{PCA}^f = \mathbf{W}_{PCA}^f \mathbf{x}^f. \tag{2.35}$$

The dimension of $\mathbf{x}_{PCA}^{f}$ has been reduced from M to N and the influence of noise has also been reduced by this step. Now IVA can be applied to the mixtures $\mathbf{x}_{PCA}^{f}$, which were also whitened by the PCA.

After the separation it is important to remove the influence of the PCA, because whitened mixtures do not sound like natural speech. This can be included into the spectral compensation stage.

In Figure 2.3 it is shown, how the subspace method can be integrated into IVA.

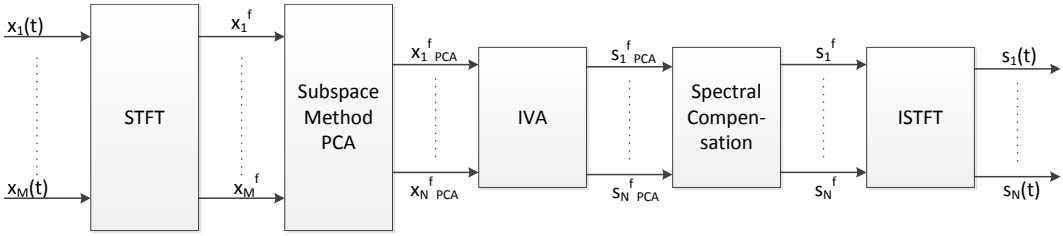


Figure 2.3.: Integration of the subspace method into IVA

2.2. Speaker Recognition

In this chapter, the fundamentals of *speaker recognition* are explained. Most of the theory, explained in this chapter, originates from [14].

2.2.1. Fundamentals of Speaker Recognition

The field of *speaker recognition* can be divided into three general groups [14]:

- **Speaker verification:** Here it is only checked, if a speech sample corresponds to a speaker's identity, which has to be verified. This can be used i.e. for access control.
- **Speaker identification:** We can distinguish between to types of speaker identification: *open-set* and *closed-set* identification. In closed-set identification, one speech sample is compared to all available speaker models in a set of models and the model, that is most likely, is chosen as the speakers identity. In open-set identification, also unknown speakers, that are not included in the set of speaker models, can be detected. Therefore a closed-set speaker identification is extended by a speaker verification to an open-set identification. When no speaker identity can be verified, a new model will be created.
- **Speaker detection:** Speaker detection determines, which speakers are active in an audio stream, that can contain multiple speakers. If we additionally want to know, at what time which speaker was active, speaker detection can be extended by a

segmentation, that identifies the parts in an audio stream, belonging to one speaker. This is also called *speaker diarization*.

A typical speaker recognition system can consist of the following processing steps, as shown in [14]:

- Preprocessing
- Feature Extraction
- Classification

This is also called pattern recognition. These three steps are now explained in more detail.

2.2.2. Preprocessing

Due to the fact, that human voice does not exceed frequencies above $f_{max} = 8\,\mathrm{kHz}$ [14], a sampling frequency of $f_A \geq 2 \cdot f_{max} = 16\,\mathrm{kHz}$ is sufficient for speaker recognition. So, if the input signal has a higher sampling frequency, it can be downsampled to $16\,\mathrm{kHz}$.

Because speech signals have a low-pass characteristic, applying a preemphasis filter in order to amplify the higher frequencies is very useful, since many speaker dependent information is contained in the high-frequent formants [14]. For this we can use a filter with the transfer function

$$H_{pre}(z) = 1 - \alpha \cdot z^{-1}, \tag{2.36}$$

where different values for α yield different frequency responses of the filter.

In order to analyze the spectral characteristics of a digital speech signal $s(k)$, the signal has to be transformed into the frequency domain. Since speech signals are in general non-stationary, a *short-time Fourier transform* (STFT) has to be performed. The STFT of a signal $s(k)$ is defined as [14]

$$\mathrm{STFT}\{s(k)\} \equiv S_k(m,\omega) = \sum_{k=-\infty}^{+\infty} s(k)w(k-m)e^{-j\omega k}, \tag{2.37}$$

where $w(k)$ is a window function that weights and cuts out a short time interval of the signal. By keeping the window size very small, stationarity can be assumed for this signal part. As window function, a *hamming window* [7, 14]

$$w(\tau) = 0.54 + 0.46 \cdot \cos(2\pi\frac{\tau}{T}) \tag{2.38}$$

with $\tau = -\frac{T}{2}, ..., +\frac{T}{2}$ can be used. In [14] a window length of $20-30\,\mathrm{ms}$ with a progress in $5-25\,\mathrm{ms}$ steps is suggested to obtain well extracted features.

2.2.3. Feature Extraction

For describing the characteristics of a speech sample, features have to be found that describe the voice spectrum accurately with a very small number of features, to reduce the dimension. For this task *mel frequency cepstral coefficients* (MFCCs) are a good choice [14]. For calculating the MFCCs, first the signal energy has to be filtered by triangular-band filters that are adjusted to the human auditory system. These triangular filters are called *mel filters*. The *mel-energy* can be calculated by [14]

$$E_{mel}^{(w)} = \sum_{n=0}^{K/2-1} F_{mel}^{(w)}(n)|S(k)|^2 \quad 1 \leq w \leq W, \tag{2.39}$$

where $F_{mel}^{(w)}(n)$ is the frequency response of the w-th filter and K is the number of samples of a frequency segment.

Now, the MFCCs can be calculated by applying a *discrete cosine transform* (DCT) to the logarithm of the mel-energy:

$$c_{MFCC}^{(i)} = \sum_{w=1}^{W} \log(E_{mel}^{(w)}) \cos[i(w-0.5)\frac{\pi}{W}] \quad 1 \leq i \leq M \tag{2.40}$$

Finally, we obtain M MFCCs $c_{MFCC}^{(i)}$, which can now be used as features.

The features can be represented by a feature vector

$$\vec{x}_i = \begin{pmatrix} x_{i,1} \\ \vdots \\ x_{i,N}, \end{pmatrix} \tag{2.41}$$

where for each frame i one vector is calculated. As vector elements $x_{i,j}$ the previously calculated MFCCs $c_{MFCC}^{(i)}$ can be used.

The feature vectors $\vec{x}_i$ of all frames can be summarized to a *feature matrix*

$$\vec{X} = [\vec{x}_1, \vec{x}_2, \dots]. \tag{2.42}$$

Using such a feature matrix has the advantage, that a speech sample can be represented by much less data.

2.2.4. Classification

In the classification step the feature matrix is compared to precalculated speaker models and it is decided to which model the speech sample fits best. In [14], *Gaussian mixture models* (GMM) are suggested for modeling speakers in a conference, since the recognition

has to be text-independent. With a GMM a speaker identity can be represented statistically as a weighted sum of *unimodal Gaussian densities*

$$\mathcal{N}(\vec{x}|\vec{\mu}_k,\Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\{-\frac{1}{2}(\vec{x}-\vec{\mu}_k)^T\Sigma_k^{-1}(\vec{x}-\vec{\mu}_k)\}, \tag{2.43}$$

where $\vec{\mu}_k$ is a mean vector and Σ_k is a covariance matrix, that can be summarized to

$$\vec{\mu} = \{\vec{\mu}_1,\ldots,\vec{\mu}_K\}, \tag{2.44}$$

$$\Sigma = \{\Sigma_1,\ldots,\Sigma_K\}, \tag{2.45}$$

where K is the number of mixture components. A feature vector $\vec{x}$ with dimension N can be modeled by a probability density function

$$p(\vec{x}|\lambda) = \sum_{k=1}^{K} w_k \mathcal{N}(\vec{x}|\vec{\mu}_k,\Sigma_k). \tag{2.46}$$

The weighting factors w_k have to satisfy the constraint

$$\sum_{k=1}^{K} w_k = 1 \quad 0 \leq w_k \leq 1 \tag{2.47}$$

and can be summarized to a vector

$$\vec{w} = \{w_1,\ldots,w_K\}. \tag{2.48}$$

The parameters of the density model can by expressed as

$$\lambda = \{w_k,\vec{\mu}_k,\Sigma_k\} \quad k = 1,\ldots,K. \tag{2.49}$$

2.3. Evaluation

2.3.1. Evaluation Criteria for Source Separation

In order to evaluate the separated source signals, obtained by the source separation algorithms, we need a measure that shows us how good the separation performs.

In this work we use three different measures for the performance measurement, the *Source to Distortion Ratio (SDR)*, the *Source to Interferences Ratio (SIR)* and the *Sources to Artifacts Ratio (SAR)* [21].

If $\hat{s}_j$ is the estimated source with source index j, we can decompose it into

$$\hat{s}_j = s_{target} + e_{interf} + e_{noise} + e_{artif}. \tag{2.50}$$

The term $s_{target} = f(s_j)$ is a version of the original signal s_j modified by an allowed distortion $f \in \mathcal{F}$, where $\mathcal{F}$ is a family of allowed distortions which can be chosen by the

user. The terms e_{interf}, e_{noise} and e_{artif} are the errors arising from interferences, noise and algorithmic artifacts. To calculate all these terms, we need to know the original source signals.

The decomposition can be obtained by means of orthogonal projections. Let us define the three orthogonal projectors [21]

$$P_{s_j} := \prod\{s_j\}, \tag{2.51}$$

$$P_{\mathbf{s}} := \prod\{(s_{j'})_{1\leq j'\leq n}\}, \tag{2.52}$$

$$P_{\mathbf{s,n}} := \prod\{(s_{j'})_{1\leq j'\leq n}, (n_i)_{1\leq i\leq m}\}, \tag{2.53}$$

where $\prod\{y_1,\ldots,y_k\}$ is the orthogonal projector onto the subspace spanned by the vectors $y_1,\ldots,y_k$.

With these three projectors we can calculate the terms of Equation (2.50) as follows [21]:

$$s_{target} := P_{s_j}\hat{s}_j, \tag{2.54}$$

$$e_{interf} := P_{\mathbf{s}}\hat{s}_j - P_{s_j}\hat{s}_j, \tag{2.55}$$

$$e_{noise} := P_{\mathbf{s,n}}\hat{s}_j - P_{\mathbf{s}}\hat{s}_j, \tag{2.56}$$

$$e_{artif} := \hat{s}_j - P_{\mathbf{s,n}}\hat{s}_j. \tag{2.57}$$

Further details can be found in [21].

After the decomposition of the estimated signal $\hat{s}_j$ we can now calculate our performance measures.

The *Source to Distortion Ratio* is defined as the energy ratio of the target signal s_{target} to the sum of all three noise terms e_{interf}, e_{noise} and e_{artif} [21]:

$$SDR := 10\log\frac{\|s_{target}\|^2}{\|e_{interf}+e_{noise}+e_{artif}\|^2}. \tag{2.58}$$

The *Source to Interferences Ratio* is defined as the energy ratio of the target signal s_{target} to the noise error signal e_{noise}:

$$SIR := 10\log\frac{\|s_{target}\|^2}{\|e_{interf}\|^2}. \tag{2.59}$$

The *Sources to Artifacts Ratio* is defined as the energy ratio of the sum of the target signal s_{target} and the interference and noise error signals e_{interf} and e_{noise} to the artifacts error term e_{artif}:

$$SAR := 10\log\frac{\|s_{target}+e_{interf}+e_{noise}\|^2}{\|e_{artif}\|^2}. \tag{2.60}$$

Here, mostly the SDR will be used as measure for the quality of the separation, because it takes into acount all three error types e_{interf}, e_{noise} and e_{artif}.

2.3.2. Evaluation Criteria for Speaker Recognition

In [14, 20] the *diarization error rate* DER is suggested as measure for speaker diarization tasks. The DER is defined as

$$DER = \delta_{miss-error} + \delta_{false-alarm} + \delta_{speaker-error}. \tag{2.61}$$

The components of Equation (2.61) are error rates, that indicate how often different types of errors occur over time. The different error rates are defined as follows:

- **Miss error $\delta_{miss-error}$:** Rate of speech segments that are not assigned as speech. This error occurs, if the voice activity detection (VAD) detects no speech.
- **False alarm $\delta_{false-alarm}$:** Rate of segments that are incorrectly declared as speech. This error occurs, if the voice activity detection declares a segment as speech although there are no active speakers.
- **Speaker error $\delta_{speaker-error}$:** Rate of falsely detected speakers. This error occurs, if the name of a wrong speaker is assigned to a speech segment.

As the DER is mainly designed for detecting single speakers, some modifications have to be done in combination with source separation. For example, if we want to know how the speaker recognition performs after the separation of multiple speakers, it is mostly interesting for us, how often all speakers, talking at the same time, are detected correctly. The question is, how to treat the case, when only one of two active speakers has been detected correctly. Should we calculate the DER for each speaker independently or should we calculate the DER for the detected speaker combination as a whole? Since here the effect of the separation on the speaker recognition is investigated, I decided that it is more useful to treat only the case that all speakers are detected correctly as a right detection. All other cases are treated as errors. But we can divide the *speaker error* into different cases depending on the number of falsely detected speakers.

For the case of two simultaneously active speakers I define the following:

- **Right detection:** Both speakers are detected correctly.
- **Only one speaker correct:** Only one of the two speakers is detected correctly.
- **False detection:** Both speakers detected wrong.
- **Missed detection:** Both speakers are not detected as active.
- **False alarm:** A segment, containing no speech, is detected as speech.

So the *DER* for two active sources can be calculated as

$$DER = \delta_{miss-error} + \delta_{false-alarm} + \delta_{false-detection} + \delta_{only-one}, \tag{2.62}$$

where $\delta_{only-one}$ is the is the error rate of the case when only one speaker has been detected correctly.

If we only want to know, how successful the speaker recognition was, we can also calculate the *accuracy*, which is the rate of correctly detected segments.

The advantage of the DER as measure for speaker recognition is, that we can see the different error types that lead to a bad performance. So it is more useful to find the error causes.

3. Overdetermined Independent Vector Analysis

This chapter deals with the application of blind source separation in teleconferences, using a microphone array. In a typical conference situation, only one person is talking for most of the time. But people do not always wait until the current speaker has stopped talking before they say something. So there can occur some overlaps between two or more speakers. Also interjections from other people, while someone else is speaking, can happen. When more than one speakers are active at the same time, the most likely situation is that there are only two persons talking. Only when different groups of people in the conference room are talking with each other at the same time, or maybe in very emotional discussion everybody wants to say something simultaneously, more than two speakers could be active. But in most of these situations, if more than two people are talking, a separation of their utterances would not make much sense. Thus, a microphone array with two or three microphones would be enough to perform source separation. But here we want to use eight microphones, which is much more than needed. We want to spend some redundancy and see what is possible. The goal is to obtain better separation results in this overdetermined case than in an determined case. We have to find a way, how we can obtain the best separation results by using all eight microphones. It also has to be studied, which microphone combinations with less than eight microphones yield the best results, so that we can compare our results in order to see which method really works best.

3.1. Microphone Array

Here, a circular, planar microphone array, consisting of eight microphones, is used. This microphone array has also been used in the diploma thesis by Johannes Feldmaier [9], who performed geometric source separation. The microphones are uniformly distributed on a circle with radius $r = 0.12\,\mathrm{m}$ and with an angular distance of 45° between the microphones. Figure 3.1 shows the plan view of the microphone array. The microphones are numbered from 1 to 8, increasing counterclockwise. The center of microphone 1 is defined as 0°. This array configuration and the here defined coordinate system are used for all experiments in this thesis. In [9], also some volumetric array configurations were use. This means, that not all microphones are located in one plane, but in all three dimensions. For the case of blind source separation I prefer a circular array, because it is symmetric. Also

no localization is done, contrary to [9], where one microphone of the array is centered and raised to achieve better localization results.

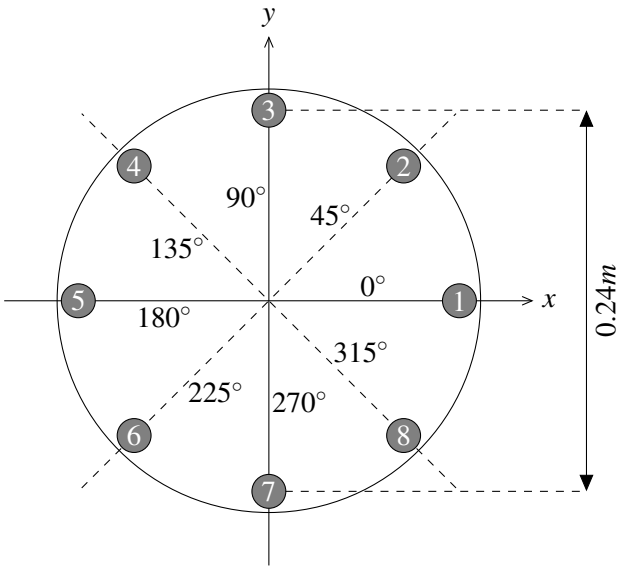


Figure 3.1.: The circular microphone array, consisting of eight microphones, numbered from 1 to 8 at the angular positions 0° to 315° and an angular distance of 45°.

3.2. Basic IVA Implementation

For IVA, a Matlab implementation has already been developed at the Institute for Data Processing by Christian Denk [8]. So this implementation can be used as basis for performing blind source separation in this thesis.

The IVA algorithm consists of the following steps, as described in [8]:

1. **STFT:** A short-time fourier transform is applied to the input mixtures, in order to obtain short blocks that are stationary. At a sampling rate of $48\,\mathrm{kHz}$ and with a window size of 1024 samples, these blocks are $21.3\,\mathrm{ms}$ long. Applying a STFT to convolutive mixtures yields multiple instantaneous ICA problems in the frequency domain. Hence, in each frequency bin exists one instantaneous ICA problem.

2. **Whitening:** Before the signals are separated, whitening is performed to yield uncorrelated mixtures with variance (power) 1.

3. **Separation:** Using a standard instantaneous ICA algorithm in each frequency bin leads to permutation ambiguities among the frequency bins, so permutation alignment would have to be performed after the separation. IVA tries to overcome this

problem by assuming that between the frequency bins there are dependencies, so the separation process prevents permutations and no postprocessing for permutation alignment is needed.

4. **Spectral compensation:** Prior to the separation process, whitening was performed and the signals have the same power over all frequencies. Due to this fact and the scaling ambiguity, a spectral compensation has to be performed, to obtain signals that sound natural.

5. **Inverse STFT:** After the separation and the spectral compensation, the signals can be transformed back into the time domain by applying an inverse STFT. Now we can listen to the separated signals.

This IVA implementation has been designed for determined mixtures. This means, that it tries to find as many independent components as there are input mixtures. So, if there are eight input mixtures, we obtain eight output signals. The question is, what happens, if there are only one or two sources, but eight input mixtures, recorded by the microphones. Does the separation work correctly, or does it influence it in a negative way? Because IVA tries to yield as many independent components as there are input mixtures (in our case eight). But what is, if there are only two independent components? So it has to be determined, how the performance of the separation is affected, when the number of mixtures is higher than the number of sources.

The first thing, we have to investigate is how the separation quality changes with the number of microphones, used as input for the basic IVA algorithm, if there are two active speakers.

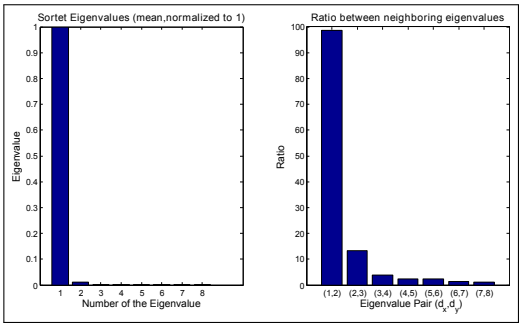
3.3. PCA Subspace Method Implementation

As described in Chapter 2.1.5, the subspace method can be used to solve the overdetermined separation problem. Additionally, it promises to remove some noise.

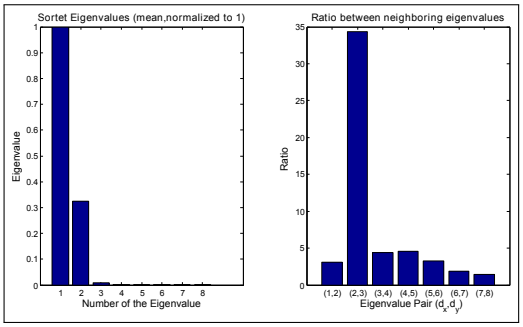
The integration of the subspace method into the basic IVA implementation was simple, since in the whitening stage of IVA, a PCA is already performed to obtain uncorrelated mixtures. But in IVA, all eight principal components are used for the separation, if there are eight microphones available. For the implementation of the subspace method, only as much principal components as needed are used for separation. When N is the number of principal components, only the N eigenvectors, belonging to the N biggest eigenvalues are selected after the eigenvalue decomposition to create the whitening matrix. So after the whitening there are only N remaining signals instead of the original M signals. This reduces the dimension of the input mixtures and also reduces the complexity of the separation problem.

To perform the subspace method, a function (*iva_pca.m*, see Appendix A.3) was created, where the number of desired principal components for the subspace selection can be entered as an input parameter.

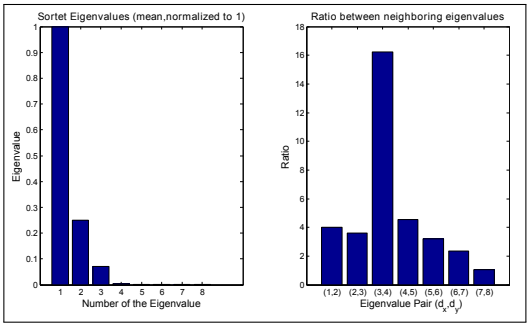
It has to be noted that the number of needed principal components, which is depending on the number of sources, has to be known before applying separation with the subspace method. So we also have to estimate, how much sources are active. The theory tells us, that in a mixture of N source signals there are also N dominant eigenvalues (see Equation (2.33)). So the number of sources could be determined by analyzing the eigenvalues. In Figure 3.2, on the left part of each subfigure, the eigenvalues for different numbers of sources in an anechoic room are shown, sorted by their magnitude. The values in each plot are normalized to the first column.



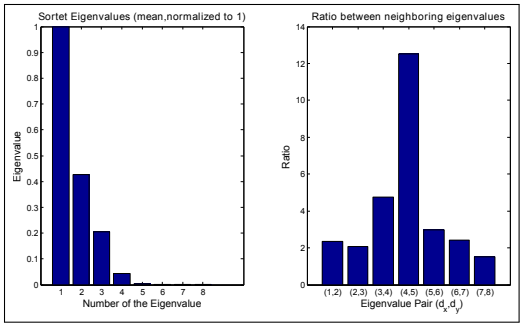
(a) 1 Source



(b) 2 Sources



(c) 3 Sources



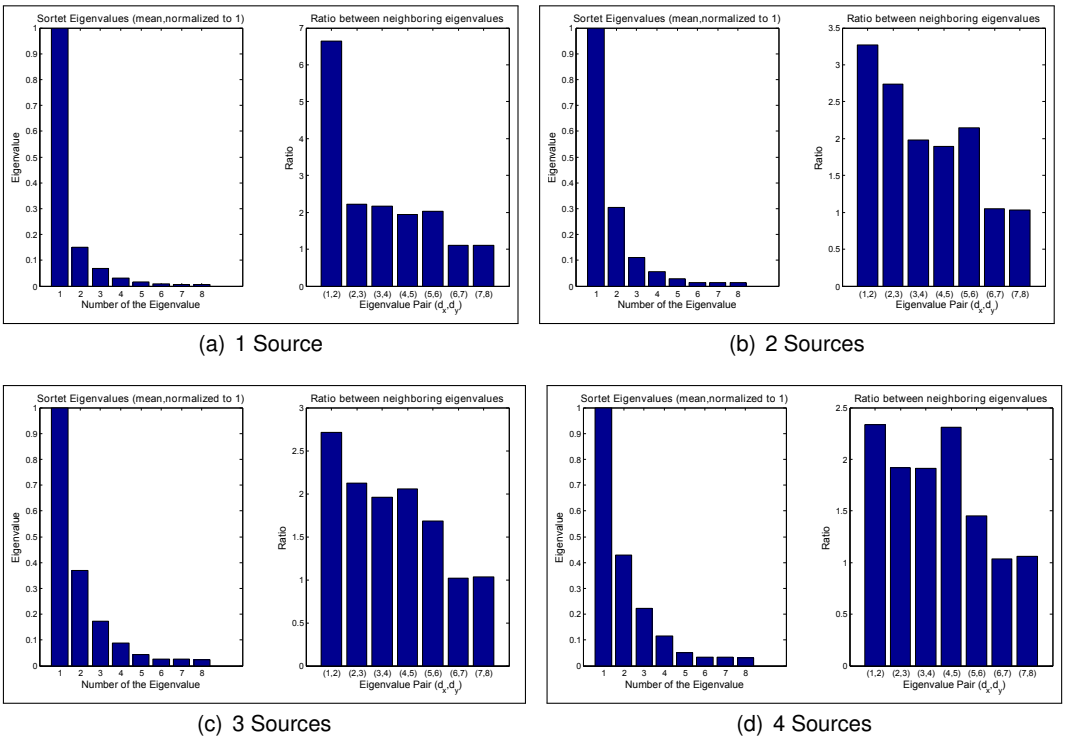
(d) 4 Sources

Figure 3.2.: Eigenvalues for different numbers of active sources in the anechoic room, sorted by their magnitude, on the left of each subfigure. On the right of each subfigure, the ratio between neighboring eigenvalues of the left part of each subfigure can be seen.

As you can see, the first eigenvalue has always the biggest magnitude and depending on the number of sources N, the next $N-1$ eigenvalues are also dominant. The remaining $M-N$ eigenvalues are very small. In order to see the ratio between the eigenvalues, on each subplot's right side, the ratio between two neighboring eigenvalues has been calculated, so that we can see, how big the change from one eigenvalue to the next is.

The position with the greatest ratio corresponds to the number of sources. So we can use this ratio to determine the number of active sources. In the anechoic room, this method works pretty good for the detection of the number of sources. The best results could be achieved, when only frequencies between $700\,\mathrm{Hz}$ and $8\,\mathrm{kHz}$ had been analyzed. This value has been determined by extensive experiments.

When trying to determine the number of sources by this method in an office room, some problems arise. In Figure 3.3 the same arrangement as in Figure 3.2 is shown for an echoic office room.



(a) 1 Source (b) 2 Sources

(c) 3 Sources (d) 4 Sources

Figure 3.3.: Eigenvalues for different numbers of active sources in the office room, sorted by their magnitude, on the left of each subfigure. On the right of each subfigure, the ratio between neighboring eigenvalues of the left part of each subfigure can be seen.

As you can see, now there are more dominant eigenvalues than sources. The number of active sources cannot be determined by just looking at the eigenvalues. Also the ratio between neighboring eigenvalues does not show how many active sources there are. For determining the number of sources, a threshold could be set, calculated for each eigenvalue distribution. But finding a threshold is not easy and for other recordings the threshold can be completely different.

3.4. Evaluation Data Set for IVA

When determining, which number of microphones yields the best results with the basic IVA implementation, we can also investigate, if we find any rules or regularities according to the position of the microphones. When using only two microphones, we can choose $\binom{8}{2} = 28$ different microphone pairs as input signal for the separation. So it would be interesting to see, which microphone combinations yield good results, and which combinations yield poor results, when the positions of the microphones and the speakers are known.

Number of microphones	**Number of combinations**
$m = 2$	$\binom{8}{2} = 28$
$m = 3$	$\binom{8}{3} = 56$
$m = 4$	$\binom{8}{4} = 70$
$m = 5$	$\binom{8}{5} = 56$
$m = 6$	$\binom{8}{6} = 28$
$m = 7$	$\binom{8}{7} = 8$
$m = 8$	$\binom{8}{8} = 1$
Sum	**247**

Table 3.1.: Number of all possible microphone combinations (order not taken into account), when there are 8 microphones available

In Table 3.1 it is shown, how many different combinations are possible for performing the separations with m microphones for $m = 2, \ldots, 8$. Altogether, there are 247 different possibilities to perform BSS with the basic IVA implementation, when there are eight microphones available (in the case of one or two active sources).

In order to evaluate IVA for different microphone combinations, recordings in an anechoic room and in an echoic office room were made. With the recordings in the anechoic room it shall be examined, how the source separation behaves with minimal room reflections and if some regularities can be found relating to the microphone combinations and their geometry. Then, with the recordings in the echoic office room, the influence of room reflections will be investigated and it will be reviewed, if the regularities, which have been found in the anechoic room, are also true in an echoic room.

For the recordings, speech signals of different speakers were played back through loudspeakers, which have been recorded with the microphone array, as introduced in Section 3.1. In a conference, people can sit or stand anywhere around the table. Hence, different recordings for different angular distances between the loudspeakers were made. Always, a pair of speakers, talking simultaneously, was recorded, because this is the most common case, when speaker overlaps occur. So two loudspeakers were positioned around

the array. The first loudspeaker *LS1* was always at the same position at a azimuth of $0°$, related to the coordinate system introduced in Figure 3.1, such that the next microphone to *LS1* is microphone 1. The second loudspeaker *LS2* was positioned at different angular distances to *LS1*. Because the microphone array is symmetric, only distances not greater than $180°$ are considered. Here, the angular distances $25°$, $45°$, $90°$,$135°$ and $180°$ were treated.

In Figure 3.4 the recording configurations in the anechoic room are shown. The distance between the loudspeakers and the middle of the microphone array was $r = 1.5\,\mathrm{m}$. For each of these configurations, the recordings have been performed from two different elevation angles. First, the elevation angle was set to $0°$, as the microphone array and the loudspeakers were at the same height. And second, an elevation of $20°$ was set, where the loudspeakers were at a higher position than the array.

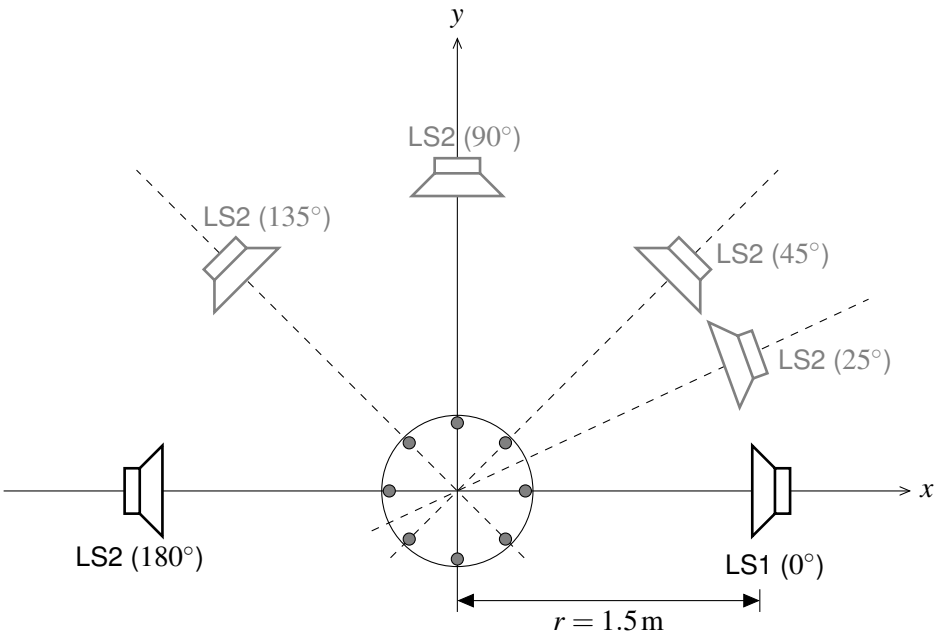


Figure 3.4.: The recording configurations in the anechoic room. Loudspeaker 1 ($LS1$) is located at $0°$ and loudspeaker 2 ($LS2$) was positioned at different angles. Here an angular distance of $180°$ between the loudspeakers is indicated in black. All other configurations for $LS2$ with the angular positions $135°$, $90°$, $45°$ and $25°$ are indicated in gray.

As source signals four different $10\,\mathrm{s}$ long mono files were used, containing only one speaker. The source files are named with the speaker's name and are called *diana.wav*, *gernot.wav*, *martin.wav* and *ricarda.wav*. As can be seen from the file names, there are two male speakers and two female speakers. These files were chosen, because they were also used in [9], so the results can be compared better. Four different combinations of

these sources were played back for each angular distance and elevation. In Table 3.2 these combinations are listed.

Speaker combination	**Source** $LS1$	**Source** $LS2$
diana-gernot	diana	gernot
martin-gernot	martin	gernot
martin-ricarda	martin	ricarda
ricarda-diana	ricarda	diana

Table 3.2.: Speaker combinations, used for the recordings for the BSS evaluation. Shows which sources have been played back through loudspeaker 1 (*LS1*) and loudspeaker 2 (*LS2*).

The same recordings as in the anechoic room were also made in an echoic office room. The only difference is, that the distance between the loudspeakers and the array was $r = 1.0\,\mathrm{m}$, because the room was very small and a distance of $1.0\,\mathrm{m}$ is in this case more realistic for a conference. The dimensions of the office room were about $4.6\,\mathrm{m} \times 3.4\,\mathrm{m} \times 3.10\,\mathrm{m}$. In Figure 3.5 the recording configurations in the echoic office room are shown. The array had been placed on a round table. The position of the table was on a randomly chosen position in the room, not in the middle. The exact position can be seen in Figure 3.5. The top of the array was at a height of $1,33\,\mathrm{m}$. For the recording with 0° elevation the loudspeakers were also positioned at a height of $1.33\,\mathrm{m}$. For 20° elevation the height of the loudspeakers was $1.67\,\mathrm{m}$.

It has to be mentioned, that the hardware, that was used for the recordings, caused some latency between playing back the signals and capturing the microphone signals. Of course, due to the signal propagation time there is also some delay, which is normal for acoustic signals and also between the microphones there are some delays, but this propagation delay is no problem, because the evaluation algorithm can handle about 400 samples of propagation delay. But the delay caused by the recording hardware is much more than 400 samples. In order to determine this delay, a sound file was played back and recorded through a loopback from the output to the input of the recording device. By calculating the maximum cross-correlation between the played and recorded file, the delay due to the recording hardware could be determined. Also by graphically comparing the two time signals, the same delays as calculated by the cross-correlation could be observed. For the recordings in the anechoic room, the delay was 8353 samples. In the echoic office room, the delay was 6305 samples, since a different buffer size was used for recording. So, when evaluating the separated signals, this delay has to be compensated.

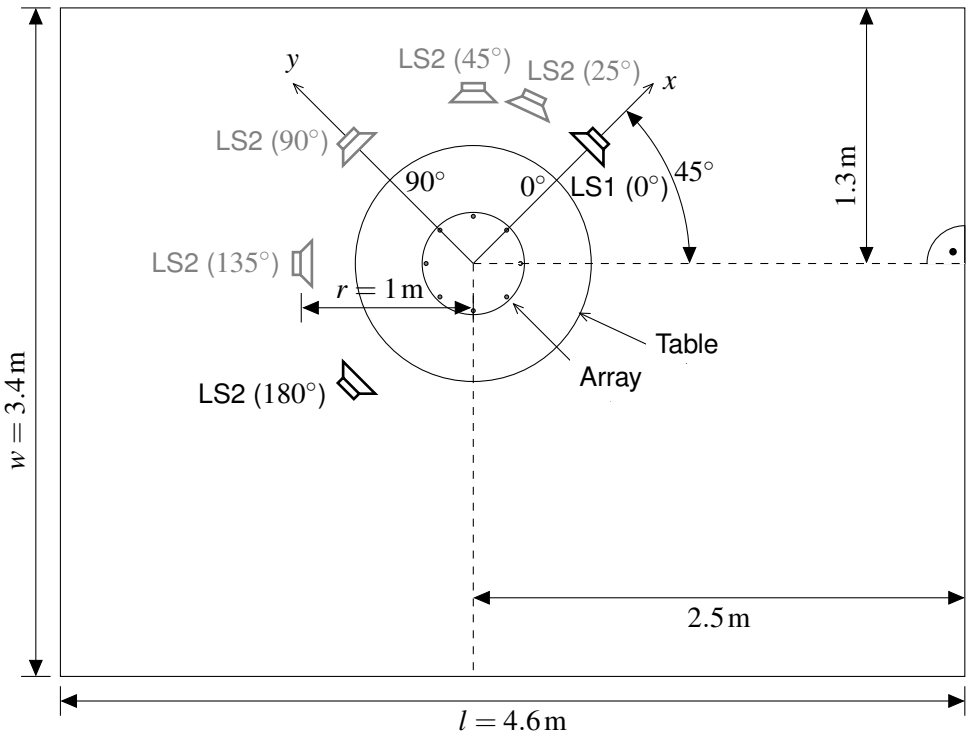


Figure 3.5.: The recording configurations in the echoic office room. The configurations are the same as in the anechoic room. The array has been rotated by 45°, so the coordinate system is also rotated by 45°.

3.5. Graphical User Interface

It is very inconvenient, always having to type in long commands with all necessary parameters into the command window, if you want to perform source separation and then listen to the different channels of the separated signals, I have built a graphical user interface in Matlab. This simplifies the usage of the separation algorithms vastly. This is also advantageous to people who do not know all the different functions but also want to listen how the separated signals sound.

This graphical user interface is very important, because we have to verify if the separation performance measures, such as SDR, SIR and SAR, really tell us the truth about the quality of the separation. Obtaining good evaluation results does not automatically mean that humans perceive the separation quality in the same way. So this graphical user interface is a useful tool to check for oneself, if the separation went well.

All sound files that were used for the evaluation of the source separation can be selected in a list just by one mouse click and all important parameters for the source separation can be selected by drop-down menus.

It is even possible to select the microphones, which are used for the source separation, by check boxes that are arranged in a circle. So it is easier for the user to select the desired microphones.

Depending on the number of selected microphones, play buttons appear that allow us to listen to the separated signals. When a play button is pressed, first the source separation for the selected configuration is performed and afterwards the separated signal is played. The separation of the signals can take some time, so the separated signals can't be heard immediately. Because a precalculation of all possible configurations would take too much time and also require a lot of storage, only the separated signals for configurations that have already been calculated are stored. So if a configuration is selected that has already been separated, the user can immediately listen to the separated sources.
Of course it is also possible to listen to the unseparated signals.

Figure 3.6 shows a screenshot of the graphical user interface.

3.6. Evaluation Results for the Anechoic Room Recordings

For the evaluation, the function *bss_eval_sources.m* of the *BSS Eval toolbox Version 3.0* [22] is applied to the separated signals, which have been obtained by the basic IVA implementation (*iva_general.m*). This function calculates the SDR, SIR and SAR values, as described in Chapter 2.3.1. As reference signals for the calculations the original sound files were used. For each recorded file the SDR, SIR and SAR values for all 247 possible microphone combinations, as listed in Table A.1, were calculated. If there are at each recording two active speakers, we get two SDR, SIR and SAR values per microphone combination, one value for speaker 1 and one value for speaker 2. The source separation has been performed with the parameters, shown in Table 3.3.

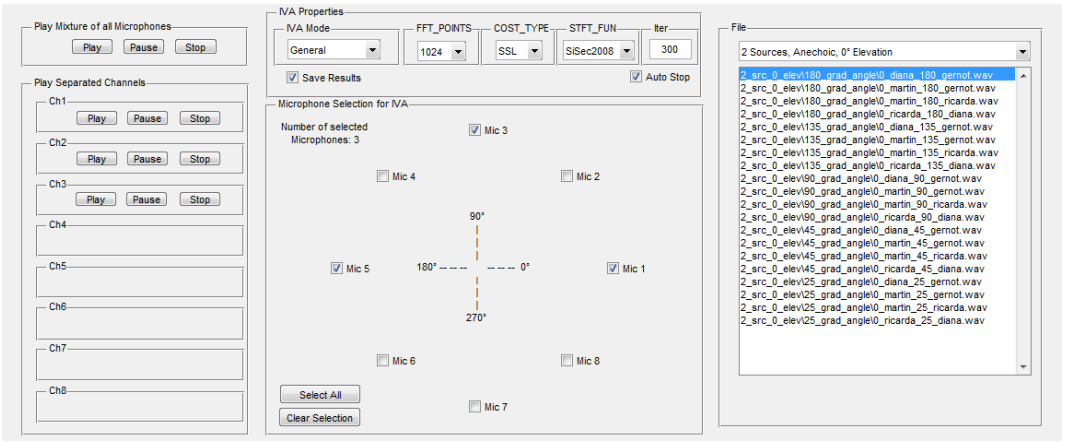


Figure 3.6.: The graphical user interface for performing IVA

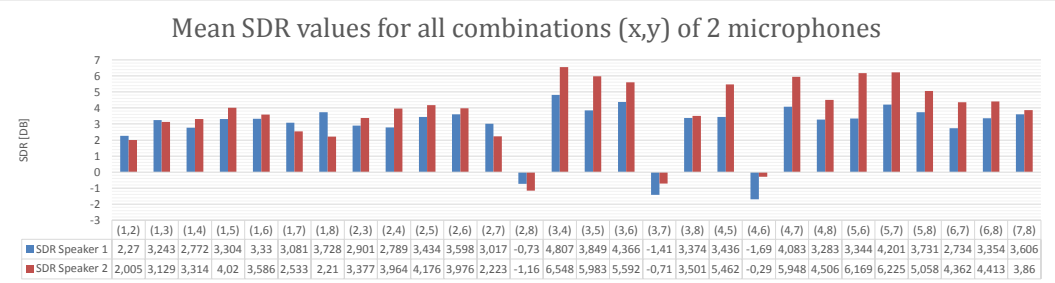
STFT window size	1024 samples
STFT function	SiSec2008
Cost function	SSL
Iterations	300
Auto stop	enabled
Sampling frequency	48 kHz

Table 3.3.: Separation parameters, used for IVA

3.6.1. Evaluation Results for IVA with Two Microphones

First, let us take a look at the case that there are two active speakers. The theory tells us, that for this case 2 microphones are sufficient to separate the two source signals by IVA. But is this also true in reality or does the separation with more than two microphones yield better results? So we will look first, what results we can achieve with two microphones.

In Figure 3.7 the separation results for the case that the angular distance between the speakers is $180°$ are shown. This means that the speakers sit opposite to each other. For each microphone combination, the mean of eight SDR, SIR and SAR values has been calculated, since we have four speaker combinations at an elevation of $0°$ and four combinations at an evaluation of $20°$.



	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)	(3,4)	(3,5)	(3,6)	(3,7)	(3,8)	(4,5)	(4,6)	(4,7)	(4,8)	(5,6)	(5,7)	(5,8)	(6,7)	(6,8)	(7,8)
SDR Speaker 1	2,27	3,243	2,772	3,304	3,33	3,081	3,728	2,901	2,789	3,434	3,598	3,017	-0,73	4,807	3,849	4,366	-1,41	3,374	3,436	-1,69	4,083	3,283	3,344	4,201	3,731	2,734	3,354	3,606
SDR Speaker 2	2,005	3,129	3,314	4,02	3,586	2,533	2,21	3,377	3,964	4,176	3,976	2,223	-1,16	6,548	5,983	5,592	-0,71	3,501	5,462	-0,29	5,948	4,506	6,169	6,225	5,058	4,362	4,413	3,86

Figure 3.7.: The mean SDR values for all combinations (x, y) of two microphones for two active speakers with $180°$ speaker distance. (Anechoic Room)

The figure shows, that the SDR values strongly vary depending on the selected microphones. With some combinations we can reach good results, for example with the combination $(3,4)$ or $(3,6)$. But with the combinations $(2,8)$, $(3,7)$ and $(4,6)$ the separation results are poor. This observation is very interesting, because it shows how important the positions of the microphones in relation to the speakers are when using only two microphones for the separation. In Figure 3.8 the best and the worst combinations for $180°$ speaker distance are visualized. On the left, the combinations that are not good for separation are indicated by red lines. Combinations that yield good results are indicated by green lines on the right. The black arrows indicate, from which direction the speech signals are arriving. So for the case of $180°$ speaker distance we can see, that combinations where both microphones have the same distance to the sources yield bad separation results.

The separation results of the remaining speaker distances $135°$, $90°$, $45°$ and $25°$ have also been visualized in the Figures 3.9, 3.10, 3.11 and 3.12. Comparing all these results, we can see also here, that there are always some combinations that yield very poor results and some combinations that yield very good results.

The complete evaluation results for the anechoic room, containing all SDR, SIR and SAR values for all evaluated speaker angles can be found in A.4.

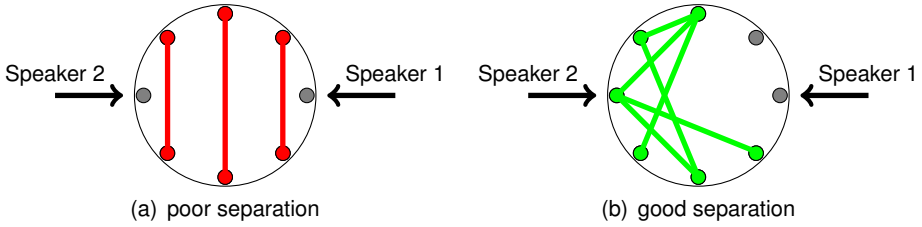


Figure 3.8.: Visualization of the separation results for two microphones and two active speakers with distance $180°$. On the left, the combinations yielding the worst results are indicated by red lines. On the right, the combinations yielding the best separation results are indicated by green lines.

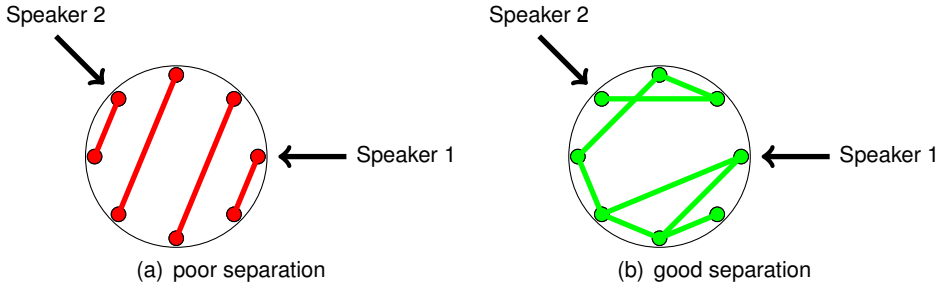


Figure 3.9.: Visualization of the separation results for two microphones and two active speakers with distance $135°$

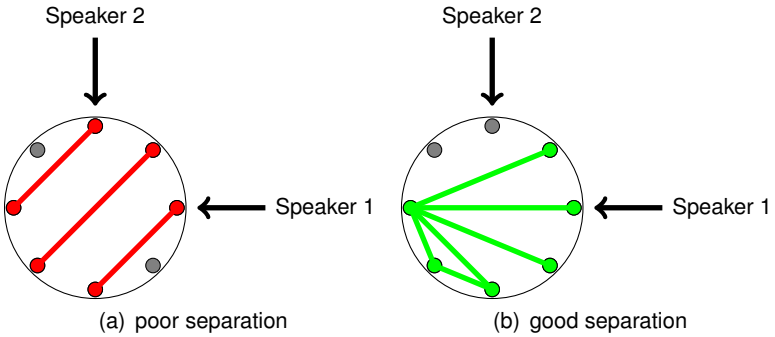


Figure 3.10.: Visualization of the separation results for two microphones and two active speakers with distance $90°$

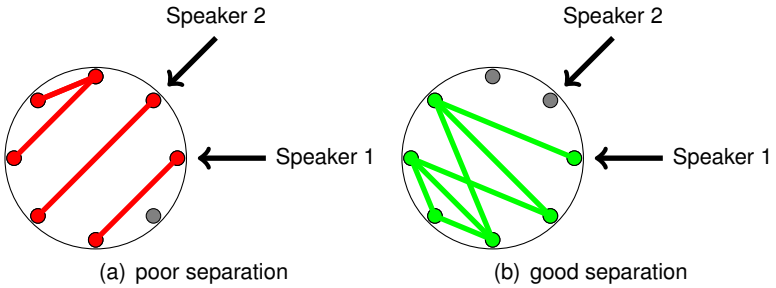


Figure 3.11.: Visualization of the separation results for two microphones and two active speakers with distance 45°

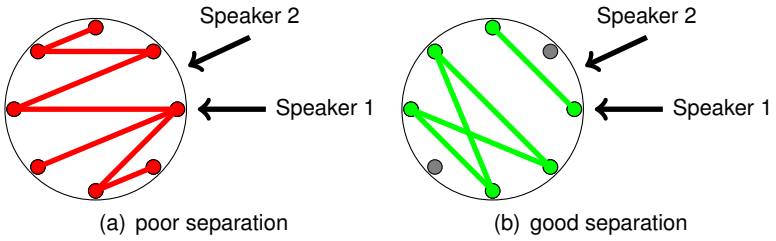


Figure 3.12.: Visualization of the separation results for two microphones and two active speakers with distance 25°

Having seen all the results above, the question arises, why there are always some combinations that achieve very poor separation results in comparison to all other results. In Figure 3.13 the geometry of a configuration that yields poor separation results is depicted. The observation of the evaluation results from above was, that the separation yields bad results, when both microphones are located on the black solid line, which is the perpendicular bisector of the locations of source 1 (s_1) and source 2 (s_2). In this case the distance to both sources at each microphone is the same. Also when the microphones are located on lines parallel to the perpendicular bisector with a small distance, the separation results are also not good. If we assume that there are perfect conditions for sound propagation and

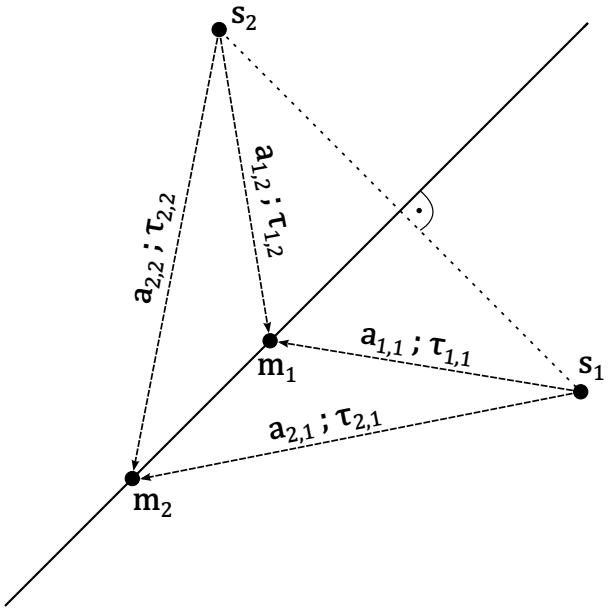


Figure 3.13.: Geometrical interpretation of the configurations that yield poor separation results. There are shown two sources, s_1 and s_2, and two microphones, m_1 and m_2. When m_1 and m_2 are located at the black solid line or at a line parallel to it, the separation yields bad results.

there are no reflections, which means that the signals arriving at the microphones are only influenced by an attenuation factor a and a time delay τ, the microphone signals $m_1(t)$ and $m_2(t)$ can be calculated as

$$\begin{aligned} m_1(t) &= a_{1,1} \cdot s_1(t-\tau_{1,1}) + a_{1,2} \cdot s_2(t-\tau_{1,2}) \\ m_2(t) &= a_{2,1} \cdot s_1(t-\tau_{2,1}) + a_{2,2} \cdot s_2(t-\tau_{2,2}), \end{aligned} \tag{3.1}$$

where $s_1(t)$ and $s_2(t)$ are the source signals. If we assume that a and τ are only influenced by the distance, the attenuation factors and the time delays at each microphone from both sources become the same:

$$\begin{aligned} \tau_{1,1} = \tau_{1,2} \quad \& \quad \tau_{2,1} = \tau_{2,2} \\ a_{1,1} = a_{1,2} \quad \& \quad a_{2,1} = a_{2,2}. \end{aligned} \tag{3.2}$$

So after some transformations we see, that the signal

$$m_1(t) = a \cdot m_2(t - \tau), \tag{3.3}$$

arriving at microphone 1 is the same as the signal, arriving at microphone 2, attenuated by a factor a and delayed by a value of τ.

Such a mixture is the worst case for the source separation algorithm, because the second microphone signal does not contain any further information about the source signals.

Even though under real conditions the signals, arriving at both microphones are not exactly the same, this constellation is not advantageous, because the signals are still very similar to each other.

Since the separation performance, using two microphones, strongly depends on the geometry of the sources and the microphones, the positions of the sources have to be known in order to achieve a good source separation. But, because here we have a BSS scenario, we cannot make any assumptions about the positions of the sources. So, for this scenario, two microphones are not enough to yield good separation results for all possible positions.

3.6.2. Evaluation Results for IVA with More Than Two Microphones

For all microphone combinations with more than two microphones the SDR, SIR and SAR values have also been calculated for all speaker angles. Diagrams like in Figure 3.7 and A.4 have also been created for all cases. But showing all these diagrams would be to spacious and some of them are also too big to show them on one page. For example in the case of four microphones there are 70 values. For those, who are interested in these diagrams, I refer to the attached DVD. This DVD contains all separation results, stored and visualized in Excel files.

In the following, only the best combinations are shown. In order to find the best microphone combinations for more than two microphones, the mean of the SDR values over all evaluated source position has been calculated in order to find combinations, that yield good separation results for all possible positions.

In Figure 3.14 the best microphone combinations for three microphones are shown. The separation results for combinations with three microphones were in general very good and did not vary as much as in the case of two microphones. Nevertheless combinations, that are shown in this figure, achieved the best separation results. So, when there are two speakers active and we select one of the combinations, shown in Figure 3.14, there can be achieved good separation results for all possible positions of the speakers.

The microphone combination that achieve the best SDR values for two active sources, when four microphones are used, are shown in Figure 3.15.

For every number of microphones, one combination that yields good separation results over all position, has been selected and the results have been compared. Figure 3.16 shows the SDR values for these combinations in the anechoic room, depending on the speaker angles.

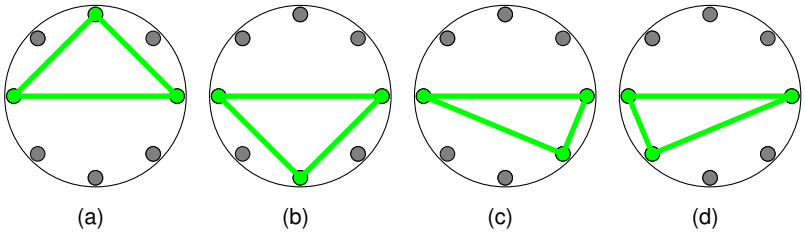


Figure 3.14.: Microphone combinations with three microphones, that achieve the best separation results, averaged over all possible source positions. (2 active speakers)

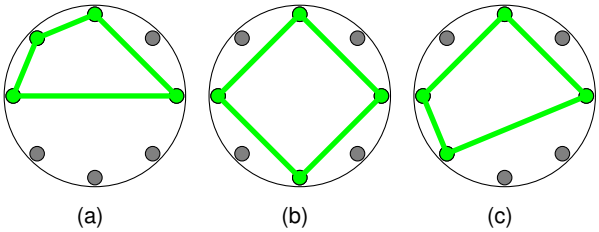
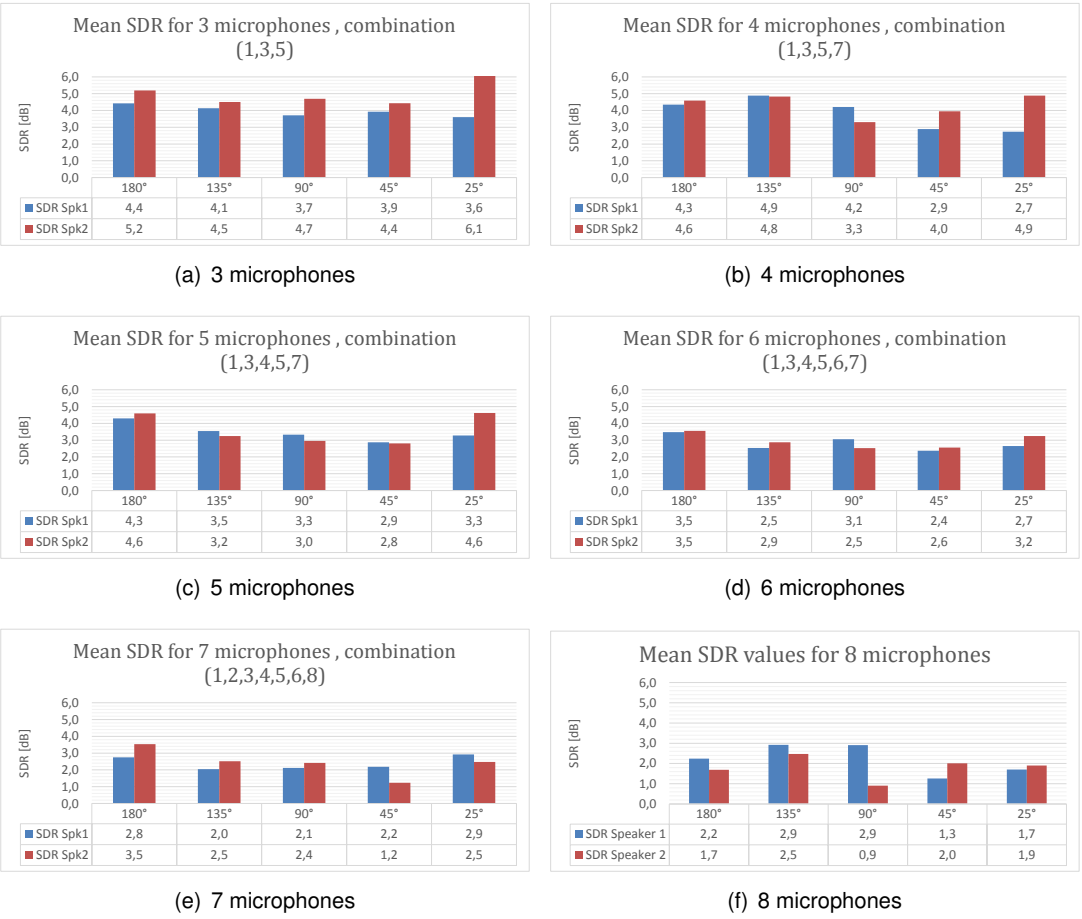


Figure 3.15.: Microphone combinations with four microphones, that achieve the best separation results, averaged over all possible source positions. (2 active speakers)

As we can see, the best and most stable separation results over all positions can be achieved by using three microphones, if there are two active speakers. With increasing number of microphones, the SDR values are decreasing. The worst SDR values are achieved, when all eight microphones are used for the separation. Also by listening to the separated signals, we can confirm these results. So, just taking all eight microphones for the separation is no good idea. The reason for that could be, that IVA tries to obtain as many independent signals as there are input signals. When there are only two sources and eight microphone signals are used for the separation, IVA tries to obtain eight independent signals, although there are only two sources.

Hence, when using IVA for the separation of two sources, selecting three microphones is the best choice for anechoic recordings, in order to obtain a good separation. Also two microphones can be used, but when using two microphones, the positions of the sources should be known, because, as shown above, the separation performance can vary extremely, depending on the selected microphone pair.



(a) 3 microphones

(b) 4 microphones

(c) 5 microphones

(d) 6 microphones

(e) 7 microphones

(f) 8 microphones

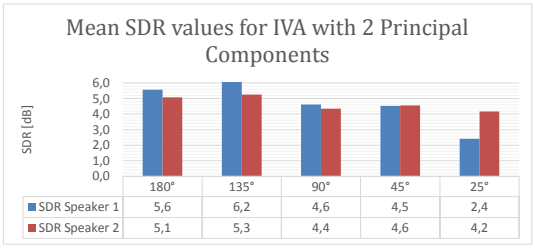
Figure 3.16.: The mean SDR values for different numbers of microphones, depending on the speaker angle, for the anechoic recordings. For each number of microphones the combination yielding the best mean SDR values, was chosen.

3.6.3. Evaluation Results for IVA with PCA Subspace Method

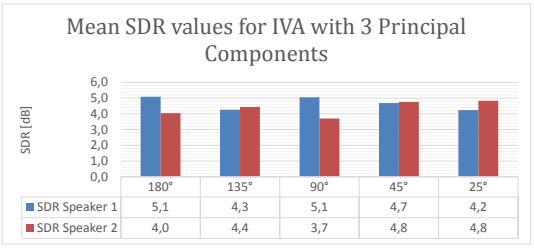
The subspace method promises to solve the overdetermined BSS problem. The advantage of this method is, that we can use all eight microphones. We only have to choose, with how many principal components we want to perform the separation. Another advantage of the subspace method is, that some noise is removed as well, since we only keep the signal subspace.

The SDR, SIR and SAR values have been calculated for the subspace method. In Figure 3.17 the mean SDR values in the anechoic room are shown, depending on the speaker angle. To find out, how many principal components we really need for the separation of two sources, the evaluation has been done for two, three and four principal components.

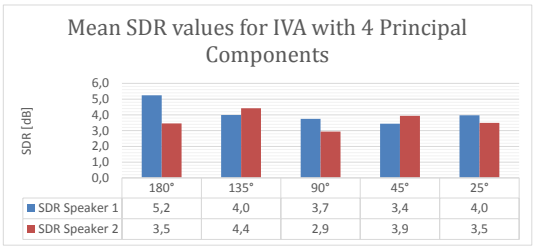
As one can see, the separation with two or three principal components achieves the best results for anechoic room recordings. The separation results for three principal components are more stable, also for small angles. But for large speaker angles, the separation with two principal components show better results.



(a) 2 principal components



(b) 3 principal components



(c) 4 principal components

Figure 3.17.: The mean SDR values for different numbers of principal components, using the PCA subspace method depending on the speaker angle, for the anechoic recordings.

3.7. Evaluation Results for the Echoic Office Room Recordings

After having evaluated the different separation methods for the anechoic room set-up, the same has to be done for the office room recordings, in order to see, how the separation performs in an echoic environment. Do the separation methods, that show a good performance in anechoic rooms also perform well in echoic rooms, or do they behave completely different?

In the following, there will be first evaluated the basic IVA implementation and then IVA with the subspace method.

3.7.1. Evaluation Results for the Basic IVA Implementation

For the anechoic recordings, the basic IVA implementation showed the best results, if three microphones are used. Using all eight microphones produce worse separation results.

Now, the question is, how IVA behaves for different numbers of microphones in the office room, when there are reflections. For this reason, the separation performance of the basic IVA implementation has been evaluated in the office room.

In Figure 3.18, the mean SDR values, obtained in the office room, for different numbers of microphones with the basic IVA implementation are shown. Here, the same microphone combinations as used in the anechoic room are shown, to be able to make a better comparison between the two environments. The complete evaluation results for all possible microphone combinations and speaker angles are available on the attached DVD.

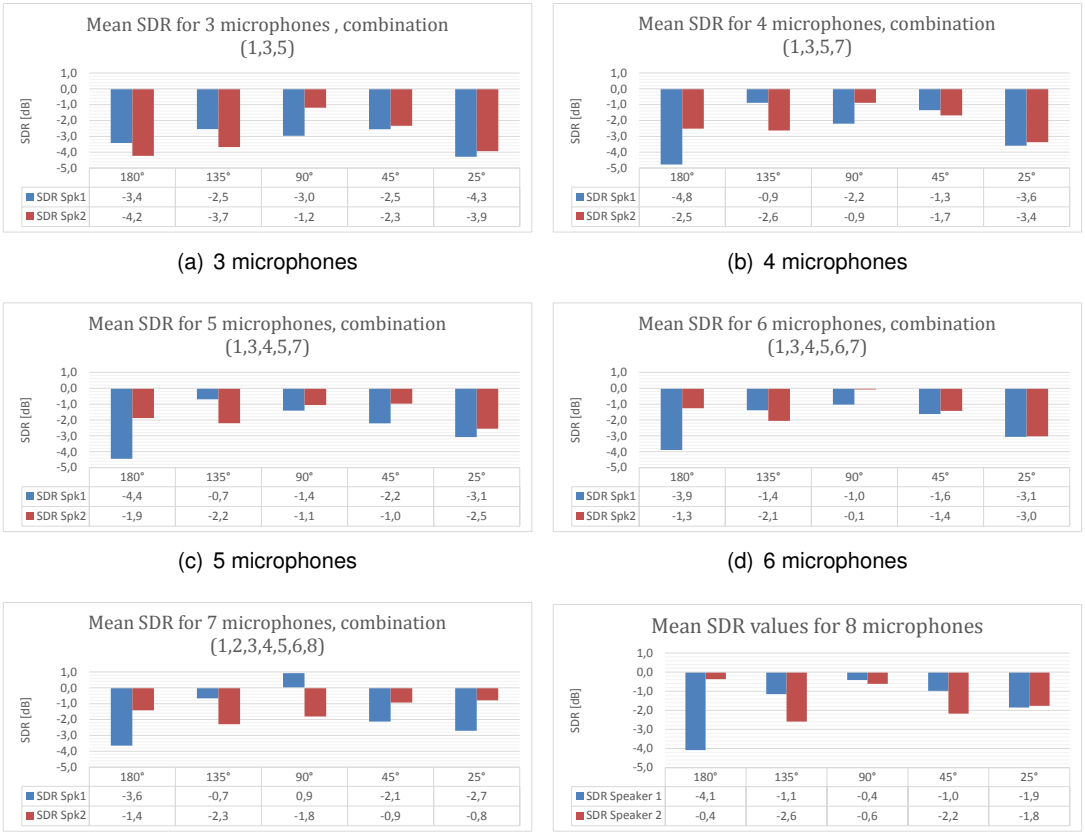
As one can see, the results in the office room are completely different from the results in the anechoic room. The SDR values are much worse and with increasing number of microphones the SDR values become better. The best results could be achieved with seven or eight microphones. In the anechoic room it was the other way around and the SDR values became worse with increasing number of microphones.

So, using more microphones yields better separation results for the office room set-up. It seems, that due to reflections, more independent components than source signals are present. Otherwise, IVA would not yield better separation results with more microphones, as we could see from the separation results in the anechoic room.

Another interesting observation is, that for an speaker angle of 90° the separation results are better than for all other angles. For the angles 180° and 25° the separation results are worst.

3.7.2. Evaluation Results for IVA with PCA Subspace Method

For the recordings, made in the anechoic room, the subspace method has achieved very good separation results, when using two or three principal components. Now, the question is, if the subspace method also yields good separation results in the office room. For that reason, the subspace method has also been evaluated in the office room for two, three and four principal components.



(a) 3 microphones

(b) 4 microphones

(c) 5 microphones

(d) 6 microphones

(e) 7 microphones

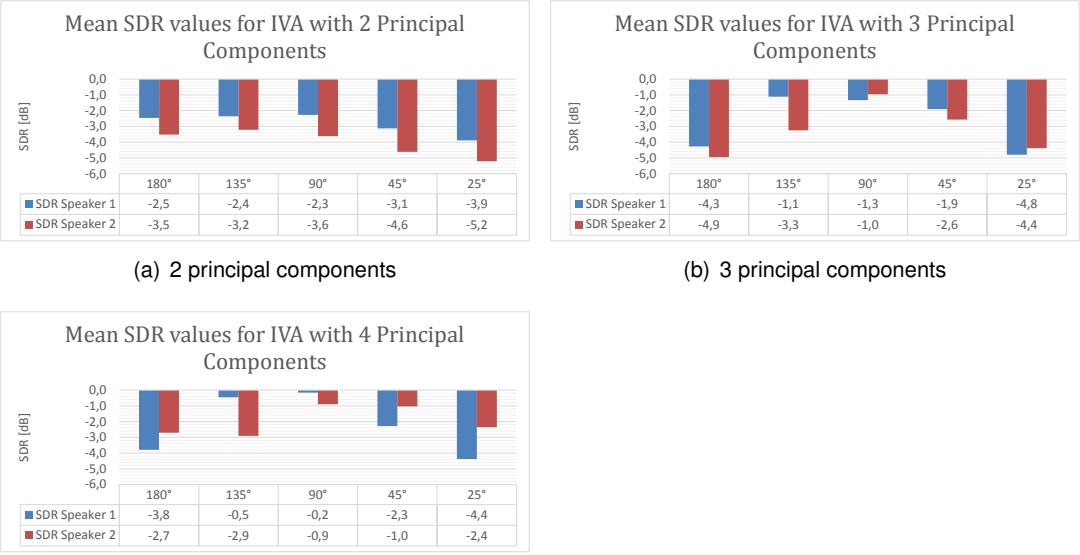
(f) 8 microphones

Figure 3.18.: The mean SDR values for different numbers of microphones, depending on the speaker angle for the office room recordings. For each number of microphones, the combination, yielding the best mean SDR values, was chosen.

In Figure 3.19 the mean SDR values for the office room recordings, depending on the speaker angles, are shown. It becomes obvious, that the subspace method in this case does not yield better separation results. The worst results have been achieved by using two principal components.

So, also here, using more principal components leads to better separation results.

It can again be recognized, that for a speaker angle of $90°$, the separation results are much better than for $180°$ and $25°$.



(a) 2 principal components

(b) 3 principal components

(c) 4 principal components

Figure 3.19.: The mean SDR values for different numbers of principal components, using the PCA subspace method, depending on the speaker angle for the office room recordings.

Now, the question arises, why with the subspace method no better separation could be achieved on the office room recordings. In [3], this problem is discussed. Due to room reflections it is very difficult to estimate a separation matrix, especially for rooms with a high reverberation time. As a solution it is supposed to choose a short window length for the STFT, so that the time interval between the direct sound and the reflection exceeds the window length. It is assumed, that then the reflections behave like "incoherent additive noise", since speech is nonstationary. So, the subspace method can reduce some reflections, if the window length is short.

To see, if this method really works, different window lengths for the STFT have been tested; also longer window lengths, in order to see, how the window length affects the separation results. The standard window length, used for all other evaluation was 1024 samples, which corresponds to $21\,\mathrm{ms}$ at a sampling frequency of $48\,\mathrm{kHz}$. To see the influence of the window length to the separation results, window length from 128 samples to

8192 samples have been evaluated. Figure 3.20 shows, how the subspace method performs for different STFT windows lengths. The mean of the SDR values over all positions has been calculated, to see the overall performance. Here, the subspace method with four principal components is shown, because the best results have been yielded with it. As we can see, the SDR values become better, the shorter the window length is. For a window length of 128 samples, which corresponds to 2.7 ms, the SDR value is highest. Also by listening to the separated signals, this results could be confirmed. Even for the subspace method with two and three principal components, similar results have been observed.

But to make reliable statements, how the window length affects the separation results in general, more evaluation and further work would be needed, extending this point of view. Since, this is not in the main scope of this thesis, for now let us just notice, that the window length can also influence the separation performance, but here, mostly a window length of 1024 samples will be used, so that all results are consistent.

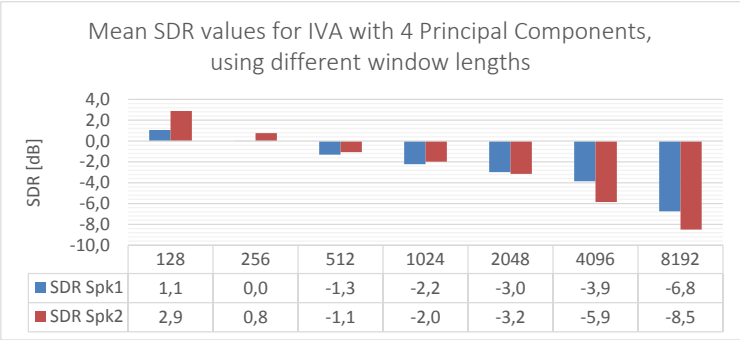


Figure 3.20.: Comparison of different STFT window lengths, using the subspace method with 4 principal components and office room recordings.

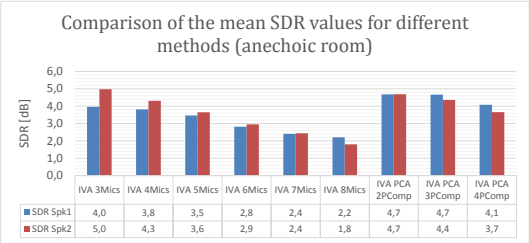
3.8. Summary of the Evaluation Results

After a lot of evaluation, all results for both environments are summarized here. In Figure 3.21, the mean of the SDR values over all positions has been calculated, in order to be able to compare the overall performance of the different methods. On the left, the results for the anechoic recordings are shown, and on the right, the results for the office recordings. So, we can also directly compare the evaluation results of both environments.

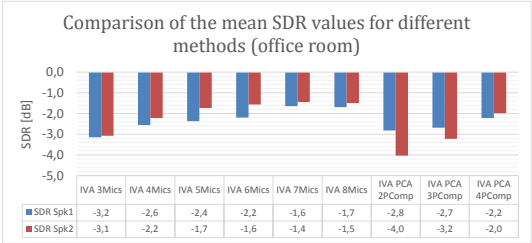
There can be made the following statements:

- For the anechoic recordings, the separation of two sound sources performs best with a small number of microphones or principal components. Using all eight microphones[1], yields the worst results. The best results could be achieved with the subspace method, using two principal components.

[1] IVA with 8 microphones = IVA subspace method with 8 principal components



(a) anechoic room



(b) office room

Figure 3.21.: Overview of all evaluated separation methods for the different environments with two sources.

- For recordings, made in the office room, the separation performs best with more microphones or principal components. The best results can be achieved, using seven or all eight microphones.
- Acoustic reflections reduce the separation performance strongly.

In order to achieve good separation results, different separation methods should be used, depending on the environment. When there are a lot of reflections, it is better to choose more microphones or principal components. In anechoic rooms, less microphones or principal components are better. Thus, we have to know, how reverberant the environment of the conference room is. To yield optimal results, it would be beneficial to find a connection between reverberation and the best configuration for IVA.

As mentioned above, also the window length of the STFT, when applying the subspace method, is an important parameter, which can influence the separation performance a lot.

So, there are a lot of possibilities for the selection of the parameters for the source separation, that should be chosen differently, depending on the environment. To find the optimal parameters for every possible environment, more research is still needed.

4. Joint Source Separation and Speaker Recognition

This chapter deals with the connection of BSS with a speaker recognition system. Most speaker recognition systems suffer from overlapping speech and can only detect one speaker at one time. So it would be great, if the performance of speaker recognition systems could be improved by applying BSS prior to the speaker recognition.

4.1. The Speaker Recognition System

The speaker recognition system, used in this thesis has been developed by Christoph Kozielski [14] during his diploma thesis at the Institute for Data Processing. This speaker recognition works as described in Chapter 2.2 by building GMMs on MFCCs. This recognition can be used online, but the signals have to be downsampled to $16\,\mathrm{kHz}$. For offline recognition downsampling is not required, but we will first also sample down the signals in order to be able to compare the separation results to other work, where this speaker recognition was used. This speaker recognition is a *closed set* recognition, which means, that all possible speakers have to be known to the system. Hence, for every speaker a model has to be trained, before the speaker recognition can be started.

4.1.1. Model Training

For the model training we need speech signals that only contain the speaker who has to be trained. In the scenario of a real conference the data for the model training could be realized by an introduction round at the beginning of the conference, where every speaker has to say something for a certain time. So we can make sure, that only one speaker is active. This recordings can then be used to train the speaker models. Figure 4.1 shows the basic steps of the implementation of the model training, which have already been explained in Chapter 2.2. Prior to the feature extraction, a voice activity detection (VAD) is applied to the signals, that discards segments, containing no speech.

The first question was, how long the data for the model training should be. To find out, which training length is the best, different models were trained with the evaluation data (the used evaluation data set will be introduced in Chapter 4.2). Then with the calculated models a speaker recognition was performed on data, containing only one speaker. The tested training lengths were $30\,\mathrm{s}$, $60\,\mathrm{s}$, $90\,\mathrm{s}$ and $120\,\mathrm{s}$.

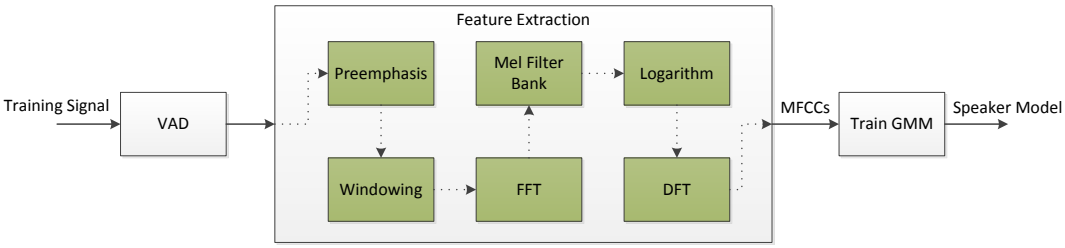
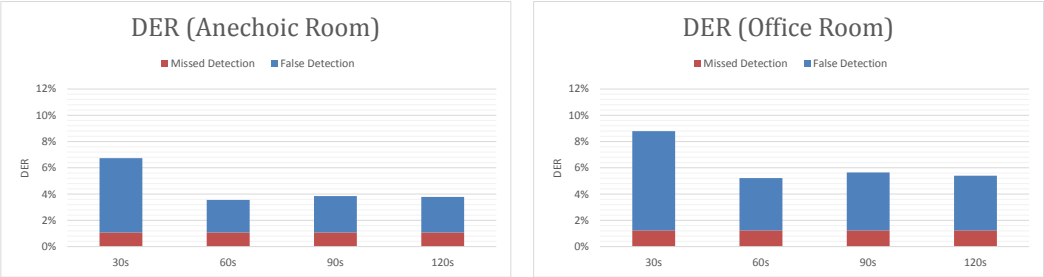


Figure 4.1.: Model Training



(a) Anechoic Room

(b) Office Room

Figure 4.2.: DER for one active speaker, for different training lengths ($30\,\mathrm{s}$, $60\,\mathrm{s}$, $90\,\mathrm{s}$,$120\,\mathrm{s}$) for a) an anechoic room, and b) an office room

Figure 4.2 shows the achieved values for the diarization error rate (DER) for the different models, depending on the training length, for one active speaker, as defined in Equation (2.61). On the left the DER for anechoic room recordings are shown and on the right the values for office room recordings are shown. The figure clearly shows that a training length of $60\,\mathrm{s}$ is the best choice, since it has the smallest error rate in both environments. So for the rest of this thesis this length is used for model training.

In Table 4.1 all important parameters, used for the model training are listed.

Number of features	39 (12 MFCCs + Spectral energy; 1^{st} & 2^{nd} order delta regression coefficients)
Number of Gaussian mixture components	128
STFT window size	1024 samples
Sampling frequency	$16\,\mathrm{kHz}$

Table 4.1.: Model training and recognition parameters

4.1.2. Speaker Recognition

When all models are trained, the speaker recognition can begin. The structure of the recognition system can be seen in Figure 4.3. The feature extraction works in the same way as in the model training, which was already shown in Figure 4.1. After the feature extraction, the log-likelihood to all available speaker models is calculated and the most likely model is detected as speaker.

For the speaker recognition the same parameters are used as for the model training, as listed in Table 4.1.

When evaluating the speaker recognition, it is important to use different data for the model training and the speaker recognition. Applying a speaker recognition on the training data can falsify the recognition results.

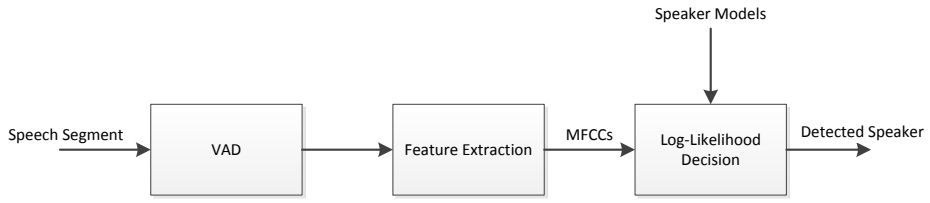


Figure 4.3.: Speaker Detection

4.1.3. Application of the Speaker Recognition to the Separated Signals

When connecting the speaker recognition with BSS, there are many possibilities in the selection of the separation method when training the model and when applying speaker recognition. Since in Chapter 3 several methods to perform overdetermined IVA were utilized, the question arises, which method should be used for the separation, prior to the speaker recognition. Do the separation methods that showed a good separation performance also yield good recognition results?

We also have to know, how the models should be trained to yield the best recognition results. Should the models be trained on the recordings without separation or should IVA or a PCA be applied to the training data, before the model training is carried out?

To find out, which methods works best, we will have to evaluate several possibilities and compare the recognition results. In Table 4.2 all methods that have been evaluated are listed. Since it is confusing, explaining always how the separation in the different cases was performed, some abbreviations are defined, which represent the selected separation method. In the left column you can see the abbreviation and on the right column there are detailed explanations of the used separation method.

For the separation method *IVA 3Mics* always the microphone combination $(1,3,5)$ was used, because this combination showed good separation results for all speaker angles (see Chapter 3.6.2).

Abbreviation	Detailed explanation
No Sep	Use the recorded signals without separation, calculate the mean of all 8 channels
PCA 1PComp	Perform PCA, selecting 1 principal component, no separation
IVA 2PComp	Perform IVA PCA subspace method, using 8 microphones, choosing 2 principal components
IVA 3PComp	Perform IVA PCA subspace method, using 8 microphones, choosing 3 principal components
IVA 4PComp	Perform IVA PCA subspace method, using 8 microphones, choosing 4 principal components
IVA 5PComp	Perform IVA PCA subspace method, using 8 microphones, choosing 5 principal components
IVA 6PComp	Perform IVA PCA subspace method, using 8 microphones, choosing 6 principal components
IVA 7PComp	Perform IVA PCA subspace method, using 8 microphones, choosing 7 principal components
IVA 3Mics	Perform basic IVA, using 3 microphones (i.e. IVA PCA subspace method, using 3 microphones, choosing 3 principal components)
IVA 8Mics	Perform basic IVA, using 8 microphones (i.e. IVA PCA subspace method, using 8 microphones, choosing 8 principal components)

Table 4.2.: Separation methods, used prior to the speaker recognition and the model training

4.2. Evaluation Data Set for Speaker Recognition

For the evaluation of the speaker recognition we need a big data set, containing speech of multiple speakers. For the simulation of a conference we need recordings, containing only one active speaker as well as recordings, containing simultaneously active speakers. It is also important not to use the same data for the model training and the speaker recognition. So the idea was to play back speech by a loudspeaker and record it with the microphone array, so that we can simulate a conference. The requirements for these playback files are that they contain only speech of one single speaker with no noise or music in the background. Another requirement is that there is no influence of the recording room that affects these files. So recordings that were made in a professional studio are preferred. Since it is not easy to find recordings, that satisfy all these requirements, audio books

that are available in the internet [6] were chosen as basis for the playback signals. The advantage of these audio books is, that they were sorted by the speakers name, so there are available hours of recordings for several speakers. The audio books have a good quality and are free from noise or music. So they are ideal for the evaluation of the speaker recognition system.

From the audio books recordings from eight different speakers were picked. With these recordings two data sets that should simulate a conference with four participants were built. The first data set is composed of two female and two male speakers. The second data set contains four male speakers. The second data set was used as a reference data set to make sure that the obtained recognition results not only depend on the selected data.

Each data set contains the following parts:

- **Training data:** $2\,\mathrm{min}$ of each speaker
- **1 speaker active:** $10 \times 5\,\mathrm{s}$, $10 \times 10\,\mathrm{s}$, $10 \times 20\,\mathrm{s}$ and $10 \times 30\,\mathrm{s}$ of each speaker
- **2 speakers active:** $30 \times 10\,\mathrm{s}$ with different speaker combinations (data set 2 only $6 \times 10\,\mathrm{s}$)
- **3 speakers active:** $6 \times 10\,\mathrm{s}$ with different speaker combinations (only available in data set 1)
- **4 speakers active:** $3 \times 10\,\mathrm{s}$ with different speaker combinations (only available in data set 1)
- **overlaps at the end/beginning:** $30 \times 13\,\mathrm{s}$, each speaker $7\,\mathrm{s}$, overlap $1\,\mathrm{s}$ (data set 2 only $6 \times 13\,\mathrm{s}$)

Parts like these typically occur during a conference. Because most of the time there is only one speaker talking, more data with only one active speaker is contained in the data set. Later, there can be composed artificial conversations, based on this data set.

The data set has been played and recorded in an anechoic room and in an office room. In Figure 4.4 the recording configuration in the anechoic room is shown. There are 4 loudspeakers ($LS1$ - $LS4$), arranged around the microphone array with a distance of $1\,\mathrm{m}$. The angular distance between the loudspeakers was set to $90°$. The position of loudspeaker 1 is defined as $0°$, so that microphone 1 is the next to loudspeaker 1.

For each data set, each of the 4 speakers is assigned to one, fixed loudspeaker.

The Figures 4.5 and 4.6 show, how the recordings in the office room were realized. The configuration was the same as in the anechoic room, but at a randomly chosen position in the room and the coordinate system was rotated by $45°$. Both, the microphone array and the loudspeakers were positioned in the same height, so that the elevation was $0°$. The height of the loudspeakers and the array was about $1.58\,\mathrm{m}$.

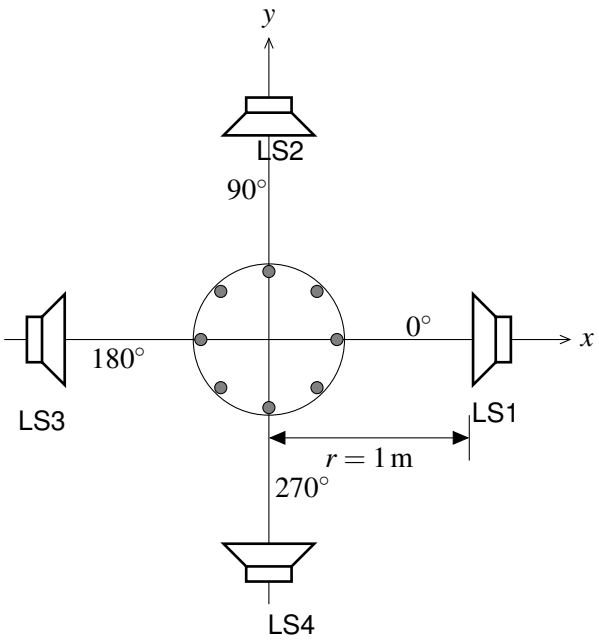


Figure 4.4.: The recording configuration for the evaluation data set for the speaker recognition in the anechoic room

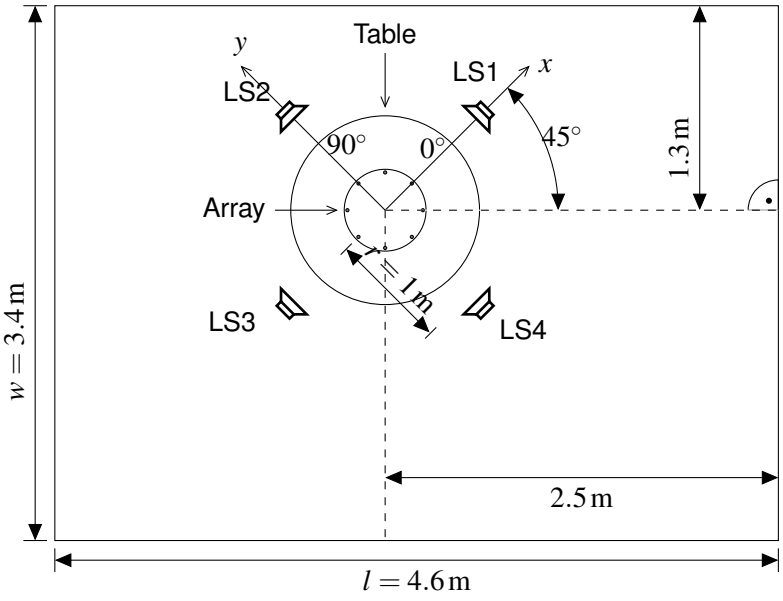


Figure 4.5.: The recording configuration for the evaluation data set for the speaker recognition in the office room

Figure 4.6.: Picture of the recording set-up in the office room

4.3. Evaluation of the Joint Source Separation and Speaker Recognition

As already mentioned above, there are a lot of possibilities how to perform joint source separation and speaker recognition. First, a separation method has to be selected, which is applied prior to the speaker recognition. And second, a separation method for the model training has to be selected. In Table 4.3 some possibilities to perform joint separation and recognition are shown. The methods, marked by an **x** have been evaluated during this thesis.

Used Data \ Used Model	No Sep	PCA 1PComp	IVA 2PComp	IVA 3PComp	IVA 4PComp	IVA 5PComp	IVA 6PComp	IVA 7PComp	IVA 3Mics	IVA 8Mics
No Sep	x	x								
PCA 1PComp	x	x								
IVA 2PComp	x	x	x							
IVA 3PComp	x	x		x						
IVA 4PComp	x	x			x					
IVA 5PComp	x	x								
IVA 6PComp	x	x								
IVA 7PComp	x	x								
IVA 3Mics	x	x							x	
IVA 8Mics	x	x								x

Table 4.3.: Possibilities for joint source separation and speaker recognition. All combinations, marked by an **x** have been evaluated. Used Data denotes, which separation method is used for speaker recognition. Used Model denotes, which separation method is used for the model training.

First, the evaluation has been performed for only one active speaker, and later, the case of two active speakers has been evaluated.

4.3.1. Evaluation for One Active Speaker

As first step, the speaker recognition has been evaluated for only one active speaker without separation, in order to determine the performance of the speaker recognition system.

Then different separation methods have been tested in order to find out, if applying BSS can improve the performance, when only one speaker is active. Only separation methods, that showed good separation results in the evaluation of the source separation have been selected.

When a source separation is applied prior to the speaker recognition, it is also interesting, which model fits best to the separated signals.

Figure 4.7 shows the diarization error rate (DER), achieved for the anechoic recordings and the office recordings, for one active speaker. Both data sets were evaluated and then the mean of both results was computed. Altogether about $87\,\mathrm{min}$ of speech have been evaluated to obtain the DER for one column in the diagram. So for creating these two diagrams, about $55\,\mathrm{h}$ of speech have been evaluated.

For the speaker recognition, every part was segmented into segments of a length of $1\,\mathrm{s}$ and for each segment a speaker detection was performed. For the case, that a BSS was applied before the recognition, first each part had been separated as a whole and then the separated part had been segmented into segments of $1\,\mathrm{s}$, on which a speaker detection was performed.

To be able to reproduce all these results, for every case one separate script has been written to perform the speaker recognition and all individual separation results have been stored in Excel files. All scripts and Excel files can be found on the attached DVD.

As we can see in Figure 4.7, the DER of the speaker recognition, when applying no source separation, is at 3.6% in the anechoic room and 5.2% in the office room case. This are pretty good results, because an error rate of 3.6% means, that 96.4% of the speech segments are assigned to the right speaker in the anechoic room and in the office room 94.8% are assigned correctly. In both cases the missed detection rate is at about 1%. A speech segment is assigned as missed, when the voice activity detection does not detect it as active speech although a speaker is active. So this means that about 1% of the DER is caused by the VAD. Hence, this 1% cannot be changed by an improved speaker recognition, because it only depends on the VAD. It also might occur that a speech segments falls into a breathing pause, but we can't take this case into account either. The rest of the DER without missed detections[1] is the false detection rate. This is the rate of speech segments where the false speaker has been detected. Altogether we can say, that the recognition performance for single speech without separation is very good in both environments.

Our hope was to improve the recognition performance also for one active speaker by applying IVA. But when looking at the results, it becomes obvious that this is not the case. When a source separation is applied prior to the recognition, the DER rises in every case. The only thing we can learn from these results is, that the models, trained by applying a PCA, or the same separation method as used for the recognized data, yield better results than the models, that were trained without applying a separation. It has to be mentioned, that the recorded speech was very clean and almost no background noise was present. But altogether we can say, that applying BSS to signals, containing only one source, makes no sense as preliminary stage to speaker recognition.

The only useful method is to apply a PCA before the recognition is started. For the anechoic recordings, the recognition performance could be increased in the case of using a PCA with one principal component. The DER could be lowered to 2.6%, but in the office

[1] Note, that the rate of false alarms, as defined in Chapter 2.3.2, doesn't exist in these evaluations, because we only evaluate segments containing active speech

room case the DER almost stayed constant. When applying a PCA as preliminary stage to speaker recognition, it is important to also use a model, where a PCA has been applied prior to the model training.

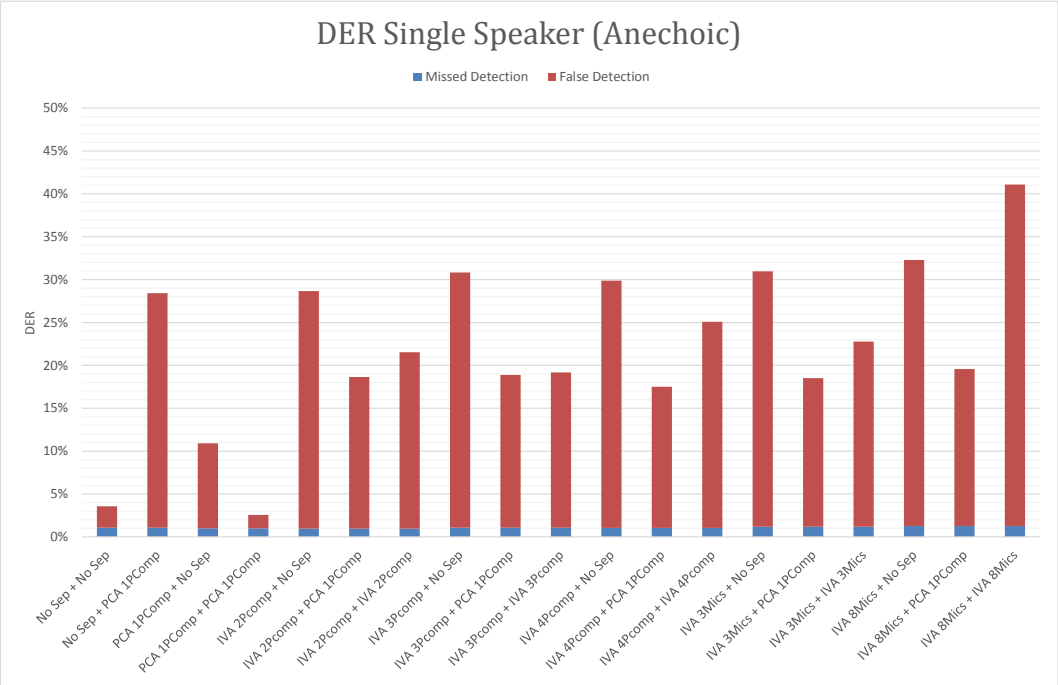
4.3.2. Evaluation for Two Active Speakers

Let us now focus on the case, that there are two speakers talking at the same time. Since we have found out, that for one active speaker, source separation makes no sense, the case of two active speakers is more interesting. Here, source separation might be really useful to improve the performance of the speaker recognition system.

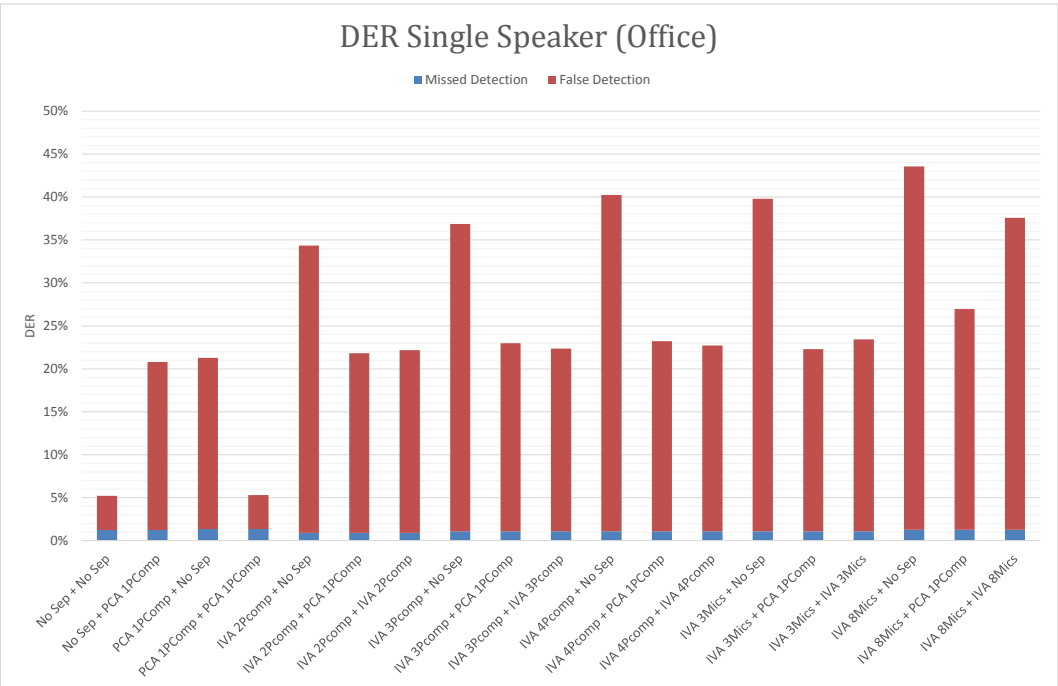
Before we can analyze, how the source separation affects the performance of the speaker recognition, when there are two active speakers, we have to determine, how the speaker recognition behaves without separation. Figure 4.8 shows the recognition results, measured for two active speakers. The first column of each diagram shows the DER in the case that no separation has been performed. Since there were two active speakers, the two most likely speakers have been chosen within the likelihood decision. It can be seen that in the anechoic room as well as in the office room the DER is 100%, which means, that in no case both speakers have been detected correctly, without applying separation to the speech data. When looking at the rate, describing that only one of both speakers has been detected correctly, in about 46% of the segments only one of the two speakers have been detected correctly for the anechoic recordings and for the office recordings about 36%. This means, that the speaker recognition works not completely incorrect, when there are two active speakers, because one of the two speakers can be detected in some cases. But altogether the recognition results are poor for two active speakers without separation, even when trying to detect only one of the two speakers.

The remaining columns in Figure 4.8 show the DER for all cases that were evaluated in combination with IVA. For the calculation of the DER first IVA was applied and then on the first two separated channels a speaker recognition was performed. If in both channels the correct speakers had been detected, the recognition result has been classified as correct. If only one of the two speakers had been detected correctly, the result has been classified as *only one speaker correct*. And for the case, that in both channels the wrong speaker had been detected, the result has been classified as *false detection*. So we can see, how strong the different errors influence the DER. For each column $6\,\mathrm{min}$ of overlapping speech have been evaluated, so that altogether $4.4\,\mathrm{h}$ of speech have been evaluated for creating these two diagrams.

When looking at the results in the anechoic room, we can see that the DER is more than halved, when applying IVA prior to the recognition. The best results can be achieved with the models, trained with *PCA 1PComp*. For the data, that has to be recognized, *IVA 3Mics*, *IVA 3PComp* and *IVA 4PComp* showed the lowest DER. A DER of about 34% could be achieved in the best case. This means, that about 66% of the segments have been detected correctly. This detection rate also seems to be not good, but its a big improvement compared to the recognition rate without separation, where 0% of the segments were

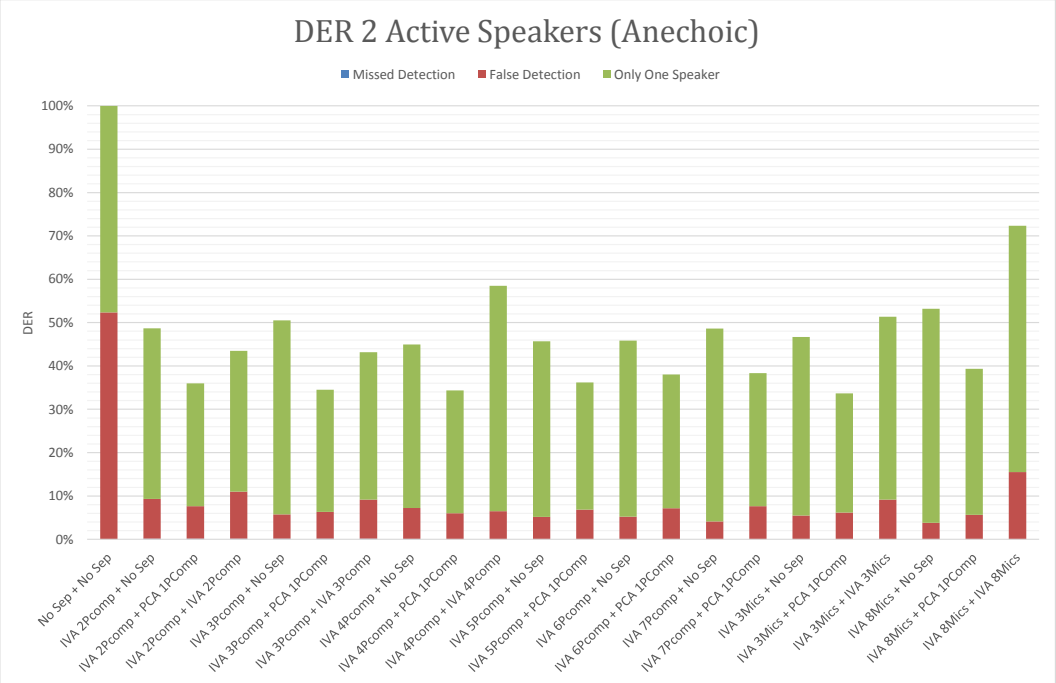


(a) Anechoic room

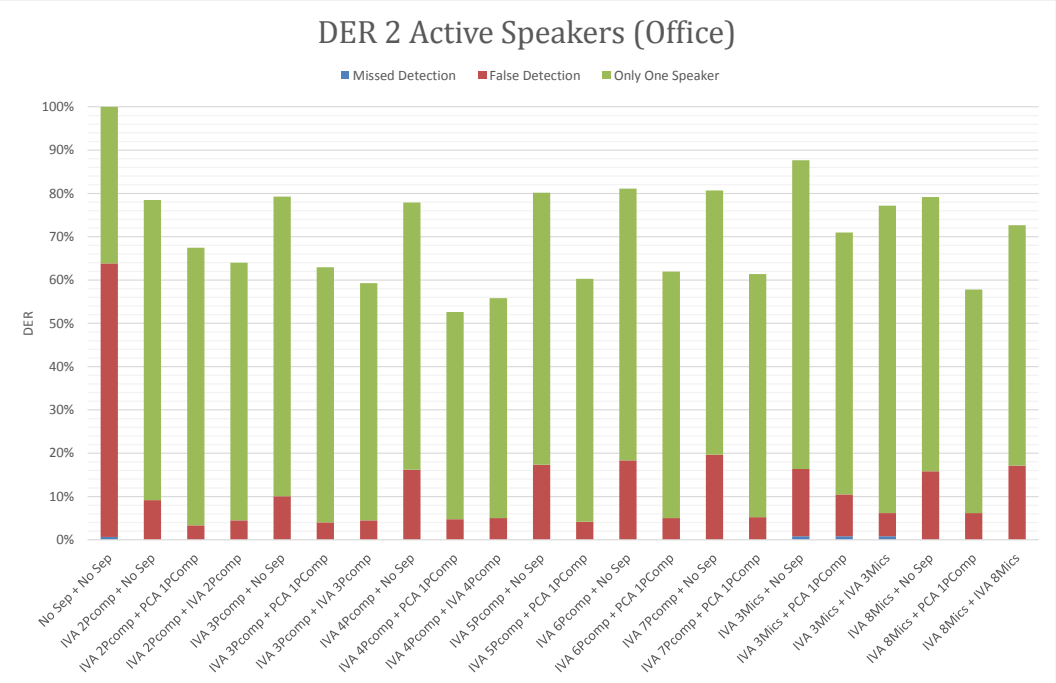


(b) Office room

Figure 4.7.: Diarization error rate for one active speaker in different environments for different methods. The labels of the columns are composed of *"<used data> + <used model>"*.



(a) Anechoic room



(b) Office room

Figure 4.8.: Diarization error rate for two active speakers in different environments for different methods. The labels of the columns are composed of "<used data> + <used model>".

detected correctly. When taking into account, that the rate of segments, where only one speaker has been detected correctly, only 5% of the segments, that have been detected completely false, are remaining. For the case, where no separation has been applied, the rate of completely false assigned segments was about 50%. Thus, a false detection rate of 5% is also a big improvement.

For the office room recordings, the performance has decreased. The best results could be achieved with the method *IVA 4PComp + PCA 1PComp* with a DER of 53%. Compared to the anechoic room, this result is not so good. But the false detection rate could also be improved from about 64% to about 5%. Although the overall DER is high, the performance is still better than without applying separation.

In Table 4.4, the performance of the speaker recognition without separation is compared to the methods with the best performance for each environment, for the case of two active speakers. As we can see, although the DER in the office room is at about 53%, this does not mean, that the speaker recognition is completely wrong in 53% of the segments. It only means, that in 53% not both speakers have been detected correctly. When looking at the rate, that at least one speaker has been detected correctly, the result is much better. For the office room recordings, in 95% of the segments, there has at least one of the two speakers been detected correctly. This is much more than without separation. Without separation, in only 37% of the segments at least one speaker has been detected correctly.

Environment	anechoic		office	
Method	No Sep + No Sep	IVA 3Mics + PCA 1PComp	No Sep + No Sep	IVA 4PComp + PCA 1PComp
Accuracy/Both speakers detected correctly	0%	66.3%	0%	47.4%
DER	100%	33.7%	100%	52.6%
Both speakers detected falsely	52.3%	6.2%	63.2%	4.8%
One speaker detected falsely	47.7%	27.5%	36.2%	47.9%
Missed detection	0%	0%	0.1%	0%
At least one speaker detected correctly	47.7%	93.8%	36.8%	95.2%

Table 4.4.: Comparison of the speaker recognition results for two active speakers without and with separation. For each environment the separation method, yielding the lowest DER has been selected.

So, we can say, that the source separation improves the recognition performance a lot, when two speakers are talking simultaneously, although the DER values are high, yet. For the anechoic recordings, in 66.3% of all segments, both speakers have been detected correctly, and in 93.8%, at least one speaker has been detected correctly. For the office recordings, in 47.4% of all segments, both speakers have been detected correctly, and in 95.2%, at least one speaker has been detected correctly. These results clearly show, that source separation can improve the performance of a speaker recognition.

After having done all evaluations, some additional tests have been carried out, to see, if the speaker recognition can be improved with different parameters. In the previous evaluations, always a sampling rate of $16\,\mathrm{kHz}$ has been used, since we first wanted to investigate the performance of the online speaker recognition system. But here, we do not have to recognize the signals online, so one test was made with a sampling rate of $44,1\,\mathrm{kHz}$ in the office room case. As we can see above, the best DER, achieved for the office recordings, is at about 53%. When performing the speaker recognition and the model training with $44,1\,\mathrm{kHz}$ instead of $16\,\mathrm{kHz}$, this DER could be decreased to 43%, which is an improvement of 10%. This clearly shows, that an increased sampling rate at the speaker recognition as well as at the model training can improve the performance of the speaker recognition a lot. But, to see, how the recognition rate improves by an increased sampling rate, more evaluation has to be done.

In Chapter 3.7.2, it was found out, that the subspace method shows better separation results for shorter STFT window lengths. So, also some tests with shorter window lengths have been performed. But these tests preliminary showed no improvements of the recognition performance, meaning that also here, more tests are needed.

5. Concluding Remarks

In this chapter, the most important facts, that have been achieved during this theses, are summarized and some proposals for future work are given.

5.1. Conclusion

In the first part of this thesis, several methods to perform IVA have been evaluated for two different environments. The first method performed the basic IVA implementation with different numbers of microphones. The second method performed IVA, using the PCA subspace method with different numbers of principal components. The subspace method is preferable, since all available microphones can be utilized and only the number of principal components has to be chosen. For performing basic IVA, the number of microphones and a microphone combination have to be chosen. The performance of both methods depends on the number of selected microphones or principal components and the environment, in which the recordings were made.

For the anechoic room scenario, a small number of microphones or principal components showed the best results. With the subspace method, the separation results in the anechoic case were better, than with selecting a small number of microphones.

In the office room scenario, a higher number of microphones or principal components should be used. Here, the basic IVA implementation, using all eight microphones, which is the same as the subspace method, using eight principal components, showed the best results.

It was discovered, that the performance of the subspace method strongly depends on the STFT window length. For very short window lengths, the performance in the office room for small numbers of principal components could be improved. But for making reliable statements, more evaluation would be needed.

Overall, when performing IVA, the separation method and the number of used microphones or principal components should be chosen with respect to the environment. Finding the optimal configuration was done by experiments.

In the second part of this thesis, it was studied, how the source separation can be connected with a speaker recognition system. A lot of different methods have been combined for the model training and the speaker recognition, in the case of one active speaker and the case of two active speakers.

In the case of one active speaker, IVA did not improve the recognition performance. Only by applying a PCA, choosing one principal component without separation, the recognition

rate can be improved slightly. So, we can say, that applying source separation in the case of one active speaker makes no sense.

In the case of two active speakers, the recognition performance of the speaker recognition could be improved. Without separation, the speaker recognition was not able to detect both speakers correctly in the anechoic scenario, as well as in the office scenario, which corresponds to a DER of 100%. And also in 48% of the speech segments, at least one speaker has been detected correctly in the anechoic case. In the office case, this rate was even worse, with 37% of correctly detected segments.

By applying IVA, the recognition performance could be improved. In the anechoic case, the DER could be decreased from 100% to 34%, which means, that for 66% of the segments, both speakers have been detected correctly. For 94%, at least one of the two speakers has been detected correctly, which is a great improvement, since without separation this rate was only 48%. In the office scenario, the DER could be decreased from 100% to 52.6%, which means, that in 47.4% of the segments, both speaker have been detected correctly. The rate of detecting at least one speaker could be increased to 95% in the office scenario.

Additional tests have shown, that using a higher sampling rate for the speaker recognition and the model training can improve the recognition rate by 10%.

So, we can say, that BSS can improve the performance of the speaker recognition. But also here, it is important to chose the separation methods depending on the environment. The results of the speaker recognition in combination with the source separation are not completely consistent with the evaluation results of source separation. For example, in the evaluation of the source separation in the office room case, eight microphones showed the best separation results. But in the evaluation of the speaker recognition, the subspace method, using four principal components showed better results.

Overall, when combining BSS with speaker recognition, it is important to treat segments, that contain only one speaker differently than segments, containing two speakers. When only one speaker is active, it is better to apply no separation or a PCA with one principal component, to reduce some noise. When more than two speakers are active, IVA should be performed. Which method should be used, depends on the environment. Also here, it is not easy to find the optimal configuration.

A reliable detection of the number of active speakers is needed, which is also not easy, especially in echoic environments. In an anechoic room, the number of speaker can be estimated reliable by applying PCA.

Altogether, it can be said, that to build a reliable system for combined BSS and speaker recognition, a lot of improvements have to be made, since the performance of the complete system strongly depends on the selected parameters, used for the separation.

5.2. Future Work

As can be seen from the results above, there are some points, that have to be improved, in order to build a joint BSS and speaker recognition system for teleconferences.

The first point is to find a connection between the reverberation time of a room and the ideal parameters for IVA, to yield the best separation results for every environment.

Also the optimal STFT window length has to be found, since the subspace method achieved better results for shorter window lengths.

Another important point is to find a reliable method for the detection of the number of active speakers. For the anechoic scenario, the number of speakers can already be determined by analyzing the distribution of the eigenvalues. But for echoic rooms, it is hard to determine the number of sources by the eigenvalues, since, due to reflections, there are more dominant eigenvalues than active sources. In [23], two information theoretic criteria are proposed to estimate the number of signals, that could be implemented.

Also an automatic segmentation is needed, that divides the signal into segments, depending on the number of active speakers. But for this, also a reliable detection of active speakers is needed. With an automatic segmentation, the speaker recognition could be also tested on data, recorded in a real conference scenario.

A. Appendix

A.1. DVD Content

All Matlab functions and scripts, used in this thesis, are contained in the attached DVD. Also, all evaluation results are stored and visualized in excel files.

The DVD is structured as follows:

- **[\Matlab\]** : Contains all Matlab functions and scripts
- **[\Separation Results\]** : Contains all results of the source separation evaluation
- **[\Recognition Results\]** : Contains all results of the speaker recognition evaluation
- **[\Diplomarbeit - Latex Files\]** : Contains all LaTeX files of this thesis
- **[\Projektplan\]** : Contains the project proposal of this thesis
- **[\Quellen\]** : Contains all papers, listed in the bibliography
- **[\Fotos Versuchsaufbau\]** : Contains some pictures of the recording set-up in the office room

A.2. List of the Evaluated Microphone Combinations

On the following page, a table (Table A.1), listing all evaluated microphone combinations is shown.

Number of microphones	Microphone combinations (a,b, ...)
$m = 2$	(1,2); (1,3); (1,4); (1,5); (1,6); (1,7); (1,8); (2,3); (2,4); (2,5); (2,6); (2,7); (2,8); (3,4); (3,5); (3,6); (3,7); (3,8); (4,5); (4,6); (4,7); (4,8); (5,6); (5,7); (5,8); (6,7);(6,8); (7,8);
$m = 3$	(1,2,3); (1,2,4); (1,2,5); (1,2,6); (1,2,7); (1,2,8); (1,3,4); (1,3,5); (1,3,6); (1,3,7); (1,3,8); (1,4,5); (1,4,6); (1,4,7); (1,4,8); (1,5,6); (1,5,7); (1,5,8); (1,6,7); (1,6,8); (1,7,8); (2,3,4); (2,3,5); (2,3,6); (2,3,7); (2,3,8); (2,4,5); (2,4,6); (2,4,7); (2,4,8); (2,5,6); (2,5,7); (2,5,8); (2,6,7); (2,6,8); (2,7,8); (3,4,5); (3,4,6); (3,4,7); (3,4,8); (3,5,6); (3,5,7); (3,5,8); (3,6,7); (3,6,8); (3,7,8); (4,5,6); (4,5,7); (4,5,8); (4,6,7); (4,6,8); (4,7,8); (5,6,7); (5,6,8); (5,7,8); (6,7,8);
$m = 4$	(1,2,3,4); (1,2,3,5); (1,2,3,6); (1,2,3,7); (1,2,3,8); (1,2,4,5); (1,2,4,6); (1,2,4,7); (1,2,4,8); (1,2,5,6); (1,2,5,7); (1,2,5,8); (1,2,6,7); (1,2,6,8); (1,2,7,8); (1,3,4,5); (1,3,4,6); (1,3,4,7); (1,3,4,8); (1,3,5,6); (1,3,5,7); (1,3,5,8); (1,3,6,7); (1,3,6,8); (1,3,7,8); (1,4,5,6); (1,4,5,7); (1,4,5,8); (1,4,6,7); (1,4,6,8); (1,4,7,8); (1,5,6,7); (1,5,6,8); (1,5,7,8); (1,6,7,8); (2,3,4,5); (2,3,4,6); (2,3,4,7); (2,3,4,8); (2,3,5,6); (2,3,5,7); (2,3,5,8); (2,3,6,7); (2,3,6,8); (2,3,7,8); (2,4,5,6); (2,4,5,7); (2,4,5,8); (2,4,6,7); (2,4,6,8); (2,4,7,8); (2,5,6,7); (2,5,6,8); (2,5,7,8); (2,6,7,8); (3,4,5,6); (3,4,5,7); (3,4,5,8); (3,4,6,7); (3,4,6,8); (3,4,7,8); (3,5,6,7); (3,5,6,8); (3,5,7,8); (3,6,7,8); (4,5,6,7); (4,5,6,8); (4,5,7,8); (4,6,7,8); (5,6,7,8);
$m = 5$	(1,2,3,4,5); (1,2,3,4,6); (1,2,3,4,7); (1,2,3,4,8); (1,2,3,5,6); (1,2,3,5,7); (1,2,3,5,8); (1,2,3,6,7); (1,2,3,6,8); (1,2,3,7,8); (1,2,4,5,6); (1,2,4,5,7); (1,2,4,5,8); (1,2,4,6,7); (1,2,4,6,8); (1,2,4,7,8); (1,2,5,6,7); (1,2,5,6,8); (1,2,5,7,8); (1,2,6,7,8); (1,3,4,5,6); (1,3,4,5,7); (1,3,4,5,8); (1,3,4,6,7); (1,3,4,6,8); (1,3,4,7,8); (1,3,5,6,7); (1,3,5,6,8); (1,3,5,7,8); (1,3,6,7,8); (1,4,5,6,7); (1,4,5,6,8); (1,4,5,7,8); (1,4,6,7,8); (1,5,6,7,8); (2,3,4,5,6); (2,3,4,5,7); (2,3,4,5,8); (2,3,4,6,7); (2,3,4,6,8); (2,3,4,7,8); (2,3,5,6,7); (2,3,5,6,8); (2,3,5,7,8); (2,3,6,7,8); (2,4,5,6,7); (2,4,5,6,8); (2,4,5,7,8); (2,4,6,7,8); (2,5,6,7,8); (3,4,5,6,7); (3,4,5,6,8); (3,4,5,7,8); (3,4,6,7,8); (3,5,6,7,8); (4,5,6,7,8);
$m = 6$	(1,2,3,4,5,6); (1,2,3,4,5,7); (1,2,3,4,5,8); (1,2,3,4,6,7); (1,2,3,4,6,8); (1,2,3,4,7,8); (1,2,3,5,6,7); (1,2,3,5,6,8); (1,2,3,5,7,8); (1,2,3,6,7,8); (1,2,4,5,6,7); (1,2,4,5,6,8); (1,2,4,5,7,8); (1,2,4,6,7,8); (1,2,5,6,7,8); (1,3,4,5,6,7); (1,3,4,5,6,8); (1,3,4,5,7,8); (1,3,4,6,7,8); (1,3,5,6,7,8); (1,4,5,6,7,8); (2,3,4,5,6,7); (2,3,4,5,6,8); (2,3,4,5,7,8); (2,3,4,6,7,8); (2,3,5,6,7,8); (2,4,5,6,7,8); (3,4,5,6,7,8);
$m = 7$	(1,2,3,4,5,6,7); (1,2,3,4,5,6,8); (1,2,3,4,5,7,8); (1,2,3,4,6,7,8); (1,2,3,5,6,7,8); (1,2,4,5,6,7,8); (1,3,4,5,6,7,8); (2,3,4,5,6,7,8);
$m = 8$	(1,2,3,4,5,6,7,8)

Table A.1.: All different microphone combinations

A.3. List of all Functions and Scripts

[IVA\]	Contains all functions for source separation
detect_num_spk_pca.m	Tries to detect the number of active speakers by analyzing the eigenvalues. Can also plot the eigenvalues. (Works only in anechoic rooms)
gui.m	Graphical user interface for applying IVA all different parameters and methods
iva_pca.m	Perform IVA, using the PCA subspace method
pca.m	Performs a PCA in frequency domain
pca_analyze.m	Performs a PCA in frequency domain, but without inverting the spectral transformation. Outputs whitened signals in time domain.

[IVA\denk\]	Contains the basic IVA implementation
inv_st_fft.m	Inverse short-time Fourier transform
istft_SiSec2008.m	Multichannel inverse short-time Fourier transform (ISTFT) using half-overlapping sine windows
iva_data.m	Class that holds all data of the IVA sound source separation algorithm
iva_general.m	Separates sound mixtures
short_time_fft.m	Splits up several input mixtures in Frequency bins and performs FFT on each bin
stft_SiSec2008.m	Multichannel short-time Fourier transform (STFT) using half-overlapping sine windows

[IVA\evaluation\]	Evaluation functions
estimate_delay.m	Estimates the delay between two signals
evaluate_all_combinations_1src.m	Performs IVA for all possible microphone combinations and then evaluates the separation results (for 1 speaker)
evaluate_all_combinations_2src.m	Performs IVA for all possible microphone combinations and then evaluates the separation results (for 2 speakers)

evaluate_directory_2src.m	Evaluation script for evaluating all files in the current directory for 2 sources by basic IVA
evaluate_directory_iva_pca.m	Evaluation script for evaluating all files in the current directory for 2 sources by IVA PCA subspace method
evaluate_directory_iva_pca_different_-window_sizes.m	Evaluates all files in a directory by IVA PCA subspace method, using different window lengths for the STFT
evaluate_iva_pca_2src.m	Performs IVA with PCA subspace method and then evaluates the separation results (for 2 speakers)
plot_results.m	Plot the SDR, SIR and SAR values for basic IVA
plot_results_iva_pca.m	Plot the SDR, SIR and SAR values for IVA PCA subspace method

[IVA\evaluation\bss_eval_3.0\]	Contains evaluation function of the BSS toolbox
bss_eval_sources.m	Calculate SDR, SIR and SAR values

[Speaker Recognition\]	Contains all functions for speaker recognition
detect_speaker.m	Detects the speaker of a speech segment.
trainModel_iva.m	Trains a GMM, using IVA before the training
trainModel_iva_pca.m	Trains a GMM, using PCA subspace method before the training

[Speaker Recognition\kozielski]	Contains the basic speaker recognition functions
EM.m	Implementation of the EM algorithm
enframe.m	Window a signal
extractFeatures.m	Extract features out of a signal
initEM.	Initialize EM algorithm by k-means
logLikelihood.m	Calculate log-likelihood
map.m	MAP adaptation
mel2frq.m	Transform mel scale to linear frequency
melcepst.m	Calculate the MFCCs

PROPERTIES.m	Defines all important parameters for speaker recognition centrally
trainGMM.m	Train a GMM
vad.m	Voice activity detection
vad_old.m	Old VAD version with a different approach

[Speaker Recognition\kozielski\tools\]	Tools from Voicebox
activlev.m	Estimates the active speech level
estnoisem.m	Estimates the ground noise level
frq2mel.m	Transform linear frequency into mel scale
gaussmix.m	Fits a Gaussian mixture pdf to a set of data observations
gaussPDF.m	Computes PDF of a Gaussian distribution
lmultigauss.m	Computes multigaussian log-likelihood
logsum.m	log(sum(exp()))
lsum.m	Sum up logarithmically
m2htmlpwd.m	Creates a HTML documentation of the current folder
maxfilt.m	Find max of a filter
mel2frq.m	Transform mel scale to linear scale
melbankm.m	Mel bank filter function
nearnonz.m	Create a value close to zero
rdct.m	Calculate DCT of real data
rfft.m	Calculate DFT of real data

[Speaker Recognition\]	Folders
[recognition scripts\]	Contains all recognition scripts
[speakerModels_dialog1\]	Contains all speaker models for dialog 1
[speakerModels_dialog2\]	Contains all speaker models for dialog 2
[training scripts\]	Contains all model training scripts

A.4. SDR, SIR, SAR Values for 2 Microphones for the Anechoic Recordings

On the following pages you can find the complete evaluation results for all combinations of 2 microphones for the anechoic recordings, containing the SDR, SIR and SAR values for all evaluated speaker distances $(180°, 135°, 90°, 45°, 25°)$.

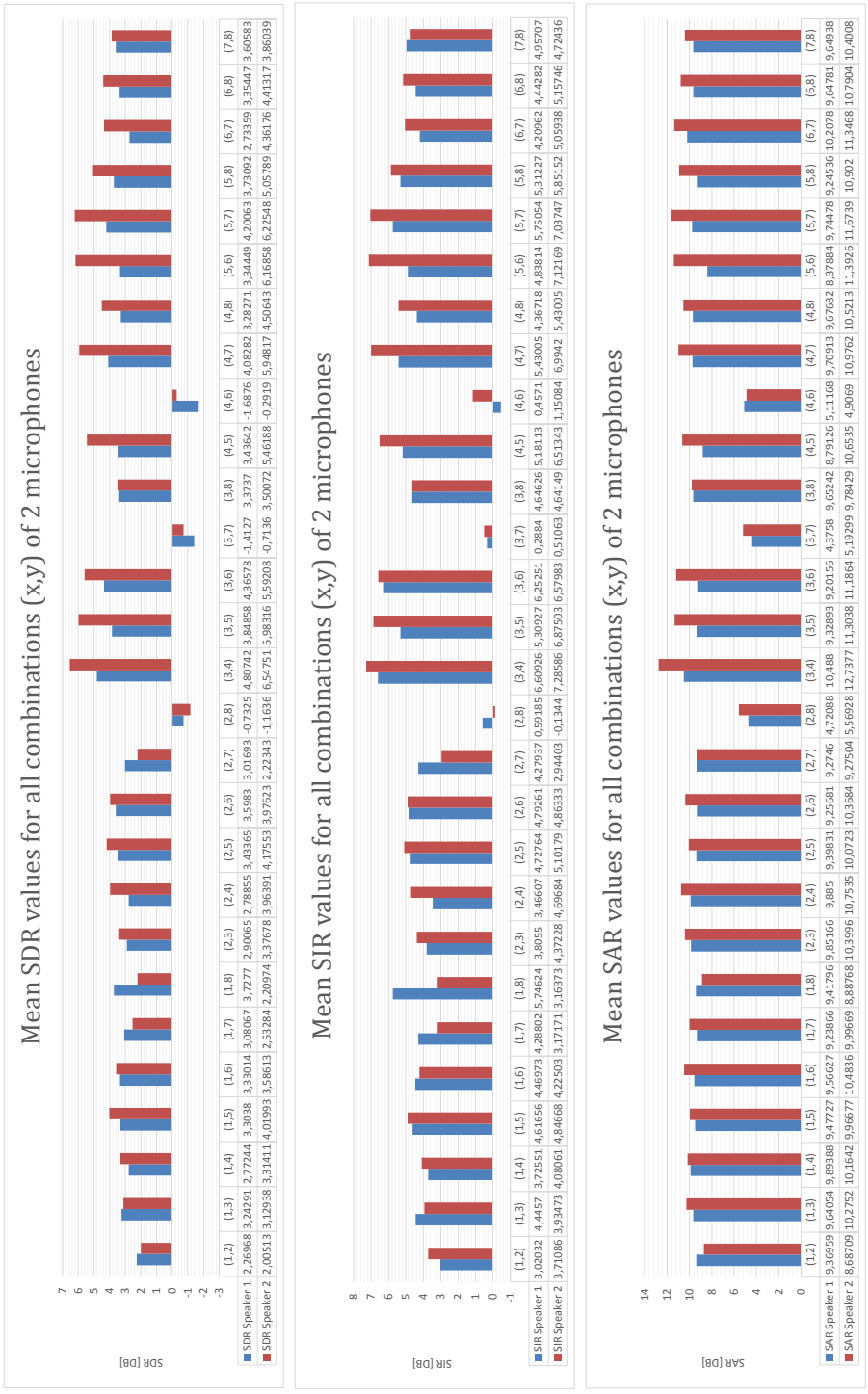


Figure A.1.: SDR, SIR, SAR for 2 microphones, 180° speaker distance, anechoic

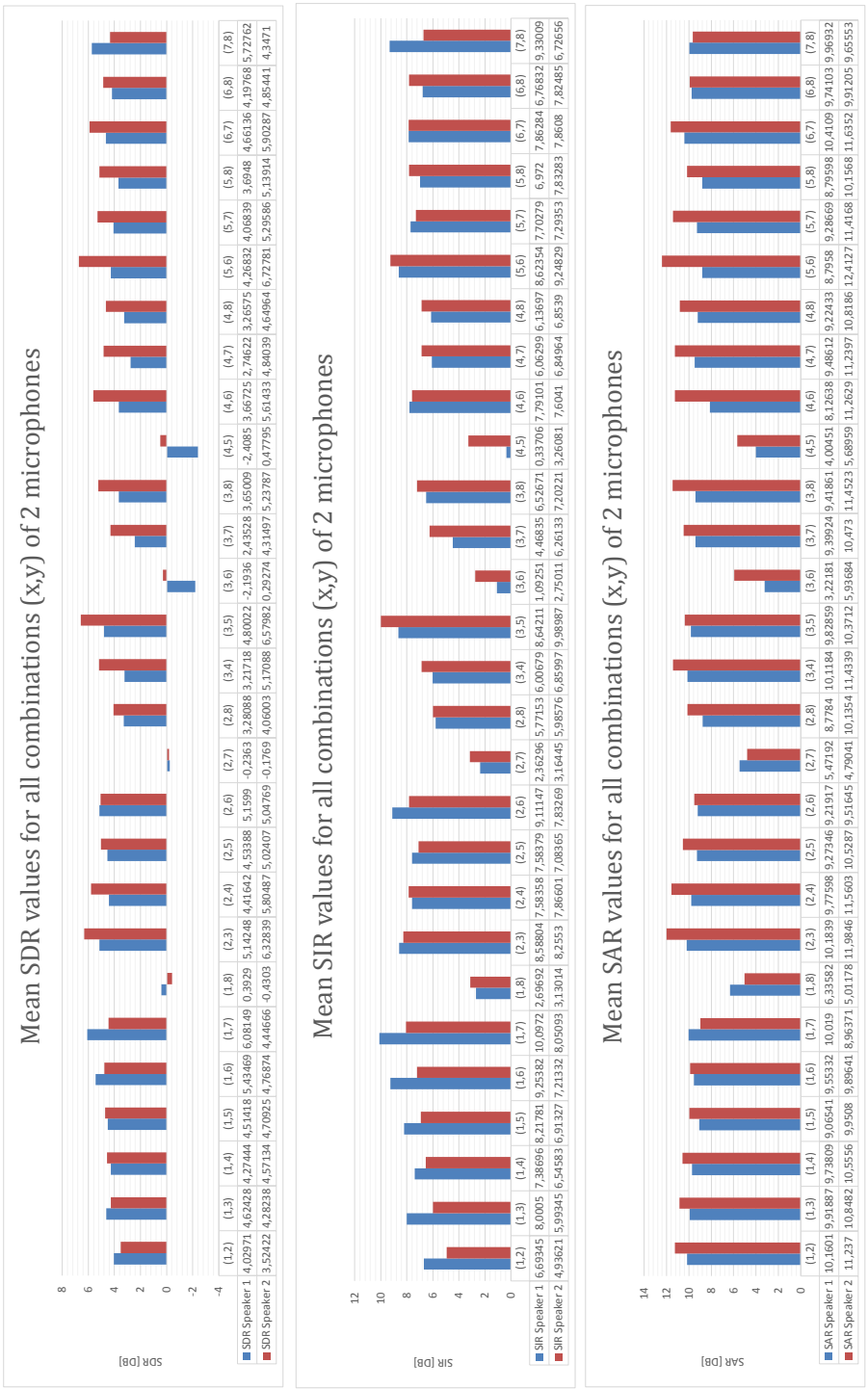


Figure A.2.: SDR, SIR, SAR for 2 microphones, 135° speaker distance, anechoic

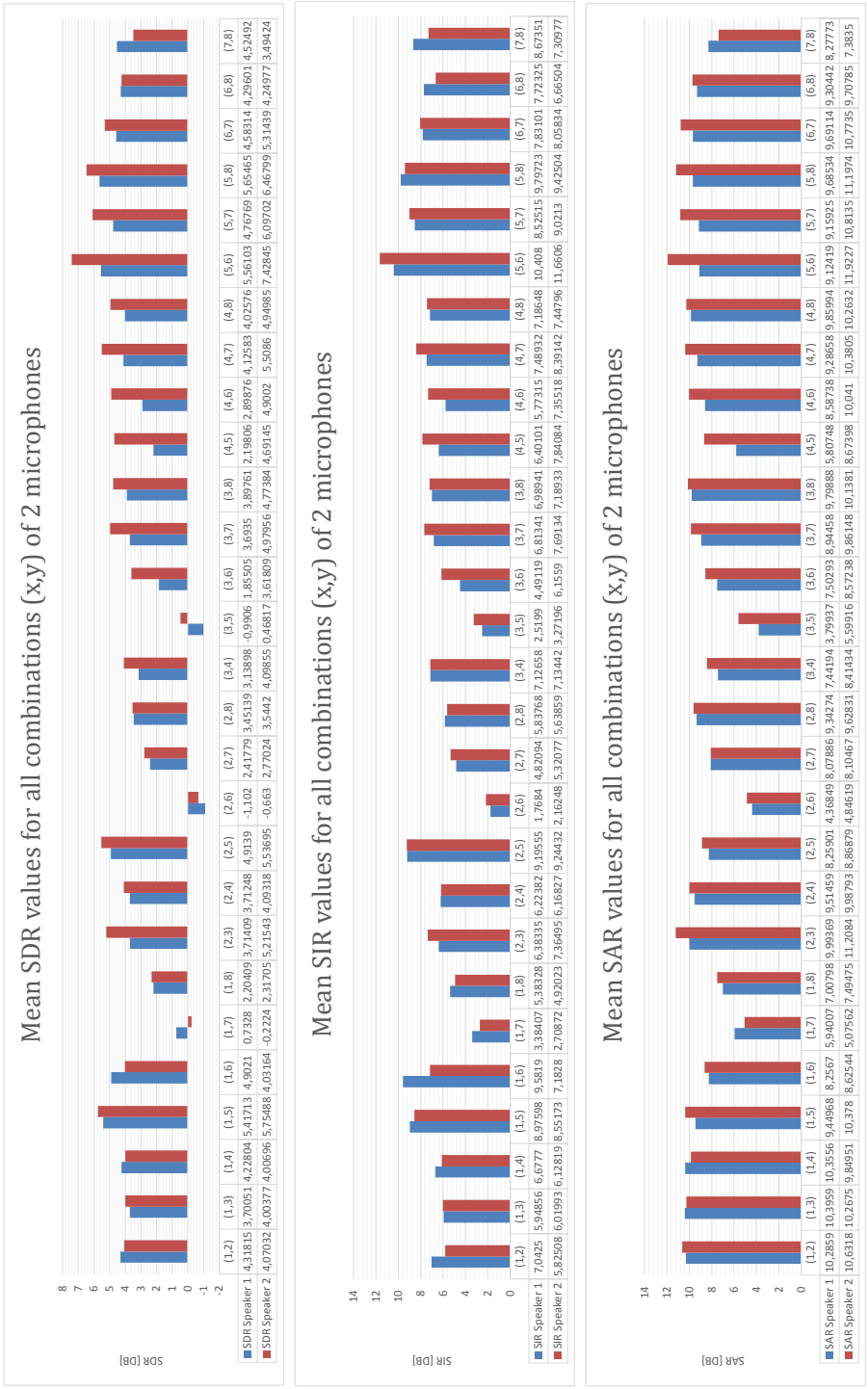


Figure A.3.: SDR, SIR, SAR for 2 microphones, 90° speaker distance, anechoic

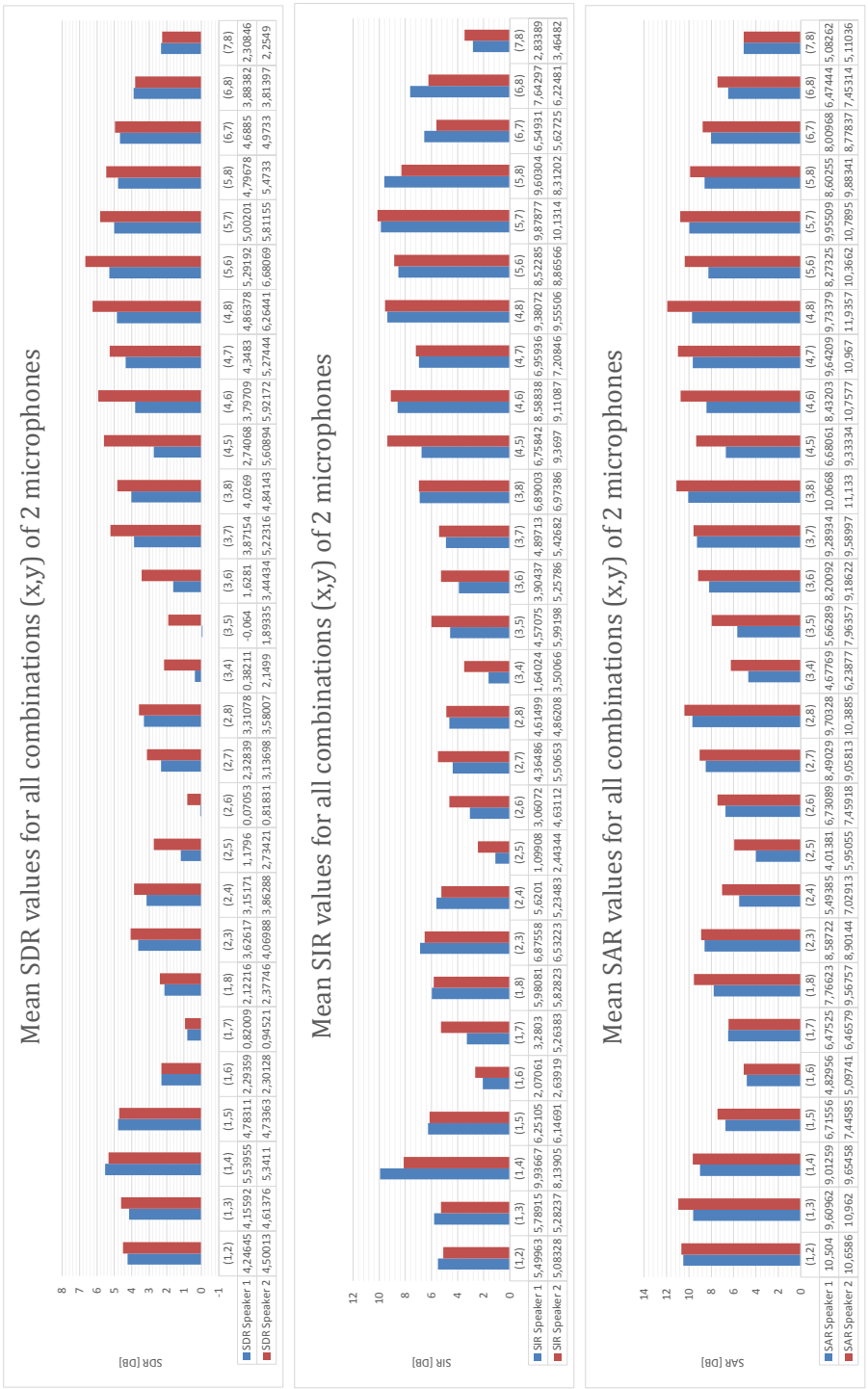


Figure A.4.: SDR, SIR, SAR for 2 microphones, 45° speaker distance, anechoic

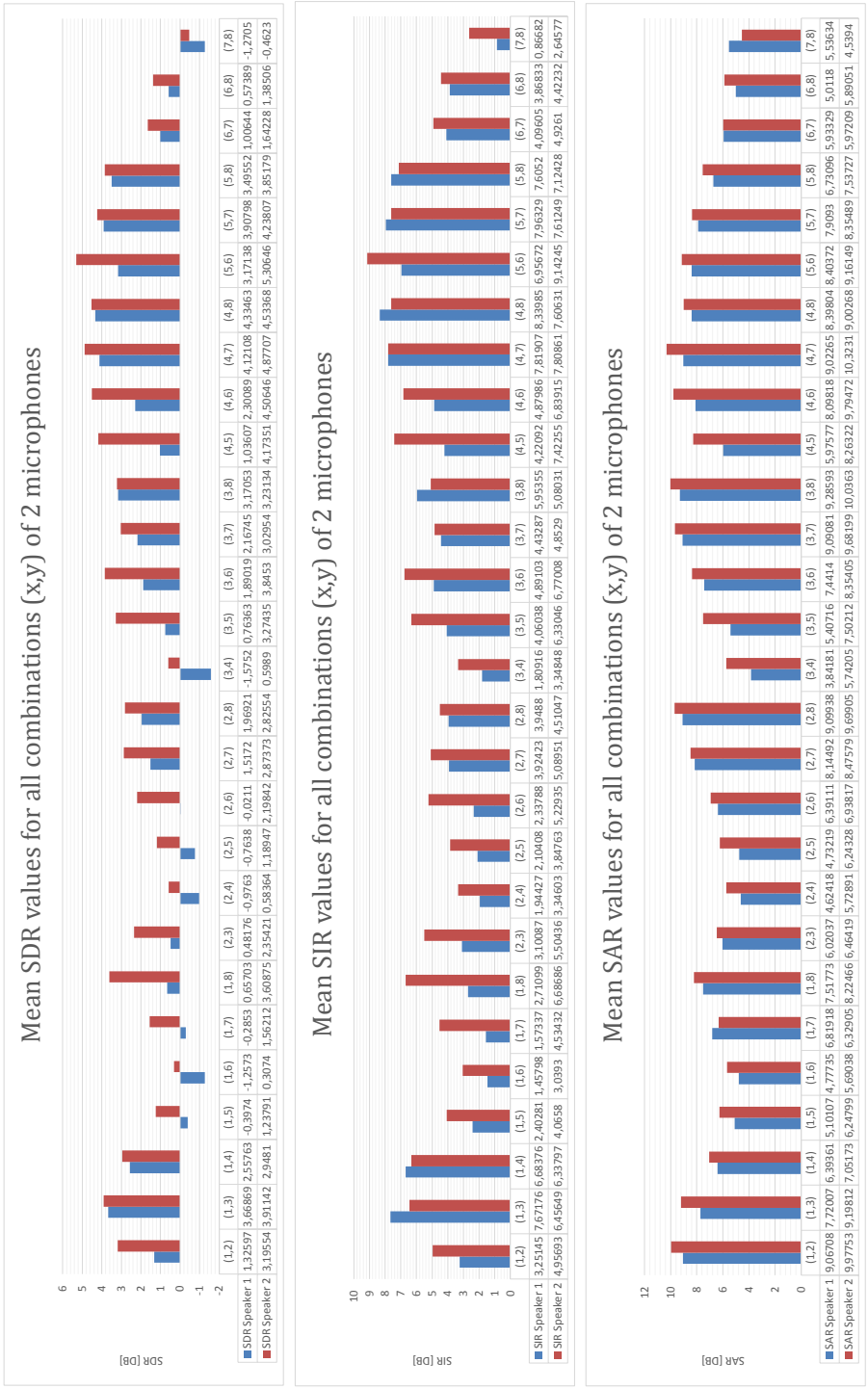


Figure A.5.: SDR, SIR, SAR for 2 microphones, 25° speaker distance, anechoic

Bibliography

[1] AMIDA Project. Conversational multi-party speech recognition using remote microphones. 2007. State-of-the-art overview, *AMI Consortium*.

[2] F. Asano, S. Ikeda, M. Ogawa, H. Asoh, and N. Kitawaki. Combined approach of array processing and independent component analysis for blind separation of acoustic signals. In *IEEE Transactions on Speech and Audio Processing*, 11(3), pp. 204 – 215, 2003.

[3] F. Asano, Y. Motomura, H. Asoh, and T. Matsui. Effect of PCA filter in blind source separation. In *International Workshop on Independent Component Analysis and Blind Signal Separation*, pp. 57–62. 2000.

[4] J. Benesty, M. Sondhi, and Y. Huang. *Springer Handbook of Speech Processing*. Springer-Verlag Berlin Heidelberg, 2008.

[5] K. Boakye, B. Trueba-Hornero, O. Vinyals, and G. Friedland. Overlapped speech detection for improved speaker diarization in multiparty meetings. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2009.

[6] BUCHFUNK Verlag GbR. Vorleser.net - literatur hören. Online. URL `http://www.vorleser.net/`. Last visit: 03.09.2012.

[7] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. 1st edition. Academic Press, Elsevier Oxford, 2010.

[8] C. Denk and M. Rothbucher. Robotic sound source separation using independent vector analysis. 2011. Project thesis at the *Institute for Data Processing, Technische Universität München*.

[9] J. Feldmaier. Sound localization and separation for teleconferencing systems. 2011. Diploma thesis at the *Institute for Data Processing, Technische Universität München*.

[10] G. Friedland, H. Hung, and C. Yeo. Multi-modal speaker diarization of real-world meetings using compressed-domain video features. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4069–4072. 2009.

[11] J. Hao, I. Lee, T. Lee, and T. Sejnowski. Independent vector analysis for source separation using a mixture of gaussians prior. In *Neural Computation*, 22(6), pp. 1646 –1673, 2010.

[12] S. Hirayanagi and N. Hamada. A solution for the permutation proplem of overdetermined source separation using subspace method. In *International Workshop on Acoustic Echo and Noise Control*, pp. 101–104. 2005.

[13] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, Inc. New York, 2001.

[14] C.D. Kozielski. Online speaker recognition for teleconferencing systems. 2011. Diploma thesis at the *Institute for Data Processing, Technische Universität München*.

[15] I. Lee, T. Kim, and T.W. Lee. Fast fixed-point independent vector analysis algorithms for convolutive blind source separation. In *Signal Processing*, 87(8), pp. 1859 – 1871, 2007.

[16] T. Lee. *Independent Component Analysis: Theory and Applications*. Kluwer Academic Publishers Boston, 1998.

[17] S. Makino, T.W. Lee, and H. Sawada. *Blind Speech Separation*. Signals and Communication Technology. Springer Dordrecht, 2007.

[18] A.K. Nandi. *Blind Estimation Using Higher-Order Statistics*. Kluwer Academic Publishers Boston, 1999.

[19] M. Rothbucher, M. Kaufmann, J. Feldmaier, T. Habigt, M. Durkovic, C. Kozielski, and K. Diepold. 3D audio conference system with backward compatible conference server using HRTF synthesis. In *Journal of Multimedia Processing and Technologies*, 2(4), pp. 159–175, to appear.

[20] S. Tranter and D. Reynolds. An overview of automatic speaker diarization systems. In *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5), pp. 1557 –1565, 2006.

[21] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. In *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4), pp. 1462 –1469, 2006.

[22] E. Vincent. BSS Eval A toolbox for performance measurement in (blind) source separation. Online. URL `http://bass-db.gforge.inria.fr/bss_eval/`. Last visit: 27.07.2012.

[23] M. Wax and T. Kailath. Detection of signals by information theoretic criteria. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33(2), pp. 387 – 392, 1985.

[24] S. Winter, H. Sawada, and S. Makino. Geometrical interpretation of the PCA subspace method for overdetermined blind source separation. In *International Symposium on Independent Component Analysis and Blind Signal Separation*, pp. 775–780. 2003.