The University Of Sheffield.

Diploma thesis

Online Signal Separation Based on Microphone Arrays in a Multipath Environment

Stefan Richardt

Signal Processing and Speech Communications Laboratory
Graz University of Technology, Austria

Department of Electronic & Electrical Engineering
University of Sheffield, United Kingdom

Supervisors:
Dipl.-Ing. Dr.sc.ETH Harald Romsdorfer
Dr. Wei Liu

Assessor:
Univ.-Prof. Dipl.-Ing. Dr.techn. Gernot Kubin

Graz, March 2011

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, _____ _____

Place, Date Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, _____ _____

Ort, Datum Unterschrift

Abstract

This thesis aims at constructing a system, being able to separate the signals of simultaneously speaking persons in a room. The desired source signal is supposed to be extracted from the actual mix of appearing sources. It shall be recovered from any noise or interfering sources, such as other speakers.

The approach is based on a microphone array whereas the received data are processed in two successive stages. Firstly, a beamforming network, consisting of several fixed beamformers steering in different directions, scans the room. The second stage, a Blind Source Separation algorithm, controls the individual beams in order to separate the desired signal from any interferences as well as possible.

Furthermore, it was required to construct twenty analogue amplifiers in order to complete the available hardware setup. The final result is a fully functional system consisting of a Matlab Graphical User Interface utilizing the associated hardware. It enables the user to listen separately to the individual speakers in a room.

Kurzfassung

Ziel dieser Arbeit ist es ein System zu erstellen, welches in der Lage ist Sprachsignale mehrerer zeitgleich agierender Sprecher in einem Raum zu extrahieren. Das Originalsignal eines bestimmten Sprechers soll dabei so gut wie möglich zurück gewonnen werden, wobei es von etwaigen Störungen wie anderen Sprechern oder Rauschen befreit werden soll.

Der auf einem *Mikrophone Array* basierende Ansatz arbeitet in zwei Stufen. Ein Netzwerk aus mehreren unveränderlichen *Beamformern* bildet die erste Stufe. Jeder dieser *Beamformer* ist richtungsabhängig, wobei nur Signale aus einer bestimmten Richtung durchgelassen und Signale aus anderen Richtungen gedämpft werden. Die *Beamformer*, welche in verschiedene Richtungen zeigen, werden von der zweiten Stufe, einem *Blind Source Separation* Algorithmus gesteuert, um das gewünschte Signal bestmöglich von Störungen bzw. anderen Sprechern zu befreien.

Zur Vervollständigung der vorliegenden Hardware war es notwendig 20 analoge Verstärker zu entwerfen und aufzubauen. Als Ergebniss liegt letztendlich ein voll funktionsfähiges System vor, wobei ein Matlab *Graphical User Interface* auf die zugehörige Hardware zugreift. Es wird dem Benutzer ermöglicht, sich die einzelnen Sprecher im Raum unabhänging voneinander anzuhören.

Acknowledgement

I would like to thank my supervisors Wei Liu and Harald Romsdorfer, who gave my the opportunity to accomplishment my diploma thesis at the University of Sheffield straightforwardly and who advised and supported me throughout my thesis. Thanks to the people in the lab, especially to James and Jon for helping me out wherever possible and to Bo for helpful inputs and interesting conversations.

I wish to thank all my friends and fellow students especially Ben for proofreading this thesis and for various discussions providing valuable suggestions. Special thanks to Franzi, always supporting and encouraging me even when I was planing to write my diploma thesis in Sheffield. Foremost, I want to thank my parents for enabling my studies and for their great support.

Graz, March 2011 Stefan Richardt

Contents

1 Introduction

The objective of this thesis is to apply an algorithm being able to separate speech signals in a reverberant environment by using a microphone array. As there are many different ways to access this topic this thesis focuses on a specific approach proposed in [Liu and Mandic, 2005] and [Liu, 2010]. Frequency Invariant Beamformer (FIB) technique as well as Blind Source Separation (BSS) technique are utilized and combined. Several fixed beamformers with different and unchangeable main steering directions, which are uniformly distributed on a half plane, are weighted by an adaptive BSS algorithm in order to recover different speakers in a room while disturbing interferences are suppressed. Therefore, the thesis can be allocated to two main fields of research:

Acoustic Array Signal Processing / Beamforming

Blind Source Separation (BSS) and Independent Component Analysis (ICA)

Both fields have been studied extensively. The main sources used in the course of this thesis are introduced here. 'Independent Component Analysis' [Hyvärinen et al., 2001] describes the basic principles of ICA in a comprehensible and descriptive way. 'Adaptive Blind Signal and Image Processing' [Cichocki and Amari, 2002] examines several algorithms in detail. Acoustic Array Processing and beamforming has been researched for example in 'Microphone Array Signal Processing' [Benesty et al., 2008] or 'Wideband Beamforming: Concepts and Techniques' [Liu and Weiss, 2010].

The main goal of this work is to prove and modify the proposed algorithm [Liu, 2010] in practice aiming to implement a real time version. All simulations as well as a Graphical User Interface (GUI) have been implemented in Matlab. Furthermore, it was claimed to build parts of the required hardware in the course of this thesis. I built twenty analogue amplifiers for what a proper board layout had to be designed. The amplifiers had to be integrated in the existing hardware (microphone array, data acquisition card, PC) whereas effects of electromagnetic compatibility had to be considered. The MATLAB GUI was developed in order to enable the user to easily record data and process these immediately by the algorithm. The algorithm itself is implemented modularly to ensure compatibility and exchangeability of certain stages.

The thesis is divided into three parts: Part I: Theory, Part II: Practical Approach and Part III: Results.

Part I contains the fundamentals of beamforming which are explained in Chapter 2 followed by an introduction of blind source separation in Chapter 3. A combination of both techniques is described in Chapter 4. Furthermore, a series of Matlab simulations is presented in order to work out suitable parameters for the practical approach.

Part II describes the practical work and is divided in two chapters. Chapter 5 includes the hardware setup whereas the practical problems and the development of the amplifiers are in the main focus. In Chapter 6 the developed Matlab GUI is introduced shortly.

Part III contains testing and evaluation of the system. In Chapter 7 the conducted experiments are described with the results being presented and interpreted in Chapter 8. The conclusion as well as the prospects are given in Chapter 9.

Part I

Theory

2 Beamforming

2.1 Introduction

Array processing is a broad research field having a long history in various application areas. It can be found in radar, sonar, communications, seismology but also in medical diagnosis and treatment [Benesty et al., 2008]. There are many purposes acoustic arrays respectively beamformers can be used for, such as estimating the Direction of Arrival (DOA) or gaining a desired signal with enhanced quality by recovering it from noise, different sources or reverberations. Facing a huge number of applications we need to classify our approach. Regarding the relative location of the sensors we can separate sensor arrays in three categories [Liu and Weiss, 2010]:

linear arrays (1-D)

planar arrays (2-D)

volumetric arrays (3-D)

Further, each of them can be divided into two classes:

regular spaced arrays with an either uniform or nonuniform sensor distribution

irregular or random spaced arrays

In general a beamformer can be described as a spatial filter supposed to form a certain beam certain pattern, also known as directivity pattern. As the requirements to a beamformer and its pattern differ a lot the complexity can reach from a very simple approach such as a delay and sum (D&S) beamformer to more complex structures like the Generalized Sidelobe Canceller (GSC). Therefore, it is useful to distinguish between narrow band and wide band beamformers. In the following sections a simple narrow band beamformer, the Delay and Sum beamformer, is explained. Next, a Filter and Sum (F&S) beamformer is deduced whereas a specific design process of the filter coefficients (window method) is described in particular. As the focus of this thesis lays on a particular approach, which combines a Frequency Invariant Beamformer (FIB) and Blind Source Separation (BSS) technique, it is abandoned to give an introduction of adaptive beamforming algorithms.

Temporal and Spatial Signal Phase

A common perspective of beamformers is the spatial filter. A spatial filter utilizes the different phases of the impinging signals. The phase of a signal, arriving at a certain point of the array, is not just time dependent but also dependent on the location of the sensor. It is common to treat the resulting delays, caused by the different positions of the sensors, as spatial sampling. Assuming an impinging signal with a specific frequency we sample all sensors signals at a specific point in time. Treating the values the same as a signal sampled in the time domain the frequency of the signal will be angle dependent. Several sensors at different locations will cause different phases for an instant time t. In further consequence, the signal phase is dependent on the direction of arrival (DOA) as we will show in this section.
A propagating plane wave from a certain DOA causes a different time delay respectively a

different phase at each sensor. Summing up all sensor signals various output levels according to the DOA are to be seen. The resulting magnitudes are caused by an either constructive or deconstructive interference of the sensor signals and leads to the corresponding beam pattern. Note that this pattern is frequency dependent, therefore, it is three dimensional.
The DOA is described by an azimuth angle ϕ and an elevation angle θ. The reference plane for the azimuth angle is at constant height whereas the elevation angle defines the deflation angle of this plane.

Assuming an impinging signal with a specific frequency we sample all sensors signals at a specific time. Treating the values the same as a signal sampled in the time domain the frequency of the signal will be angle dependent. For simplicity, we firstly consider a plane wave signal with

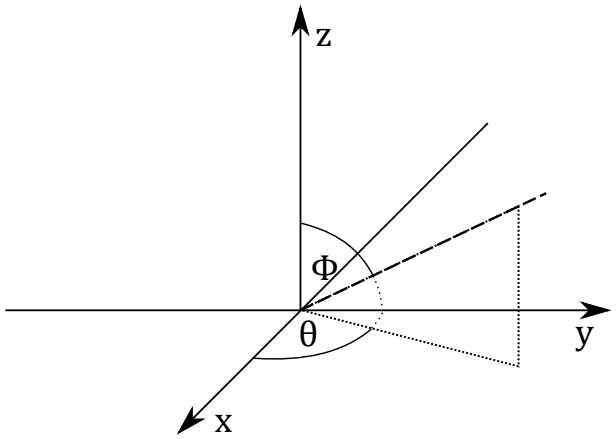

Figure 2.1: Cartesian coordinate system with an azimuth angle θ and an elevation angle ϕ describing the direction of arrival

a frequency f propagating in direction of the z axis. Figure 2.2 shows a plane wave with constant z plane in the Cartesian coordinate system.

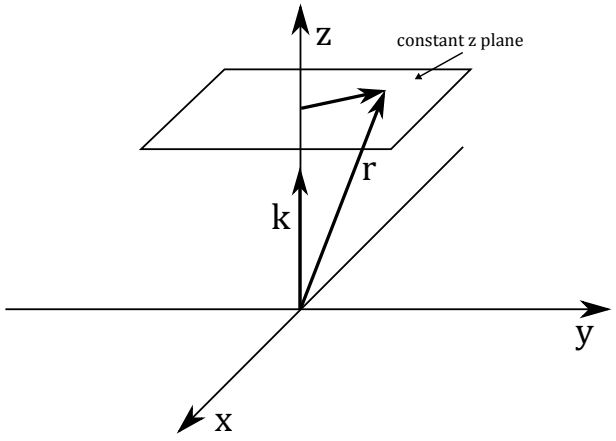

Figure 2.2: Plane wave with constant z plane propagating in z direction with spatial frequency $\boldsymbol{k}$

As mentioned earlier, the phase of a signal is crucial when it comes to summing up the signals. The phase term of a signal $\phi(t, z)$ is not longer exclusively a function of the time t but also of the position z and can be described as follows:

$$\phi(t, z) = \underbrace{\omega t}_{temporal} + \underbrace{kz}_{spatial} \tag{2.1}$$

In order to express the position term we need to introduce the wavenumber k:

$$k = \frac{\omega}{c} = \frac{2\pi f}{c} = \frac{2\pi}{\lambda} \tag{2.2}$$

k contains the temporal, angular frequency ω respectively the signal frequency f. Additionally, λ denotes the resulting wavelength and c is the propagation speed in a particular medium. Form Equation (2.1) it can be seen that ω is used as a fixed factor for the time t whereas k is used as fixed factor for the position z. As ω is called temporal frequency it is common to name k the spatial frequency. Instead of giving the number of periods per seconds k represents the number of periods per meter. Denote that k is also dependent on the signal frequency f. For a more detailed and well illustrated explanation of the wavenumber k see [Williams, 1999].
Assuming to shift the observation point along the z axis the phase changes according to the spacial frequency k whereby the phase shift is the same for all points in this plane.
Unlike the temporal frequency ω the spacial frequency k is three dimensional and points in the directions of the propagating wave. In Cartesian coordinates it can be denoted as a three dimensional vector:

$$\mathbf{k} = [k_x \; k_y \; k_z]^T \tag{2.3}$$

The length can be calculated as follows:

$$k = \sqrt{k_x^2 + k_y^2 + k_z^2} \tag{2.4}$$

In the particular case shown in Figure 2.2 $\mathbf{k}$ consits of $k_x = k_y = 0$ and $k_z = k$. Defining the unity vector $\hat{\mathbf{z}}$ along the z axis:

$$\hat{\mathbf{z}} = [0 \; 0 \; 1]^T \tag{2.5}$$

leads to:

$$\mathbf{k} = k\hat{\mathbf{z}} \tag{2.6}$$

If we want to show the relation between k and f we need to refer to Equation (2.2).
Within the next steps we describe any point in a 3D space. For that reason we introduce $\mathbf{r}$, which directly gives the coordinates in the Cartesian system. Using $\mathbf{k}$ we are now able to formulate the phase $\phi(t, \mathbf{r})$ as function of $\mathbf{r}$, enabling to give the phase of a signal at every possible position in a room. Therefore, we rewrite Equation (2.1):

$$\phi(t, \mathbf{r}) = \underbrace{\omega t}_{temporal} + \underbrace{\mathbf{k}^T\mathbf{r}}_{spatial} \tag{2.7}$$

with:

$$\mathbf{r} = [r_x \; r_y \; r_z]^T \tag{2.8}$$

In order to gain more generality we replace the coordinates by the angles θ and ϕ. As we assume

plane waves, analogous to the far field assumption, there is no need to include the distance from the source to the sensor. Finally we are able to determine a time independent phase term, which is a function of the DOA as only the angles of an impinging signal are significant.

$$\mathbf{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} = k \begin{bmatrix} sin\,\theta\,cos\,\phi \\ sin\,\theta\,sin\,\phi \\ cos\,\theta \end{bmatrix} \tag{2.9}$$

Applying Equation (2.8) and Equation (2.9) in Equation (2.7) and focusing on the time independent phase term we obtain:

$$\mathbf{k}^T\mathbf{r} = sin\,\theta\,cos\,\phi\,r_x + sin\,\theta\,sin\,\phi\,r_y + cos\,\theta\,r_z \tag{2.10}$$

Thus, we have expressed the phase of a signal as function of time and DOA. As we are able to describe the phase of a signal at every point in 3D space we want to utilize the spatial phase shift. Placing sensors at different locations constitutes an array and in further consequence a beamformer.

2.2 Problem description, Assumptions and Specifications

Before explaining the D&S beamformer as well as the FIB it is useful to give a general problem description. Besides, we need to make some basic assumptions of the environment conditions. In addition, we give the specifications of the particular approach, implemented in course of this thesis.
Our approach is an one-dimensional linear array with an uniform spacing d, the distance between two microphones. Therefore, our considerations are constrained to a plane. As DAO of the impinging signals we consequently only regard the azimuth angle θ, which is constrained to $\theta \in [-\pi/2\ \pi/2]$. It is useful just to consider a half plane as the results are mirrored at the broadside of the beamformer. Figure 2.3 shows a linear array structure with an impinging plane wave.

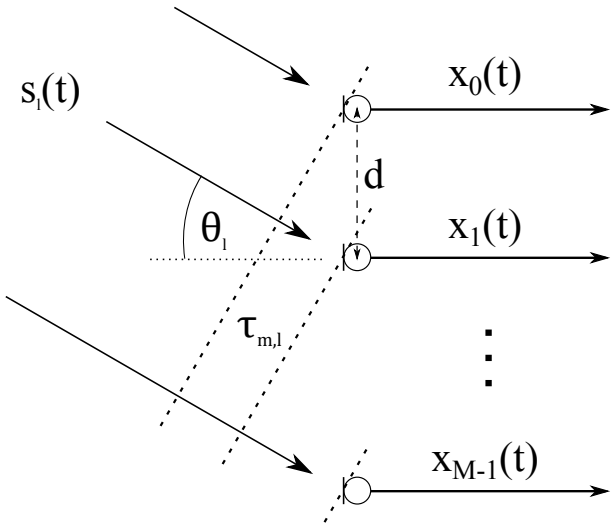

Figure 2.3: Linear array with uniform spacing and impinging plane wave

s_l ... l-th source signal
L ... number of sources
x_M ... x-th microphone signal
M ... number of microphones
d ... distance between microphones
θ_l ... DOA / angle of impinging signal of l-th source
$\tau_{m,l}$ ... time delay to reference micrphone

We suppose that $s_l(t)$ is the l-th impinging signal with $l = 0, 1, \ldots, L-1$ where L is the number of source signals. According to each of the source signals we describe the direction of arrival by θ_l. Basically our model contains M sensors, which provide the received signals $x_0(t), x_1(t), \ldots, x_{M-1}(t)$. We consider the zeroth sensor to be our reference. Hence, we are able to define $\tau_{m,l}$ as time delay from the reference sensor to the m-th sensor according to the l-th source. The delay $\tau_{m,l}$ depends on the angle of the impinging signal θ_l and the distance of the m-th sensor form the reference. As we have an uniform sensor distribution with a fixed distance d between the microphones the delay $\tau_{m,l}$ can be considered as:

$$\tau_{m,l} = sin(\phi_l)\,\frac{m \cdot d}{c} \tag{2.11}$$

The wave propagation speed of sound waves in air is:

$$c \approx 340\,\frac{m}{s} \tag{2.12}$$

In general the signal at the m-th sensor can be described by:

$$x_m(t) = \alpha_m s(t - t_l - \tau_{m,l}) + \upsilon_n(t) \tag{2.13}$$

with:

α_m ... attenuation factor of microphone m
t_l ... time delay from the source l to the reference sensor
$\upsilon_n(t)$ ... uncorrelated noise

Ideally the attenuation factor α_m is the same for all microphones as we assume plane waves respectively far field conditions. Apart from the additional noise we basically obtain the same signal $s(t)$ whose differences in terms of time delay are exclusively determined by $\tau_{m,l}$. The noise $\upsilon_n(t)$ is caused by the microphones themselves and by the Analogue Digital (AD) converters. It is supposed to be uncorrelated.
Furthermore, we want to introduce a signal model for a multipath environment. In general, reflections in a room can be described as attenuated and delayed versions of the original signal. All basic assumptions remain. Based on Equation (2.13) we add a term for a certain number of

reflections:

$$x_m(t) = \alpha_m s(t - t_l - \tau_{m,l}) + v_n(t) + \sum_{r=1}^{R} \alpha_{m,r} s(t - t_{l,r} - \tau_{m,l,r}) \tag{2.14}$$

with:

R	...	Number of reflections
$t_{l,r}$	...	time delay of the r-th path from source l to the reference sensor
$\alpha_{m,r}$	...	attenuation factor of the m-th microphone according to the r-th path

We assume that all array sensors have the same characteristics. This means that the varieties of gains and frequency responses in terms of magnitude and phase should ideally be zero. It is also supposed that the sensors are omni directional meaning their responses to any impinging signals are independent of their DOA. As we deal exclusively with acoustic signals the used microphones need to provide a capsule with an omni directional directivity pattern.
As mentioned earlier we assume that all impinging signals are plane waves. Depending on the array aperture and the distance between microphones and source this assumption can not always complied. A more detailed discussion can be found in Chapter 7.3.

2.3 Delay and Sum Beamformer

The D&S beamformer can be described by a general model of a narrow band beamformer. A D&S beamformer normally consists of two stages.

The first stage delays the sensor signals whereas the seconds sums up all the signals. Without the delaying stage the main steering direction would be zero degree. Delaying the incoming signals at first enables to adjust the main steering direction θ_0. Due to the fact of being a narrow band beamformer the output of the D&S beamformer is a function of the frequency. Optimal results in terms of attenuating interferences or rejecting noise can just achieved at a very narrow frequency range. Figure 2.4 shows the general structure of a narrow band beamformer:

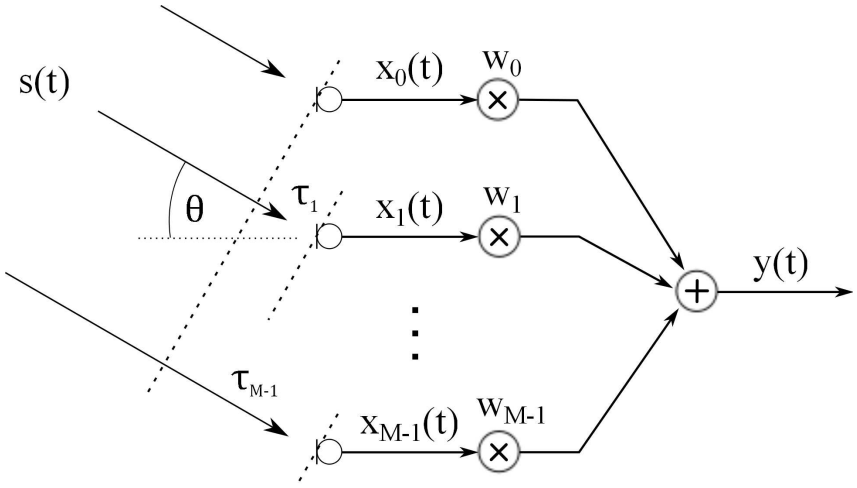

Figure 2.4: A general narrow band beamformer

Note that this model does not contain the delaying stage. The output $y(t)$ is a weighted sum of

the sensor signals, whereas w_m is the weighting factor according to the m-th signal:

$$y(t) = \sum_{m=0}^{M-1} x_m(t) w_m \tag{2.15}$$

In order to describe the frequency response of the beamformer in the following we define the complex sensor signal $x(t)$ for one source with a DOA angle θ. We assume a signal with the angular frequency ω and zero phase shift at the reference sensor:

$$x_0(t) = e^{j\omega t} \tag{2.16}$$

Hence, the signal at all sensors can be described by utilizing τ_m introduced in Equation (2.11):

$$x_m(t) = e^{j\omega(t-\tau_m)} = e^{j\omega t} e^{-j\omega\tau_m} \tag{2.17}$$

Applying Equation (2.17) to Equation (2.15) leads to:

$$y(t) = e^{j\omega t} \sum_{m=0}^{M-1} e^{-j\omega\tau_m} w_m \tag{2.18}$$

Excluding the temporal part we are able to define the beampattern $P(\omega, \theta)$:

$$P(\omega, \theta) = \sum_{m=0}^{M-1} e^{-j\omega\tau_m} w_m \tag{2.19}$$

The beampattern is a function of the temporal angular signal frequency ω and the DAO angle θ, contained by the time delay τ_m. Dealing with a discrete aperture brings along the problem of spatial aliasing. Comparable to temporal aliasing a certain frequency of the incoming signal should not be exceeded. The Nyquist criterion is also valid in terms of spatial sampling. For a more detailed explanation and several examples concerning this problem please see [Williams, 1999] and [Dollfuß, 2010]. We consider λ to be the smallest possible wavelength of the source signal. According to this we can set a suitable distance d.

$$d = \frac{\lambda_{min}}{2} \tag{2.20}$$

Defining the weighting vector $\mathbf{w}$ and the steering vector $\mathbf{d}(\omega, \theta)$ enables to rewrite Equation(2.19) in vector form:

$$P(\omega, \theta) = \mathbf{w}^T \, \mathbf{d}(\omega, \theta) \tag{2.21}$$

with:

$$\mathbf{w} = [\, w_0 \; w_1 \; w_2 \; \ldots \; w_{M-1} \,]^T \tag{2.22}$$

and:

$$\mathbf{d}(\omega, \theta) = [\, 1 \; e^{j\omega\tau_1} \; e^{j\omega\tau_2} \; \ldots \; e^{j\omega\tau_{M-1}} \,]^T \tag{2.23}$$

The weighting vector $\mathbf{w}$ can be seen as spacial window function. Regarding parameters such as main beam width or side lobe attenuation various spatial windows are possible. The very simplest case is the D&S beamformer with:

$$\mathbf{w} = \frac{1}{M} [\, 1 \;\; 1 \;\; \ldots \;\; 1 \,]^T \tag{2.24}$$

This conforms a rectangular window. In general, spatial windows conform temporal windows, which has been studied extensively in [Oppenheim et al., 2004]. Considering a window function in time it is useful to transfer the function from the time domain in the frequency domain via Fourier transformation. Transforming a rectangular leads to a sinc function. The results are the same if we transfer a spatial filter. The sinc function is to be seen in the beam pattern. The beam pattern (BP) is calculated as follows:

$$BP = 20log_{10} \frac{|P(\theta, \omega)|}{max\,|P(\theta, \omega)|} \; [dB] \tag{2.25}$$

As an example a beam pattern of a D&S beamformer with a main steering direction of zero degree and a spatial, rectangular window $\mathbf{w}$, defined in Equation (2.24), is shown in Figure 2.5.

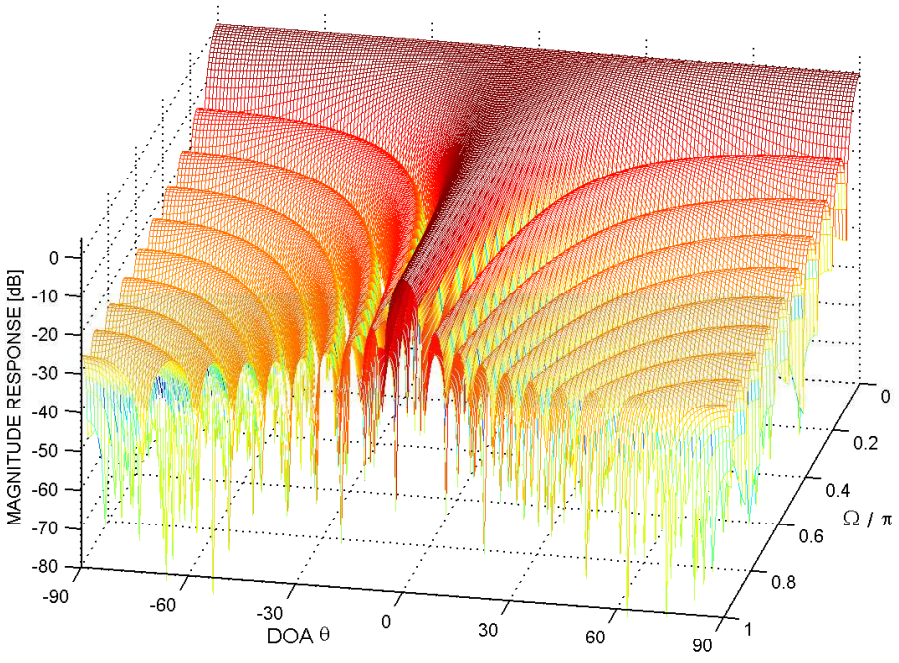

Figure 2.5: 3D beam pattern of a D&S beamformer; main steering direction: $\theta_0 = 0°$; number of sensors: $M = 19$

2.4 Frequency Invariant Beamformer

Unlike the D&S beamformer the Frequency Invariant Beamforming (FIB) technique is an array design for wideband signals. Ideally, the response is exclusively a function of the DOA angle of the impinging signal and, thus, not frequency dependent. However, it is practically not possible to design a frequency invariant beamformer valid for the whole spectrum. Consequently, the signals have to be restricted to the valid range.

Regarding a FIB also known as Filter and Sum (F&S) beamformer every sensor signal $x_m(t)$ is convoluted by a Finite Impulse Response (FIR) filter. The aim is to design a fixed set of filter coefficients (non-adaptive) in order to achieve frequency independence. Furthermore, the delaying stage is implemented by the filters, hence, a set of filter coefficients for each steering direction is required.

The method which is used here is proposed in [Liu and Weiss, 2008], [Sekiguchi and Karasawa, 2000]. To achieve frequency-independence we exploit the Fourier transform. The spatio-temporal distribution $P(\omega, \theta)$ of a 1D array, which is dependent on the DOA angle θ of the impinging signal and its frequency ω, can be characterized by a two dimensional discrete Fourier transformation. With the help of specific substitutions we are able to design a beam pattern dependent on normalized angular frequencies such as $P(\Omega_1, \Omega_2)$. If we can define an appropriate beam pattern it is possible to apply the 2D inverse Fourier transformation to gain the desired filter coefficients. A detailed description of the process and the filter design is given below.

2.4.1 Design Procedure

Figure 2.6 shows a wideband beamforming structure with M sensors and J filter taps.
Each real valued signal $x_m(n)$, with $m \in [0, 1, \ldots, M-1]$, sampled with a sampling period of T_s is processed by a FIR filter $\mathbf{w}_m$. Every individual filter of the m-th sensor contains J

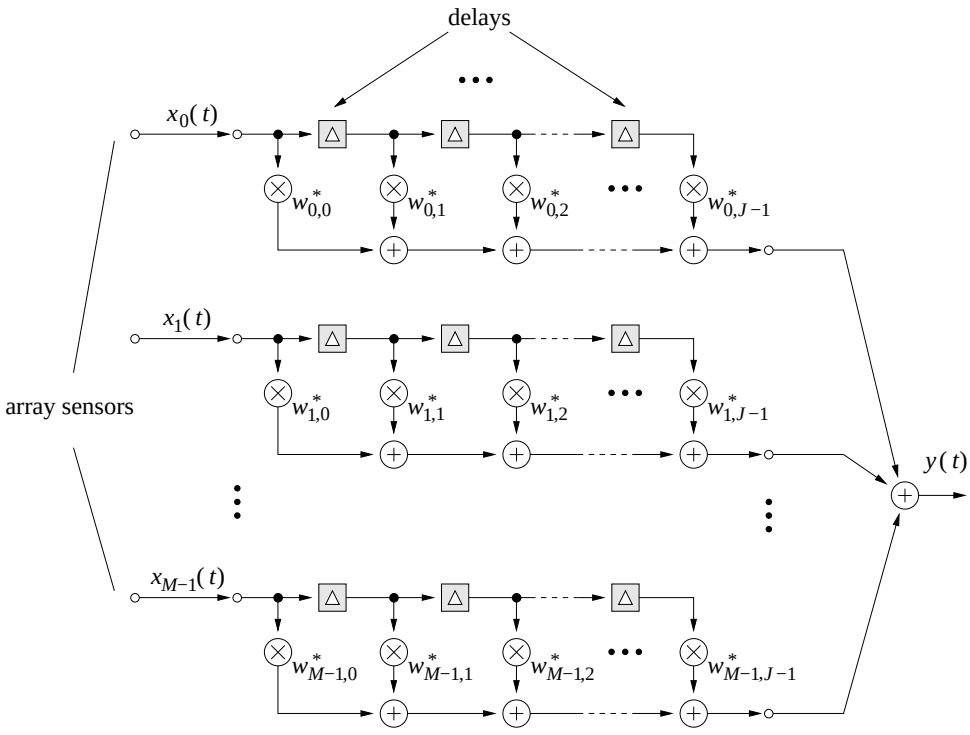

Figure 2.6: filter and sum beamformer
(with friendly permission of Dr Wei Liu)

coefficients $w_{m,j}$, which are also claimed to be real. These coefficients need to be calculated. The acquired 2D filter exceeds to a 3D filter as a 2D set for each main steering direction θ_0 has to be calculated. Criteria for the performance of these filters are for example side lobe attenuation, the valid frequency range but also error robustness and stability [Pape, 2005].

In order to avoid aliasing in the time domain the sampling frequency f_s should be larger then twice of the maximum frequency of interest f_{max}. To prevent spatial aliasing the distance between two microphones d should be less then half of the minimal wavelength of interest λ_{min} according to the maximal frequency. Finally we can denote two simple requirements:

$$2f_{max} < f_s = \frac{1}{T_s} \tag{2.26}$$

$$\lambda_{min} = \frac{c}{f_{max}} < 2d \tag{2.27}$$

Combining both conditions we set the sensor spacing d pursuant to the maximal signal frequency:

$$d = \frac{\lambda_{min}}{2} = cT_s \tag{2.28}$$

According to the signal model for a linear, equally spaced array in chapter 1.2 the beam response $P(\omega, \theta)$ from Equation (2.19) can be exceeded to:

$$P(\omega, \theta) = \sum_{m=0}^{M-1} \sum_{k=0}^{J-1} w_{m,k} \cdot e^{-jm\omega\tau} \cdot e^{-jk\omega T_s} \tag{2.29}$$

whereas the time delay between two adjacent microphones is:

$$\tau = sin\theta \cdot \frac{d}{c} \tag{2.30}$$

With consideration of the assumption being made in Equation (2.28) we define a constant μ, however, we are instantly able to simplify the problem:

$$\mu = \frac{d}{cT_s} = \frac{cT_s}{cT_s} = 1 \tag{2.31}$$

As a result we can rewrite Equation (2.29) as follows:

$$P(\Omega, \theta) = \sum_{m=0}^{M-1} \sum_{k=0}^{J-1} w_{m,k} \cdot e^{-jm\mu\Omega\ sin\theta} \cdot e^{-jk\Omega} \tag{2.32}$$

with the normalized angular frequency:

$$\Omega = \omega T_s \tag{2.33}$$

The next step is to apply a spectral transformation. To achieve a beam pattern $P(\Omega_1, \Omega_2)$, which is only dependent on normalized angular frequencies, we substitute as follows:

$$\Omega_1 = \mu\Omega sin\theta = \Omega sin\theta \tag{2.34}$$
$$\Omega_2 = \Omega \tag{2.35}$$

which yields to:

$$P(\Omega_1, \Omega_2) = \sum_{m=0}^{M-1} \sum_{k=0}^{J-1} w_{m,k} \cdot e^{-jm\Omega_2} \cdot e^{-jk\Omega_1} \tag{2.36}$$

Ω_1 represents the spatial normalized angular frequency, while Ω_2 stands for the temporal normalized angular frequency. Denote that the simplification of $\mu = 1$ which yields to a much easier equation is only possible as we set d according to the sampling frequency f_s as already done in Equation (2.28). Regarding the relationship: $\Omega_1 = \Omega_2\, sin\theta$ it can easily be seen that the impinging signals must comply the following expression:

$$|\Omega_1| \leq |\Omega_2| \tag{2.37}$$

The area for valid signals is located between the two lines $\Omega_1 = \Omega_2$ and $\Omega_1 = -\Omega_2$. The possible locations for a spatial temporal spectrum of an impinging signal can is depicted in Figure 2.9.
As we have to derive a proper beam pattern $P(\Omega_1, \Omega_2)$ later on, we are allowed to chose the remaining area arbitrarily as no signals occur. For a better understanding of how the 1-D and 2-D spectra are connected it is useful to pick out a few specific lines of a 2-D spectrum. Each line represents a 1-D spectrum according to a fixed angle θ being represented by the spatial angular frequency Ω_2.

As an example lets take a look at the line according to $\theta = 45\,\text{deg}$, which results to $\Omega_1 = 1/\sqrt{2}$ for a maximum temporal frequency $\Omega_1 = 1$. The magnitude of this line is the 1D spectrum for this specific angle. Indeed, it is reasonable to consider these lines as we want to obtain a

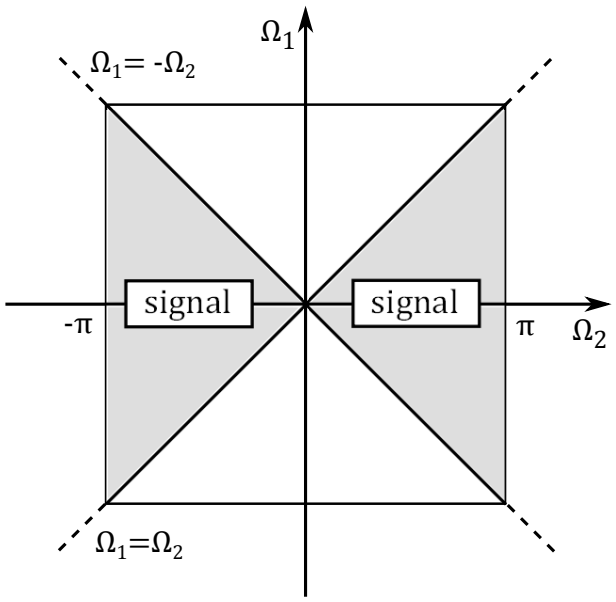

Figure 2.7: possible location of an impinging signal at the spatial temporal spectrum on the (Ω_1, Ω_2) plane

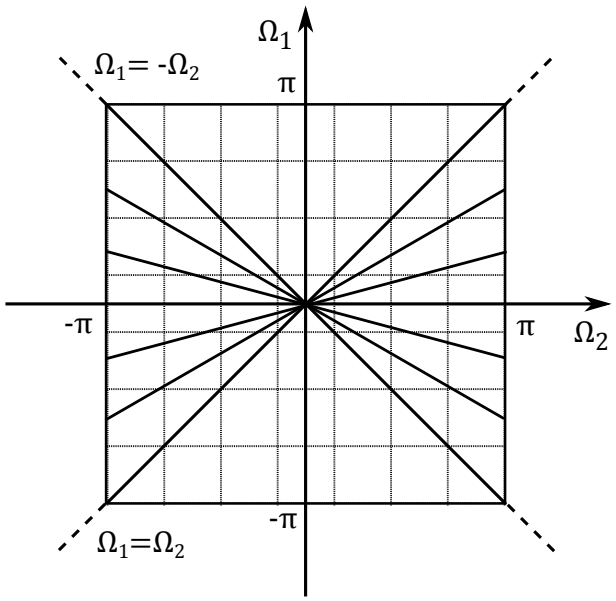

Figure 2.8: Relationship between 1D and 2D frequency; $d = c/f_s$; main steering direction $\theta_0 = 0$

frequency invariant beamformer. The ideal 2D pattern for a 100 per cent frequency invariant beamformer would exclusively contain straight lines of a constant magnitude.

Unfortunately, we are not able to achieve the same performance for lower frequencies as it is possible for higher frequencies. It is already proven that a design approach, treating all frequencies equally yields to a non robust beamformer [Pape, 2005]. We can consider this in our design process by using a low pass filter as prototype function in order to build $P(\Omega_1, \Omega_2)$. Thus, the beamformer can not gain the optimal results below a certain frequency.

2.4.2 Applying the Window Method

To make the design procedure more comprehensible it is useful to outline the following three steps in detail. At first we have to find a beam pattern $P(\Omega_1, \Omega_2)$ which is achievable in reality. Having found an appropriate pattern we can apply the inverse Fourier transform which finally yields to the desired coefficients.

1D Prototype FIR Filter

The first task is to design a 1D prototype narrow band zero phase FIR low pass filter with a filter length of L. For simplicity we chose L to be odd. Furthermore, L is crucial as the overall number of all lobes is determined directly by L. The frequency response $P(\Omega)$ is expressed as:

$$P(\Omega) = p(0) + \sum_{l=1}^{(L-1)/2} p(l)\, cos(2\pi l \Omega) \tag{2.38}$$

We set our prototype low pass filter function to:

$$p(l) = 1/L \tag{2.39}$$

where L is the length of the prototype. Conducting the simulations L is either set to $L = 5$ or $L = 7$.

Transform the 1D Filter in a 2D Filter

Next, we have to apply the transformation to the derived frequency response. The main steering direction should be considered at this point. Each main direction leads to an own 2D filter. The 2D pattern can be calculated if Ω in Equation (2.38) is replaced by T_D as follows:

$$P(\Omega_1, \Omega_2) = p(0) + \sum_{l=1}^{(L-1)/2} p(m) cos(2\pi l T_D) \tag{2.40}$$

with:

$$T_D = \frac{\Omega_2}{\Omega_1} - sin\theta_0 \tag{2.41}$$

Denote, that in the area of $|\Omega_1| \geq |\Omega_2|$ the values can be chosen arbitrary. For this area we can set a fixed value [Sekiguchi and Karasawa, 2000]:

$$T_D = 1 - sin\theta_0 \tag{2.42}$$

Figure 2.9 and Figure 2.10 show two different 2D prototypes for different main steering directions θ_0 and different side lobe numbers L.

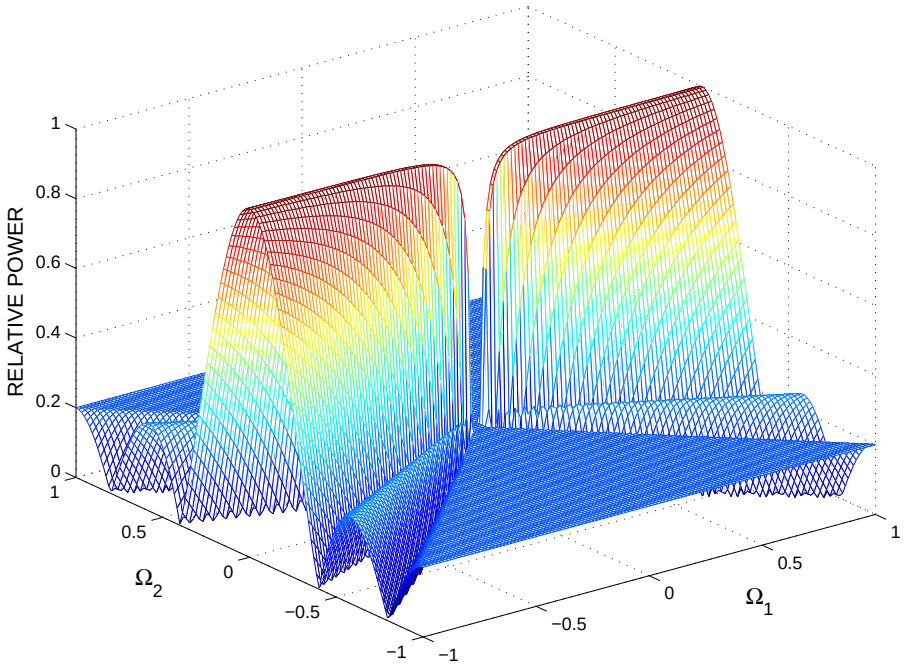

Figure 2.9: 2D frequency response of prototype filter $P(\Omega_1, \Omega_2)$; main steering direction $theta_0 = 0^\circ$; $L = 5$

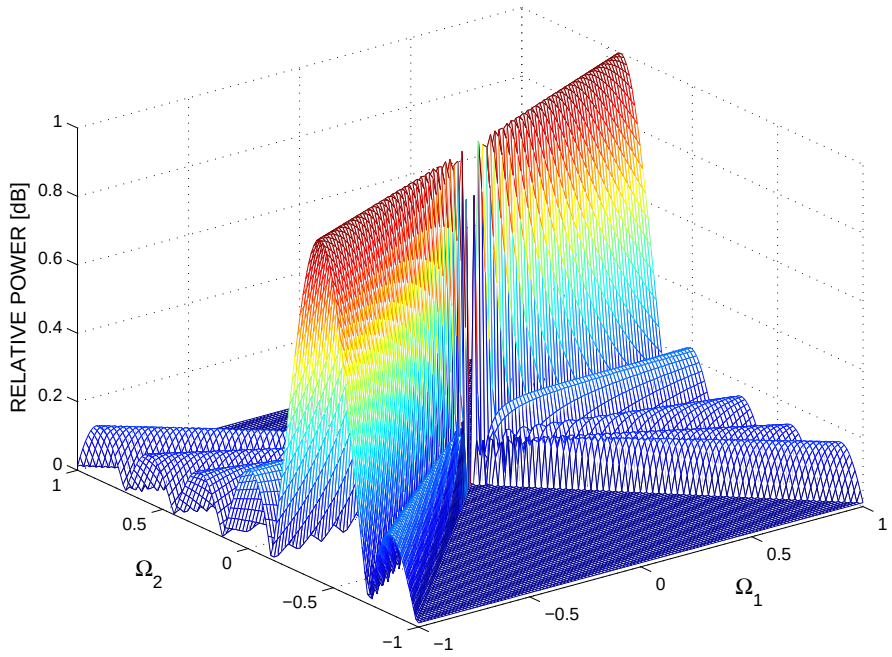

Figure 2.10: 2D frequency response of prototype filter $P(\Omega_1, \Omega_2)$; main steering direction $theta_0 = -25^\circ$; $L = 7$

Fourier transformation

As we have finally derived $P(\Omega_1, \Omega_2)$ we can now apply the two dimensional inverse discrete Fourier transformation of our desired beam pattern. For the desired beam pattern we need to choose a resolution which affects the maximal sensor number as well as the number of filter taps. Setting the resolution to [128 × 128] leads to a maximal number of filter taps J and sensors M: $J = M = 128$, which has to be truncated according to requirements of the beamformer. Note that it is necessary to apply a shift of the FFT results.

The resulting filter coefficients, computed with a different main steering direction θ_0 are displayed in the Figure 2.11 and Figure 2.12. Both filters are truncated to a dimension of [19 × 19].

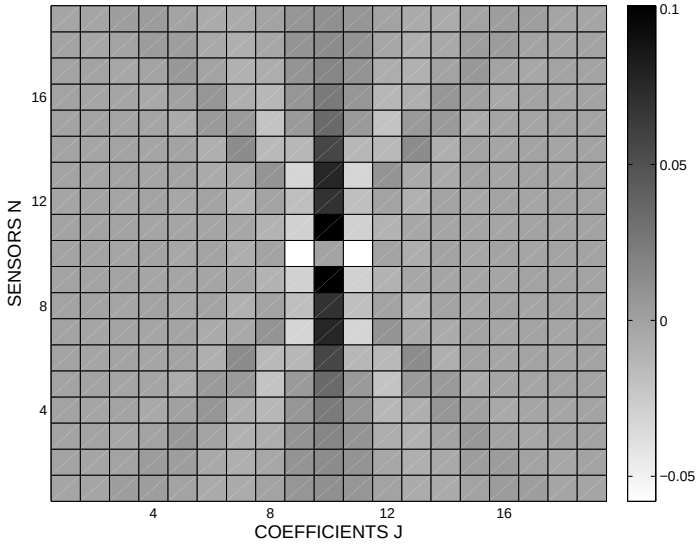

Figure 2.11: Final filter coefficients with: $M = 19$; $J = 19$; $L = 5$; $\theta_0 = 0°$

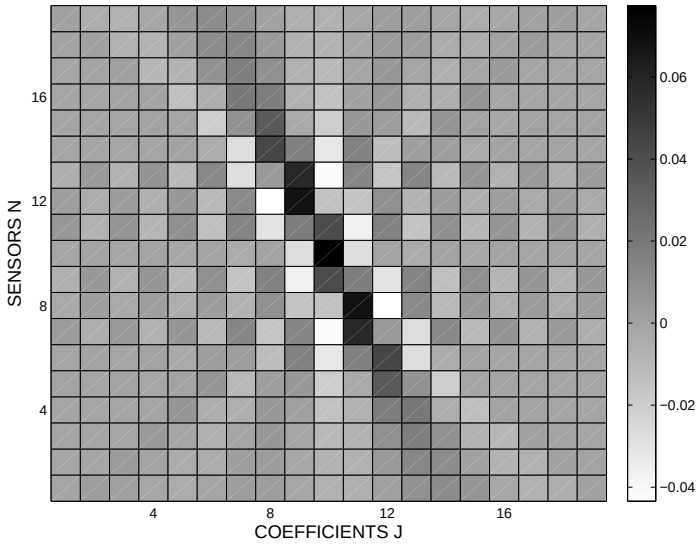

Figure 2.12: Final filter coefficients with: $M = 19$; $J = 19$; $L = 7$; $\theta_0 = -25°$

2.4.3 Simulations

The following simulation serves as a basis for the choosing the parameters utilized in the practical approach later on. Some simulation results of the designed FIB are shown in order to compare significant parameters. Its influence of the beamformers performance is evaluated and illustrated. All shown results refer to the former suggested method and are implemented in Matlab. Different possibilities of depicting a beam pattern are introduced.

Even if different beam patterns are produced, the method of simulation is basically the same. For every possible DOA angle and every possible frequency a set of data respectively source signals are produced and applied to the beamformer. By measuring the RMS power of the all resulting signals we are able to depict a beam pattern by calculating the ratio. As we apply this process step by step for different DOA angles (with a resolution of one degree) and different frequencies (with changing resolution) the pattern is a function of DAO and frequency. Hence, a three dimensional representation is useful.

As source signal a sinusoidal signal with a certain frequency is utilized. To comply the requirement of a certain number M of microphone signals the DOA angle is simulated by delaying each channel signal by the according delay $\tau_{m,l}$.

A common way to present the results is a 3D plot where x axis and y axis are frequency and DOA angle whereas the z axis depicts the magnitude. Such a 3D plot can be seen in Figure 2.13 Figure 2.18 and Figure 2.19. A flat version of this plot is possible when a colored representation for the z axis is used as shown in Figure 2.16 and Fig 2.17 .

If we want illustrate a single beam pattern for one frequency a 2D plot can be chosen, representing a pattern for a specific frequency per line. Such a pattern can be seen in Figure 2.15 and Figure 2.14. It is also possible to utilize broadband signals, such as bandpass noise, as source in order to combine several frequencies if they are supposed to cause the same pattern. This is conducted in experiment Ia in the anechoic chamber (chapter 7.1).

Fig 2.13 shows a 3D pattern for a number of sensors $M = 19$, a number of filter taps $J = 121$, a side lobe number of $L = 5$ and a main steering direction of $\theta_0 = 0°$.

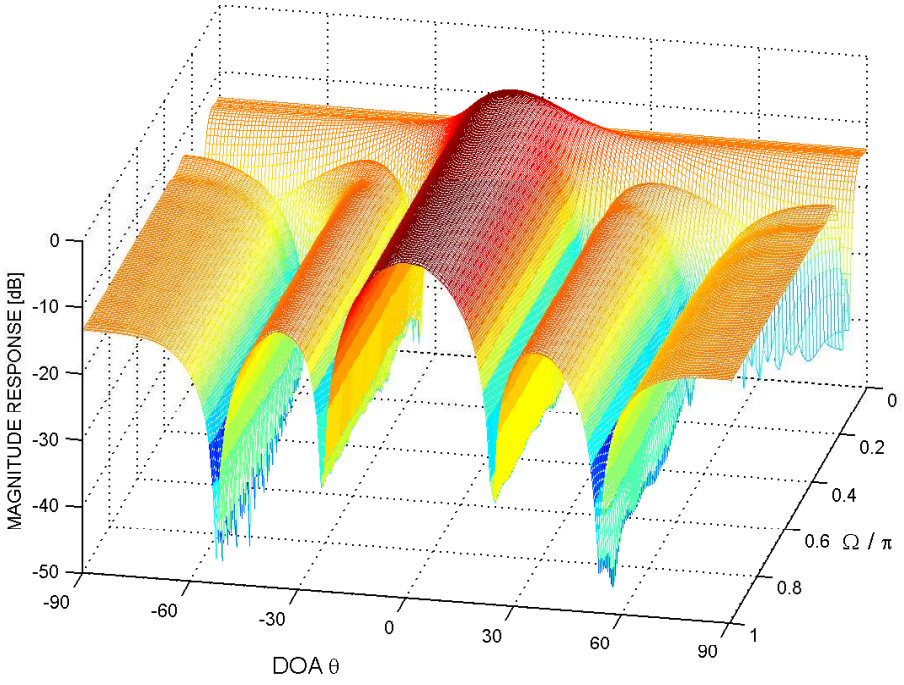

Figure 2.13: 3D Beam pattern: $M = 19$; $J = 121$; $L = 5$; $\theta_0 = 0°$

2D Pattern Considering the Number of Microphones M and Different Normalized Frequencies Ω with a Constant Filter Length J

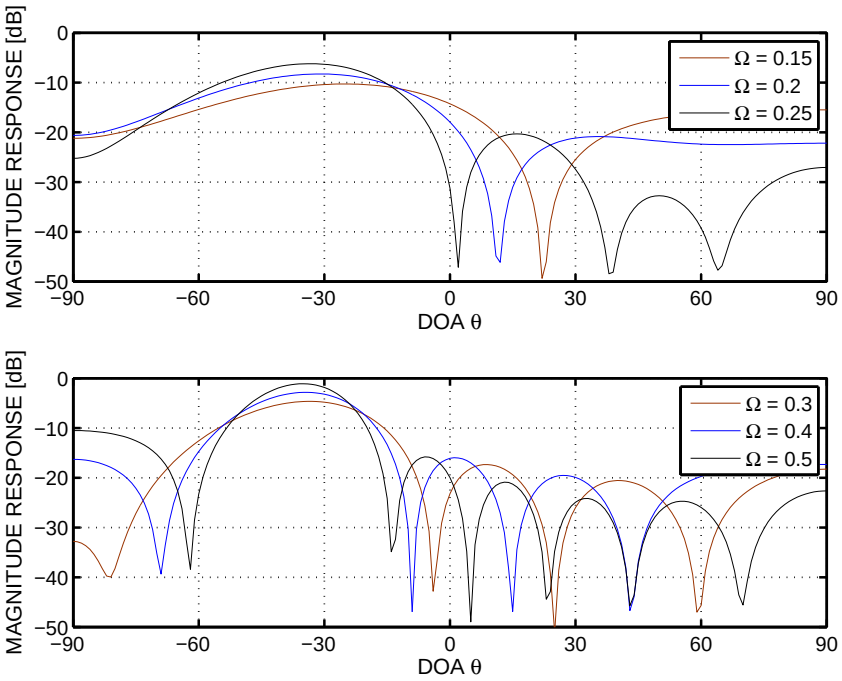

Figure 2.14: 2D Beam pattern; 6 different frequencies $0.15 \leq \Omega \leq 0.5$*;* $M = 11$*;* $J = 121$*;* $L = 7$*;* $\theta_0 = -34°$

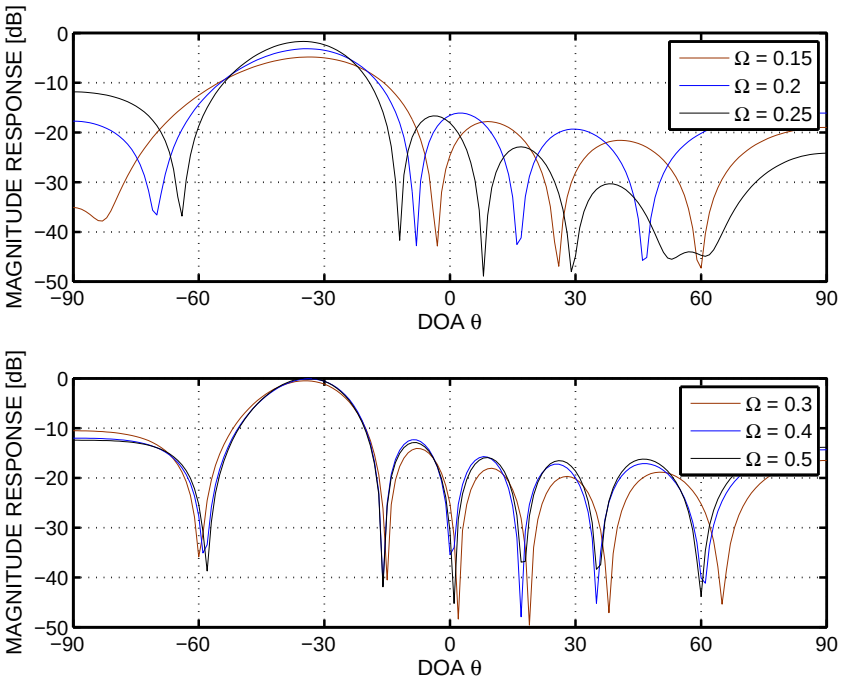

Figure 2.15: 2D Beam pattern; 6 different frequencies $0.15 \leq \Omega \leq 0.5$*;* $M = 19$*;* $J = 121$*;* $L = 7$*;* $\theta_0 = -34°$

Flat 3D Pattern Considering the Number of Filter Taps J with a Constant Microphone Number M and Same Steering Direction θ_0

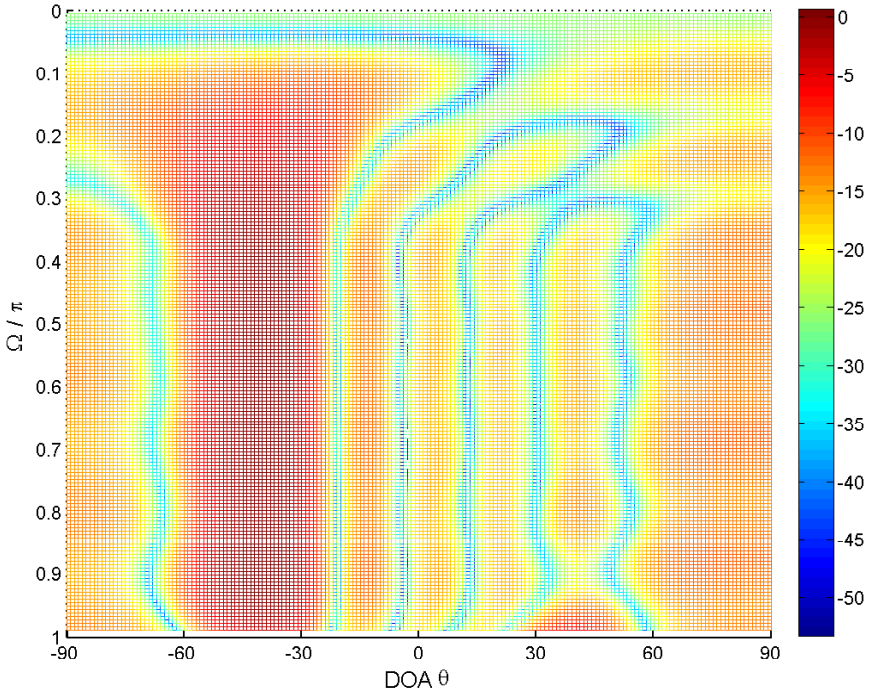

Figure 2.16: Flat 3D beam pattern; $M = 19$; $J = 19$; $L = 7$; $\theta_0 = -39°$

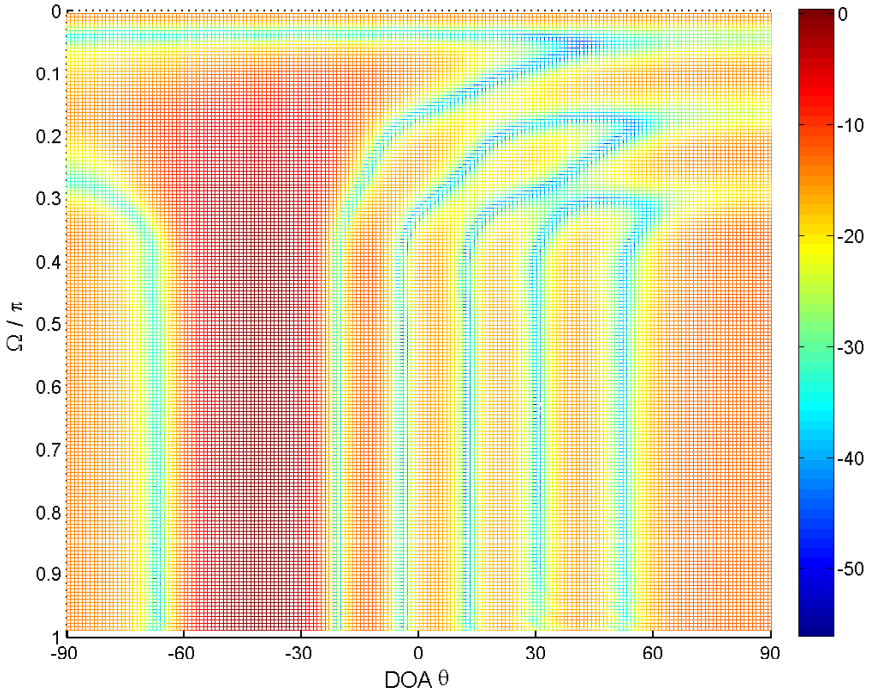

Figure 2.17: Flat 3D beam pattern; $M = 19$; $J = 121$; $L = 7$; $\theta_0 = -39°$

3D Pattern with Parameters Chosen in the Practical Approach Compared to a Pattern with a Lower Number of Microphones and Filter Taps

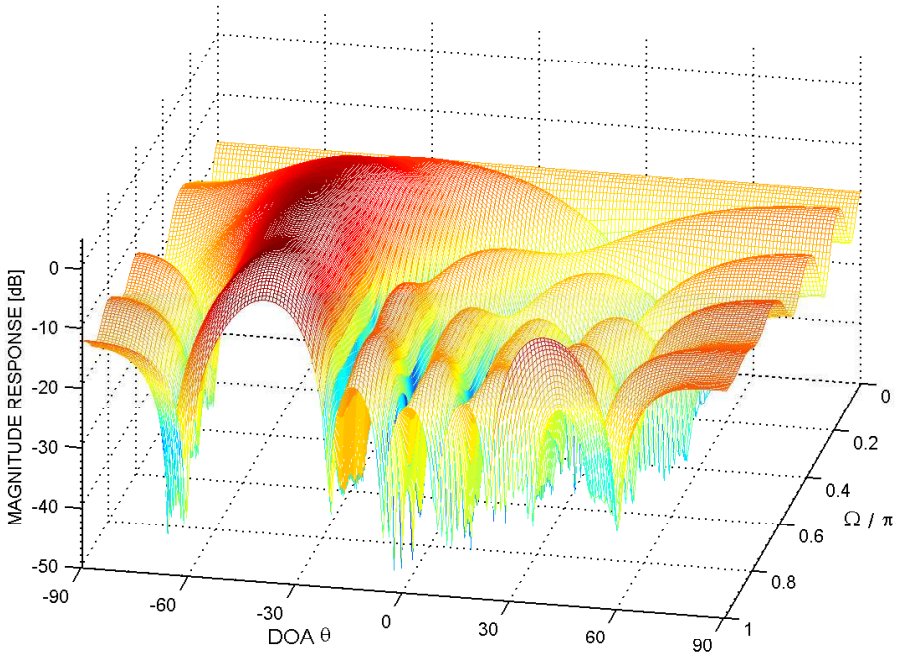

Figure 2.18: 3D beam pattern with insufficient performance: $M = 11$; $J = 19$; $L = 7$; $\theta_0 = -37°$

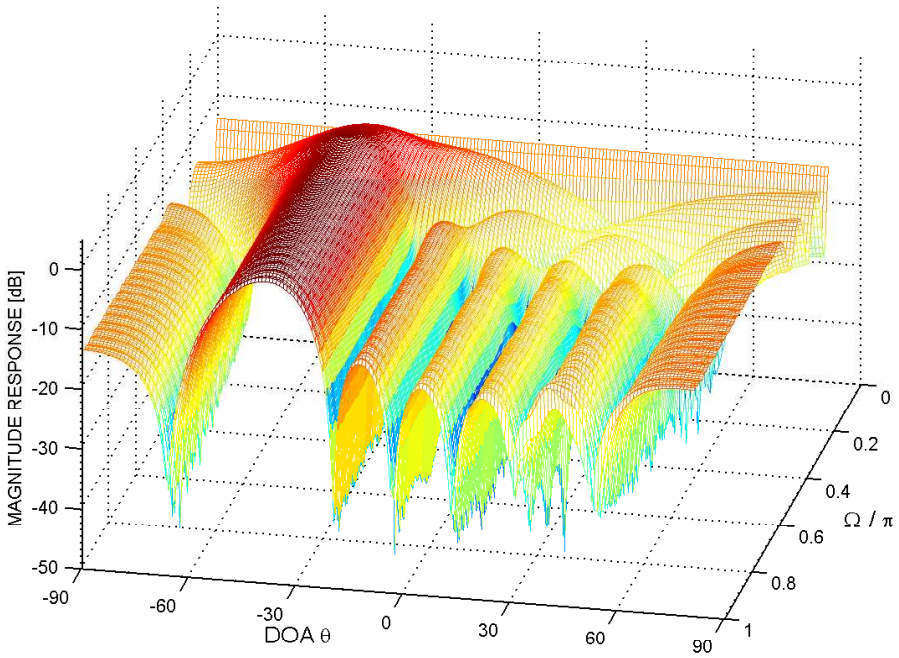

Figure 2.19: 3D beam pattern with parameters chosen for the practical approach: $M = 19$; $J = 121$; $L = 7$; $\theta_0 = -37°$

2.4.4 Conclusion

The former results illustrate the impact of three parameters:

Filter length of the 1D prototype L ... respectively the side lobe number $L-1$

Number of microphones M

Number of filter taps J

As already mentioned in Chapter 2.4.2 the choice of the filter length L of the 1D prototype directly determines the number of side lobes. A certain number is needed as we want to cancel several interfering signals. This will be discussed later in Chapter 4 as well as in Chapter 8.2.3. For now, a number of $L = 7$ is chosen.

The Number of microphones M is evaluated in Figure 2.14 and Figure 2.15. It can be seen that an increasing number of microphones M extends the frequency range where the beamformer is working frequency independently. Hence, the maximal, practically available number is chosen: $M = 19$.

The Number of filter taps J is considered in Figure 2.16 and Figure 2.17. It is shown that a smaller number of filter taps J has an impact of the beamformers performance in the valid frequency range as the main lobe as well as some side lobes are distorted. Therefore, a sufficiently high number of filter taps is chosen: $J = 121$.

Concluding with Figure 2.19 a simulation with the finally chosen parameters is conducted: $L = 7$, $M = 19$ and $J = 121$.

2.4.5 Fractional Delay

When dealing with beamforming the topic of delaying signals is important. For example, it is possible to change the steering direction of a beamformer by adding appropriate delays (Chapter 2.3). Furthermore, it is essential for any kind of simulation. A filter delaying a signal by an integer number of samples is defined as follows:

$$h[n] = \delta[n - n_0] \tag{2.43}$$

Utilizing exclusively integer delays is far and away insufficient. Therefore, some considerations are given to assemble a fractional delay FIR filter. As a result we want to gain a filter which delays the signal by convoluting it with this filter.

Basically, the approach works like a conversion from a digital signal to an analogue one, or more general, from a discrete to a continuous one. Considering the discrete signal $x[n]$ we interpolate as exactly as possible. Afterwards we pick out the specific points according to the required decimal delay. There are many ways to accomplish an interpolation of a signal. The most exact one can be achieved by using a sinc function. Hence, we take a look at a model of an ideal reconstruction system.
The input is the discrete signal $x[n]$ which we want to delay by a fractional number of samples. As an output the filter yields the reconstructed, continuous signal $x_c(t)$. Intermediately we obtain the weighted, periodic impulse train $x_s(t)$, which is already a continuous signal. As reconstruction filter we are using an ideal low pass filter with a frequency response $H_r(j\omega)$ and an impulse response $h_r(t)$. The output of the system can be described as follows:

$$x_c(t) = \sum_{n=-\infty}^{\infty} x[n] h_r(t - nT) \tag{2.44}$$

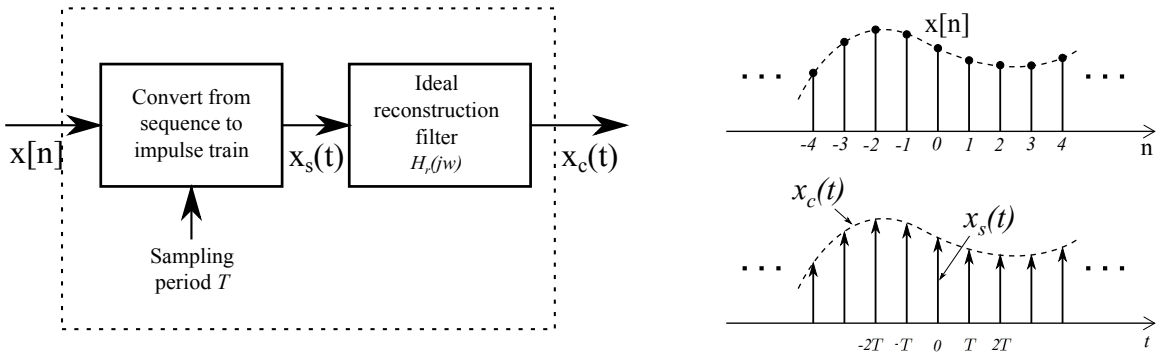

Figure 2.20: Ideal reconstruction system with discrete input $x[n]$, weighted impulse train $x_s(t)$ and continuous signal $x_c(t)$

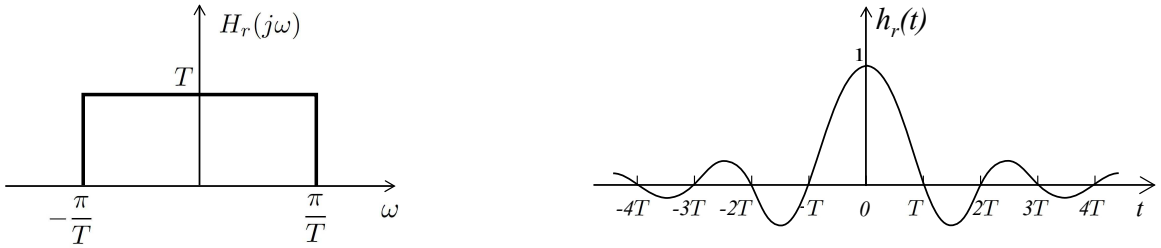

Figure 2.21: reconstruction filter in frequency domain $H_r(j\omega)$ and time domain $h_r(t)$

The frequency response of the ideal low pass filter is shown below. It is common to choose $\omega_c = \omega_s/2 = \pi/T$ with regard to aliasing. For simplicity the filter has a gain of T. If we apply the inverse Fourier transform to the given frequency response $H_r(j\omega)$ we obtain the impulse response $h_r(t)$, which is generally known as sinc function.

$$H_r(j\omega) = \begin{cases} T & |\omega| \leq \pi/T \\ 0 & |\omega| \geq \pi/T \end{cases} \tag{2.45}$$

$$h_r(t) = \frac{sin(\pi t/T)}{\pi t/T} \tag{2.46}$$

Applying the ideal low pass filter of Equation (2.46) in Equation (2.44) yields to:

$$x_c(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{sin(\pi(t-nT)/T)}{\pi(t-nT)/T} \tag{2.47}$$

This equation is the result of an ideal reconstruction. We are now able to calculate the value x for every point in time t which obviously includes points between the original samples. Utilizing the former consideration we rewrite this equation and describe the reconstructed signal $x_r(t)$.

$$x_r(t) = \sum_{k=-\infty}^{\infty} x[k] \frac{sin(\pi(t-kT)/T)}{\pi(t-kT)/T} \tag{2.48}$$

Furthermore, we want to find a solution of the following problem. $h[n]$ is demanded whereas

$y[n]$ shall be the final fractionally delayed signal.

$$y[n] = x[n] * h[n] \tag{2.49}$$

If we delay $x_r(t)$ it corresponds to $y[n]$. To fit this requirement we delay $x_r(t)$ about αT whereas α is the fractional delay factor with $0 \leq \alpha \leq 1$.

$$x_r(t - \alpha T) = \sum_{k=-\infty}^{\infty} x[k] \frac{sin(\pi(t - \alpha T - kT)/T)}{\pi(t - \alpha T - kT)/T} \tag{2.50}$$

With the definition of $t = nT$ we finally replace $x_r(t - \alpha T)$ by $y[n]$.

$$y[n] = x_c(t - \alpha T)|_{t=nT} \tag{2.51}$$

After a few simplifications we accomplish our calculations. As a result we obtain $h[n]$.

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{sin(\pi(nT - \alpha T - kT)/T)}{\pi(nT - \alpha T - kT)/T} \tag{2.52}$$

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{sin(\pi(n - \alpha - k))}{\pi(n - \alpha - k)} \tag{2.53}$$

$$y[n] = x[n] * h[n] \tag{2.54}$$

$$h[n] = \frac{sin(\pi(n - \alpha))}{\pi(n - \alpha)} \tag{2.55}$$

Using the filter $h[n]$ with infinite length the results will be a perfect and frequency independent interpolation. The necessary truncation of the filter leads to inaccuracies especially at higher frequencies near $\omega_c/2$ also called the Gibbs phenomenon. For further information please see [Oppenheim et al., 2004]. Effectively this will not be a big issue if the number of taps is chosen high enough. The simulations of the beamformer are conducted with $M = 101$ which does not cause any problems.

3 Blind Source Separation by Independent Component Analysis

3.1 Introduction

According to Hyvärinen the research field of Independent Component Analyses (ICA) is very closely related to Blind Source Separation (BSS), also known as Blind Signal Separation. ICA is supposed to be the most widely used method in BSS [Hyvärinen and Oja, 2000]. Hence, it does not seem useful to strictly separate this two disciplines.

In this chapter, the basic idea of BSS and the corresponding mathematic model is shown. Furthermore, the natural gradient approach is presented shortly, although, it is not derived in this thesis.

The goal of BSS is to separate or to recover a number of original sources, which has been mixed in an unknown way. The process is called blind as there is very few a priori knowledge and just weak assumption on the original signals. A source in the BSS case means an independent component such as a speaker in a cocktail party situation. The mixing process is described by a mixing matrix. As well as the source signals the mixing matrix is unknown and needs to be estimated in order to recover the original signal. Therefore, we have to consider certain conditions. Furthermore, the restrictions and ambiguities of the approach are shown.

3.2 Model

We assume a number of simultaneously speaking persons L and a number of sensors M. We determine that the number of sources and the number of microphones is equal:

$$M = L \tag{3.1}$$

For simplicity we choose three as random number to be the amount of sources and speakers. The examples in this chapter will refer to this number. Note that this could be any number larger than one and is chosen to explain the problem in a simple and comprehensible way.

Therefore, the original source signal is denoted by $s_0(t), s_1(t)$ and $s_2(t)$ whereas the sensor signals are $x_0(t), x_1(t)$ and $x_2(t)$. Each signal $x(t)$, arriving at a microphone, is a mixture of source signals which are individually weighted by a factor $\alpha_{m\,l}$. The set of linear equations can be expressed as follows:

$$x_0(t) = \alpha_{00}s_0(t) + \alpha_{01}s_1(t) + \alpha_{02}s_2(t) \tag{3.2}$$
$$x_1(t) = \alpha_{10}s_0(t) + \alpha_{11}s_1(t) + \alpha_{12}s_2(t) \tag{3.3}$$
$$x_2(t) = \alpha_{20}s_0(t) + \alpha_{21}s_1(t) + \alpha_{22}s_2(t) \tag{3.4}$$

As example the mixing and demixing process is shown in the following figures. The independent source signals $s_0(t), s_1(t)$ and $s_2(t)$ in Figure 3.1 are multiplied by the unknown mixing matrix A which leads to the observed signals $x_0(t), x_1(t)$ and $x_2(t)$ in Figure 3.2. The estimated signals $\hat{s}_0(t), \hat{s}.1(t)$ and $\hat{s}_2(t)$ are depicted in Figure 3.3.

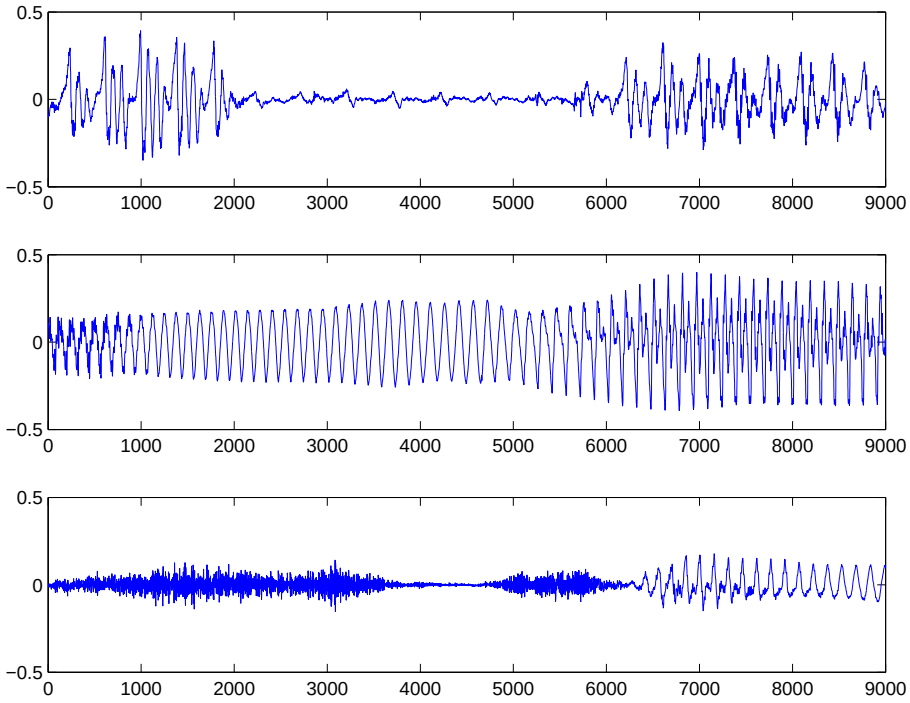

Figure 3.1: original source signals s_1, s_2 and s_3 of an BSS Instantaneous Mixing Problem

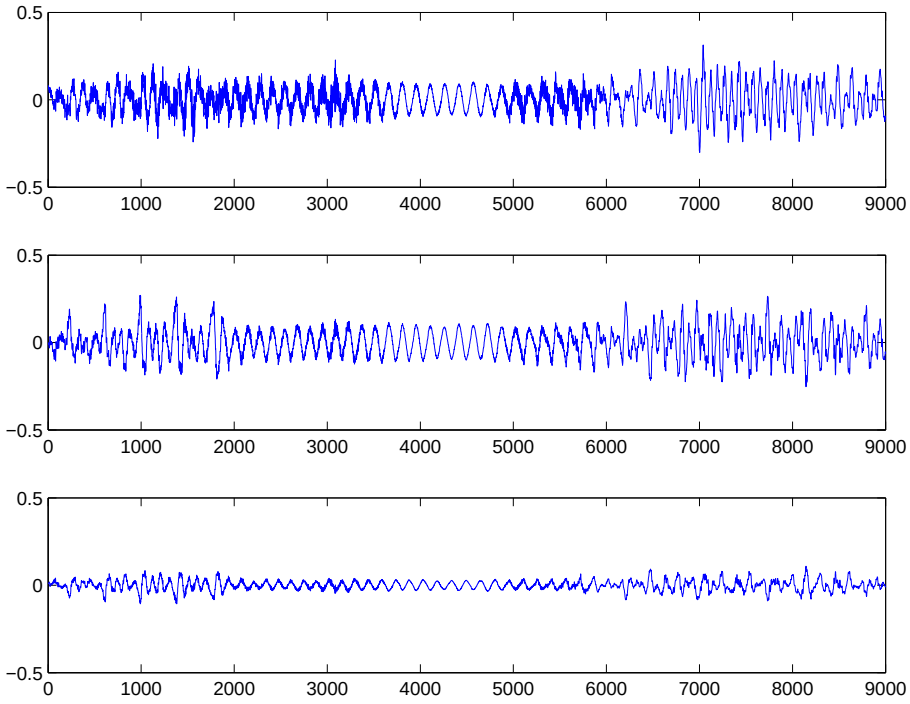

Figure 3.2: mixed signals x_1, x_2 and x_3 of an BSS Instantaneous Mixing Problem

Remember that $x_0(t), x_1(t)$ and$x_2(t)$ are the only available signals in order to recover $\mathbf{s}$. Despite from signal order and amplitudes (see Ambiguities) it is possible to recover the signals.

A general notation for a certain number of sources L and microphones M whereas $M = L$

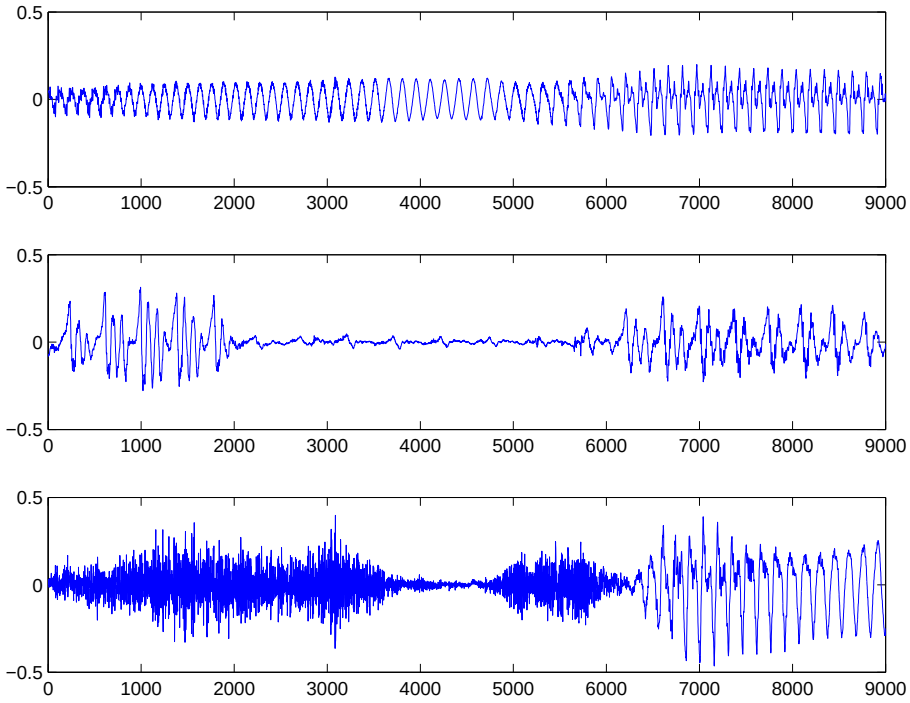

Figure 3.3: recovered source signals $\hat{s}_1$, $\hat{s}_2$ and $\hat{s}_3$ of an BSS Instantaneous Mixing Problem

can be given as follows:

$$x_m(t) = \alpha_{m0} s_0(t) + \alpha_{m1} s_1(t) + \ \ldots \ + \alpha_{ml} s_l(t) \tag{3.5}$$

Note that this is a simplified model where any kind of time delays or additional noise is omitted. In order to rewrite the former equations in matrix notation we need to drop the time index t. This is possible because the mixture x_m as well as the independent source signal s_l are supposed to be random variables. As this model neglects any time delays it is usually called the instantaneous mixing model:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{3.6}$$

comprised of:

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} \alpha_{00} & \alpha_{01} & \ldots & \alpha_{0l} \\ \alpha_{10} & \alpha_{11} & \ldots & \alpha_{1l} \\ \vdots & & \ddots & \vdots \\ \alpha_{m0} & \alpha_{m1} & \ldots & \alpha_{ml} \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_l \end{pmatrix} \tag{3.7}$$

As already shown our goal is to recover the original signals $\mathbf{s}$ only by knowing $\mathbf{x}$. The problem can not be solved directly as we do not know neither the weighting matrix $\mathbf{A}$ nor the source signals $\mathbf{s}$. These variables are called latent variables as they can not be directly observed. $\mathbf{A}$ is also known as mixing matrix. If we can find an inverse of the matrix $\mathbf{A}$ we can multiply this inverse by $\mathbf{x}$ and hence recover $\mathbf{s}$. The task is to find the inverse of the mixing matrix: $\mathbf{A}^{-1}$ often called demixing matrix and denoted by $\mathbf{D}$.

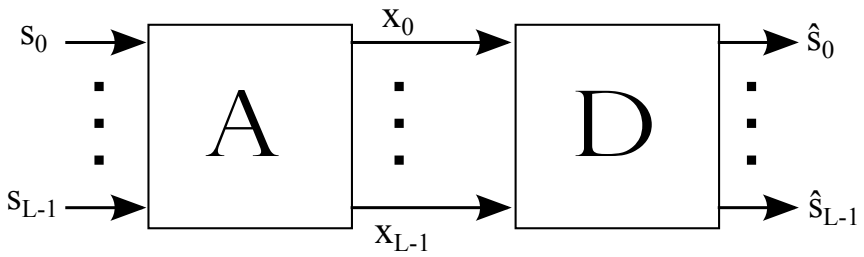

Figure 3.4: Instantaneous Mixing Problem with mixing matrix A and demixing matrix D

The calculated demixing matrix is always an estimate, thus, we define the estimated recovered signal vector $\hat{\mathbf{s}}$. The flow chart of the mixing and demixing process is shown in Figure 3.4.

3.2.1 Restrictions

To make sure that an algorithm, such as the natural gradient algorithm, can successfully be applied we need to make some restrictions and assumptions.

All sources must statistically be independent.

Basically this means that random variables $y_0, y_1, \ \dots \ y_{N-1}$ are declared to be independent if the information given by the value y_i does not contribute any information on a different value y_j for $i \neq j$. A common way to define independence can be done with help of the Probability Density Function (PDF). We consider the joint PDF of y_i by $p(y_0, y_1, \ \dots \ y_{N-1})$ and the marginal PDF of a specific signal by $p_i(y_i)$. Independence is given if the joint PDF can be factorized by all marginal PDF:

$$p(y_0, y_1, \ \dots \ y_{N-1}) = p_0(y_0)p_0(y_0) \ \dots \ p_{N-1}(y_{N-1}) \tag{3.8}$$

The distribution of all sources must be non Gaussian.

To apply ICA we need to rely on higher order statistics. As we will show in Chapter 3.3 the third and fourth cumulant for a Gaussian distribution is zero. This information is essential. Further, if the observed signals do not contain any higher order statistics it is impossible to separate the signals. Note that it is not required to know the exact distribution, although this would considerably simplify the problem. This restriction becomes more obvious when the mixing process for different distributions such as Gaussian and super Gaussian signals is regarded [Hyvärinen and Oja, 2000].

The mixing matrix has to be squared.

From Figure 3.4 can be seen that the optimal solutions for the demixing matrix $\mathbf{D}$ would be an inverted matrix $\mathbf{A}^{-1}$. This requires not just a squared matrix $\mathbf{A}$ but also invertibility. This assumption simplifies our model although we have to find a solution for the non squared case later on. There are two cases: the number of sources is higher then the number of microphones $L > M$ or the number of microphones is higher then the number of sources: $M > L$. In our approach we have to deal mostly with case number two. There are different possibilities to solve the problem, which will we discussed later in Chapter 4.4.

3.2.2 Ambiguities

There are two ambiguities of ICA. It is important to consider them when implementing the algorithm as this will affect but also restrict the algorithm.

The signal energy can not be determined.

The problem is that neither $\mathbf{A}$ nor $\mathbf{s}$ is known. Introducing an additional random scalar γ_l we scale the source vector $\mathbf{s}$ while dividing every column of $\mathbf{A}$ by the corresponding γ_l. Although the variance of $\mathbf{s}$ is changed the same result is provided. Based on Equation (3.6) the problem can be noted as follows:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{3.9}$$

introducing the scalar γ_l:

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} \frac{1}{\gamma_0}\alpha_{00} & \frac{1}{\gamma_1}\alpha_{01} & \dots & \frac{1}{\gamma_l}\alpha_{0l} \\ \frac{1}{\gamma_0}\alpha_{10} & \frac{1}{\gamma_1}\alpha_{11} & \dots & \frac{1}{\gamma_l}\alpha_{1l} \\ \vdots & & \ddots & \vdots \\ \frac{1}{\gamma_0}\alpha_{m0} & \frac{1}{\gamma_1}\alpha_{m1} & \dots & \frac{1}{\gamma_l}\alpha_{ml} \end{pmatrix} \cdot \begin{pmatrix} \gamma_0\, s_0 \\ \gamma_1\, s_1 \\ \vdots \\ \gamma_l\, s_l \end{pmatrix} \tag{3.10}$$

Hence, there is a need in constraining the signal energy in our approach. For example, this can be done by a simple normalization of the recovered signals.

The order of the independent components can not be determined.

Again, we are facing the problem that neither $\mathbf{A}$ nor $\mathbf{s}$ is known. It is possible to freely change the order of the independent sources. Any of the sources can considered to be first. We can easily change the order of sources $\mathbf{s}$ without affecting $\mathbf{x}$. As an example, referring again to Equation (3.6), we interchange s_0 and s_1 while shifting the corresponding columns of $\mathbf{A}$:

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} \alpha_{01} & \alpha_{00} & \dots & \alpha_{0l} \\ \alpha_{11} & \alpha_{10} & \dots & \alpha_{1l} \\ \vdots & & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m0} & \dots & \alpha_{ml} \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_0 \\ \vdots \\ s_l \end{pmatrix} \tag{3.11}$$

Having explained the basic idea of BSS respectively ICA with all its assumptions and ambiguities we need to find an algorithm in order to estimate the demixing matrix $\mathbf{D}$ referring to figure 3.4. The algorithm should work reliably and fast. At this point we want to introduce the natural gradient algorithm.

3.3 Natural Gradient Algorithm

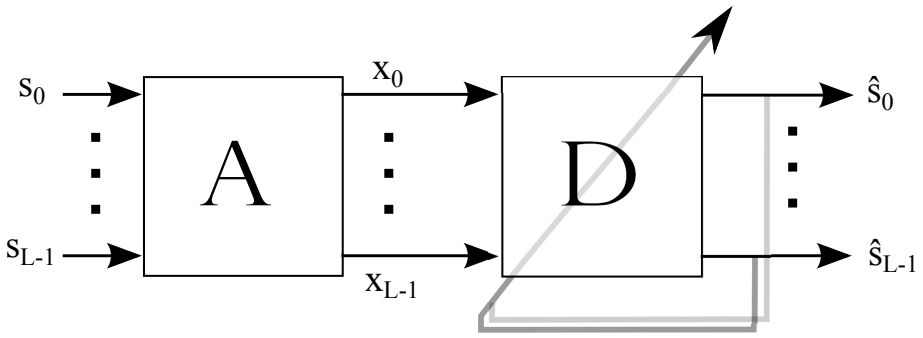

Figure 3.5: Instantaneous Mixing Problem with mixing matrix A and adaptive demixing matrix D

The Natural Gradient Algorithm is a standard learning algorithm for adaptive blind source separation (BSS) and independent component analyses (ICA). The algorithm is supposed to solve the former explained demixing problem. It can be allocated to Maximum Likelihood (ML) estimation which is a fundamental method of statistical estimation. The parameter values with the highest possibility for the observation are estimated. Those are the results of the algorithm. Regarding Figure 3.5 it can be seen that the best solution for D would be the inverse Matrix of A.

$$\mathbf{D} \stackrel{!}{=} \mathbf{A}^{-1} \tag{3.12}$$

The adaptive estimation process of D is defined as follows. A detailed derivation of this algorithm is given in [Cichocki and Amari, 2002].

$$\mathbf{D}[n+1] = \mathbf{D}[n] + \mu\Big[\mathbf{I} - \mathbf{f}(\hat{\mathbf{s}}[n])\,\hat{\mathbf{s}}^T[n]\Big]\;\mathbf{D}[n] \tag{3.13}$$

with:

$$\hat{\mathbf{s}}[n] = [\hat{s}_0[n], \hat{s}_1[n],\ \ldots\ , \hat{s}_{M-1}[n]]^T \tag{3.14}$$

while $\hat{\mathbf{s}}[n]$ is calculated as follows:

$$\hat{\mathbf{s}}[n] = \mathbf{D}[n]\mathbf{x}[n] \tag{3.15}$$

Furthermore, $\mathbf{I}$ is the unity matrix and μ the step size of the adaptation. Note that the choice of an appropriate step size can be difficult as the convergence is dependent on the signal. Moreover, we have to define the function $\mathbf{f}(y)$. In our case it is defined as follows:

$$\mathbf{f}(\hat{\mathbf{s}}[n]) = [sign(\hat{s}_0[n]),\ sign(\hat{s}_1[n]),\ \ldots\ ,\ sign(\hat{s}_{M-1}[n])]^T \tag{3.16}$$

This function is introduced as *activating function* $\mathbf{f}(x)$ [Cichocki and Amari, 2002]. It relies on the estimated statistics of a signal and can directly be derived from the Probability Density

Function (PDF), $q(x)$.

$$\mathbf{f}(x) = -\frac{d\ log\ q(x)}{dx} \tag{3.17}$$

Knowing or estimating the PDF of a signal, the activating function can be derived and applied to the Natural Gradient Algorithm. Therefore, in order to implement the Natural Gradient Algorithm we need to know the statistics of the expected signals. A different PDF leads to a different activating function. Hence, two well known examples are given: The PDF $q(x)$ and the corresponding, activating function $\mathbf{f}(x)$ of a Gaussian and a Laplace distribution.

$$\text{Gauss:} \quad q(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\ e^{-}\frac{(x-\mu)^2}{2\sigma^2} \quad \Rightarrow \quad \mathbf{f}(x) = \frac{x}{\sigma^2} \tag{3.18}$$

$$\text{Laplace:} \quad q(x) = \frac{1}{2\sigma}\ e^{-}\frac{|x-\mu|}{\sigma} \quad \Rightarrow \quad \mathbf{f}(x) = \frac{sign(x)}{\sigma} \tag{3.19}$$

For other distributions and its corresponding activating functions see [Cichocki and Amari, 2002]. Note that zero-mean signals with $\mu = 0$ are assumed and for simplicity the variance is set by $\sigma^2 = 1$.

It is common to classify signals due to their statistical properties. Statistical moments like the mean value μ or the variance σ^2 describe the distribution of a signal. From Equation 3.18 can be seen that the normal or Gaussian distribution can fully be described by these two parameters. Another distribution, fitting the features of speech best, is the Laplace distribution. At this point we want to introduce the third moment and fourth satistical moment which are: skewness γ and kurtosis γ_2. Although the third moment is zero the fourth moment of the Laplace distribution is known to be $\gamma_2 = 3$. It is not needed to define the distribution, however, it is inherent. According to [Hyvärinen and Oja, 2000], where a detailed explanation is given, higher order statistics are necessary if a successful BSS using ICA shall be applied. This leads to a nonlinear activating function as shown in Equation 3.19. A good overview of stochastic signals and estimations is given in [Vary and Martin, 2006].

The Laplace distribution is known to match the statistical properties of speech best. Consequently, the corresponding activation function $\mathbf{f}(x)$ is implemented. It is obvious that if another distribution is expected its corresponding function needs to be implemented.

4 Blind Source Separation by Frequency Invariant Beamforming

The applied approach, suggested in [Liu and Mandic, 2005] and [Liu, 2010], combines the former explained techniques. The algorithm contains two successive stages:

Frequency Invariant Beamforming Network

Blind Source Separation Algorithm

A beamforming network, consisting of several FIB, scans the room while the BSS algorithm is supposed to estimate a set of coefficients whereby the beamformers are weighted. The relationship of the beamformers coefficients determines from which DOA angle signals are either enhanced or suppressed.

4.1 Concept

In order to explain the concept of this approach the principle signal flow is depicted in Figure 4.1. All notations of the former chapters remain. The beamforming network, consisting of N parallel working FIB, is the first stage and processes the microphone signals $x_0, x_1 \ \ldots \ x_{M-1}$. The beamformers are denoted by $w_0, w_1, \ \ldots \ w_{N-1}$ whereas the corresponding outputs are denoted by $y_0, y_1, \ \ldots \ y_{N-1}$, which is the input of the BSS algorithm with its adaptive demixing matrix D. Finally, the estimated and recovered source signals are denoted by $\hat{s}_0, \hat{s}_1, \ \ldots \ \hat{s}_{N-1}$.

A number of unknown source signals L, denoted by $s_0, s_1, \ \ldots \ s_{L-1}$ with corresponding DOA angle θ_l shall be recovered as best as possible. The demixing process provides N estimated respectively recovered signals.

Any estimated source signal $\hat{s}$, is a mixture of several beams, weighted by a set of coefficients. These coefficients are calculated by the proposed BSS algorithm. Each column of of the demixing matrix D represents a set of coefficients. Weighting every beamformer with its corresponding coefficient and add up these signals yields to one of the recovered signals $\hat{s}_l$. Each resulting beam is a linear combination of all FIB. It is possible to create a mixture of the beamformers which enhances parts of the half plane while others are suppressed. This mixing process is described

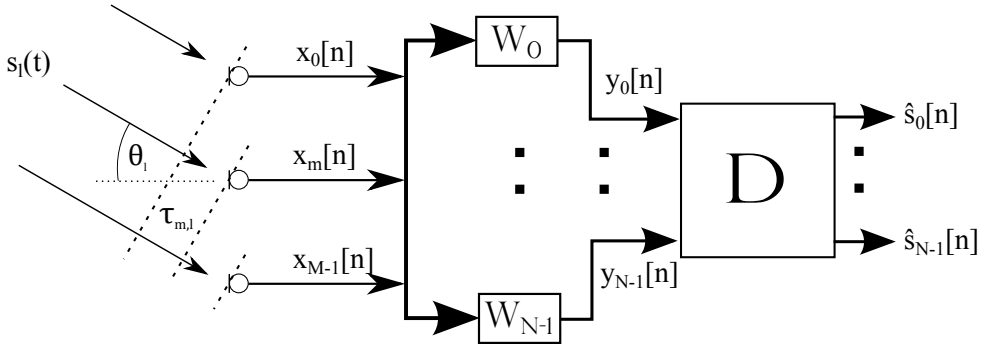

Figure 4.1: Frequency Invariant Beamforming network for instantaniuos BSS

and depicted in Chapter 4.3. The optimization process of the adaptive coefficients is performed by the BSS algorithm, described in Chapter 3. Note that the input signals in this case are already the outputs of the beamformers.

As we most likely have more then one source we want to recover all source signals. Hence, we determine several mixes in parallel, which are supposed to recover the inputs from different DOA angles as best as possible. The BSS algorithm is creating several mixes of the beamformers outputs. According to Equation (3.1) the demixing Matrix D has to be squared and with a number of beamformers $N = 7$ the same number of recovered signals is received accordingly, even if we expect or know that the number of speakers is lower. Hence, we have to reduce the number of recovered signals or simply spot the right results. The method of singular value decomposition reduces the number of signals before the BSS algorithm is applied and is explained in chapter 4.4. Another possibility would be a modified BSS algorithm addressing the problem of an overdetermined model, which is proposed in [Zhang et al., 1999] but has not been implemented yet.

4.2 Frequency Invariant Beamforming Network

To cover the whole half plane several beams in different directions scan the room whereas adaptive weights determine the value of the individual beams as these weighted beamformers are added up in the end. Interference signals can be canceled by choosing the right linear combination. A similar idea has already been introduced in [Sekiguchi and Karasawa, 2000]. In order to suppress interferences by a linear combination of the beams effectively two conditions for the beamforming network have been postulated:

a) The beamformers must have an identical phase characteristic which allows the output signals of the beamformers to be added without any delay compensation.

b) The beam pattern of each beamformer must be virtually frequency independent for every angle of the half plane in a way that an interference signal received in any sidelobe direction is not distorted. If this complies replicas of interferences signals can be generated and cancelled.

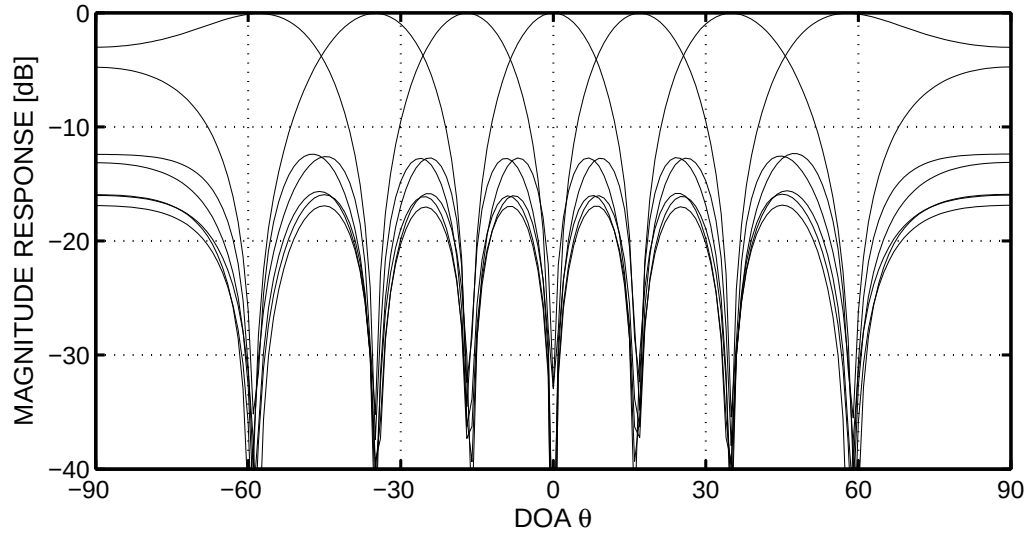

Figure 4.2: beam shapes of seven frequency invariant beamformers

Both of these conditions are accomplished by the the FIBs proposed in chapter 2.4. The following specifications of the parameters are chosen:

Number of microphones: $M = 19$

Number of filter taps: $J = 121$

Valid frequency range: $0.4 \leq \Omega \leq 1$

Furthermore, a number of beamformers $N = 7$ seems to be sufficient. The steering direction of the beamformers are chosen so that the entire energy of all beamformers is constant for all DOA angles. The main steering directions of all seven beams are:

$$\theta_{0,n} = [-59 \quad -35 \quad -17 \quad 0 \quad 17 \quad 35 \quad 59] \tag{4.1}$$

Note that for the main steering direction of one beamformer the magnitude response of all other beamformer is almost zero. Figure 4.2 shows seven parallel working beamformers distributed in a half plane.

4.3 Scaling the Beamforming Network

Consider one source from a certain direction it would be imaginable that all weights are zero except the one for the beamformer having the best matching steering direction. This case is very rare, especially in a multipath environment, where interferences shall be suppressed.

To illustrate the functioning of the scaled beamforming network an example, based on real data, is given. A desired source signal with a DOA angle of $\theta = 40°$ is corrupted by an interfering signal with a DOA angle of $\theta = -10°$. The coefficient set with the best performance in terms of SNR is the following:

$$D(:,1) = \begin{pmatrix} 9.0118 \\ 3.5604 \\ -8.4623 \\ -5.3323 \\ 2.5859 \\ 70.3469 \\ 7.7584 \end{pmatrix} \tag{4.2}$$

The beam patterns of Figure 4.2 are scaled by these coefficients. The beams, which are normalized to the beam with the highest coefficient, yield in a beam pattern enhancing the source while suppressing the interference. Figure 4.3 depicts all beams with a positive coefficient whereas Figure 4.4 shows the beams with a negative coefficient. The bold line represents the sum of all beams. If positive and negative beams have the same magnitude at a certain DOA angle any impinging signal disappears. The negative beam contains the same signal as the positive one but with a phase rotated by 180°. Consequently, a deconstructive interference cancels the signal.

This can be seen in Figure 4.5 where the resulting beam pattern is shown. The data from Equation (4.2) are taken from the experiment in a reverberant environment and is explained in Chapter 7.2. It can be seen that the main steering direction is directed towards the desired source ($\theta = 40°$) whereas th interference ($\theta = -10°$) is canceled.

Note that the small variations concerning the main steering direction and the direction of the canceled signals may result from inaccuracies of the experiment and the hardware. However, it

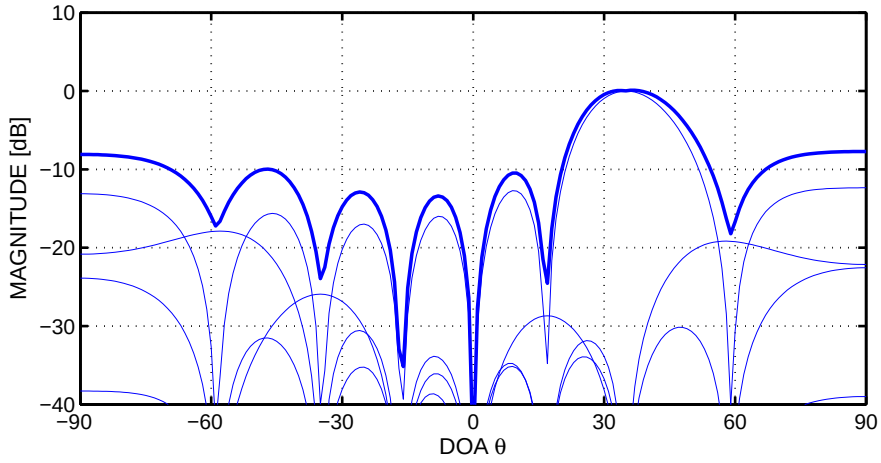

Figure 4.3: Weighted beams with positive coefficients; bold line: resulting beam pattern; signal phase: 0°

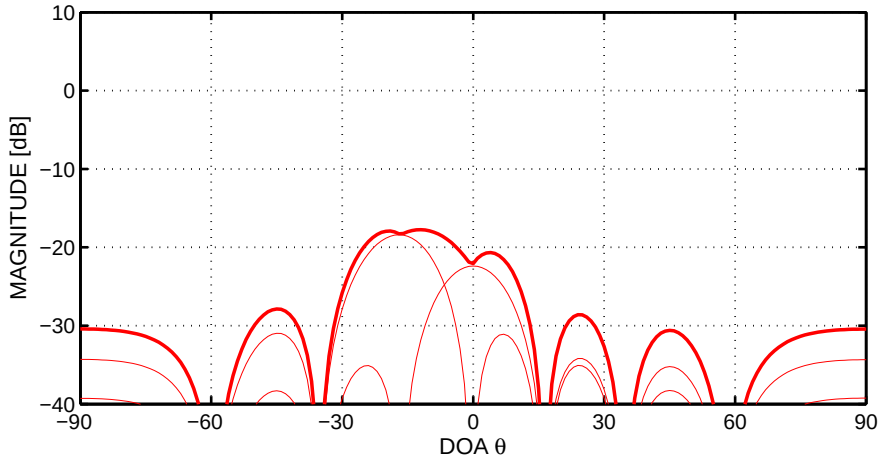

Figure 4.4: Weighted beams with negative coefficients; bold line: resulting beam pattern; signal phase: 180°

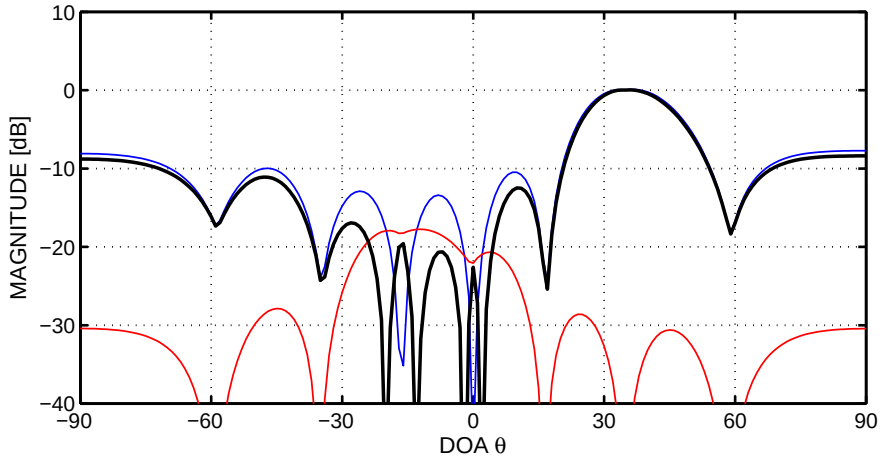

Figure 4.5: Resulting beam pattern (black line); combination of zero phase pattern (blue line) and 180° *phase pattern (red line)*

is shown that the final beamformer is working properly. Even if the interference source matches not exactly the minimum of the resulting beam the interference is at least attenuated by 20 dB. Figure 4.5 shows that more than one minimum exists. Hence, the algorithm is capable to cancel multiple interferences. It is conceivable that a major interference is a specific reflection of an interfering source whereas the beamformer is capable to cancel this reflection.

4.4 Singular Value Decomposition

As Singular Value Decomposition is utilized in order to reduce the number of resulting signals a short introduction to the topic is given. Furthermore, it is explained how this method can be integrated in the proposed algorithm.

Singular Value Decomposition (SVD) is a method to factorize a non squared matrix. It is closely related to Eigen Value Decomposition (EVD), also known as eigendecomposition, whereas this mathematical problem requires a squared matrix.

A general mathematical description of eigendecomposition can be given by:

$$A = Q\Lambda Q^H \tag{4.3}$$

A squared Matrix A with dimensions $[N \times N]$ and N linear independent eigenvectors q_i with $i \in [1, 2, \ldots, N]$ can be factorized by the eigen vector matrix Q. It contains the eigen vectors q_i in the i-th column whereas the diagonal matrix Λ contains the corresponding eigenvalues λ_i. Hence, A is represented by its eigen values and its eigen vectors. In Matlab the function `eig` is available in order to execute a EVD.

Singular value decomposition is closely related to eigendecomposition. A non squared matrix Y with dimensions $[N \times S]$ can be factorized as follows:

$$Y = U\Lambda V^H \tag{4.4}$$

Similar to the eigen vector matrix Q the columns of U are the eigenvectors of YY^H and are called left singular vectors. The columns of V are the eigenvectors of Y^HY and named right singular vectors. The non zero elements of Λ, labeled as non zero singular values, are the square roots of the non zero eigenvalues of either Y^HY or YY^H. The corresponding Matlab function, which is also utilized in course of this work is called `svd`.

SVD in our case is applied to reduce the number of recovered signals. In order to make use of the BSS algorithm, which provides the same number of outputs as inputs, the number of input signals has to be reduced before applying the BSS algorithm. We assume that the number of sources is known and described by L. It is supposed that the number of beamformers N is higher then the number of sources:

$$N > L \tag{4.5}$$

Additionally, we need to define S as the number of samples. The goal of the applied SVD is to reduce the number of output vectors of the beamforming network to L. We suppose that the outputs of the beamformer, described by the matrix Y, only contain L linear independent signals. Hence, the rank of the matrix Y equals L, this means that only L eigenvalues with $\lambda \neq 0$ exist. The problem can be explained best by regarding the matrices and the according

dimensions from Equation (4.4):

$$\underbrace{Y}_{[N\times S]} = \underbrace{U}_{[N\times N]} \cdot \underbrace{\Lambda}_{[N\times S]} \cdot \underbrace{V^H}_{[S\times S]} \tag{4.6}$$

We assume a number of non zero singular values σ_i with $i \in [1, \ldots, L]$, corresponding to the source number L. As the diagonal of the singular value matrix partly equals zero because of $N > L$ it is useful to separate U in two Matrices U_r and $\tilde{U}_r$ whereas the second part is completely nullified by Λ:

$$Y = \left[U_r, \tilde{U}_r\right] \cdot \begin{pmatrix} \sigma_1 & & 0 & \\ & \ddots & & \mathbf{0} \\ 0 & & \sigma_L & \\ & \mathbf{0} & & \mathbf{0} \end{pmatrix} \cdot V^H \tag{4.7}$$

If we apply U_r to the original signal we obtain a Matrix $\tilde{Y}$ with the required dimensions of $[L \times S]$ containing the linear independent vectors of Y.

$$\tilde{Y} = {U_r}^H \cdot Y \tag{4.8}$$

The Matlab function `svd` sorts the results by starting with the highest singular value. In practice the singular values do not equal zero $\sigma_{i>L} \neq 0$. Nevertheless, the vectors corresponding to the highest σ_i contain the source signals.

This method is implemented in the Graphical User Interface, which is described in Chapter 6.2. Note that this is only one possibility to reduce the number of resulting signals. An alternative would be to adjust the BSS algorithm itself, as proposed in [Zhang et al., 1999], [Joho et al., 2000], [Amari, 1999]. Conceivable would also be the development of an algorithm choosing the right signals after the BSS is applied, by considering the properties of the weighting coefficients. Most likely an implementation of a the former suggested approach yields to a more robust algorithm. Note that further research concerning this topic is not subject of this work.

Part II

Practical Approach

5 Hardware

The hardware setup used in course of this work consists of a microphone array, 20 suitable amplifiers, a data acquisition card and a PC (Windows 7; 32-bit). Whereas the array was already constructed the amplifiers had to be built. Note that the data acquisition card as well as the PC were already available. Figure 5.2 shows the general hardware setup.

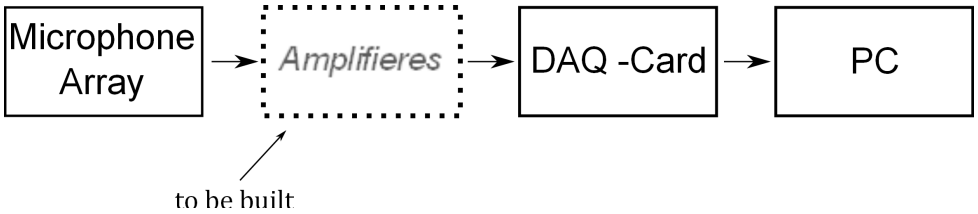

Figure 5.1: hardware setup

In this chapter the components of the hardware setup are documented. Moreover, the design process of the amplifiers is described. Furthermore, the occurring problems due to the hardware are outlined and solutions are given.

5.1 Microphone Array

The utilized microphone array was built in course of a previous work. It consists of 20 microphones. Sub miniature electret microphone cartridges are used and fitted in an appliance made of plastic. Each can be mounted easily on a metal bar with a length of $1m$ constructed for this purpose. Concerning the microphone the following specifications are offered by the distributer.

The microphone is supposed to have a omni directional directivity pattern. It requires a DC voltage between $1.5V$ and $10V$ by a maximum current consumption of $0.5mA$. The sensitivity at $1kHz$ is indicated by $5mV/Pa$. As diameter $6mm$ are denoted.

Figure 5.2: microphone array consisting of 20 sub-miniature electret microphone cartridges

It is worth mentioning that a conventional audio amplifier does not fit the specifications of these microphones since they usually provide $48V$ phantom power coming with a 3-pin connector.

Furthermore, the frequency response of the individual cartridges differs a lot. This can be stated by simply listen to recorded signals without conducting any measurements. Moreover, I believe that the directivity pattern of these microphones is not omni directional. Especially higher frequencies are concerned meaning the attenuation is increased by choosing a DOA angle apart from zero degree.

5.2 Amplifiers

Building the required 20 amplifiers was the first main step of my project. The department of Electronic and Electrical Engineering (EEE) of Sheffield University suggests a circuit, which was already proven in previous works, and is described immediately. The task was to create a proper board layout for 20 of these amplifiers. Furthermore, I had to build them and integrate them in the existing hardware. Above all, Electro Magnetic Compatibility (EMC) had to be considered in order to avoid attributable errors. After a first test setup consisting of 5 amplifiers I finally constructed a modified version which included 20 amplifiers.

5.2.1 Utilized Circuit

The built up amplifiers have two main purposes. They must supply the microphones with a certain voltage and, of course, amplify the signals while a low pass filter is implemented.

The suggested circuit is based on the operational amplifier TL072 and works in two stages. One single TL072 IC includes two operational amplifiers (data sheet see appendix). From Figure 5.3 it can be seen that the circuit consists of two successive stages. The first one has a fixed gain of 100, while the second stage is adjustable with a gain factor between 1 and 100. The ratio of the resistors R_4 to R_3 at the first stage and the ratio of R_6 to R_7 at the second stage set the

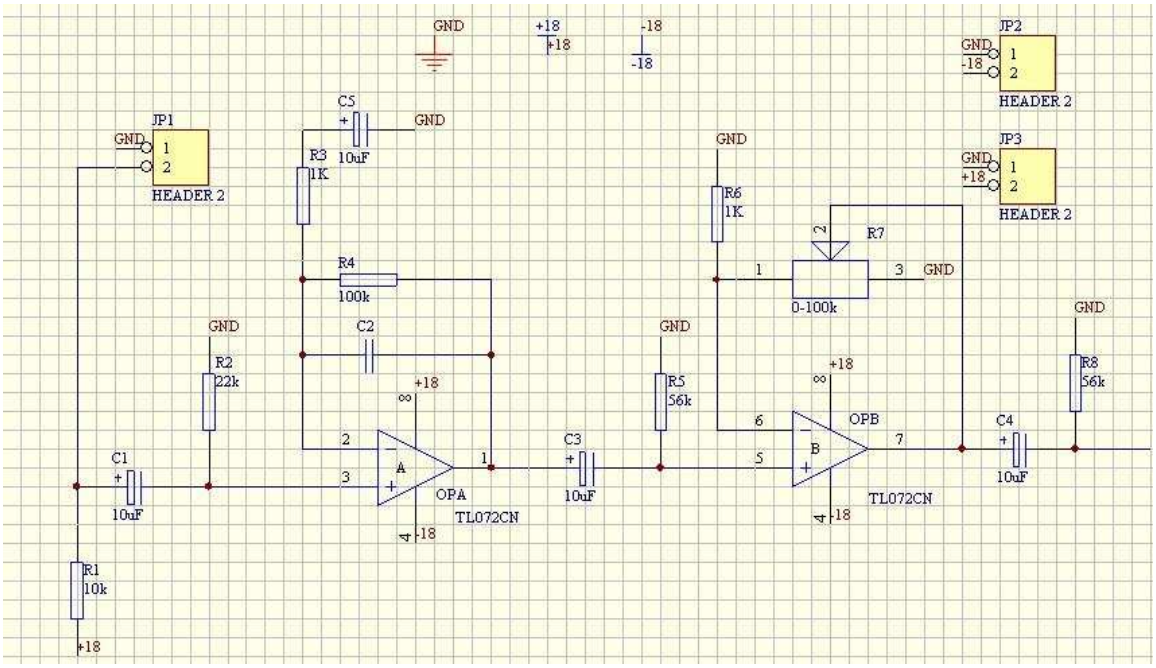

Figure 5.3: suggested circuit containing to successively working operational amplifiers TL072

gain:

$$A_1 = \frac{R_3}{R_4} = \frac{100\,k\Omega}{1\,k\Omega} = 100 \tag{5.1}$$

$$A_2 = \frac{R_6}{R_7} = \frac{1\ldots100\,k\Omega}{1\,k\Omega} = 1\ldots100 \tag{5.2}$$

Furthermore, the dimensioning of C_2 is crucial in terms of the cutoff frequency of the first order low pass at the first stage. A value of $C = 18pF$ was chosen which entails a cutoff frequency of 22 kHz. To preserve flexibility we did not chose a lower cutoff frequency although it would have been more effectively in order to avoid aliasing. Besides, a second order or even higher order low pass would have been desirable. At first appearance another low pass at the second stage would have made sense. Unfortunately, the cutoff frequency changes when adjusting R_7 in order to change the gain. Consequently, we must be satisfied with a first order low pass.
The electrolytic capacitors C_1, C_3 and C_4 are for common mode rejection. As the built up circuit provides a DC power of $8V$ the purpose of C_1 is to suppress the DC-voltage in the signal path. Summarizing the specifications of the circuit are:

Gain adjustable from $A = 100 \ldots 10000$

DC-voltage supply: $8V$

First order low pass with a cutoff frequency of 22 kHz

5.2.2 Creating the Board Layout

The discussed circuit has to be implemented. Therefore, we have to create a proper board layout by using the software *Proteus*. As mentioned earlier we have to consider the electromagnetic compatibility of our device. Basically this means that any possible unwanted effects or interferences caused by electromagnetic energy are to be avoided. A simple and very common example for an unwanted reception of electromagnetic energy in audio equipment is a $50Hz$ hum. It is obviously caused by the conventional power supply but can be receipted in many different ways. Starting to build the amplifiers the first prototype was a board with five amplifiers in a row. When testing the amplifiers it turned out that they worked in principle. Nevertheless, some irregularities occurred as a result of an insufficient electromagnetic compatibility. A few channels had a significant voltage offset, some unknown broadband noise occurred and the $50Hz$ hum was large. As we do not have any lines carrying $50Hz$ AC voltage in our circuit this interference is caused by an external source. A grounded housing would have been a solution. Since the lower frequencies are cut off anyway the performance is not affect essentially. To solve the other issues the second layout was developed by following the principles:

Separating the power supply from any signal path as well as possible.

Avoiding parallel lines. If parallel lines are necessary separate them with a grounded line.

Avoiding long signal lines by putting the input and output plugs close to the individual amplifier.

The resulting, improved layout can be seen on the next picture. The screen shot is an extract from the Proteus design showing one amplifier and the electrical connections for the power supply.

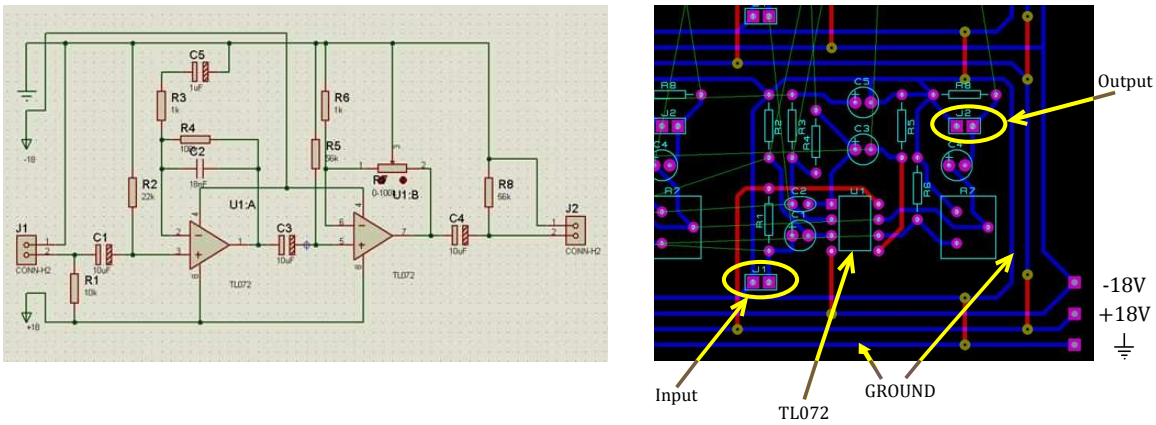

Figure 5.4: Proteus screen shot of the implemented circuit (one amplifier) // Proteus screen shot of the board layout (one out of twenty amplifiers including the connectors; blue lines button side; red lines top side)

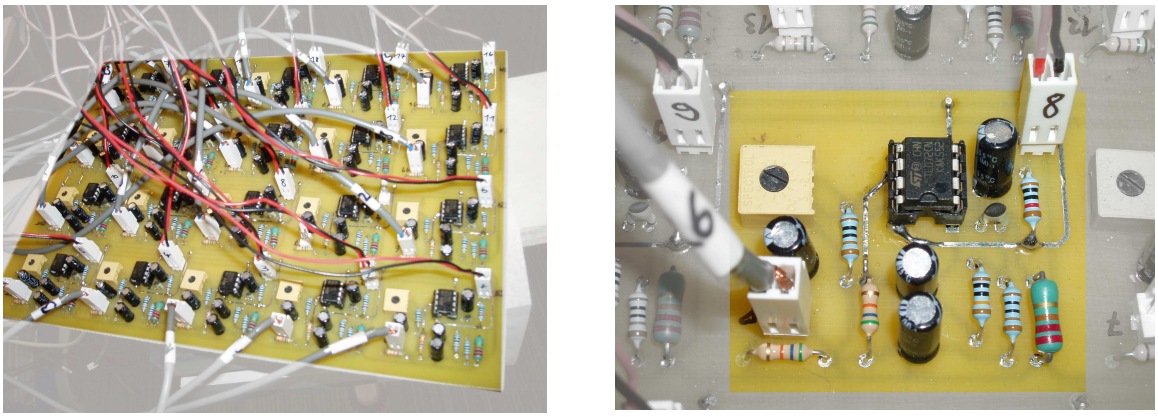

Figure 5.5: Photo of all twenty amplifiers built on the board
Photo of one out of the twenty amplifiers

For practical reasons most lines are located at the bottom of the board and are depicted as blue lines while wires on the top side are marked red. The three connectors at the bottom right corner starting from the top are $-18V$, $+18V$ and the ground. Unlike the first design the positive and negative power supply is separated by ground lines from every individual amplifier circuit. With the exception of the $50Hz$ hum every preceding EMC problem could be solved due to this design.

5.3 DAQ - card

5.3.1 Specifications

The utilized data acquisition card Adlink DAQ-2205 is the main interface of our system. It links the analogue equipment directly with the PC via PCI bus. In interaction with the Matlab data acquisition toolbox a direct data access with Matlab is possible. The data acquisition card supports 64 single ended analog input channels or 32 differential analog input channels. As the occurring signals share a common ground the maximal possible utilization of the card are 64 channels in theory. The A-D converter operates with a resolution of 16 bit with an input range of +/- 10 V . Furthermore, the card allocates programmable gains of x1, x2, x4 and x8, which allows the user to adapt the systems in terms of signal range by software. The card also provides two analogue outputs with 12-bit resolution and several digital in - and outputs.

Utilizing Adlink DAQ-2205 a certain number of problems occurred, which are discussed in the following sections. The manufacturer claims a 500 $\frac{kS}{s}$ acquisition. The sampling rate has to be divided by the number of used channels as a multiplexer for AD conversion is used. Due to the multiplexer two major problems result. Firstly, we face a limited sample rate, thus, the problem of a proper anti aliasing filter occurs. Secondly, we have to deal with a non-simultaneously data acquisition. Furthermore, coupling appears mainly caused by the breakout cable which connects the DAQ card and the amplifiers.

5.3.2 Limited Sample Rate and Aliasing

The first issue we have to deal with is a decreasing sampling rate when increasing the number of channels. Using all of the 64 channels yields to a theoretical sampling rate of:

$$f_s = \frac{f}{M_{chan}} = \frac{500000\,Hz}{64} \approx 7800\,Hz \tag{5.3}$$

Considering the Nyquist theorem we obtain the following maximum frequency:

$$f_{max} = \frac{f_s}{2} = 3900\,Hz \tag{5.4}$$

To effectively achieve the calculated frequency an ideal, analogue, rectangular anti aliasing filter would be required. However, even higher order analogue filters do not fully accomplish these requirements consequently the maximum frequency decreases again. In our case the built up amplifiers provide just a first order low pass so that oversampling is absolutely nessecairy when additional higher order filters are not used. Our specific approach with 19 channels leads to a theoretical sampling rate of:

$$f_s = \frac{f}{M_{chan}} = \frac{500000\,Hz}{19} \approx 26316\,Hz \tag{5.5}$$

In practice the sample rate is determined by $f_s = 24999\,Hz$. A good explanation how to avoid aliasing by utilizing oversampling and a simple analogue filter is given in [Oppenheim et al., 2004]. Our approach is quite similar. We use oversampling to simplify an anti aliasing filter while making it more cost effective via a software implementation. A simple low pass filter is provided by the amplifiers. Note the the cutoff frequency actually is too high. Due to the fact that the signal energy at higher frequencies ($f > 12.5\,kHz$) is low enough aliasing of these frequencies does not affect the results significantly. According to the signal specification we now

apply the A-D conversion in a way that we are not affected by any aliased signals or noise. The sample rate of the converter is high enough to ensure that no significant signal components over $f_s/2$, which would cause aliasing, occur. In our case this is accomplished when using a sampling rate of 24999 Hz.

Under these conditions we can easily apply a sharp digital low pass. The cutoff frequency in our case is 4150 Hz and, therefore, matched with our sampling rate reduction of $F = 3$. As a result $f_s = 8333\,Hz$ and, consequently, $f_{max} = 4161\,Hz$.

As mentioned earlier, dealing with speech signals a maximum frequency of at least 4000 Hz is desirable. The reason for that is that the majority of the information is situated in frequency ranges below 4000 Hz. Besides, we can assume that signals with frequencies over 12500 Hz, which equals $f_s/2$ are not crucial simply because of their low magnitudes. In our case this is really important whereas it is not possible to avoid aliasing in this frequency range due to the lack of a appropriate analogue filters.

Regarding the current setup leads to the conclusion that its maximum of channels is reached. If it is necessary to increase the number of channels, higher order analogue low pass filters would be required. Increasing the number of channels without adding additional analogue filters would entail either a reduction of the sampling rate or a significant deterioration of the anti aliasing filter. This in turn, would lead to clearly perceptible distortions in the recordings.

5.3.3 Non Simultaneously Acquisition

The second issue is the successive A-D conversion of every single channel. Unfortunately, the card does not include any on board sample and hold devices. To ensure simultaneous data acquisition sample and hold devices are necessary when using a multiplexer. Usually they are already part of a multichannel A-D converter. The impact caused by not reading in the channels simultaneously can easily be compared with a shift of the angle of the incoming signal. Considering a signal with an impinging angle of zero degree we should obtain the same signal at every microphone if the above mentioned conditions are complied. If the processing of the A-D conversion is successive every channel is delayed by a multiple of the time delay T_{AD}. This leads to a constant shift of the steering direction of the hole system. According to the manufacturer the additional time delay between two sampled channels can be calculated as followed:

$$T_{AD} = \frac{counter\ sampling\ interval}{timebase} = \frac{80}{40\,MHz} = 2\mu s \tag{5.6}$$

which equals:

$$T_{AD} = \frac{1}{f_s} = \frac{1}{500000\,Hz} = 2\mu s \tag{5.7}$$

To circumvent the problem Adlink provides a signal for external sample and hold devices. So it would be possible to insert a sample and hold device for each of the channels between the amplifiers output and the signal input of the card. The card itself would provide a control signal for every of these devices. As a result the analog inputs would be sampled at the same time.

Making use of this feature the problems could be solved in hardware. Compared with a software solution no additional latency is caused while saving performance of the PC which could be a crucial point in terms of a real time implementation. Nevertheless, it would be quite a big effort to build and include all the hardware.

Another solution is to compensate the delay via software. This solution requires one FIR filter per channel to apply the appropriate delay. For further information concerning fractional delay please refer to Chapter 2.4.5.

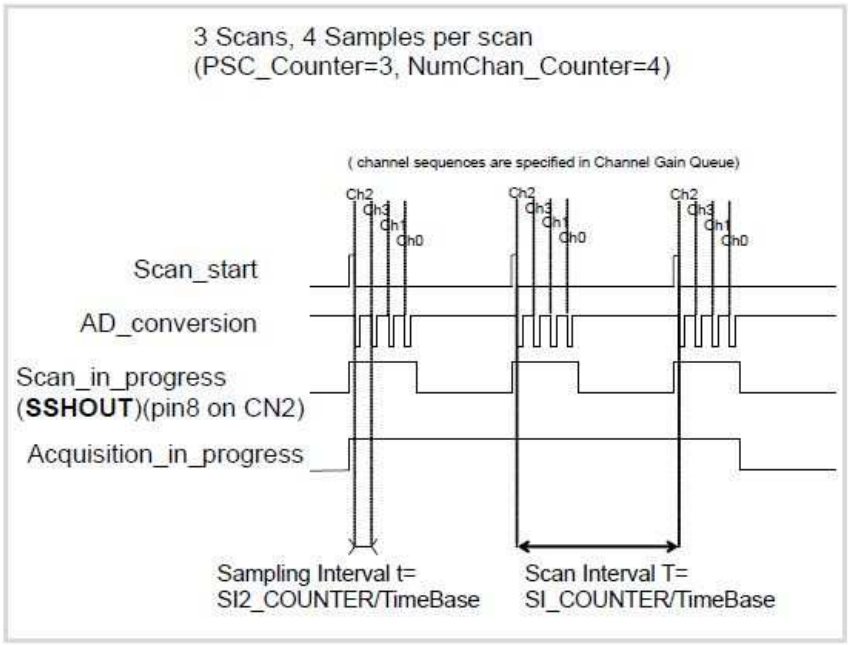

Figure 5.6: AD conversion process of DAQ-2205 with a multiplexer; providing a control signal SSHOUT for additional sample and hold devices

The last possibility is to simply accept the delay of $2\mu s$ knowing we have a shift of the steering direction. As we know the delay and the dimensions of our array we can calculate the shift as followes:

$$\phi = arcsin\left(\frac{T_{AD}}{T}\right)\frac{360}{2\pi} = 0.96^{\circ} \tag{5.8}$$

with T the time difference between two microphones:

$$T = \frac{d}{c} = \frac{0.0408m}{340\frac{m}{s}} = 0.119ms \tag{5.9}$$

In our specific case an inaccuracy of not even one degree is acceptable as considering the blind beamforming approach. However, note that a change of the array dimensions and a change of the sampling rate can cause a much higher shift than in our approach.

5.3.4 Coupling

The term coupling describes an unwanted transmission from two adjoining channels. It is also known as cross talk. In the field of analogue audio technology this is a well known problem. When processing several parallel analogue signals it can be difficult to avoid coupling completely. Hence, it appears in many analogue device to a certain degree. Usually the problem is caused by inducted voltage of another analogue channel. The cross talk of two adjoining channels can be calculated as follows:

$$C = 20\,log_{10}\left(\frac{RMS_{xtalk}}{RMS_{orig}}\right) \quad [dB] \tag{5.10}$$

RMS_{orig} is the original signal energy of an unwanted source and RMS_{xtalk} is the signal energy captured at another channel. Note that coupling can be frequency dependent, which is not regarded here. An reasonable standard value could be $C = -80\,dB$. In our case coupling is caused by unshielded, parallel wires. The connection cable, linking the accessory connectors box with the DAQ card in the PC, contains 68 wires which are not shielded. With a length of one metre coupling occurs in a non tolerable magnitude. A coupling of $C = -10\,dB$ occurred.

In order to measure the energy of every channel I implemented the matlab GUI `crosstalk.m`. Measuring the impact of one channel to its surrounding channels the most effective method was to ground as much as possible channels in between the data channels. As 20 channels were required it was possible to ground 44 of the 64 available data channels.

Figure 5.7 shows the used channels and the remaining grounded channels. Due to the additional grounding a minimum value of C_{min} was achieved:

$$C_{min} = -35\,dB \tag{5.11}$$

Note that this is the best result in terms of coupling which can be achieved by using the accessory cable and connectors box of the DAQ card when using 20 channels.

AI0 (AIH0)	1	35	(AIL0) AI48
AI1 (AIH1)	2	36	(AIL1) AI49
AI2 (AIH2)	3	37	(AIL2) AI50
AI3 (AIH3)	4	38	(AIL3) AI51
AI4 (AIH4)	5	39	(AIL4) AI52
AI5 (AIH5)	6	40	(AIL5) AI53
AI6 (AIH6)	7	41	(AIL6) AI54
AI7 (AIH7)	8	42	(AIL7) AI55
AISENSE	9	43	AIGND
AI8 (AIH8)	10	44	(AIL8) AI56
AI9 (AIH9)	11	45	(AIL9) AI57
AI10 (AIH10)	12	46	(AIL10) AI58
AI11 (AIH11)	13	47	(AIL11) AI59
AI12 (AIH12)	14	48	(AIL12) AI60
AI13 (AIH13)	15	49	(AIL13) AI61
AI14 (AIH14)	16	50	(AIL14) AI62
AI15 (AIH15)	17	51	(AIL15) AI63
AI16 (AIH16)	18	52	(AIL16) AI64
AI17 (AIH17)	19	53	(AIL17) AI65
AI18 (AIH18)	20	54	(AIL18) AI66
AI19 (AIH19)	21	55	(AIL19) AI67
AI20 (AIH20)	22	56	(AIL20) AI68
AI21 (AIH21)	23	57	(AIL21) AI69
AI22 (AIH22)	24	58	(AIL22) AI70
AI23 (AIH23)	25	59	(AIL23) AI71
AIGND	26	60	AIGND
AI24 (AIH24)	27	61	(AIL24) AI72
AI25 (AIH25)	28	62	(AIL25) AI73
AI26 (AIH26)	29	63	(AIL26) AI74
AI27 (AIH27)	30	64	(AIL27) AI75
AI28 (AIH28)	31	65	(AIL28) AI76
AI29 (AIH29)	32	66	(AIL29) AI77
AI30 (AIH30)	33	67	(AIL30) AI78
AI31 (AIH31)	34	68	(AIL31) AI79

Figure 5.7: resulting CN1 Pin Assignment for DAQ card

6 Software

6.1 Data Acquisition with Matlab Data Acquisition Toolbox

6.2 Graphical User Interface

The Matlab GUI `Beamformer_Gui2_1.m`, which I developed in the course of this thesis, is an easy to use development tool. It enables the user to record data and apply the algorithm immediately. For a maximum number of three speech signals the results are represented. In addition, the user can listen to every recovered signal. Furthermore, previously recorded data can be loaded and processed. Moreover, if data have either been recorded or loaded onces, the algorithm can be applied again while changing significant parameters.

6.2.1 Parameters and Handling

To accomplish the requirements of a simple and easy accessible interface the options of the user interface are restricted. In the section on the top left side new data can be recorded while the maximum input voltage of the DAQ card is required in order to control the signal level and

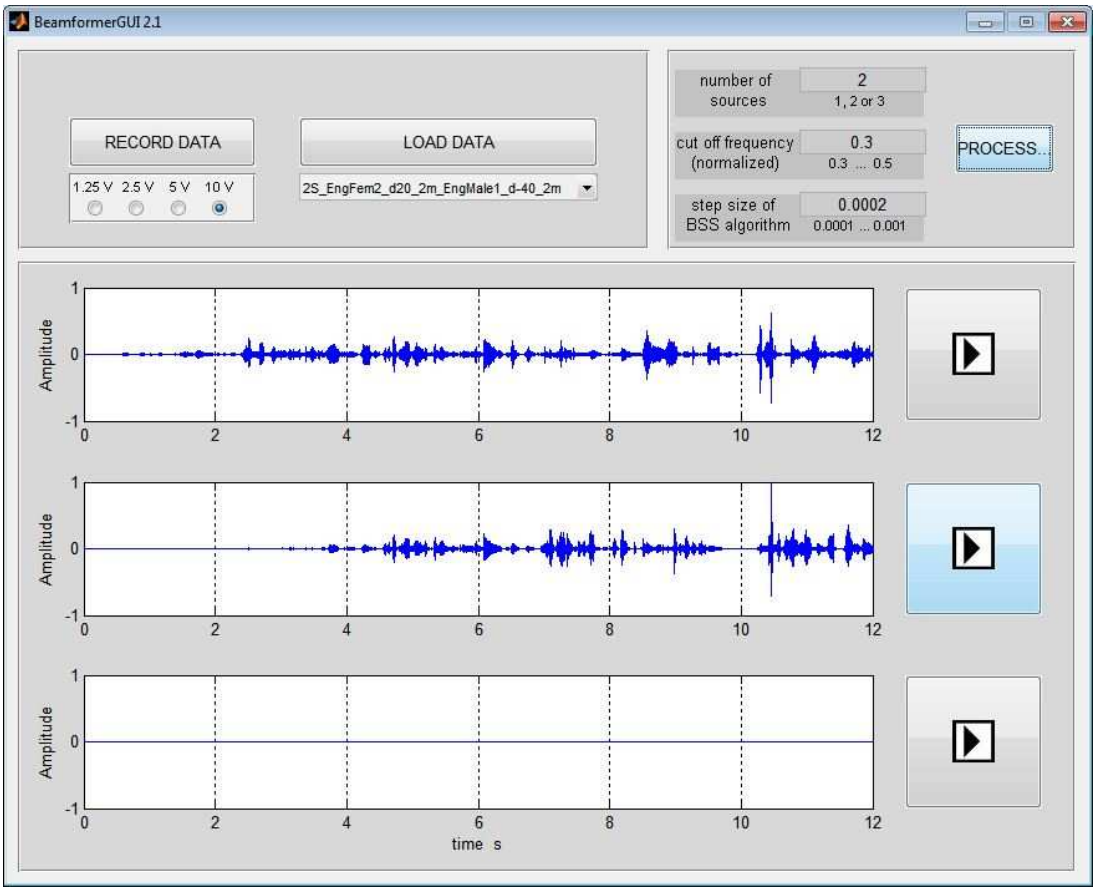

Figure 6.1: Graphical User Interface

increase the accuracy of the AD converters. If the user decides to load data all files previously saved in the folder *mixed_signals_RoomB59_21_01_11* are available. The current data set is a part of the recordings described in Experiment II. Activating the Record Data or the Load Data button includes the prompt execution of the algorithm with the chosen parameters. If the user wants to process the data again with a different set of parameters these can be changed in the top right section.

The number of sources is required, whereas the GUI is restricted to a maximum number of three sources as a signal separation of more then three speakers under real conditions utilizing the proposed system seems to be an unrealistic task. It is possible to change the normalized lower cutoff frequency. The suggested value is $\Omega = 0.4$, which corresponds to $f = 1666Hz$ because it is known that the Beamforming network is working frequency independent at least in a range of $0.4 \leq \Omega \leq 1$. Nevertheless, good results might be obtained by setting a lower cutoff and, therefore, having a wider spectrum of the signals. Note that a modification of the sample rate $f_s = 8333Hz$ is not necessary as the sample rate is already matched to the spectrum of a speaking person. Moreover, the array would have to be adjusted according to a different sample rate. The last accessible parameter is the step size of the BSS algorithm which sets the adaption speed. An appropriate range of values is suggested.

The bottom section depicts the recovered signals in time. Comparing the time signals a visual clue for a fast evaluation of the signal separation is provided. The recovered signals can be listened by pressing the corresponding play button.

6.2.2 Structure

The current algorithm is implemented in modular way. Every major step is implemented as a single Matlab function. Therefore, it is easily possible to adjust and manipulate parts of the algorithm. Moreover, it would be conceivable to remove a segment and replace it by an alternative function. The idea to reduce the number of signals before the BSS algorithm is applied by Single Value Decomposition, described in Chapter 4.4, is implemented in this GUI.

The Matlab script `Beamformer_Gui2_1.m` creates the GUI. All actions taken by the user are executed by one of the following callback functions, whereas `pb_call_processing.m` is the function containing the algorithm:

`pb_call_record.m`: Callback function recording data from the microphone array and saves the unprocessed data.

`pb_call_loading`: Callback function loading requested data which are saved in the folder Data_RoomB59_21_01_11/mixed_signals_RoomB59_21_01_11. It is possible and recommended to store new files in this folder.

`pb_call_gui2.m`: Callback function enabling the user to listen to the final results.

`pb_call_processing.m`: Callback function applying the whole algorithm. The main steps are implemented in separate functions, which are shortly introduced next.

The following functions are executed step by step by `pb_call_processing.m`:

`postprocess.m`: Scales the recorded data according to the maximum input voltage. An anti aliasing filter is applied before the signals are down sampled by the factor 3 to $f_s = 8333$. Additionally, a low cut filter with an adjustable cutoff frequency is applied.

`FIBeam.m`: Applies several frequency invariant beamformers. The filter coefficients are loaded from `coeff_3D_m7_K121.mat`

`svd_reduce_B.m`: Reduces the number of signals by utilizing single value decomposition.

`BSS.m`: A blind source seperation algorithm based on the natural gradient algorithm is applied.

`reconstruct_singnal.m`: Reconstructs the signals with the coefficients estimated by the BSS algorithm. The current set of coefficients is utilized. The recovered signals are normalized.

Moreover, the GUI requires a few additional functions: `norm_signal.m, design_lowcut.m, getfilenames.m`. The filter coefficients for the anti aliasing filter are loaded from `AA_fs24999.mat` whereas the coefficients for the beamforming network are loaded from `coeff_3D_m7_K121.mat`. Both files could be replaced if, for example, a different filter design shall be tested.

Due to the fact that the algorithm is implemented separately from the interface it is possible to develop new algorithms while using the same framework. The algorithm, implemented in a modular way, enables the user to change or replace parts quite easily in order to observe the resulting signals. On this account, this application is supposed to be used as basis for further development.

Part III

Results

7 Experiments

7.1 Experiment I

The first experiment was accomplished in a non reverberant environment. For that reason, we utilized the anechoic chamber in the Portobello Centre of the University of Sheffield. With respect to errors caused by curved waveforms the distance from the source to the microphone array was chosen with $r = 6\,m$. For a more detailed discussion of this problem please see Chapter 7.3. With a microphone number $M = 19$ and a distance of two adjoining microphones $d = 0.0408\,m$ the dimension of the array was $a = 0.7344\,m$. The microphone array and the source were located at the same height. As we had a non reverberant environment, meaning a single path scenario, we were allowed to turn the array in order to change the DOA angle instead of moving the source.

7.1.1 Experiment I a

First and foremost the purpose of this measurement was to review the performance of the beamformer. A customary loudspeaker was utilized as source. The source signal was white Gaussian noise. Applying the noise from a DOA angle in a range from $0 \leq \theta \leq 90$ with a resolution of $5°$ we are able to calculate the beam pattern later on.

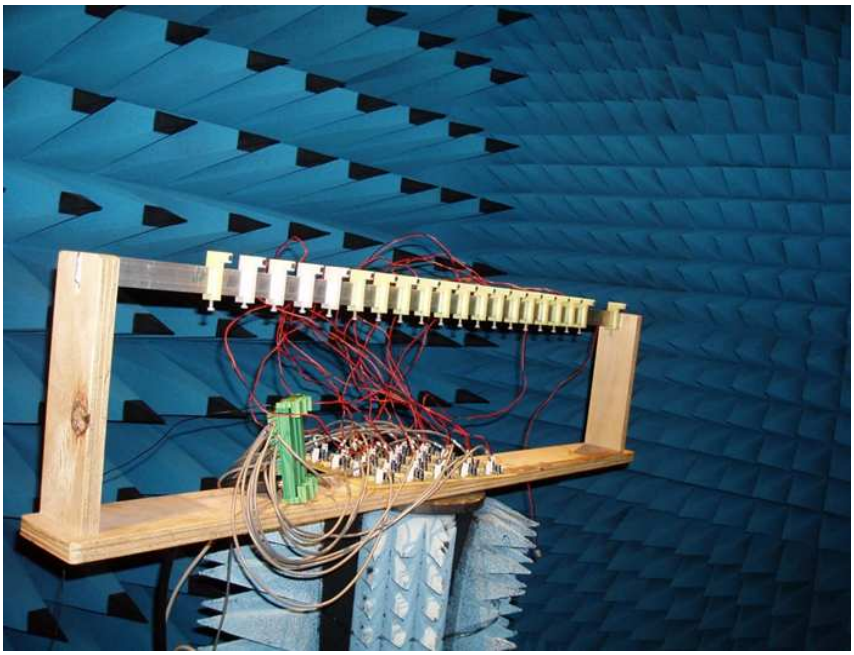

Figure 7.1: Microphone array in anechoic chamber

7.1.2 Experiment I b

Furthermore, we realized a small experiment with real speaking persons as sources. The voices of two male speakers are recorded from a certain DOA angle. To ensure that the male speakers are easily to distinguish one of them talked in Chinese while the other person spoke in German. Due to flexibility the voices are recorded successive and the data was combined after the experiment.

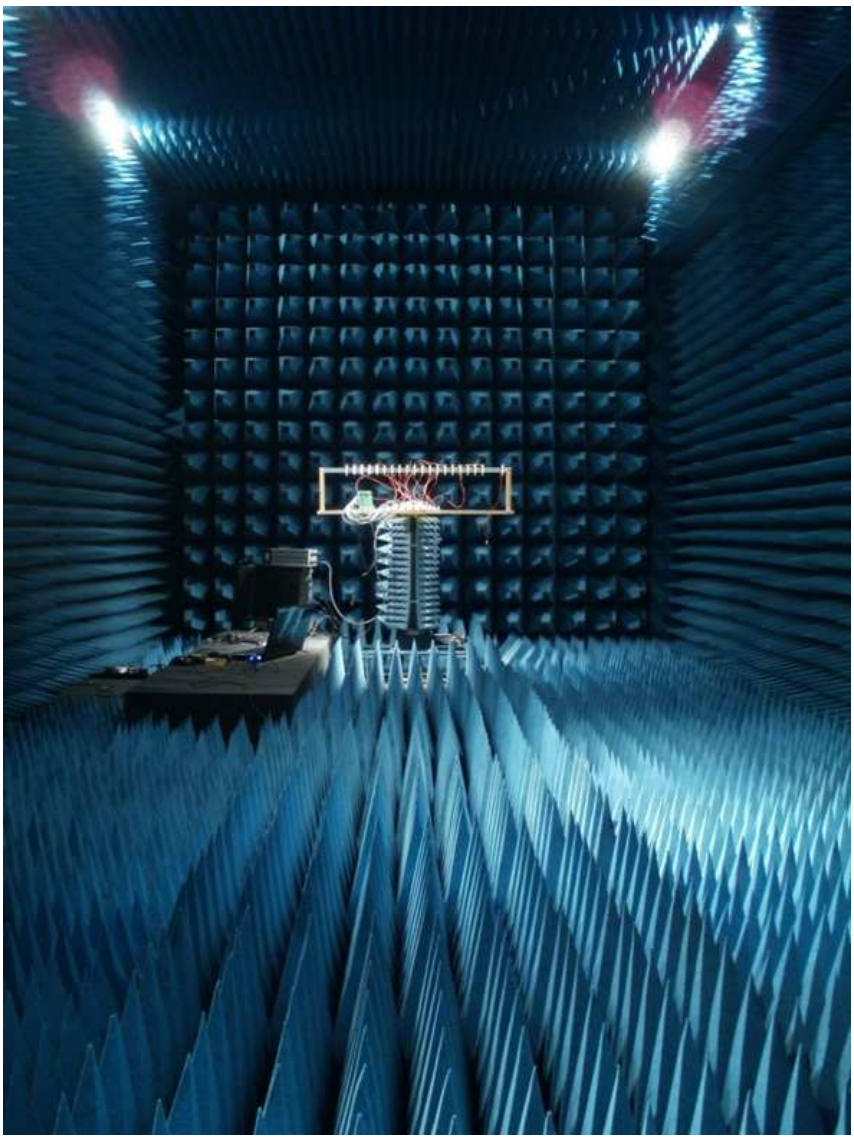

Figure 7.2: Microphone array in anechoic chamber

7.2 Experiment II

The second experiment took place in a multipath environment. The lecture hall B59 in the Portobello Centre of the University of Sheffield with around 110 m^2 was utilized. The exact dimensions of the room were: $[11.10 \times 9.80 \times 3.30]$ m. To comply the assumption of a constant plane both the microphone array and the source were located at a height of $h = 1.50$.

Basically a huge number of speech signals was recorded in order to combine them afterwards. As sources four different female speech samples and four different male samples were available. They were played back by a customary loudspeaker.

With a microphone number $M = 19$ and a distance of two adjoining microphones $d = 0.0408m$ the dimension of the array is $a = 0.7344\,m$. Three different distances were chosen: $r \in [2, 3, 4]\,m$. Note that with the current dimensions of the array and $r = 2m$ the effect of the curved wave is noticeable. On the other hand the direct signal to reverberation ratio is significantly higher at this point, which increase the performance of the algorithm in terms of signal separation.

The range of DOA angles was set from $-70 \leq \theta \leq 70$ with a resolution of 10°. The speaker was moved from one position to another successively while the microphone array remained on its position. Figure 7.3 gives a true to scale view on the measurement situation in the lecture hall. The sources s_0 and s_1 are examples with a DOA angle $\theta_0 = -30°$ and $\theta_1 = 50°$ and a distance from source to array of $r = 3\,m$. In total 45 different positions were measured. In the main steering direction $\theta = 0$ all eight different speech samples were recorded. On all other positions two female and two male samples, which were chosen more randomly, were recorded.

Additionally a few moving sources were recorded. This means that the loudspeaker was moved during the recording.

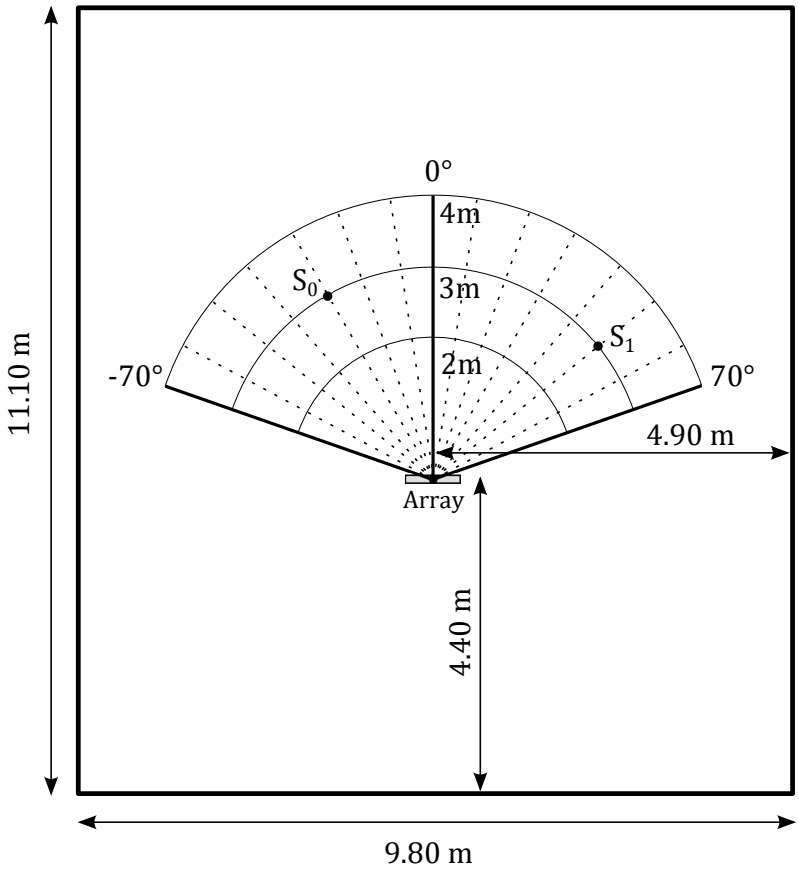

Figure 7.3: Plan of Measurement setup in Room B59; Portobello Centre; Sheffield University (true to scale)

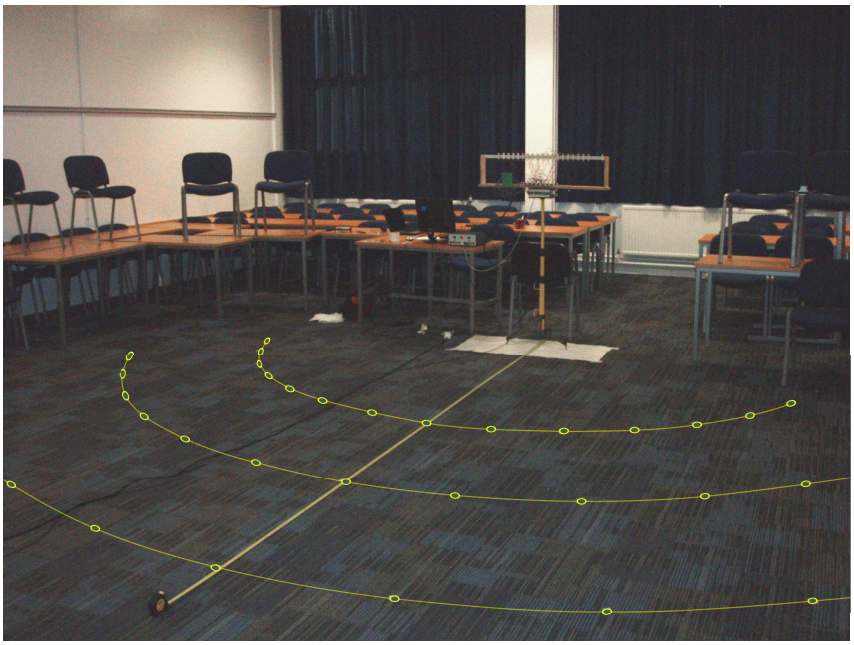

Figure 7.4: Test setup in an echoic environment; Room B59, Portobello Centre, Sheffield

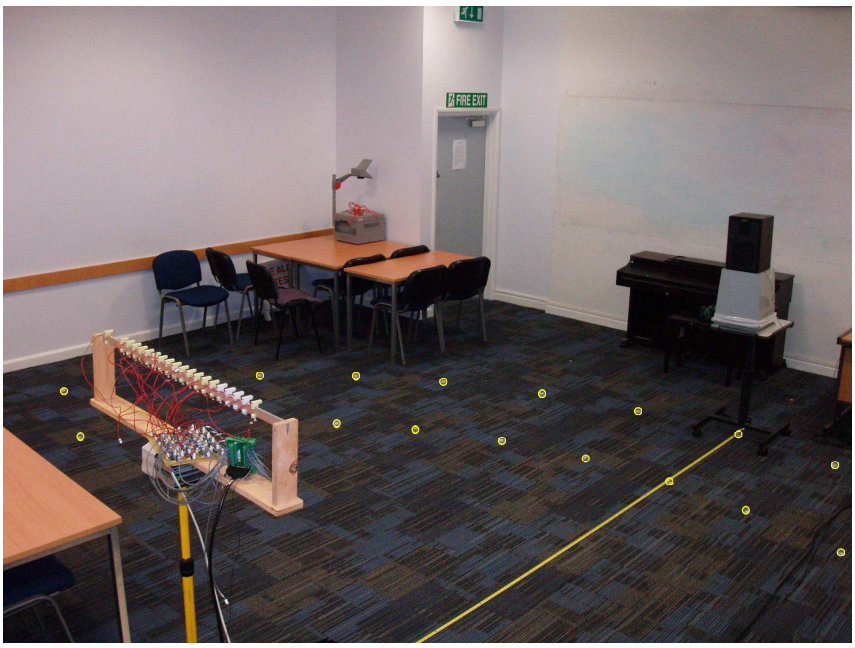

Figure 7.5: Measurement situation in an echoic environment; loudspeaker with a distance of $d = 4m$; DOA angle of $\theta = 0°$

7.3 Considerations of the Effect of Curved Waveforms

Though, the input signal was assumed to be a plane wave. This assumption makes it possible to define a delay $\tau_{m,l}$, only according to the m-th microphone and the DOA angle of the the l-th source signal. However, this assumption does not fit to the real world very well. One possibility to qualify sources is to regard them as point sources with an isotropic radiation. In fact, this model suits as the propagating wave is curved. It should be noted that real sources are usually not isotropic radiators but have a main radiation direction.

As the condition of a plane wave can not be fulfilled completely errors are caused and the delay $\tau_{m,l}$ is biased by an error.

Regarding the simplest case of an impinging signal with $\phi = 0°$ all delays are zero $\tau_m = 0$ if we assume plane waves. This is not the case if a curved wave is supposed to be the source. Considering a point source in one line with the microphone array, the more the sensors diverge from the centre of the array the bigger the delay shift becomes. Figure 7.6 depicts two identical arrays with different distances from the source to the array. The resulting errors τ_{e1} and τ_{e2} are dependent on the ratio of the overall array dimensions to the distance from the source to the microphone array. Increasing the distance the curved wave front converges more and more into a plane wave and consequently the error disappears. On the other hand, increasing the overall dimensions of the array leads to bigger delay shift. These thoughts become crucial when it gets to the practical approach. To put it simply, if the speaker gets too close the beamformer will work significantly worse.

In Figure 7.7 the simulation results of the effect of a curved waveform is depicted. In order to estimate the shortest possible distance approximately for the measurements in Experiment II different distances are compared with a maximum distance of $r = 1000m$. Utilizing the pythagoreom theorem we are able to calculate the additional delay τ_e. It is a function of the distance from source to the centre of the array r, the DOA angle and the distance between the different microphones d. It describes the delay (in space) of a sensor next to the centre microphone of the array:

$$\tau_e(\theta) = \sqrt{r^2 + cos(\theta)\, d^2} \; - r \tag{7.1}$$

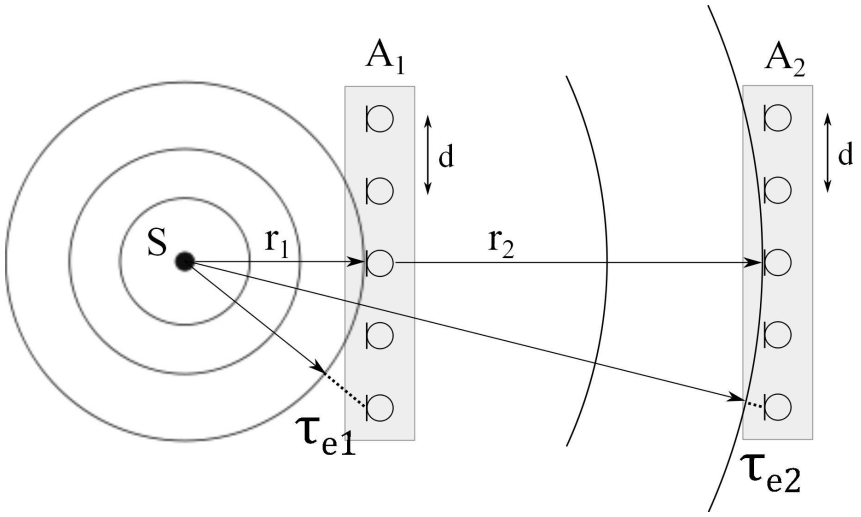

Figure 7.6: Comparison of errors τ_{e1} and τ_{e2} evolved by a curved waveform for two arrays; distances r_1, r_2 from the source S to the arrays A_1, A_2; distance between two microphones: d

The delay of any microphone having with a distance of $k \cdot d$ from the centre microphone can be calculated as follows:

$$\tau_e(\theta, k) = \sqrt{r^2 + cos(\theta)\,(kd)^2} - r \tag{7.2}$$

Note that the case shown in Figure 7.6 and simulated in Figure 7.7 with a DOA angle of $\theta = 0°$ is the worst case. From Equation (7.2) can be seen that the effect has its biggest impact for $\theta = 0°$, whereas for $\theta = 90°$ or $\theta = -90°$ it is zero and no errors occur. The simulation shows that even if the source is in a distance of two metres the results might be still sufficient. It was the basis for our decision in terms of distances in Experiment II.

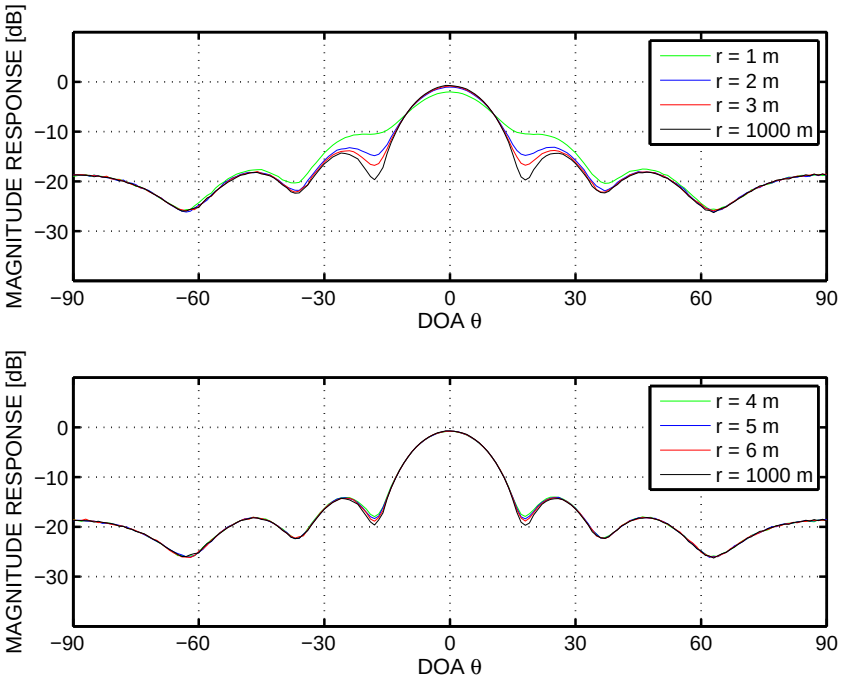

Figure 7.7: Comparing the responses of a beamformer for different distances from the source to the array; main steering direction of $\theta_0 = 0°$; DOA angle $\theta = 0°$

8 Results

8.1 Anechoic chamber

This section refers to the measurement setup described in Experiment Ia. Its main purpose is to test the performance of the frequency invariant beamformer proposed in Chapter 2.4. To evaluate the quality of the results they are compared to appropriate simulations.

The source signal is Gaussian noise. The recorded data are processed as it is described in Chapter 6.2 excluding SVD and BSS. A low cut filter with a normalized cutoff frequency $\Omega = 0.4$ ensures that the FIB is exclusively applied in a valid frequency range. To calculate the signal energy a sample of five seconds is taken and the RMS is derived. The beam pattern is calculated by building the ratio of the signals energy in steering direction $RMS_{\theta=0}$ and the signals energy of the current DOA angle RMS_θ as follows:

$$BP = 20\, log_{10}\left(\frac{RMS_{\theta=0}}{RMS_\theta}\right) \quad [dB] \tag{8.1}$$

Note that with this setup it is not possible to evaluate the frequency dependence of the beamformer as sinusoidal signals would be required. However, the overall performance can be measured. The simulations are conducted with the identical bandpass filter. In order to provide vividness the results for $0 \leq \theta \leq 90$ are mirrored to depict the whole half plane. The effect of a curved waveform is neglected in Figure 8.1 and considered in Figure 8.2.

It can be perceived that the results fit the simulation in general. The slight divergences can be caused by many effects. First of all, the accuracy of the setup is restricted. The DOA angle θ and the distance of the microphones d are affected by errors. Furthermore, the anechoic chamber is not 100 per cent anechoic, hence, reflection affect the results. Moreover, the attenuation of the propagating wave alters the magnitude at the microphones, which is not simulated. Another source of errors is the hardware itself as described in Chapter 5. It is a noticeable fact that the zeros of the real beamformer are not as clearly distinct as the simulation results are. First of all, this is caused by the limited resolution of the DOA angle of 5° whereby it is not always possible to match the corresponding zeros. Furthermore, the inaccuracies described above are the reason for the incomplete cancellation.

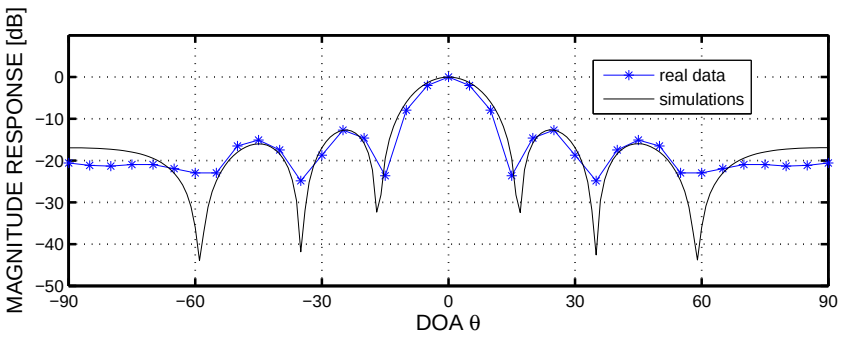

Figure 8.1: Comparing simulations results with the results of real data; recorded in the anechoic chamber

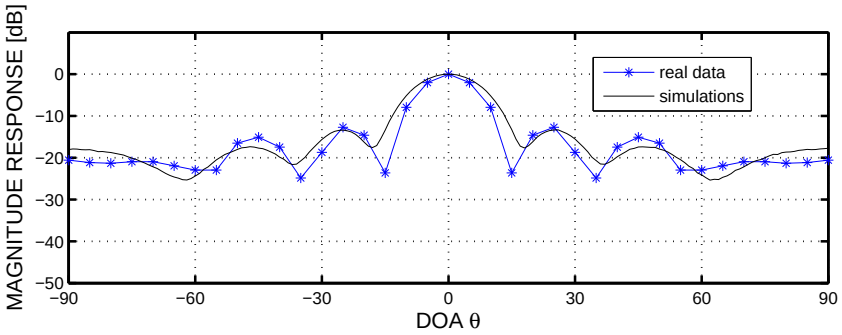

Figure 8.2: Simulations results include the effect of a curved waveform in 6m distance compared to real data recorded in the anechoic chamber

Additionally, short real speech samples were recorded, referring to Experiment Ib. The algorithm was able to recover the signals sufficiently. The reflection of the results is abandoned here. A comprehensive evaluation of the algorithm in terms of signal separation is done in course of the experiments in a multipath environment and described in the following section.

8.2 Reverberant Environment

This section refers to the measurement setup described in Experiment II. It is the final performance test as a lecture hall seems to be a suitable environment for this application. Although, not every relevant parameter can be evaluated, the recorded data constitute a data base enabling the analyses of several aspects of the approach.

In order to keep a maximum of flexibility every source was recorded at its own. Hence, if we want to evaluate the signal separation of several signals two or more recordings are mixed. Eight different sources, whereas four are male and four are female speech samples, are utilized. Before applying the algorithm we need to filter the signal by a bandpass to ensure that the beamformers work properly. The relevant frequency range is $0.4 \leq \Omega \leq 1$, which corresponds to a frequency of $1666\,Hz \leq f \leq 4166\,Hz$. If we use a bandpass filter with a smaller low cut frequency the pass band of the speech signals is increased while the performance of the beamformers decreases. The Matlab GUI provides an input where it is possible to chose the lower cutoff. Another way to let lower frequencies pass through would be a lower sampling rate. This would entail the modification of the array because the distance between two microphones would have to be increased.

The utilized sources have different energies in the relevant frequency range of $0.4 \leq \Omega \leq 1$.

Name of Source	**Energy** [dB]
Female1	0
Female2	-4.5
Female3	-0.5
Female4	-8.0
Male1	-8.5
Male2	-6.5
Male3	-2.0
Male4	-7.0

Table 8.1: signal energy in valid frequency range

An overview of the signals and its energy is given in Table 8.1 whereas the signal with the most energy within this frequency range is supposed to be the reference.

In order to keep generality it is abandoned to include SVD if not declared differently. As there are seven beamformers implemented the BSS algorithm provides seven output signals. The main goal is to test the performance of this algorithm not being affected by the reduction of the signal number. Hence, the right recovered signals have to be found. Calculating the SNR we simply chose the signal or the signals with the best SNR.

The SNR is computed with the intention to compare various impacts affecting the approach. Its calculation turns out to be more sophisticated then assumed. The difficulty is to estimate the noise. The approach of calculating the residual by subtracting the original time signal directly from the recorded one failed. Hence, the SNR has to be calculated in the frequency domain.

8.2.1 Calculation of SNR in Frequency Domain

A Short Time Fast Fourier Transformation (SFFT) is applied first to both signals: The original and the recorded signal. A hanning window with a length of 512 samples, which corresponds to 60ms, is utilized. With a hop size of 256 samples the energy is kept constantly. The Power Spectral Density (PSD) can be calculated as follows:

$$\phi_t(\omega) = \frac{F_t(\omega)\, F_t^*(\omega)}{2\pi} \tag{8.2}$$

where $F_t(\omega)$ is supposed to be the Fourier transformation of a signal at a certain time instance. Note that every 256 samples a new PSD is calculated so alterations in time are considered. If we subtract the PSD of the original signal from the PSD of the recorded one we obtain the residual.

$$\phi_{t,res}(\omega) = \phi_{t,record}(\omega) - \phi_{t,orig}(\omega) \tag{8.3}$$

As a result the development of the power spectral density of the recorded signal in time, the original signal and the residual are obtained. Note that it is required to align the PSDs in time before applying Equation (8.3).

In order to build a meaningful, time dependent energy value every frequency bin for a certain time instance is averaged.

$$E(t) = \frac{1}{N} \int_{\omega=0}^{\pi} \phi_t(\omega) \tag{8.4}$$

$N = 256$ denotes the number of relevant frequency bins. At this stage it is possible to build the ratio of the energies:

$$SNR = 20log_{10}(\frac{E_{orig}}{E_{res}}) \tag{8.5}$$

Due to the ambiguities of the BSS described in Chapter 3.2.2 the recovered signal has to be normalized. If this is done at once for the whole adaption process the result is altering a lot and no convincing results can be gained. In order to evaluate the quality of the coefficients a single set of coefficients is applied to the signal on its complete length. This is done for several points in time. So it can be ensured that not the absolute values, which have no relevance actually, but the development of the relation of the coefficients is considered.

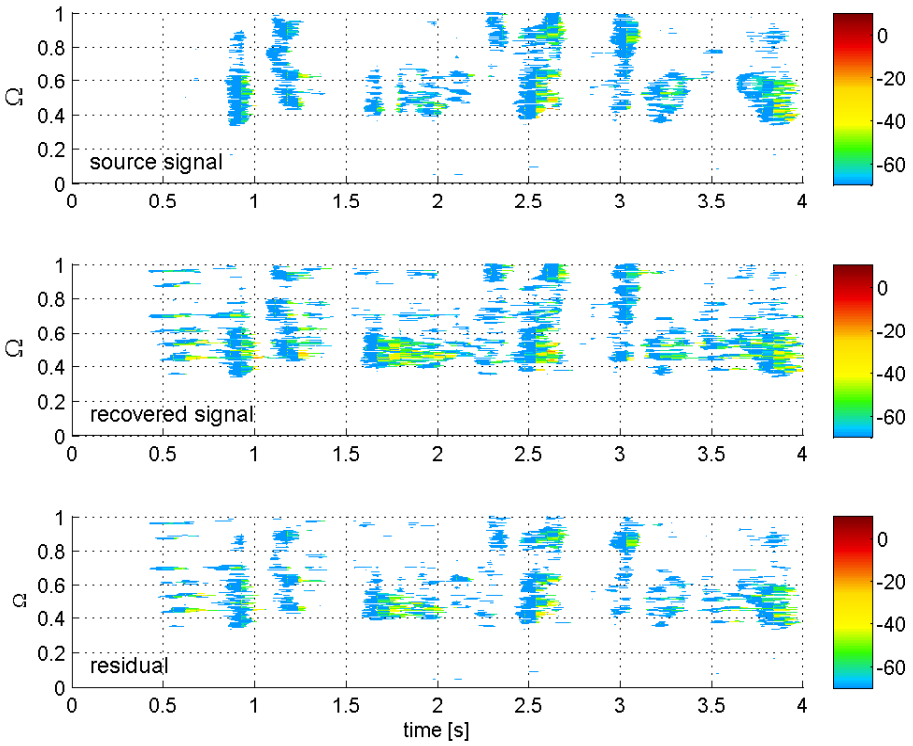

Figure 8.3: 3 spectrograms: 1. corresponding original source signal (male1); 2. recovered signal; 3. residual; original sources: female1 $\theta = 0°$ and male1 $\theta = 40°$

Figure 8.3 depicts the spectrogram of the original signal, the recovered signal, and the corresponding residual, which are the relevant signals for a SNR calculation.

8.2.2 Single Source

Regarding a single speech source the noise in this case consists of echoes as well as noise produced by various other non speech sources. The echoes represent a major part of the noise. Compared to any kind of noise echoes are perceived less disturbing as they contain original information. Therefore, the following SNR calculations are to be treated as relative measure to consider the influence of several parameters.

Figure 8.4, Figure 8.5 and Figure 8.6 show the influence of r, the distance from a source to the array. Depicted is the SNR, which increases slightly while its adaption process. Changing the DOA angle leads to different results, which can be explained by the influence of a curved waveform, explained in Chapter 7.3.

Referring to Chapter 8.2.1 note that a normalization of the reconstructed signal is applied to the data before calculating every single SNR value. For that reason, the alteration of the resulting SNRs is in such a small range. The results are determined with a step size of 0.0001 and a valid frequency range of $0.4 \leq \Omega \leq 1$. The depicted SNRs are an average of 5 trials each, as the start values of the demixing matrix are chosen randomly.

Figure 8.4 shows the development of the SNR for three different distances $r \in [2, 3, 4]\, m$ and a DOA angle of $\theta = -40°$. The utilized source signal is Female1. Due to the fact that the ratio of direct sound to reverberation decreases with an increasing distance the SNR is best at $r = 2m$. Regarding Figure 8.5, the results of a different DOA angle $\theta = 20°$ and a different source Female3 the statement according to Figure 8.4 still remains although the varieties become less.

The third experiment, depicted in Figure 8.6, seems to be in contrast to the former statement. The DOA angle in this case is $\theta = 0°$ whereas the source Male3 is utilized. Effectively, it shows the influence of a curved waveform again. The effect has its biggest impact for a DOA angle of $\theta = 0°$ and converges to zeros for $\theta = -90°$ or $\theta = 90°$. The performance of the beamformer is significantly worse and, therefore, the SNR in distance of $r = 3m$ is better.

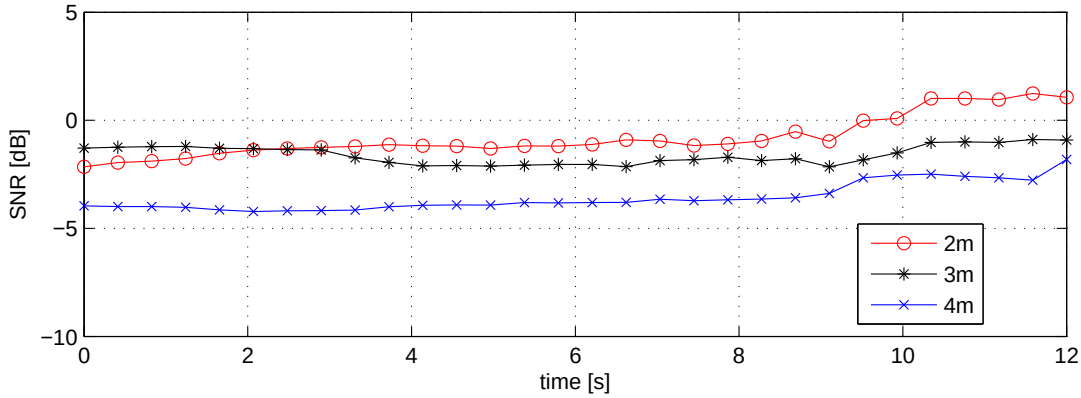

Figure 8.4: SNR of a single signal; varying distances $r \in [2, 3, 4]\, m$; Source Female1; DOA angle $\theta = -40°$;

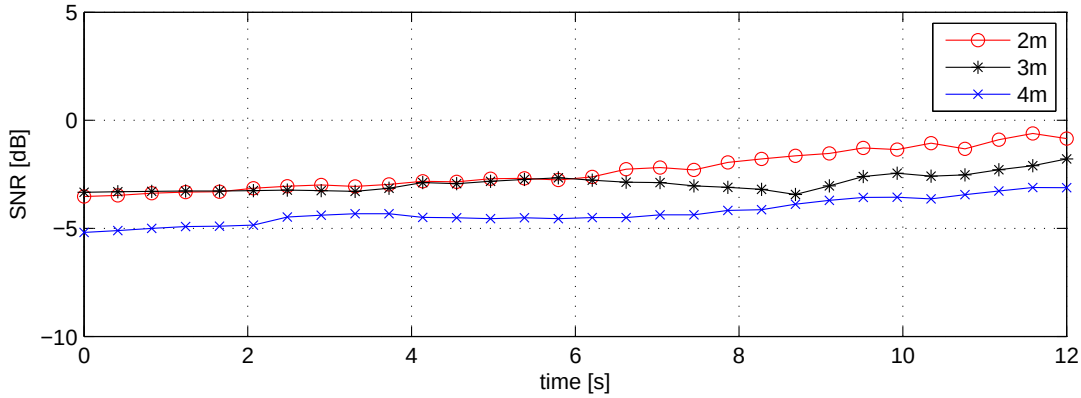

Figure 8.5: SNR of a single signal; varying distances $r \in [2, 3, 4]\, m$; Source Female3; DOA angle $\theta = 20°$;

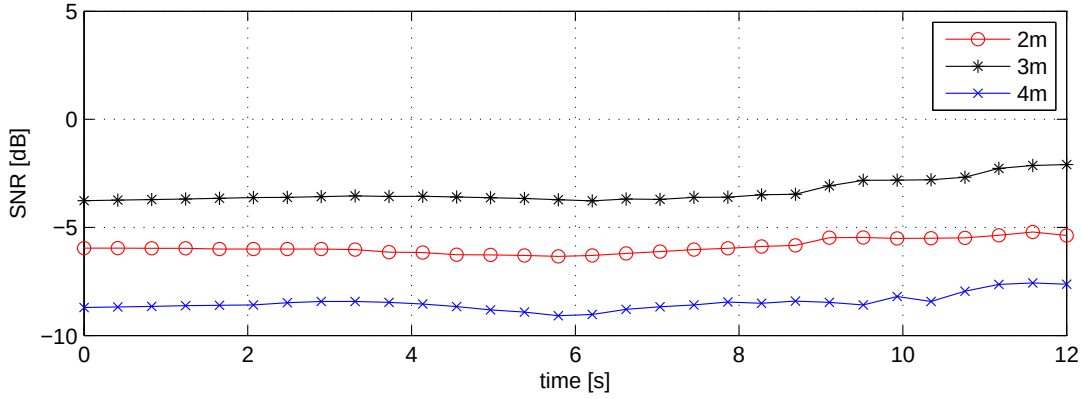

Figure 8.6: SNR of a single signal; varying distances $r \in [2, 3, 4]\, m$; Source Male3; DOA angle $\theta = 0°$;

8.2.3 Multiple Sources

The main purpose of this approach is to separate different speech sources in a multipath environment. This is evaluated in the following section. We consider one speech signal to be our desired signal where all other speech sources are regarded as interferences. Therefore, the former introduced Signal to Noise Ratio (SNR) is renamed Signal to Noise and Interference Ratio (SNIR). The computing of the SNIR remains while SNIR incorporates the appearance of an interfering source.

First of all, we can declare that the proposed algorithm as well as the applied hardware enables a signal separation in general.

An example is shown in Figure 8.7 and Figure 8.8. Referring to Experiment II two independent sources from different DOA angles are supposed to be separated in a reverberant environment. The results of the former Chapter 8.2.2 suggest a constant distance of $r = 3m$. The processed signal contains two sources: Female1, with a DOA angle $\theta = 0°$, and Male1, with a DOA angle $\theta = 30°$. Again, the results are determined with a step size of 0.0001 and a valid frequency range of $0.4 \leq \Omega \leq 1$. Each figure depicts three spectrograms:

1. Spectrogram of both original source signals.
2. Spectrogram of the desired source signal.
3. Spectrogram of the recovered signal.

Comparing the desired and the recovered spectrogram illustrates that the characteristic sequence remains. Referring to Table 8.1 the signal energy of Female1 is significantly higher. Hence, the corresponding SNIR of this signal is better. In Figure 8.8, which shows the spectrogram of the second source with lower energy, parts of the first source can be spotted. In contrast, it is much more difficult to spot parts of the second signal in Figure 8.7, which is mainly caused by the higher energy and the density of the first source itself. This is reflected by the resulting SNIR.

Another interesting question is: How close are two sources allowed to be placed nearby without effecting the performance of the algorithm?

In order to answer this question the following analysis is accomplished. In a scenario with two speakers several SNIRs, which are a function of the relative angle ΔDOA, are calculated. Two sources, whereas one is regarded as desired signal and one as interference are played back simultaneously. The difference of the impinging angle ΔDOA determines the condition of the SNIR. While one source is kept at the broadside of the beamformer (DOA angle $\theta = 0°$) several positions for the second source are tested whereas both SNIRs are computed.

In Figure 8.10 the results are outlined. The red lines represent the signals with a constant DAO angle. The blue lines show the SINRs of sources with a varying angle. Furthermore, the data are summarised by the bold lines. The results are obtained by computing the SINR from five adaptation trials. The last 6 values, which corresponds to 2 seconds, of all five signals are averaged. An example of the calculation process of an appropriate average is shown in Figure 8.9. Again, it is mentioned that the step size is a significant parameter. In order to achieve converging results the step size is set to 0.00015. Figure 8.11 shows the overall average of the appearing signals.

To answer the former question, a good separation seems to be achieved if the sources have at least a angle difference of $\Delta\theta = 20°$. This result is actually no surprise. The main steering directions of the beamforming network are arranged in a distance of about $\Delta\theta_0 \approx 20°$. This corresponds to the measured ΔDOA, where an increasing ΔDOA does not significantly enhance the SNIR. The conclusion is, that the beamwidth of one beamformer respectively the distance of two beams is crucial in terms of a minimum spacing of multiple signals. Hence, increasing the number of beams would enhance the efficiency if sources are in close positions.

In order to evaluate the improvement of our algorithm, the SNIR of a single microphone is calculated. The computation is conducted in the same way as it is done to assess the results of the algorithm. Both are compared in Figure 8.11. A clear improvement can be seen. We can calculate the difference between the SNIRs with and without using the algorithm as follows:

$$\Delta SNIR = SNIR_{algo} - SNIR_{mic} \tag{8.6}$$

where $SNIR_{algo}$ is the signal to noise and interference ratio when the algorithm is applied and $SNIR_{mic}$ the calculated measure for a single microphone. Figure 8.12 shows the improvement in terms of SNIR for the case of two speaking persons in a distance of $r = 3m$. The following overall enhancement of $\Delta SNIR$ can be achieved when neglecting values of $\Delta DOA \in [-10, 0, 10]$:

$$\Delta SNIR \approx 9\, dB \tag{8.7}$$

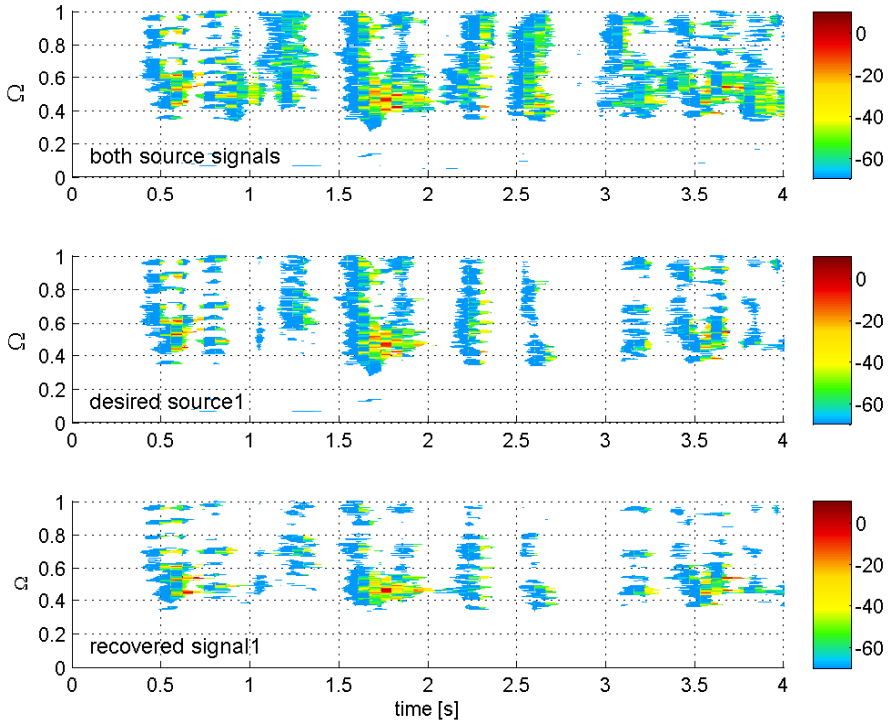

Figure 8.7: 3 spectrograms: 1. original mixed source signals Female1 and Male1; 2. original source1 in order to be recovered; 3. recovered signal; DOA: female1 $\theta = 0°$ and male1 $\theta = 30°$

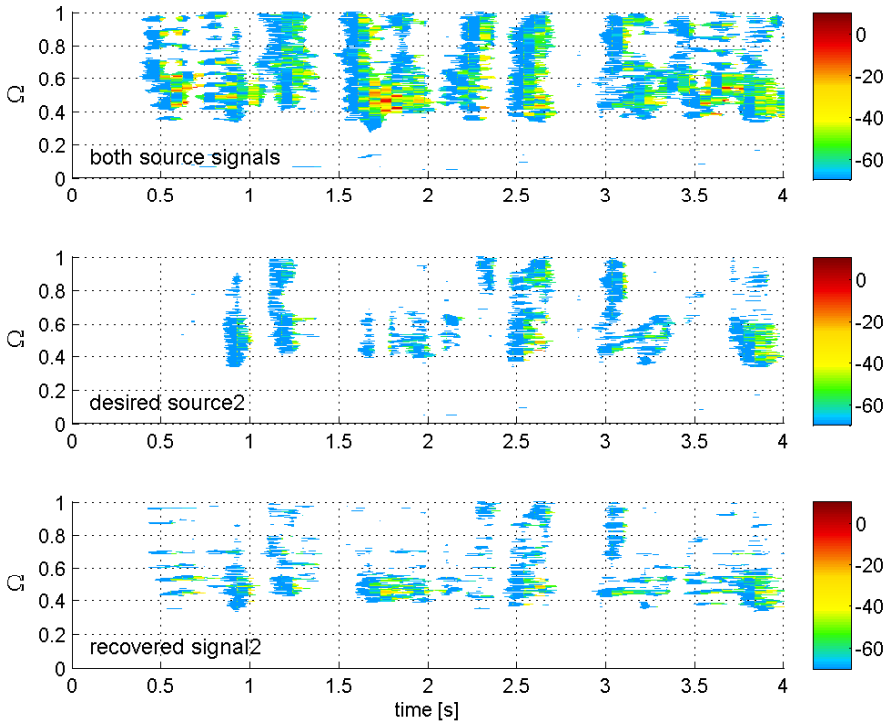

Figure 8.8: 3 spectrograms: 1. original mixed source signals Female1 and Male1; 2. original source2 in order to be recovered; 3. recovered signal ; DOA: female1 $\theta = 0°$ and male1 $\theta = 30°$

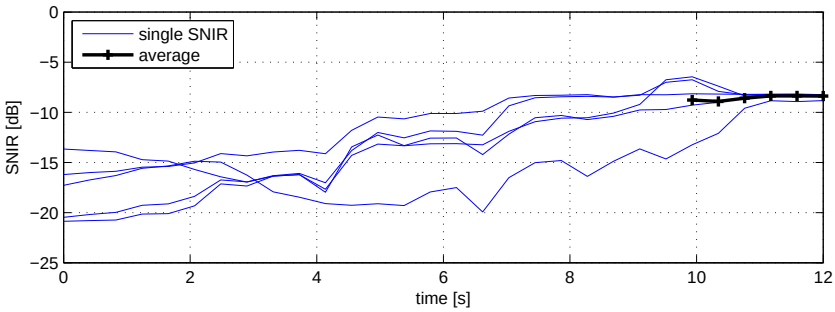

Figure 8.9: SNIR development of five signals with random start values; desired source: Male2 $\theta = 10°$; interfering source: Female2 $\theta = 0°$,

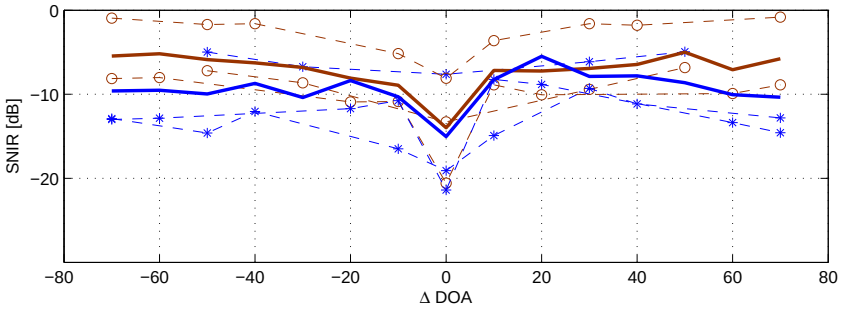

Figure 8.10: SNIR of six signals with fixed distances $d = 3m$ while playing back two sources simultaneously; red lines: signals with constant DOA angle $\theta = 0°$; blue lines: signals with varying DOA angle

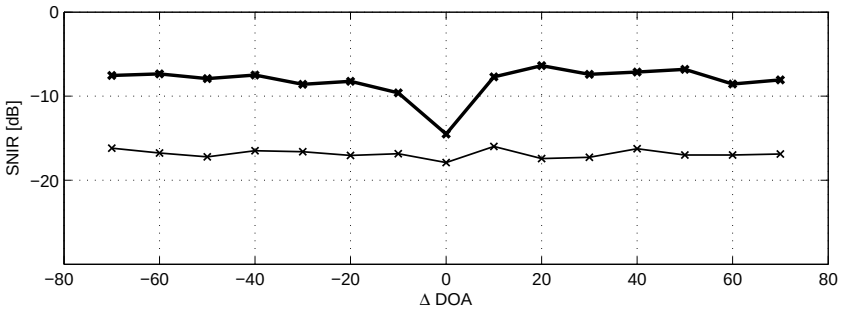

Figure 8.11: averaged results of two corresponding sources with a relative angle ΔDOA; $SNIR$ of signals recovered by the algorithm (bold line) compared to $SNIR$ of signals recorded by one microphone

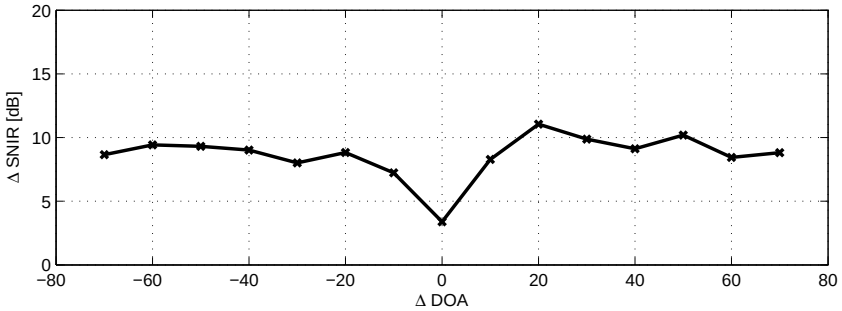

Figure 8.12: SNIR improvment: $\Delta SNIR$ compares the SNIR of signals processed by the algorithm and SNIR of a single microphone; dependent on the relative angle ΔDOA of two corresponding sources; distance 3m

9 Conclusion and Prospects

The scope of this work was to build a system that enables the user to apply the proposed algorithm and to apply access to the results immediately. As a result a fully functional hardware setup as well as an easy to use Matlab GUI was developed.

9.1 Conclusion

It is shown that the proposed algorithm can be applied under real conditions in a reverberant environment whereby the limits of the approach are exposed. The performance of the system is signal and environment dependent. The experiment in a lecture hall demonstrates the dependency on the source location and the relative location of multiple sources in detail. The performance decreases with an increasing ratio of reverberations to direct sound at the location of the microphone array. Therefore, smaller rooms are unsuitable as the source location with a proper ratio can be in a very close position to the array. Noise or interfering signals which are non directional can not be canceled by this algorithm. However, even if the issue of a multipath environment can not fully be solved, the system is capable to enhance the SNIR by suppressing interfering sources effectively. Furthermore, the algorithm is able to cancel a small number of reflected interferences, if the DOA angle does not match with the DOA angle of the desired source.

The Matlab GUI `Beamformer_Gui2_1.m` is a fully functional tool, enabling the user to easily apply experiments in order to evaluate the performance of the algorithm. Due to its modular structure it is suggested to modify or replace certain parts of the algorithm gaining an improvement of the approach.

The single stages of the proposed algorithm can be modified separately. The FIB network and the BSS algorithm are independent and work successively. Hence, a separate development is possible. If one of these parts is improved it can be replaced while retaining the other part. Besides, it is achievable to modify the BSS algorithm in order to recover different kinds of signals, as the applied approach is optimized in terms of speech separation. Thereby, an extension of the valid frequency range of the FIB network would be desirable. Further to this approach, which aims at a good speech discrimination, the frequency range is sufficient.

Different from comparable beamforming systems a priori knowledge of the DOA is not required. Moreover, it should be possible to calculate the DOA angles of the desired sources by evaluating the corresponding set of coefficients. The only a priori knowledge needed is the number of sources L.

Summarizing, the proposed system can be described as unconventional and valid approach in order to separate several speech signals.

9.2 Prospects

The system is supposed to work in real time. To a certain level this requirement is complied. The computation of a certain recording can be performed in a shorter time than the lengths of this recording. The system shall be extended with the final goal of enabling the user to directly listen to the results while one or more persons are speaking. Additionally, the optimization of the system can be further refined. A remaining question is if an increasing number of beamformers leads to a better performance. As block processing is required in order to replay the processed data with a minimum delay it is suggested to abandon the SVD stage. Therefore, a modified BSS needs to be implemented. Suggestions can be found in Chapter 4.4. Furthermore, the constraints of the coefficients respectively the normalization of the signals could become a problem.

In principle, the system including the hardware can be used as basis for a further development of such a real time system. The data acquisition toolbox enables a direct data access in Matlab. It is recommended to implement a software being able to record and replay data virtually simultaneously while a simple FIR filter is applied. If such a application is doable, a block by block implementation of the whole algorithm should be possible.

Last but not least, I want to mention an alternative possibility of an online, Matlab-based signal processing approach. In the course of the diploma thesis [Dietze, 2010], conducted at the Graz University of Technology, a convincing concept, utilizing Matlab for real time data processing is introduced. The Matlab-compatible software 'playrec' is used in order to achieve access to a standardized multichannel audio format such as ASIO (Audio Stream Input/Output), which is provided by a various number of sound cards. This, of course, would require professional audio equipment, which would solve the majority of the hardware caused problems on the other hand.

I hope that my work will be continued and contributes to the research and development accomplished by the Department of Electronic and Electrical Engineering in Sheffield.

Bibliography

[Amari, 1999] Amari, C. (1999). Natural gradient learning for over- and under complete bases in ica. *Neural Computation*, 11:1875–1883.

[Benesty et al., 2008] Benesty, J., Chen, J., and Huang, Y. (2008). *Microphone Array Signal Processing*. Springer.

[Cichocki and Amari, 2002] Cichocki, A. and Amari, S. (2002). *Adaptive Blind Signal and Image Processing*. John Wiley & Sons Ltd.

[Dietze, 2010] Dietze, B. M. (2010). Room response equalization and loudspeaker crossover networks. Master's thesis, Graz, Technical University.

[Dollfuß, 2010] Dollfuß, P. (2010). Driving direction detection with microphone arrays. Master's thesis, Graz, University of Technology.

[Hyvärinen et al., 2001] Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons Ltd.

[Hyvärinen and Oja, 2000] Hyvärinen, A. and Oja, E. (2000). Indenpendent component analysis:algorithms and applications. *Elsevier, Neural Networks*, Vol.13:411–430.

[Joho et al., 2000] Joho, M., Mathis, H., and R., L. (2000). Overdetermined blind source separation: Using more sensors than source signals in a noisy mixture. *Indenpendent Component Analysis and Blind Source Seperation ICA*, pages 81–86.

[Liu, 2010] Liu, W. (2010). Blind beamforming for multi-path wideband signals based on frequency invariant transformation. In *Proc. 4th Int Communications, Control and Signal Processing (ISCCSP) Symp*, pages 1–4.

[Liu and Mandic, 2005] Liu, W. and Mandic, D. P. (2005). Semi-blind source separation for convolutive mixtures based on frequency invariant transformation. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '05)*, volume 5.

[Liu and Weiss, 2008] Liu, W. and Weiss, S. (2008). Design of frequency invariant beamformers for broadband arrays. *Signal Processing IEEE*, 56(2):855–860.

[Liu and Weiss, 2010] Liu, W. and Weiss, S. (2010). *Wideband Beamforming: Concepts and Techniques*. John Wiley & Sons Ltd.

[Oppenheim et al., 2004] Oppenheim, A., Schafer, R., and Book, J. (2004). *Zeitdiskrete Signalverarbeitung*. Pearson Studium.

[Pape, 2005] Pape, L. (2005). Vergleich robuster beamformer arrays. Master's thesis, Graz, Technical University.

[Sekiguchi and Karasawa, 2000] Sekiguchi, T. and Karasawa, Y. (2000). Wideband beamspace adaptive array utilizing fir fan filters for multibeam forming. *Signal Processing IEEE*, 48(1):277–284.

[Vary and Martin, 2006] Vary, P. and Martin, R. (2006). *Digital Speech Transmission*. John Wiley & Sons Ltd.

[Williams, 1999] Williams, E. (1999). *Fourier Acoustics; Sound Radiation and Nearfield Acoustical Holography*. Academic Press.

[Zhang et al., 1999] Zhang, L.-Q., Cichocki, A., and Amari, S. (1999). Natural gradient algorithm for blind separation of overdetermined mixture with additive noise. *Signal Processing Letters IEEE*, 6(11):293–295.