Development and Evaluation of Source Localization Algorithms for Coincident Microphone Arrays

Diploma Thesis
by

Karl Freiberger

at the
Institute of Electronic Music and Acoustics (IEM)
University of Music and Performing Arts Graz, Austria

submitted at
Graz University of Technology

in partial fulfilment of the requirements for the degree of
Diplomingenieur

F750 Electrical Engineering - Audio Engineering

Supervisor: DI Dr. Alois Sontacchi (IEM)
Assessor: Univ.Prof. Mag. DI Dr. Robert Höldrich (IEM)

Graz, April 2010

Abstract

A typical application of microphone arrays is to estimate the position of sound sources. The term microphone array is usually related to an arrangement of several microphones placed at different locations. Within this thesis, however, acoustic source localization (ASL) using coincident - and thus inherently space-saving and handy - microphone arrays is tackled.
Besides established ASL-method based on analyzing the direction of the intensity vector, a pattern recognition approach for ASL is presented. A minimum distance classifier is employed, i.e. feature vectors calculated frame by frame from the array signals are compared with a prerecorded feature-database. The characteristics of the presented approaches are discussed with the help of a mathematical model of first order gradient microphones, as well as with measurements with a planar 4-channel coincident array prototype.

Particular focus is given to robust single speaker-tracking in noisy environments. In this context, several advances to the basic algorithm for improving robustness and accuracy are proposed. In addition to source localization, a brief outline of beamforming using coincident arrays is provided. The performance of the presented ASL-algorithms is experimentally evaluated using array recordings of static and moving sound sources. Different signal to noise ratios are considered. As a basis for quantification of the estimation error, the actual position of the sound source was captured with an optical tracking system.
The results are very promising and show the practicability of the presented algorithms. The similarity approach outperforms the intensity vector approach, in particular at low SNR. At 0 dB SNR (1.8s male speech in a diffuse pink noise field) the azimuth of all (100%) individual frames is correctly estimated if 15° absolute error is allowed (82.5% at 5°, 98% at 10°). The corresponding mean absolute azimuth estimation error is 3°. Though accurate for static sources, the algorithm is able to track rapid azimuth changes.

Keywords: Acoustic source localization, speaker tracking, microphone arrays, B-format, directional microphones, minimum-distance classifier

Zusammenfassung

Mit Hilfe eines Mikrofonarrays ist es möglich, die örtliche Position einer Schallquelle zu bestimmen. Üblicherweise wird dabei unter dem Begriff Mikrofonarray eine Anordnung von räumlich verteilten Mikrofonen verstanden und zur Anwendung gebracht. In der vorliegenden Arbeit werden jedoch Algorithmen zur Schallquellenlokalisation mit einer koinzidenten - und somit prinzipbedingt besonders platzsparenden - Mikrofonanordnung vorgestellt.
Neben einer gängigen Methode zur Richtungsbestimmung über den Intensitätsvektor, wird in dieser Arbeit ein Mustererkennungsverfahren zur Quellenlokalisation vorgeschlagen. Dabei wird ein Minimum-Distanz-Klassifikator verwendet, der blockweise aus den Array-Signalen extrahierte Merkmalsvektoren mit einer Merkmalsdatenbank vergleicht. Die Eigenschaften dieses Ansatzes werden anhand einer mathematischen Modellierung von Gradienten-Mikrofonen erster Ordnung sowie anhand von Messungen mit einem planaren 4-Kanal Array-Prototypen aufgezeigt.
Ein besonderer Schwerpunkt dieser Arbeit liegt auf dem robusten Erkennen einer einzelnen, zeitabhängigen Sprachsignalquelle bei Vorhandensein von Umgebungsgeräusch. Daher spielen Überlegungen zur Verbesserung der Robustheit der Positionsschätzung bei schlechtem Signal-Rauschverhältnis (SNR) eine wichtige Rolle. Neben der Quellenlokalisation mit koinzidenten Arrays wird auch das sogenannte Beamsteering, also das Richten des Aufnahmefokus in eine bestimmte Richtung, kurz vorgestellt.
Die Leistungsfähigkeit der vorgestellten Lokalisationsalgorithmen wird für statische Quellpositionen sowie für eine bewegte Schallquelle evaluiert, wobei unterschiedliche SNR-Situationen Betracht finden. Die Ergebnisse bei der Bestimmung des Azimut-Winkels sind vielversprechend und zeigen die praktische Relevanz der vorgestellten Algorithmen. Der Minimum-Distanz Algorithmus erziehlt besonders bei schlechtem SNR bessere Ergebnisse als der Intensitätsvektor-Algorithmus. Bei 0 dB SNR (1.8 s Sprachsample vs. räumlich diffuses pinkes Geräuschfeld) werden im Versuch alle (100%) Blöcke richtig erkannt, wenn 15° absoluter Schätzfehler zugelassen sind (98% bei 10°, 82.5% bei 5°). Der entsprechende mittlere absolute Winkelfehler ist 3°. Trotz der Genauigkeit bei statischen Quellen, ist der Algorithmus in der Lage sprunghaften Änderungen schnell zu folgen.

Schlagwörter: Akustische Quellenlokalisation, Tracking, Mikrofonarrays, B-Format, Direktionale Mikrofone, Minimum-Distanz-Klassifikator

Contents

List of Figures

Chapter 1

Introduction

The task of acoustic source localization (ASL) is to estimate the spatial position of one or several sound sources given acoustic sensor signals.
In the following, a review of ASL in the context of applications such as teleconferencing systems, intelligent robots and surveillance systems is given. A brief overview of established methods for ASL with microphone arrays is provided. Finally, the coincident ASL approach presented in the remainder of this thesis is motivated and introduced.

1.1 Acoustic Source Localization and Microphone Arrays

Humans are able to localize sound by analyzing the sound pressure picked up at their eardrums. To put it simple, the ears function as sensors and the brain does further processing of information. In technical systems, microphones and appropriate signal processing algorithms are employed for that purpose, respectively. The algorithms are usually implemented in digital domain on computers, particularly on digital signal processors (DSPs). As a sensor front-end, an arrangement of two ore more microphones commonly referred to as a *microphone array* is typically used. However, ASL can[1] also be performed with a single microphone only [45, 49].
An important feature provided by microphone arrays is their ability to pick up sound from a desired direction (signal) well, while sound from other directions (noise) is attenuated. Such spatial filtering is known as *beamforming* or array steering. The particular difference to a single directive microphone is that the steering direction of a beamformer can be changed by mere signal processing rather than by moving the microphones mechanically. Furthermore, appropriate arrays can achieve very high directivity, compared with standard first order polar patterns such as the cardioid

[1] see page 4 for details

pattern[2] [7, 19].

In systems which involve acoustic pickup, beamforming is thus a very popular technology for noise reduction. Noise reduction, in turn, can be indispensable because the performance of many algorithms and applications such as speech recognition, classification of sounds or speaker identification, is often drastically worsened if the signal to noise ratio (SNR) is too low. Poor SNR conditions can be expected in real-world environments, e.g. reverberant and noisy conference rooms, particularly when the sound source is not very close to the microphone. The improvement of SNR obtained with beamforming can therefore be essential to many practical applications.

The localization estimate provided by an ASL algorithm can be used as an input to the beamformer, i.e. the beamformer can be steered automatically in the direction of the sound source. A corresponding system that combines ASL and beamforming is referred to as a *self-steered* microphone array in the following.

In addition to a microphone array beam, a video camera can be automatically directed toward the current speaker [52]. This can be useful in video conferencing and surveillance systems. Another large application area focuses on human-like robots. These need ASL for proper interaction with their environment, in particular with humans [39, 29].

Typically, all these applications require accurate and robust localization results in adverse environments, i.e. noisy and reverberant rooms. Numerous research papers have been dedicated to master this challenge, e.g. [14, 52, 34].

Besides performance, an important issue when it comes to practical implementation is size, cost and practicability of the hardware, i.e. microphones, signal processing unit and wiring. Especially in this respect, the *coincident* microphone array approach presented in this thesis seems to be a very interesting alternative to the more widely used *spaced* arrays.

1.1.1 Spaced arrays

Up till now, most of the research in the field of ASL has been dedicated to spaced arrays. In this thesis, a microphone array is termed *spaced* if the individual microphones are placed at such different locations, that *time differences* can be exploited for ASL.

The key physical basis for ASL with spaced arrays is the propagation delay of sound waves due to the finite speed of sound in air. If the acoustic signal[3] emitted by a sound source is sampled with sensors placed at different locations, a time delay $\tau_{ij}(\boldsymbol{\theta})$ dependent on the source direction $\boldsymbol{\theta}$ occurs between each pair ij of sensors. This *time difference of arrival* (TDOA) can be extracted from the microphone signals by means of appropriate algorithms for time delay estimation (TDE). Common approaches to the TDE-problem are for instance the generalized cross correlation (GCC) [6] and

[2] For more on microphones and polar patterns see chapter 3.1.4

[3] the concept of acoustic signals is detailed in chapter 2.3

adaptive eigenvalue decomposition [10].
Given the array layout and the speed of sound, the *direction of arrival* (DOA) $\boldsymbol{\theta}$ of the incident sound wave can be derived from the time delay $\tau(\boldsymbol{\theta})$. However, simple mapping from TDOA to DOA is only reasonable if certain assumptions on the sound waves radiated by the source can be made. Well known examples are the *near-field* and the *far-field* assumption.
A sound source is said to be in the far-field if the distance between the sound source and the array is much larger than the aperture-size, i.e. the dimensions of the array [15]. Then, a plane wavefront can be assumed and all pairs of adjacent microphones of a linearly spaced array can be expected to have the same time delay. Consequently, using more than 2 sensors brings in redundancy which can be exploited for improving robustness [7]. In the near-field, the wave curvature cannot be neglected and thus different time delays occur. This allows for estimation of the distance between sound source and array. However, assuming far field sources is often favored because this leads to less complex algorithms [41].

Spaced arrays usually consist of several omni-directional microphones. The TDOA principle is however also applicable to directive microphones [44]. Besides planar apertures, circular arrays have been considered [27]. The latter have the advantage of avoiding front-back confusion. Furthermore, there is typically no preferred direction if the array is placed in the center of the region of interest. These benefits are also offered by the coincident array described in the next section.
All spaced arrays do however share the property that, as a matter of principle, there is a relationship between frequency, spacing and performance. This is a major drawback of spaced arrays compared to coincident ones. For good ASL-performance at low frequencies, large distances between the individual microphones are necessary. Therefore, spaced arrays are sometimes quite large and bulky.

1.1.2 Coincident Arrays

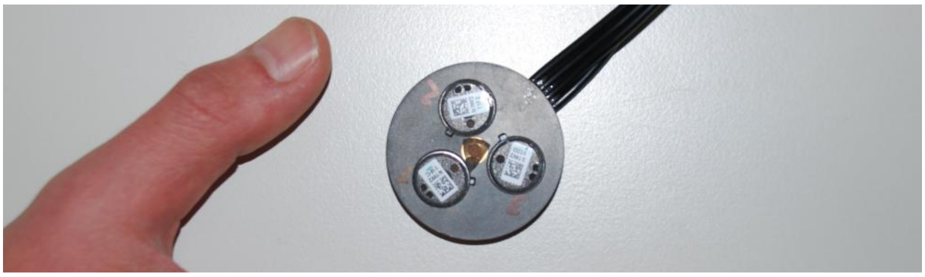


Figure 1.1: Coincident microphone array: prototype array

A *coincident* microphone array (CMA) comprises of two or more microphone capsules placed at the same point, at least as far as this is possible from a manufacturing point of view. Such an array is therefore very handy compared to spaced microphone arrays. Because no time differences occur if all microphones are placed at the same point, the key principle behind ASL with CMAs is to use *level differences* caused by directive microphones oriented towards different directions.

Some researchers have already addressed ASL using coincident arrays. Frequently, a *soundfield microphone* (SFM) is used as the sensor front-end. This arrangement of four tetrahedrally arranged cardioid microphones was devised by M. Gerzon [23] for first order Ambisonic recordings. It is well established and commercially available[4]. The raw cardioid microphone signals, the so called *A-format*, can be converted to the *B-format* which consists of an omni-directional (W) and three figure-eight components that span a Cartesian coordinate system (X, Y, Z) [4]. Given the B-format, the azimuth and elevation angle of the sound source can be estimated by simply mapping the intensity vector components from the Cartesian coordinate system to a spherical one.

In the simplest case, the azimuth angle φ is estimated by relating the levels p_X and p_Y of the X and Y component respectively [16], i.e the estimated azimuth $\hat{\varphi}$ is

$$\hat{\varphi} = \operatorname{atan2}\left(p_Y, p_X\right) \tag{1.1}$$

This can be easily extended to consider time and frequency dependence, which basically leads to a replacement of p_X and p_Y in equation 1.1 by a suitable time-frequency transform of the corresponding microphone channels X and Y. The short time Fourier transform (STFT) [37], the modified cosine transform (MDCT) [26] and wavelet packages [25] have already been applied for that purpose. In this thesis, the coincident ASL approach outlined by eq. (1.1) is referred to as *intensity vector* approach.

Another possibility for ASL with CMAs is to employ the localization principle of *steered response power* (SRP). Using the soundfield microphone signals it is possible to steer a first order directional pattern, e.g. a cardioid, in any desired direction. By directing the beam toward different steering angles, e.g. an azimuthal grid with 1 degree resolution, the source direction can be estimated as that angle on the grid that produces the maximum signal power [24]. SRP - algorithms are widely-used with spaced arrays as well [7].

Especially in the field of human-like robots, efforts have been made to mimic the human sound localization system. Humans localize sound by appraising different cues such as interaural time and level differences (ITD and ILD) as well as so called *monaural cues* [9].

Monaural localization is possible because, due to reflections and diffraction by head, pinna, and torso, an acoustic signal is filtered differently depending on the source location relative to head and torso. Measuring the corresponding filter frequency

[4] www.soundfield.com

responses for different source positions leads to a set of head related transfer functions (HRTFs). By learning the relation between filtering effect and source position monaural localization is possible [45]. However, broadband signals are required for single channel ASL.
The relation between localization cues and the source position can be given by an analytic relation deduced from measurements or model considerations. Another way is to store features belonging to certain reference positions in a table and compare the actual observed feature with it [50, 36]. Moreover, the mapping can be learned by pattern recognition methods [51].

The similarity-based algorithm presented in this thesis picks up these ideas and presents ASL with CMAs in the light of a pattern classification problem. Instead of using a human-like dummy head with two omni-directional microphones, a CMA consisting of three cardioid microphones and one omni-directional microphone is used in this thesis. As can be seen in Figure 1.1, the present prototype array is very small and handy and can for instance be easily placed on a conference table.
Another nice feature of the coincident approach is that no assumptions such as a restriction to far-field sources have to be made. Following the definition of far- and near-field typically used in the context of spaced microphone arrays (see page 3), only far-field sources have to be considered with CMAs, because the aperture-size is theoretically infinitely small and hence always smaller than the wavelength. In other words, no matter whether a point-source or a plane-wave model of sound radiation is assumed, the wave-front curvature on a sufficiently small aperture is the same. The point source model is however advantageous, because not only the direction, but also the distance of the sound source can be specified.
The similarity-based algorithm detailed in chapter 4.1 allows for estimation of the distance of sound sources by utilizing the proximity effect. As outlined in section 3.1.6, the microphone array must be placed in the 'near-field' of the sound source. Here the term 'near-field' is however not related to the array aperture. Instead, a common definition only dependent on the wavenumber $k = 2\pi/\lambda$ and the distance r between sound source and the point of observation [30] is used.

- Near-field: $kr \ll 1$

- Far-field: $kr \gg 1$

This follows from solving the wave equation for sound velocity in spherical coordinates and is directly related to the proximity effect that occurs with pressure gradient microphones. The proximity effect is discussed in detail in section 3.1.6. More on acoustics and the point source model can be found in section 2.3.

1.2 Structural Organisation

In the remainder of this thesis,

chapter 2, *Scientific Background*, provides the foundations for the subsequent chapters. Definitions, principles and model assumptions concerning acoustics, signal processing and identification of linear time-invariant systems are provided.

chapter 3, *Coincident Microphone Arrays*, presents coincident microphone arrays and corresponding mathematical models to describe their behavior. Furthermore, beamforming with coincident arrays is reviewed.

chapter 4, *Localization Algorithm: Foundations*, presents the similarity-based localization principle and discusses simulation results for static, single frequency sources. This reveals many characteristics relevant to the practical algorithms presented in the following chapter. Furthermore the intensity vector approach is reviewed.

chapter 5, *Practical Algorithm*, brings in time dependence and presents practical algorithms for stable localization of non-stationary signals such as speech.

chapter 6, *Evaluation*, evaluates the performance of the presented algorithms with data originating from real-world recordings.

chapter 7, *Summary, Conclusions and Outlook*, summarizes the previous chapters and outlines ideas for further research.

Finally, the *Appendix* describes the implementation of the algorithm in Matlab and contains a detailed parameter list for the corresponding localization function.

Chapter 2

Scientific Background

2.1 Coordinate System

Throughout this thesis, a spherical coordinate system according to Figure 2.1 is used to describe the position of sound sources. The origin of the coordinate system is defined to be the place of the microphone. The position vector $\mathbf{\Theta}$ is defined as

$$\mathbf{\Theta} = \begin{pmatrix} \varphi, & \vartheta, & r \end{pmatrix}^T , \tag{2.1}$$

where φ, ϑ and r are the spherical coordinates

- azimuth angle φ
- elevation angle ϑ
- radius r

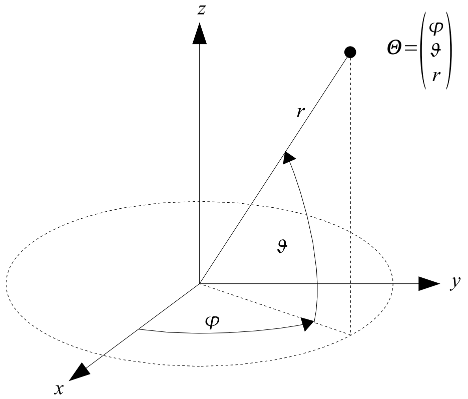


Figure 2.1: Coordinate System

The position of a sound source s is denoted $\mathbf{\Theta}_s$. The transformation from spherical to Cartesian coordinates is given as

$$\begin{aligned} x &= r \cdot \cos\vartheta \cdot \cos\varphi \\ y &= r \cdot \cos\vartheta \cdot \sin\varphi \\ z &= r \cdot \sin\vartheta \end{aligned} \tag{2.2}$$

The transformation from Cartesian to spherical coordinates is

$$\begin{aligned} \varphi &= \mathrm{atan2}(y, x) \\ \vartheta &= \mathrm{atan2}(z, \sqrt{x^2 + y^2}) \\ r &= \sqrt{x^2 + y^2 + z^2} \qquad , \end{aligned} \tag{2.3}$$

where

$$\mathrm{atan2}(y,x) := \begin{cases} \arctan\frac{y}{x} & \text{für } x > 0 \\ \arctan\frac{y}{x} + \pi & \text{für } x < 0,\ y \geq 0 \\ \arctan\frac{y}{x} - \pi & \text{für } x < 0,\ y < 0 \\ +\pi/2 & \text{für } x = 0,\ y > 0 \\ -\pi/2 & \text{für } x = 0,\ y < 0 \\ 0 & \text{für } x = 0,\ y = 0 \end{cases} \tag{2.4}$$

2.2 Signal Processing Preliminaries

This section is intended to introduce the notation of basic relations from linear systems theory and mathematics, needed in the remainder of this paper. For a detailed introduction to signal processing, please refer to the classic textbook by Oppenheim, Schafer and Buck [40].

2.2.1 Continuous Time

The Fourier transform of a continuous time signal $x(t)$ is given as

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \tag{2.5}$$

Given an input signal $s(t)$, the output $x(t)$ of a linear time-invariant system described by its impulse response $h(t)$ is

$$x(t) = s(t) * h(t) = \int_{-\infty}^{\infty} s(\tau) h(t - \tau) d\tau \tag{2.6}$$

Eq. (2.6) is referred to as convolution integral and $*$ is the convolution operator. The corresponding relation in frequency domain is

$$X(f) = S(f) H(f) \tag{2.7}$$

2.2.2 Discrete Time

A sequence $x[n]$ can be obtained from a continuous time signal $x(t)$ by sampling with a samplerate $f_s = 1/T$.

$$x[n] = x(nT) \; , \tag{2.8}$$

where the integer n is the discrete time index.

In this thesis, a *signal of length* N denotes a sequence that is equal to zero outside a finite domain $n = 0, 1, \ldots, N-1$, i.e. its support $\mathcal{S}$ is given as

$$\mathcal{S} = \{0 < n < N-1\} \; . \tag{2.9}$$

The information contained in a sequence $x[n]$ of length N can be put in a signal vector $\underline{x}$.

$$\underline{x} = \Big[x[0], \quad x[1], \quad \ldots, \quad x[N-1]\Big]^T \tag{2.10}$$

The vector notation is helpful for describing operations like for instance sorting of the sequence. $\underline{x}$ is not to be mistaken with the notation of a multi-variate signal $\boldsymbol{x}[n]$ used for instance for conjoint description of all microphone array channels.

$$\boldsymbol{x}[n] = \Big[x_0[n], \quad x_1[n], \quad \ldots, \quad x_{N-1}[n]\Big]^T \tag{2.11}$$

Fourier Transform

The discrete time Fourier transform (DTFT) of a sequence $x[n]$ is

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \tag{2.12}$$

The discrete Fourier transform (DFT) of a signal $x[n]$ of length N is

$$X[k] = \mathrm{DFT}\{x[n]\} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \tag{2.13}$$

The corresponding inverse discrete Fourier transform (IDFT) is given as

$$x[n] = \mathrm{IDFT}\{X[k]\} = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N} \tag{2.14}$$

In practice, the Fast Fourier Transform (FFT) can be used to speed up the computation of the DFT. For use with a radix-2 FFT, N must be a power of 2. The computational complexity increases from N^2 (DFT) to $N \cdot \log_2 N$ (FFT). Since the FFT achieves the same result as the DFT, the operator $\mathrm{FFT}\{\cdot\}$ may be used instead of $\mathrm{DFT}\{\cdot\}$ without affecting the result.
If the signal is real valued, i.e. $x[n] \in \mathbb{R}$, only the first half of the N frequency bins

must to be considered, the other half is redundant. This yields a relevant frequency index range $0 < k < N/2$.
Instead of using the index k, the spectrum can also be looked at in terms of a discrete frequency grid in Hz. The relation between index k and discrete frequency f_k is

$$f_k = kf_s/N \tag{2.15}$$

The frequency spacing, i.e. the distance between two neighboring frequency bins is

$$\Delta_f = f_s/N \tag{2.16}$$

If the samplerate is reduced, the FFT-length N can also be reduced by the same factor without affecting Δ_f. What changes is however the maximum frequency $f_{N/2} = f_s/2$. Because the square brackets clearly indicate discrete frequency, the index k may be omitted, i.e. the notation $X[f]$ is used instead of $X[f_k]$.

Convolution

The linear convolution of two sequences $s[n]$ and $h[n]$ is given as

$$x[n] = s[n] * h[n] = \sum_{m=-\infty}^{\infty} s[m]h[n-m] \tag{2.17}$$

The output spectrum can be obtained by multiplication in frequency domain.

$$X(e^{j\omega}) = S(e^{j\omega})H(e^{j\omega}) \tag{2.18}$$

The convolution of two sequences $s[n]$ and $h[n]$ of finite length N_s and N_h, respectively, is given as

$$x[n] = \sum_{m=0}^{N_{h,s}-1} s[m]h[n-m] \ , \tag{2.19}$$

where $N_{h,s} = \max(N_h, N_s)$. The length of $x[n]$ is $N = N_h + N_s - 1$. Again, the convolution theorem holds:

$$X[k] = S[k]H[k] \tag{2.20}$$

However, for obtaining $X[k] = \mathrm{DFT}\{x[n]\}$, i.e. perform linear convolution using eq. (2.20), $S[k]$ and $H[k]$ must be padded with zeros such that both have the length $N = N_h + N_s - 1$. Zero padding of $s[n]$ to length N yields

$$\tilde{s}[n] = \begin{cases} s[n] & O \leq n \leq N_s - 1. \\ 0 & N_s \leq n \leq N - 1. \end{cases} \tag{2.21}$$

For a concise description, the operator $\mathrm{DFT_N}$ is defined as

$$\mathrm{DFT_N}\{s[n]\} = \mathrm{DFT}\{\tilde{s}[n]\} \tag{2.22}$$

where $\tilde{s}[n]$ is defined according to eq. (2.21). Hence, the operator $\mathrm{DFT_N}$ combines the following two steps

1. zero padding of time domain signal of length $N_s < N$ to length N
2. consecutive DFT of length N

With that, linear convolution in frequency domain can be expressed as

$$x[n] = \mathrm{IDFT}\left\{ \mathrm{DFT_N}\{s[n]\} \cdot \mathrm{DFT_N}\{h[n]\}\right\} \tag{2.23}$$

2.2.3 Sample Statistic

The expected value $\mu_\xi = \mathrm{E}\left\{\xi[n]\right\}$ of an ergodic, discrete time stochastic process $\xi[n]$ is given as

$$\mu_\xi = \mathrm{Mean}\left\{x[n]\right\} = \lim_{N\to\infty} \frac{1}{N+1} \sum_{n=-N/2}^{N/2} x[n] \ , \tag{2.24}$$

where $x[n]$ is a realization (sample) of the process $\xi[n]$.
The variance $\sigma_\xi^2 = \mathrm{E}\left\{(\xi[n] - \mu_\xi)^2\right\}$ is a measure for variation around the mean value. From (2.24) follows

$$\sigma_\xi^2 = \mathrm{Var}\left\{x[n]\right\} = \lim_{N\to\infty} \frac{1}{N+1} \sum_{n=-N/2}^{N/2} \left(x[n] - \mu_\xi\right)^2 \ , \tag{2.25}$$

The arithmetic mean $\bar{x}$ of a finite length sample sequence $x[n]$ is an unbiased estimator of the expected value μ_x.

$$\bar{x} = \mathrm{mean}\big\{x[n]\big\} = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \ , \tag{2.26}$$

where N is the length of the sequence. The sample variance as defined in (2.27) is an unbiased estimator of the population variance σ_ξ^2.

$$\mathrm{var}\big\{x[n]\big\} = \frac{1}{N-1} \sum_{n=0}^{N-1} \big(x[n] - \bar{x}\big)^2 \tag{2.27}$$

The sample standard deviation is given as

$$\mathrm{std}\big\{x[n]\big\} = \sqrt{\mathrm{var}\big\{x[n]\big\}} \tag{2.28}$$

A valuable alternative to the sample mean is the sample median. It is defined as that value of $x[n]$, that separates the higher 50% values from the lower 50%. A more precise definition can be given by examining how the median is actually computed. Sorting the values of $\underline{x}$ by ascending magnitude yields

$$\underline{\acute{x}} = \mathrm{sort}\{\underline{x}\} = \Big[\acute{x}_0 = \min\{x[n]\},\ \ \acute{x}_1,\ \ \ldots\ ,\ \ \acute{x}_{N-2},\ \ \acute{x}_{N-1} = \max\{x[n]\}\Big]^T \tag{2.29}$$

With that, the median can be defined as

$$\operatorname{median}\{x[n]\} = \begin{cases} \acute{x}_{N/2} & \text{if N is odd} \\ \frac{1}{2}\left(\acute{x}_{N/2-1} + \acute{x}_{N/2}\right) & \text{if N is even} \end{cases} \tag{2.30}$$

The median is more robust to outliers than the mean. This is best illustrated with a simple example. Consider the 6-element list $[-1, -1, 0, 1, 1, 180]$. The median is 0.5, whereas the mean is 30. The value 180, which is likely to be an outlier, is suppressed by the median, whereas it has a strong effect on the mean.

2.2.4 Directional Statistics

Processing and analyzing of circular and spherical data requires specific regard. A simple but fundamental problem is for example the difference between two angles. This is needed for computation of the angular estimation error. The values $\varphi_1 = 179°$ and $\varphi_2 = -179°$ yield for instance a difference $\varphi_{\Delta,1} = \varphi_1 - \varphi_2 = 358°$. This high value would indicate that the angles are very different. In fact however, the angles are very close to each other and a value $\varphi_{\Delta,2} = -2°$ would be appropriate. The solution to this problem is is provided by the *principle argument* function [3] which maps a number $\varphi \in \mathbb{R}$ to the interval $]\pi, \pi]$. The principle argument function is defined as

$$\operatorname{princarg}(\varphi) = \operatorname{mod}\{\varphi + \pi, -2\pi\} + \pi \ , \tag{2.31}$$

where

$$\operatorname{mod}\{a, b\} = a - b \cdot \operatorname{floor}\left(\frac{a}{b}\right) \ , \tag{2.32}$$

is the modulo operation and $\operatorname{floor}(x) = \max\{m \in \mathbb{Z} \mid m \leq x\}$ maps a real number $x \in \mathbb{R}$ to the next smallest integer. Referring to the example above, we have[1] $\operatorname{princarg}(358°) = -2°$.

Now consider a list of angles $\underline{x} = [-179, 179]°$. Such a list could for instance be the result of estimating the source azimuth angle at 2 different frequencies. The mean as defined in (2.26) yields $0°$. A more meaningful mean value of $180°$ can however be obtained with the *circular mean* [11] defined as

$$\operatorname{cmean}\{x[n]\} = \arctan\left(\frac{\sum_{n\in\mathcal{S}} \cos x[n]}{\sum_{n\in\mathcal{S}} \sin x[n]}\right) \ , \tag{2.33}$$

where $\mathcal{S}$ is the support of the sequence $x[n]$. The circular mean operator in (2.33) transforms an angular sequence to Cartesian coordinates, takes the linear mean and transforms it back to the corresponding angular value.

[1] Certainly, angles in degree must be converted to *rad* before use with angular functions like sin, cos or princarg by $\varphi_{rad} = \varphi_{deg} \cdot \pi/180$

2.3 Acoustics

2.3.1 Acoustic Model

In this thesis, a simple yet common model for the acoustics of source and media is used: the concept of point source and linear acoustic channel. The model is illustrated in Figure 2.2. In general, sound radiation can be described in terms of a space-time field of an acoustic field quantity $p_s(t, \mathbf{\Theta})$, e.g. sound pressure. Here, t denotes continuous time and $\mathbf{\Theta}$ is a 3-dimensional spatial parameter that defines the point of observation.

A typical model assumption is to claim that acoustic wave propagation can be described by the linear wave equation [2], i.e. that $p_s(t, \mathbf{\Theta})$ obeys

$$\frac{\delta^2 p_s(t, \mathbf{\Theta})}{\delta t^2} - c^2 \Delta p_s(t, \mathbf{\Theta}) = q_s(t, \mathbf{\Theta}) \ , \tag{2.34}$$

where Δ is the Laplace operator, i.e. the sum of second order spatial derivatives. c is referred to as the speed of sound and $q_s(t, \mathbf{\Theta})$ is an excitation term that describes sources of acoustic energy. The linear model in Eq. (2.34) is appropriate for sound propagation in ideal, homogeneous, non-dispersive fluid media at moderate sound levels [35].

The solutions $p_s(t, \mathbf{\Theta})$ can be obtained by a linear systems theory approach [53] : $p_s(t, \mathbf{\Theta})$ can be interpreted as the output of a linear time-variant, space-variant filter with impulse response $g(t, \mathbf{\Theta}; t_s, \mathbf{\Theta}_s)$.

$$p_s(t, \mathbf{\Theta}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} q_s(t_s, \mathbf{\Theta}_s) \ g(t, \mathbf{\Theta}; t_s, \mathbf{\Theta}_s) \ dt_s \ d\mathbf{\Theta}_s \tag{2.35}$$

With the assumptions

- $q_s(t_s, \mathbf{\Theta}_s) = s_s(t) \ \delta(\mathbf{\Theta} - \mathbf{\Theta}_s)$, i.e. the case of an acoustic point source $s_s(t)$.
- time-invariant system behavior: motion, e.g. a moving source, is not allowed.
- the output $p_s(t, \mathbf{\Theta})$ is observed at a fixed position $\mathbf{\Theta} = \mathbf{\Theta}_z = [0, 0, 0]^T$ and is denoted $s(t) = p_s(t, \mathbf{\Theta}_z)$

eq. (2.35) simplifies to

$$s(t) = \int_{-\infty}^{\infty} s_s(t_s) g(t - t_s | \mathbf{\Theta}_s) dt_s = s_s(t) * g(t|\mathbf{\Theta}_s) \tag{2.36}$$

Eq. 2.36 states that the signal $s(t)$ at the place $\mathbf{\Theta}_z$ is given as the linear convolution of a source signal $s_s(t)$ at the place $\mathbf{\Theta}_s$ and the acoustic impulse response (AIR) $g(t|\mathbf{\Theta}_s)$. $g(t|\mathbf{\Theta}_s)$ describes the acoustic transmission channel between the points $\mathbf{\Theta}_s$ and $\mathbf{\Theta}_z$. In practice, $s(t)$ can be figured as the output of an ideal sound pressure transducer and $s_s(t)$ could be seen as the input signal of an ideal loudspeaker. When sound propagation in enclosed spaces is considered, $g(t|\mathbf{\Theta}_s)$ is referred to as *room impulse response* (RIR).

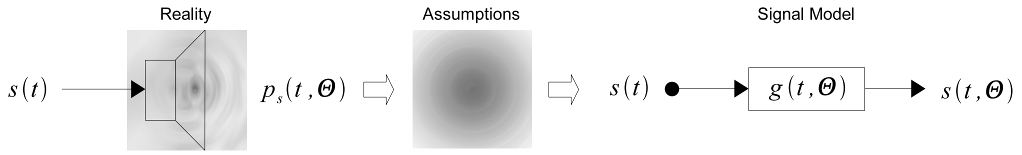


Figure 2.2: A sound source generates a sound field $p_s(t, \mathbf{\Theta})$. With the assumptions given in the text, a signal model can be introduced: A source signal $s(t)$ drives an LTI-system $g(t, \mathbf{\Theta})$ which models the dependence on the place of observation $\mathbf{\Theta}$.

2.3.2 Room Impulse Response

In general, a RIR varies depending on the location of the sound source $\mathbf{\Theta}_s$, which is indicated by the notation $g(t|\mathbf{\Theta}_s)$, i.e. g(t) given $\mathbf{\Theta}_s$. Even so, the acoustic properties of a room are often described by a single impulse response $g(t)$ only. The basic characteristic of a room can be even further broken down to a single scalar value, the well known reverberation time RT_{60} [30].

$g(t|\mathbf{\Theta}_s)$ can be estimated by means of measurements or assessed analytically by means of model assumptions. Figure 2.3 shows a measured IR and an IR generated by an *image source model* [30] of a small room. Appropriate techniques for measuring IRs are discussed in chapter 2.4. Modeling $g(t)$ requires knowledge about acoustic signal transmission. Two types of acoustic signal transmission can be distinguished:

- Single path transmission (SPT): The signal reaches the receiver via a single, direct path (DP). There are no reflections. This is also known as the *free-field* case.

- Multi-path transmission (MPT): The receiver signal is an overlay of the DP-signal and reflections of the signal.

A room impulse response (RIR) $g[n]$ can be divided into the subsets

- Direct sound
- Early reflections
- Late reverberation

The early reflections are a set of discrete reflections, which are often simply assumed as all reflections within the first 50 or 80 ms after the direct sound [48]. Late reverberation usually refers to the huge number of diffuse reflections. These are typically described by a statistical model [30]. In rooms with good intelligibility much energy is contained within the first 50ms after direct sound, i.e. within the early reflections (cf. the definition of the *clarity index* C_{50} [31], which compares the energy within the first 50 ms with the energy received 50 ms after the DP-signal).

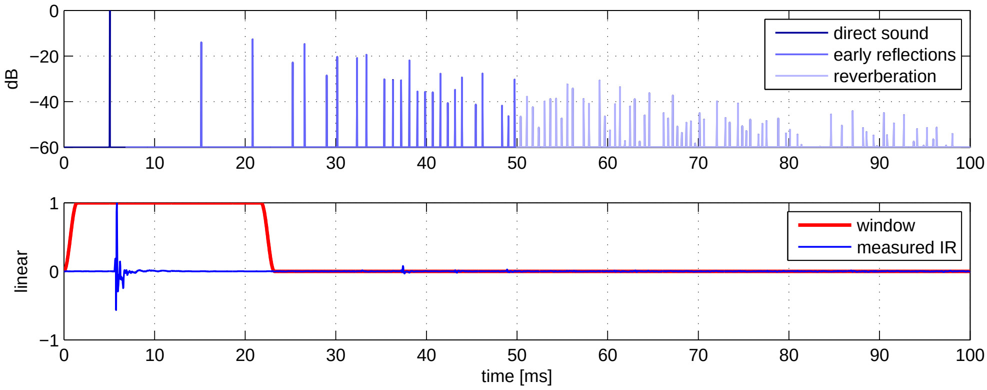


Figure 2.3: Upper picture: schematic IR of a small room as produced by an image source model. Lower picture: IR of a large hall measured with the omni-directional microphone of the array CMA1. The first reflections arrive very late (¿30 ms). The DP-signal is not a perfect impulse $\delta[n]$, because the microphone impulse response $h_0[n|\mathbf{\Theta}_s = (0,0,1)]$ is contained. The DP-signal - and hence $h_0[n|\mathbf{\Theta}_s = (0,0,1)]$ without influence of the room - can be easily extracted by means of windowing.

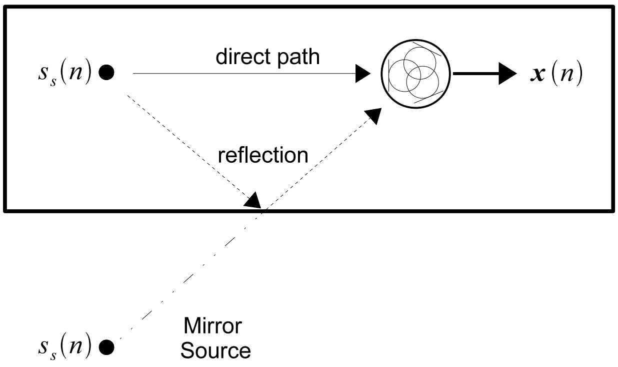


Figure 2.4: Schematic illustration of a single wall reflection. The microphone array has do deal with two sources (the source and its image source) which are placed at different locations, i.e. the reflection arrives from a different direction than the direct path signal.

At a point of observation, e.g the microphone array location, the reflections arrive from different directions than the DP-signal. This is a very important issue for source localization. However, no information on the directions of the reflections is contained in a single RIR.

The directional information can however be obtained from an image-source model. This model is appropriate for early reflections with wavelengths significantly smaller than the texture of the reflecting objects (geometric acoustics) [30]. Figure 2.4 schematically illustrates the image-source model by showing the direct path signal and a single reflection. A reflection can be understood as an additional source placed at a different location than the original sound source. The single source multiple path - problem is hence transformed to a multiple source single path (free-field) model.

Because they arrive from other directions than the DP-signal, reflections are disturbances or *noise*[2] to an ASL-localization algorithm tailored to estimate the position of a single sound source.

In section 4 and 6, the influence of reflections and noise on the localization performance is examined. As outlined above, early reflections can be modeled by adding the array signals of several recordings of the same excitation signal from different locations. To model the absorption properties of the room and different acoustic paths, the reflection signals can be delayed and attenuated or filtered appropriately.

2.3.3 Acoustic Signal Model

Equation (2.36) accounts for both SPT and MPT, since the IR $g(t)$ can include both direct path and reflections. In the free-field case (SPT) we have

$$g(t) = a_0 \delta(t - \tau_0) \tag{2.37}$$

where $\delta(t)$ is the Dirac delta distribution and τ_0 and a_0 delay-time and attenuation due to the distance between source and receiver. Inserting (2.37) in eq. (2.36) yields

$$s(t) = a_0 s_s(t - \tau_0) \tag{2.38}$$

In ASL-research, eq. (2.38) is a widely used signal model for the microphone signal. It is however only proper if ideal pressure transducers are assumed. Since the focus of this thesis lies in incorporating microphone directivity, eq. (2.38) cannot be used directly as a model for a microphone signal. The microphone signal is therefore denoted $x(t)$ instead of $s(t)$.

In the remainder of this thesis, digital signal processing (DSP) algorithms are developed for analyzing the microphone signals. Therefore, a discrete time signal model

[2] Here, the term *noise* refers to the distribution of the directions of arrival of multiple reflections. The reflection signal is of course highly correlated to the direct path signal, i.e. energy is delivered at the same frequencies as it is delivered by the direct path signal.

is advantageous. Though the acoustic signal $s(t)$ is continuous by nature, it is common practice ([10], [6], [7]) to use discrete time notation for it as well. This unifies notation and is no cutback as long as the Nyquist sampling theorem is kept in mind, i.e. no aliasing occurs [40]. Therefore, all further references to time-signal are made in discrete time. The discrete time equivalent of eq. (2.36) is

$$s[n] = s_s[n] * g[n|\mathbf{\Theta}_s] \tag{2.39}$$

where $g[n|\mathbf{\Theta}_s]$ is the impulse response from the source position $\mathbf{\Theta}_s$ to the place of the microphone, i.e. the origin of the coordinate system.

Eq. (2.39) does not explicitly model the direction of the reflections. As outlined above, this can be achieved by a simple discrete reflection (image source) model. If only the first I early reflections are considered, the source signal can be written as

$$s[n] = \sum_{i=0}^{I} s_i[n|\mathbf{\Theta}_i] \ , \tag{2.40}$$

where $s_i[n|\mathbf{\Theta}_i] = s_s[n - N_i] * a_i[n]$ is the i^{th} image source which time delay N_i, frequency dependent attenuation $a_i[n]$. $\mathbf{\Theta}_i$ is the image source direction.

2.4 System Identification

In chapter 2.3 The acoustic channel and the microphone array are both modeled as LTI-systems. In the following, techniques for measuring impulse responses of acoustic systems are reviewed.

2.4.1 Problem formulation

For LTI-systems, the relation between the any input signal $s[n]$ and output $x[n]$ is given by linear convolution

$$x[n] \quad = \quad s[n] * h[n] \tag{2.41}$$

where $h[n]$ is the system impulse response and $*$ is the convolution operator. If $h[n]$ is known, the system is fully identified. In practice, a loudspeaker is used to play back the excitation signal $s[n]$, which is then picked up by the microphone and recorded. Loudspeakers always exhibit slight nonlinear system behavior, i.e. introduce distortions. Additionally, noise is produced by the LS, the microphone and the recording system. In acoustic systems ambient noise comes along. The real input-output relation is thus better described by a 'black box' system $T\{\cdot\}$ with an additional noise term $\eta[n]$.

$$x[n] = T\{s[n]\} + \eta[n]$$

The system $T\{\cdot\}$ comprises of the linear system of interest $h[n]$ and unwanted system $\tilde{T}\{\cdot\}$, that accounts for the recording system described above.

$$T\{s[n]\} = h[n] * s[n] + \tilde{T}\{s[n]\}$$

Different methods exist for estimating $h[n]$ from a measured signal $x[n]$. Their practical value arises mainly from the quality of the estimate $\hat{h}[n]$ in the presence of nonlinearities, transient disturbances and noise, i.e. how well the influence of $\tilde{T}$ and $\eta[n]$ is suppressed. Another point that can be of interest, is the time needed for measurement and post-processing.

Different excitation signals were proposed for identification of acoustic systems. The basic requirement is that the excitation must provide sufficient energy at all frequencies of interest. Another desired feature is that the signal should be deterministic, i.e. exactly reproducible. Established excitation signals that meet these demands are

- MLS (Maximum Length Sequence)
- ESS (Exponential Sine Sweep)

A well known method for system identification with linear sweeps is time delay spectrometry (TDS). The pros and cons of these methods are well studied [38], [47]. For our purpose, the ESS - method proposed by Farina [20] seems well suited. The ESS - method makes it possible to separate the linear and nonlinear part of the system [21], i.e. the influence of loudspeaker nonlinearity can be effectively suppressed. The sensitivity of the ESS-method to transient disturbances is no problem in our case, because the room is supposed to be quiet during the measurements.

2.4.2 IR - measurement with exponential sine sweeps

An ESS signal that crosses an instantaneous frequency of f_1 Hz at a time $t = 0$ and f_2 Hz at $t = T$ seconds is given as

$$s(t) = \sin\left[A\left(e^{\frac{t}{\tau}} - 1\right)\right], \tag{2.42}$$

where $\tau = \frac{T}{\ln\frac{f_2}{f_1}}$ and $A = \tau \cdot 2\pi f_1$. The discrete time equivalent of the sweep $s(t)$ is given as

$$s[n] = \sin\left[A\left(e^{\frac{n}{f_S \cdot \tau}} - 1\right)\right] \ . \tag{2.43}$$

Typically, a finite length sequence $n = 0, \ldots, N-1$, with $N = T \cdot f_s$ is considered. With that, it makes sense to refer to f_1 as the start frequency and f_2 as the stop frequency of the sweep. T is the sweep duration.

The sweep response $x[n]$ of an LTI system $h[n]$ is given in eq. 2.41. Deconvolution must be performed to obtain $h[n]$ from excitation $s[n]$ and $x[n]$. This can be accomplished either in time domain or frequency domain. From $X[f] = S[f]H[f]$ follows

$$H[f] = \frac{X[f]}{S[f]} \tag{2.44}$$

The impulse response is given as

$$h[n] = \mathrm{IDFT}\{H[f]\} \tag{2.45}$$

If $s[n]$ is of length N_s and $h[n]$ of length N_h, linear convolution yields a signal $x[n]$ of length $N_x = N_s + N_h - 1$. In practice, the sweep response $x[n]$ is truncated when it reaches the noise-floor (e.g. RT_{60} seconds after the excitation stops). This defines N_x. To allow for the point-wise division in (2.44), $s[n]$ must be zero-padded to N_x samples.

$$S[f] = \mathrm{FFT}_{\mathrm{N_x}}\{s[n]\} \tag{2.46}$$

Chapter 3

Coincident Microphone Arrays

3.1 Microphones

Microphones are transducers that convert acoustic sound waves into electric signals. A categorization into different types of microphones can be made according to the

- *Principle of operation*, e.g. condenser, dynamic
- *Physical quantity* to be measured, e.g. pressure, pressure gradient
- *Directivity pattern*, e.g. omni-directional, cardioid

In this section, the common concept of directivity patterns is extended to a more general *position dependent LTI-system* description, that incorporates

- Directivity
- Frequency dependence
- The proximity effect

The directivity of a microphone describes its sensitivity with respect to the direction $\boldsymbol{\theta} = (\varphi, \vartheta)$ of sound incidence. The proximity effect explains the influence of the distance r between sound source and a gradient microphone on the frequency response.

Similar to section 2.3, a system model that relates an input signal $s[n]$ to an output signal $x[n]$ is established in the following. Ideal microphone equations are compared with measurements of the prototype array.
In order to avoid digression, standard material concerning the principle of operation, i.e. transduction mechanism and construction of microphones, is not replicated here. The interested reader is referred to [17] and [8].

3.1.1 Position Dependent LTI Model

Because of directivity and the proximity effect, a microphone IR depends - in general - on the source position, i.e. a microphone has a position dependent impulse response (PDIR). In this thesis, a microphone is hence modeled as a system that relates an input signal[1] $s[n]$, to an output signal $x[n]$ according to

$$x[n] = s[n] * h[n|\mathbf{\Theta}_s] \tag{3.1}$$

where $h[n|\mathbf{\Theta}_s]$ is PDIR of the microphone, i.e. the microphone IR given a source position $\mathbf{\Theta}_s$. $x[n]$ is the microphone signal and $s[n]$ is the acoustic signal at the place of the microphone, i.e. the origin of the coordinate center, due to a point source $s_s[n]$ placed at a position $\mathbf{\Theta}_s$. In eq. (3.1) free-field transmission is assumed. As outlined in eq. (2.40) in the previous section, multi path transmission can be however be modeled as a sum of mirror sources under free-field conditions. The microphone output with several sound sources $s[n|\mathbf{\Theta}_i]$ from different positions $\mathbf{\Theta}_i$ can therefore be expressed as a sum over all individual contributions.

$$x[n] = \sum_i s[n|\mathbf{\Theta}_i] * h[n|\mathbf{\Theta}_i] \tag{3.2}$$

For a time-invariant system description as in eq.(3.1), a static source, i.e. no change in the position $\mathbf{\Theta}_s$ must be assumed. Considerations on moving sources are made in section 3.1.2.

Applying the convolution theorem [40] to eq. (3.1) leads to the following expression in discrete time Fourier transform (DTFT) domain.

$$X(e^{j\omega}) = H(e^{j\omega}|\mathbf{\Theta}_s)\; S(e^{j\omega}) \tag{3.3}$$

In practice, the position dependent frequency response (PDFR) $H(e^{j\omega}|\mathbf{\Theta}_s)$ can be measured by means of linear system identification methods as presented in section 2.4.

3.1.2 Moving Sources

A time-invariant system description as in eq. (3.1) requires that the sound source is static. With the help of frame-wise signal processing, tracking of moving sources can however be tackled. This only implies the assumption that the source is static within one frame. In the following, it is roughly assessed whether this assumption is feasible in practice or not.

Consider a sound source that is moving around the microphone array along a circle with radius r at a constant tangential speed v. Moving once along the whole circle

[1] The discrete time signal description is introduced in section 2.3.2

takes $T_{2\pi} = 2\pi r/v$ seconds. Hence, in T_l seconds the azimuthal range $\Delta\varphi$ that is passed is

$$\Delta\varphi = \frac{v \cdot T_l}{r}\,\mathrm{rad} \tag{3.4}$$

Evaluating eq. (3.4) for a typical human walking speed [29] $v = 1\,\mathrm{m/s}$, $r = 1\,\mathrm{m}$ and a frame duration $T_l = 23\,\mathrm{ms}$ results in $\Delta\varphi \approx 1.3°$. Since the IR of a microphones (and the room) does typically not change significantly if the angle changes by such a small value, the source may be fairly assumed to be static within a frame of duration T_l.

3.1.3 Omni-directional Microphones

In air, sound pressure is a non-directional, i.e. scalar, field quantity [35]. This means that, at any time t, the pressure $p(t, \mathbf{\Theta})$ is independent of the direction of a sound wave traveling trough the point $\mathbf{\Theta}$. As a consequence, an ideal transducer that responds to sound pressure does not exhibit any directional behavior and is therefore called omni-directional (omni). The PDFR of such a perfect pressure microphone is thus a constant factor K_α, independent of the position and frequency.

$$H_0(f|\mathbf{\Theta}) = K_\alpha \tag{3.5}$$

Real manufactures, however, do not exhibit a perfectly uniform sensitivity over all directions and the whole audible frequency range. Especially at high frequencies diffraction effects become significant and a preferential direction can be observed [17]. Figure 3.1a shows a measured polar pattern of the pressure transducer of the prototype array. This microphone is very small and thus the omni-directional behavior is well pronounced up to the 8 kHz frequency band.

3.1.4 First Order Microphones

The polar pattern of a first order pressure gradient microphone, i.e. a transducer responding to the pressure gradient ∇p, can be derived to [12]

$$H_1(\boldsymbol{\theta}) = \cos(\varphi)\cos(\vartheta)\ , \tag{3.6}$$

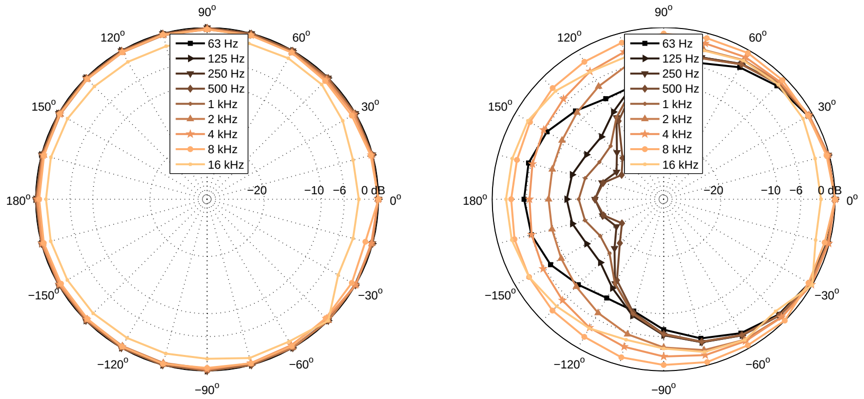
where $\boldsymbol{\theta} = (\varphi, \vartheta)$ is a vector of angular coordinates φ and ϑ. Eq. (3.6) is known as the *'figure 8'*, 'dipole' or 'bidirectional' response.

By a weighted summation of the omni-directional and the bidirectional pattern in eq. (3.5) and eq. (3.6) respectively, different intermediate patterns can be realized. Cotterell [12] refers to such microphones as *first order microphones* and this nomenclature is adopted here. The directivity of an ideal first order microphone with look-direction, i.e. direction of maximum sensitivity, $\boldsymbol{\theta_s} = (\varphi_s, \vartheta_s)$ is given as

$$H_\beta(\boldsymbol{\theta}) = (1 - \beta) + \beta\cos(\varphi - \varphi_s)\cos(\vartheta - \vartheta_s)\ , \tag{3.7}$$

where the parameter β , $0 \leq \beta \leq 1$ controls the type of first order directivity. Certain values of β achieve familiar polar patterns such as 'cardioid' or 'hypercardioid'. In particular, $\beta = 0$ gives the omni-directional and $\beta = 1$ the bidirectional response. Table 3.1 lists some well-established patterns, the corresponding β, directivity factor Q and directivity index DI.

As mentioned above, a polar pattern similar to eq. (3.7) can be generated by combining the output of a separate omni-directional and figure-8 microphone capsule. In practice, it is however more common to achieve the directivity with a single capsule and appropriate design of acoustic delay-paths, because this is usually less expensive [31].



(a) 'Omni-directional' microphone. The omni-directional pattern of the pressure transducer is well pronounced up to and including the 8 kHz band.

(b) 'Cardioid' microphone. The frequency dependence is clearly visible: At low frequencies f, the proximity effect causes an off-axis (180°) boost. For high f, the pattern becomes broad. Only 3 dB off-axis attenuation are achieved at 8kHz.

Figure 3.1: Polar plots of prototype array microphones for different frequency bands. Microphone and speaker were placed on the floor (AKG1-measurement: elevation $\vartheta = 0°$, distance $r = 1m$).

3.1.5 Directivity Measures

The directivity factor[2] Q of a microphone $H(\boldsymbol{\theta})$ is defined as the ratio between the sound power pick-up of a perfect omni-directional microphone to that of $H(\boldsymbol{\theta})$, assuming the same sensitivity in look direction of $H(\boldsymbol{\theta})$ [31], [17].

$$Q = \frac{4\pi}{\iint_S |H(\boldsymbol{\theta})|^2 \, d\boldsymbol{\theta}} \ , \tag{3.8}$$

[2] The German translation of directivity factor is *Bündelungsgrad*

Name	β	Q	DI
omni-directional	1	1	0 dB
figure eight	0	3	4.77 dB
cardioid	0.5	3	4.77 dB
supercardioid	0.63	3.732	5.72 dB
hypercardioid	0.75	4	6.02 dB

Table 3.1: Common microphone directivity patterns.

where $\iint_S d\boldsymbol{\theta} = \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} \cos(\vartheta)\ d\vartheta\ d\varphi$. The numerator in eq. (3.8) is the surface of a sphere with radius $r = 1$, i.e. $\iint_S 1\ d\boldsymbol{\theta} = 4\pi$. Evaluating Q for an ideal first order microphone as defined in eq. (3.7) yields

$$Q = \frac{3}{4\alpha^2 - 2\alpha^2 + 1}$$

The directivity index is defined as

$$DI = 10\log_{10}(Q) \tag{3.9}$$

3.1.6 Proximity Effect

Microphones that have a pressure gradient component, e.g. the first order microphones described in section 3.1.4, boost low frequencies of sound sources that are very close to the microphone. This is known as the proximity effect. The PDFR of an ideal gradient microphone including the proximity effect is [12]

$$H_1(f, \boldsymbol{\Theta}) = \cos(\varphi - \varphi_0)\cos(\vartheta - \vartheta_0)\frac{1 + jkr}{jkr}\ , \tag{3.10}$$

where $k = 2\pi f/c$ is the wavenumber, $\boldsymbol{\Theta} = (\varphi, \vartheta, r)$ is the position of the sound source and (φ_0, ϑ_0) is the look-direction of the microphone. $H_1(f, \boldsymbol{\Theta})$ can be split into two parts, i.e.

$$H_1(f, \boldsymbol{\Theta}) = H_1(\boldsymbol{\theta})\tilde{B}(kr) \tag{3.11}$$

$H_1(\boldsymbol{\theta})$ is the bidirectional polar pattern as defined in eq. (3.6) and the factor

$$\tilde{B}(kr) = \frac{1 + jkr}{jkr} \tag{3.12}$$

describes the proximity effect, i.e. the dependence on the product kr.
The magnitude of $\tilde{B}(kr)$ is termed *boostfactor* $B(kr)$ [12] and given as

$$B(kr) = |\tilde{B}(kr)| = \frac{\sqrt{1 + k^2r^2}}{kr} \tag{3.13}$$

It is worthwhile to examine the following two cases:

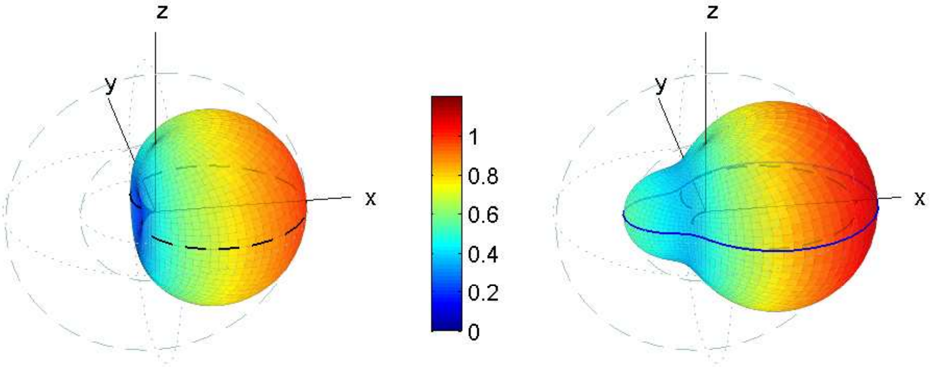


Figure 3.2: Proximity Effect: The picture on the left shows $|H_{0.5}(f, \mathbf{\Theta})|$ for $f = 10\,\mathrm{kHz}$ and $r = 0.5\,\mathrm{m}$, i.e. $kr \approx 100$. It virtually resembles the polar pattern of the ideal cardioid microphone without the proximity effect considered. Right: $f = 100\,\mathrm{Hz}$, i.e. $kr \approx 1$. The boost compared to the left picture is clearly visible and especially pronounced off-axis, i.e. at 180°.

- $kr \gg 1$: The microphone is located in the far-field of the sound source. There is no boost, i.e. $B(kr) = 1$. This is the case if the distance r between sound source and the microphone is large compared to the wavelength $\lambda = 2\pi/k$.
- $kr \ll 1$: The microphone is located in the near-field of the sound source. From eq. (3.13) follows $B(kr) \approx \frac{1}{kr}$. This means that, at a fixed distance in the near-field, the microphone output rises by a factor of 2 (6 dB) if the frequency drops by one octave. This is the well known bass boost at close distances.

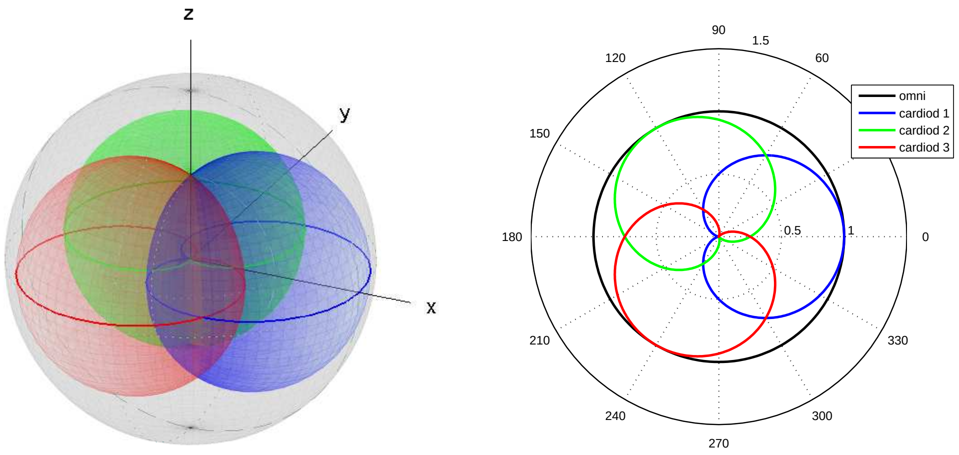
Rewriting eq. (3.7) with a gradient component $H_1(f, \mathbf{\Theta})$ including the proximity effect yields a frequency dependent first order microphone equation.

$$H_\beta(f, \mathbf{\Theta}) = (1 - \beta) + \beta\; H_1(f, \mathbf{\Theta}) \tag{3.14}$$

As pointed out before, the figure-8 component $H_1(f, \mathbf{\Theta})$ boosts for low kr. Hence, for small kr a first order microphone, e.g. a cardioid, behaves more and more like a figure-eight microphone. This effect can be seen in Figure 3.2, where eq. (3.14) is visualized.

3.2 Coincident Microphone Arrays

In the introduction (section $1.1.2$) a coincident microphone array (CMA) was defined as an arrangement of two or more microphone capsules placed at the same point, at



(a) Linear polar plot (3D): The sensitivity for sources from above, i.e. $\vartheta = \pi$, is the same for all microphones.

(b) Linear polar plot (2D): The look directions of the cardioid microphones $\varphi = (0^\circ, 120^\circ, 240^\circ)$ are clearly visible.

Figure 3.3: Model of the planar prototype array shown in Fig. 1.1. Ideal microphone equation (3.7) were used for generating the depicted polar plots. The array consists of 1 omni-directional and 3 cardioid microphones with azimuth look-directions 0°, 120° and 240° , respectively. This setup is named CMA1 and used throughout this thesis.

least as far as this is possible from a manufacturing point of view. For theoretical considerations, it is useful to employ an idealized mathematical description for a CMA, where perfect coincidence is assumed.

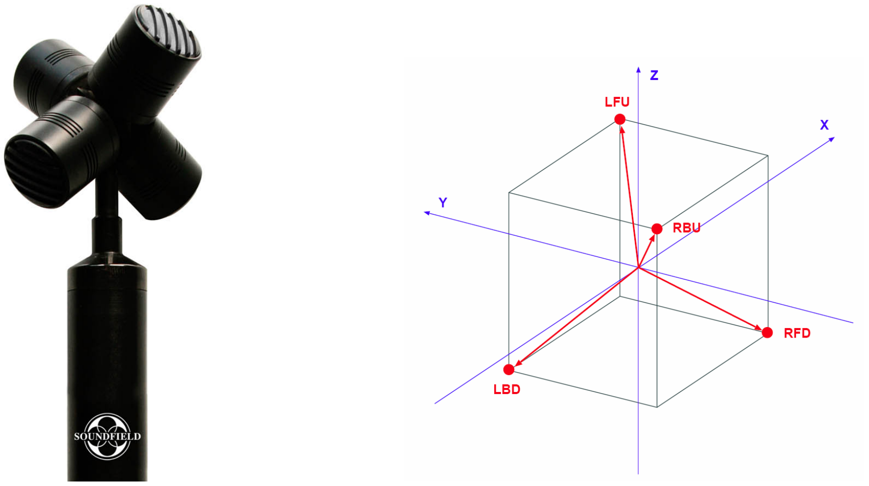
As outlined in the previous chapter, first order microphones can be described by eq. (3.14). Consequently, the PDFR of an individual microphone belonging to an ideal CMA is

$$H_m(f, \mathbf{\Theta}) = (1 - \beta_m) + \beta_m cos(\varphi - \varphi_m) cos(\vartheta - \vartheta_m) \tilde{B}(kr) \ , \qquad (3.15)$$

where the integer m $m = 0, \dots, M-1$ is the channel index. A coincident microphone array can be defined by specifying the parameters in (3.15). As an alternative to eq. (3.15), measured PDFRs can be used for a description closer to reality. In that case, $H_m(f, \mathbf{\Theta})$ is only given at a certain, discrete positions $\mathbf{\Theta}$.

3.2.1 Prototype Array

The planar configuration depicted in Figure 3.3 forms the basis of the prototype array used for practical recordings and measurements throughout this thesis. This array configuration is referred to as CMA1 the following. The CMA1 consists of 4 microphones: 3 cardioids which are directed towards different directions. The cardioid look-directions of the CMA1 are $\underline{\varphi} = (0, 120, 240)^\circ$. The corresponding parameters for the array microphone equations in eq. (3.15) can be found in Table 3.2.



(a) SoundField SPS200

(b) Array configuration (from [1])

Figure 3.4: Tetrahedral microphone array

3.2.2 Tetrahedral Array

Another important coincident array is the following tetrahedral configuration, also known as *soundfield microphone*[3] (SFM). It is depicted in figure 3.4b and defined in Table 3.2[4]. The raw cardioid microphone signals are referred to as *A-format* signals.

Array	m	M	β_m	φ_m	ϑ_m
CMA1	$0,\ldots,M-1$	4	$[0,1,1,1]/2$	$2\pi/3\cdot(m-1)$	0
SFM	$1,\ldots,M-1$	5	$[1,1,1,1]/2$	$\pi/4\cdot(2m+1)$	$\pi/4\cdot(-1)^m$

Table 3.2: Array configuration for prototype array (CMA1) and sound field microphone (SFM) according to eq. (3.15).

By matrixing and filtering they can be converted to the B-format which comprises of an omni-directional and 3 orthogonal figure-8 components. This is described in more detail in the next section, sec. 3.3.

3.3 Steering of Coincident Arrays

The focus of the following review of beamforming with CMAs is restricted to the first order arrays presented in the previous chapter, in particular to the prototype

[3] www.soundfield.com

[4] As can be seen in Table 3.2, for the SFM, the microphone index m starts at 1 and not at 0 as with the CMA1. This is preferential for a general description of algorithms both valid for CMA1 and SFM, because the channel $m = 0$ is a pressure transducer in case of CMA1, whereas the soundfield microphone comprises of cardioids only. $m = 0$ was chosen to denote the omnidirectional channel of CMA1 for keeping consistency with previous practical work.

array CMA1. The approach and nomenclature is however based on established work on spherical arrays (e.g. [19],[55]) and could therefore be easily adopted to arrays of higher order.

It is well known that a circular planar array such as the CMA1 can steer a first order directivity pattern toward a desired azimuth angle φ_s [13]. The SFM additionally allows for specifying a desired elevation angle ϑ_s. Steering a first order directivity pattern means that a 'virtual' first order microphone

$$H_\beta(\boldsymbol{\theta}) = (1-\beta) + \beta\cos(\varphi - \varphi_s)\cos(\vartheta - \vartheta_s)\ , \tag{3.16}$$

can be computed for any desired steering angle $\boldsymbol{\theta_s} = (\varphi_s, \vartheta_s)$ and polar pattern type β. In the case of the planar CMA1 ϑ_s is fixed to $\vartheta_s = 0$. The single-channel beamformer signal $x_s[n]$ is obtained by weighting and summing the array channels.

$$x_s[n] = \sum_m w_m x_m[n] = \boldsymbol{w}^T \boldsymbol{x}[n] \tag{3.17}$$

In the following it is shown how to obtain the steering vector $\boldsymbol{w}$ from β, $\boldsymbol{\theta_s}$ and the array configuration $\boldsymbol{\theta_m}$. The steering procedure can be split in two parts [19]:

1. Eigenbeamforming: The microphone signals are transformed into an orthogonal space. Typically, spherical harmonics are used as basis functions. In the first order case this leads to an omnidirectional and three (3D) orthogonal figure-8 components.

2. Modal beamforming: The eigenbeam-signals are weighted and summed according to the desired steering angle and beampattern shape.

3.3.1 Eigenbeamforming

A well known example for eigenbeamforming is the matrixing involved in A- to B-format conversion of soundfield microphone signals. As the term *matrixing* indicates, the mapping from the original signal space $\tilde{\boldsymbol{x}}$ to a new space $\boldsymbol{\chi}$ is achieved by means of matrix multiplication

$$\boldsymbol{\chi} = \boldsymbol{Y_n}\tilde{\boldsymbol{x}} \tag{3.18}$$

The signal vector $\tilde{\boldsymbol{x}}$ contains the directional microphones, i.e. $\tilde{\boldsymbol{x}} = [x_1, \ldots, x_{M-1}]^T$. The matrix $\boldsymbol{Y_n}$ is given as

$$\boldsymbol{Y_n} = \Big[\boldsymbol{Y}_1, \quad \ldots \quad, \boldsymbol{Y}_{M-1}\Big] \tag{3.19}$$

If the rows $\boldsymbol{Y}_m$ are defined

$$\boldsymbol{Y}_m = \left[\tfrac{1}{\sqrt{2}}, \quad \cos(\varphi_m)\cos(\vartheta_m), \quad \sin(\varphi_m)\cos(\vartheta_m), \quad \sin(\vartheta_m)\right]^T\ , \tag{3.20}$$

where $m = 1, \dots, M-1$ is the microphone index and (φ_m, ϑ_m) is the look direction of the m^th cardioid microphone, the resulting signal vector $\boldsymbol{\chi}$ comprises of spherical harmonic eigenbeams[5], i.e. an omnidirectional channel (w), and three figure-8 components in x-, y- and z-direction, respectively.

$$\boldsymbol{\chi} = \begin{bmatrix} \chi_w, & \chi_x, & \chi_y, & \chi_z \end{bmatrix}^T \tag{3.21}$$

$\boldsymbol{\chi}$ is commonly referred to as the B-format signal vector and eq. (3.20) represents the first order Ambisonic encoding format. The first order case described above can be understood as a simple mapping from spherical to cartesian coordinates. The notation $\boldsymbol{Y_n}$ is a tribute to the notation commonly used the context of the discrete spherical harmonic transform (DSHT). It must be noted that the sensor arrangements of the CMA1 and the SFM are extraordinary in that they are specifically regular. More precisely, these arrays achieve orthogonal sampling [55]. This is a premise for the simple and exact transformation to a spherical harmonic eigenbeams according to eq. (3.18).

Now consider the planar (2D) array CMA1, where $\vartheta_m = 0\ , \forall m$. From eq. (3.18) and (3.20) follows

$$\begin{bmatrix} \chi_w \\ \chi_x \\ \chi_y \\ \chi_z \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos(\varphi_1) & \cos(\varphi_2) & \cos(\varphi_3) \\ \sin(\varphi_1) & \sin(\varphi_2) & \sin(\varphi_3) \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3.22}$$

Eq. (3.22) indicates that an omnidirectional virtual microphone χ_w can be obtained by simply summing the indiviual cardioids. Furthermore, it can be seen that the planar arrangement (cf. figure 3.3a) produces a redundant z-channel, $\chi_z = \frac{1}{\sqrt{2}}\chi_w$, which can therefore be omitted. The lack of information in z-direction is the reason why only a desired azimuth φ_s, but no elevation el_s can be specified for steering. Inserting the cardioid look-directions $(0, 2\pi/3, 4\pi/3)$ and defining $a = \sin(2\pi/3) \approx 0.866$ yields a transform matrix

$$\boldsymbol{Y_n} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 1 & -0.5 & -0.5 \\ 0 & a & -a \end{bmatrix} . \tag{3.23}$$

3.3.2 Modal Beamforming

The process of transforming the eigenbeams contained in $\boldsymbol{\chi}$ to a single channel signal x_s is refered to as modal beamforming . Modal beamforming is performed by simply weighting and summing of the eigenbeams [19].

$$x_s = \boldsymbol{\omega}^T \boldsymbol{\chi} \tag{3.24}$$

[5] orthogonal sampling (see page 30) is presumed

As stated in eq. (3.7), a first order microphone is obtained by weighted adding of an omni-directional and a figure-8 microphone. Appropriate weighted adding of χ_x, χ_y and χ_z generates a figure-8 in a desired direction (φ_s, ϑ_s). Hence, an arbitrary first order pattern defined by β can be achieved with x_s by using the weight vector $\boldsymbol{\omega} = \boldsymbol{\omega}_\beta$, where

$$\boldsymbol{\omega}_\beta = \left[\tfrac{1-\beta}{\sqrt{2}}, \quad \beta \cos\varphi_s \cos\vartheta_s, \quad \beta \sin\varphi_s \cos\vartheta_s, \quad \beta \sin\vartheta_s\right]^T \tag{3.25}$$

3.3.3 Practical Considerations

Combination of eigen- and modal beamformer

For practical implementation it might be favourable to have a beamformer equation as anticipated in eq. (3.17), because no intermediate signal vector $\boldsymbol{\chi}$ must be stored. Inserting eq. (3.18) in eq. (3.24) yields

$$x_s = \boldsymbol{\omega}^T \boldsymbol{Y_n} \tilde{\boldsymbol{x}} \ .$$

With the definition

$$\boldsymbol{w} = \boldsymbol{Y_n^T} \boldsymbol{\omega} \ , \tag{3.26}$$

the beamformer output signal x_s can therefore be computed as

$$x_s = \boldsymbol{w}^T \tilde{\boldsymbol{x}} \tag{3.27}$$

Using the Pressure Transducer

Up till now only the cardioid microphones $\tilde{\boldsymbol{x}}$ have been used. In case of the CMA1, which includes a pressure microphone x_0, it is however possible to use x_0 instead of χ_w. For use with the practical prototype, this makes a difference because the pressure transducer has a different frequency response and self-noise than the directive microphones. By defining a 4-element weight vector

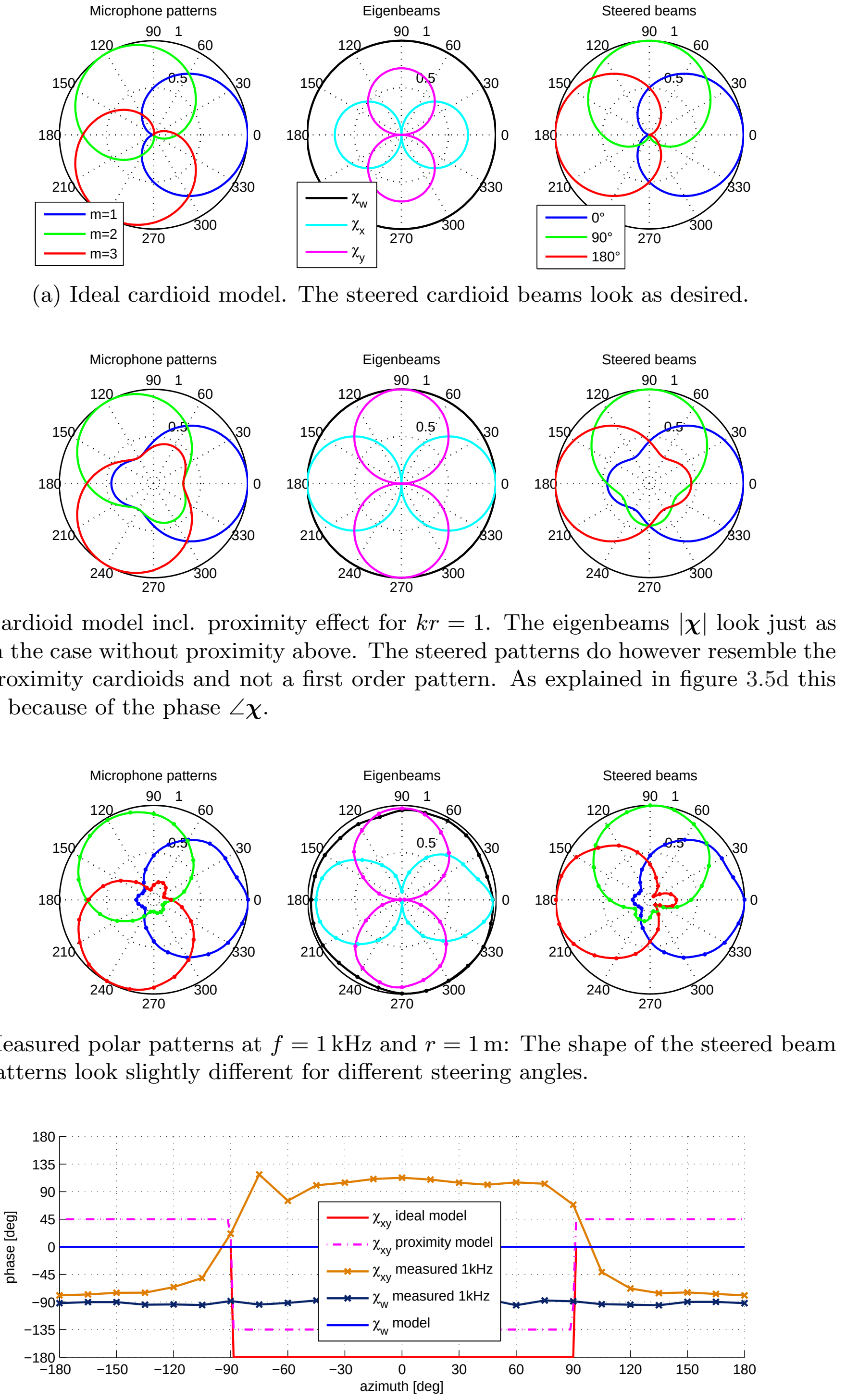
$$\boldsymbol{w} = \left[1-\beta, \quad \boldsymbol{w}_1^T\right]^T \ , \tag{3.28}$$

where $\boldsymbol{w}_1 = \boldsymbol{Y_n^T} \boldsymbol{\omega_1}$ is a 3-element figure-8 steering vector, beamforming with the CMA1 including the pressure channel can be desribed by

$$x_s = \boldsymbol{w}^T \boldsymbol{x} \tag{3.29}$$

Requirements of a Self-steered Array to Localization Accuracy

If the result from ASL is used solely as a steering angle for beamforming with same array, the results obtained above allow conclusions on the required localization accuracy.



(a) Ideal cardioid model. The steered cardioid beams look as desired.

(b) Cardioid model incl. proximity effect for $kr = 1$. The eigenbeams $|\boldsymbol{\chi}|$ look just as in the case without proximity above. The steered patterns do however resemble the proximity cardioids and not a first order pattern. As explained in figure 3.5d this is because of the phase $\angle\boldsymbol{\chi}$.

(c) Measured polar patterns at $f = 1\,\mathrm{kHz}$ and $r = 1\,\mathrm{m}$: The shape of the steered beam patterns look slightly different for different steering angles.

(d) Phase of the Eigenbeams: For perfect first order steering beams as in fig. 3.5a the on-axis half-plane must provide 0° phase-shift between χ_w and the steered figure 8 pattern χ_{xy}. Off-axis, 180° phase-shift are necessary for full cancellation. Only the ideal model fulfills these conditions perfectly.

Figure 3.5: Steering the CMA1 array: The left picture in (a,b,c) shows the array polar patterns $\tilde{\boldsymbol{x}}$, the center picture depicts the eigenbeams $\boldsymbol{\chi}$ and the right picture depicts the beam pattern for 3 different steering angles and a desired pattern $\beta = 0.5$. Not only the magnitude, also the phase of the eigenbeams is of importance to the resulting steering pattern.

Consider a sound source located at φ_s and 2 different beamformers $x_{s,1}$ and $x_{s,2}$ with steering angles $\varphi_1 = \varphi_s$ and $\varphi_2 = \varphi_s + \tilde{\varphi}$, respectively. If the offset $\Delta = |\tilde{\varphi}|$ is small, the signals $x_{s,1}$ an $x_{s,2}$ are similar, i.e. they almost have the same level. Below a certain bound D of level difference, a listener cannot distinguish $x_{s,1}$ and $x_{s,2}$ anymore. Assuming that differences $D < 1\text{dB}$ are not audible, a localization error of $\Delta = 30^\circ = \pi/6$ would be acceptable for a hypercardioid steering pattern because

$$|20 \log_{10} \left(0.25 + 0.75 \cos \frac{\pi}{6}\right)| \approx 0.9 < 1 \tag{3.30}$$

The result $\Delta = 30^\circ$ is based on a single sound source. In practice however, multiple disturbing sources, reverberation and ambient noise can be present. Then the allowed error Δ for non-audible difference in x_1 and x_2 can be significantly smaller.

Specifying a particular value for Δ, would require a comprehensive listening test which is beyond the scope of this thesis. No such studies are known to the author. Subjective experiments made by author with recordings of speech in a medium size room (see chapter 6) indicate that errors $\Delta \leq 10^\circ$ do not degrade the quality of x_1 compared to x_2 significantly.

Perfect Coincidence

If the array is not perfectly coincident, cancellation effects due to phase differences at high frequencies, i.e. small wavelengths $\lambda = c/f$, must be expected when summing the channels. The maximum phase difference that may[6] occur when $d = \lambda/4$ is $\pi/2$. With $\pi/2$ phase shift, there is an amplitude attenuation of 3 dB compared to summation without phase shift. If 3 dB maximum attenuation is accepted, the frequency f should therefore meet

$$f \leq f_h = \frac{c}{4d} \, . \tag{3.31}$$

The combination $c = 340\,\text{m/s}$ and $d = 2\,\text{cm}$ yields $f_h = 4250\,\text{Hz}$.

To reduce the cancellation effects at high frequencies, Gerzon [23] proposed to introduce appropriate filters that compensate for the microphone spacing. Besides the theoretically motivated filters proposed by Gerzon [23], such filters can also be obtained from measurements. An advantage of measured filters is that they allow for compensation of microphone mismatch [1].

[6] The actual phase difference between the channels is not only dependent on the microphone spacing but also on the source angle.

Chapter 4

Localization Algorithms: Foundations

This chapter explains the basic functionality of two different approaches to ASL using coincident microphone arrays (CMAs). Both were already addressed in the introduction, in section 1.1.2. They are termed

1. Intensity vector approach
2. Similarity approach

These approaches are first investigated for a single frequency bin only. This provides the basis for robust localization of real-world signals such as speech which is tackled in chapter 5.

As a starting point, the definitions and models from chapter 2 are used. In particular, a coincident array with M channels indexed by $m = 0, 1, \dots M-1$ is presumed. The first microphone $m = 0$ is assumed to be omni-directional, the others are cardioid microphones with look-directions (φ_m, ϑ_m). As shown in section 3.2, this is general enough to account for the prototype array as well as for the soundfield microphone. Assuming a single sound source s located at a position $\mathbf{\Theta}_s$, the output signal of the m^{th} microphone is given as[1]

$$x_m[n] = s[n] * h_m[n|\mathbf{\Theta}_s] \quad . \tag{4.1}$$

In frequency domain, this is

$$X_m[f] = S[f] \cdot H_m[f|\mathbf{\Theta}_s] \quad . \tag{4.2}$$

[1] Noise-free, free-field conditions are presumed. The microphone is located in the center of the coordinate system (cf. section 3.2).

Instead of eq. (4.2), the following simplified notation is used throughout this chapter for the sake of brevity and clarity.

$$X_m = S \cdot H_{m|\boldsymbol{\Theta}_s} \tag{4.3}$$

4.1 Similarity Approach

Humans are able to localize sound sources because they know from experience how to relate features[2] deduced from the signals picked up at their eardrums to the position of the sound source. Machine learning or pattern recognition algorithms operate quite similarly:

- A feature vector $\boldsymbol{Y}$ is computed from observations.
- A classifier decides to which class q the feature vector $\boldsymbol{Y}$ belongs to.

In the case of ASL, the observed signals are microphone signals and different classes q correspond to different source positions $\boldsymbol{\Theta}_q$. Classification based on a training set of labeled features (training data) is referred to as *supervised* [42]. In this thesis, a simple *minimum distance classifier* (MDC) is used for supervised classification.

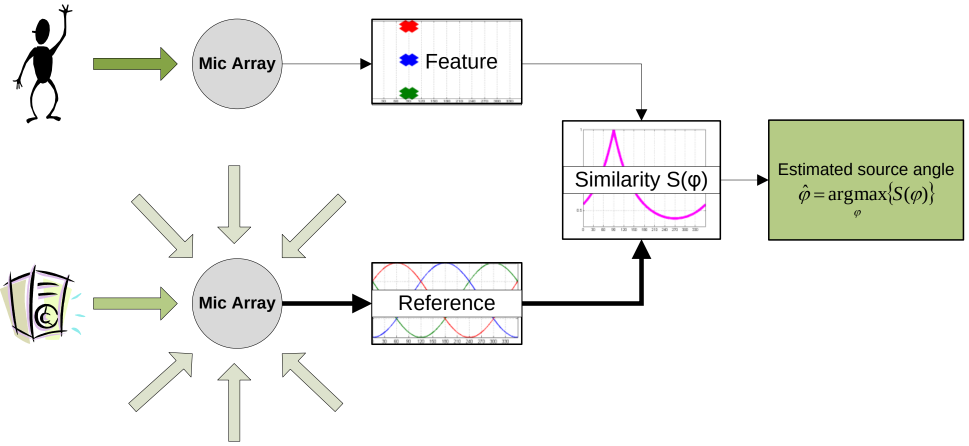


Figure 4.1: Supervised pattern classification principle: A reference database is obtained from measurements with a loudspeaker placed at different positions relative to the microphone array. An actual sound source, e.g. a human speaker, can be localized by computing a feature vector from the microphone signals and comparing it to a reference feature database. The database contains features belonging to different source positions.

[2] ITD, ILD, HRTFs - see section 1

Basically, an observed feature vector $\boldsymbol{Y}$ is compared with a reference database $\underline{\boldsymbol{\mathcal{Y}}}$ consisting of a number Q of feature vectors $\boldsymbol{\mathcal{Y}}(\boldsymbol{\Theta}_q)$ (prototypes, training data). Each vector $\boldsymbol{\mathcal{Y}}(\boldsymbol{\Theta}_q)$ belongs to a different source position $\boldsymbol{\Theta}_q$. The source position is estimated as that position $\boldsymbol{\Theta}_q$ where $\boldsymbol{\mathcal{Y}}(\boldsymbol{\Theta}_q)$ is most similar to $\boldsymbol{Y}$. This principle is illustrated in Figure 4.1.

4.1.1 Features

In the following, appropriate features for ASL with CMAs are derived based on the layout of the prototype array CMA1 presented in section 3.2. Eq. (4.3) states that the incoming signal S is multiplied with different gain factors $H_{m|\boldsymbol{\Theta}_s}$, dependent on the microphone m and the source position $\boldsymbol{\Theta}_s$. The array configuration plot in figure 3.3b shows $H_{m|\boldsymbol{\Theta}_s}$ as a function of the source azimuth angle φ_s for the case of ideal microphones as defined in (3.7). Combining all microphone gains in a vector gives

$$\boldsymbol{H}_{\boldsymbol{\Theta_s}} = \Big[H_{0|\boldsymbol{\Theta}_s}, \quad \ldots, \quad H_{M-1|\boldsymbol{\Theta}_s}\Big] \tag{4.4}$$

If a sound source is placed at a position $\boldsymbol{\Theta}_{s,1} = [0, 0, 1]$, i.e. 0° azimuth, 0° elevation and 1m distance relative to the array, the microphones multiply the incoming signal with

$$\boldsymbol{H}_{\boldsymbol{\Theta}_{s,1}} = \Big[1, \quad 1, \quad 0.25, \quad 0.25\Big] \tag{4.5}$$

Because the first cardioid microphone $m = 1$ looks directly towards the source it picks up sound without attenuation, i.e. $H_{1|\boldsymbol{\Theta}_{s,1}} = 1$. The other two cardioids attenuate the signal s by a factor of 4.

If the source azimuth angle changes however by 180°, i.e. $\boldsymbol{\Theta}_{s,2} = [\pi, 0, 1]$, the vector of gain-factors becomes

$$\boldsymbol{H}_{\boldsymbol{\Theta}_{s,2}} = \Big[1, \quad 0, \quad 0.75, \quad 0.75\Big] \tag{4.6}$$

Obviously, the two sound source positions $\boldsymbol{\Theta}_{s,1}$ and $\boldsymbol{\Theta}_{s,2}$ produce different patterns $\boldsymbol{H}_{\boldsymbol{\Theta}_{s,1}}$ and $\boldsymbol{H}_{\boldsymbol{\Theta}_{s,2}}$. The two positions can therefore be easily distinguished by looking at $\boldsymbol{H}_{\boldsymbol{\Theta}_s}$. In fact[3], every azimuth angle $\varphi_s \in [-\pi, \pi[$ has its distinct pattern $\boldsymbol{H}_{\boldsymbol{\Theta}_s}$ (cf. Figure 3.3b) Consequently, if $\boldsymbol{H}_{\boldsymbol{\Theta}_s}$ is known, φ_s is known as well.

In practice however, the source position $\boldsymbol{\Theta}_s$ - and hence $\boldsymbol{H}_{\boldsymbol{\Theta}_s}$ - is unknown. Only the microphone signals X_m are available. Taking S to the left side of eq. 4.3 yields

$$\frac{X_m}{S} = H_{m|\boldsymbol{\Theta}_s} \tag{4.7}$$

Eq. (4.7) is not directly suited to determine $H_{m|\boldsymbol{\Theta}_s}$ because the source signal S cannot be expected to be given. However, the omni-directional microphone signal X_0 can

[3] Ideal conditions are assumed

be used instead of S because a perfect omni-directional microphone $H_{0|\boldsymbol{\Theta}_s} = 1$ leads to $S = X_0$. This motivates the definition of microphone ratios

$$Y_m = \frac{X_m}{X_0} \tag{4.8}$$

Combining the channels $m = 1, \ldots, M-1$ yields the *feature vector*

$$\boldsymbol{Y} = \Big(Y_1, \ldots, Y_M\Big)^T \tag{4.9}$$

Y_0 is not a part of the feature vector $\boldsymbol{Y}$ because it does not contain any information, i.e. $Y_0 = 1$ regardless of the position $\boldsymbol{\Theta}_s$. Hence, $\boldsymbol{Y}$ has 3 elements in case of the prototype array CMA1. $\boldsymbol{Y}$ is independent of the source signal S, because inserting (4.7) in (4.8) gives

$$Y_m = \frac{S \cdot H_{m|\boldsymbol{\Theta}_s}}{S \cdot H_{0|\boldsymbol{\Theta}_s}} = \frac{H_{m|\boldsymbol{\Theta}_s}}{H_{0|\boldsymbol{\Theta}_s}} \tag{4.10}$$

As shown in section 3.3.1 on eigenbeamforming with the CMA1, the sum of the cardioids delivers an omni-directional signal χ_W. This signal can be used in eq. (4.8) instead of the pressure transducer signal X_0, i.e. the pressure transducer is redundant and not essential to the similarity approach. Hence, the algorithms presented on basis of the CMA1 can be used with a SFM as well.

Reference Features

The minimum distance classifier uses a database $\underline{\boldsymbol{\mathcal{Y}}}$ of reference features $\boldsymbol{\mathcal{Y}}_q$ which are compared with the observed feature vector $\boldsymbol{Y}$. To compare like with like, the reference features are certainly computed just as the observed features. With features as in eq. (4.8) the reference features are given as

$$\mathcal{Y}_{m,q} = \frac{H_{m|\boldsymbol{\Theta}_q}}{H_{0|\boldsymbol{\Theta}_q}} \tag{4.11}$$

The feature vector belonging to the q^{th} position is

$$\boldsymbol{\mathcal{Y}}_q = \Big(\mathcal{Y}_{1,q}, \ldots, \mathcal{Y}_{M-1,q}\Big)^T \tag{4.12}$$

With that, a reference feature matrix $\boldsymbol{\mathcal{Y}}$ can be defined:

$$\underline{\boldsymbol{\mathcal{Y}}} = \Big(\boldsymbol{\mathcal{Y}}_1, \ldots, \boldsymbol{\mathcal{Y}}_Q\Big) \tag{4.13}$$

The set of all reference positions is denoted as

$$\boldsymbol{\mathcal{Q}} = (\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_Q) \tag{4.14}$$

The reference database comprises of $\underline{\boldsymbol{\mathcal{Y}}}$ and the related positions $\boldsymbol{\mathcal{Q}}$.
As an alternative to the feature matrix $\underline{\boldsymbol{\mathcal{Y}}}$, a vector-valued function notation may also be used. The feature vector as a function of the position $\boldsymbol{\Theta}$ is denoted $\boldsymbol{\mathcal{Y}}(\boldsymbol{\Theta})$ and $\boldsymbol{\mathcal{Y}}[\boldsymbol{\Theta}]$ in case of a discrete position grid, respectively.

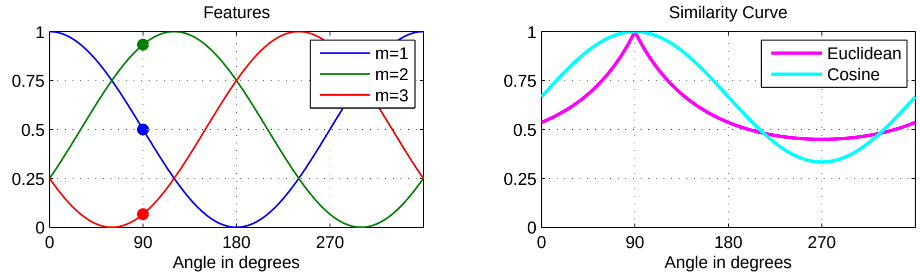


Figure 4.2: Similarity Principle: Assuming an ideal array (CMA1), the 3 dots in the left plot form the 3-element feature vector $\boldsymbol{Y}$ produced by a source located at $\varphi_s = 90°$. The curves represent the reference feature matrix $\boldsymbol{\mathcal{Y}}(\varphi_q)$, i.e. a collection of feature vectors for different angles φ_q.
The right plot shows the similarity curve $S(\varphi_q)$, i.e. the similarity between Y and $\boldsymbol{\mathcal{Y}}(\varphi_q)$, for two different similarity measures: Euclidean and cosine similarity. Both reach a maximum value of $S_{max}(\varphi_q) = 1$ at $\varphi = 90°$, because no noise in the features was assumed and thus $\boldsymbol{Y}(\varphi_s) = \boldsymbol{\mathcal{Y}}(\varphi_q = \varphi_s)$. Consequently, the estimated angle $\hat{\varphi}_s = \operatorname{argmax}\{S(\varphi_q)\}$ is identical to the true source angle φ_s, i.e. the azimuth estimation works perfectly.

4.1.2 Similarity Measures

The similarity of two vectors can be quantified by means of an appropriate similarity measure. Two common similarity measures are presented in the following.

p-Norm

The *p-Norm* of two vectors $\boldsymbol{Y} = (Y_1, Y_2 \ldots, Y_M - 1)^T$ and $\boldsymbol{\mathcal{Y}} = (\mathcal{Y}_1, \mathcal{Y}_1, \ldots, \mathcal{Y}_{M-1})^T$ is given as

$$D = ||(\boldsymbol{Y} - \boldsymbol{\mathcal{Y}})||_p \tag{4.15}$$

$$= \left(\sum_{m=1}^{M-1} |Y_m - \mathcal{Y}_m|^p \right)^{1/p} \tag{4.16}$$

Setting $p = 2$ gives the well-known *Euclidean distance*. The norm is a measure of distance D. It can however be easily converted to a similarity measure S according to

$$S = \frac{1}{1 + D} \tag{4.17}$$

Because $0 < D < \infty$, S is bound between 0 (completely dissimilar) and 1 (vectors are the same). In summary, the p-Norm based similarity $\mathrm{Sim_p}$ can be written as

$$\mathrm{Sim_p}\,\{\boldsymbol{Y}, \boldsymbol{\mathcal{Y}}\} = \frac{1}{1 + ||(\boldsymbol{Y} - \boldsymbol{\mathcal{Y}})||_p} \tag{4.18}$$

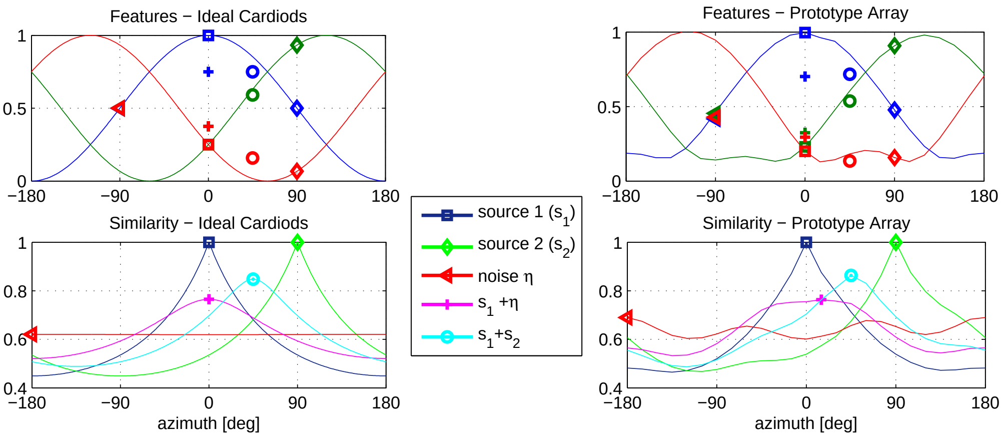


Figure 4.3: Similarity curves for multiple sources and noise. The upper graphs represent the reference features (solid lines) and feature vectors (markers) for 5 different source configurations: Source s_1 at 0° ($\Box$), source s_2 at 90° ($\diamond$), omni-directional noise[4] η ($\triangleleft$) , mixture $s_1 + \eta$ ($+$), mixture $s_1 + s_2$ ($\circ$). The lower plots show the corresponding similarity curves (SC).

Left: Ideal microphone without proximity: Adding omni-noise to a source makes the SC flatter (pink curve, $+$). If there are 2 sources with the same level, the SC peaks in the middle of them ($\circ$) and gets flatter as in the case of a single source. The maximum value depends on how different the angles are to each other. In the extreme case of 180° difference the SC resembles the omni-noise case ($\triangleleft$), i.e. the SC is a horizontal line.

Right: Prototype measurement[5]: The basic characteristics are the same as with the ideal microphones. There are however increases in the omni-noise SC at $[-180, -60, 60]^\circ$. The estimation is therefore pulled towards these angles if the SNR is bad.

Cosine Similarity

Another widely-used measure for similarity is the *cosine* similarity, defined as

$$\mathrm{Sim}_{\cos}\{\boldsymbol{Y}, \boldsymbol{\mathcal{Y}}\} = \frac{\boldsymbol{Y}^T \cdot \boldsymbol{\mathcal{Y}}}{||(\boldsymbol{Y}||_2 \cdot ||\boldsymbol{\mathcal{Y}})||_2} \tag{4.19}$$

4.1.3 Similarity Curve

The similarity between $\boldsymbol{Y}$ and $\boldsymbol{\mathcal{Y}}(\boldsymbol{\Theta})$ is given as

$$S(\boldsymbol{\Theta}_q) = \mathrm{Sim}\left\{\boldsymbol{Y}, \boldsymbol{\mathcal{Y}}(\boldsymbol{\Theta}_q)\right\} \tag{4.20}$$

[4] The term 'omni-directional noise' relates to sound coming from all directions at equal levels.

[5] AKG1-measurement: $f = 1\,\mathrm{kHz}$, $r = 1\,\mathrm{m}$, $\vartheta = 0$. The omni-noise features are generated by adding up contributions from all available directions (24 angles, $[-180, -165, \ldots, 165]^\circ$).

It is a function of the reference position $\mathbf{\Theta}_q$ and named *similarity curve* (SC) in the following. Because minimum distance (MD) equals maximum similarity, the MD-classifier estimate for the source position $\mathbf{\Theta}_s$ is given as

$$\hat{\mathbf{\Theta}}_s = \underset{\mathbf{\Theta}_q}{\operatorname{argmax}} \{S(\mathbf{\Theta}_q)\} \tag{4.21}$$

The principle of maximum similarity and the difference between Euclidean and cosine similarity is illustrated in Figure 4.2. The Euclidean similarity produces a more pronounced maximum in the SC. Only because the maximum is more obvious to the human eye does however not necessarily mean that the Euclidean distance works better. Practical tests revealed that the performance is quite similar with slight favor to the Euclidean distance.

In Figure 4.2, the SC for the case of multiple sources is shown. The term *omnidirectional noise* is used to describe sound coming from all directions at equal levels. In case of multiple sources, the SC gets flatter as in the single source case, i.e. the difference between maximum and minimum of the SC decreases.

If the individual cardioid microphones are well matched, all information is captured within $120°$. The other two thirds of the circle may be constructed by interchanging the cardioid microphone channels. This means that the time needed for measurement of the database can be reduced by a factor of three.

4.1.4 Position Interpolation

The position estimate in (4.21) is tied to the reference position grid, i.e. $\hat{\mathbf{\Theta}}_s \in \mathcal{Q}$. In practice, this means that only the finite number of positions where the microphone responses were actually measured can be detected. If, for instance, the reference position azimuth is given in $15°$ steps, the estimated azimuth can only change in $15°$-steps as well. Now a method is presented that allows estimates $\hat{\mathbf{\Theta}}_s$ between the grid.

The key idea is to interpolate between the maximum of the SC and its neighbors. A simple interpolation method that can be applied to this problem is parabolic interpolation. This method is widely used for interpolation in a DFT-spectrum for finding exact peak frequencies [22], [46][6].

Consider an ordered, discrete grid of Q different angles

$$\mathcal{Q} = [\varphi_0, \ldots, \varphi_{Q-1}] \tag{4.22}$$

In case of an azimuth database sampled with $15°$, we could have for instance $\mathcal{Q} = [-180, -165, \ldots, 165]°$. The reference angles are indexed by the integer

[6] https://ccrma.stanford.edu/~jos/parshl/Peak_Detection_Steps_3.html

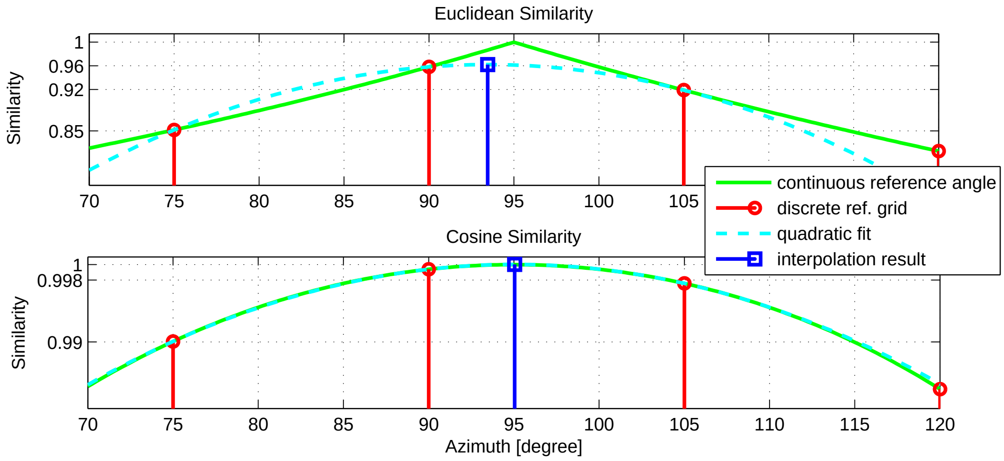


Figure 4.4: Parabolic Interpolation (PI): The true source angle is $\varphi_s = 95°$. This angle is not contained in the discrete position grid $\underline{\varphi_q} = [75, 90, 105, \ldots]°$. Without interpolation the estimated direction is hence $\hat{\varphi}_s = \operatorname{argmax}\{S(\varphi_q)\} = 90°$. With PI however, an estimate in between $90°$ and $105°$ can be computed.
The ideal Euclidean similarity curve does not resemble a parabola. Hence PI does not deliver a very accurate result. The ideal cosine similarity matches very well with a quadratic function. Therefore, PI yields a very good estimate $\hat{\varphi}_s \approx 95°$.

$q = 0, 1, \ldots, Q-1$. Let p denote the index where the similarity curve $S[q]$ reaches its maximum, i.e.

$$p = \operatorname{argmax}\{S[q]\} \tag{4.23}$$

$$S_p = \max\{S[q]\} = S[p] \tag{4.24}$$

The neighbors of p are

$$p_{m1} = \begin{cases} Q-1 & \text{if} \quad p = 0 \\ p-1 & \text{otherwise} \end{cases} \tag{4.25}$$

$$p_{p1} = \begin{cases} 0 & \text{if} \quad p = Q-1 \\ p+1 & \text{otherwise} \end{cases} \tag{4.26}$$

The case differentiation is necessary to stay within the range $[0,\ Q-1]$. Referring to the example grid mentioned above, this allows to interpolate between $-180°$ and $165°$, because the linear index is effectively wrapped to a circle.
According to [22], the offset to the peak-index p in the parabolic fit of the three values $[S_{p_{m1}}, S_p, S_{p_{p1}}]$ is given as

$$\Delta_p = \frac{S_{p_{m1}} - S_{p_{p1}}}{2\left(S_{p_{m1}} - 2S_p + S_{p_{p1}}\right)} \tag{4.27}$$

The spacing $\varphi_{spacing}$ between the two best azimuth candidates ($15°$ for our example) can be defined as

$$\varphi_{spacing} = \begin{cases} \mathrm{princarg}(\varphi_p - \varphi_{p-1}) & \text{if} \quad \Delta_p < 0 \\ \mathrm{princarg}(\varphi_{p+1} - \varphi_p) & \text{otherwise} \end{cases} \tag{4.28}$$

This definition allows for using a non-uniformly sampled azimuth grid. With that, the estimated position using parabolic interpolation is given as

$$\hat{\varphi}_s = \varphi_p + \Delta_p \cdot \varphi_{spacing} \tag{4.29}$$

Figure 4.4 shows the effect of interpolation for the case of an Euclidean and a cosine SC, respectively.

The ideal Euclidean SC does not resemble a quadratic function and consequently, the parabolic interpolation does not work well. In the practice (observed features contain noise) the Euclidean SC is however not as sharp as in the ideal case without noise (cf. Fig. 4.3). Because of the flatter and rounder shape, the parabolic interpolation of an Euclidean SC can be expected to work better in practice as in the ideal, noise-free case.

4.2 Intensity Vector Approach

The basic idea of this approach is to compute the intensity of sound pickup in direction of the Cartesian coordinate axes, i.e. determine the *intensity vector*

$$\boldsymbol{I} = [I_x, I_y, I_z]^T \ . \tag{4.30}$$

The DOA of an incident sound wave can be estimated as the direction of the vector $\boldsymbol{I}$, i.e. once $\boldsymbol{I}$ is known, all that is left to do is simple conversion of Cartesian to spherical coordinates according to eq. (2.4). Consequently, the azimuth angle φ can be estimated by

$$\hat{\varphi}_s = \mathrm{atan2}\left(I_y, I_x\right) \ . \tag{4.31}$$

The estimate for the elevation angle ϑ is given as

$$\hat{\vartheta}_s = \mathrm{atan2}\left(I_z, \sqrt{I_x^2 + I_y^2}\right) \ . \tag{4.32}$$

An illustration of the intensity vector localization principle is shown in Figure 4.5. In [37], Merimaa and Pulkki derive how to obtain $\boldsymbol{I}$ from the B-format signals of a soundfield microphone. As outlined in section 3.3, the B-format $\boldsymbol{\mathcal{X}}$ comprises of an omni-directional component $\mathcal{X}_w$ and gradient (figure eight) components $\mathcal{X}_x$, $\mathcal{X}_y$ and $\mathcal{X}_z$ looking in x-, y- and z-direction, respectively. With this, the intensity vector component I_α can be written as

$$I_\alpha = \Re\{\mathcal{X}_w^* \mathcal{X}_\alpha\} \tag{4.33}$$

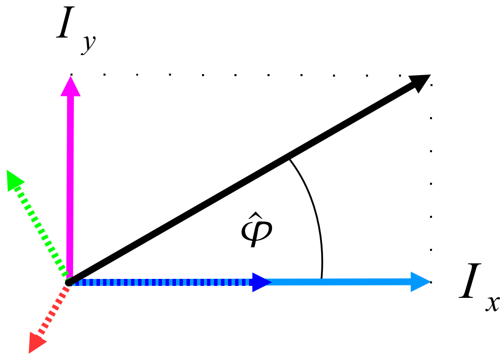


Figure 4.5: Intensity vector principle: The blue, red and green arrow indicate the look directions of the microphones. The length of these arrows represents the strength of the corresponding channel, the magnitude of an FFT-bin. Projection onto Cartesian coordinates leads to I_x and I_y. Given these intensity components in x- and y-direction respectively the azimuth angle φ can be estimated as $\hat{\varphi} = \tan^{-1}(I_y/I_x)$.

where $\mathcal{X}_w^*$ is the complex[7] conjugate of $\mathcal{X}_w$ and the variable α can be either x, y or z.

Now we know how to estimate the direction from the intensity vector $\boldsymbol{I}$ and how to compute $\boldsymbol{I}$ from the B-format signals $\boldsymbol{\mathcal{X}}$. What is left, is to relate $\boldsymbol{\mathcal{X}}$ to the actual microphone signals $\boldsymbol{X}$. This task has already been examined in the context of eigenbeamforming in section 3.3: The B-format signal vector $\boldsymbol{\mathcal{X}} = [\mathcal{X}_w, \mathcal{X}_x, \mathcal{X}_y, \mathcal{X}_z]^T$ can be obtained from the original microphone signals $\boldsymbol{X}$ by

$$\boldsymbol{\mathcal{X}} = \boldsymbol{Y_n X} \ . \tag{4.34}$$

The transform matrix $\boldsymbol{Y_n}$ is defined in eq. (3.19). It is a 3x3 (CMA1) or 4x4 (SFM) matrix defined by the microphone look directions. The signal vector $\boldsymbol{X}$ only contains the cardioid signals. As an option, the omni-directional microphone of the CMA1 can be used instead of the sum of the cardioids, i.e. $\mathcal{X}_w = X_0$.
In case of the CMA1, a figure-eight component $\mathcal{X}_z$ in z-direction can not be obtained (cf. eq. 3.22). Consequently, intensity-vector based estimation of the elevation angle according to eq. (4.32) can not be performed with the CMA1.

[7] As mentioned in the introduction to this chapter, we deal with frequency domain components. These are complex by nature.

Chapter 5

Practical Algorithms

Tracking the position of sound sources demands that a new localization result is available within a certain time interval after the previous result. Thus, the practical algorithms presented in this chapter, work on a frame-level basis, i.e. the signal is segmented into blocks (frames) and a separate result is calculated for each block.

The algorithm should be able to follow rapid changes of the source position. At the same time however, the position estimate should not be easily distracted if for instance a speaking pause occurs. Classic filtering[1] of the position estimate output sequence always results in a tradeoff between speed/smoothness.

An alternative or additional option is to somehow discern between 'good' and 'bad' frames. Many real-world signals such as speech are temporally discontinuous: some frames (e.g. a loud speech vowel) may exhibit good SNR whereas others contain virtually no signal but only background noise (e.g. a short speaking pause between to words). The position estimate computed from a noise-frame cannot be assumed to coincide with the actual sound source position (SSP). If the influence of such 'bad' frames is not suppressed, the position estimate is hence likely to jump wildly instead of staying focussed at the SSP.

As outlined in section 2.3, reflections and ambient noise constitute additional sources from positions different to the SSP. This means that even the best frames are 'noisy'. The algorithm must be able to cope with such situations. In summary, the main challenges for designing a functional localization algorithm are

- *Bad SNR*: The algorithm should perform satisfactorily even when the SNR is bad (constant over time, e.g. fan-noise, heavy reflections).
- *Frame reliability*: The algorithm should be able to suppress the influence of unreliable frames (e.g. speaking pauses).

[1] averaging - e.g. 1-pole low pass, moving average, median - with a constant speed factor

- *All-purpose*: The algorithm should not be dependent on predetermined assumptions on signal (e.g. speech only), noise, microphone position, etc.

In the remainder of this chapter, concepts to tackle the problems described above are presented.

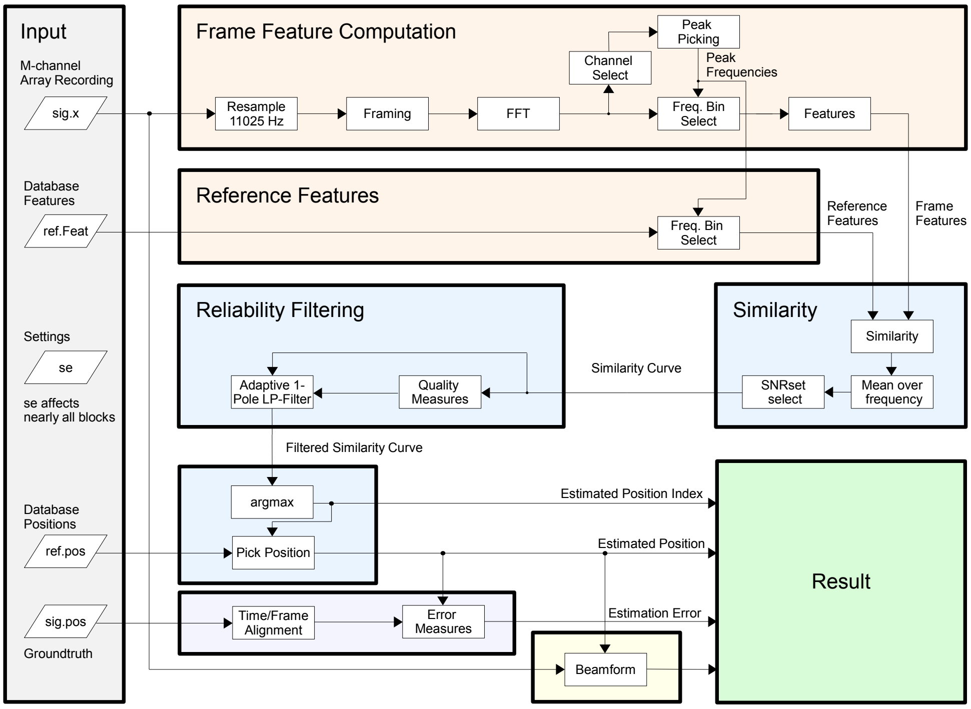


Figure 5.1: Basic scheme of the Matlab localization function. On the input-side, 3 structures may be provided: one for audio signal, reference database and settings respectively. The settings affect nearly all blocks of the algorithm. Signal and reference-features are compared at each peak frequency, which gives a new set of similarity curves for every frame. To suppress the influence of unreliable signal frames, reliability filtering is performed. If a ground-truth file is provided, localization error measures are calculated. The array can be steered towards the detected direction and generate a mono audio beamformer output.

5.1 Similarity Approach

The similarity approach has already been outlined in section 4.1. The main ideas and contributions presented in this section are the following:

- The algorithm works in frequency domain, i.e. the first step is computation of the short time fast Fourier transform (STFFT) of the array signals. Only strong frequency components are used for ASL.
- The performance in adverse environments can be effectively enhanced by using a larger database, where the influence of noise is explicitly modeled.
- As a measure for frame quality, the shape of the similarity curve is rated. Instead of processing the position estimate output sequence, the SC is filtered appropriately.

Figure 5.1 shows an overview of the practical algorithm. In the following, the algorithmic stages are explained one by one.

5.1.1 Short-Time Spectrum

The STFFT consists of two operations: i) framing (windowing) of the time signal and ii) consecutive FFT of the resulting block. A well known property of all time-frequency transforms is the time/frequency resolution trade-off, i.e. the uncertainty principle. The longer the time window (coarse time resolution), the finer the frequency resolution and vice versa.

A simple way to analyze low frequencies with high accuracy and high frequencies with good time-resolution is to use different window-lengths, i.e. perform two or more separate transforms. Since the number of samples of the low-frequency analysis window is high, the corresponding FFT has high computational cost. To reduce complexity, the low-frequency frame (or channel) can be downsampled (cf. Sampling, below). Basically, this multi-resolution approach resembles the principle of the discrete wavelet transform.

Sampling

The first step involved in digital signal processing of the array signals is analog to digital conversion. As outlined in section 2.2, the maximum frequency that can be investigated by means of FFT is dependent on the samplerate f_s.

As can be seen in Figure 3.1b, the directivity of the cardioid microphones of the prototype array is not well pronounced at high frequencies. In practice, frequencies up to 5 kHz proved to be relevant for ASL. Hence, a samplerate $f_s = 11025\,\mathrm{Hz}$ seems appropriate. Because the recordings were made with $f_s = 44.1\,\mathrm{kHz}$, resampling to the lower rate (anti-aliasing low pass-filtering followed by downsampling) is performed. The advantage of lowering the samplerate is a reduction of computational complexity.

If the spectrum is only of interest within a certain frequency band complex modulation and resampling can be performed. This is known as the zoom-FFT [28].

Framing

Each time domain array channel[2] $x[n]$ is divided into frames (blocks) of length N_l samples. The l^{th} frame is given as

$$x_l[n] = x[l\ N_{hop} + n], \quad O \leq n \leq N_l - 1\ , \tag{5.1}$$

where the hop size N_{hop} specifies the overlap of consecutive frames in samples; i.e. if $N_{hop} = N_l$ the blocks are contiguous whereas if $N_{hop} < N_l$ the blocks overlap. N_{hop} can be defined using an overlap factor $Olap$.

$$N_{hop} = \mathrm{round}\left(N_l \cdot Olap\right) \tag{5.2}$$

$Olap = 0.25$ results in 25 % overlap. The round-function is used to assure that N_{hop} is an integer. Since, in the following, the radix-2 FFT algorithm is applied to $x_l[n]$, N_l should be a power of 2.

Fourier Transform

The windowed signal frame $x_l[n]$ is transformed to frequency domain by means of the FFT (cf. section 2.2.2).

$$X_l[f] = \mathrm{FFT_N}\{x_l[n]\} \tag{5.3}$$

With a zero padding factor $Zpad$, the amount of zero padding relative to frame-length N_L can be adjusted

$$N = N_l \cdot Zpad \tag{5.4}$$

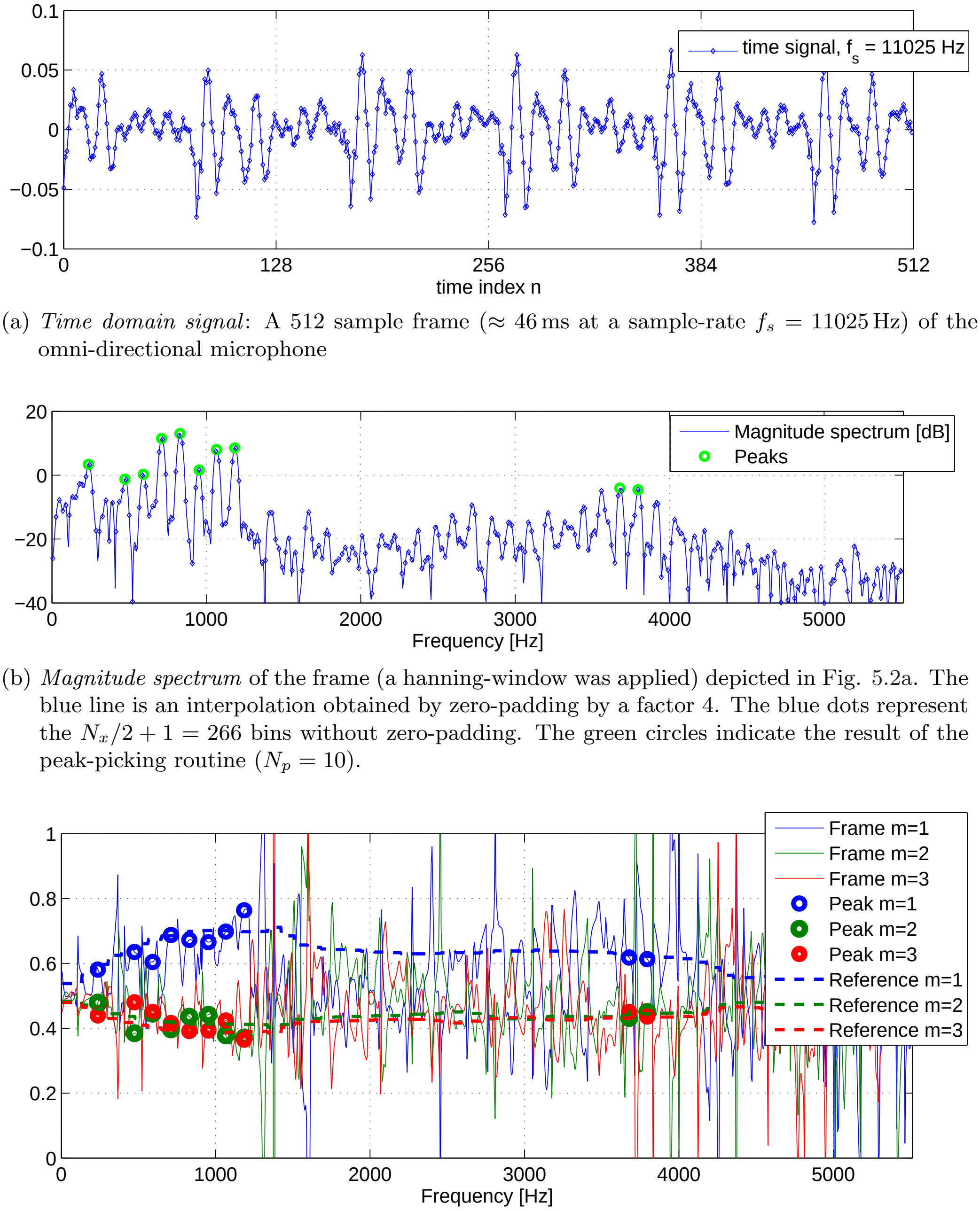
This can be used to increase the number of frequency bins from N_l to N. Basically, zero-padding of the time-domain frame results in a sinc-interpolation of the spectrum [40].

In the following, only the magnitude spectrum is investigated. Since the array signals are real valued, this yields a relevant frequency index range $0 < k < N/2$ (cf. sec. 2.2). Because all the following steps are made on a frame-level basis, the frame index l will be omitted in the following, i.e. $X[f]$ is written instead of $X_l[f]$.

5.1.2 Peak Picking

For each of the $N_f = N/2 + 1$ frequency bins $X[f]$, a basic ASL-routine as described in section 4 can be computed separately. If all spectral bins are taken into account,

[2] the microphone index m is omitted if not relevant, i.e. if the same processing is applied to all channels.



(a) *Time domain signal*: A 512 sample frame ($\approx 46\,\text{ms}$ at a sample-rate $f_s = 11025\,\text{Hz}$) of the omni-directional microphone

(b) *Magnitude spectrum* of the frame (a hanning-window was applied) depicted in Fig. 5.2a. The blue line is an interpolation obtained by zero-padding by a factor 4. The blue dots represent the $N_x/2 + 1 = 266$ bins without zero-padding. The green circles indicate the result of the peak-picking routine ($N_p = 10$).

(c) *Feature curves*: At $\varphi = 0°$, the first (blue) cardioid is stronger than the red and green one, which have the same level. This is reflected in the reference feature curves. The frame-features do only resemble the $\varphi = 0°$-reference features at those frequencies where the source provides sufficient energy. Elsewhere, noise prevails and the $\varphi = 0°$ characteristic is not visible.

Figure 5.2: Framing, peak-picking and features of a male speech vowel [a:] positioned at $\varphi_s = 0°$ with added noise (omni-directional, 12dB SNR). At the peak-frequencies, the frame-features are very similar to the reference features from $\varphi = 0°$. At frequencies with low energy, however, the frame-features are more ore less random and do not provide information on the source-position. For good performance in noisy-conditions it is therefore expedient to only consider strong frequency components.

the computational load would however be very high. There is however yet another point why it is disadvantageous to use all available frequency bins: If the source does not provide energy at a certain frequency, the corresponding ASL-result is likely to be wrong, because the influence of noise dominates at these bins. Using only a certain number $N_p < N_f$ of strong frequency components is hence likely to improve the ASL-performance. The basic step proposed for selecting the N_p bins is peak-picking, i.e. the search for local maxima in the magnitude spectrum. It works as follows:

First, the frame spectrum of one of the array channels is selected. A reasonable choice is the omni-directional microphone. Another possibility is to always pick that microphone, that is believed to be on-axis, i.e. looking towards the sound source. The respective channel can be selected according to the localization result of the previous frame.
The magnitude spectrum of the selected channel is then searched for peaks. The original N_f frequency bins $\{f\}$, are reduced to $N_{\tilde{p}}$ peak frequencies $\{f_{\tilde{p}}\}$.

$$X[f_{\tilde{p}}] = \mathrm{pkpick}\left\{|X[f]|\right\} \tag{5.5}$$

The peak-picking function $\mathrm{pkpick}\{|X[f]|\}$ computes the first difference of $|X[f]|$ and takes the sign of it. The resulting sequence $X_{sd}[f]$ indicates the slope of $|X[f]|$, i.e. $X_{sd}[f]$ is 1 for positive and -1 for negative slope. Samples f where the slope changes from positive to negative are local maxima. Hence, the peak locations are those indices, where the first difference of $X_{sd}[f]$ is negative.
Several additional conditions can be put on the peak candidates (cf. [22]). For instance, only peaks higher than an absolute or relative threshold can be considered. To limit the candidate frequency band to a band $\{f_b\}$, the magnitude of frequencies outside that band is set to zero.

$$\tilde{X}[f_{\tilde{p}}] = \begin{cases} X[f_{\tilde{p}}] & f_{\tilde{p}} \in \{f_b\} \\ 0 & f_{\tilde{p}} \notin \{f_b\} \end{cases} \tag{5.6}$$

Sorting $\tilde{X}_l[f_{\tilde{p}}]$ by descending magnitude moves strong peak frequencies to the left side and weak peak frequencies - including those set to zero in eq. (5.6) - to the right side. Thus, by simply taking the first, i.e. leftmost, N_p bins of the sorted peak spectrum, the N_p strongest frequency components can be selected.
With that, the frame spectrum $\boldsymbol{X}[f]$ of all M channels can be reduced to peak frequency components $\boldsymbol{X}[f_p]$ only.
Basically, the above is how the peak picking stage works. There is however an exception: The number $N_{\tilde{p}}$ of peaks found by the peak-picking algorithm varies from frame to frame and can possibly even be smaller than the number of peak frequencies that shall be considered; i.e. $N_{\tilde{p}} < N_p$. In that case, the remaining $N_p - N_{\tilde{p}}$ bins are provided in the following way:

- First, frequencies found in the regular way as described above are removed from the full frequency spectrum.
- Then, the frequency band is selected similar to eq. (5.6).
- The resulting list is then sorted by descending magnitude and the first $N_p - N_{\tilde{p}}$ bins are selected.
- Finally, these bins are added to those found by the peak-picker algorithm, which again gives N_p frequency components.

The procedure described above usually boosts areas around the strongest peak frequencies. Basically, this adds redundancy to the information provided by the peak frequency bins and is therefore a welcome effect.

It must be noted that in the current implementation, always N_p components are picked, even if the input signal is e.g. a sinusoidal. The possible influence of weak noise or silence components is however eliminated in the following algorithmic stages (reliability weighting). In the case that the signal is known to be a sinusoid it would however be meaningful to set $N_p = 1$. For practical use with speech and other broadband signals a value $N_p = 10$ seemed to work well with the similarity localizer.

5.1.3 Features

As a feature for pattern matching, the ratio between the magnitude spectrum of the cardioids and the omni-directional microphone is used (cf. eq. 4.8). Only strong peak frequencies $\{f_p\}$ are considered. The frame features are calculated as

$$Y_c[f_p] = \frac{|X_c[f_p]|}{|X_0[f_p]|} \tag{5.7}$$

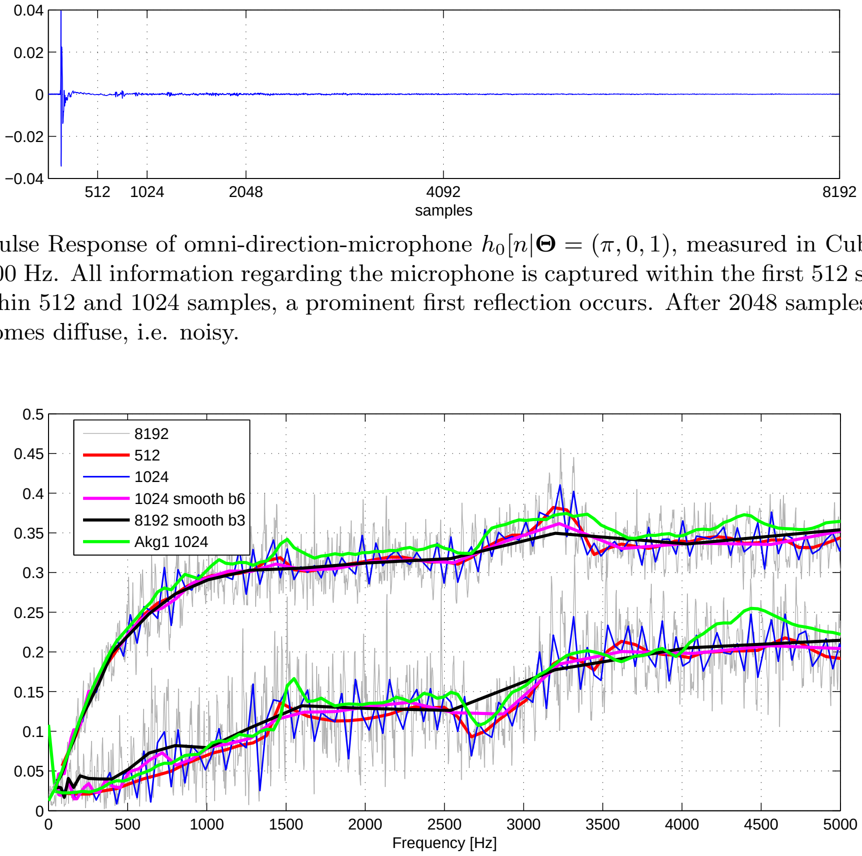
where c is the cardioid microphone index, $1 < c < M - 1$. Consequently, each peak frequency has its $M-1$-component feature vector $\boldsymbol{Y}[f_p] = (Y_1[f_p], \dots, Y_{M-1}[f_p])^T$. The reference frequency responses $\boldsymbol{H}[f_q, \boldsymbol{\Theta}_q]$ are processed in the same way as the frame spectra.

$$\mathcal{Y}_c[f_q, \boldsymbol{\Theta}_q] = \frac{|H_c[f_q, \boldsymbol{\Theta}_q]|}{|H_0[f_q, \boldsymbol{\Theta}_q]|} \tag{5.8}$$

This gives the reference feature matrix $\boldsymbol{\mathcal{Y}}[f_q, \boldsymbol{\Theta}_q]$ All computations concerning the creation of reference feature database are performed only once, outside the algorithm. The actual localization routine only loads the data.

Frequency aligning

The frame features are given at certain frequencies $\{f_p\}$, whereas the reference features are available on a frequency grid $\{f_q\}$. For comparison we need the reference features at the peak frequencies, i.e. $\boldsymbol{\mathcal{Y}}[f_p, \boldsymbol{\Theta}_q]$ instead of $\boldsymbol{\mathcal{Y}}[f_q, \boldsymbol{\Theta}_q]$.

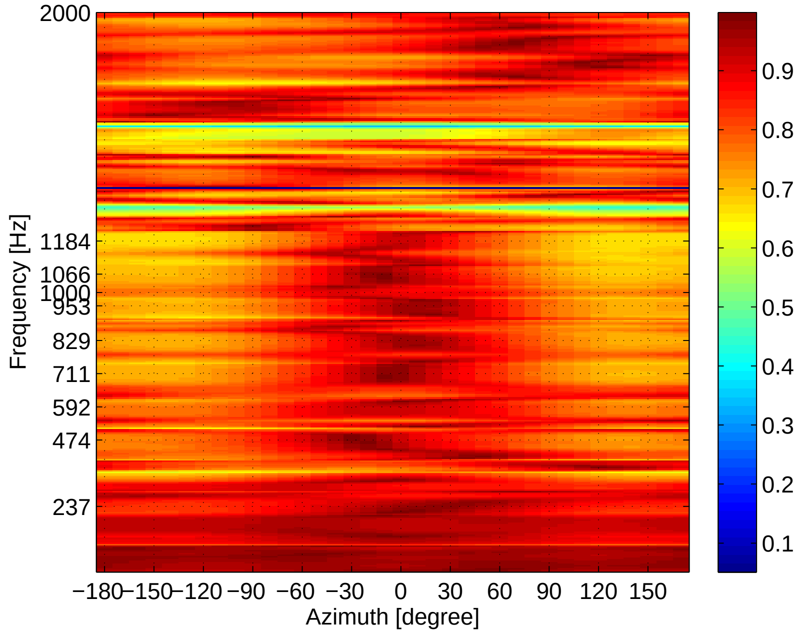


(a) Impulse Response of omni-direction-microphone $h_0[n|\mathbf{\Theta} = (\pi, 0, 1)$, measured in Cube, $f_s =$ 44100 Hz. All information regarding the microphone is captured within the first 512 samples. Within 512 and 1024 samples, a prominent first reflection occurs. After 2048 samples the IR becomes diffuse, i.e. noisy.

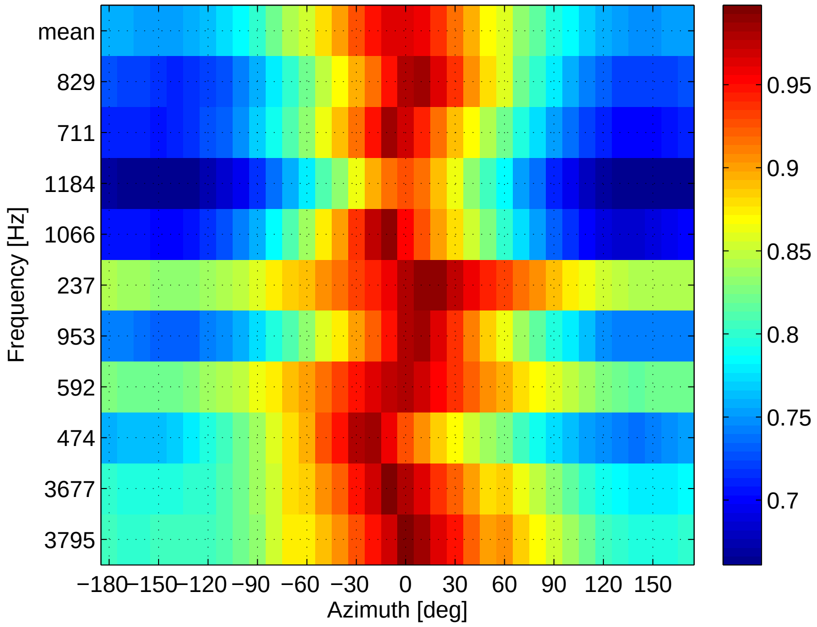
(b) Reference feature curves $Y_m(f|\mathbf{\Theta} = (\pi, 0, 1))$ for different window-lengths and smoothing coefficient b. The lower curves $\mathcal{Y}_1$ belong to the first cardiod microphone $m = 1$ which is off-axis, the upper curves show $\mathcal{Y}_2 = |\frac{H_2}{H_0}|$. A comparable curve measured in another room is shown (AKG1, hall-measurement).

Figure 5.3: Effect of window-length on the reference features: Using a FFT-window of $N_{FFT} = 512$ samples leads to a smooth feature curve. With $N_{FFT} = 1024$, the comb-filter produced by the first room reflection is included (cf. 5.3a). This causes a ripple of approx. 80 Hz. $N_{FFT} = 8192$ yields a very noisy feature curve. Smoothing leads to a very similar result as with 512 samples, because the amount of energy in the IR after 512 samples is quite low.

Since the peak frequencies $\{f_p\}$ need not necessarily be included in $\{f_q\}$, interpolation has to be performed. The simplest and fasted possibility is to take the nearest neighbor, i.e. the frequency f_q closest to f_p. Other possibilities would be linear or spline interpolation. In practice, neighboring reference feature frequency bins do usually not differ very much, which makes nearest neighbor search work fine. Problems can however arise if the reference impulse responses are not truncated appropriately and if no smoothing of the reference features is performed. Then, neighboring frequency bins may differ significantly. This can be seen in Figure 5.3.



(a) Similarity matrix up to 2 kHz: At some frequencies, e.g. between 1200 and 2000 Hz, the SC is flat or peaks at wrong angles. This is due to the fact that the source signal does not provide much energy at these frequencies (cf. Fig. 5.2).



(b) Similarity matrix for peak frequencies only. At the top, the mean similarity curve over these peak frequencies is shown.

Figure 5.4: SC as a function of frequency - similarity matrix. This figure is directly related to Fig. 5.2: The source signal is a speech vowel positioned at $\varphi_s = 0°$. Hence, the SC is supposed to peak at $\varphi_q = 0°$. Because of noise (12 dB SNR) this is however not the case for all frequencies. At the peak frequencies however, the maximum of the SC is distributed quite well around $\varphi_q = 0°$. By taking the mean of the SC over these frequencies, and average SC is obtained. This mean SC is well-shaped and seems to be a good choice for estimation of the azimuth.

5.1.4 Similarity

The similarity of frame and reference feature vector is now evaluated at each frequency bin f_p and for each reference position $\mathbf{\Theta}_q$. This results in a similarity matrix $\tilde{S}[f_p, \mathbf{\Theta}_q]$

$$S[f_p, \mathbf{\Theta}_q] = \mathrm{Sim}\{\boldsymbol{Y}[f_p], \boldsymbol{\mathcal{Y}}[f_p, \mathbf{\Theta}_q]\} \tag{5.9}$$

The position $\mathbf{\Theta}_q$ that achieves the highest similarity can be used as an estimate $\hat{\mathbf{\Theta}}_l[f_p]$ of the position of the sound source that produced $\boldsymbol{Y}_l[f_p]$.

$$\hat{\mathbf{\Theta}}_s[f_p] = \underset{\mathbf{\Theta}_q}{\mathrm{argmax}}\{S[f_p, \mathbf{\Theta}_q]\} \tag{5.10}$$

The position estimate $\hat{\mathbf{\Theta}}_l[f_p]$ is however a function of frequency f_p. Ideally, all frequency components produce the same position estimate $\hat{\mathbf{\Theta}}_s$. In practice however, the position estimate varies with frequency due to noise in the spectral components, i.e. there are N_p potentially different results. In Figure 5.4, the similarity matrix is depicted. It is clearly visible, that for the peak-frequencies, the similarity peaks approximately at the right position, whereas for other frequencies the SC may be flat or peak at a completely wrong angle.

Though the frequency dependence is the key to localization of multiple sources, in case of a single source the frequency dependence is unwanted, i.e. a single source position estimate is desired. The simplest way to obtain a a single result $\hat{\mathbf{\Theta}}_s$ is certainly to consider a single frequency only, i.e. set $N_p = 1$. The redundancy introduced by using multiple peak frequencies is however very important for improving the robustness of the algorithm.

Therefore, two different methods for achieving a single position estimate out of frequency dependent frame features are proposed. They basically differ in the point when to get rid of frequency-dependence:

1. Before taking the argmax, the frequency-dimension of the similarity matrix gets eliminated by taking the mean over frequencies. The position estimate is then basically the argmax of the resulting mean SC.

$$S[\mathbf{\Theta}_q] = \underset{f_p}{\mathrm{mean}}\{S[f_p, \mathbf{\Theta}_q]\} \tag{5.11}$$

$$\hat{\mathbf{\Theta}}_s = \underset{\mathbf{\Theta}_q}{\mathrm{argmax}}\{S[\mathbf{\Theta}_q]\} \tag{5.12}$$

2. An estimation result is calculated separately for all peak frequencies. The position estimate $\hat{\mathbf{\Theta}}_s$ is the mean or median of the frequency-dependent estimate.

$$\hat{\mathbf{\Theta}} = \underset{f_p}{\mathrm{mean}}\{\hat{\mathbf{\Theta}}_s[f_p]\} \tag{5.13}$$

The mean of the position vector $\hat{\mathbf{\Theta}}_s[f_p]$ in (5.13) is performed separately for the components φ,ϑ,r. For the angular components φ and ϑ, the circular mean as defined in 2.33 must be used.

In practice, the first method seemed to produce better results. This seems evident: 'Bad' frames do usually have a rather flat SCs. Such SCs do only make the mean SC flatter but do not change the basic shape, i.e. the location of the maximum.

The second method takes the argmax of each of the SCs and trusts them equally. A bad, i.e. very flat, SC may peak at a completely wrong position and hence have a significant, bad influence on the overall result. The advantage of the second method is however that it is not only applicable to the similarity approach but also to the intensity vector approach. Method 2) is further discussed in the section 5.2.2.

5.1.5 Search-space and Noisy Features

Up till now, a general reference position grid $\mathbf{\Theta}_q$ was used. This notation allows for an arbitrary set of reference positions $\mathfrak{Q}$. The database could for instance consist of only 3 positions, e.g. $[0°, 25°, 1\,\mathrm{m}]$, $[30°, 50°, 0.5\,\mathrm{m}]$, $[190°, 10°, 4\,\mathrm{m}]$. To ensure equal localization-performance for arbitrary source directions, it seems however more convenient to use a more uniformly sampled position grid and a finer resolution, i.e. larger number of positions.

For practical use with the CMA1-Prototype, a database[3] of $Q = 216 = 24{\cdot}3{\cdot}3$ different positions was created. The azimuth was measured in $15°$-steps ($-180°$,$-165°$,..., $165°$) for 3 different elevations ($0°$,$30°$,$60°$) and 3 different distances ($0.5\,\mathrm{m}$,$1\,\mathrm{m}$,$2\,\mathrm{m}$).

In the simplest case, the frame-feature vector is compared with all the available reference positions $\mathbf{\Theta}_q = (\varphi_q, \vartheta_q, r_q)$. A search over so many positions is however quite costly, because the similarity has to computed for each position. The source distance does only have a significant influence on the features when the frequency is low ($kr < 1$, cf. the proximity effect in sec. 3.1.6). This means that feature databases that contain positions that differ only in their distance are highly redundant at high frequencies ($kr > 2$). For estimation of the direction, the search-space can therefore be reduced to a single distance. In case of the above described database, only that part where $r = r_{max} = 2\,\mathrm{m}$ may be used. The direction estimate is then given as

$$\hat{\boldsymbol{\theta}}_s = \underset{\boldsymbol{\theta}_q}{\operatorname{argmax}} \left\{ S(\boldsymbol{\theta}_q, r_{max}) \right\} \ . \tag{5.14}$$

The distance estimator can use the result of the direction-estimator as a given constant. This means that in case of our example database, only three comparisons have to be made.

$$\hat{r}_s = \underset{r_q}{\operatorname{argmax}} \left\{ S(\hat{\boldsymbol{\theta}}_s, r_q) \right\} \tag{5.15}$$

[3] This database is termed 'AKG1'. The measurements were performed in large hall at AKG Acoustics, Vienna.

The direction $\boldsymbol{\theta}$ comprises of azimuth φ and elevation ϑ. In case of an ideal CMA1, the influence of the elevation can be summarized as follows: The directivity of the cardioids decreases[4] with increase of $|\vartheta|$. In the extreme case of a source located at $\vartheta = 90°$ all cardiod microphones have the same output signal. The corresponding feature vector is $\boldsymbol{Y}_\eta = [0.5, 0.5, 0.5]^T$.
As can be seen in Fig. 4.3, this is the same feature that is produced by an omni-directional noise-source. Actually, an elevated sound source behaves just as a sound-source mixed with omni-noise, where $|\vartheta|$ controls the SNR. In practical situations, where we have an unknown amount of noise, the elevation can therefore not be estimated with the CMA1.
The performance of the azimuth estimator can however be improved if there are several feature vector prototypes for each reference azimuth direction. The reference features could for instance be measured for a range of different elevations and SNRs. Since elevation and omni-directional noise have in theory the exact same effect, known effect on the reference features such measurements are however not essential. Instead, the *noisy reference features* can be computed as follows: As shown in section 6.1.2, different SNR-conditions can be simulated by mixing a clean signal with a noisy one in different ratios. The response[5] of the m^{th} microphone due to a mix i of a sound source positioned at φ_q and sound coming uniformly from all directions (omni-noise) is given as

$$H_m(\varphi_q, i) = H_m(\varphi_q) + G_{SNR,i}\, H_{m,\eta}\,. \tag{5.16}$$

where the factor $G_{SNR,i}$ controls the amount of noise and $i = 0, \ldots, I-1$ indexes different SNRs. $G_{SNR,i}$ is related to a certain SNR-value in dB by eq. (6.7). $H_{m,\eta}$ is the microphone response due to omni-noise and may be estimated by summing equal contributions of all available directions

$$H_{m,\eta} = \sum_{q=0}^{Q-1} H_m(\varphi_q) \tag{5.17}$$

In case of ideal microphones we have $\boldsymbol{H}_\eta = K_Q \cdot [1, 0.5, 0.5, 0.5]^T$. With the 'noisy responses' in (5.16), the 'noisy reference features' can be computed just as in eq. (5.8)

$$\mathcal{Y}_c(\varphi_q, i) = \frac{|H_c(\varphi_q, i)|}{|H_0(\varphi_q, i)|} \tag{5.18}$$

If ideal microphones are considered the corresponding reference feature vectors may be written as

$$\boldsymbol{\mathcal{Y}}(\varphi_q, i) = \frac{\boldsymbol{\mathcal{Y}}(\varphi_q) + G_{SNR,i}\, \boldsymbol{\mathcal{Y}}_\eta}{1 + G_{SNR,i}}\,. \tag{5.19}$$

[4] cf. the ideal microphone equation in 3.7

[5] the frequency dependence is omitted for now

The similarity $S(\varphi_q, i)$ is computed for every all φ_q and i and the global maximum is searched, i.e. the azimuth angle estimate is given as

$$\hat{\varphi} = \underset{\varphi_q}{\operatorname{argmax}} \left\{ \max_i \{S(\varphi_q, i)\} \right\} \tag{5.20}$$

Because the noisy features effectively model elevated sources, the azimuth localization performance for elevated sources is increased as well. The computational effort in the similarity stage increases however by a factor I. Practical evaluation[6] revealed that the azimuth estimation performance in adverse SNR-conditions is however improved significantly.

5.1.6 Reliability Weighting

In the beginning of this chapter the problem that some frames produce better position estimates than others was pointed out. 'Good' frames shall be trusted, whereas the influence of 'bad' frames must be suppressed. In this thesis, this task is described by the term *reliability weighting*. First, the crucial task of how to define 'good' and 'bad', i.e. rate the frame quality, is tackled by introducing some frame quality measures.

Frame quality measures

Frame power

An obvious choice for a frame quality measure (QM) is the frame energy. Frames with high energy are assumed to produce a more reliable position estimate than very silent, low energy frames. To be independent from the frame length N_l, the power P should however be used instead of energy. The power of a time-signal frame $x_l[n]$ is given as

$$P = \frac{1}{N_l} \sum_{n=0}^{N_l-1} x_l[n]^2 \tag{5.21}$$

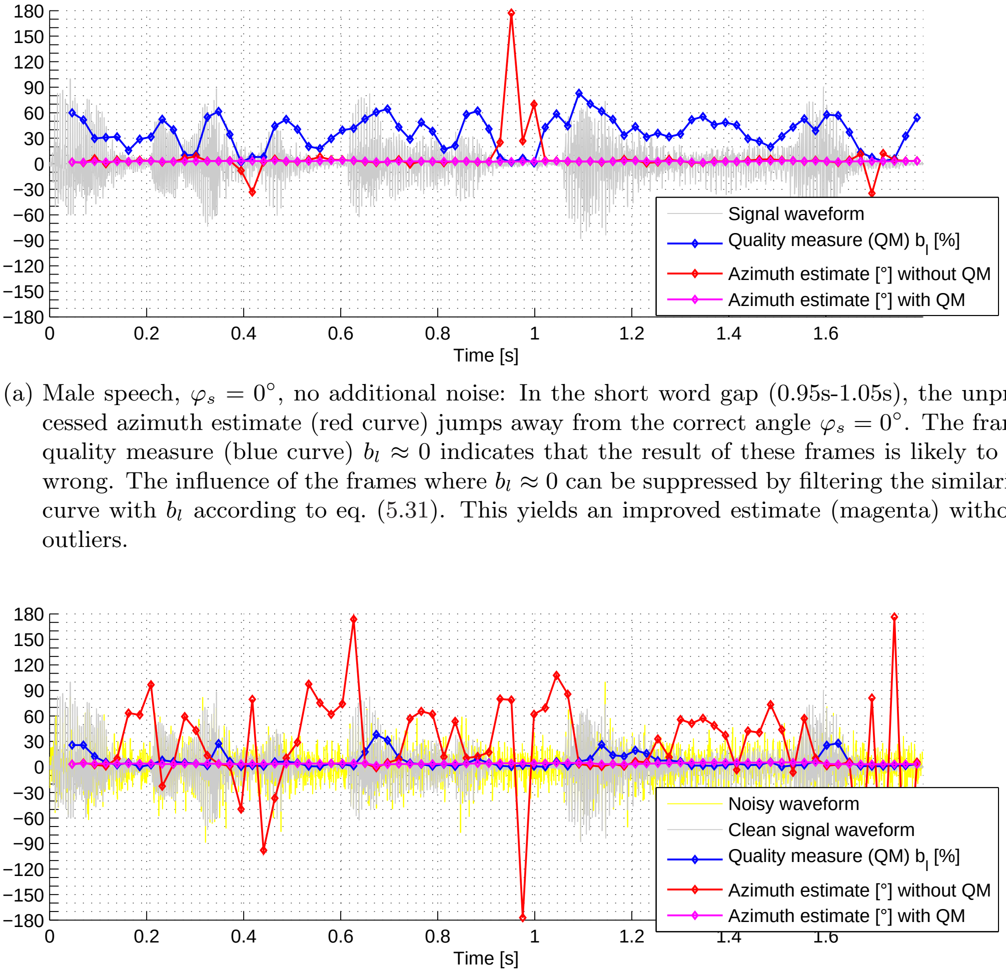
Parseval's relation states that the energy can also be calculated in frequency domain [40], i.e. the frame power P can be written as

$$P = \frac{2}{N \cdot N_l} \sum_{n=0}^{N/2} |X_l[k]|^2 \tag{5.22}$$

Since only peak frequencies f_p are used for the position estimation task, it seems natural to take only their power into account. The *peak frequency power* is given as

$$P_{f_p} = \frac{1}{N_p^2} \sum_{n=0}^{N_p-1} |X_l[f_p]|^2 \tag{5.23}$$

[6] cf. chapter 6



(a) Male speech, $\varphi_s = 0°$, no additional noise: In the short word gap (0.95s-1.05s), the unprocessed azimuth estimate (red curve) jumps away from the correct angle $\varphi_s = 0°$. The frame quality measure (blue curve) $b_l \approx 0$ indicates that the result of these frames is likely to be wrong. The influence of the frames where $b_l \approx 0$ can be suppressed by filtering the similarity curve with b_l according to eq. (5.31). This yields an improved estimate (magenta) without outliers.

Noisy waveform
Clean signal waveform
Quality measure (QM) b_l [%]
Azimuth estimate [°] without QM
Azimuth estimate [°] with QM

(b) Added omni-directional pink noise (SNR=0dB) makes the unprocessed estimate (red) very erratic. The similarity variance quality measure (QM) is however still able to suppress all the 'bad' frames. The resulting magenta curve is again correct without outliers.

Figure 5.5: Reliability filtering: The similarity variance quality measure b_l rates the reliability of the frame l with a value between 0 (unreliable) and 1 (reliable). With that, the influence of unreliable frames can be successfully suppressed.

Voice activity

A typical approach to the problem of discerning reliable and unreliable frames is to employ a separate voice activity detection (VAD) algorithm [32]. Numerous algorithms have been investigated for VAD. Most of them use spectral differences between speech and noise [43],[33]. In the simplest case, a fixed threshold is applied to a feature quantity FQ. A binary measure VA is set to 1 if voice and 0 if no voice is detected.

$$VA = \begin{cases} 1 & \text{if} \quad FQ \leq TH \\ 0 & \text{otherwise} \end{cases} \tag{5.24}$$

A simple feature for discerning voiced and unvoiced frames is the zero-crossing rate (ZCR) of the time-signal. The ZCR of a signal is the number of zero crossings per second. A zero crossing is a point in time, where the sign of the signal changes. The zero crossing rate of a frame $x_l[n]$ can therefore be computed by

$$ZCR = \frac{f_s}{2N_l} \sum_{n=1}^{N_l-1} \left|\operatorname{sgn}\{x_l[n]\} - \operatorname{sgn}\{x_l[n-1]\}\right| \tag{5.25}$$

For using ZCR as a feature quantity FQ in eq. (5.24), a threshold value $TH = 1500$ seemed to work quite well for classifying speech vowels.

Similarity Curve Variance

Up till now, the frame quality measures were directly derived from the source signal $x_l[n]$. A different approach is to look at the shape of the SC.

The SC of reliable frames can be expected to have a single[7] peak at the source position. Unreliable frames however, tend to have either a flat or rippled SC. This property can be seen in Figure 4.3.

In practical experiments (cf. Figure 5.5), the sample variance of the similarity curve over its position index proved to be a simple, yet powerful quality measure.

$$V_S = \operatorname{var}\{S[\varphi_q]\} \tag{5.26}$$

High values of V_S indicate reliable frames.

Filtering

Assume that a normalized frame quality measure coefficient b_l , $\quad 0 \leq b_l \leq 1$ is given. The higher the value of b_l, the more reliable is the frame, i.e. if $b_l = 0$ the frame is completely unreliable and if $b_l = 1$ the frame can be fully trusted. The simplest method to use of b_l for suppressing unreliable frames is to accept or reject the current

[7] Similar to the chapter interpolation 4.1.4 an ordered set of azimuth angles is assumed.

frame estimation result, depending on if b_l exceeds or falls below a certain threshold b_{TH}.

$$\tilde{\boldsymbol{\Theta}}_l = \begin{cases} \hat{\boldsymbol{\Theta}}_l & \text{if} \quad b_l \geq b_{TH} \\ \hat{\boldsymbol{\Theta}}_{l-1} & \text{otherwise} \end{cases} \tag{5.27}$$

Eq. (5.27) can also be written as

$$\tilde{\boldsymbol{\Theta}}_l = b_{l,01} \cdot \hat{\boldsymbol{\Theta}}_l + (1 - b_{l,01}) \cdot \tilde{\boldsymbol{\Theta}}_{l-1} \tag{5.28}$$

where $b_{l,01}$ is the binary quality coefficient

$$b_{l,01} = \begin{cases} 0 & \text{if} \quad b_l \geq b_{TH} \\ 1 & \text{otherwise} \end{cases} \tag{5.29}$$

The estimation result $\tilde{\boldsymbol{\Theta}}_l$ only changes if a reliable frame is detected. Otherwise it sticks to the old result. This can lead to discrete jumps in the estimation result sequence $\tilde{\boldsymbol{\Theta}}_l$. A smoother run of the sequence can be achieved if b_l is not thresholded, i.e. if $b_{l,01}$ in eq. (5.28) is replaced by b_l. Because this means that some sort of averaging is performed, the spherical coordinates $\tilde{\boldsymbol{\Theta}}_l$ must be converted to Cartesian coordinates $\tilde{\boldsymbol{P}}_l$ to prevent errors in the angular components (cf. circular mean in eq. 2.33).

$$\tilde{\boldsymbol{P}}_l = b_l \cdot \hat{\boldsymbol{P}}_l + (1 - b_l) \cdot \tilde{\boldsymbol{P}}_{l-1} \tag{5.30}$$

This is the difference equation of a 1-pole low pass filter with a time-varying pole $1 - b_l$ where $\hat{\boldsymbol{P}}_l$ is the input sequence and $\tilde{\boldsymbol{P}}_l$ is the output, i.e. the improved version of the estimation result. The response time and hence the amount of smoothing is dependent on the quality of the frame.

An alternative to processing the result is to perform the filtering in an earlier stage in the algorithm, i.e. filter an intermediate sequence. In our case the SC seems appropriate. The SC produced by the l^{th} frame is denoted[8] S_l. Before taking the argmax, S_l is passed through the adaptive 1 - pole low pass filter, which yields $\tilde{S}_l$. The corresponding difference equation is

$$\tilde{S}_l = b_l \cdot S_l + (1 - b_l) \cdot \tilde{S}_{l-1} \tag{5.31}$$

The enhanced position estimate is then given as

$$\hat{\boldsymbol{\Theta}}_l = \underset{\boldsymbol{\Theta}_r}{\operatorname{argmax}} \{\tilde{S}_l(\boldsymbol{\Theta}_r)\} \tag{5.32}$$

In eq. (5.31), the time-varying coefficient b_l $,0 \leq b_l \leq 1$ determines how much influence the actual frame has compared to the history $\tilde{S}_{l-1}$. It seems fruitful to again discuss the following extremes:

[8] In this section, the dependence of the similarity curve $S[\boldsymbol{\Theta}_q]$ on the reference position $\boldsymbol{\Theta}_q$ is omitted. The frame-index l is however introduced again.

- $b_l = 1$: In this case, equation eq. (5.31) simplifies to $\tilde{S}_l = S_l$ and the estimation is equal to eq. (5.12), where no reliability weighting was employed. In other words, all history is deleted and the estimation is based on the actual similarity curve only; i.e the actual frame is fully trusted.
- $b_l = 0$: Here we have $\tilde{S}_l = \tilde{S}_{l-1}$. The influence of the current frame l is fully suppressed. The outcome $\hat{\mathbf{\Theta}}_l$ of eq. (5.32) is the same as $\hat{\mathbf{\Theta}}_{l-1}$, the result of the previous frame $l - 1$.

Filter coefficient b_l

The following frame quality measures were introduced in eq. (5.23),(5.25) and (5.26), respectively:

- Peak frequency power P_{f_p}
- Zero crossing rate ZCR
- Variance of the similarity curve V_S

These quantities do have different units and scaling. The aim is to have a quality coefficient b_l, $0 < b_l < 1$ that is 0 if the frame is unreliable and 1 if it is reliable, i.e. b_l must use its range between 0 and 1 properly. If b_l is not close to 1 for very good frames the algorithm will be slow, i.e. not able to follow quickly changing source positions. On the other hand, if b_l does not get close to 0 for very bad frames, the position estimate may be erratic and loose track of a source easily.

A simple method to assure that any quality measure Q_l, e.g. V_{S_l}, ZCR or P_{f_p}, can be used to achieve a b_l with $0 \leq b_l \leq 1$ is to introduce a threshold and make a hard decision between good and bad frames, i.e. set b_l to 0 and 1 respectively. This method can be applied to all the above quality measures and gives new, binary quality measures that can be used with eq. (5.36). The threshold must however be chosen carefully and the drawbacks of a binary filter coefficient mentioned above occur.

A better method is to find an appropriate scaling factor Q_{max} This can be done by recording a sequence Q_l over time l from experimental data and retrieve the maximum as a scaling factor.

$$Q_{max} = \max_l \{Q_l\} \tag{5.33}$$

$$\tilde{b}_l = \frac{Q_l}{Q_{max}} \tag{5.34}$$

Because Q_{max} is retrieved by experiment, it could however happen that $\tilde{b}_l$ may still be greater than 1 for conditions that are better as in the experiment. To assure that the filter coefficient b_l is never larger than 1 a clipping function can be introduced.

$$b_l = \begin{cases} 1 & K\tilde{b}_l > 1 \\ \tilde{b}_l & K\tilde{b}_l < 1 \\ 0 & K\tilde{b}_l < 0 \end{cases} \tag{5.35}$$

The factor $K \in \mathbb{R}$, $K \geq 0$ is termed *speed factor*, because large values of K drive b_l towards 1, which makes the algorithm faster (cf. eq. 5.31).
Please note that all the above mentioned quality measures fulfill $Q_l >= 0$ and thus the condition $b_l \geq 0$ is fulfilled in any case.
$\tilde{b}_l$ may also be represented as a product of N_Q quality coefficient $\tilde{b}_{l,q}$ originating from different quality measures

$$\tilde{b}_l = \prod_{q=0}^{N_q-1} \tilde{b}_{l,q} \tag{5.36}$$

If $0 \leq \tilde{b}_{l,q} \leq 1$, $\forall q$, the condition $0 < \tilde{b}_l < 1$ is fulfilled. Multiplication of several quantities $\tilde{b}_{l,q}$, $0 \leq \tilde{b}_{l,q} \leq 1$ can however easily lead to $\tilde{b}_l << 1$. The speed factor K can however be used to compensate for this.

Practical Notes

In the practical implementation the similarity curve filtering method in (5.31) in combination with the similarity curve variance quality measure worked well.
The use of frame power has the following drawbacks:

- If the source distance increases, the energy decreases and the algorithm gets slower.
- If the SNR is very bad, all frames may have similar power.

The drawback of the ZCR is, that it is very specific only for speech vowels. A strong source with a noise-like spectrum cannot be detected. Furthermore it fails in low SNR situations - defining a threshold gets increasingly difficult. The performance of many VAD-algorithms suffers significantly in low-SNR situations [43].
The similarity variance V_S is superior to ZCR and frame power. It works for arbitrary source spectra and can quantify frame quality even in very low SNR-conditions. The only thing that requires specific attention is the scaling of V_S.
As an alternative to filtering the mean (over frequency) similarity curve obtained by eq. (5.12), the whole similarity matrix (all frequency bins) could be filtered. A separate coefficient $b_l(f)$ can be computed for every frequency bin f. However, in a few practical experiments this did not seem to perform better as filtering the mean SC. The computational load is however much higher.

5.2 Intensity Vector Approach

The intensity vector approach has already been introduced in section 4.2. A practical implementation can be made in time domain (TDIV) or frequency domain (FDIV). The TDIV approach is very simple and computationally efficient. The FDIV allows for a simple form of reliability weighting and performs better in bad SNR conditions (cf. sec. 6).

5.2.1 Time Domain

One possibility to implement the IV-approach in time domain is to calculate the RMS-value of each time-frame $x_l[n]$ in (5.1).

$$X_l = \sqrt{\frac{1}{N_l}\sum_{n=0}^{N_l-1}\left\{x_l^2[n]\right\}} \tag{5.37}$$

Then the transformation to Cartesian coordinates is performed.

$$\boldsymbol{\mathcal{X}}_l = \boldsymbol{Y_n}\boldsymbol{X}_l \tag{5.38}$$

Here, $\boldsymbol{X}_l = [X_{l,1}, \ldots, X_{l,M-1}]^T$ contains the RMS-values of all cardioid channels and $\boldsymbol{Y_n}$ is a transform matrix as defined in eq. (3.19). The vector $\boldsymbol{\mathcal{X}}_l$ consists of components in the Cartesian coordinate directions. In case of the planar CMA1, we have

$$\boldsymbol{\mathcal{X}}_l = \begin{bmatrix}\mathcal{X}_{w,l}, & \mathcal{X}_{x,l}, & \mathcal{X}_{y,l}\end{bmatrix}^T \tag{5.39}$$

A simple 1-pole filter may be used to smooth the run of $\boldsymbol{\mathcal{X}}_l$. The smoothed version is denoted by $\boldsymbol{I}_l$.

$$\boldsymbol{I}_l = (1-a)\boldsymbol{\mathcal{X}}_l + a\boldsymbol{I}_{l-1} \tag{5.40}$$

The pole a is a constant value. In practical experiments, a value $a = 0.9$ performed well, if the frame length was approximately $50\,\mathrm{ms}$. The azimuth estimate is given as

$$\hat{\varphi}_l = \mathrm{atan2}\left\{I_{y,l}, I_{x,l}\right\} \tag{5.41}$$

The integration time is defined by the frame length N_l and the additional smoothing filter pole a. In practice, it is however hard to find a good tradeoff between smoothness (no outliers) and speed.

5.2.2 Frequency Domain

As a starting point we take the frame spectrum $X_l[f]$ as defined in (5.3). The number of considered spectral components may be reduced by peak-picking and/or by restricting the frequency band (cf. section 5.1.2). The following steps are completely analog to the TDIV-approach.

$$\boldsymbol{\mathcal{X}}_l[f] = \boldsymbol{Y_n}\boldsymbol{X_l}[\boldsymbol{f}] \tag{5.42}$$

$$\tilde{\boldsymbol{\mathcal{X}}}_l[f] = (1-a)\tilde{\boldsymbol{\mathcal{X}}}_l[f] + a\boldsymbol{\mathcal{X}}_{l-1}[f] \tag{5.43}$$

The intensity vector components are given as (cf. 4.32)

$$I_{x,l}[f] = \Re\left\{\tilde{\boldsymbol{\mathcal{X}}}^*_{w,l}[f] \cdot \tilde{\boldsymbol{\mathcal{X}}}_{x,l}[f]\right\} \tag{5.44}$$

$$I_{y,l}[f] = \Re\left\{\tilde{\boldsymbol{\mathcal{X}}}^*_{w,l}[f] \cdot \tilde{\boldsymbol{\mathcal{X}}}_{y,l}[f]\right\} \tag{5.45}$$

The frequency dependent azimuth estimate is given as

$$\hat{\varphi}_l[f] = \mathrm{atan2}\left\{I_{y,l}[f], I_{x,l}[f]\right\} \tag{5.46}$$

For tracking of a single sound source, a single estimation result is desirable. A simple possibility to achieve this, is to take the circular mean or median over frequency.

$$\hat{\varphi}_l = \underset{f}{\mathrm{cmean}}\left\{\hat{\varphi}_l[f]\right\} \tag{5.47}$$

This method has already been mentioned in section 5.1.4.

Reliability Weighting

As shown in section 5.1.6, the sequence $\hat{\varphi}_l$ can be filtered with an adaptive 1-pole lowpass-filter, where the zero is controlled by a frame quality measure. For the FDIV approach, two such quality measures are presented in the following.

Standard deviation over frequency

The frequency distribution of the estimation result $\hat{\varphi}_l[f]$ can be used to define a frame quality measure. The idea can be illustrated by the following two extremes.

- If all frequency bins produce the same position estimate, the variance of the estimate over frequency is zero. This is the ideal case for a single sound source. It is meaningful to sum up the result $\hat{\varphi}_l[f]$ by a single estimate $\hat{\varphi}_l$.

- If $\hat{\varphi}_l[f]$ strongly varies with frequency f, the variance is high. A single source position cannot be clearly identified. Summarizing $\hat{\varphi}_l[f]$ to a single position estimate $\hat{\varphi}_l$ by taking the mean does therefore not seem very meaningful.

Instead of the variance, the standard deviation $s_{\hat{\varphi}_l}$ may be used because it has the same units as $\hat{\varphi}_l$, which makes interpretation easier. Again, a circular version (see [11]) must be used.
A binary filter coefficient can be generated by introducing a threshold. If the standard deviation exceeds the threshold, e.g. $s_{\hat{\varphi}_{TH}} = 30°$, $\hat{\varphi}_l$ is not updated. It is however hard to define a universal threshold for different SNR-conditions. The second possibility is to use a properly scaled version of $s_{\hat{\varphi}_l}$ directly as filter coefficient.

Test for significance

A different quality measure may be provided by a hypothesis test that examines whether the circular sample mean (or median) in eq. 5.47 equals the true population mean (or median). Corresponding MATLAB functions are provided by CircStat circular statistics toolbox [5].

Again, the result of the hypothesis test can be used to control the adaptive 1-pole lowpass-filter. A few practical experiments, indicated that this method performs better than the standard-deviation approach presented above.

Chapter 6

Experimental Evaluation

.

6.1 Experimental Setup

To evaluate the performance of the proposed localization algorithms, recordings with the prototype array depicted in Figure 1.1 were made. The recordings were carried out at the Institute of Electronic Music and Acoustics in the room 'IEM-CUBE', an approximately 10m x 12m x 4m large multipurpose room. Beside the usage as a lab, this room is mainly used for lectures and electro-acoustic music [54]. The CUBE is equipped with an optical tracking system[1] and a 24-channel hemispherical loudspeaker array [18].

[1] a V624 data station and 15 M2 cameras by Vicon (cf. http://www.vicon.com)

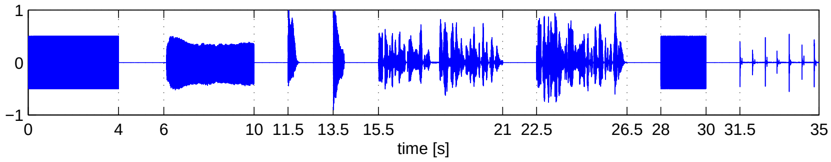
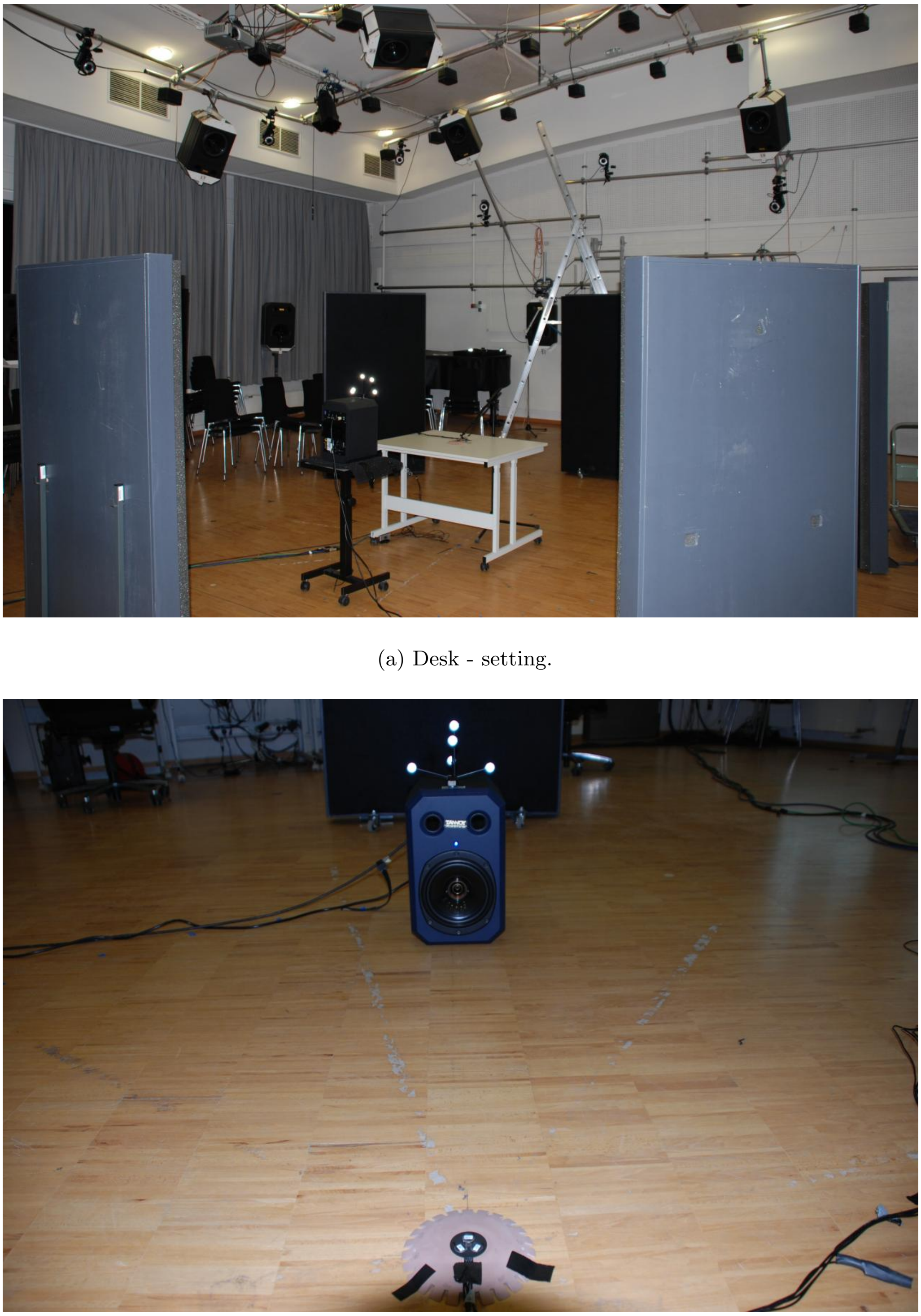


Figure 6.1: A versatile *excitation signal* comprising of different constituent signals separated by silence was used for measurements. The signals are: Exp. sweep, male speech vowel [a:], single word (male speaker), single word (female), male speech, female speech, white noise, footsteps.



(a) Desk - setting.

(b) Floor - setting.

Figure 6.2: Measurements were conducted in the *IEM Cube*, a medium-sized room. Additional absorber elements were used to reduce reverberation. Two measurement scenarios '*desk*' and '*floor*', relating to the location of the microphone array, were considered. Besides measurements with a loudspeaker, a male human speaker acted as a sound-source.

6.1.1 Test Signals

As a sound-source, a loudspeaker and a male human came into operation. Static and moving sound sources, as well as different source signals were considered.

- *Loudspeaker*: Active near-field monitor Tannoy System 600A
  - Static source: At various static positions, the test signal depicted in Figure 6.1 was played back. It consists of an exp. sweep, speech samples, noise and footsteps (transient signal) separated by silence. This allows for
    - *System identification*, i.e. computation of reference features.
    - *Evaluation of ASL-performance* for different static positions. The exactly same speech sample was played back from different locations. With this, the effect of single, heavy reflections can be simulated (cf. eq. (6.5)).
  - *Moving source*: Speech or pink noise was played back while the loudspeaker was moving, i.e. carried around manually.
- *Human*: A male human (the author) was the sound-source.
  - *Speech*: Talking while moving around the array in different walking speeds, sitting at the desk.
  - *'Real-world noises'*: Person falling down on the floor, footsteps, newspaper, plastic coffee cup etc.

Two different locations of the microphone array, were considered. The setting is depicted in Figure 6.2.

- Desk: The array was placed on a small desk.
- Floor: The array was placed on the floor.

To allow for simulation of different SNR-conditions noise was recorded.

- Background noise: The room was quiet (30dBA). Hence, the self noise of the microphones and amplifiers contributes considerably.
- Diffuse (omni-directional) pink noise: Pink noise played back with the same level from all 24 Tannoy System 1200 loudspeakers in the CUBE. This constitutes a diffuse noise field, i.e. the noise is arriving at the array more or less[2] equally from all directions.
- Directive noise: The loudspeaker excitation signal contained a white noise sample. Such a white noise signal can either be used as static source or, when added to another source, as a disturbance signal.

[2] The array was placed in the center of the loudspeaker array. No calibration was performed however.

6.1.2 Simulation of different SNR-Conditions

Various multi-source and SNR scenarios can be set up by simple weighting and adding of recordings of different sources. This means that a noisy signal $x[n]$ can be generated from a clean signal $x_s[n]$ and noise $x_\eta[n]$ as given in eq. (6.5). With this, a considerable reduction of measurement time can be achieved. Certainly, linearity of acoustics, microphones and recording setup[3] must be premised for using eq. (6.5) with the array recordings.

The SNR is defined as the ratio of signal energy and noise energy in Decibel.

$$SNR = 10 \log_{10} \left(\frac{P_s}{P_\eta} \right) , \tag{6.1}$$

where

$$\begin{aligned} P_\eta &= \frac{1}{N} \sum_{n=0}^{N-1} (x_\eta[n])^2 && (6.2) \\ P_s &= \frac{1}{N} \sum_{n=0}^{N-1} (x_s[n])^2 && (6.3) \\ & && (6.4) \end{aligned}$$

The SNR of the recorded speech signals compared to background noise can be estimated by cutting out different parts, i.e time-ranges, of the recording:

- A 'stationary' speech part is used as $x_s[n]$,
- A signal pause is used as $x_\eta[n]$:

Referring to the excitation signal in Figure 6.1 reasonable time-ranges are $t_{range,s} = [15.55, 17.85]$ for the male speech signal and $t_{range,\eta} = [4.5, 5.5]$ for background noise, respectively. With this, the following SNR-values can be reported for the desk setting at 1m distance:

- omni-directional microphone: 43 dB
- on-axis cardioid: 37.5 dB
- off-axis cardioid: 26.5 dB

A noisy array signal $\boldsymbol{x}[n]$ with a desired SNR can be generated from a clean signal $\boldsymbol{x}_s[n]$ and noise $\boldsymbol{x}_\eta[n]$ as follows

$$\boldsymbol{x}[n] = \boldsymbol{x}_s[n] + G_{SNR} \cdot \boldsymbol{x}_\eta[n] , \tag{6.5}$$

where

$$\begin{aligned} G_{SNR} &= \sqrt{\frac{P_d}{P_\eta}} && (6.6) \\ P_d &= P_s \, 10^{(-SNR/10)} && (6.7) \end{aligned}$$

[3] The nonlinearity of the loudspeakers is no problem in this regard. It plays however a role in system identification.

The omni-directional channel was used to determine a factor G_{SNR}, which was then applied to all channels according to eq. 6.5. Because a recorded signal $x_s[n]$ already contains background noise (43 dB SNR) it is not to meaningful to specify very high SNR-values ($>$ 30 dB).

6.1.3 Audio Recording and Optical Tracking Setup

For evaluation of the ASL-performance, the true position of the source must be known. In many papers regarding ASL of moving sound sources, these ground-truth position tags are manually generated from video data. This is neither very accurate nor elegant. By using the optical tracking system installed in the CUBE this obstacle can be overcome. Highly accurate groundtruth-tags can be automatically generated in sync with the audio recordings.
To record optical tracking in sync with audio, a measurement patch in the graphical real-time signal processing environment Pure Data (PD) was created. Basically, it works as follows: PD receives the positional information obtained by optical tracking from the tracking software VICON iQ 2.5 via OSC. When a new recording is started, a filename must be entered. A 4-channel wave-file and a text-file of the same name is generated and the audio recording starts. At the same time, the incoming OSC data stream is sampled in PD with $T = 25\,\mathrm{ms}$ and written to the text-file. The audio signals were played back and recorded with a samplerate $f_s = 44.1\,\mathrm{kHz}$.

6.2 Performance Metrics

A simple and powerful way for a human viewer to examine the ASL-performance is to plot the sequences φ_l and $\hat{\varphi}_l$, i.e. groundtruth and estimation result, respectively, on top of each other. Additionally, the microphone signal waveform can be superimposed. Such a plot delivers detailed insight, because it becomes apparent at what points t_l in time $\hat{\varphi}_l$ is close to φ_l . This is for instance important for checking if the algorithm is fast enough to follow rapid changes. For comparison of the overall performance of different settings and algorithms it may however favorable to get rid of the frame-index l, i.e. consolidate the information. This can be achieved by means of the following performance measures.

The *estimation error* $\tilde{\varphi}_l$ is defined as the difference between true value φ_l and the estimated value $\hat{\varphi}_l$. In case of angular data, the principle argument as defined in (2.31) must be applied, i.e.

$$\tilde{\varphi}_l = \mathrm{princarg}\left\{\hat{\varphi}_l - \varphi_l\right\} \tag{6.8}$$

Basic error metrics are the bias, mean absolute error (MAE), mean square error

(MSE) and root mean square error (RMSE). They are defined as follows:

$$\begin{aligned} Bias &= \text{mean}\{\tilde{\varphi}_l\} && (6.9)\\ MAE &= \text{mean}\{|\tilde{\varphi}_l|\} && (6.10)\\ MSE &= \text{mean}\left\{\tilde{\varphi}_l^2\right\} && (6.11)\\ RMSE &= \sqrt{MSE} && (6.12) \end{aligned}$$

Please notice that even though $\tilde{\varphi}_l$ is an angular value, the circular mean is not appropriate for calculation of the $Bias$[4]. In contrast to the MSE, the RMSE has the same unit as the quantity φ. This is helpful for interpretation. Due to the squaring, the RMSE is very sensitive to large errors, i.e. it increases significantly if a only a few large errors occur.

A performance metric also used in [27] is the percentage of frames where the absolute error $\tilde{\varphi}_l$ is smaller than a defined maximum acceptable error Δ. It is referred to as frame *accuracy* and defined as:

$$\text{ACC}_\Delta = \frac{1}{L}\sum_{l=0}^{L-1} \delta_\Delta(\tilde{\varphi}, \varphi)\ , \qquad (6.13)$$

where

$$\delta_\Delta(\hat{\varphi}, \varphi) = \begin{cases} 1 & |\tilde{\varphi}_l| \leq \Delta \\ 0 & \text{otherwise} \end{cases} \qquad (6.14)$$

$\text{ACC}_{\Delta=5} = 90\%$ means for instance that 90% of all frames achieve an estimation error $|\tilde{\varphi}_l| \leq 5°$. A lot of information at a glance is provided by plotting the accuracy as a function of the true source angle φ and the error bound Δ (cf. Figure 6.4).

6.3 Evaluation Results

6.3.1 Static Sources

The main excitation signal used for evaluation of the performance of static sources was a $1.8\,\text{s}$ long male speech signal[5] played back by a loudspeaker with the array placed on the desk. Figures 6.4, 6.5 and 6.6 depict the azimuth estimation accuracy for different algorithms, different reference databases and different parameter settings, respectively. These accuracy image plots are very detailed, because they show the performance as a function of the source azimuth and the parameter Δ. Figure 6.3 shows a summarized plot, achieved by taking the mean over all source angles and by specifying a specific bound Δ. Besides accuracy, the mean (over all source angles) $RMSE$ and MAE is shown.

The basic settings used with the similarity algorithm were

[4] This can be easily checked with a simple example values for the estimation error, e.g. $[90, -120]°$. The circular mean yields 165° whereas the linear mean delivers −15°

[5] '... course of a December tour in York ...'. Part $15.815\,\text{s} - 17.615\,\text{s}$ of the excitation signal. The signal waveform is depicted in Figure 5.5.

- Framing: $46.44\,\mathrm{ms}$ frame-length, 50% overlap, Hamming window
- Spectral Analysis: $200 - 4000\,\mathrm{Hz}$ band, $N_p = 10$ peaks
- Similarity: Euclidean similarity, noisy features database, similarity variance reliability weighting.

For the IV-approach a smoothing pole $a = 0.8$ was used.
From $46.44\,\mathrm{ms}$ frame-length and 50% overlap follows that every $23.22\,\mathrm{ms}$ a new estimation result is available. The $1.8\,\mathrm{s}$ long signal this gives a total of 77 frames. Hence, the ACC increases by approximately 1.3% for every frame with a wrong estimation result.

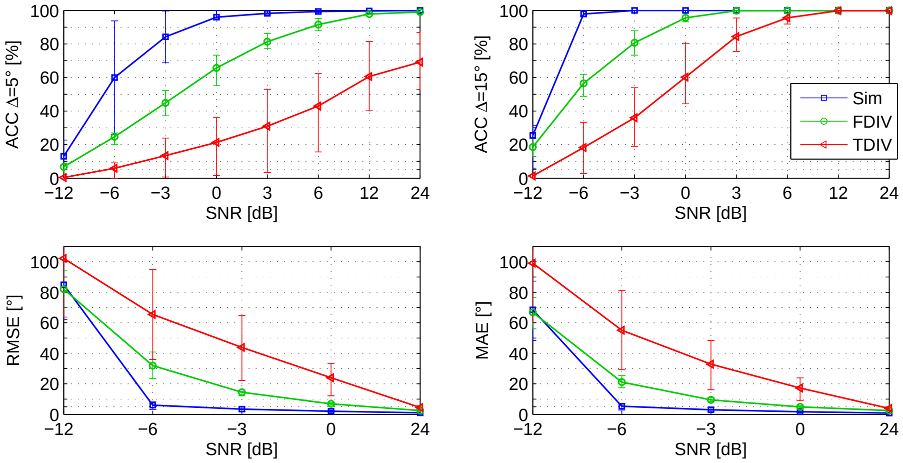
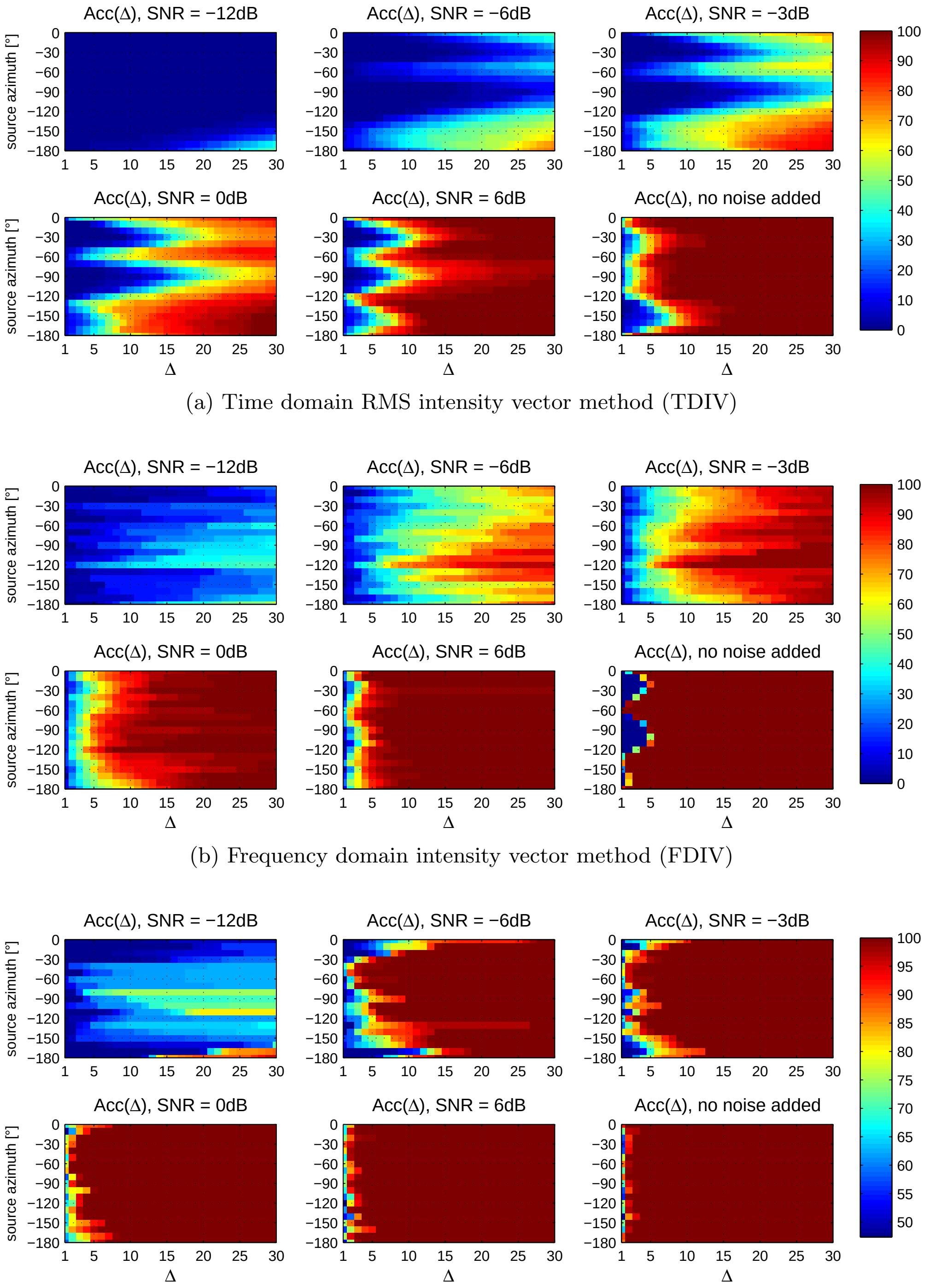


Figure 6.3: Different algorithm approaches: Mean over all source angles $(-180, -170, \ldots, 0)°)$ of $ACC_{\Delta=5}$, $ACC_{\Delta=15}$, $RMSE$ and MAE, respectively. The errorbars indicate the first and the third quartile. This figure shows the same data as Fig. 6.4

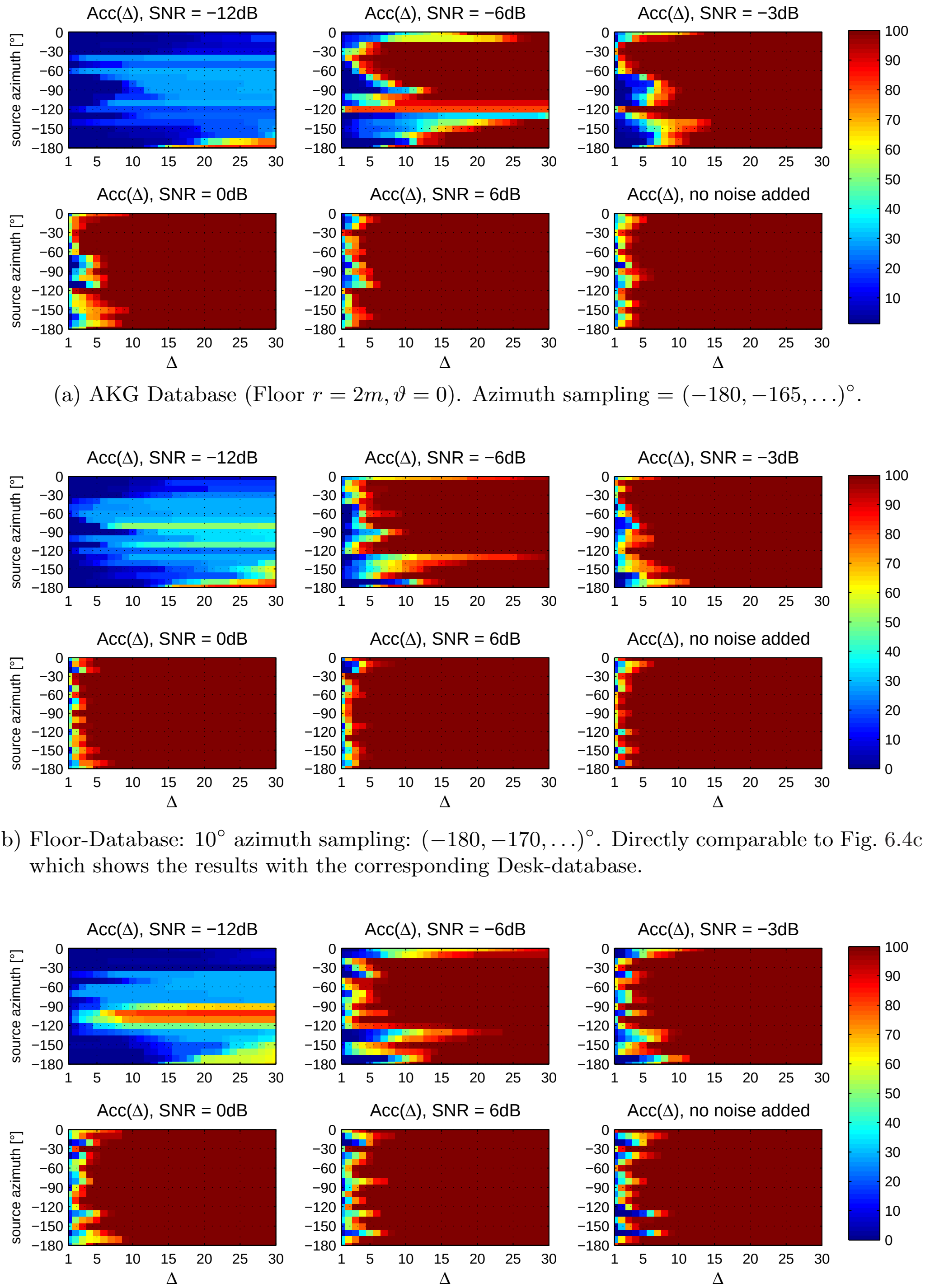


(a) Time domain RMS intensity vector method (TDIV)

(b) Frequency domain intensity vector method (FDIV)

(c) Similarity approach: Desk-database, i.e. 'best case'.

Figure 6.4: Different algorithm approaches: Accuracy as a function of source angle and SNR. Desk-Setting, signal: 1.8s male speech, noise: pink, omni-directional.
The similarity-approach works very well even below 0 dB SNR. The FDIV-approach is a good alternative if the SNR is $\geq$ 6 dB. The simple TDIV-approach is not as accurate as the competitors at good SNR and fails $\leq$ 6 dB SNR.

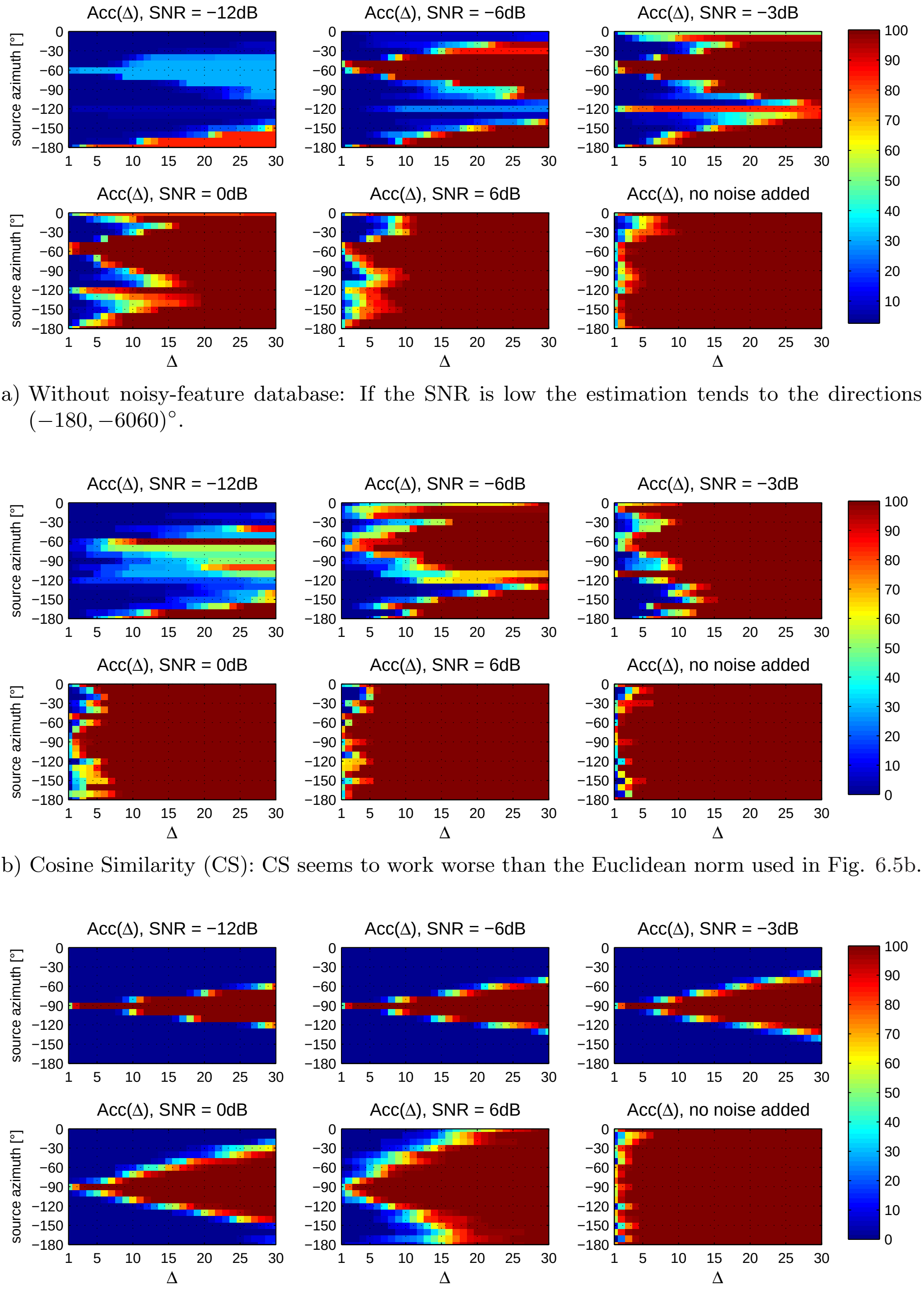


(a) AKG Database (Floor $r = 2m, \vartheta = 0$). Azimuth sampling $= (-180, -165, \ldots)^\circ$.

(b) Floor-Database: 10° azimuth sampling: $(-180, -170, \ldots)^\circ$. Directly comparable to Fig. 6.4c which shows the results with the corresponding Desk-database.

(c) 30° sampled Floor-Database (only contains $(-180, -150, \ldots)^\circ$).

Figure 6.5: Accuracy for different reference databases. Recordings as in Fig. 6.5. Though the microphone is placed on the desk, the floor-databases work fairly well. As could be suspected, a reduced dataset (less reference angles) has a negative effect on the accuracy at the missing angles.



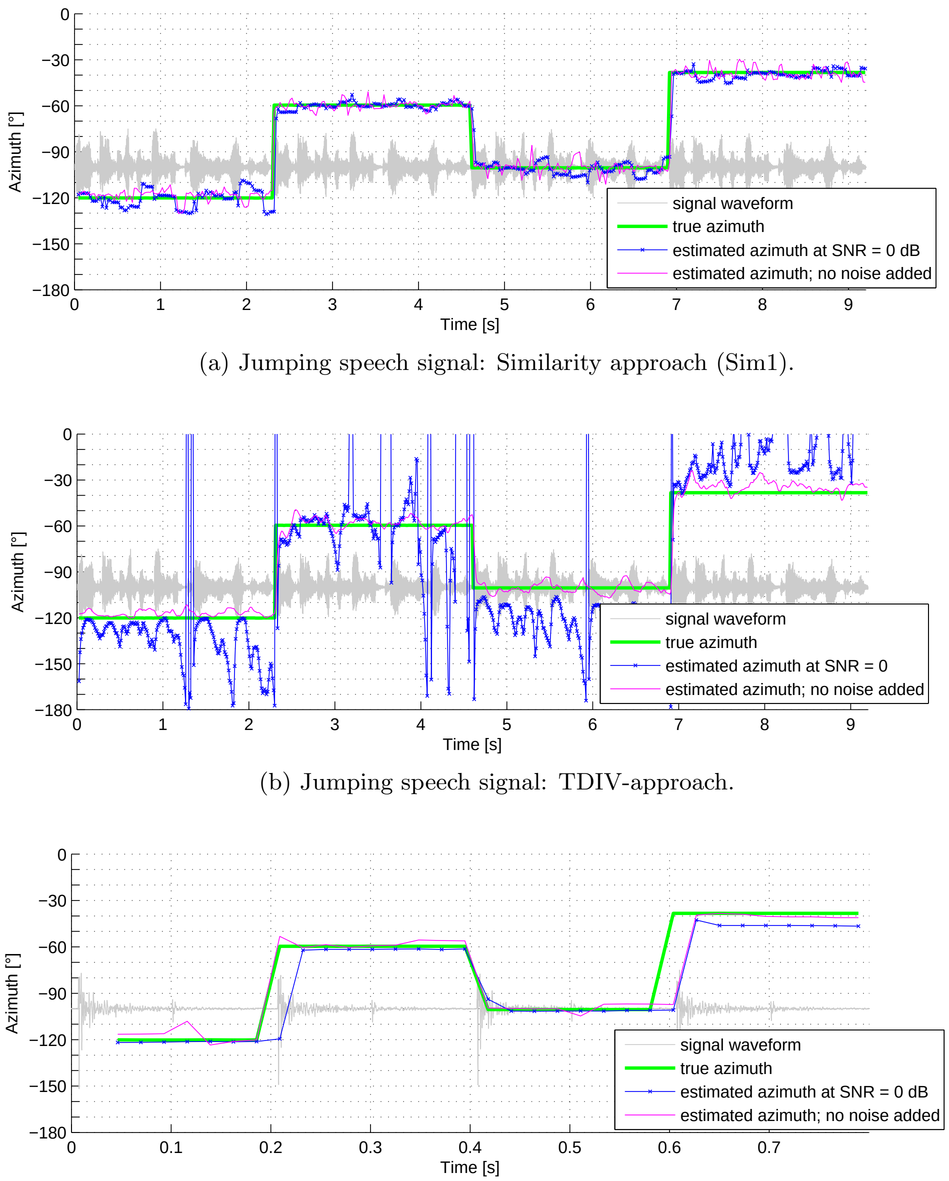
(a) Without noisy-feature database: If the SNR is low the estimation tends to the directions $(-180, -6060)^\circ$.

(b) Cosine Similarity (CS): CS seems to work worse than the Euclidean norm used in Fig. 6.5b.

(c) Single reflection: a 10ms delayed version of the source signal from −90°.

Figure 6.6: Accuracy for different algorithm settings.

6.3.2 Moving Sources

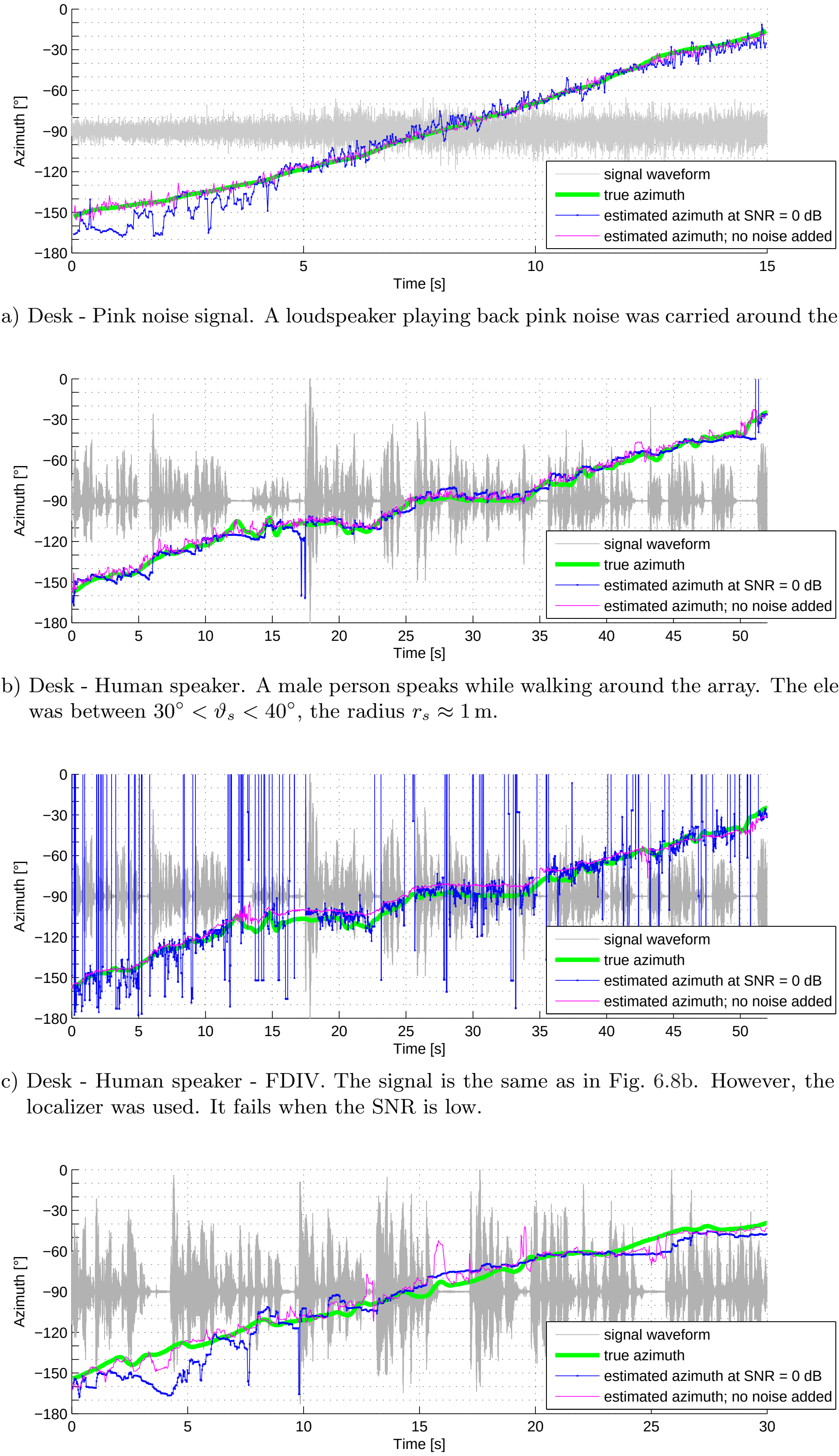


(a) Jumping speech signal: Similarity approach (Sim1).

(b) Jumping speech signal: TDIV-approach.

(c) Jumping footsteps-signal: Sim1.

Figure 6.7: Jumping sources: The localizer is fast enough to follow sudden changes of the source angle. The TDIV-approach does not work reliably at 0 dB.



(a) Desk - Pink noise signal. A loudspeaker playing back pink noise was carried around the array.

(b) Desk - Human speaker. A male person speaks while walking around the array. The elevation was between $30^\circ < \vartheta_s < 40^\circ$, the radius $r_s \approx 1\,\mathrm{m}$.

(c) Desk - Human speaker - FDIV. The signal is the same as in Fig. 6.8b. However, the FDIV localizer was used. It fails when the SNR is low.

(d) Floor - Human speaker: Here the array was placed on the floor. Hence the elevation is higher ($\vartheta_s \approx 60^\circ$). This is probably the reason, why the result is less accurate as in the case of the desk-setting.

Figure 6.8: Moving Sources. The generic floor-database was used.

Chapter 7

Summary, Conclusions and Outlook

This thesis dealt with tracking of a single sound source in a noisy environment using a coincident microphone array (CMA). A basic practical advantage of a CMA compared to a spaced array is its small dimensions.

In chapter 3, a mathematical description of CMAs was presented. Each microphone of a CMA can be described by a set of impulse responses dependent on the source position. As an alternative to measurements, appropriate microphone model equations were reviewed. A discussion of beamforming with coincident arrays revealed the property that the beampattern of a CMA is limited to first order.

In chapter 4 and 5, the principles and practical aspects of two different localization approaches were discussed, respectively. The *similarity approach* uses a minimum distance classifier approach, i.e. it works with a database of reference features that are compared with the observed feature vector. For tracking of real-world sources, the observed feature vector was computed frame-wise and in frequency domain. To suppress the influence of noise, only frequencies where the source spectrum delivers sufficient energy were used for comparison. The result of the comparison was named similarity curve (SC). In case of a single sound source, the SC peaks at the actual source position. In case of sound reaching the array equally from all directions, i.e. omni-directional noise, the similarity curve is completely flat.

A particular focus of the practical algorithm design was that the localizer should not be easily distracted from the actual source position by unreliable frames, e.g. a pause in a speech signal. Usually, a separate voice activity detector is used for that purpose. In chapter 5 of this thesis however, a so called *frame reliability weighting* system was incorporated directly within the actual algorithm. The quality with respect to localization was considered good if a single sound source with high energy (good SNR) is present.

The variance of the SC was found to be a powerful feature for the frame quality, because it represents the flatness of the SC. As mentioned above, the SC turned out to be flat if there is no clear, single source direction. In contrast to speech related features, the concept of rating the shape of the SC is in principle independent of the source spectrum. Furthermore, no threshold must be defined. The SC-variance was used to control an adaptive first order 1-pole filter, that processes the change of the similarity curve over time. Basically, the SC is updated quickly if the frame quality is high, and updated slowly if the frame quality is low. With this, the influence of pauses in the signal can be effectively suppressed.

For additional improvement of robustness in bad SNR-conditions, the database was expanded with noisy feature prototypes. The azimuth search was restricted to a frequency band where the proximity effect is negligible. Due to the proximity effect, the similarity approach is in principle capable of estimating the distance of a sound-source. In contrast to estimation of the azimuth this did however not work very well in practice. Estimation of the elevation is impracticable with the planar prototype array CMA1.

With the *intensity vector* (IV) approach, an established method for coincident source localization was reviewed. Two practical types of implementation were presented: A time domain (TDIV) and a frequency domain (FDIV) intensity vector localizer. The presented FDIV-algorithm is comparable to similarity approach in the way that a single source position estimate was demanded and frame reliability weighting was introduced. As a frame quality measure a hypothesis test for the significance of the circular median of the frequency distribution of the localization result was proposed.

An experimental evaluation of the proposed algorithms with real array recordings was made. Different signals played back by a loudspeaker as well as a human speaker were recorded. To provide ground-truth position tags, the actual source position was tracked optically. An omni-directional noise-field was recorded and used for simulation of different SNR-conditions. The experiments showed that the similarity algorithm outperforms the IV-algorithms. It is fast enough to follow quickly changing source positions and still not sensitive to noise and silence gaps. However, the FDIV-approach proved to be a viable alternative to the similarity approach in case that the SNR is not too bad ($> 6\,\mathrm{dB}$).

7.1 Outlook

Possible future work may include the following tasks:

- Evaluation in different rooms: The evaluation was performed in a single, rather dry room. To explicitly evaluate the influence of reverberation, the periphonic loudspeaker system in the CUBE to generate various reverberation settings.

Recordings in several conference rooms would be a good basis for further evaluation of the practicability of the presented algorithms.

- Multi-source tracking: If the source-spectra do not overlap too much in frequency domain it should be possible to simultaneously track multiple sources. The necessary extensions to the presented algorithm may include clustering of the SC in frequency domain and a more advanced frequency-tracking system/peak-picking system.

- Different microphone configurations: The algorithm was tested only with 2-D prototype array. It should however be a simple task to transfer the algorithm to a 3-D CMA such as the sound-field microphone. Furthermore, the pattern recognition approach may also be applied to a spaced spherical array that incorporates several directive microphones that are distributed on a circle(2D) or sphere(3D). Such arrays have the advantage that beamforming with higher order is possible [19].

- Different pattern recognition approaches: A minimum distance classifier was used in this thesis. However, classifiers such as a Bayes-classifier or a multi-class support vector machine (SVM) might also be tried for classifying the sound source direction.

- Further use and elaboration of the microphone model. Given a certain source distribution of a room, i.e. source signals and their positions, the signal recorded by a microphone may be simulated by use of the position dependent impulse response model proposed in chapter 3 of this thesis. This could for instance be useful for simulating the sonic properties of different microphone setups for surround-recordings.

Bibliography

[1] F. Adriaensen. A tetrahedral microphone processor for ambisonic recording. In *LAC2007 - 5th International Linux Audio Conference*, 2007. [cited at p. 28, 33]

[2] P. Annibale, R. Rabenstein, S. Spors, and P. Steffen. A short review of signals and systems for spatial audio. In *European Signal Processing Conference (EUSIPCO 2009), Glasgow*, Aug. 2009. [cited at p. 13]

[3] D. Arfib, F. Keiler, and U. Zölzer. *Time-Frequency Processing*, chapter 8. John Wiley & Sons, 2002. [cited at p. 12]

[4] J-M. Batke. The B-format microphone revised. In *Proc. 1st Ambisonics Symposium*, 25-27 Juli 2009. [cited at p. 4]

[5] P. Behrens. Circstat: A matlab toolbox for circular statistics. *Journal of Statistical Software*, 31(10):1–21, Sept. 2009. [cited at p. 65]

[6] J. Benesty, J. Chen, and Y. Huang. *Acoustic MIMO Signal Processing (Signals and Communication Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. [cited at p. 2, 17]

[7] J. Benesty, J. Chen, and Y. Huang. *Microphone Array Signal Processing*. Springer, Berlin, 2008. [cited at p. 2, 3, 4, 17]

[8] L. Beranek. *Acoustics*. Amer. Inst. of Physics, 1986. [cited at p. 21]

[9] J. Blauert. *Spatial Hearing - Revised Edition: The Psychophysics of Human Sound Localization*. The MIT Press, 1996. [cited at p. 4]

[10] M. Brandstein and D. Ward. *Microphone Arrays: Signal Processing Techniques and Applications*. Springer, Berlin, 2001. [cited at p. 3, 17, 84]

[11] R. C. Cabot. An introduction to circular statistics and its application to sound localization experiments. In *58th AES Convention, November 4-7,1977 New York*, 1977. [cited at p. 12, 64]

[12] P. Cotterell. *On the Theory of the Second Order Soundfield Microphone*. PhD thesis, University of Reading, 2002. [cited at p. 23, 25]

[13] R. Derkx. Optimal azimuthal steering of a first-order superdirectional microphone response. In *Proceedings of the 11th International Workshop on Acoustic Echo and Noise Control Seattle, Washington USA*, 2008. [cited at p. 28]

[14] J. DiBiase, H. Silverman, and M. Brandstein. *Robust Localization in Reverberant Rooms*, chapter 8, pages 157–180. In Brandstein and Ward [10], 2001. [cited at p. 2]

[15] J. H. DiBiase. *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays*. PhD thesis, Brown University, Providence RI, USA, May 2000. [cited at p. 3]

[16] C. A. Dimoulas, K. A Avdelidis, G. M. Kalliris, and G. V. Papanikolaou. Sound source localization and B-format enhancement using soundfield microphone sets. In *Proceedings of the 122nd AES Convention, Vienna*, May 2007. Convention Paper 7091. [cited at p. 4]

[17] J. Eargle. *The Microphone Book*. Focal Press, 2nd edition, 2005. [cited at p. 21, 23, 24]

[18] G. Eckel, D. Pirrò, and G.K. Sharma. Motion-enabled live electronics. In *Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference, 23-25 July 2009, Porto - Portugal*, 2009. [cited at p. 67]

[19] G.E. Elko and J.Meyer. Microphone arrays. In J. Benesty, M. M. Sondhi, and Y. Huang, editors, *Springer Handbook of Speech Processing*, chapter 50, pages 1021–1041. Springer, 2007. [cited at p. 2, 28, 29, 30, 81]

[20] A. Farina. Simultaneous measurement of impulse response and distortion with a swept-sine technique. In *108th AES Convention, Paris*, pages 18–22, 2000. [cited at p. 18]

[21] A. Farina. Advancements in impulse response measurements by sine sweeps. In *122nd AES Convention, Vienna*, May 2007. [cited at p. 18]

[22] C. Feldbauer. Analyse, Resynthese und Interpolation von KFZ-Innengeräuschen. Master's thesis, Institute of Electronic Music and Acoustics, University of Music and Dramatic Arts, Graz, Austria, 2000. [cited at p. 41, 42, 50]

[23] M. Gerzon. The design of precisely coincident microphone arrays for stereo and surround sound. In *50th AES Convention, London*, March 1975. [cited at p. 4, 33]

[24] B. Gunel. Loudspeaker localization using B-format recordings. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 154–157, October 2003. [cited at p. 4]

[25] B. Gunel, H. Hachabiboglu, and A.M. Kondoz. Wavelet packet based analysis of sound fields in rooms using coincident microphone arrays. *Applied Acoustics*, vol. 68(7):778–796, 2007. [cited at p. 4]

[26] B. Gunel, H. Hachabiboglu, and A.M. Kondoz. Acoustic source separation of convolutive mixtures based on intensity vector statistics. *IEEE Transactions on Audio, Speech and Language Processing*, 16(4):154–157, May 2008. [cited at p. 4]

[27] T. Habib, M. Kepesi, and L. Ottowitz. Experimental evaluation of the joint position-pitch estimation (POPI) algorithm in noisy environments. In *Proc. 5th IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 369–372, 2008. [cited at p. 3, 72]

[28] E. Hoyer and R. Stork. The zoom FFT using complex modulation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '77*, volume 2, pages 78–81, 1977. [cited at p. 48]

[29] H-D. Kim, K. Komatani, T. Ogata, and H. G. Okuno. Localization over entire azimuth range for moving talkers. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France*, Sept, 22-26 2008. [cited at p. 2, 23]

[30] H. Kuttruff. *Room acoustics*. Taylor & Francis, October 2000. [cited at p. 5, 14, 16]

[31] H. Kuttruff. *Acoustics - An Introduction*. Taylor & Francis, 2007. [cited at p. 14, 24]

[32] G. Lathoud. *Spatio-Temporal Analysis Of Spontaneous Speech With Microphone Arrays*. PhD thesis, Ecole polytechnique fédérale de Lausanne EPFL, 2006. [cited at p. 59]

[33] E. A. Lehmann and A. M. Johansson. Particle filter with integrated voice activity detection for acoustic source tracking. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007. Article ID 50870, 11 pages. [cited at p. 59]

[34] A. Lombard, T. Rosenkranz, H. Buchner, and W. Kellermann. Multidimensional localization of multiple sound sources using averaged directivity patterns of blind source separation systems. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan*, pages 233–236, Apr. 2009. [cited at p. 2]

[35] J. Strutt (Lord Rayleigh). *The Theory of Sound - Volume Two*. Dover Publications, 2nd edition, 1945. [cited at p. 13, 23]

[36] J.A. MacDonald. A localization algorithm based on head-related transfer functions. *J. Acoust. Soc. Am.*, 123:4290–6, June 2008. [cited at p. 5]

[37] J. Merimaa and V. Pulkki. Spatial impulse response rendering 1: Analysis and synthesis. *J. Audio Eng. Soc.*, 53(12):1115–1127, December 2005. [cited at p. 4, 43]

[38] S. Mueller and P. Massarani. Transfer function measurement with sweeps. *J. Audio Eng. Soc.*, 49(6), June 2001. [cited at p. 18]

[39] K. Nakadai, H. Nakajima, Y. Hasegawa, and H. Tsujino. Sound source separation of moving speakers for robot audition. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 0:3685–3688, 2009. [cited at p. 2]

[40] A. Oppenheim, R. Schafer, and J. Buck. *Discrete Time Signal Processing*. Prentice Hall, 2 edition, 1999. [cited at p. 8, 17, 22, 48, 57]

[41] L. Ottowitz. Acoustic source localization with a circular microphone array. Master's thesis, Signal Processing and Speech Communication Lab (SPSC), Graz University of Technology, 2008. [cited at p. 3]

[42] D. G. Stork R. O. Duda, P. E. Hart. *Pattern Classification (2nd Edition)*. Wiley - Interscience, 2001. [cited at p. 36]

[43] J. Ramirez, J. C. Segura, C. Benitez, A. De La Torre, and A. Rubio. Efficient voice activity detection algorithms using long-term speech information. *Speech Communication*, 42:271–284, 2004. [cited at p. 59, 62]

[44] Y. Rui, D. Florencio, W.Lam, and J. Su. Sound source localization for circular arrays of directional microphones. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. (Proceedings ICASSP '05)*, volume 3, pages 93–96, 18-23 March 2005. [cited at p. 3]

[45] A. Saxena and A. Ng. Learning sound location from a single microphone. In *2009 IEEE International Conference on Robotics and Automation*, May 2009. [cited at p. 1, 5]

[46] J. O. Smith. *Physical Audio Signal Processing, December 2008 Edition*. http://ccrma.stanford.edu/ jos/pasp/, accessed 24-09-09. [cited at p. 41]

[47] G. Stan, J.J. Embrechts, and D. Archambeau. Comparison of different impulse response measurement techniques. *J. Audio Eng. Soc.*, 50(4):249, April 2002. [cited at p. 18]

[48] R. Stewart and M. Sandler. Statistical measures of early reflections of room impulse responses. In *Proc. of the* 10^{th} *Int. Conference on Digital Audio Effects (DAFx-07), Bordeaux, France*, September 10-15 2007. [cited at p. 14]

[49] R. Takashima, T. Takiguchi, and Y. Ariki. Monaural sound-source-direction estimation using the acoustic transfer function of an active microphone. In *FUSION '09, 12th International Conference on Information Fusion*, July 2009. [cited at p. 1]

[50] H. Viste and G. Evangelista. Binaural source localization. In *Proc. of the* 7^{th} *Int. Conference on Digital Audio Effects (DAFx'04), Naples, Italy*, Oct. 5-8 2004. [cited at p. 5]

[51] V. Willert, J. Eggert, J. Adamy, R. Stahl, and E. Korner. A probabilistic model for binaural sound localization. In *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, volume 36, pages 982–994, October 2006. [cited at p. 5]

[52] C. Zhang, D. Florencio, D.E. Ba, and Z. Zhang. Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings. *IEEE Transactions on Multimedia*, 3(3):538–548, 2008. [cited at p. 2]

[53] L. Ziomek. *Fundamentals of Acoustic Field Theory and Space-Time Signal Processing*. CRC Press, 1994. [cited at p. 13]

[54] J. M. Zmoelnig, A. Sontacchi, and W. Ritsch. The IEM-Cube, a periphonic re-/production system. In *AES 24th International Conference on Multichannel Audio*, June 2003. [cited at p. 67]

[55] F. Zotter. *Analysis and Synthesis of Sound Radiation with Spherical Arrays*. PhD thesis, Institute of Electronic Music and Acoustics, University of Music and Performing Arts, Austria, 2009. [cited at p. 28, 30]

Appendices

Appendix A

Implementation Details

The localization algorithm was implemented in Matlab. Basically, array recordings can be loaded and the corresponding position estimate over time is calculated. The algorithm works offline, so no real-time input from a soundcard is supported. The code could however be modified quite easily to do so, e.g. by using the freeware Matlab utility `playrec`[1] . However, in the present version the only package needed in addition to standard-Matlab is the signal processing toolbox.

A.1 Overwiew

The implementation should meet the following demands:

- flexible design, open to further development
- simple usage for fast and easy results

This tradeoff was tackled by putting all algorithmic functionality into a single Matlab-function which includes several sub-functions. It gives access to source localization and array steering of recorded audio by a single line of code:

```
result = CATfunction(sig);
```

In the simplest case, `sig` is the M-channel input audio signal matrix and `result` is a structure that contains the resulting variables, such as the estimated position for each frame. Neither signal samplerate, nor a reference database have to be specified, though they are required for operation. This is possible because all signal and algorithm parameters have default values. The signal samplerate is e.g presumed

[1] http://www.playrec.co.uk

to be 44.1kHz and default reference features are used. To advance from simple to flexible usage, all these default values can be overridden by user input.
Up to three structures may be provided as an input. These are:

- signal
- reference database
- settings

Their fields and default values are be detailed below. On the output side, in addition to the structure `result` all used settings `se` and changes to the default settings `seUser` can be logged.
A function call including all possible input and output arguments looks like this:

```
[result, seUser, se] = CATfunction(sig, mySe, ref);
```

The output structure `se` lists all settings that were used in the current function call. This can be used for documentation and check as well as for getting an overview of parameters that can be altered.
With the fields of structure `mySe` it possible to adjust the algorithm in detail. A simple example shall illustrate the principle: If `mySe` does not contain a field `.olap` the default value 0.25 is used for frame overlap. If this field however exists, e.g. `mySe.olap = 0.5`, the default frame overlap is overruled. In the given case, 50 % overlap would be used for framing the audio signal.
Similar to this, many algorithm parameters can be altered from their default value, which can change localization performance quite drastically. Since many parameters influence each other, even in a nonlinear way, an optimal parameter set can hardly be found. Thus, the default values were defined based on a experimental method.
Figure 5.1 shows a basic flow graph of the localization function.

A.2 Parameter List and Description

This section lists and describes the input and output variables of the Matlab - function `CATfunction` which was introduced in chapter A. As a basis for the naming of the input and output structures, consider the function call:

```
[result, seUser, se] = CATfunction(sig, mySe, ref);
```

This function call consists of 3 input structures and 3 output structures. The input structures are

- sig: Array signal data
- mySe: Changes to default algorithm settings
- ref: Reference database

The output structures are

- result: Position estimate, error measures, beamformer signal, etc.
- seUser: Changes to default settings, basically a copy of mySe
- se: All settings

In the following, the fields of these structures are discussed in detail. To keep things short, a telegram-style is used.

A.2.1 Signal input

All data related to the array signal is passed to the function by a single structure, which is referred to as `sig` in this document. `sig` consists of the following fields:

sig.x Input signal matrix
M channels of digital audio. Each channel is a column of the matrix, i.e. a M-channel array recording has M columns. The channel definition is important: first omni-microphone, then cardioids. If the coordinate system definition or order of cardioids is different from the database, the detected angles will be shifted. If the omni-microphone is not the first channel the result can be arbitrarily wrong, both with similarity and vector approach.

sig.fs Samplerate of sig.x in Hz

sig.pos Groundtruth position tags
For performance evaluation only. If sig.pos is not specified, no error measures can be computed. Format is `[t,az,el,r]` with units `[s,degrees,degrees, m]`

Each new time-mark, i.e change in position, is a new row of the matrix.

A.2.2 Settings

All settings regarding the algorithm can be passed to the function by a single structure `mySe`. As described in the previous chapter,`mySe` need not contain all possible fields, but only those that are desired to be altered from their default. Below, all fields of the settings structure `se` are listed. These can all be altered by passing a structure `mySe` with the corresponding fields to the localization function.

Basic Settings

se.principle Similarity or intensity vector approach
Three different algorithm principles can be chosen. The string `'Sim'` selects the similarity-, `'TDIV'` the time domain intensity vector approach. `'FDIV'` the frequency domain vector intensity vector approach. Example: `se.principle = 'Sim'`

se.compute Activate localization for az, el, r
Example: `se.compute = [1 0 0]`. Here, only azimuth detection is enabled. No elevation or radius detection is made. Can save computation time if e.g. only azimuth is relevant.

se.refFile Filename of default reference database.
Example: se.refFile = 'akg1' loads data file 'akg1.mat', which must contain a reference database structure called `ref` with fields as described in the reference database section.

se.resample Resampling to 11025 Hz
Binary flag. If 1, signal is resampled to 44100 Hz to 11025 and 48000 Hz is resampled to 12000 Hz. If 0 or if sig.fs is 11025 or 12000: no resampling is performed. Resampling is usefull for restricting frequency range, i.e. reducing data amount for the localizer. Beamformer always works with original and not downsampled signal!

Frame settings

se.frameDur Maximum duration of short frame in seconds.
Equivalent frame length in samples is rounded down to be a power of 2. Example: `se.frameDur = 0.05` equals 551.25 samples at 11025 Hz. Thus, resulting frame length is 512 samples.

se.olap frame overlap of short frame.
Example: `se.olap = 0.25` results in 50% frame overlap.

se.Mfact Length ratio long vs. short frame.
Radius detection works with longer frames for better frequency resolution. Angular detection however should use shorter frames for better time-resolution. M is an integer specifying how many times the long framelength is larger than short framelength. It is only relevant for radius detection. Example: `se.Mfact = 4` makes long frame 4 times longer than short frame.

Frequency analysis and peak picking settings

se.winType Window type
Example: `se.winType ='hamming'` windows frames with a hamming window. Window function from signal processing toolbox is used. Thus a wide range of window-functions can be used. For more information type `help window` in Matlab.

se.fb Frequency band
Upper and lower frequency bound. Example: `[400 4000; 1500, 2500; 50 200]` estimates azimuth between 400 and 4000 Hz, elevation in between 1500 and 2500 Hz and radius from 50 to 200 Hz.

se.fbExcl Frequency band exclusion
Frequencies to be excluded. Example: `[1500 2500; 0, 0; 0 0];` excludes the range 1500-2500 Hz from azimuth estimation.

se.zpadFact Zero padding factor
FFT - length of a length L frame is NFFT = L if no zero padding is applied. NFFT = zpadFact*L Example: `se.zpadFact = [1 4]`

se.Np Number of peaks
Example: `se.Np = [10 3 2]`

se.pkthresh Peak picking threshold
Absolute threshold that peaks must exceed for being recognized by peak - picker

Feature, database and similarity settings

se.featType Feature type
Type of features to be used: 1 = abs(card)/abs(omni) and 2 = log(card/card) Choice 2 was only better than 1, when gain of omni was different than gain of omni in ref.database

se.interp Result interpolation
Binary flag for switching interpolation of result az,el,r on/off If 0, only grid positions of the reference database can be detected. Example: `se.interp = [1 0 0]`

se.nsyFeat Noisy feature reference database
Binary flag. se.nsyFeat = 1 activates the noisy reference feature method.
se.nsyFeat = 0 deactivates it.

se.SNRsets SNR sets used with noisy features
The standard setting is se.SNRsets = [-30:10:40]; This means SNRs from -30 dB up to 40 dB with 10 dB stepsize are simulated. Using more SNR-values results in heavier computational load.

se.meanFree Mean free features
This should be activated, i.e. set to 1, if se.nsyFeat = 1. Basically, this removes the mean/trend from the features. Feat = Feat-mean(Feat)+ 0.5;

se.refIntMet ; Reference database interpolation method
Interpolate frequency bins of reference database to meet the peak frequencies found in current frame (frequency aligning). 'none' employs fast nearest neighbor search. All other possibilities (e.g 'linear') use Matlabs `interp1` function and consume much more computation time. For more information see `help interp1` .

se.simMeas Similarity measure
Integer that selects similarity measure.

- se.simMeas = 0: Cosine similarity
- se.simMeas = $p > 0$: Lp-norm based similarity

The setting `se.simMeas = 2` is equal to `se.simMeas = 'euclid'`

Frame reliability settings

Here, settings concerning the frame reliabilty weighting are listed. Most of them affect the coefficient b_l in eq. (5.31).

se.QMoff Quality measures switch-off
Binary flag for switching off all quality measures. se.QMoff = 1 results in $b_l = 1$. Consequently, every frame is trusted and no reliability weighting is made.

se.K Speed factor K in eq. (5.35)
$K > 1$ makes the estimate faster, $K < 1$ makes it slower. The filter coefficient b_l is defined as

$$b_l = K \cdot \cdot V_{S_l}$$

, where V_{S_l} is the scaled similarity curve variance (see sec. 5.1.6).

Fieldname	Description	Size	Preset Value
.x	Input signal matrix	Lx, M	-
.fs	Samplerate of sig.x	1	44100
.pos	Groundtruth position tags	x, 4	-

Table A.1: Fields of signal input structure `sig`

Beamformer settings

Now, settings concerning the beamformer are listed.

se.beamform Beamforming on/off

Binary flag. Only set $se.beamform = 1$ if you are interested in the resulting beamformer audio signal. Otherwise it is a waste of computation time.

se.omnibeam Omni-microphone W-channel

Binary flag. If $= 0$, only the cardioid microphones are used for beamforming. If $= 1$, the omni-directional microphone is used as the beamformer W-channel. This usually achieves a nicer low end sound. It can however ruin the beamformer directivity if the gain of the omni-channel is different as expected.

se.wbeam W-channel gain

This value defines the directivity. If se.wbeam=0, the beam is a figure 8. If se.wbeam=1 it is a cardioid (provided that the gain of the omni-channel is as assumed).

se.beamAz Fixed azimuth steering angle

se.beamAz is the desired azimuth steering angle in degree. With this, steering towards a certain, fixed direction can be made. If the field se.beamAz is not empty, the position estimate is ignored, and se.beamAz is used for steering.

A.2.3 Result

`result` is the main output structure and provides the results of the position estimation. They can be divided into 4 categories. These are

- position estimate
- position estimate error
- beamformer
- internal algorithm parameters

pos Position estimate $\hat{\mathbf{\Theta}}_l$ over time
The l^{th} column of `pos` is the 3-dimensional position estimate, i.e. azimuth, elevation and radius, for frame number l. The total number of frames is L. Thus, the size of `pos` is $[L, 3]$.

truepos Groundtruth position $\mathbf{\Theta}_l$
`truepos` is the groundtruth position, in the same format as `pos`. Consequently, the difference `truepos−pos` is the estimation error.

pos_noqm Position estimate $\hat{\mathbf{\Theta}}_l$ with no frame quality measures employed.
Here no frame reliability measures are employed, i.e. $b_l = 1$. This is useful for debugging, i.e. finding settings.

errorm Error measures structure
A structure that includes several error-measures. The fields of `result.errorm` are the metrics described in detail in chapter 6.2.

- `acc`: Accuracy Δ_d in percent.
- `mae`: Mean absolute error
- `rms`: Root mean square
- `bias`: Bias
- `std`: Standard deviation

The size of each field is $[1, 3]$. The only exception is `acc`, a $[D, 3]$ matrix for the accuracy Δ_d . Here the first dimension is used for varying the maximum tolerated error d. For the angular quantities there is currently `D=30` with `d=1:D`, i.e. the tolerated error angle is varied between 1 and 30 degrees with 1 degree resolution.

sig Beamformer output signal
The Beamformer is steered towards φ_l. It depends on the beamformer settings whether the position estimate, the groundtruth, or a fixed angle is used as a steering angle φ_l. If φ_l is equal the source angle, `result.sig` reproduces the sound source with good SNR.

sigb Beamformer output reverse steered
`sigb` is the beamformer output signal t, when the steering angle is $\varphi_b = \varphi + 180$, i.e. exactly opposite to the angle used for `result.sig`.