Technische Universität Braunschweig

Institut für Nachrichtentechnik

Master Thesis

Localization of Multiple Sound Sources

Jan Philip Janssen
Matrikelnummer: 4689492

26.02.2018

Technische Universität Braunschweig
Institute for Communications Technology
Schleinitzstraße 22 · 38106 Braunschweig

Examiner: Prof. Dr.-Ing. Tim Fingscheidt
Supervisor: Dr.-Ing. Tobias Wolff (Nuance Communications)
Maximilian Strake

Eidesstattliche Erklärung

Jan Philip Janssen

Contents

List of Tables

List of Figures

List of Abbreviations

IOT internet of things

HMI human machine interface

STFT short-time Fourier transform

TDOA time delay of arrival

DOA direction of arrival

EM expectation–maximization

ASL acoustic speaker localization

D&S delay and sum

IIR infinite impulse response

kNN k-nearest neighbor

ANN artificial neural networks

SVM support vector machines

GMM Gaussian mixture model

WGMM wrapped Gaussian mixture model

ML maximum likelihood

MAP maximum a posteriori

PHAT phase transform

SRP steered response power

SRP-PHAT steered response power with phase transform

GCC generalized cross correlation

MUSIC multiple signal classification

CSSM coherent signal-subspace method

MVDR minimum variance distortionless response

GM-PHD Gaussian mixture probability hypothesis density

UCA+C uniform circular array with center microphone

ULA uniform linear array

TTL time-to-live

CISC circular integrated cross spectrum

RMSE root mean square error

FP false positive

TP true positive

PDF probability density function

MEMS MicroElectrical-Mechanical system

GT ground truth

ST system

IDC ID change

List of Symbols

M	number of microphones
Q	number of sources
$\underline{\breve{r}}_m$	position of microphone m
$\underline{\overset{*}{r}}_q$	position of source q
n	discrete time index
$x_m(n)$	signal recorded by microphone m
$s_q(n)$	signal emitted by source q
$h_{mq}(n)$	room impulse response between the q-th source and the m-th microphone
$v_m(n)$	noise considered at microphone m
$*$	convolution operator
$(\cdot)^T$	transpose operator
f	frequency
f_s	sampling rate
Ω	normalized frequency
$X_m(\Omega)$	signal recorded by microphone m
$S_q(\Omega)$	signal emitted by acoustic source q
$H_{mq}(\Omega)$	acoustic transfer function between the q'th source and the m'th microphone
$V_m(\Omega)$	noise considered at microphone m
$\underline{A}_q(\Omega)$	propagation vector of source q
r_0	reference point's position
τ_{mq}	time delay of a microphone position
$\Delta\tau_{mq}$	TDOA = difference between time delays
φ	angle representing the direction of arrival (DOA) in the assimuth plane
c	speed of sound
$E\{\cdot\}$	expectation value operator
$(\cdot)^*$	complex conjugate operator
$(\cdot)^H$	Hermitean operator
$\mathbf{\Phi}_{ss}(\Omega)$	source correlation matrix
$\mathbf{\Phi}_{vv}(\Omega)$	noise correlation matrix

$\underline{A}_q^{\mathrm{ff}}$	propagation vector function in free field model
j	imaginary unit
$\mathbf{\Phi}_{xx}(\Omega)$	spatial covariance matrix
$Y(\Omega)$	output signal of the beamformer
ϕ_{yy}	power of output signal
$\underline{W}(\Omega)$	vector of beamformer filter weights
$\underline{D}(\theta)$	steering vector
θ	steering direction, azimuth angle
$P(\Omega, \theta)$	steered power
$\overline{P}^{\mathrm{b}}(\theta)$	average of steered power over frequencies
ϵ	small additive component
$\mathbf{I}$	the unit matrix
$\mathbf{J}_{xx}(\Omega)$	coherence matrix
l	is the discrete frequency bin
$\mathcal{W}(n)$	the window function
O	frameshift in samples
b	frame index
L	STFT window length
N_E	number of samples
$\hat{\varphi}$	DOA estimate
α	smoothing constant
$\mathcal{N}(\hat{\varphi}\|\mu, \sigma)$	single Gaussian probability density function (PDF)
K	number of Gaussians
σ	standard deviation
μ	mean of the Gaussian
π	mixing coefficients
N_k	the number of observations of class k
$p_{\mathrm{GMM}}(\underline{\hat{\varphi}}\|\underline{\pi}, \underline{\mu}, \underline{\sigma})$	joint likelihood for observations $\underline{\hat{\varphi}}$ given the GMM parameter
$p_k(\hat{\varphi})$	single Gaussian PDF of class k
γ_{kn}	responsibilities of observation n to class k
Γ	threshold parameter
$c(l, b)$	confidence value
$\underline{\lambda}$	all parameter of the GMM
μ_{v}	sensitivity for confidence calculation
α_{o}	offset for confidence calculation
β	MAP-adaption parameter
b_{TTL}	time-to-live variable

1 Introduction

With the rise of virtual assistants like Amazon Alexa, Apple Siri or the Google Assistant the trend of voice enabled internet of things (IOT) devices and smart speakers began. Audiences and consumers are rapidly adopting smart speakers. At the moment (early 2018) the ownership is at 16% of the U.S. population after 3 years [MR17]. Gartner, a provider for market research results and analysis, predicts that 75% of U.S. households will have smart speakers by 2020 [FM18]. Smart speakers are using speech recognition as the human machine interface (HMI). Therefore, new challenges for speech enhancement algorithms arise to enable a robust speech recognition in difficult acoustic situations. For example the desired speaker may be several meters away from the device while other interfering sound sources are present as well. In state of the art speech enhancement systems microphone arrays are deployed to 'conduct' a spatial filtering known as beamforming. This is done to capture and enhance the desired acoustic signal and suppress other interfering sound sources.

For acoustic beamforming it is crucial to know where sound sources are located. Otherwise spatial filtering might degrade quality of the incoming desired signal. Many basic acoustic source localization methods have been developed which, however, can only identify the most dominant sound source at a time. Well known examples are the generalized cross correlation (GCC) or the steered response power (SRP) method [BW13, Chapter 8]. Also, more complex algorithms such as multiple signal classification (MUSIC) or coherent signal-subspace method (CSSM) have been proposed for localizing multiple sources simultaneously [KS89; WK85]. In order to control beamformer-steering in noisy environments robust DOA estimates are required. In practical scenarios the aforementioned localization methods, however, suffer from noise and reverberation and therefore, their DOA estimates cannot be used directly to control a beamformer. Furthermore, multi-source localizers such as MUSIC are computationally more complex as compared to the SRP method.

The goal of this work is therefore to develop a computationally efficient localization algorithm for multiple speakers which can be used with a uniform circular array with center microphone (UCA+C). Circular microphone arrays allow for azimuth-independent beamsteering and are therefore being used in many smart speaker applications. As a baseline, the techniques to be developed should build

upon the broadband SRP method as this has proven robust and represents the current state of the art technique at Nuance.

The focus lies on the development of a post-processor for the SRP as core localizer which classifies the raw localization results into several classes whereas each class represents an acoustic source. In the smart speaker application the azimuth angle is practically more important than the elevation angle. Therefore, source localization is only considered with respect to the azimuth plane in this work. While being simple and robust, the SRP method, however, can only identify the most dominant sound source at a time. Even if two sources differ with respect to their spectral content they cannot be distinguished. Therefore, it is desirable to develop a classifier that is not limited to the SRP as a core localizer and that can be extended to work with more sophisticated localization methods using spectral localization.

The classifier proposed in this thesis uses a Gaussian mixture model (GMM) to represent the raw DOA results. To allow for tracking of time varying source positions the mean of each Gaussian is estimated adaptively over time. To achieve this maximum a posteriori (MAP) estimation known from speaker verification with GMMs has been adopted [GL94]. In the application of smart speakers a 360° localization is required which motivates the use of circular arrays. The classification methods known from the literature, however, do not allow for classification of circular data. This problem is solved here by using a so called wrapped Gaussian mixture model (WGMM) which is a periodic extension of a standard GMM. As mentioned above, the raw DOA estimates usually suffer from noise and reverberation. Because of these effects the observed data are not always reliable. In order to increase the robustness of the classification process a confidence metric has been incorporated into the proposed classifier. Finally, the effect of spatial aliasing has also been considered to achieve a robust performance for the considered microphone array.

All considered algorithms have been implemented in *Matlab*. The described multi-source localization is analyzed extensively using a large number of recordings covering a variety of acoustic scenes. The recorded data have been labeled manually to obtain the ground truth and an analysis framework has been implemented to judge the estimated source positions using a set of metrics. As a reference a well known multi-source localizer as proposed by Madhu has been chosen [MM08]. This method also uses a GMM but is computationally more complex and does not use MAP adaptation. The proposed method is analyzed using 4 different localizers.

The work is structured as follows: In chapter 2 the basics are stated for the localization and the post-processing. Here, the signal model is defined and beamforming is

introduced. Based on this, the acoustic source localization method SRP is discussed. Next, the source classification with a GMM and an expectation–maximization (EM)-algorithm is introduced. To adapt the classification method for circular arrays the wrapped Gaussians are stated. In the end of the basic chapter the state of the art method is discussed. In chapter 3 the proposed classification algorithm is described in detail. Chapter 4 then contains the evaluation and discusses all results and findings for each considered scenario. Finally, in chapter 5 the whole work is summarized and the prospects are stated.

2 Basics

In this chapter, the foundations for the developed multi-source localization algorithm are laid. First, the signal model is introduced, and the basic assumptions, as well as the setup, are explained. Based on this, beamforming and the estimation of the closely related DOA are discussed. Furthermore, the source classification using GMMs is introduced, as an important aspect of the developed classification algorithm. Finally, the state of the art of multi-source localization algorithms will be discussed and put in relation to each other.

2.1 Signal Model

In the beginning, the signal model has to be determined, which is the basis for the description of beamforming and acoustic localization. Considering a microphone array of M microphones, positioned at $\underline{\breve{r}}_m = (\breve{r}_{m,x}, \breve{r}_{m,y})^T$, receiving multiple signals over an acoustic channel from Q sources at position $\underline{\overset{*}{r}}_q = (\overset{*}{r}_{q,x}, \overset{*}{r}_{q,y})^T$ as illustrated in figure 2.1. In this thesis, an underscore is used to denote a column vector, and ^{T} is the transposed. The acoustic channel is formed by a superposition of the propagation on a direct path and a multitude of propagations on the indirect path, originating from the surrounding sources.
In figure 2.2 the equivalent block diagram for the signal arriving at microphone m is shown which can mathematically be described as

$$x_m(n) = \sum_{q=1}^{Q} s_q(n) * h_{mq}(n) + v_m(n), \ \forall \ m \in \{1, ..., M\}, \tag{2.1}$$

where $x_m(n)$ represents the signal recorded by a microphone m, $s_q(n)$ represents the signal emitted by source q and $h_{mq}(n)$ represents the room impulse response between the q'th source and the m'th microphone. They are connected by the convolution operator $*$. The microphones used are ideal and omnidirectional. Additionally, noise $v_m(n)$ is considered at any microphone m, which is uncorrelated across the microphone channels. For convenience, all signals are considered as time discrete signals, where n denotes the discrete time index.

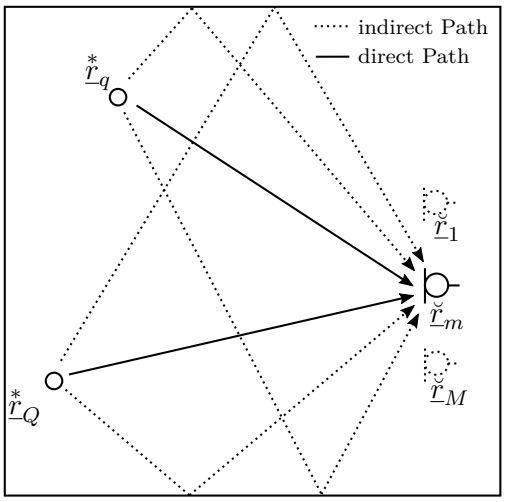


Figure 2.1: Schematic diagram of sound propagation in a room from Q sources to the microphone m

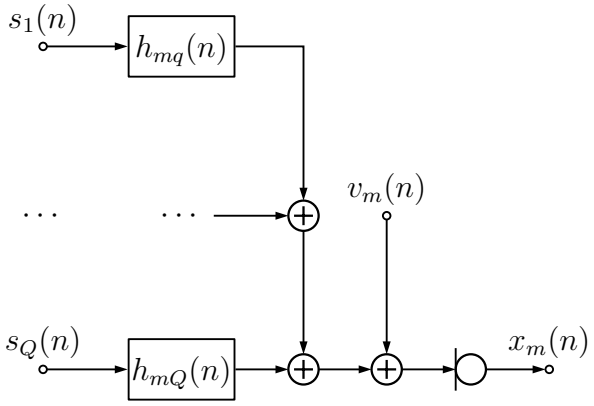


Figure 2.2: Equivalent block diagram for one microphone

Equation 2.1 may be brought from the time domain to the frequency domain by the Fourier transform:

$$X_m(\Omega) = \sum_{q=1}^{Q} S_q(\Omega) \cdot H_{mq}(\Omega) + V_m(\Omega), \ \forall \ m \in \{1, ..., M\}, \tag{2.2}$$

where $\Omega = 2\pi f / f_s$ denotes the normalized frequency variable, where f is the frequency and f_s is the sampling rate. Here and in the following, capital letters are used to denote frequency domain variables. Equation 2.2 may also be written in a matrix representation

$$\begin{aligned} \begin{pmatrix} X_1(\Omega) \\ \vdots \\ X_M(\Omega) \end{pmatrix} &= \begin{pmatrix} H_{11}(\Omega) & \dots & H_{1Q}(\Omega) \\ \vdots & \ddots & \vdots \\ H_{M1}(\Omega) & \dots & H_{MQ}(\Omega) \end{pmatrix} \begin{pmatrix} S_1(\Omega) \\ \vdots \\ S_Q(\Omega) \end{pmatrix} + \begin{pmatrix} V_1(\Omega) \\ \vdots \\ V_M(\Omega) \end{pmatrix} \\ \underline{X}(\Omega) &= \mathbf{H}(\Omega)\underline{S}(\Omega) + \underline{V}(\Omega), \end{aligned} \tag{2.3}$$

where bold letters are used for matrices.
Since the absolute acoustic transfer functions $H_{mq}(\Omega)$ can only be identified up to a filtering, a reference point r_0 is introduced. Typically one microphone is chosen as a reference microphone, whose transfer function $H_{Rq}(\Omega)$ is related to the source spectrum $S_q(\Omega)$.

This turns the matrix $\mathbf{H}(\Omega)$ into the matrix of *relative* transfer functions $\mathbf{A}(\Omega)$:

$$\begin{pmatrix} X_1(\Omega) \\ \vdots \\ X_M(\Omega) \end{pmatrix} = \begin{pmatrix} \frac{H_{11}(\Omega)}{H_{R1}(\Omega)} & \cdots & \frac{H_{1Q}(\Omega)}{H_{RQ}(\Omega)} \\ \vdots & \ddots & \vdots \\ \frac{H_{M1}(\Omega)}{H_{R1}(\Omega)} & \cdots & \frac{H_{MQ}(\Omega)}{H_{RQ}(\Omega)} \end{pmatrix} \begin{pmatrix} S_1(\Omega)H_{R1}(\Omega) \\ \vdots \\ S_Q(\Omega)H_{RQ}(\Omega) \end{pmatrix} + \begin{pmatrix} V_1(\Omega) \\ \vdots \\ V_M(\Omega) \end{pmatrix} \tag{2.4}$$

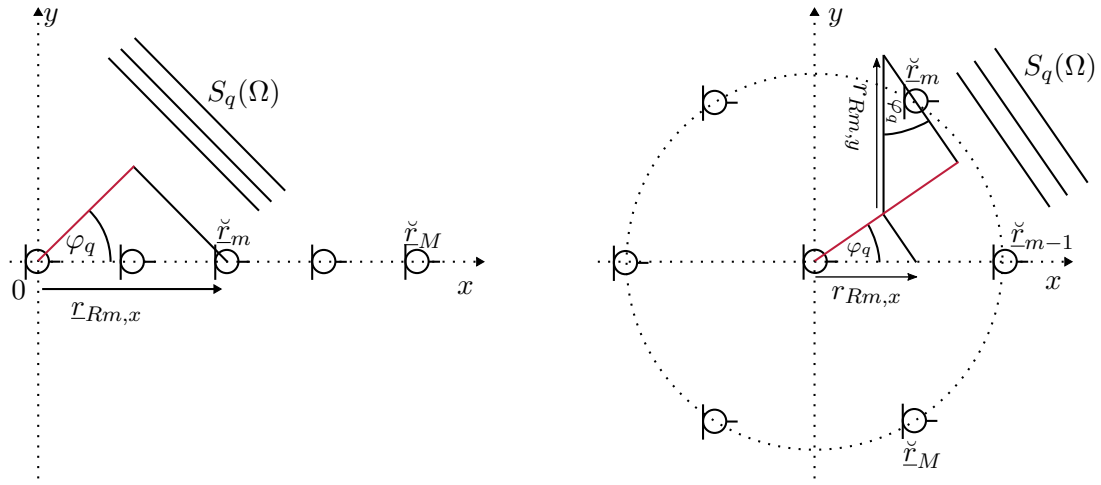
$$\underline{X}(\Omega) = \mathbf{A}(\Omega)\underline{S}'(\Omega) + \underline{V}(\Omega),$$

where $\mathbf{A}(\Omega)$ is the transfer function matrix in regards to the reference point. Each column of matrix $\mathbf{A}(\Omega)$ can be called propagation vector $\underline{A}_q(\Omega)$ for source q: $\mathbf{A}_q(\Omega) := (\underline{A}_1(\Omega), ..., \underline{A}_Q(\Omega))$. The transfer functions between the microphones and the reference point multiplied with the source signal is denoted as $\underline{S}'(\Omega)$. When dividing the elements of $\mathbf{A}(\Omega)$ in 2.4 into magnitude and phase, the time delay of arrival (TDOA) becomes visible

$$A_{mq}(\Omega) = \frac{H_{mq}(\Omega)}{H_{Rq}(\Omega)} = \frac{|H_{mq}(\Omega)|}{|H_{Rq}(\Omega)|} e^{-j\Omega f_s(\tau_{mq} - \tau_{Rq})}. \tag{2.5}$$

The difference between the time delays of a microphone position and the reference point $\Delta\tau_{mq} = \tau_{mq} - \tau_{Rq}$ represents the relative time delay or TDOA. [Mad10, Chapter 2]
Due to the unknown position of the acoustic source, the absolute time delays τ_{mq} and τ_{Rq} cannot be obtained. The difference, however, can be observed from 2.4. To later obtain the angle φ, which is the DOA, a model for the TDOA and the transfer matrix $\mathbf{A}(\Omega)$, respectively, is needed. Therefore the array geometry and the position of the reference point r_0 has to be known. Furthermore, some constraints are needed. Far-field is assumed, which means that the distance between sources and microphones is much greater than the distance between the different microphones $\|\underline{\breve{r}}_q^* - \underline{\breve{r}}_m\| \gg \|\underline{\breve{r}}_i - \underline{\breve{r}}_j\|,\ \forall\ i, j \in \{1, ...M \mid i \neq j\}$. In this case, the signal propagation can be assumed as a plane wave. Moreover, free field model is assumed which means that anechoic conditions or no reflections are considered. [Mad10, Chapter 2] In general, this time delay can be stated with the help of the vectors from the reference point to the microphone positions $\underline{r}_{Rm} = \underline{\breve{r}}_m - \underline{r}_0$ as

$$\Delta\underline{\tau}(\varphi_q) = \left(\frac{r_{R1,x}\cos(\varphi_q) + r_{R1,y}\sin(\varphi_q)}{c}, \quad \cdots, \right. \\ \left. \frac{r_{RM,x}\cos(\varphi_q) + r_{RM,y}\sin(\varphi_q)}{c} \right)^T, \tag{2.6}$$



(a) ULA (b) UCA+C

Figure 2.3: Outline of different array geometries, to calculate the time delay of arrival . The red line is the difference of distance the sound coming from direction φ_q has to travel between microphone m and the reference point.

where c is the speed of sound. To illustrate equation 2.6, two fundamental array geometries are considered. First, the uniform linear array (ULA) geometry as shown in figure 2.3a is used with a reference point $\underline{r}_0$ set to the first microphone. When utilizing this array in 2.6 the following is obtained for the TDOA:

$$\Delta\underline{\tau}_{\mathrm{ULA}}(\varphi_q) = \left(0, \quad \frac{r_{R2,x}\cos\varphi_q}{c}, \ \cdots \ , \ \frac{r_{RM,x}\cos\varphi_q}{c} \right)^T, \tag{2.7}$$

where φ_q is the DOA of source q in the azimuth plane and c is the speed of sound. Another important geometry is the UCA+C geometry seen in figure 2.3b. The reference point $\underline{r}_0$ is commonly set into the center which results in

$$\Delta\underline{\tau}_{\mathrm{UCA+C}}(\varphi_q) = \left(0, \quad \frac{r_{R2,x}\cos\varphi_q + r_{R2,y}\sin\varphi_q}{c}, \quad \cdots \ , \quad \frac{r_{RM,x}\cos\varphi_q + r_{RM,y}\sin\varphi_q}{c} \right)^T, \tag{2.8}$$

as the TDOA for an UCA+C. As stated in chapter 1 only a UCA+C with a radius of 42 mm is considered in this work. When using the free field model and the further constraints for the TDOA calculation in the propagation vector following is obtained

$$\underline{A}_q^{\text{ff}}(\Omega) = \exp\Big(- j\Omega f_s \Delta\underline{\tau}(\varphi_q)\Big), \tag{2.9}$$

where $A_q^{\text{ff}}(\Omega)$ is the propagation vector function for a source q in the free field model.

2.1.1 Spatial Covariance

Another important signal property is the covariance that describes the interdependencies between the microphone signals $\underline{X}(\Omega)$. To obtain this covariance, it is presumed that the signals are stochastic. When only considering one source ($Q = 1$), the spatial covariance matrix can be denoted as

$$\begin{aligned} \mathbf{\Phi}_{xx}(\Omega) &= \mathrm{E}\{\underline{X}(\Omega)\underline{X}^H(\Omega)\} \\ &= \underline{A}(\Omega)\mathrm{E}\{S'(\Omega)S'^*(\Omega)\}\underline{A}^H(\Omega) + \mathrm{E}\{\underline{V}(\Omega)\underline{V}^H(\Omega)\} \\ &= \mathbf{\Phi}_{ss}(\Omega) + \mathbf{\Phi}_{vv}(\Omega), \end{aligned} \tag{2.10}$$

where $E\{\cdot\}$ represents the expectation value operator, * denotes the complex conjugate operator, $\mathbf{\Phi}_{ss}(\Omega)$ represents the source correlation matrix, $\mathbf{\Phi}_{vv}(\Omega)$ the noise correlation matrix and $(\cdot)^H$ the Hermitean operator. For uncorrelated source signals $S_q(\Omega)$ and $Q > 1$ the spatial covariance is obtained as:

$$\mathbf{\Phi}_{xx}(\Omega) = \sum_{q=1}^{Q} \mathbf{\Phi}_{ss}^q(\Omega) + \mathbf{\Phi}_{vv}(\Omega), \tag{2.11}$$

where $\mathbf{\Phi}_{ss}^q(\Omega)$ is the covariance matrix of the q'th source signal. It is evident that the covariance matrix $\mathbf{\Phi}_{xx}(\Omega)$ is a superposition of the Q source covariance matrices $\mathbf{\Phi}_{ss}^q(\Omega)$ and the noise covariance matrix $\mathbf{\Phi}_{vv}(\Omega)$.

2.2 Beamforming

Beamforming or spatial filtering is an array processing technique used to improve the quality of the desired signal in the presence of noise. This filtering is accomplished by a linear combination of the recorded signals $X_m(\Omega)$ and the beamformer weights $W_m(\Omega)$. In other words, the filtered microphone signals are summed together (compare with figure 2.4). When the filter weights are configured correctly, the desired signal is superimposed constructively.

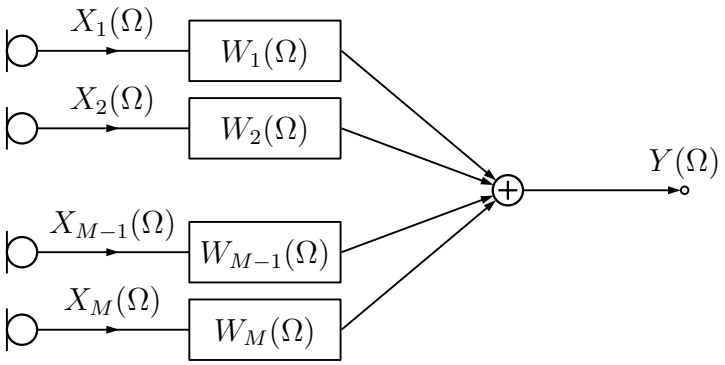


Figure 2.4: Filter and sum beamformer. Microphone signals $\underline{\mathrm{X}}(\Omega)$ are multiplied with the beamformer weights $\underline{\mathrm{W}}(\Omega)$ and then accumulated to the beamformer output signal $Y(\Omega)$.

In vector notation, the beamformer output signal can be written as

$$Y(\Omega) = \underline{W}^H(\Omega)\ \underline{X}(\Omega), \tag{2.12}$$

where $\underline{X}(\Omega)$ is defined in 2.4 as vector of microphone spectra, $\underline{W}(\Omega)$ represents the vector of beamformer filter weights and $Y(\Omega)$ is the output signal of the beamformer. $(\cdot)^H$ is the hermitian (self-adjoint) operator. This beamformer weighting vector $\underline{W}(\Omega)$ may be determined, by solving a constrained optimization problem. These constraints are using the power $\phi_{yy}(\Omega)$ of $Y(\Omega)$ which can be written as,

$$\begin{aligned}\phi_{yy}(\Omega) &= E\{Y(\Omega)Y^*(\Omega)\} \\ &= E\{\underline{W}^H(\Omega)\underline{X}(\Omega)\underline{X}^H(\Omega)\underline{W}(\Omega)\} \\ &= \underline{W}^H(\Omega)\mathbf{\Phi}_{xx}(\Omega)\underline{W}(\Omega).\end{aligned} \tag{2.13}$$

A very prominent approach to determine $\underline{W}(\Omega)$ is to minimize its output power $\phi_{yy}(\Omega)$ without distorting the desired signals. In the following, this beamforming approach is discussed in more detail.

2.2.1 Minimum Variance Distortionless Response Filter

The minimum variance distortionless response (MVDR) optimization criterion states that the output variance in equation 2.13 shall be minimized, under the constraint of no distortion in steering direction θ. This optimization can mathematically be described as

$$\begin{aligned} \underline{W}^H(\Omega)\mathbf{\Phi}_{xx}(\Omega)\underline{W}(\Omega) &\rightarrow \min_{\underline{W}(\Omega)} \\ \text{subject to } \underline{W}^H(\Omega)\underline{D}(\Omega,\theta) &= 1, \end{aligned} \tag{2.14}$$

where $\underline{D}(\Omega,\theta)$ is the steering vector, which is calculated similarly to the modeled propagation vector $\underline{A}_q^{\text{ff}}(\Omega)$ and uses the TDOA from equation 2.6: $\underline{D}(\Omega,\theta) = \exp(-j\Omega f_s \Delta\underline{\tau}(\theta))$. However, it represents the steering direction θ of the array and not the propagation direction φ also called DOA of the signal $S_q(\Omega)$. For convenience, the normalized frequency Ω is neglected but will be reintroduced when it is needed. The optimization may be solved using the technique of Lagrange multipliers [Lag11, Section 2, Chapter 4][Bro+08, Chapter 6.2.5.6], which results in

$$\underline{W}_{\text{CAP}}(\theta) = \frac{\mathbf{\Phi}_{xx}^{-1}\underline{D}(\theta)}{\underline{D}^H(\theta)\mathbf{\Phi}_{xx}^{-1}\underline{D}(\theta)}. \tag{2.15}$$

This beamforming technique is also known as Capon beamformer by the name of its inventor. [Cap69; KV96]

2.2.2 Delay and Sum Beamformer

The delay and sum (D&S) beamformer is a special case of the MVDR beamformer. The beamformer is optimized for the uncorrelated sound field as noise. To derive the D&S beamformer, the optimization criterion has to be changed from minimizing the output power ϕ_{yy} to minimizing the noise after the beamformer weights $\underline{W}^H\mathbf{\Phi}_{vv}\underline{W}$. The resulting weight vector changes from 2.15 to [1]

$$\underline{W}(\theta) = \frac{\mathbf{\Phi}_{vv}^{-1}\underline{D}(\theta)}{\underline{D}^H(\theta)\mathbf{\Phi}_{vv}^{-1}\underline{D}(\theta)}. \tag{2.16}$$

Assuming a spatial uncorrelated noise field, the noise power $\mathbf{\Phi}_{vv}$ reduces to a unit matrix $\mathbf{\Phi}_{vv} \rightarrow \phi_{vv}\mathbf{I}$. With this assumption used in 2.16, the D&S weights can be stated as

$$\begin{aligned} \underline{W}_{\text{DS}}(\theta) &= \frac{\phi_{vv}\underline{D}(\theta)}{\phi_{vv}\underline{D}^H(\theta)\underline{D}(\theta)} \\ &= \frac{\underline{D}(\theta)}{M}. \end{aligned} \tag{2.17}$$

[1] Note that equation 2.15 and 2.16 are equal only if $\underline{D}(\theta) = \underline{A}_q(\varphi)$. If the assumed steering vector $\underline{D}(\theta) \neq \underline{A}_q(\varphi)$, which is usually the case in realistic acoustic environments, equation 2.15 will lead to signal distortions. Equation 2.16 on the other hand is more robust to steering vector mismatch as only the noise is minimized.

The number of microphones M is seen in the denominator because squaring reduces the steering vector, which consists of M unit vectors to the number of elements. This beamformer can be seen as time compensation before the summation. Therefore the name *delay & sum* beamformer is used. [BW13, Chapter 2]

2.3 Acoustic Speaker Localization

The acoustic speaker localization (ASL) techniques known from literature can broadly be divided into two parts. On the one hand the parametric approach exists. Here a multidimensional search has to be deployed to find all estimates at once. One of the most frequently used parametric approaches is the maximum likelihood (ML) technique. On the other hand there are the spectral-based algorithms, which are computationally more feasible than the preceding one, but may suffer from inaccuracy. In this approach, a spectrum-like function is formed, which can be searched for the parameters of interest. Spectral-based methods can again be subdivided into beamforming and subspace-based techniques. One of the most popular subspace-based techniques is MUSIC, which utilizes an eigenvalue decomposition, that again is computationally very complicated. Subspace-based techniques is a field of great research interests with the development of multiple techniques like CSSM [WK85], TOPS [YKM06], FRIDA [Pan+17] or WAVES [DP01]. In this work, the beamforming technique is used and will be discussed in more detail. [KV96]

2.3.1 Steered Response Power

The SRP method tries to estimate the DOA by steering a beamformer towards a set of possible directions and calculating the resulting power. In a second step, the angle that maximizes the power is chosen as the DOA estimate. This method can be applied with different kinds of beamformers.
The output power of the Capon beamformer can be written by inserting 2.15 in 2.13 as

$$\begin{aligned} P_{\mathrm{CAP}}(\Omega,\theta) &= \underline{W}_{\mathrm{CAP}}^{H}(\Omega,\theta)\boldsymbol{\Phi}_{xx}(\Omega)\underline{W}_{\mathrm{CAP}}(\Omega,\theta) \\ &= \frac{\underline{D}(\Omega,\theta)^{H}\boldsymbol{\Phi}_{xx}^{-1}(\Omega)}{\underline{D}^{H}(\Omega,\theta)\boldsymbol{\Phi}_{xx}^{-1}(\Omega)\underline{D}(\Omega,\theta)}\boldsymbol{\Phi}_{xx}(\Omega)\frac{\boldsymbol{\Phi}_{xx}^{-1}(\Omega)\underline{D}(\Omega,\theta)}{\underline{D}^{H}(\Omega,\theta)\boldsymbol{\Phi}_{xx}^{-1}(\Omega)\underline{D}(\Omega,\theta)} \\ &= \ldots \\ &= \frac{1}{\underline{D}^{H}(\Omega,\theta)\ \boldsymbol{\Phi}_{xx}^{-1}(\Omega)\ \underline{D}(\Omega,\theta)}. \end{aligned} \tag{2.18}$$

Respectively, when using 2.17 in 2.13, the power output of the D&S beamformer can be stated as

$$P_{\mathrm{DS}}(\Omega,\theta) = \frac{\underline{D}^H(\Omega,\theta)\ \mathbf{\Phi_{xx}}(\Omega)\ \underline{D}(\Omega,\theta)}{M^2}. \tag{2.19}$$

In the next step, the angle for the maximum steered power has to be found. This can be done in two ways. The broadband method takes the mean over all frequencies and searches for the maximum afterwards. In the frequency DOA estimation, the averaging over frequencies is dropped, and the maximum search is done in every frequency. The broadband method can be stated as following:

$$\begin{aligned}\hat{\varphi} &= \arg\max_{\theta} \frac{1}{2\pi}\int_{-\pi}^{\pi} P(\Omega,\theta)\mathrm{d}\Omega \\ &= \arg\max_{\theta} \overline{P}^{\mathrm{b}}(\theta).\end{aligned} \tag{2.20}$$

When averaging across frequencies is neglected, it yields a DOA estimate for every frequency, which is noted as

$$\hat{\varphi}(\Omega) = \arg\max_{\theta} P(\Omega,\theta). \tag{2.21}$$

Receiving one value per frequency may enable a real multi-source detection, under the assumption, that the sources are spectrally disjoint. Broadband and frequency SRP values are distinguished by the dependence on frequency Ω. The broadband SRP may also be weighted before mean taking. This can be done in different ways. One of them is the phase transform (PHAT). Mathematically this can be stated as

$$\begin{aligned}\hat{\varphi} &= \arg\max_{\theta} \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{\phi_{xx}(\Omega)} P(\Omega,\theta)\mathrm{d}\Omega \\ &= \arg\max_{\theta} \overline{P}^{\mathrm{b}}_{\mathrm{PHAT}}(\theta).\end{aligned} \tag{2.22}$$

The PHAT weighted steered mean power is stated as $\overline{P}^{\mathrm{b}}_{\mathrm{PHAT}}(\theta)$. It is assumed that the sound field is homogeneous, which means that the microphones have the same amount of variance $\phi_{xx}(\Omega)$.[2] Furthermore, the assumption decreases the computational load. This weighting leads to canceling out the absolute power. Thus only the phase is holding all information, and all spectral parts are having equal share in the broadband SRP in respect to the mean. [BW13, Chapter 8][MM08]

[2]Note that the variance ϕ_{xx} are the values on the main diagonal of the covariance matrix

2.3.2 Theoretical Steered Power Spectrum and Aliasing

The theoretical Steered Power Spectrum can be used to evaluate different types of beamformers. They are calculating the response of a microphone array to a wavefront with a certain DOA for all parameters. As stated in chapter 1 in this work only the azimuth-plane is considered. So the response is dependent on the azimuth angle θ and the normed frequency Ω. The theoretical steered power spectrum can be calculated, when considering a signal as in 2.4 but with neglecting the noise $\underline{V}(\Omega) = 0$ and setting the modified source signal $S'(\Omega) = 1$ which can be seen as a white noise signal. In dB, the SRP then reads[3]:

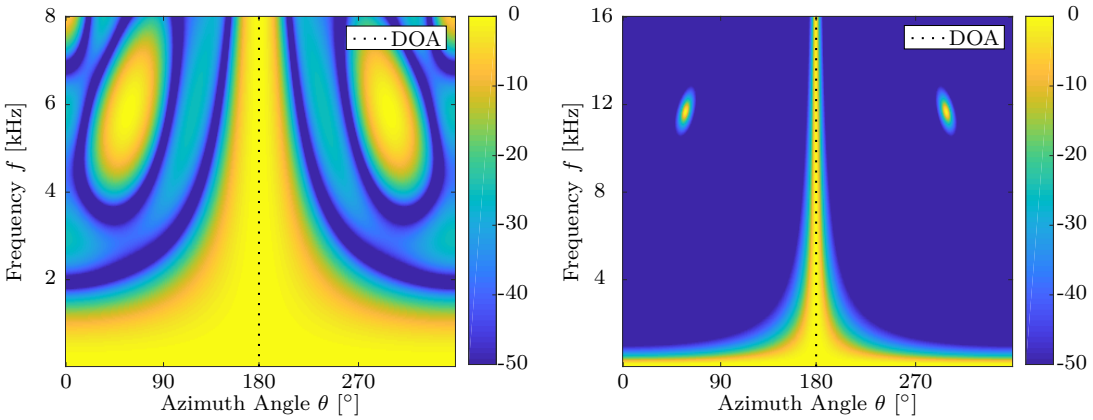
$$P_{\text{theory}}(\Omega, \theta) \mid_{\text{db}} = -10 \log_{10}(|\underline{W}^H(\Omega, \theta)\underline{A}(\Omega)|^2). \tag{2.23}$$

Different beamformer weight vectors $\underline{W}(\Omega, \theta)$ can be compared, for instance those of D&S beamformer 2.17 or Capon beamformer 2.15. Steering vector $\underline{D}(\Omega, \theta)$ and propagation vector $\underline{A}(\Omega)$ are dependent on the array geometry and can for example be calculated with 2.7 and 2.8 in 2.9. [BW13, Chapter 2] Referring to these beamformers the theoretical steered power spectra for an UCA+C arrangement are shown in figure 2.5. When comparing both spectra, it is observed that the spectrum for the Capon beamformer is much more distinct than the one for the D&S beamformer. However, in both plots the DOA $\varphi = 180°$ is visible. Mathematically the DOA can be obtained by employing $P_{\text{theory}}(\Omega, \theta)$ in 2.20 for broadband DOA or in 2.21 frequency DOA.

Another observation from figure 2.5 concerns the so called *grating lobes* in the upper frequencies at 60° and 300° azimuth angle θ. Reason is the spatial aliasing, which occurs when the microphone distance is greater than half of the sound's wavelength. The grating lobes have the same amount of power as the main lobe and therefore cannot be distinguished by the SRP algorithm when looking at single frequency bands. Aliasing is not only impacting the DOA estimation in specific frequencies but also the broadband SRP when it is desired to detect more than one source. For example, when multiple sources are present, and one source is more active than the other, one cannot decide between a second source which is less active or the power peak stimulated by aliasing of the first source.

To illustrate the SRP localization algorithm and how aliasing affects the localization, two polar plots are shown in figure 2.6. In plot 2.6a the broadband power $\overline{P}^{\text{b}}(\theta)$ is shown for the spectrum depicted in figure 2.5a. This spectrum is used in 2.20 to estimate the DOA. Here, the DOA of $\varphi = 180°$ is also visible. The aliasing has only a small impact in the broadband power spectrum. The spatial power spectrum $P(\Omega, \theta)$ for different frequencies, like it is used in 2.21 to estimate DOA per frequency, is shown in 2.6b. In that figure, the DOA can be obtained for the lower frequencies ($f = 2\,\text{kHz}$, $f = 4\,\text{kHz}$). However, for $f = 6\,\text{kHz}$ grating lobes on

[3]Note that this is very similar to the well known beampattern. Here (equation 2.23), the beamformer weights are changed as apposed to changing $\underline{A}(\Omega)$ depending on θ.

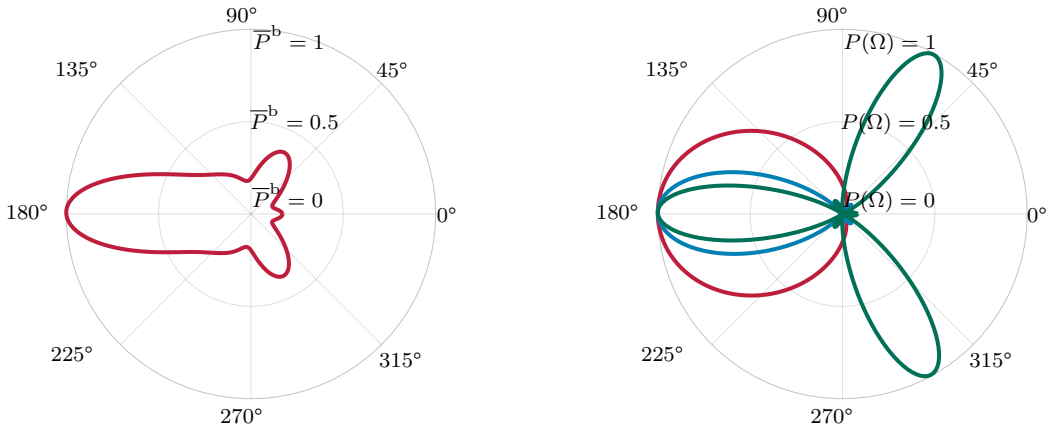


(a) steered response power with phase transform (SRP-PHAT) Delay & Sum

(b) SRP-PHAT Capon

Figure 2.5: Theoretical steered power spectrum of a 7 microphone uniform circular array with center microphone with a 42 mm radius. The direction of arrival ϕ of the acoustic signal is 180°. Grating lobes at 60° and 300° are visible due to spatial aliasing.

the basis of aliasing occur massively, and are due to this ambiguity disqualifying the DOA obtained with this frequency. Furthermore, it is visible that the main lobe becomes narrower in the upper frequencies which in contrary may lead to a more precise DOA estimation.



(a)

(b)

Figure 2.6: Polar plots for the theoretical steered power spectrum from figure 2.5a. In (a) broadband power spectrum $\overline{P}(\theta)$ is plotted and in (b) the spatial power spectrum $P(\Omega, \theta)$ for different frequencies f are plotted. Red: $f = 2\,\mathrm{kHz}$, Blue: $f = 4\,\mathrm{kHz}$, Green: $f = 6\,\mathrm{kHz}$

2.3.3 Practical Realization

In practice some adjustments and assumptions are made to reduce the processing time or to make algorithms more robust. The latter is the reason to adjust the Capon beamformer (Equation 2.18). The main problem is that the covariance matrix may be singular and therefore the inverse of the matrix may not exist. Reason for the singular matrix may be, that fewer sources are dominant as the number of microphones. Thus the spatial covariance matrix has no full rank. The singularity problem can be fixed by a technique called diagonal loading. This is done by adding a small component ϵ to the main diagonal. Now the weights of the modified Capon beamformer reads as

$$\underline{W}'_{\mathrm{CAP}}(\Omega,\theta) = \frac{(\mathbf{\Phi}_{xx}(\Omega)+\epsilon\mathbf{I})^{-1}\underline{D}(\Omega,\theta)}{\underline{D}^H(\Omega,\theta)(\mathbf{\Phi}_{xx}(\Omega)+\epsilon\mathbf{I})^{-1}\underline{D}(\Omega,\theta)}, \tag{2.24}$$

where $\mathbf{I}$ is the unit matrix. However, the impact of the weighting in 2.24 is heavily depended on the absolute values in covariance matrix $\mathbf{\Phi}_{xx}(\Omega)$. To cancel out the dependencies the so called coherence matrix is taken instead:

$$J^{ij}_{xx}(\Omega) = \frac{\phi_{ij}(\Omega)}{\sqrt{\phi_{ii}(\Omega)\phi_{jj}(\Omega)}},\ \forall i,j \in \{1,...,M\}, \tag{2.25}$$

where $J^{ij}_{xx}(\Omega)$ are the entries of the $M \times M$ coherence matrix $\mathbf{J}_{xx}(\Omega)$ and $\phi_{ij}(\Omega)$ are the elements of the covariance matrix $\mathbf{\Phi}_{xx}(\Omega)$. When doing the diagonal loading on the coherence matrix, one wants to preserve the magnitude of one on the main diagonal, so instead of increasing the main diagonal by a small amount ϵ all other elements are decreased:

$$J'^{ij}_{xx}(\Omega) = \begin{cases} \frac{\mathbf{\Phi}^{ij}_{xx}(\Omega)}{\phi_{xx}(\Omega)(1+\epsilon)} & i,j \in \{1...M \mid i \neq j\} \\ \frac{\mathbf{\Phi}^{ij}_{xx}(\Omega)}{\phi_{xx}(\Omega)} & i,j \in \{1...M \mid i = j\} \end{cases}. \tag{2.26}$$

The loaded coherence matrix is then used in 2.15 as $\mathbf{\Phi}_{xx}(\Omega)$ to gain the beamformer weights:

$$\underline{W}''_{\mathrm{CAP}}(\Omega,\theta) = \frac{J'_{xx}(\Omega)^{-1}\underline{D}(\Omega,\theta)}{\underline{D}^H(\Omega,\theta)J'_{xx}(\Omega)^{-1}\underline{D}(\Omega,\theta)} \tag{2.27}$$

Equation 2.27 may be incorporated in 2.13 to obtain the power

$$P''_{\mathrm{CAP}}(\Omega,\theta) = \frac{\underline{D}^H(\Omega,\theta)\mathbf{J}'^{-1}_{xx}(\Omega)\ \mathbf{\Phi}_{xx}(\Omega)\ \mathbf{J}'^{-1}_{xx}(\Omega)\underline{D}(\Omega,\theta)}{\left(\underline{D}^H(\Omega,\theta)\mathbf{J}'^{-1}_{xx}(\Omega)\underline{D}(\Omega,\theta)\right)^2}. \tag{2.28}$$

This method is also stated in [BW13, Chapter 2].
A further adjustment has to be done because in real-time applications the incoming stream of data requires frame-wise processing. Another reason is the non-stationarity of speech and DOA. This can be omitted when considering only a short segment of microphone signals. As a result the L-point discrete Fourier transform is utilized, with an overlapping window function:

$$X(l,b) = \sum_{n=0}^{L-1} \mathcal{W}(n) x(bO+n) e^{-j2\pi n \frac{l}{L}}. \tag{2.29}$$

The l is the discrete frequency bin, $\mathcal{W}(n)$ is the window function, O represents the frameshift in samples, and b is the frame index. This method is called the short-time Fourier transform (STFT). [Loi13, Chapter 2.5.4]
Based on the STFT, the equations 2.20 and 2.21 changes as follows

$$\begin{aligned} \hat{\varphi}(b) &= \arg\max_{\theta} \sum_{l=0}^{L/2} P(l,b,\theta) \\ &= \arg\max_{\theta} \overline{P}^{\mathrm{b}}(b,\theta), \end{aligned} \tag{2.30}$$

$$\hat{\varphi}(l,b) = \arg\max_{\theta} P(l,b,\theta). \tag{2.31}$$

The DOA estimate $\hat{\varphi}$ now depends on the discrete frequency l and the frame index b. Due to symmetry properties only the frequency bins from 0 to $L/2$ are used. Furthermore, the frame dependent SRP is now called sub-band SRP, because of the sub-band frequency division by STFT.
Furthermore, the effort to calculate the spatial covariance matrix can be reduced by only considering B_E frames which may according to [KV96] be stated as

$$\hat{\mathbf{\Phi}}_{xx}(l,b) = E\{\underline{X}(l,b)\underline{X}(l,b)^*\} = \frac{1}{B_E} \sum_{b=1}^{B_E} \underline{X}(l,b)\underline{X}^H(l,b). \tag{2.32}$$

A further simplification of the covariance matrix estimation is to use a first order infinite impulse response (IIR) filter, which can be noted as

$$\hat{\mathbf{\Phi}}_{xx}(l,b) = \alpha \hat{\mathbf{\Phi}}_{xx}(l,b-1) + (1-\alpha)\underline{X}(l,b)\underline{X}^H(l,b), \tag{2.33}$$

where α is the smoothing constant.

2.4 Source Classification with GMMs

From chapter 2.3 and the equations and 2.31 'raw' localization date $\hat{\varphi}$ are resulting. Based on this data a robust localization for multiple sound sources shall be obtained. To perform multi-source localization, these raw results $\hat{\varphi}$ have to be classified. In other words, the parameters of the underlying model that represent the observations shall be estimated. This can be done by many different algorithms like k-nearest neighbor (kNN), artificial neural networks (ANN) or support vector machines (SVM). Another often used approach is a GMM. This method forms a statistical model with the use of a mixture of Gaussian PDFs. GMMs have an advantage against kNN because they do not need to store a large set of data, but only the statistical model. [MHA08, Chapter 17.2.3]

2.4.1 Gaussian Mixture Models

In the following, the basics of modeling data with a GMM are explained. The observations are the estimated DOAs $\hat{\varphi}$ in the azimuth plane. It is assumed that these observations have an underlying stochastic process, which shall be modeled by a set of one dimensional Gaussians. A single Gaussian PDF can be written in this form

$$\mathcal{N}(\hat{\varphi}|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\hat{\varphi}-\mu}{\sigma})^2}, \tag{2.34}$$

where σ is the standard deviation and μ the mean of the Gaussian. To get a GMM one needs to form a linear superposition of K Gaussians stated as

$$p_{\mathrm{GMM}}(\hat{\varphi}|\underline{\mu},\underline{\sigma},\underline{\pi}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\hat{\varphi}|\mu_k,\sigma_k). \tag{2.35}$$

where $\underline{\mu} = \{\mu_1, \mu_k, ..., \mu_K\}$, $\underline{\sigma} = \{\sigma_1, \sigma_k, ..., \sigma_K\}$ and $\underline{\pi} = \{\pi_1, \pi_k, ..., \pi_K\}$ are sets of means, standard deviations and mixing coefficients for every class k. The mixing coefficients are used to normalize the GMM PDF. One of the main properties of a PDF is that the integral over its variable has to be one $\int_{\hat{\varphi}} p(\hat{\varphi})\mathrm{d}\varphi = 1$. Therefore, the sum over all single mixing coefficients π_k must also be one and in the range between zero and one for any k, which can be stated as

$$\sum_{k=1}^{K} \pi_k = 1, \qquad 0 \leqslant \pi_k \leqslant 1. \tag{2.36}$$

2.4.2 Parameter Estimation

At first hand, the GMM is not a classification algorithm, but rather a method to model data. If it is known to which class a given observation belongs to (the representations of every observation is known), one can separate this superposition of Gaussians into an estimation of k single Gaussians. This can be done with the help of the ML estimators,

$$\begin{aligned} \mu_k &= \frac{\sum_{n=1}^{N_k} \hat{\varphi}_n}{N_k} \\ \sigma_k &= \frac{1}{N_k} \sum_{n=1}^{N_k} (\hat{\varphi}_n - \mu_k)^2, \end{aligned} \tag{2.37}$$

where the data set consists of $\underline{\hat{\varphi}} = \{\hat{\varphi}_1, ..., \hat{\varphi}_{N_k}\}$ observations, N_k is the number of data which are representatives of class k. However, if the information about a class representation is not observable, the parameter cannot be estimated by a closed form solution. Unknown representations are also called latent variables. This problem can be stated as the joint probability of 2.35 over all N observations $\underline{\hat{\varphi}} = \{\hat{\varphi}_1, \hat{\varphi}_n, ..., \hat{\varphi}_N\}$

$$p_{\mathrm{GMM}}(\underline{\hat{\varphi}}|\underline{\mu}, \underline{\sigma}, \underline{\pi}) = \prod_{n=1}^{N} \left[\sum_{k=1}^{K} \pi_k \mathcal{N}(\hat{\varphi}_n|\mu_k, \sigma_k) \right] \rightarrow \max, \tag{2.38}$$

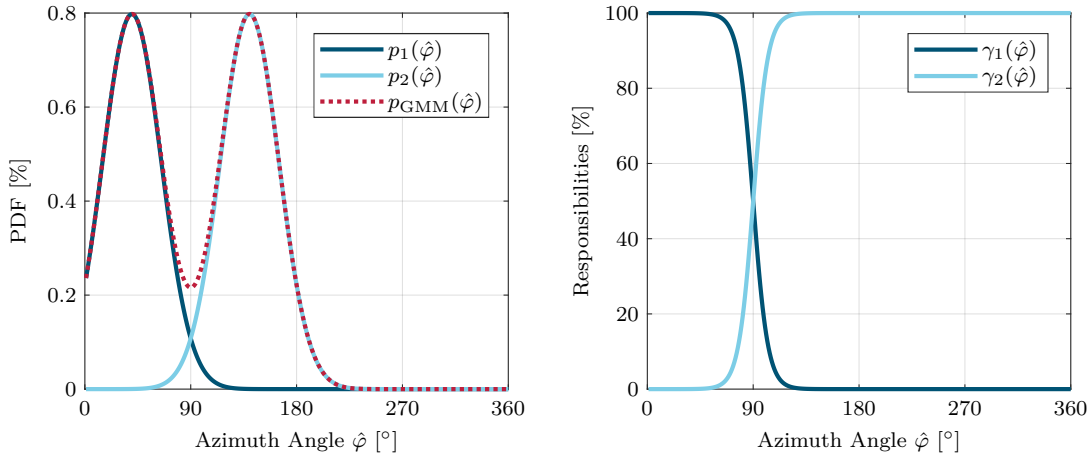
where $p_{\mathrm{GMM}}(\underline{\hat{\varphi}}|\underline{\pi}, \underline{\mu}, \underline{\sigma})$ is the joint likelihood how probable the observations $\underline{\hat{\varphi}}$ are given the model consisting of $\underline{\pi}, \underline{\mu}$ and $\underline{\sigma}$. This likelihood shall be maximized, which is also known as ML estimation. Alternatively, the logarithmic probability can be maximized

$$\ln p_{\mathrm{GMM}}(\underline{\hat{\varphi}}|\underline{\mu}, \underline{\sigma}, \underline{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\hat{\varphi}_n|\mu_k, \sigma_k) \right\} \rightarrow \max . \tag{2.39}$$

Finding the ML solution for a GMM with latent variables can be accomplished using the EM-algorithm, which is a recursive process, as illustrated in figure 2.8. First, the parameter set has to be initialized as seen in figure 2.8.a. This can be done randomly or with an educated guess. Afterwards, the recursive process starts with the expectation step, where the latent variables for each observation $\hat{\varphi}_n$ are estimated (figure 2.8.a). They are called responsibilities γ_{kn} and may be expressed as

$$\gamma_{kn} = \frac{\pi_k \mathcal{N}(\hat{\varphi}_n|\mu_k, \sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\hat{\varphi}_n|\mu_{k'}, \sigma_{k'})} = \frac{p_k(\hat{\varphi}_n)}{p_{\mathrm{GMM}}(\hat{\varphi}_n|\underline{\mu}, \underline{\sigma}, \underline{\pi})}, \tag{2.40}$$

where p_k represents a single Gaussian PDF of class k. The calculation of these responsibilities can also be comprehended in figure 2.7. Here you can see that the total probability $p_{GMM}(\hat{\varphi})$ (red) will be set into relation to the single probabilities $p_1(\hat{\varphi})$ or $p_2(\hat{\varphi})$ (blue or teal) to result in the responsibilities γ_1 or γ_2. In the overview figure 2.8 the estimation of responsibilities in (b) is shown as a coloration of the observations in class colors.



(a) (b)

Figure 2.7: In plot (a) the probability density function for two classes and the total Gaussian mixture model distribution is shown. In plot (b) the resulting responsibilities are shown, which are calculated by dividing a single probability density function by the the total distribution.

Next, in the maximization step, the parameters of the GMM are estimated (figure 2.7.c). The ML estimation of the mean μ_k can be written as

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_{kn} \hat{\varphi}_n}{N_k}, \tag{2.41}$$

where N_k is the number of observations associated with class k

$$N_k = \sum_{n=1}^{N} \gamma_{nk}. \tag{2.42}$$

It can be noted that this estimation for every class k is a weighted mean of all N observations in the set. The standard deviation can be written as

$$\sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{kn} (\hat{\varphi}_n - \mu_k)^2. \tag{2.43}$$

Last, the mixing coefficients are stated as

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma_{kn}}{N}. \tag{2.44}$$

After the maximization step is done, the recursive algorithm has to evaluate if it has to do another iteration I on the set $\underline{\hat{\varphi}}$ of observations or to terminate the algorithm. This may be done by evaluation if the log likelihood in 2.39 is converging. If the convergence criterion is not satisfied, the algorithm shall be starting again at the expectation step. A full derivation of the EM-algorithm can be found in [Bis06, Chapter 9.2].

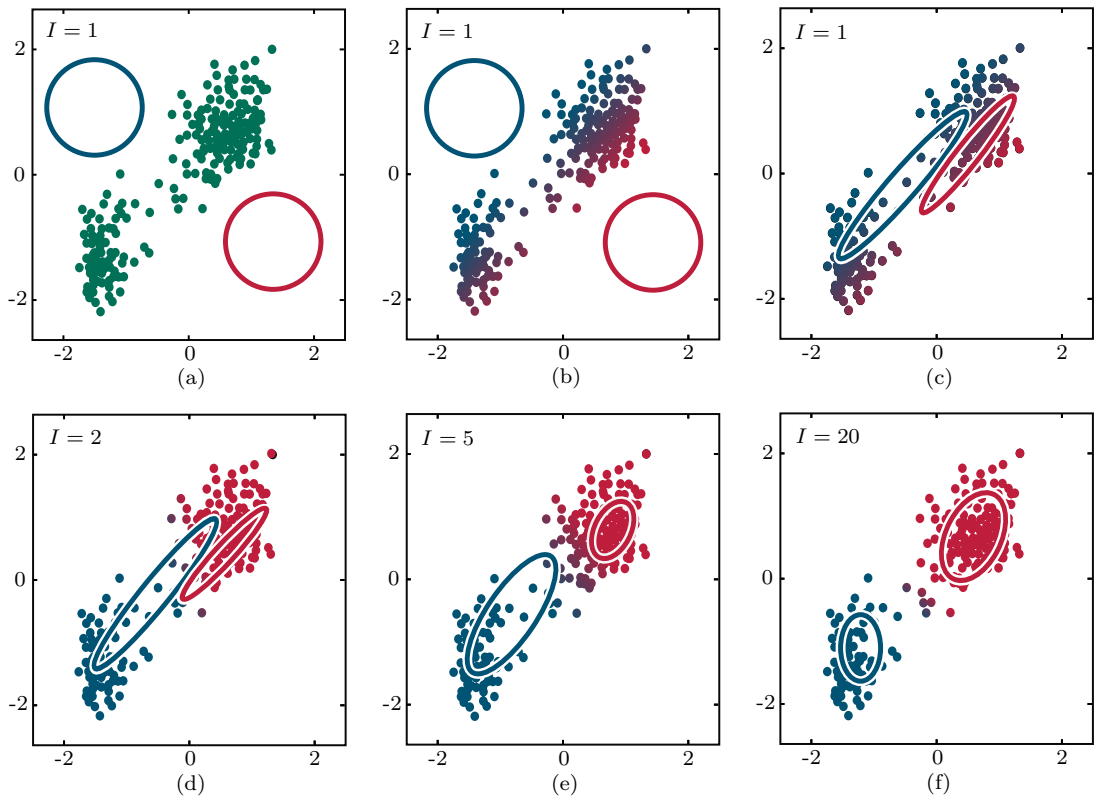


Figure 2.8: Parameter estimation of a two dimensional Gaussian mixture model using an expectation-maximization algorithm according to [Bis06, Chapter 9.2.2]. After the initialization (a), the responsibilities to the classes for the observations are calculated in the E-step (b). Based on the responsibilities, the Gaussian mixture model parameter are recalculated in the M-step (c). Step (b) and (c) are repeated in every iteration I, till the log likelihood in 2.39 is converging (f).

2.4.3 Use of Wrapped Gaussian Mixture Model for Periodic Observations

Since this work is restricted to the use of circular arrays (stated in chapter 1), the GMM has to deal with cyclic observations. For instance, there are two observations given at $\hat{\varphi}_1 = 359°$ and $\hat{\varphi}_2 = 1°$ and they should be modeled as one Gaussian distribution. With the ML estimator for standard deviation 2.43 this gives different results of the underlying Gaussian distribution than the observations on the other side of the circle ($\hat{\varphi}_3 = 179°$ and $\hat{\varphi}_4 = 181°$). This inconsistency problem has to be handled with a new approach. One approach could be the von-Mises-distribution, which can be derived by taking a two-dimensional Gaussian and map it onto the unit circle. Von Mises distribution has the drawback that it makes use of the Bessel function which can be quite complex to handle when deriving the ML-estimators. Another approach is the wrapped Gaussian, which is defined as a 2π periodical recurrent Gaussian.

$$\mathcal{N}_{\mathrm{w}}(\hat{\varphi}|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{i=-\infty}^{\infty} e^{-\frac{1}{2}(\frac{\hat{\varphi}-\mu+2\pi i}{\sigma})^2},\ \hat{\varphi} \in [0,..,2\pi). \tag{2.45}$$

This corresponds to 'wrapping' the real axis around the unit circle. Finally, incorporating 2.45 in 2.35 results in an WGMM. [Bis06, Chapter 2.3.8]

2.5 State of the Art

Multi-source localization is an active field of research. It can be separated in approaches which use distributed microphone networks [BOS08; SH05; HQF11] or approaches that use compact arrays. This work focuses on the compact arrays that can be integrated in smart speakers. In this section, first the general state is given for localization with compact arrays. Then an algorithm which utilizes also a GMM from Nilesh Madhu is described in more detail. This algorithm will be used later as an external reference algorithm.

2.5.1 General State

In most papers a general strategy is to utilize first a core localizing technique to get raw localization results. Afterwards the data are used by a post-processing algorithm to separate the results into classes. The post-processing may also be a tracking algorithm integrated like a Kalman-Filter or a particle filter. In figure 2.9 this approach is depicted.

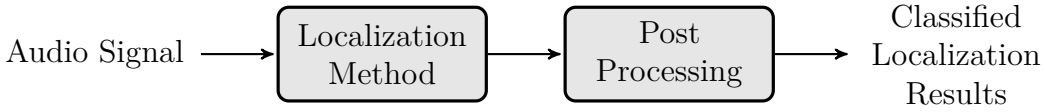


Figure 2.9: Block diagram for the general principal of multi-source localization.

An interesting real-time approach for circular arrays was made by Pavlidi et al.. They transform the signal with a Fourier transform to have the time-frequency representation. Here they are searching for *single source* zones, where only one source is active. Therefore, they can use a single source DOA algorithm over these zones by searching maximums over the so called circular integrated cross spectrum (CISC) which is quite similar to the SRP method. From this they generate a smoothed histogram of DOA estimations. Finally they employ a method, called *matching pursuit*, to estimate the number of active sources and the corresponding DOAs. [Pav+13]
Another method by Jean-Mark Valin et al. was created for detecting simultaneously moving sound sources for a mobile robot. The array of 8 microphones is structured like a cube with one microphone on each corner. The strategy is also close to the general approach stated before. They use SRP with a D&S beamformer and a spherical search grid to obtain a steered power spectrum. Afterwards they conduct a particle filter-based post-processing which yields source positions. [VMR07]
Nikunen at al. built a source separation which includes a multi-source localization algorithm. They use also SRP as a core localizer and then classify the broadband steered power spectrum with an EM-Algorithm and a WGMM. Afterwards they conduct a particle filter as a source tracking on the results of the EM-Algorithm.[NDV18]
The research of Evers et al. concentrate on the post-processing with the Gaussian mixture probability hypothesis density (GM-PHD) filters. This filter was developed for multi target tracking in presence of clutter and missing detections. [Eve+15]

2.5.2 Madhu's Multi-source Localization Algorithm

Nilesh Madhu's multi-source localization algorithm is based as some others before on the SRP with a D&S beamformer. He uses a ULA and therefore can only detect values between 0° and 180° due to the symmetry of the array. It uses equation 2.31 to obtain the localization results and buffers them over frequency l in a frame b which can be written as

$$\underline{\hat{\varphi}}(b) = \Big(\hat{\varphi}(l_{\text{low}}, b), ..., \hat{\varphi}(L/2, b)\Big)^T, \tag{2.46}$$

where l_{low} is a lower frequency bound. This excludes bins that are below this bound because low frequencies do not yield good localization estimates. Upon these data

a GMM is trained with an EM-algorithm. This is done on a frame basis, so the index b will be dropped till it is needed again. The general approach is to start the EM-algorithm with an over-estimation for sources K_{init}. So the starting order of the system is $K \leftarrow K_{\text{init}}$. When this EM is applied to the observations $\underline{\hat{\varphi}}(b)$ a fitted GMM with means $\underline{\mu}$, variances $\underline{\sigma}^2$ and mixing coefficients $\underline{\pi}$ is obtained. Because of the overestimation the underlying process may be overdetermined and a *shrinking* process is introduced, which reduces the model. This is done by limiting the distance of the mean estimated $\Gamma_{\text{mean}}^{\text{dist}}$. When this threshold is hit by two classes the classes are shrunken to one class by the following method:
if $\exists i, i'$, such that $|\mu_i - \mu_{i'}| \leq \Gamma_{\text{mean}}^{\text{dist}}$

$$\begin{aligned} \mu_i &\leftarrow \frac{\pi_i \mu_i + \pi_{i'} \mu_{i'}}{\pi_i + \pi_{i'}} \\ \sigma_i^2 &\leftarrow \frac{\pi_i \sigma_i^2 + \pi_{i'} \sigma_{i'}^2}{\pi_i + \pi_{i'}} \\ \pi_i &\leftarrow \pi_i + \pi_{i'} \\ K &\leftarrow K - 1. \end{aligned} \tag{2.47}$$

After the shrinking, the EM-algorithm is used again with the GMM as initial values. This process is repeated till the separation of all mean values are at least $\Gamma_{\text{mean}}^{\text{dist}}$. The number of classes K indicated the estimated number of sources. In addition to the GMM a noise class is added. This is necessary due to random distributed observations in bins where no source is active. The floor class is modeled as a *hidden component* $(K+1)$ with a constant mean $\mu_{K+1} = 90°$ and a large standard deviation. For the hidden component, only the mixing coefficients and variance is adapted during the EM-algorithm. The tracking is done with the help of an overlying GMM $(\overline{\underline{\mu}}, \overline{\underline{\sigma^2}}, \overline{\underline{\pi}})$. The overlying GMM has the purpose to handle the changing number of sources in the acoustic scene. Sometimes sources do speech pauses, new sources arise, while others disappear and some are moving. This is not depicted in looking at single frames and therefore the overlying GMM is used. To handle the 'death' of sources the time-to-live (TTL) $\overline{b}_{\text{TTL},k}$ is introduced. The update of the overlying GMM for any time frame b can be stated as the following:
If $\exists\ i, \bar{i}$, at frame b such that $|\mu_i - \mu_{\bar{i}}| \leq \Gamma_{\text{mean}}^{\text{dist}}$

$$\begin{aligned} \overline{\mu}_{\bar{i}} &\leftarrow \frac{\overline{\pi}_{\bar{i}} \overline{\mu}_{\bar{i}} + \pi_i(b) \mu_i(b)}{\overline{\pi}_{\bar{i}} + \pi_i(b)} \\ \overline{\sigma}_{\bar{i}}^2 &\leftarrow \frac{\overline{\pi}_{\bar{i}} \overline{\sigma}_{\bar{i}}^2 + \pi_i(b) \sigma_i^2(b)}{\overline{\pi}_{\bar{i}} + \pi_i(b)} \\ \overline{\pi}_i &\leftarrow \overline{\pi}_i + \pi_i(b) \\ \overline{b}_{\text{TTL},\bar{i}} &\leftarrow \min(\overline{b}_{\text{TTL},\bar{i}}(b) + 1, \Gamma_{\text{TTL}}^{\max}) \end{aligned} \tag{2.48}$$

and for each i in frame b such that $\nexists \bar{i}$, with $|\mu_i - \mu_{\bar{i}}| \leq \Gamma_{\text{mean}}^{\text{dist}}$

$$\begin{aligned} \overline{\mu}_{\bar{i}} &\leftarrow \overline{\mu}_{\bar{i}} \ \cup \ \mu_i(b) \\ \overline{\sigma}^2_{\bar{i}} &\leftarrow \overline{\sigma}^2_{\bar{i}} \ \cup \ \sigma^2_i(b) \\ \overline{\pi}_i &\leftarrow \overline{\pi}_i \ \cup \ \pi_i(b) \\ \overline{b}_{\mathrm{TTL},\bar{i}} &\leftarrow b^{\mathrm{init}}_{\mathrm{TTL}} \\ \overline{K} &\leftarrow \overline{K} + 1 \end{aligned} \tag{2.49}$$

and for each $\bar{i}$ such that $\nexists i$ in frame b, with $|\mu_i - \mu_{\bar{i}}| \leq \Gamma^{\mathrm{dist}}_{\mathrm{mean}}$

$$\overline{b}_{\mathrm{TTL},\bar{i}} \leftarrow \overline{b}_{\mathrm{TTL},\bar{i}}(b) - 1. \tag{2.50}$$

The equations 2.48 state how the overlying GMM class is updated by the current trained GMM class, if their distances to each others are below the threshold $\Gamma^{\mathrm{dist}}_{\mathrm{mean}}$. If no overlying GMM class is in distance to the current trained GMM class then a new class is opened, which is stated in equation 2.49. Equation 2.50 describes the decrease of the TTL in the overlying GMM, if no class of the current GMM is present for updating. If $\overline{b}_{\mathrm{TTL},\bar{i}} \leq 0$ the source is considered as *died* and therefore will be removed from the system. The maximum TTL threshold $\Gamma^{\mathrm{max}}_{\mathrm{TTL}}$ is introduced to limit the source lifetime and b_{TTL}init is considered as initial value for the TTL. $\overline{K}$ is considered as the number of classes of the overlying GMM. After this update the overlying GMM renormalizes the mixing coefficient to guarantee the constraint 2.36. [MM08] [Mad10, Chapter 4.3]

3 Adaptive EM-based Multi-source Localization Algorithm using a Wrapped Gaussian Mixture Model

In this chapter, the developed multi-source localization method is described. The core of this method is a GMM which is estimated by an EM-algorithm. The main challenges are the addition and deletion of new sources q where each is represented by one Gaussian. Furthermore, the change from processing on a block of static data to the processing of varying real-time data has to be tackled, which is not covered by a classical EM-algorithm.

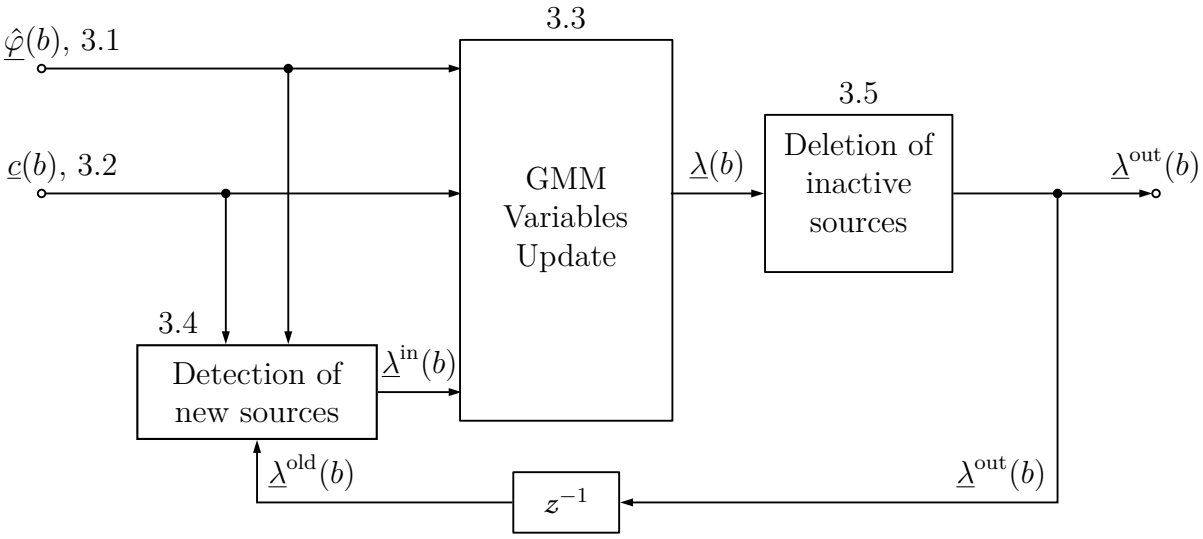


Figure 3.1: Block diagram as overview of developed multi-source localization algorithm

In figure 3.1 the overview flow chart of the algorithm is shown. The numbers above of the different parts of the algorithm represent the subsections of this chapter where they are discussed in more detail. At first the incoming data are considered, which are results of different types of localization, in detail addressed in the following chapter 2.3. The values $\underline{\hat{\varphi}}(b)$ are buffered to get a set of observations for the EM-algorithm.
Besides the DOA result, the so-called confidence $\underline{c}(b)$ is introduced, as a measure

of likelihood that the DOA is correct. Then follows the derivation and description how the GMM variables are updated by the EM-algorithm. For each observation set, only one EM-Iteration is done, due to the assumption, that incoming data are not changing rapidly, and the EM can adapt in every time step. The adopted GMM variables from the step before will be reused in the current step. This is a big difference to the Madhu approach (chapter 2.5.2) where in every iteration a full EM-Algorithm has to be done. An important fact is that the algorithm is initialized with zero classes, so the first action of the algorithm is to decide if a new class shall be opened, based on the incoming observations. This is done by the 'detection of new sources' block. As last element of the developed multi-source localization method the deletion of sources are contemplated, which is done with the help of the TTL. The output of the multi-source algorithm is given by the full GMM $\underline{\lambda}^{\text{out}}$, which represents the current scene as a model. After the complete discussion of the basic algorithm, the last section will introduce some further more heuristic adjustments and improvements.

3.1 Localization

The DOA as input data $\hat{\varphi}(b)$ for the multi-source localization are calculated with broadband or sub-band SRP method as stated in equation 2.30 or 2.31. In this work the D&S beamformer and the Capon beamformer with diagonal loading stated in 2.17 and 2.28, respectively, are used in the SRP method to calculate the spatial power spectrum $P(l, b, \theta)$. The covariance matrix $\mathbf{\Phi}_{xx}(l, b)$ will be estimated by a first order IIR filter noted in 2.33, which is assumed to be sufficient for estimating the covariance matrix $\mathbf{\Phi}_{xx}(l, b) \approx \hat{\mathbf{\Phi}}_{xx}(l, b)$.
When using the broadband SRP, the DOAs are buffered over time frames b. This buffer can be seen as a sliding window. In this fixed-size buffer of length N the oldest frame will be erased by the newest frame. The buffer can be stated as:

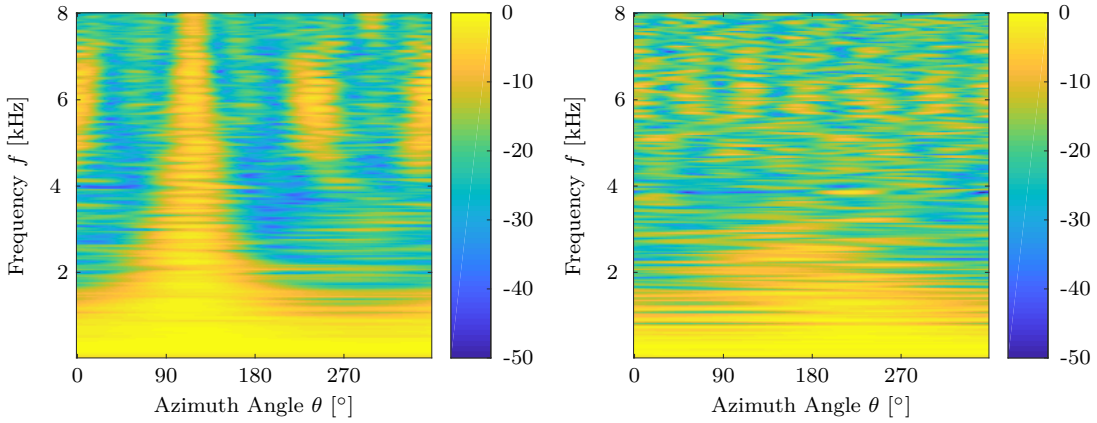
$$\underline{\hat{\varphi}}^{\text{b}}(b) = \Big(\hat{\varphi}(b-N+1),\ \hat{\varphi}(b-N+2),\ \cdots,\ \hat{\varphi}(b)\Big)^T. \tag{3.1}$$

When using the frequency SRP method, the sliding window is not buffered over time anymore rather than over frequencies. Therefore, only data from one frame at a time are in the buffer:

$$\underline{\hat{\varphi}}^{\text{s}}(b) = \Big(\hat{\varphi}(0, b),\ \hat{\varphi}(l, b),\ \cdots,\ \hat{\varphi}(L/2, b)\Big)^T. \tag{3.2}$$

3.2 Confidence

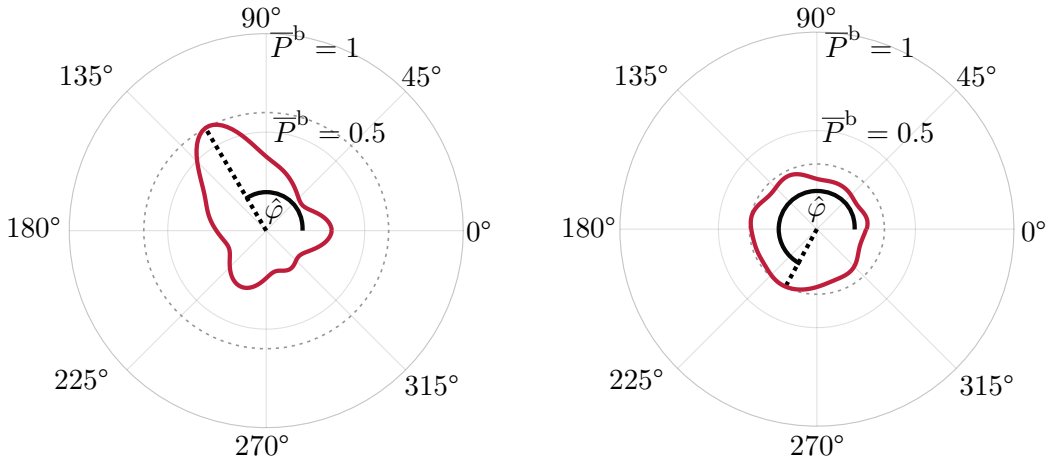
A unique feature of the developed algorithm is the use of confidence values as a second input value. This is done because of the nature of the arg max function. For this function, it does not matter how high the peak is in absolute numbers or in comparison to the other values. This leads to a loss of information. To illustrate the problem, two examples are depicted in figure 3.2. When looking at subplot 3.2a the theoretical steered power spectrum for the SRP-PHAT method with a D&S beamformer shown in 2.5a is visible. Furthermore, in the polar plot 3.3a of the mean power, a maximum for an azimuth angle of approximately $\hat{\varphi} = 120°$ is visible.



(a) Theoretical steered power spectrum visible (b) Theoretical steered power spectrum distorted

Figure 3.2: Comparison of two different spatial power spectra calculated with Steered response power and a delay & sum beamformer for real data recorded with a 7 microphone uniform circular array with center microphone

On the other hand, when looking at plot 3.2b the theoretical steered power spectrum totally vanishes. The polar plot in 3.3b is stating the same: no distinct maximum. But in the nature of the arg max function, a value has to be picked as DOA and here the value would be approximately 240°. When feeding both values in our multi-source localization algorithm without the use of confidence both values would have the same impact on the results, and the algorithm would become degraded. To tackle this problem the confidence value is introduced, which can be seen as a weighting factor for each DOA value. To calculate the confidence value, the ratio between the maximum and the minimum value is taken into account. Therefore the Wiener filter has been used as a role model. The Wiener filter is normally installed for optimal noise reduction and accounts for the power of the distorted signal in relation to the noise [Loi13, Chapter 6.5]. In practice, the noise is estimated by the minimum of the signal. On the other hand, the power of the total signal is estimated by the maximum. Figure 3.4 is depicting this min and max value.



(a) Sufficient maximum height for maximum search

(b) Insufficient maximum height for maximum search

Figure 3.3: Polar plots corresponding to figure 3.2. $\hat{\varphi}$ is the direction of arrival estimation with broadband steered response power

However, the noise is underestimated, because the absolute minimum value is taken. To compensate for this μ_{v} is introduced, which results in

$$c'(l,b) = 1 - \mu_{\mathrm{v}} \frac{\min_\theta P(l,b,\theta)}{\max_\theta P(l,b,\theta)}. \tag{3.3}$$

In this work μ_{v} is called the confidence sensitivity, because it scales the range of confidence values'. To get another degree of freedom, the one is also substituted by a tuning parameter, which is called offset α_{o}. To let this confidence stay between zero and one also when this added parameter is used, an upper and lower limit is introduced. On the basis of 3.3 and the mean power $\overline{P}^{\mathrm{b}}(b,\theta)$ in 2.30 following is obtained

$$c(b) = \begin{cases} 0 & \text{for } c'(l,b) < 1 \\ c'(b) & \text{for } 0 < c'(l,b) < 1 \\ 1 & \text{for } 1 < c'(l,b) \end{cases} \tag{3.4}$$

$$\text{with} \quad c'(b) = \alpha_{\mathrm{o}} - \mu_{\mathrm{v}} \frac{\min_\theta \overline{P}^{\mathrm{b}}(b,\theta)}{\max_\theta \overline{P}^{\mathrm{b}}(b,\theta)}.$$

To calculate the confidences for the sub-band processing, the spatial power spectrum

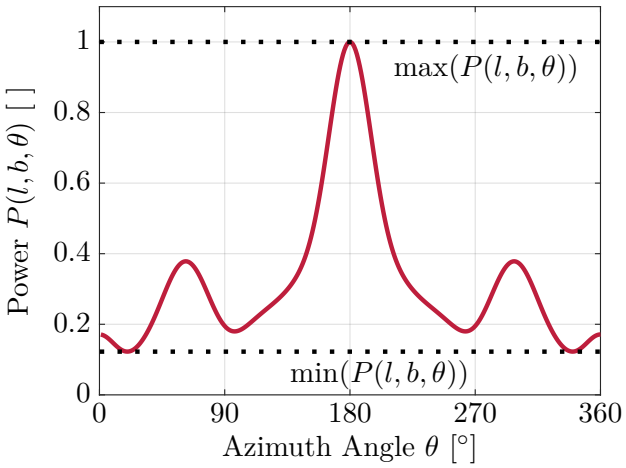


Figure 3.4: Power distribution over azimuth angle. Minimum and maximum values are used for estimating signal and noise power in the confidence calculation.

$P(l,b,\theta)$ is used:

$$c(l,b) = \begin{cases} 0 & \text{for } c'(l,b) < 1 \\ c'(l,b) & \text{for } 0 < c'(l,b) < 1 \\ 1 & \text{for } 1 < c'(l,b) \end{cases} \tag{3.5}$$

$$\text{with} \quad c'(l,b) = \alpha_{\mathrm{o}} - \mu_{\mathrm{v}} \frac{\min_\theta P(l,b,\theta)}{\max_\theta P(l,b,\theta)}.$$

To use these values in the algorithm besides the localization results, a sliding window buffer is also introduced for the confidence values:

$$\underline{c}^{\mathrm{b}}(b) = [\ c(b-N+1),\ c(b-N+2),\ \cdots,\ c(b)\ ]. \tag{3.6}$$

For the sub-band SRP the buffer is again only used with values of one frame b:

$$\underline{c}^{\mathrm{s}}(b) = [\ c(0,b),\ c(l,b),\ \cdots,\ c(L/2,b)\ ]. \tag{3.7}$$

3.3 Update of GMM Parameters

The heart of the multi-source localization algorithm is the estimation of the GMM variables. The algorithm takes the buffered observations from equation 3.1 or 3.2, the confidence value from equation 3.6 or 3.7 and the variables of the GMM from the time step before (see figure 3.1) as input.

For the input data, the origin of the incoming data (sub-band/broadband) is neglected because this has no influence on the further processing. The observations are stated as $\underline{\hat{\varphi}}(b) = \Big(\hat{\varphi}_1(b), ...\hat{\varphi}_N(b)\Big)$ and confidences as $\underline{c}(b) = \Big(c_1(b), ...c_N(b)\Big)$, where N is the number of elements in the buffer. Furthermore, the GMM variables are summarized in one variable $\underline{\lambda}(b) = \Big(\underline{\mu}(b), \underline{\pi}(b), \underline{b}_{\mathrm{TTL}}(b)\Big)$. The added variable b_{TTL} is needed for the deletion of inactive classes and will be explained in chapter 3.5 in full detail.
The variance is set to a fixed parameter $\sigma^2_{\mathrm{const}}$ in this work because this gives an increase in robustness and therefore is neglected in this variable set. Because the input values are now time (frame) dependent, the GMM parameter are so too. However, in the following the update process is explained for one step only and for convenience the frame index b is neglected. During the algorithm, the GMM is in different states and cannot directly be taken as the input for the next EM iteration step as in the normal EM-algorithm. Therefore a few modified $\underline{\lambda}$ are introduced. $\underline{\lambda}^{\mathrm{out}}$ is the GMM after one full processing step is done and can be seen as the output of the algorithm. Furthermore the GMM of the previous frame is needed and will be written as $\underline{\lambda}^{\mathrm{old}} = \underline{\lambda}^{\mathrm{out}}(b-1)$. The $\underline{\lambda}^{\mathrm{in}}$ comes from the hypothesis test and is used as the input value for the EM. It is still the GMM parameter set from the time frame before. The unmodified $\underline{\lambda}$ is the direct output of the EM iteration step. For illustration all different GMM parameter states are depicted in 3.1.
In the first section of this chapter, the M-Step and the E-Step will be derived for wrapped Gaussians in general. Then the EM iteration step will be updated to use MAP adaption rather than ML estimation, to handle the varying real-time data. Next, the confidences are integrated, and a floor class is introduced, which lays the basis for the hypothesis test. Finally, all steps are wrapped up and integrated into the developed algorithm.

3.3.1 Derivation of the EM-Algorithm for Wrapped Gaussians

At first, a simplified version of the wrapped Gaussian distribution described in 2.45 is introduced. This can be done under the constraint that $\sigma_k \ll 2\pi$. Therefore Gaussians that are further away have no significant contribution to the PDF, which is stated now as

$$\mathcal{N}_{\mathrm{w}}(\hat{\varphi}_n|\mu_k, \sigma_k) \approx \frac{1}{\sigma_k\sqrt{2\pi}} \sum_{i=-1}^{1} e^{-\frac{1}{2}(\frac{\hat{\varphi}_n - \mu_k + 2\pi i}{\sigma_k})^2} \tag{3.8}$$
$$\text{with } \mathcal{N}_{\mathrm{w}}(\hat{\varphi}_n|\mu_k, \sigma_k) \in [\ 0,\ 2\pi\ ),$$

where index i is now an element of $[-1,\ 0,\ 1]$ and $\mathcal{N}_{\mathrm{w}}(\hat{\varphi}|\mu, \sigma)$ limited to values between 0 and 2π. With equation 2.35 as basis, for the WGMM follows

$$p_{\mathrm{WGMM}}(\hat{\varphi}_n|\underline{\mu},\underline{\sigma},\underline{\pi}) = \sum_{k=1}^{K} \underbrace{\pi_k \mathcal{N}_{\mathrm{w}}(\hat{\varphi}_n|\mu_k,\sigma_k)}_{p_k(\hat{\varphi}_n)}, \tag{3.9}$$

where the class PDF $p_k(\hat{\varphi})$ is introduced. Because the sum of a wrapped Gaussian has only a few elements, this could also be written out as

$$\begin{aligned} p_k(\hat{\varphi}_n) &= \pi_k\Big(\mathcal{N}(\hat{\varphi}_n|\mu_k - 2\pi,\sigma_k) + \mathcal{N}(\hat{\varphi}_n|\mu_k,\sigma_k) + \mathcal{N}(\hat{\varphi}_n|\mu_k + 2\pi,\sigma_k)\Big) \\ &= p_k^-(\hat{\varphi}_n) + p_k^0(\hat{\varphi}_n) + p_k^+(\hat{\varphi}_n), \end{aligned} \tag{3.10}$$

where $p_k^{(\cdot)}(\hat{\varphi})$ are the PDFs for the corresponding (shifted) Gaussians. For illustration, the elements of one wrapped Gaussian class are shown in figure 3.5 over an extended value range.

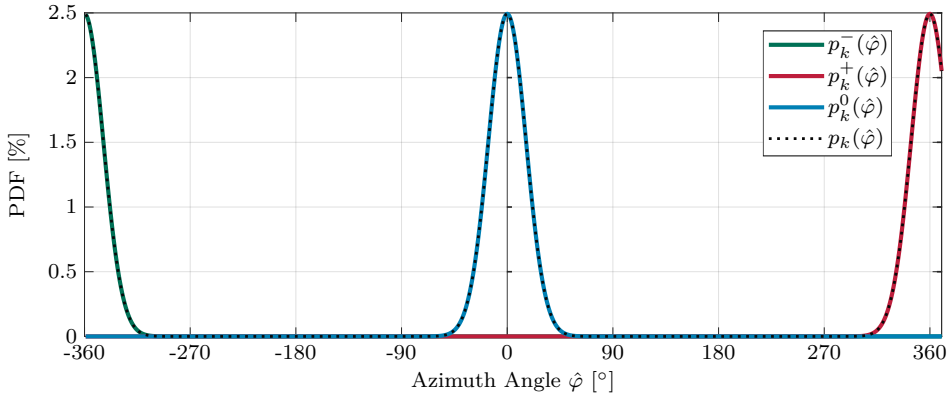


Figure 3.5: The elements of a wrapped Gaussian for a class with $\mu_k = 0$. The black dotted line is the sum of the three others. Only the 'unshifted' and the right-shifted elements have an effect on the value range 0° till 360°.

The estimation of the GMM variables is done similarly to equation 2.39 by maximizing the log-likelihood for all variables. The difference is that a wrapped GMM is used instead. This can be written as:

$$\ln p_{\mathrm{WGMM}}(\underline{\hat{\varphi}}|\underline{\mu},\underline{\sigma},\underline{\pi}) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k \mathcal{N}_{\mathrm{w}}(\hat{\varphi}_n|\mu_k,\sigma_k)\right\} \rightarrow \max. \tag{3.11}$$

The maximization for the mean of the class k is done by taking the partial derivative of equation 3.11 with respect to μ_k and set it to zero.

$$\begin{aligned}\frac{\partial \ln p_{\mathrm{WGMM}}(\underline{\hat{\varphi}}|\underline{\mu},\underline{\sigma},\underline{\pi})}{\partial \mu_k} &= \frac{1}{\sqrt{2\pi\sigma_k^2}} \sum_{n=1}^{N} \frac{\sum_{i=1}^{1} \pi_k \exp(\frac{1}{2\sigma^2}(\hat{\varphi}_n - \mu_k + i2\pi)^2) \frac{\hat{\varphi}_n - \mu_k + i2\pi}{\sigma_k^2}}{p_{\mathrm{WGMM}}(\hat{\varphi}_n|\underline{\mu},\underline{\sigma},\underline{\pi})} \\ &= 0.\end{aligned} \tag{3.12}$$

When solving 3.11 for μ_k and use 3.14 following is obtained

$$\begin{aligned}\mu_k &= \frac{\sum_{n=1}^{N} \gamma_{kn}(\hat{\varphi}_n - \gamma_{kn}^{-}2\pi + \gamma_{kn}^{+}2\pi)}{\sum_{n=1}^{N} \gamma_{kn}} \\ \text{with} \quad \gamma_{kn} &= \frac{p_k(\hat{\varphi}_n)}{p_{\mathrm{WGMM}}(\hat{\varphi}_n)}, \gamma_{kn}^{+} = \frac{p_k^{+}(\hat{\varphi}_n)}{p_k(\hat{\varphi}_n)}, \gamma_{kn}^{-} = \frac{p_k^{-}(\hat{\varphi}_n)}{p_k(\hat{\varphi}_n)},\end{aligned} \tag{3.13}$$

where the inner responsibilities γ_{kn}^{-} and γ_{kn}^{+} are introduced. They represent responsibilities inside a wrapped Gaussian, if an observation $\hat{\varphi}_n$ is assigned to a left shifted Gaussian PDF $p_k^{-}(\hat{\varphi}_n)$ or the right shifted Gaussian PDF $p_k^{+}(\hat{\varphi}_n)$. When an observation is assigned to either of them, the DOA $\hat{\varphi}_n$ is shifted by 2π as stated in equation 3.13. The inner responsibilities are calculated with the responsibilities γ^{kn} in the E-step. It can be seen as a cyclic correction to the center PDF $p_k^0(\hat{\varphi}_n)$. Furthermore, the responsibilities for the E-Step are yielded as byproduct:

$$\gamma_{kn} = \frac{p_k(\hat{\varphi}_n)}{p_{\mathrm{WGMM}}(\hat{\varphi}_n)} = \frac{\pi_k \mathcal{N}_{\mathrm{w}}(\hat{\varphi}_n|\mu_k,\sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}_{\mathrm{w}}(\hat{\varphi}_n|\mu_j,\sigma_j)}. \tag{3.14}$$

When summarizing the cyclic correction with the DOA $\hat{\varphi}$ the following can be written

$$\begin{aligned}\mu_k &= \frac{\sum_{n=1}^{N} \gamma_{kn}\hat{\varphi}_{\mathrm{cor},n}}{\sum_{n=1}^{N} \gamma_{kn}} \\ \text{with} \quad \hat{\varphi}_{\mathrm{cor},n} &= \hat{\varphi}_n - \gamma_{kn}^{-}2\pi + \gamma_{kn}^{+}2\pi,\end{aligned} \tag{3.15}$$

where $\hat{\varphi}_{\mathrm{cor},n}$ is the corrected observation. In this stated the obvious similarity to the mean calculation with a normal GMM (equation 2.41) is evident.
Because in this work the variance of the GMM classes is taken as a constant parameter $\sigma_{\mathrm{const}}^2$, the derivation is neglected. For the mixing coefficients π the calculation is staying the same as stated in equation 2.44.

3.3.2 MAP Adaption for the EM-Algorithm

Incoming data are changing with every frame b. Therefore an adaption is integrated, that also considers past values. For this problem, the MAP adaption can be used. Here the log-likelihood is not maximized anymore (equation 3.11), rather the a posteriori probability is used. With the Bayes' theorem this probability can be stated as:

$$p(\underline{\mu}, \underline{\sigma}, \underline{\pi}|\underline{\hat{\varphi}}) = \frac{p(\underline{\hat{\varphi}}|\underline{\mu}, \underline{\sigma}, \underline{\pi})p(\underline{\mu}, \underline{\sigma}, \underline{\pi})}{p(\underline{\hat{\varphi}})}. \tag{3.16}$$

Out of 3.16 the problem statement is concluded for the (over N observations) joint log a posteriori probability:

$$\ln p(\underline{\mu}, \underline{\sigma}, \underline{\pi}|\underline{\hat{\varphi}}) = \sum_{n=1}^{N} \ln p(\hat{\varphi}_n|\underline{\mu}, \underline{\sigma}, \underline{\pi}) + \ln p(\underline{\mu}, \underline{\sigma}, \underline{\pi}) - \sum_{n=1}^{N} \ln p(\hat{\varphi}_n) \rightarrow \max. \tag{3.17}$$

When maximizing this function with respect to μ_k, by setting the partial derivation of equation 3.17 to zero, following is obtained:

$$\frac{\partial}{\partial \mu_k} \sum_{n=1}^{N} \ln p(\hat{\varphi}_n|\underline{\mu}) + \frac{\partial}{\partial \mu_k} \ln p(\underline{\mu}) = 0. \tag{3.18}$$

For convenience the dependencies $\underline{\sigma}$ and $\underline{\pi}$ in the PDFs are neglected. The probability $p(\hat{\varphi}_n)$ vanishes because it is independent of μ_k. The log-likelihood $p(\hat{\varphi}|\underline{\mu})$ is already known from 2.35. The a priori probability $p(\underline{\mu})$ can be stated in many ways. In this work it is assumed as a k dimensional Gaussian probability density function under the assumption that the means are uncorrelated to each other:

$$\begin{aligned} p(\underline{\mu}) &= \prod_{k=1}^{K} \mathcal{N}(\mu_k| \overset{*}{\mu}_k, \overset{*}{\sigma}_k) \\ &= \prod_{k=1}^{K} p(\mu_k), \end{aligned} \tag{3.19}$$

The mean $\overset{*}{\mu}_k$ and $\overset{*}{\sigma}_k$ are the a priori information. The K PDFs $p(\mu_k)$ are the marginal distributions of $p(\underline{\mu})$. When inserting both equations 2.35, 3.19 in 3.18 one obtains

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \mu_k} \sum_{n=1}^{N} \ln p(\hat{\varphi}_n|\underline{\mu}) + \frac{\partial}{\partial \mu_k} \ln \prod_{k'=1}^{K} p(\mu_{k'}) \\
&= \frac{\partial}{\partial \mu_k} \sum_{n=1}^{N} \ln p(\hat{\varphi}_n|\underline{\mu}) + \frac{\partial}{\partial \mu_k} \sum_{k'=1}^{K} \ln p(\mu_{k'}) \\
&= \sum_{n=1}^{N} \frac{1}{p(\hat{\varphi}_n|\underline{\mu})} \frac{\partial}{\partial \mu_k} p(\hat{\varphi}_n|\underline{\mu}) + \frac{1}{p(\mu_k)} \frac{\partial}{\partial \mu_k} p(\mu_k) \\
&= \sum_{n=1}^{N} \underbrace{\frac{p(\hat{\varphi}_n|\underline{\mu}_k)}{p(\hat{\varphi}_n|\underline{\mu})}}_{\gamma_{kn}} \frac{1}{\sigma_k^2} (\hat{\varphi}_n - \mu_k) - \frac{1}{\overset{*}{\sigma}{}^2} (\mu_k - \overset{*}{\mu}_k).
\end{aligned}
\tag{3.20}
$$

When solving for μ_k, follows

$$
\begin{aligned}
\mu_k &= \frac{\frac{\sigma_k^2}{\overset{*}{\sigma}{}_k^2} \overset{*}{\mu}_k + \sum_{n=1}^{N} \gamma_{kn} \hat{\varphi}_n}{\frac{\sigma_k^2}{\overset{*}{\sigma}{}_k^2} + \sum_{n=1}^{N} \gamma_{kn}} \\
&= \frac{\beta \cdot \overset{*}{\mu}_k + \sum_{n=1}^{N} \gamma_{kn} \hat{\varphi}_n}{\beta + \sum_{n=1}^{N} \gamma_{kn}}.
\end{aligned}
\tag{3.21}
$$

where β is the ratio between variance σ_k^2 and a priori variance $\overset{*}{\sigma}{}_k^2$. In this work β is called the MAP-adaption parameter. The parameter can set how fast the GMM shall adopted to new observation positions. When the MAP-adaption parameter $\beta = 0$ the MAP-adaption reduces to a ML estimation. The mixing coefficient π_k stays the same as stated in 2.44. [GL94]

3.3.3 Integration of Confidences in the Parameter Update

The heuristic integration of confidence was developed over time. In the beginning, a threshold was incorporated to make a hard decision if a localization result should be considered in the EM-algorithm. The results could further be improved when taking the EM-algorithm as a role model and making soft decisions. Therefore the confidences were incorporated in the M-step. They can be seen as another responsibility but global, independent from classes k. When restating the estimation of the mean, following is obtained

$$
\mu_k^{\mathrm{c}} = \frac{\sum_{n=1}^{N} \gamma_{kn} c_n \hat{\varphi}_n}{\sum_{n=1}^{N} \gamma_{kn} c_n}.
\tag{3.22}
$$

For illustration, in case the confidence is low for observation $\hat{\varphi}_n$, the value is not accounted for in the estimation.
The mixing coefficients from equation 2.44 shall also be rewritten as

$$\pi_k^{\mathrm{c}} = \frac{\sum_{n=1}^{N} \gamma_{kn} c_n}{\sum_{n=1}^{N} c_n}. \tag{3.23}$$

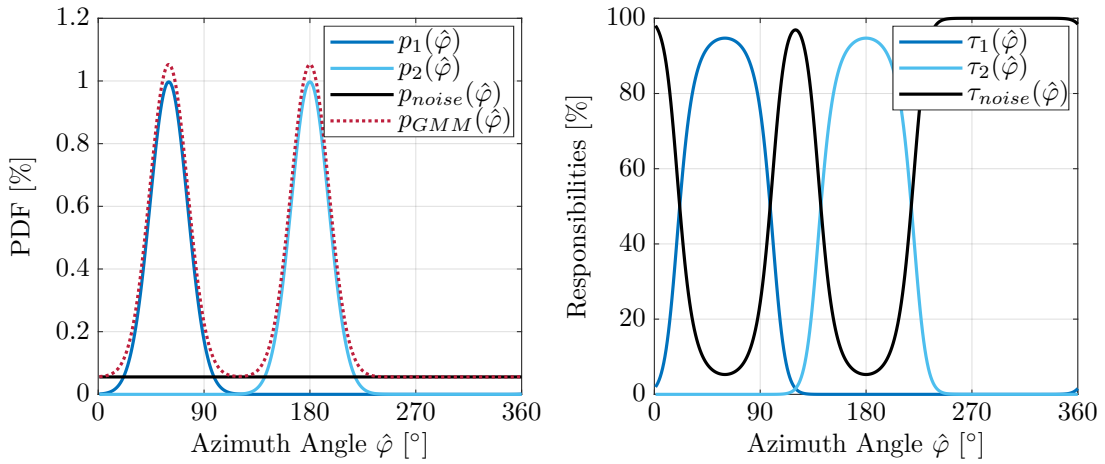
Like in 3.22 the responsibilities for observations with low confidences are not accounted for in the estimation of the mixing coefficients.

3.3.4 Adding a Floor Class to the GMM

The localization results can sometimes be degraded or totally unusable because of the absence of desired signals. This can only be impacting a few bins k in the sub-band processing but also all the frames of localization results in the broadband processing. One measure to prevent this wrong localization results is to incorporate the confidence discussed in chapter 3.2. Another measure is to introduce a floor class to the GMM, which is stated as an uniform distribution. This floor class can also handle another. This floor class can also handle another problem. If new sources arise and are not yet detected by the algorithm, the current classes of the GMM try to integrate these observations in the system. This occurrence is not desirable. Another reason is the spatial aliasing discussed in chapter 2.3.2, where the confidence will give back a high value, but the localization result is not in the direction of the speaker due to the aliasing. With the introduction of the floor class, this incidence can be cushioned by the algorithm. Furthermore the floor class is also used in the new source detection (Chapter 3.4). For illustration how the floor class influences the responsibilities an example GMM is shown in figure 3.6. In 3.6b the responsibilities for $p_1(\hat{\varphi})$ and $p_2(\hat{\varphi})$ are decreasing when observations $\hat{\varphi}$ are far away from the mean of these classes (in, e.g. between 270° to 360°). Therefore they have no impact on the maximization step of the EM-algorithm. Figure 3.6 may be compared to figure 2.7 where this floor is absent, and thus the responsibilities at least of one class are always high. Theoretically, the wrapped GMM with floor class should be stated now. However, in practice this class has only impact on the calculation of the responsibilities. For convenience the floor class is only added in the following equation. It is not necessary to update this class, it shall have a constant parameter p_{floor}:

$$\gamma_{kn} = \frac{p_k(\hat{\varphi}_n)}{p_{\mathrm{WGMM}}(\hat{\varphi}_n|\mu_k, \sigma_k, \pi_k) + p_{\mathrm{floor}}}. \tag{3.24}$$

When adding this class the mixing coefficient π_k of the other classes has to be reduced. However, because the floor class is static and therefore has not to be



(a) (b)

Figure 3.6: With the introduction of the floor class (black), the responsibilities for observations spatially far away from the sources are reduced. Therefore localizations due to aliasing or noise which are far off have less impact in the parameter update (M-step) of the expectation-maximization algorithm

re-estimated, there is no significant impact when the mixing coefficients are not renormalized. This renormalization is neglected in practices.

3.3.5 Integration of the Modified EM-Algorithm

The previous sections are now brought together and are integrated into the the developed multi-source localization algorithm. When looking at 3.1 the inputs are the localization results $\underline{\hat{\varphi}}$, the confidences $\underline{c}$ and the old GMM variables $\underline{\lambda}^{\text{in}} = (\underline{\mu}^{\text{in}}, \underline{\pi}^{\text{in}}, \underline{b}^{\text{in}}_{\text{TTL}})$ coming from the hypothesis test. First, the responsibilities are calculated with the floor class as stated in 3.24 (E-step), which can be written as

$$\begin{aligned}\gamma_{kn} &= \frac{p_k(\hat{\varphi}_n)}{p_{\text{WGMM}}(\hat{\varphi}_n|\lambda^{\text{in}}) + p_{\text{floor}}} \\ &= \frac{\pi_k \mathcal{N}_{\text{w}}(\hat{\varphi}_n|\mu_k^{\text{in}}, \sigma_{\text{const}})}{\sum_{j=1}^{K} \pi_j \mathcal{N}_{\text{w}}(\hat{\varphi}_n|\mu_j^{\text{in}}, \sigma_{\text{const}}) + p_{\text{floor}}}\end{aligned} . \tag{3.25}$$

When all responsibilities for every observation n and class k are calculated, the M-Step follows. For the mean calculation the derivation for wrapped Gaussian (equation 3.13), with MAP adaption (equation 3.21) and confidence integration (equation 3.22) are combined, which can be stated as

$$\begin{aligned}\mu_k &= \frac{\beta \cdot \mu_k^{\text{in}} + \sum_{n=1}^{N} \gamma_{kn} c_n (\hat{\varphi}_n - \gamma_{kn}^{-} 2\pi + \gamma_{kn}^{+} 2\pi)}{\beta + \sum_{n=1}^{N} \gamma_{kn} c_n} \\ &= \frac{\beta \cdot \mu_k^{\text{in}} + \sum_{n=1}^{N} \gamma_{kn} c_n \hat{\varphi}_{\text{cor},n}}{\beta + \sum_{n=1}^{N} \gamma_{kn} c_n} \\ \text{with} \quad \gamma_{kn}^{+} &= \frac{p_k^{+}(\hat{\varphi}_n)}{p_k(\hat{\varphi}_n)} = \frac{\mathcal{N}(\hat{\varphi}_n | \mu_k^{\text{in}} + 2\pi, \sigma_{\text{const}})}{\mathcal{N}_{\text{w}}(\hat{\varphi}_n | \mu_k^{\text{in}}, \sigma_{\text{const}})}, \\ \gamma_{kn}^{-} &= \frac{p_k^{-}(\hat{\varphi}_n)}{p_k(\hat{\varphi}_n)} = \frac{\mathcal{N}(\hat{\varphi}_n | \mu_k^{\text{in}} - 2\pi, \sigma_{\text{const}})}{\mathcal{N}_{\text{w}}(\hat{\varphi}_n | \mu_k^{\text{in}}, \sigma_{\text{const}})}.\end{aligned} \tag{3.26}$$

For the a priori information $\overset{*}{\mu}_k$ and $\overset{*}{\sigma}_k$ in the MAP adaption the values from the frame before are taken, which are represented by σ_k^{in} and μ_k^{in}. The weights π_k are also updated like it is stated in equation 3.23:

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma_{kn} c_n}{\sum_{n=1}^{N} c_n}. \tag{3.27}$$

After this update is done, the GMM parameter set $\underline{\lambda} = (\underline{\mu}, \underline{\pi}, \underline{b}_{\text{TTL}}^{\text{in}})$ is passed to the inactive class deletion.

3.4 Increasing the number of classes

As stated before, the GMM is initialized with zero sources $K = 0$. Therefore before anything happens, a new class must be added to the GMM. This is happening with a so-called *hypothesis test*. As seen in figure 3.1, the hypothesis test regards for the detected observations $\underline{\hat{\varphi}}$ and their confidences $\underline{c}$. On the other hand, it needs the GMM parameter $\underline{\lambda}^{\text{old}}$ of the step before. The hypothesis test is done by calculating the responsibilities of the floor class, introduced in chapter 3.3.4 and stated as

$$\gamma_{\text{floor},n} = \frac{p_{\text{floor}}}{p_{\text{WGMM}}(\hat{\varphi}_n) + p_{\text{floor}}}. \tag{3.28}$$

When summing up $\gamma_{\text{floor},n}$ over observations N the number of observations represented by the floor class in total is gained, which is compared to a threshold. When the sum is greater than this threshold, a new class is added to the GMM. This can be interpreted such, that the current GMM system is not able to represent all observations and therefore many are assigned to the floor class. Mathematically this is stated as

$$\lambda^{\text{in}} = \begin{cases} \underline{\lambda}^{\text{old}} & \text{for } \sum_{n=1}^{N} \gamma_{\text{floor},n} < \Gamma_{\text{class}}^{\text{add}} \\ \underline{\lambda}^{\text{old}} \cup \lambda^{\text{add}} & \text{for } \sum_{n=1}^{N} \gamma_{\text{floor},n} \geq \Gamma_{\text{class}}^{\text{add}} \end{cases}, \tag{3.29}$$

where $\Gamma_{\text{class}}^{\text{add}}$ is the threshold and $\lambda^{\text{add}} = (\mu^{\text{add}}, \pi^{\text{add}}, b_{\text{TTL}}^{\text{add}})$ is the class that is added to the GMM. This class must have initial values for mean μ^{add} and mixing coefficient π^{add}. For the initial mean calculation equation 3.22 with $\gamma_{\text{floor},n}$ as its responsibilities is utilized:

$$\mu^{\text{add}} = \frac{\sum_{n=1}^{N} \gamma_{\text{floor},n} c_n \hat{\varphi}_n}{\sum_{n=1}^{N} \gamma_{\text{floor},n} c_n}. \tag{3.30}$$

The initial values for mixing coefficients π^{add} and the initial TTL $b_{\text{TTL}}^{\text{add}}$ are data-independent and are used as tuning parameter. For the mixing coefficients π^{add} and the initial TTL $b_{\text{TTL}}^{\text{add}}$ the value is set constant. Further information over the TTL are given in chapter 3.5. Due to the new class in the GMM the constraint for the mixing coefficients (equation 2.36) will be violated. This could be healed by re-normalizing all mixing coefficients. However, the violation of this constraint can be neglected, because there is only a small impact for the following EM-algorithm, and after the first EM iteration is done, the mixing coefficients will be naturally normalized so that the constraint is fulfilled again.

3.5 Deleting of Inactive Sources

Now the addition of new sources is handled, the deleting of inactive sources has to be tackled as a separate case. The TTL is used in packet-switched networks, where it prevents packages for an infinitely long time to try to reach the receiving server, which may lead to a blocked channel. It is also used in Madhu's work in a similar way [Mad10, Chapter 4.3]. The deleting of inactive sources collects the GMM variables $\underline{\lambda}$ of the EM update and returns the 'shrunken' GMM variable set $\underline{\lambda}^{\text{out}}$ (see figure 3.1). The TTL is used to erase classes of sources, that are inactive for a certain time. With the TTL the GMM parameter set may be written now $\lambda = [\mu, \pi, b_{\text{TTL}}]$. The function of the TTL is straightforward. The TTL is increasing when its class is excited, which means that observations are represented by this class. The TTL decreases when its class is not excited. The activity of a class is measured in two different ways for broadband and sub-band processing. When using the developed broadband multi-source localization algorithm the current responsibility γ_{kN} is taken to compare with the threshold $\Gamma_{\text{thres}}^{\text{TTL}}$. The TTL update can be stated as,

$$b_{\mathrm{TTL},k}^{\mathrm{out,b}} = \begin{cases} b_{\mathrm{TTL},k} + b_{\mathrm{TTL}}^{\mathrm{inc}} & \text{for } \gamma_{kN} > \Gamma_{\mathrm{thres}}^{\mathrm{TTL}} \\ b_{\mathrm{TTL},k} - 1 & \text{for } \gamma_{kN} < \Gamma_{\mathrm{thres}}^{\mathrm{TTL}} \end{cases}. \tag{3.31}$$

$\Gamma_{\mathrm{thres}}^{\mathrm{TTL}}$ is the threshold parameter and $b_{\mathrm{TTL}}^{\mathrm{inc}}$ is a fixed parameter to adjust, how fast the TTL should increase. When doing sub-band processing, the mixing coefficient π_k can be used because all values are coming from the current frame. The TTL update can be stated as,

$$b_{\mathrm{TTL},k}^{\mathrm{out,s}} = \begin{cases} b_{\mathrm{TTL},k} + b_{\mathrm{TTL}}^{\mathrm{inc}} & \text{for } \pi_k > \Gamma_{\mathrm{thres}}^{\mathrm{TTL}} \\ b_{\mathrm{TTL},k} - 1 & \text{for } \pi_k < \Gamma_{\mathrm{thres}}^{\mathrm{TTL}} \end{cases}, \tag{3.32}$$

After the TTL update the erasing constraint is checked. Now the TTL is considered again as independent from broadband or sub-band processing and states as $b_{\mathrm{TTL},k}^{\mathrm{out}}$. When TTL is becoming smaller than zero, the class is removed from the GMM, which can be written as,

$$\lambda_k^{\mathrm{out}} = \begin{cases} [\,] & \text{for } b_{\mathrm{TTL},k}^{\mathrm{out}} < 0 \\ \lambda_k & \text{for } b_{\mathrm{TTL},k}^{\mathrm{out}} > 0 \end{cases}, \tag{3.33}$$

where the deletion of the class is expressed by replacing the Gaussian class k λ_k by an empty operator $[\,]$.
As a final mechanism of the TTL the $\Gamma_{\mathrm{TTL}}^{\mathrm{max}}$ threshold is introduced, which set a upper limit for TTL. Therefore one class will be always forgetting after a fixed time when this class is not active again.

3.6 Adjustments to the Basic Algorithm

In some scenarios the basic version of the developed multi-source algorithm has some shortcomings which leads to errors in the classifications. In this section the heuristic extensions for the basic algorithm are discussed to improve the classification performance.

3.6.1 Minimum Threshold for Mixing Coefficients

The mixing coefficients π can go to zero when no observations are assigned to the respective class. This would lead to a class that will never get new observations assigned to in the E-Step, which would result in the death of this class over time. However, after a short speech pause it is desirable that the mixing coefficients

rise again. This can be achieved by introducing a lower bound $\Gamma_{\text{mix}}^{\text{min}}$ for all mixing coefficients. When this is reached the constraint 2.36 is violated. However, it is tolerated in practice, because $\Gamma_{\text{mix}}^{\text{min}}$ is set so low that the violation is marginal.

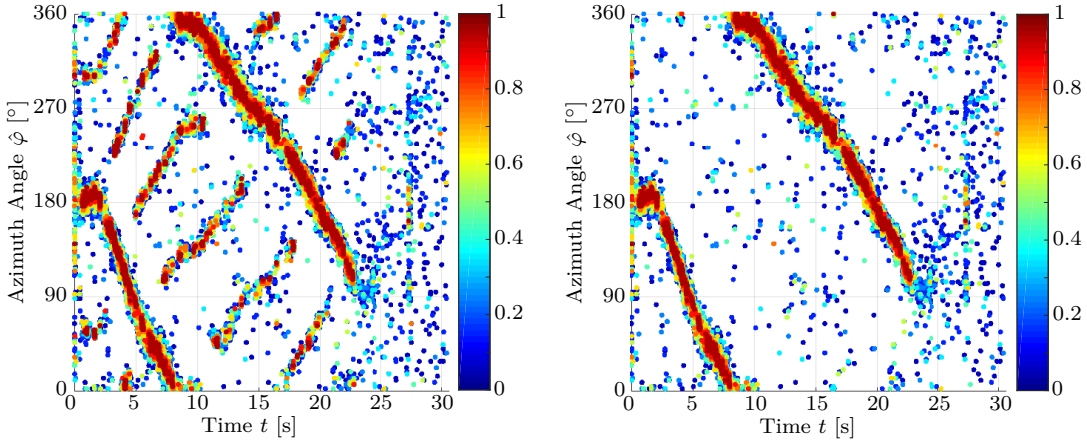
3.6.2 Handling Overlapping Sources

Another challenge for the developed multi-source localization algorithm is the overlapping of sources. This may happen if two speakers are standing close together or when they are passing by one another. The first scenario cannot be solved easily, because the two sources will cover each other and will therefore be classified as one source in the E-step. However, if this overlapping is only a time limited event, as described in the second scenario, this can be handled by a simple heuristic approach under a few assumptions:
The first assumption is that one source it not moving during the overlap. In the most real world scenarios this is acceptable because most crossovers do not occur between two moving sources. They are often happening between a human speaker and e.g. a radio, television or speaker box. The second assumption is that the fixed source is less active, therefore has less observations and a lower mixing coefficient. One reason is that fixed speaker boxes are mostly further away from the array than a human (mobile) speaker and therefore have a lower amplitude and a more diffuse noise field when the signal arrives at the array, which leads to less observations in the direction of the fixed speaker box. With this assumptions the overlap distance threshold Γ_{overlap} is defined. If two classes are closer together than this threshold, that one of them will be selected, that was more active in the last B_{overlap} frames. This is done by accumulating the mixing coefficients of these sources and comparing them. The source which has less activity will be set to the minimum mixing coefficient value π_{min}. Therefore it will not get much observation responsibilities in the next frame. This leads to a freezing of the source at the current position. If the distance of the classes is greater than the overlap distance threshold, the mixing coefficient can adapt normally again. The class freezing method is integrated after the erasing of inactive sources.

3.6.3 Handling Aliasing in the Sub-band Algorithms

Aliasing was discussed in chapter 2.3.2, which has a major impact on the developed sub-band SRP algorithm. This can be seen when looking at figure 3.7a, where the localization results for all sub-bands are shown over time. The confidences mentioned in chapter 3.2 are color coded in this plot. In this scenario there is one speaker, that moves around the microphone array for one circle. It is visible that besides the main lobe, which represents the actual speaker, also localization results of the grating lobes are detected by the sub-band SRP. The results have also a quite high confidence which can lead to false detections. The aliasing problem can

be tackled by utilizing the known class positions from the frame before. Then, the positions of the aliasing can be predicted in every frequency bin and corrected to the original value, like it is done within the wrapped Gaussians. However, hard decisions are made here to decide if an observation is moved to the main lobe. In the following this method will be described in more detail.



(a) Before aliasing correction

(b) After aliasing correction

Figure 3.7: Confidence and estimated direction of arrival over time. Confidence is color coded. Comparison before and after aliasing correction. In the depicted scenario one person is walking around the microphone array for one circle. The Capon beamformer is used.

First the knowledge of all grating lobes for every frequency and DOA has to be obtained. To this end the theoretical steered power spectrum calculation in equation 2.23 is used. All local maximum positions, besides the one at the DOA that have sufficient prominences are gathered in a list $\underline{\varphi}^{\mathrm{al}}(\hat{\varphi}, l) = (\varphi_1^{\mathrm{al}}, ...)^T$. In a next step the means for every class that have a mixing coefficient which is higher than the minimal threshold $\Gamma_{\mathrm{mix}}^{\mathrm{min}}$ will be taken as an input to obtain the grating lobe positions at every frequency bin. Then the localization results $\underline{\hat{\varphi}}$ are compared to the list of grating lobes. If the localization result has a closer distance to the grating lobe maximum position than the aliasing distance threshold Γ_{al} the observation will be corrected to the mean value of the Gaussian class. To illustrate the algorithm the pseudo code is shown in algorithm 1. In figure 3.7b the input

values that are used by the EM-update after the aliasing correction are shown.

Input : Observations $\underline{\hat{\varphi}}$, Grating lobe maximum positions $\underline{\phi}^{\mathrm{al}}(\hat{\varphi}, l)$, GMM mean $\underline{\mu}$, GMM mixing coefficients $\underline{\pi}$
Output : Aliasing corrected observations $\underline{\hat{\varphi}}^{\mathrm{al}}_{\mathrm{cor}}$
N_f is the number of unsymmetrical frequency bins of the STFT;
K is the number of classes in the GMM;
for $n = 0 : N_f$ **do**
  for $k = 1 : K$ **do**
    if $\pi_k > \Gamma^{\mathrm{min}}_{\mathrm{mix}}$ **then**
      CurrentLobePositions $= \phi^{\mathrm{al}}(\mu_k, n)$;
      for *LobePosition in CurrentLobePositions* **do**
        if $|$*LobePosition* $- \ \hat{\varphi}_n| < \Gamma_{\mathrm{al}}$ **then**
          $\hat{\varphi}_n = \mu_k$;
        end
      end
    end
  end
end
$\underline{\hat{\varphi}}^{\mathrm{al}}_{\mathrm{cor}} = \underline{\hat{\varphi}}$;

Algorithm 1: Pseudo code for the aliasing correction of the observations

4 Evaluation

In this chapter the developed broadband and sub-band post-processing algorithms are evaluated and compared to the algorithm proposed by Madhu. To realize this they were implemented with *Matlab*. The evaluation process is shown in figure 4.1. The Input signals for the ASL are the microphone signals $x(n)$ from a 7 microphone UCA+C which has a radius of 42 mm. The microphone signals are transformed by the STFT to obtain the signals $\underline{X}(l, b)$ in time-frequency representation. The ASL (described in chapter 3.1) acquires the signals and processes them by steering the D&S or Capon beamformer within a search grid of directions. The steered power spectrum is obtained and passed to the maximum search, based on the SRP for broadband or in every sub-band. The raw localization is used to calculate the confidences (Chapter 3.2). Confidences and SRP localization results are then directed to the different kinds of post-processing algorithms. For comparison the Madhu algorithm was implemented. Finally, classification results are directed to the evaluation where different metrics are calculated.

This chapter begins with discussing the modification of Madhu's post-processing algorithm, which is used as a reference. Then the evaluation setup is stated and the system setup is described, showing the parameter tuning for the different algorithms. The dataset of microphone signals is separated into a development set for tuning and a test set, where the final evaluation is conducted.[1] In the following section the results of all algorithms are stated and discussed afterwards. Finally, the results are wrapped up and a conclusion is drawn.

4.1 Reference Algorithm

As a reference the algorithm proposed by Madhu, as described in chapter 2.5.2, is used. It should be noted, that it was developed to work with ULA arrays and therefore has to be expanded for circular arrays. The classical EM-algorithm is substituted with the one in chapter 3.3.1 which is adapted for wrapped Gaussians. Furthermore the noise floor class is changed to a uniform distribution as stated in chapter 3.3.4. A new threshold on the mixing coefficients is introduced Γ_{affil} which

[1] This is done to achieve independent results and avoiding results that are overfitted on the development set.

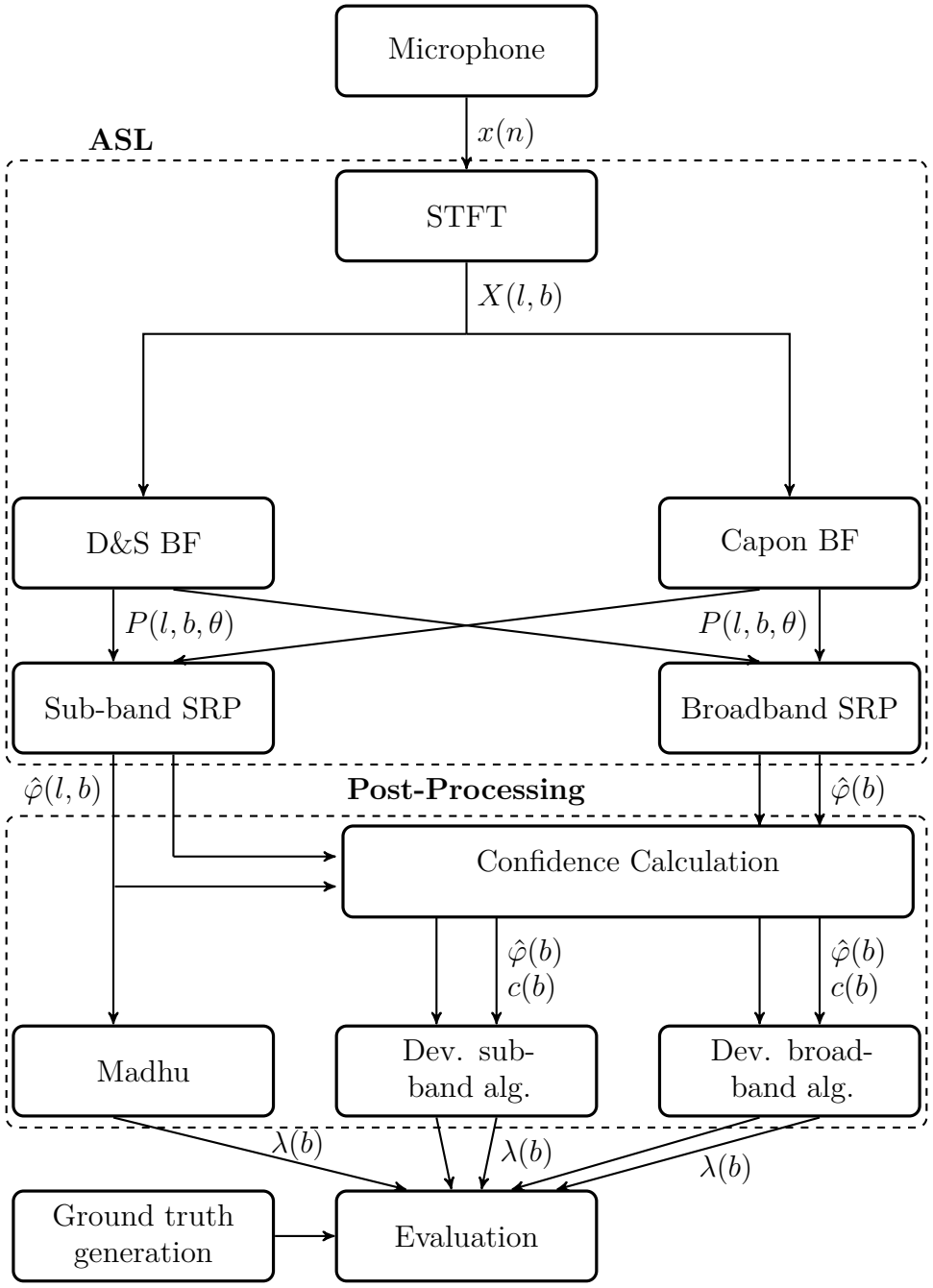


Figure 4.1: Evaluation overview. First the different raw localization results $\hat{\varphi}$ are calculated by the acoustic source localization. Then multi-source classification algorithms calculate the classification results represented by the Gaussian mixture model λ. For the evaluation the results of each algorithm are compared to the ground truth with different metrics.

is used to have a lower threshold for affiliations of observations to a class. This is added to the shrinking method: if $\exists i$, such that $\pi_i < \Gamma_{\text{affil}}$

$$\begin{aligned} \underline{\mu} &\leftarrow \underline{\mu} \setminus \mu_i \\ \underline{\sigma}^2 &\leftarrow \underline{\sigma}^2 \setminus \sigma_i^2 \\ \underline{\pi} &\leftarrow \underline{\pi} \setminus \pi_i \\ K &\leftarrow K - 1. \end{aligned} \tag{4.1}$$

The backslash $\setminus$ means the set difference. In case the mixing coefficient constraint 2.36 is violated, this is fixed with a reestimation of the parameter. The idea of using a threshold for the mixing coefficient is mentioned in [MM08], but is not integrated in his original implementation. However he used aliasing-free arrays and therefore did not have to deal with large amounts of 'false' detections. Therefore, the idea was implemented in the present work too. Last, the handling of overlapping sources, introduced in 3.6.2 is also applied to the Madhu algorithm, to ensure a fair comparison.

4.2 Evaluation Setup

In this section the evaluation setup is stated. First the microphone signal generation is presented. Then the ground truth (GT) generation is introduced which is used to compare classification results within the evaluation. Finally the different evaluation metrics are explained.

4.2.1 Microphone Signal Recordings

Microphone data were generated in an approximately 3m x 4m room with the UCA+C array placed in the center of the room on a stand in about 1.3 m of height (see figure A.1). Omnidirectional MicroElectrical-Mechanical system (MEMS) microphones were used together with phantom power supply. The microphones were installed upon a laser cut wooden plate depicted in figure A.2. The signals are relayed to a soundcard which converts them to digitals for USB-transmission to a computer, where they are finally recorded. Data acquisition was conducted during different scenarios, i.e. with multiple human speakers and speaker boxes as sound sources. Speakers could be at one static position or roaming freely in the room. Their ways were crossing and during that they were speaking with interruptions. Distances of the speaker varied from 1 m to 2 m. The full list of signal scenarios for development and test set can be viewed in the appendix A.1.

4.2.2 Metrics

To compare the developed post-processing algorithms, a broad set of metrics has to be evaluated. As a role model for the evaluation measures the field of visual object tracking is chosen. Here, a large variety of metrics exist, what can be inhibiting for the cross-paper comparison. To tackle this problem comprehensive review papers were written comparing and summarizing a broad field of measures [YMV07; Bau+08; ČLK16]. In that work the focus is on metrics for object classification but also on measures for accuracy. Because these measures are made for visual object tracking some of them had to be adapted for the developed acoustic multi-source algorithms. All measures are dependent on GT tracks (Chapter 4.2.3), which are compared against the so called system (ST) tracks of the post-processing algorithm. An ST track is defined as the mean value $\mu_k(b)$ of a class k dependent on the frame index b.
Before the evaluation metrics are introduced concepts for temporal and spatial overlap have to be defined, which are used later by the different metrics. The temporal overlap is calculated for every GT-ST pair. This is the number of frames where both tracks are present. Spatial overlap is measured by the distance of a ST track to the GT track. If this distance is smaller than a threshold, overlap is present in the frame b. The mean $\mu_k(b)$ of class k is taken as the ST track's spatial location. The GT generation is explained in a later chapter. In this work the spatial overlap threshold is set to 35°. As a more strict constraint for spatial overlap is the single source spatial overlap. Here the binary variable is only equal to one, when the ST is the only one with spatial overlap. With this stated the following metrics can be introduced.

True Positives

The true positive (TP) metric is actually a measure that is not considered in the final evaluation, but some of the following metrics are building on it. True positive is a ST track when a GT track can be assigned to it. This is done by evaluation of a temporal overlap and a spatial overlap for a pair of GT and ST. The constraint was adapted to be stricter by taking only the single source spatial overlap instead, in order to reduce double TPs for the same time span.

False Positives Duration

All ST tracks that have no TP affiliation to any GT track are considered as false positive. To have a more precise measure, in this work the false positive duration is considered as sum of the existence times of all false positives. By adopting this measure, the existence of long false positives can also be harmful in comparison to

many short false positives. In the best case scenario the false positive duration would be zero.

Root Mean Square Error

The root mean square error (RMSE) is calculated for every TP ST track and when having spatial overlap. Spatial overlap is set as a condition because this should prevent ST tracks that are totally drifting away from being considered and thus increasing the RMSE heavily. RMSE is a measure for the accuracy of the localization and should be zero in an optimal case.

Track Interruptions

Track interruptions indicate the lack of continuous tracking of a GT track. When multiple ST tracks are assigned to one GT this is counted as interruption. Assigned is defined by the TP measure. In optimal condition the track interruption should be zero.

ID Change

The metric ID change (IDC) is introduced for distinguishing if a ST track is assigned to multiple GT tracks. This can be seen as a mix-up of the classification. For this the single source spatial overlap is taken. If one ST is assigned by single source spatial overlap to one source and after a time it is assigned to another source, then an IDC happens. An example can be found in figure 4.4.

Initial Detection Lag

The initial detection lag measures the time gap between the start of the GT track and the start of the first TP ST track. This indicates how fast the algorithm can detect new source appearances, which is an important measure when the localizer is used for wake up word detection in a speech recognition device. Overall the optimal latency should be zero.

Track Completeness

This metric measures the time overlap of GT tracks and ST tracks that is assigned as TP to the GT track, divided by the total time span of the ST track. Multiple temporal overlap of multiple ST tracks are only considered once. The measure is adapted in the way, that the ST track must also have spatial overlap with the GT track. Optimal track completion would be at 100%.

Active Track Completeness

Similarly to the track completeness the *active* track completeness only allows for values of ST tracks that are active. Activity is defined over the same threshold as the increase/decrease decision of the TTL (equation 3.31). This will give a measure for the quality of detecting the sources simultaneously. Optimal active track completion would also be 100%.

4.2.3 Ground Truth Generation

The GT generation was conducted by the labeling tool programmed in *Matlab*, shown in figure A.3, and giving in the end the SRP broadband localization results. The SRP localization results have been chosen as the basis for labeling, because the main goal of this work is to compare the post-processing and not the ASL algorithms. On this basis the results were retraced with setting marks on the localization results and interpolating the data points between them linearly. These results were matched to marks that were communicated by the speaker in the recording or noted during the recording.

4.3 System Setup

The system setup states how the parameters in the different stages of the multi-source localization algorithms were tuned. First the parameters of the ASL are discussed, then confidence calculation and finally the post-processor.

4.3.1 Acoustic Speaker Localization

For the ASL parameters this is summarized in table 4.1; most of them were already mentioned. One that is not mentioned before is the steering resolution. As discussed before, when using SRP a beamformer is steered within a search grid of directions, and then the maximum power is searched over all points. The

steering resolution describes the density of the grid points. In this thesis the grid has a resolution of 2° in the azimuth plane, which means there were $360°/2° = 180$ points being examined.

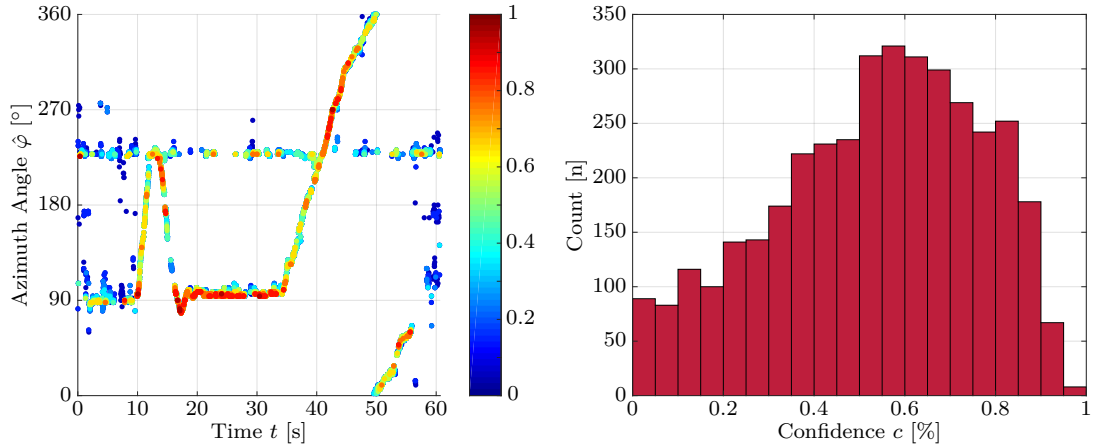
Parameter Name	Value
Array Geometry	UCA+C
Microphone Number M	7
Array radius	42 mm
Sample Frequency f_s	16 kHz
STFT Window $\mathcal{W}$	Hann
STFT length L	512
Frame Shift O	256
Smoothing Constant α	0.5
Steering resolution	2°
Capon BF diagonal loading ϵ	0.1

Table 4.1: Parameter setting for the acoustic speaker localization with steered response power.

4.3.2 Confidence calculation

The confidence calculation is executed with equation 3.4 and 3.5 for broad and sub-band processing, respectively. The parameters for the confidences were set with the help of evaluating two plots. Examples are shown in figure 4.2 for broadband and in figure 4.3 for sub-band SRP processing with a D&S beamformer. In the broadband SRP the aim for the confidence distribution was to spread the values over the full range evenly as good as possible. This is shown in the histogram 4.2b for one scenario. Spreading can be done by increasing the sensitivity μ_{v}. To lift the whole value range up, the offset α_{O} could be increased.
The distribution of confidences can also be seen in figure 4.2a. Here the estimated DOA is shown over time, where the confidence is color coded for every DOA value. It can be observed, that for example the outliers are having lower confidences.

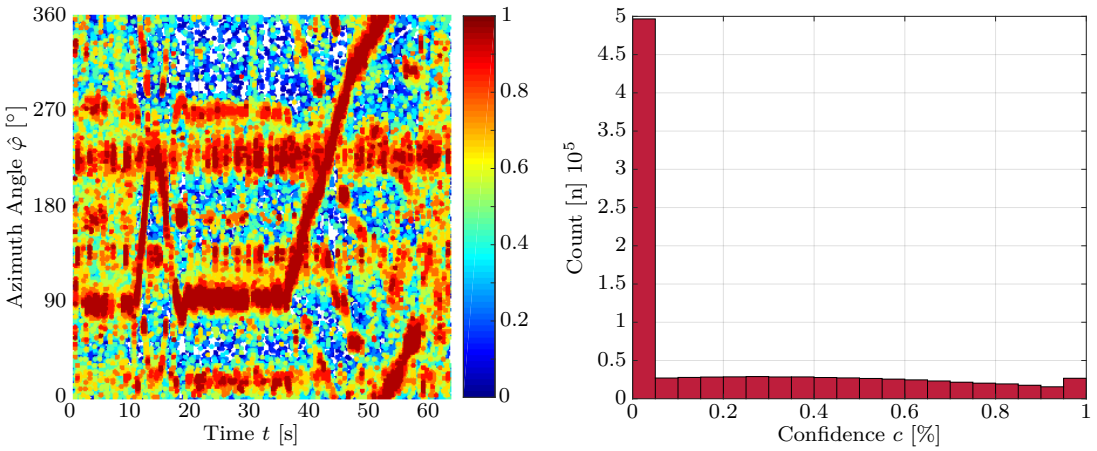
The approach of an evenly distributed confidence cannot be pursued for the sub-band processing, because there were lots of values with very low confidence. This is also shown in the theoretical spectrum (figure 2.5). In the lower bins there is not much dynamic over the azimuth plane, which results in a very low confidence for many values. When looking at the histogram 4.3b, confidences with a value of zero are much more represented than the rest. This reduced the computational load because values with zero confidence have no effect upon the mean calculation (equation 3.22) and therefore can be neglected directly. When looking at figure 4.3a and comparing it to figure 4.2a much more values are present, even when neglecting the zero values. Furthermore, the variance is also much higher as in



(a) Confidence and estimated direction of arrival over time. Confidence is color coded

(b) Histogram of confidence values

Figure 4.2: Plots for tuning the confidence of the broadband algorithm with a delay and sum beamformer. Scenario D4 is depicted as example.



(a) Confidence and estimated direction of arrival over time. Confidence is color coded

(b) Histogram of confidence values

Figure 4.3: Plots for tuning the confidence of the sub-band algorithm with a delay and sum beamformer. Scenario D4 is depicted as example.

the broadband processing. All parameter for the D&S and Capon beamformer in broadband and sub-band SRP are shown in table 4.2.

4.3.3 Post-Processing

For the algorithm of the multi-source post-processing a more sophisticated approach was used to tune the parameters. In the beginning a set was examined by try-and-

Beamformer	Delay and Sum		Capon	
SRP	Broadband	Sub-band	Broadband	Sub-band
Sensitivity μ_v	1.8	4	1.7	10
Offset α_o	1.7	1.8	1.8	1.4

Table 4.2: Confidence parameter

error for all algorithms. Afterwards a parameter sweep was done where parameters that have a big impact on the classification were varied widely, which resulted in over 250 parameter sets that had to be evaluated. With these sets the classification algorithms were run on the whole test base of signals. The Descriptions of the tests set are shown in table A.1. Afterwards the results were evaluated with the help of the introduced metrics in chapter 4.2.2. The next step was a filtering with a set of thresholds on the metrics. Priority list for the filtering was

1. ID changes
2. Track completion
3. RMSE
4. Initial detection lag
5. Interruption
6. False positive duration

However, it had to be contemplated to reduce some higher priority results against a multiple increase in lower priority ones. The found parameter sets were then used for the final evaluation, with the test data base. The parameter sets can be found in table A.2. The parameter set for the Madhu algorithm is stated in table A.3.

4.4 Results and Discussion

In this chapter the results obtained by processing the microphone signals from all the scenarios of the evaluation set are stated. The signals are described in table A.1 and the graph of GT tracks over time are shown in figure A.4 to A.10. The recording setup is described in chapter 4.2.1. The four developed algorithms with D&S and Capon beamformer using broadband SRP and sub-band SRP are compared to each other and then with the reference algorithm from Madhu. In the following the abbreviations BB D&S or BB CAP are used for the developed broadband algorithm with a D&S or Capon beamformer and SB D&S or SB CAP is used for developed sub-band algorithm with a D&S or Capon beamformer. The

complete evaluation setup is illustrated in the block diagram in figure 4.1. The chapter is structured by the different metrics introduced in chapter 4.2.2. First, in every section a bar plot is analyzed that shows the results of all five algorithms for the specific metric. Afterwards exemplary scenarios for some algorithms are picked, to explain the results and relate them to the parameters used.

4.4.1 ID Changes

IDCs are counted when the GT track changes which the ST is assigned to.

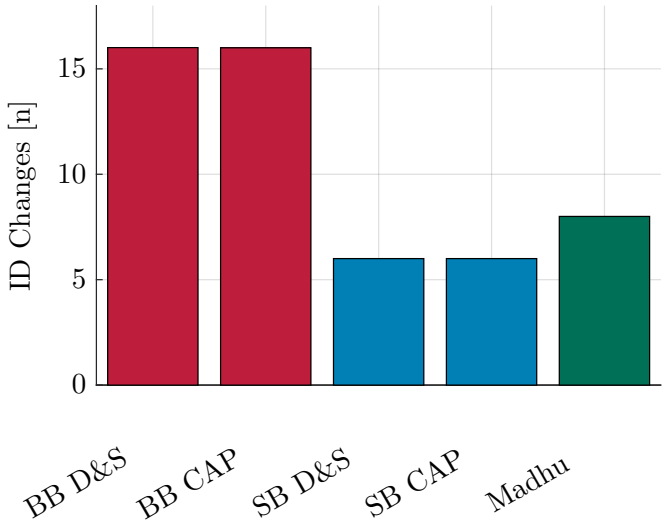


Figure 4.4: Bar plot of the number of ID changes for all considered algorithms over the evaluation set. Much more ID changes occur when processing scenarios with the developed broadband algorithms.

From figure 4.4 it is obvious, that the broadband algorithms (in red) are more predisposed to IDCs. The Madhu algorithm (green) has also a few more IDCs than the developed sub-band algorithms (blue).
When looking deeper into the different scenarios nearly all IDCs are happening in either E5 or E7 (see table A.4). At scenario E7 processed with BB D&S algorithm (see figure 4.5) it is also visible, that the algorithm cannot resolve the close distance between GT1 and GT2.

The ST track 1 is jumping between the two GT tracks which results in IDCs. The instances of IDCs are marked by a black circle. The distance between the two GT tracks are so close together that both observations caused by the two speakers can be accounted to one class.

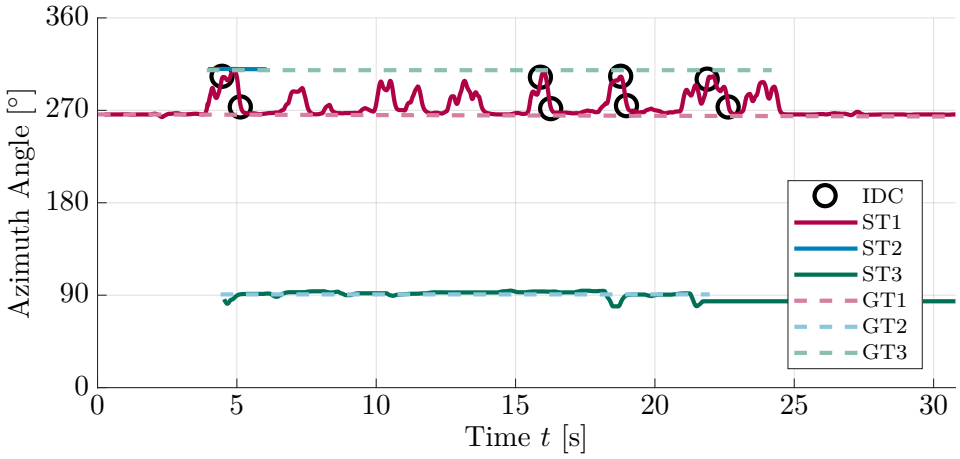


Figure 4.5: Scenario E7, processed with the developed broadband algorithm and a delay and sum beamformer, where the system track has multiple ID changes between ground truth track 1 and 2.

The constant variance parameter σ^2_{const} has an impact on this. When the variance is small only the nearest observations are getting assigned to this source. Vice versa, if the variance is high, also more distant observations are assigned to the ST track and the algorithms are adapting to this. When comparing the variances σ^2_{const} over the different algorithms in table A.2, it is visible that the deviations are higher for the broadband algorithms. Therefore the sub-band algorithms are able to solve this scenario which is shown in figure A.11.
Another reason for occurrence of IDCs can be seen in scenario E5 (figure 4.6). Here the speakers are both moving and crossing each other. To prevent IDCs when crossover happened the assumptions was made in chapter 3.6.2 that only one speaker is moving when sources are close together and only one is speaking at this time. Therefore mixing coefficient π of one source can be reduced, which can also be seen as a temporal freezing of this class at the current position. In scenario E5 both sources are moving and talking at the same time so the assumptions are violated. However the source which is less active will have reduction of the mixing coefficient and therefore freezes. This is visible in figure 4.6 at 6 seconds. ST2 is frozen because it is the less active source in comparison with ST1. Therefore ST2 cannot follow the GT track. This effects all algorithms in the same way.

4.4.2 Track Completeness

The metrics track completeness and active track completeness describe how much of the GT track is completed by the ST tracks in percent. In the active track completeness the ST track must also be active at that moment. In figure 4.7 the completeness for all algorithms over the complete evaluation set is depicted.

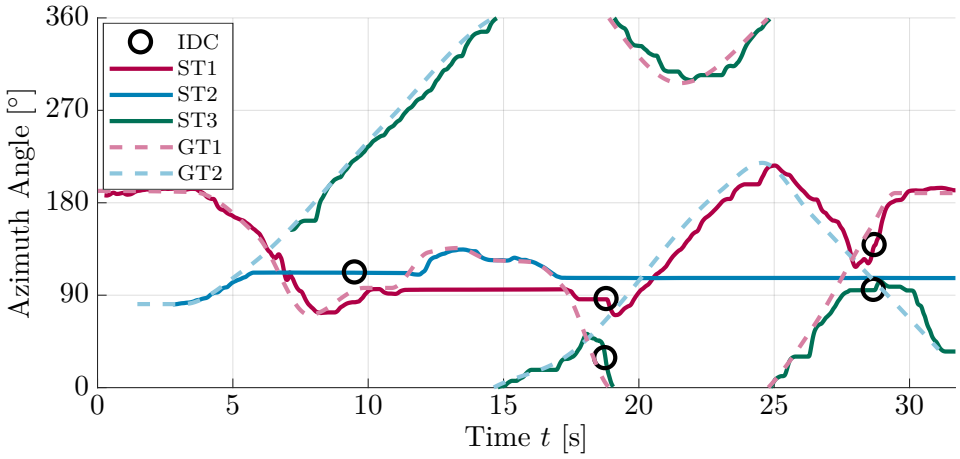


Figure 4.6: Scenario E5, processed with the developed broadband algorithm and a Capon beamformer. Multiple ID changes because of freezing source mechanism.

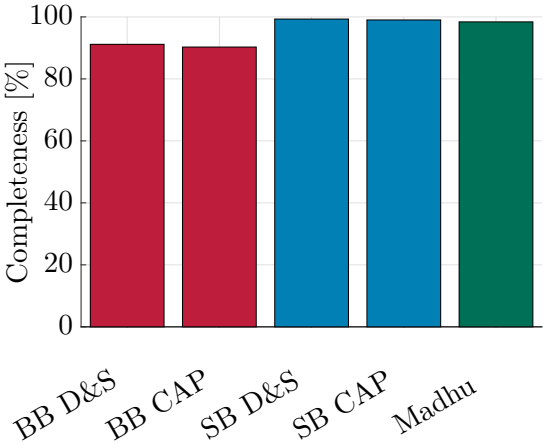


Figure 4.7: Bar plot of track completeness for all considered algorithms. Sub-band and Madhu's algorithm have nearly 100% completeness.

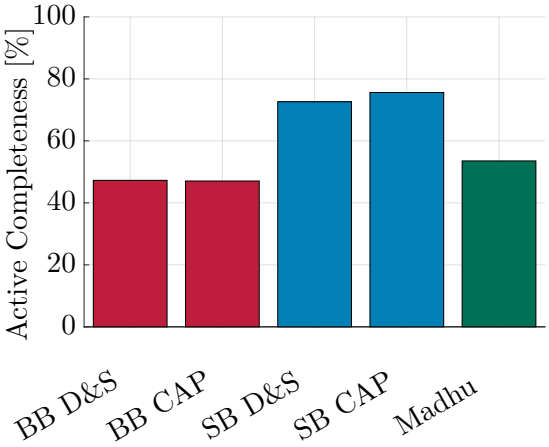


Figure 4.8: Bar plot of active track completeness for all considered algorithms. Sub-band algorithms perform 20% points better than the rest.

The broadband algorithms have lower completeness at about 90% than the sub-band algorithms and Madhu algorithm at about 98%. Main reason for the difference between broadband and sub-band is the hypothesis test which decides if new classes are added to the system. This can be seen in figure 4.9, where the developed broadband algorithm is used with a D&S beamformer. The ST track 3 has a long detection lag when comparing it to the GT track 3, which reduces the completeness. Reason for that is the hypothesis threshold $\Gamma^{\text{add}}_{\text{class}}$ in combination with the PDF of the floor class p_{floor}. The threshold stated how many values have to be represented by the floor class before a new class is opened. Table A.2 shows, that $\Gamma^{\text{add}}_{\text{class}} = 7$

(D&S) or $\Gamma_{\text{class}}^{\text{add}} = 8$ (Capon) out of $N = 16$ values have to be assigned to the floor class. Therefore the new class detection triggers only when more than half of the observations in the circular buffer comes from a direction which is not represented in the current GMM. Following this ST3 started seconds after the first observation and was visible at 90°, which results in a lower track completeness.

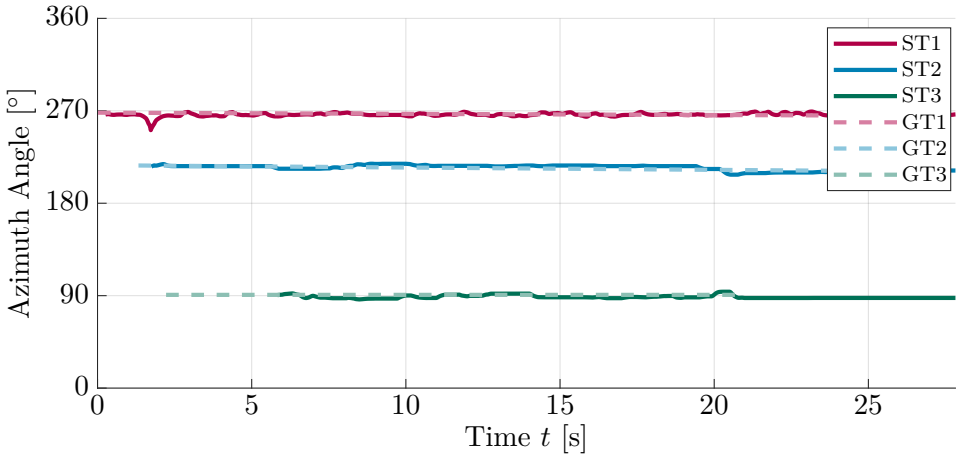


Figure 4.9: Scenario E6, processed with the developed broadband algorithm and a delay and sum beamformer. System track 3 has a long detection lag in comparison to ground truth track 3, which leads to a decrease of completion.

In figure 4.8 the active completeness drops deep for the broadband algorithm to under 50% compared to the completness. The sub-band algorithm still has over 75% active completeness. This is caused by the fundamental difference between broadband and sub-band algorithm. The broadband algorithm only delivers one value per frame and therefore only one class can be active at the same time. The sub-band algorithm can stimulate multiple classes in one frame. This higher active completeness may lead to a lower RMSE because there is less loss of detection for short times which may have impact on the RMSE when considering moving sources. The details of the difference between BB and SB in active completeness is depicted in figure A.12.
Moreover, the Madhu algorithm suffers also under low active completeness. This is due to the threshold for minimal assignments to a class. When less then this threshold are assigned to a class during the EM-algorithm the class will be deleted from the GMM. Therefore the observations in this frame have no impact anymore on the overlaying GMM. The low active completeness may also impact the RMSE in the same way as the broadband algorithm.

4.4.3 Root Mean Square Error

The root mean square is calculated between the GT track and its assigned ST track. It is a measure for accuracy of the post-processing algorithms. Figure 4.10 shows that the sub-band algorithms have the lowest RMSE at around 5°. The broadband algorithms have a RMSE of around 7.5° and the Madhu algorithm of 6.5°. This results may be affected by the poor performance of the broadband algorithm discussed in scenario E7, where multiple IDCs happen and therefore the ST track has a big deviation from the GT track. Also scenario E5 is affected by a lot of IDCs. If both scenarios are not considered in the RMSE calculation, the results depicted in figure 4.11 are obtained.

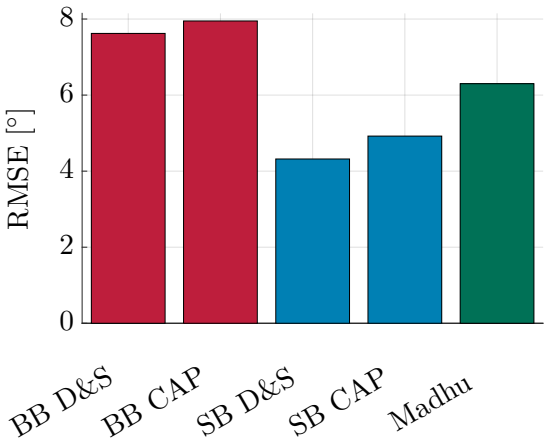


Figure 4.10: Bar plot of the root mean square error for all considered algorithms over the whole evaluation set. Sub-band algorithm have the lowest error.

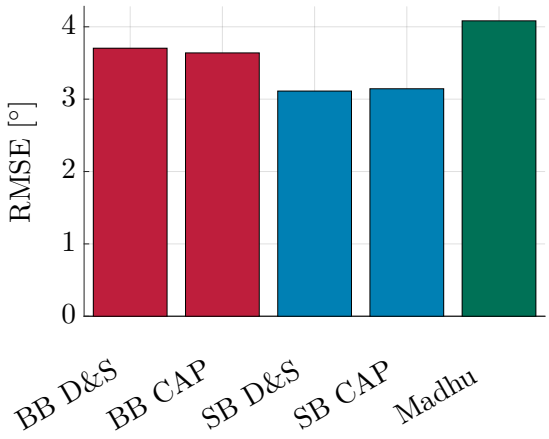


Figure 4.11: Bar plot of the root mean square error for all considered algorithms. Scenario E5 and E7 are not considered. The Madhu algorithm has now a higher root mean square error than the broadband algorithms.

It still shows that the sub-band algorithms have a better performance but the error of the broadband algorithms reduced by large and overtook the Madhu algorithm. The difference between the algorithm is now quite small in an range of 1°. However a few reasons are found for the performance differences. Therefore, the analysis has to be divided in static scenarios and dynamic scenarios, according to moving or non-moving speakers. In static scenarios the broadband algorithms have an advantage over the sub-band algorithms and the Madhu algorithm, respectively. This can be seen in table A.5, where the RMSE is listed for the different scenarios and algorithms. Static scenarios were E1, E6 and E7, however, the E7 could not correctly be solved by the broadband algorithm. In E1 and E6 the broadband algorithms generally have a lower RMSE. The broadband processing with the

D&S beamformer in E1 is seen as an outlier. The reason is, that the averaging over all frequencies delivers a more precise localization result.
In the sub-band algorithms the averaging happens by the GMM, but here only the maximum values are taken into account and moreover not over the whole steered power spectrum, which leads to a loss of information. In dynamic scenarios (E2, E3, E3), however, the sub-band processing has a lower RMSE, which has two reasons. First, the broadband algorithm buffers the localization result over time and therefore has values in the current buffer, that are already a few frames old. This results in a delayed swinging of the post-processing algorithm. The delay is also influenced by the MAP-adaption parameter β, which decides how high the a priori knowledge shall be considered in the current update. The second reason is the low active completeness of the broadband algorithms. This leads to detection loss for a short time, where the source already has moved further. The Madhu algorithm suffers from the same detection loss. Furthermore in the Madhu algorithm is no MAP-like smoothing parameter β, when updating the overlaying GMM with the current value. The ratio between current value and overlaying GMM value is determined by the mixing coefficients which can be seen in the update equations 2.48. The smoothing therefore cannot be adapted in this algorithm. An example for broadband, sub-band and Madhu is depicted in figure 4.12, showing the difference is small.

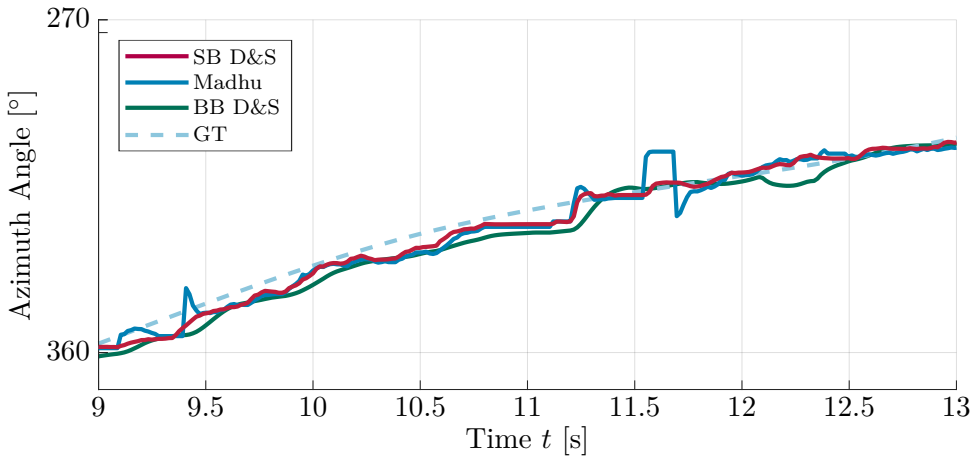


Figure 4.12: Scenario E1 for different algorithms. The Broadband algorithm (green) is always lying behind the ground truth track because of the time buffering. The Madhu algorithm (blue) has outliers because of less smoothing. The sub-band algorithm (red) has the smallest root mean square error to the ground truth track.

Another problem can be seen in scenario E7, which was excluded before. Nevertheless, this example can be generalized for all other scenarios. Madhu's algorithm was not designed with aliasing in mind, because he used an aliasing free array.

However in this work the array has spatial aliasing in the upper frequencies. This has an impact on the RMSE as visible in figure 4.14. Here the ST track 3 diverges from the GT track because of the aliasing of GT2. The aliasing can be seen in figure 4.13. This also happens in other scenarios.

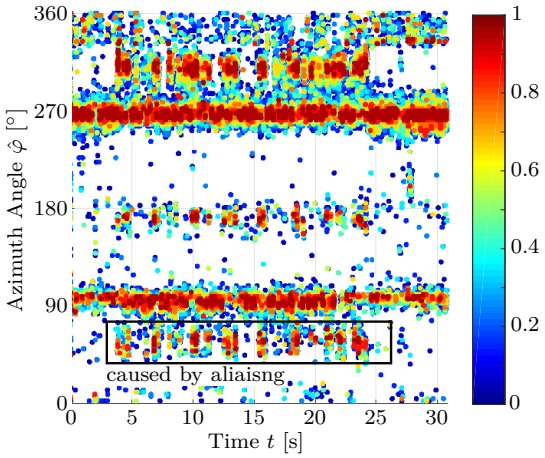


Figure 4.13: Confidence and estimated direction of arrival over time for scenario E7 over time. Confidence is color coded. Aliasing from GT2 is visible at 70°.

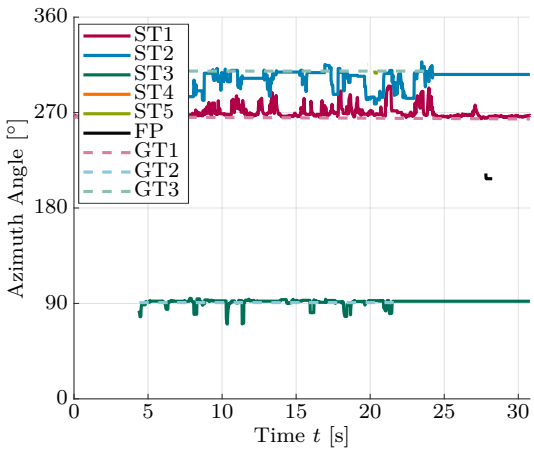


Figure 4.14: Scenario E7, processed with Madhu's algorithm. ST3 diverges from the ground truth to the aliasing caused by the speaker represented by GT2.

4.4.4 Initial Detection Lag

The initial detection lag is the time difference between the start of the GT and ST track. The mean lag is taking all GT - ST pairs over all scenarios into account. In figure 4.15 a large difference between the broadband and the sub-band algorithm can be noticed. This is due to the poor performance in the detection of new sources in scenario E1 and E6 (see table A.6). The detection error was already discussed in 4.4.2, therefore figure 4.16 shows the results for the lag when these scenarios are excluded. The mean initial lag is reducing at large in the broadband algorithm, though staying far behind the sub-band and Madhu algorithm. The mean initial lag of the broadband algorithm is around 150 ms and the sub-band algorithms' is 50 ms (D&S) or 30 ms (Capon). The lag from Madhu's algorithm is also at 30 ms. The lag times can be set into relation to the length of a frame, which can be calculated by the frameshift divided by the sampling rate $O/f_s = 256/16\,000\,\text{Hz} = 0.016\,\text{s} \equiv 16\,\text{ms}$. So it is seen that the broadband algorithm needs approximately 9 frames to detect a new class, which corresponds to the hypothesis threshold $\Gamma^{\text{add}}_{\text{class}}$ of 7 or 8. As explained before, the broadband algorithm buffers over time and therefore needs at least the number of the hypothesis threshold observations that are accounted to the floor class, to detect a new source. The other algorithms need only 2 to 3

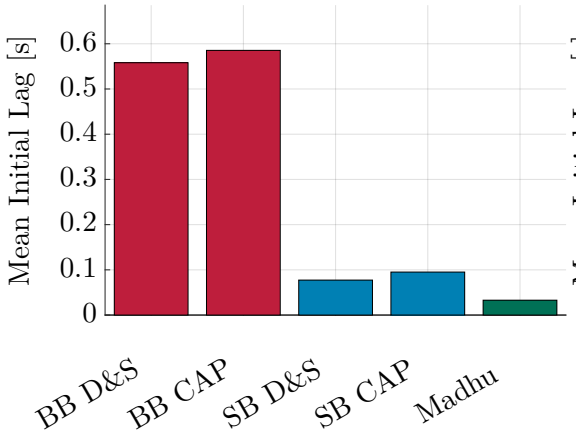


Figure 4.15: Bar plot of the mean initial detection lag for all considered algorithms over the whole evaluation set. Madhu's algorithm has the lowest initial lag.

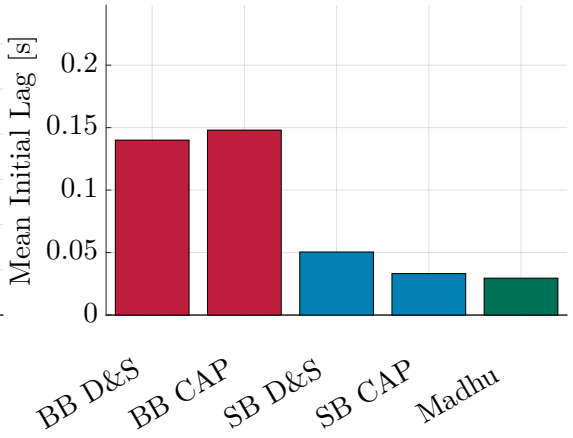


Figure 4.16: Bar plot of the mean initial detection lag for all considered algorithms. Scenario E1 and E6 are excluded from the calculation. The broadband algorithm have a much lower lag but still longer than the other algorithms.

frames to detect a new source because they are able to detect them from the first frame on, due to the buffer over frequency.

4.4.5 Track Interruptions

Track interruptions are counted when multiple ST tracks are assigned to one GT track. The results are shown in figure 4.17. All interruptions are due to the crossovers in scenario E5 (see table A.7). These interruptions are happening for the same reason as the IDCs happen. The freezing of one class prevents to track the source during the crossover. After the crossover the source went so far away that a new source is opened which counts as interrupt. However, overall can be stated that all the algorithms perform with only very few interruptions. Parameters that influence the track interruption are the variance σ^2_{const} and the MAP-adaption parameter β. They are influencing how a moving source can be followed. When the variance is large the observation can also be far away in the next frame and the class would follow. However, also the beta must be low because it decides how much of the previous estimation is influencing the current one. Other parameters would be the TTL parameters which decide how fast sources are 'dying'. When TTL increase is low, the class would vanish really fast when no observations are assigned to it.

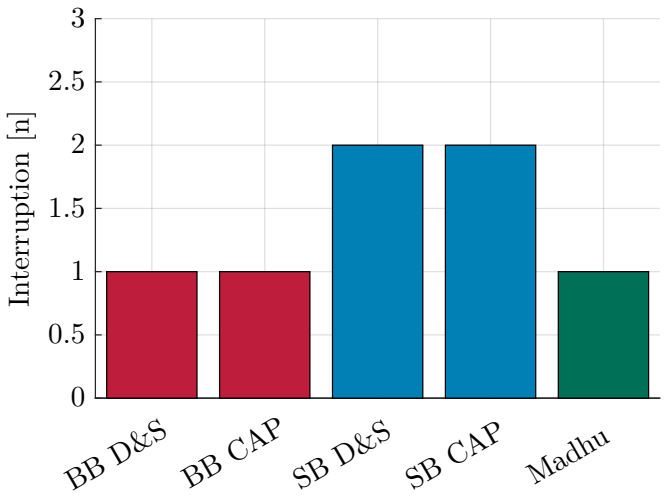


Figure 4.17: Bar plot of the interruption count. All algorithms have a very low count of interruptions.

4.4.6 False Positives

False positives are detected when ST tracks have no spatial and temporal overlap to any GT track. The sum of the FP temporal duration over all scenarios is analyzed in figure 4.18. The broadband algorithm, especially with the Capon

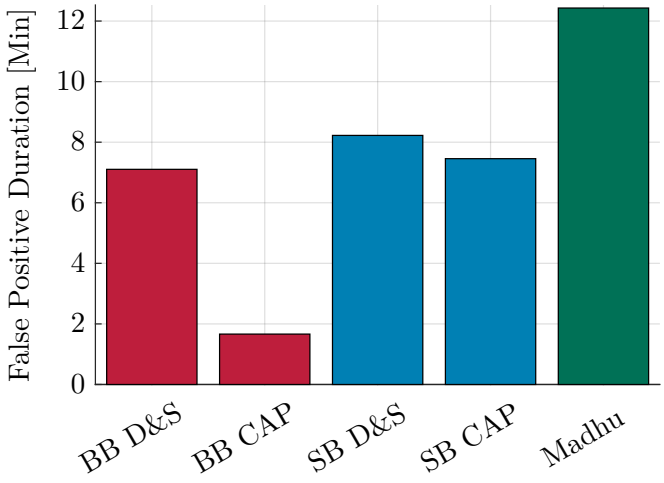


Figure 4.18: Bar plot of the false positive duration for all considered algorithms. The broadband Capon algorithm has the lowest and Madhu's algorithm the highest false positive duration.

beamformer, have the lowest FP duration. Madhu's algorithm with approximately

12 s has the longest FP duration. When looking at table A.8 it is visible that all algorithms have problems with scenario E3. Here one speaker was standing fixed and the other was going around the array and speaking from different directions. When a non speaking person walked around he or she also made noise which lead to undesired short detections. Therefore the FPs occur in this example more often than in others. The Madhu algorithm has the most problems with it which results in a long false positive duration. This is due to a threshold Γ_{affil} of the mixing coefficient which decides if a source is deleted in the EM-algorithm. Sometimes the false observations, caused for example by aliasing, are so many, that a new class is openend. Here a trade-off between FP and less overlaying GMM updating has to be made. In table A.8 is also visible that Madhu's algorithm has nearly in every scenario false detections, however, they are mostly very short because of a low TTL initial value. False positive duration can be influenced by the hypothesis threshold and the TTL parameter. To reduce the FP duration the TTL initial value or the TTL increase may be lowered. The difference between the broadband algorithm with the D&S beamformer and Capon is interesting because the influencing parameters are nearly the same. The reason for that may be the different confidence values and that the confidences are in general a little bit higher with the D&S beamformer. Therefore the class in scenario E3 is not dying as fast as when using a Capon beamformer.

4.5 Conclusion

The results drawn from the evaluation favor the sub-band algorithm with both kinds of beamformers in nearly each metric. They have the fewest ID changes, the highest completion rates and the lowest root mean square error.
However the Madhu algorithm is a whit faster (which may be due to inaccuracy of time measurement) in terms of detecting new sources and the false positive duration is longer than that of the broadband algorithm. It is difficult to decide if the Madhu algorithm performs better than the broadband algorithm. But considering the more important metrics like IDCs, completeness and initial detection lag, Madhu's algorithm is at an advantage.
When comparing the use of delay and sum and Capon beamformer, no big difference can be stated, except of the false positive (FP) duration. Here the broadband Capon has much less FP duration then the D&S beamformer, which may be due to differences in the confidence calculation.
One aim of this work was to develop a real-time algorithm and therefore the computational load shall also be regarded. In general the developed broadband algorithm has less to calculate, because it only processes $N = 16$ values per time step while the sub-band and Madhu algorithms are dealing with $N = 257$. The Madhu algorithm is doing much more efforts than the developed algorithms, because it executes a complete expectation–maximization algorithm with many iterations

in one frame. Nevertheless, for all the basic scenarios the performance of the developed algorithms is excellent and mostly outreaching the reference algorithm. When looking at aspects that could be improved, it may be the crossover handling. When one of the speakers is moving and crossing the other's direction, all the algorithms perform poorly as seen in E5. This could be improved by detecting, which source is dynamic or static and introducing certain heuristics for both cases.

5 Summary and Prospect

Aim of this work was to develop an efficient multi-source localization algorithm that is able to estimate the acoustic scene in the spatial domain by utilizing an uniform circular array with center microphone. Furthermore, the question was raised if a broadband localization is sufficient for the detection of multiple speakers, or the localization has to be done in distinct frequency bands.

A classification algorithm for localization results based on a wrapped Gaussian mixture model was developed. The WGMM parameters were estimated by a maximum a posteriori adaptive expectation–maximization algorithm under consideration of confidence values. When sub-band localization results where used, a method for handling spatial aliasing, based on knowledge about the theoretical spectrum, was developed. In the evaluation of the algorithms the delay and sum and Capon beamformer were used with the broadband and sub-band steered response power method to yield the raw localization estimates on different acoustic scenarios. These were prepared with the developed post-processing algorithm and a reference algorithm developed by Madhu. The classified localization results were compared to the ground truth. The evaluation results show, that it is possible to obtain a multi-source localization with broadband localization results. However, a large increase of performance, especially for initial detection lag and root mean square error, can be achieved by using localization results in every frequency band (or sub-band). Especially, when evaluating the processing load the broadband algorithm is greatly to favor because of a very small buffer length N. When comparing the developed algorithm to Madhu's method in terms of processing load, it can be stated, that the developed algorithm is more efficient in computation due to the usage of less EM-iteration steps. Furthermore, the sub-band algorithm outperforms or is equal to Madhu's algorithm in every metric. Especially in active track completeness and false positive duration it is considerably better than the Madhu algorithm.

This work lays the basis for further investigations in multiple directions. For better handling of situations with crossovers of sources, a tracking algorithm like a particle filter may be implemented like in other state of the art algorithms [VMR07; NDV18]. A further interesting aspect would be, to compare to other algorithms (like the GM-PHD [Eve+15]) and on a vaster signal basis. Therefore, a participation in the IEEE-AASP Challenge on Acoustic Source Localization and Tracking (LOCATA) might be contributing [Löl+18]. Furthermore, it could

also be of interest to evaluate other acoustic speaker localization methods for the generation of raw localization results like the subspace based methods [WK85; YKM06; Pan+17; DP01].

Bibliography

[Bau+08] Axel Baumann, Marco Boltz, Julia Ebling, et al. „A Review and Comparison of Measures for Automatic Video Surveillance Systems“. In: *EURASIP Journal on Image and Video Processing* 2008.2008:824726 (July 8, 2008), pages 1–30.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[BOS08] Alessio Brutti, Maurizio Omologo, and Piergiorgio Svaizer. „Localization of multiple speakers based on a two step acoustic map analysis“. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE. 2008, pages 4349–4352.

[Bro+08] I N Bronstein, K A Semendjajew, G Musiol, and H Mühlig. *Taschenbuch der Mathematik*. 7th edition. Verlag Harri Deutsch, 2008.

[BW13] Michael Brandstein and Darren Ward. *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media, 2013.

[Cap69] J. Capon. „High-resolution frequency-wavenumber spectrum analysis“. In: *Proceedings of the IEEE* 57.8 (1969), pages 1408–1418.

[ČLK16] Luka Čehovin, Aleš Leonardis, and Matej Kristan. „Visual object tracking performance measures revisited“. In: *IEEE Transactions on Image Processing* 25.3 (2016), pages 1261–1274.

[DP01] Elio D Di Claudio and Raffaele Parisi. „WAVES: Weighted Average of Signal Subspaces for Robust Wideband Direction Finding“. In: *IEEE Transactions on Signal Processing* 49.10 (2001), pages 2179–2191.

[Eve+15] Christine Evers, Alastair H Moore, Patrick A Naylor, et al. „Bearing-only Acoustic Tracking of Moving Speakers for Robot Audition“. In: *International Conference on Digital Signal Processing (DSP)*. IEEE. 2015, pages 1206–1210.

[FM18] Amy Ann Forni and Rob van der Meulen. *Gartner Says Worldwide Spending on VPA-Enabled Wireless Speakers Will Top $2 Billion by 2020*. Edited by Inc. Gartner. 2018. URL: `https://www.gartner.com/newsroom/id/3464317` (visited on 02/22/2018).

[GL94] J L Gauvain and Chin-Hui Lee. „Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains“. In: *IEEE Transactions on Speech and Audio Processing* 2.2 (1994), pages 291–298.

[HQF11] Frithjof Hummes, Junge Qi, and Tim Fingscheidt. „Robust Acoustic Speaker Localization with Distributed Microphones“. In: *19th European Signal Processing Conference*. IEEE. 2011, pages 240–244.

[KS89] Jeffrey Krolik and David Swingler. „Multiple Broad-band Source Location Using Steered Covariance Matrices“. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.10 (1989), pages 1481–1494.

[KV96] Hamid Krim and Mats Viberg. „Two Decades of Array Signal Processing Research: the Parametric Approach“. In: *IEEE signal processing magazine* 13.4 (1996), pages 67–94.

[Lag11] Joseph Louis Lagrange. *Mécanique Analytique*. Vve. Courcier, Paris, 1811.

[Loi13] Philipos C Loizou. *Speech Enhancement: Theory and Practice*. CRC press, 2013.

[Löl+18] Heinrich Löllmann, Christine Evers, Alexander Schmidt, et al. *IEEE-AASP Challenge on Acoustic Source Localization and Tracking*. 2018. URL: `https://lms.lnt.de/locata/` (visited on 02/22/2018).

[Mad10] Nilesh Madhu. *Acoustic Source Localization: Algorithms, Applications and Extensions to Source Separation*. Der Andere Verlag, 2010.

[MHA08] Rainer Martin, Ulrich Heute, and Christiane Antweiler. *Advances in Digital Speech Transmission*. John Wiley & Sons, 2008.

[MM08] Nilesh Madhu and Rainer Martin. „A scalable framework for multiple speaker localization and tracking“. In: *Proceedings of the International Workshop for Acoustic Echo Cancellation and Noise Control (IWAENC)*. IEEE. 2008.

[MR17] National Public Media and Edison Resarch. *The Smart Audio Report*. Edited by npr.org. 2017. URL: `http://nationalpublicmedia.com/wp-content/uploads/2018/01/The-Smart-Audio-Report-from-NPR-and-Edison-Research-Fall-Winter-2017.pdf` (visited on 02/22/2018).

[NDV18] Joonas Nikunen, Aleksandr Diment, and Tuomas Virtanen. „Separation of Moving Sound Sources Using Multichannel NMF and Acoustic Tracking“. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.2 (2018), pages 281–295.

[Pan+17] Hanjie Pan, Robin Scheibler, Eric Bezzam, et al. „FRIDA: FRI-based DoA Estimation for Arbitrary Array Layouts“. In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pages 3186–3190.

[Pav+13] Despoina Pavlidi, Anthony Griffin, Matthieu Puigt, and Athanasios Mouchtaris. „Real-time Multiple Sound Source Localization and Counting using a Circular Microphone Array“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.10 (2013), pages 2193–2206.

[SH05] Xiaohong Sheng and Yu-Hen Hu. „Maximum Likelihood Multiple-Source Localization Using Acoustic Energy Measurements with Wireless Sensor Networks“. In: *IEEE Transactions on Signal Processing* 53.1 (2005), pages 44–53.

[VMR07] Jean-Marc Valin, François Michaud, and Jean Rouat. „Robust Localization and Tracking of Simultaneous Moving Sound Sources Using Beamforming and Particle Filtering“. In: *Robotics and Autonomous Systems Journal (Elsevier)* 55.3 (2007), pages 216–228.

[WK85] H. Wang and M. Kaveh. „Coherent Signal-Subspace Processing for the Detection and Estimation of Angles of Arrival of Multiple Wide-Band Sources“. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33.4 (Aug. 1985), pages 823–831.

[YKM06] Yeo-Sun Yoon, Lance M Kaplan, and James H McClellan. „New DOA Estimator for Wideband Signals“. In: *IEEE Transactions on Signal Processing* 54.6 (2006), pages 1977–1989.

[YMV07] Fei Yin, Dimitrios Makris, and Sergio A Velastin. „Performance Evaluation of Object Tracking Algorithms“. In: *International Workshop on Performance Evaluation of Tracking and Surveillance, Rio De Janeiro, Brazil*. IEEE. 2007, page 25.

A Appendix

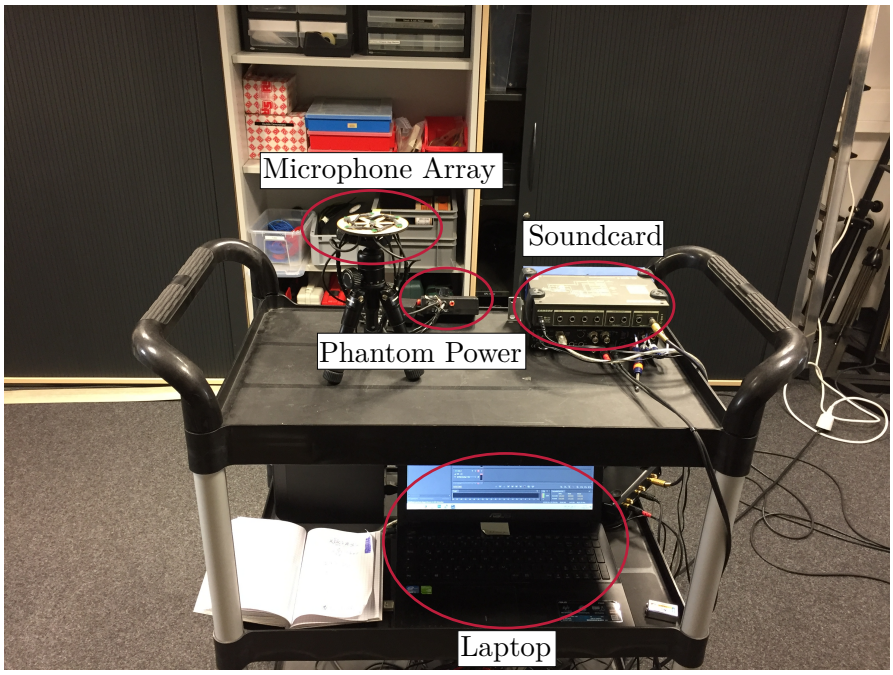


Figure A.1: Microphone signal recording setup

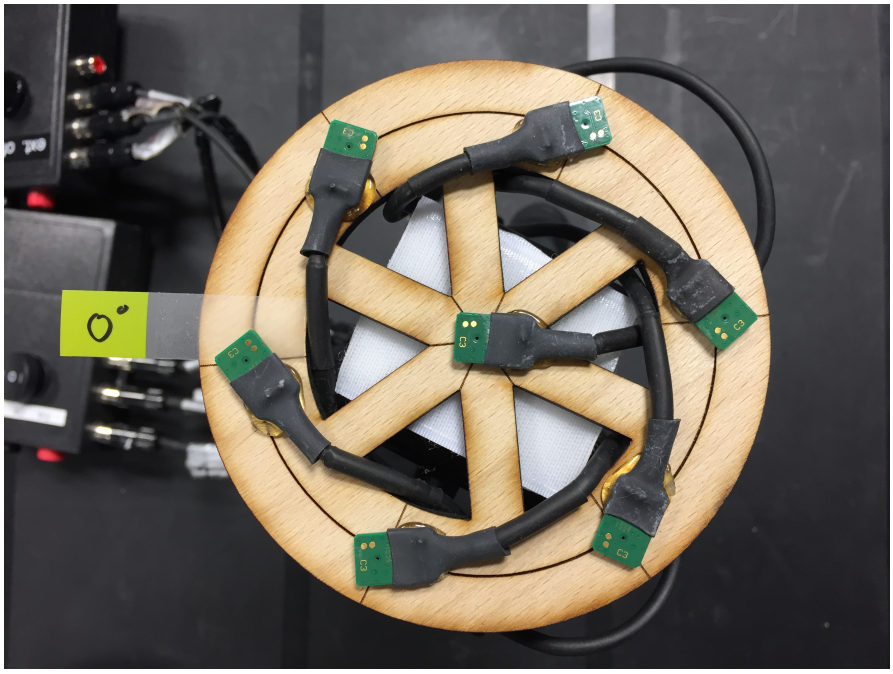


Figure A.2: Foto of the used 7 microphone Uniform circular array with center microphone

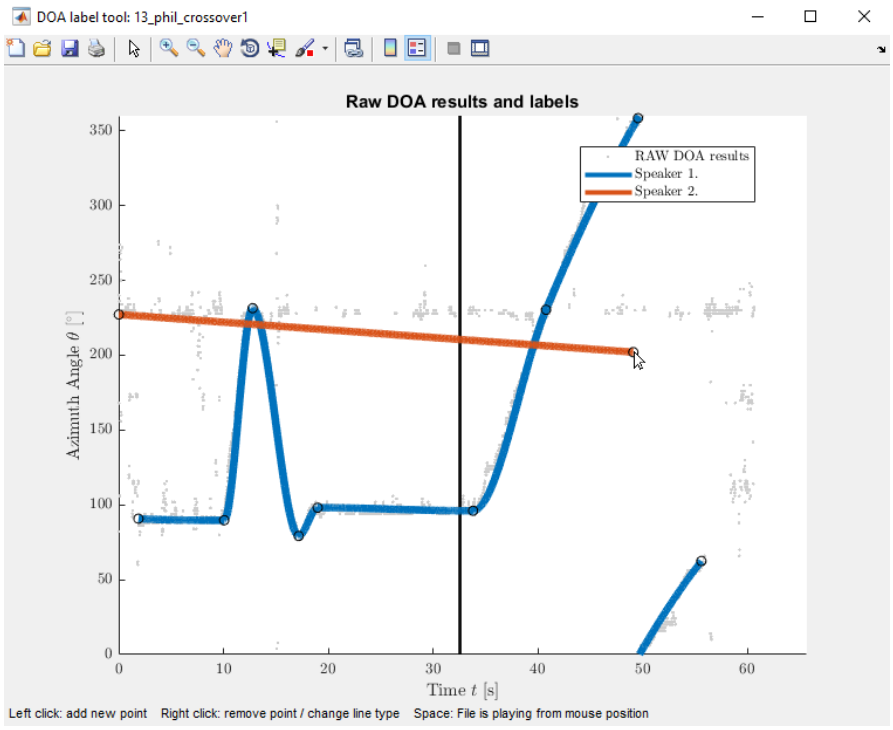


Figure A.3: Labeling Tool

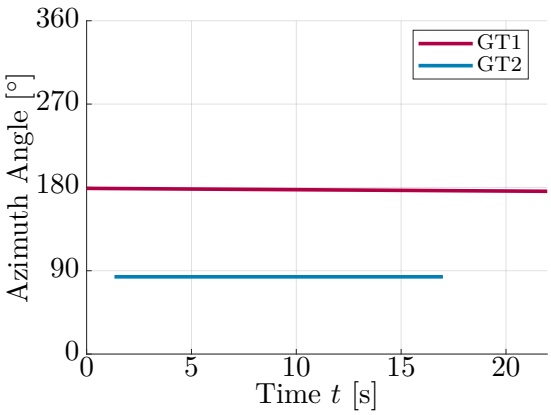


Figure A.4: Ground truth over time for scenario 1

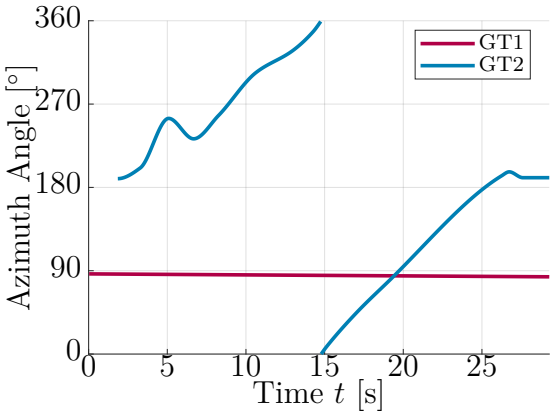


Figure A.5: Ground truth over time for scenario 2

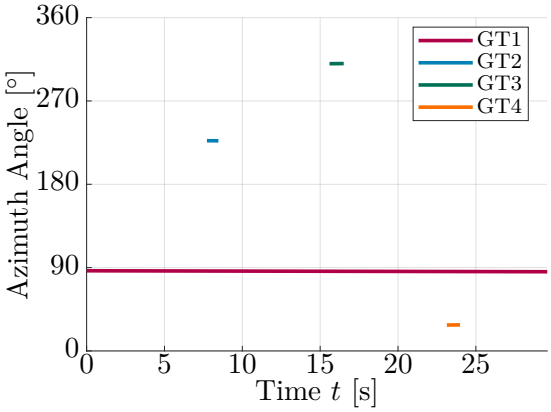


Figure A.6: Ground truth over time for scenario 3

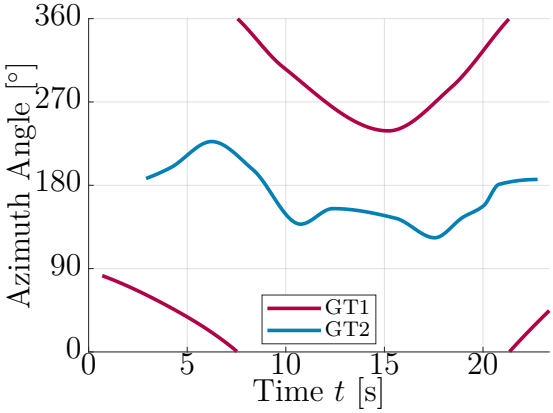


Figure A.7: Ground truth over time for scenario 4

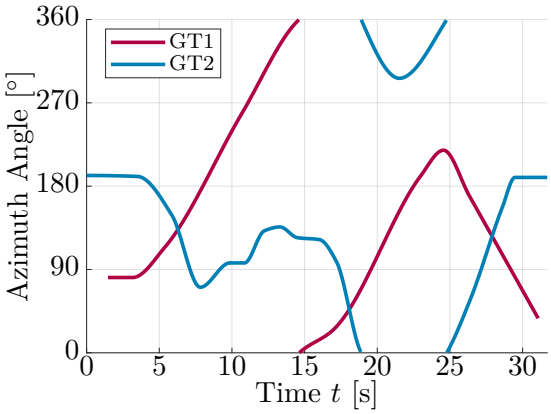


Figure A.8: Ground truth over time for scenario 5

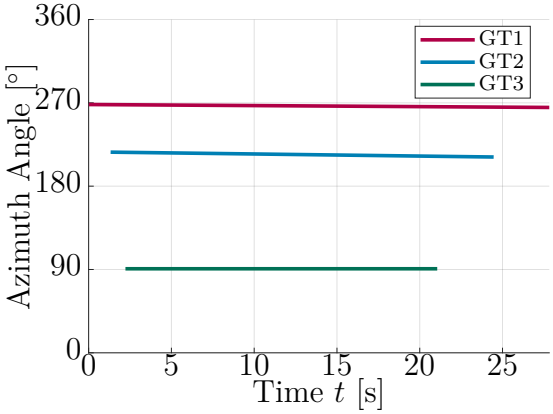


Figure A.9: Ground truth over time for scenario 6

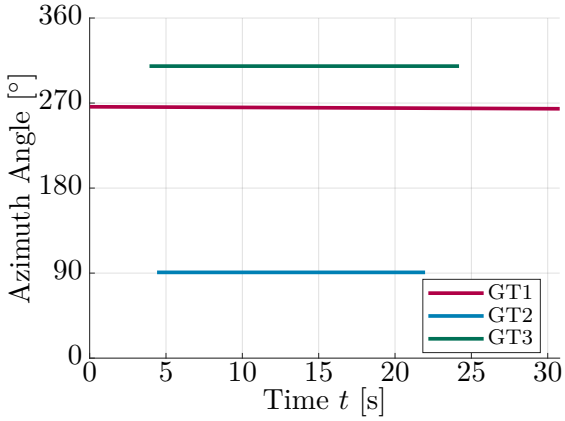


Figure A.10: Ground truth over time for scenario 7

#	Development Set
D1	One source moving around the array while speaking
D2	One source speaks successively from three positions
D3	Two sources from fixed positions speaking successively
D4	One fixed permanent music source, one moving source permanently speaking with crossover
D5	Two sources at 45° azimuth plane alternately speaking first then at the same time
D6	One source saying a wake-up-word moves and then asks a question
D7	One source saying a wake-up-word and then asks a question while music is playing

#	Evaluation Set
E1	Two fixed sources at 90° azimuth plane both simultaneously speaking
E2	One fixed source, one moving source speaking simultaneously
E3	One fixed source continuously speaking, one moving source speaking wake-up-words from different directions
E4	Two sources moving and speaking simultaneously, no crossovers
E5	Two sources moving and speaking simultaneously with crossovers
E6	Three static sources continuously speaking with minimum 80° azimuth plane distance
E7	Three static sources continuously speaking with minimum 40° azimuth plane distance

Table A.1: List of descriptions of all scenarios used for development and evaluation set

	Broadband SRP		Sub-band SRP	
Parameter	Delay and Sum	Capon	Delay and Sum	Capon
Circular buffer length N	16	16	257	257
Hypothesis threshold $\Gamma^{\text{add}}_{\text{class}}$	7	8	50	20
PDF Floor class p_{floor}	0.0001	0.0001	0.01	0.001
MAP-Adaption par. β	40	50	50	10
Min. mixing coef. $\Gamma^{\text{min}}_{\text{mix}}$	0.01	0.001	0.01	0.01
Variance σ^2_{const}	13^2	14^2	10^2	10^2
Initial TTL value $b^{\text{add}}_{\text{TTL}}$	30	30	50	50
TTL Threshold $\Gamma^{\text{TTL}}_{\text{thres}}$	0.01	0.01	0.01	0.01
TTL increase $b^{\text{inc}}_{\text{TTL}}$	30	30	8	8
TTL maximum $\Gamma^{\text{max}}_{\text{TTL}}$	20	20	15	15
Overlap Distance Γ_{overlap}	40	40	40	40
Aliasing Distance Γ_{al}	-	-	15	15

Table A.2: Parameters for the different developed multi-source classification algorithm

Parameter Name	Value
Buffer Length N	257
Number of initial sources K_{init}	5
Initial TTL $b^{\text{init}}_{\text{TTL}}$	2
TTL increase $b^{\text{inc}}_{\text{TTL}}$	12
TTL maximum $\Gamma^{\text{max}}_{\text{TTL}}$	20
Threshold Affiliations Γ_{affil}	70
Floor class p_{floor}	0.001
Overlap distance Γ_{overlap}	40

Table A.3: Parameters for Madhu's multi-source classification algorithm

	BB D&S	BB CAP	SB D&S	SD CAP	MADHU
E1	0	0	0	0	0
E2	0	0	0	0	0
E3	0	0	0	0	2
E4	0	0	0	0	0
E5	8	6	6	6	6
E6	0	0	0	0	0
E7	8	10	0	0	0
Total	16	16	6	6	8

Table A.4: ID changes for all scenarios and algorithms

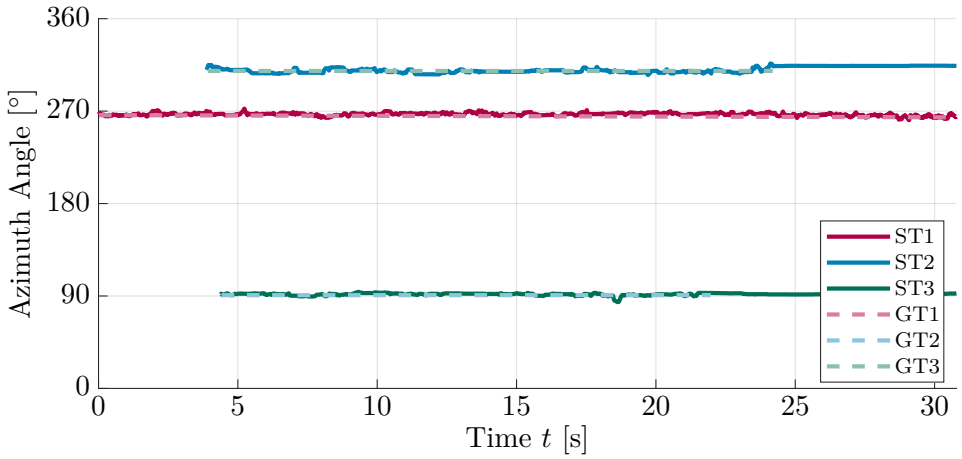
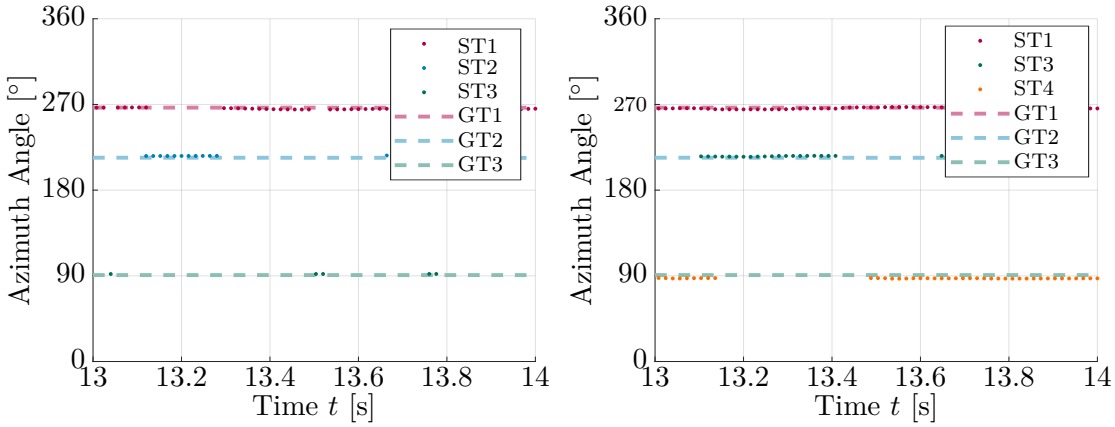


Figure A.11: Scenario E7, processed with the developed sub-band algorithm with a Capon beamformer



(a) Broadband processing

(b) Sub-band processing

Figure A.12: Active system track values over time are shown. Sub-band processing has more active system track values than the broadband processing.

	BB D&S	BB CAP	SB D&S	SD CAP	MADHU
E1	3.46°	2.86°	3.18°	3.35°	3.35°
E2	4.66°	4.09°	3.64°	3.71°	5.08°
E3	2.78°	3.64°	1.26°	2.05°	4.87°
E4	4.84°	5.09°	4.08°	3.26°	2.89°
E5	13.15°	14.37°	8.73°	9.68°	9.42°
E6	2.17°	2.17°	2.45°	2.85°	3.74°
E7	9.69°	8.32°	1.87°	1.99°	8.43°
Total	7.62°	7.95°	4.32°	4.92°	6.3°

Table A.5: Root mean square error values for all scenarios and algorithms

	BB D&S	BB CAP	SB D&S	SD CAP	MADHU
E1	1.77	2.02	0.09	0.58	0.05
E2	0.27	0.29	0.08	0.08	0.08
E3	0.28	0.22	0.15	0.16	0.15
E4	0.16	0.14	0.02	0.03	0
E5	0	0	0.13	0	0
E6	1.42	1.38	0.19	0.04	0.04
E7	0.14	0.19	0	0	0
Total	0.56	0.59	0.08	0.1	0.03

Table A.6: Initial detection lag in seconds for all scenarios and algorithms

	BB D&S	BB CAP	SB D&S	SD CAP	MADHU
E1	0	0	0	0	0
E2	0	0	0	0	0
E3	0	0	0	0	0
E4	0	0	0	0	0
E5	1	1	2	2	1
E6	0	0	0	0	0
E7	0	0	0	0	0
Total	1	1	2	2	1

Table A.7: Interrupt values for all scenarios and algorithms

	BB D&S	BB CAP	SB D&S	SD CAP	MADHU
E1	0	0	0	0	0.03
E2	0	0	0	0	0.43
E3	7.1	1.66	7.06	5.9	9.79
E4	0	0	0	0	0
E5	0	0	0	1.12	0.03
E6	0	0	0.67	0.43	1.7
E7	0	0	0.5	0	0.45
Total	7.1	1.66	8.22	7.46	12.43

Table A.8: False positive durations in seconds for all scenarios and algorithms