Graz University of Technology

Applying Beamforming on Pass-By Measurements of vehicles

Student Project

Author: Andreas Fuchs, 9773819

Date: Graz, July 28, 2015
Rev.: alpha 1.0

Contents

1 Introduction

Traffic noise evaluation is a highly standardized field. The evaluation of the noise emission of vehicles, tyres and road surfaces is spread over various standards and standardization committees (ISO, ÖNORM, UNECE). Therefore, measurements principles exist over years or decades; and seen from the signal processing perspective, it is mostly about time- and frequency-weighted levels. Some of these standards use in-situ measurement methods, i.e. under real-life conditions, where the main error sources are background noise or disturbing noise sources. For example, in the ISO 11819-1 [1] the pass-by levels of road vehicles are measured in real-life traffic. If the distance to the vehicle before or after the measured vehicle is to small, the measured level may be too high and the measurement must be discarded. By using new signal processing technologies, like beamforming, which are able to deal with unwanted noise sources, some of these measurement methods may be improved. But these signal processing technologies are on the other hand hardly standardized. They are very young and change fast, because they depend on available computing power. As mentioned before, they deal with unwanted noise sources by providing a flexible and controllable directivity. By focusing the array on the tested vehicle, the influence of disturbing noise sources on the measured level is diminished.

Inspired by the work of Püschel [2], who used a rather unusual array geometry in form of a Golomb ruler for beamforming on pass-by measurements together with a source separation algorithm, this project focused quickly on the influence of different linear array geometries. A quick way to get a deeper understanding was to design a MATLAB® GUI, where different input parameters can be changed on-the-fly and the resulting directivity pattern is visible within less than a second. But soon it became clear that only optimizing the geometry is not enough. The algorithm itself has a bigger influence on the directivity and instead of using a classic delay-and-sum beamformer, a least-squares optimized beamforming algorithm was developed. With this algorithm it is possible to capture the whole interesting sound source (car) and reduce the influence of all other sound sources including diffuse background noise. By doing so it should be possible to get a reliable level measurement, even without using a source separation algorithm. This provides two advantages: First, we do not have to think about how to do source separation, and second, because we are working in a standardized field, a more simple analysis chain introduces less error sources.

Finally the geometry and algorithm are optimized by numeric simulation. Afterwards, the result of this optimization is tested under real-life conditions by performing measurements. In controlled pass-by measurements the possible reduction of the influence of a close second car is examined as well as the ability to measure the correct level of a single pass-by.

The next chapter gives a short overview of the two fields pass-by measurements and beamforming to show the specific terminology, requirements, powers and weaknesses. In the third chapter two MATLAB GUIs (*Richt Tool DS* and *Richt Tool LS*) are presented and documented. Also the specific algorithm for the latter is derived and the numeric optimization is described. In Chapter 4 the measurements and results are presented.

2 Fundamentals

In this chapter we lay the theoretical groundwork for the application of beamforming on pass-by measurements to understand the challenges and difficulties in each field. Only then we can think about how these fields can be combined to find a satisfying solution. Section 2.1 gives an overview of pass-by measurements, which are mainly used to measure tyre and road noise. Section 2.2 contains the important parts of the topic of beamforming. In Section 2.3 the key factors for combining these two fields are presented.

2.1 Pass-By Measurements

There are various standards regarding traffic and road noise assessment, each specializing in a different aspect. The most important one for this project is the ISO 11819-1 [1]. This norm describes the Statistical Pass-By (SPB) method or measurement, which focuses on road surfaces. In contrast to this norm is the UNECE Regulation 117 [3], which describes the measurement method for tyre labels (similar to the labels for the energy efficiency on refrigerators all tyres sold in the EU after 2012 must have a label describing the sound emission, wet grip and rolling resistance of the tyre). In this regulation a measurement set-up is used, which is often described as Controlled Pass-By Method (CPB). This measurement method is specific to the combination of vehicle, tyre and road, whereas the SPB method evaluates the occurring traffic noise and, by using statistics, gives a single value to compare road surfaces.

2.1.1 Statistical Pass-By Method

To show the specialized application of beamforming on SPB measurements, the main aspects of the norm ISO 11819-1 are presented. The norm itself gives a good summary of the measuring principle:

> In the Statistical Pass-By (SPB) method, the maximum A-weighted sound pressure levels of a statistically significant number of individual vehicle pass-bys are measured at a specified road-side location together with the vehicle speeds. Each measured vehicle is classified into one of three vehicle categories: “cars”, “dual-axle heavy vehicles” and “multi-axle heavy vehicles”. Other vehicle categories are not used for this evaluation, since they do not provide any additional information regarding road surface influence.
>
> For each of three speed ranges defined in 3.3[1] as well as for each of the three vehicle categories, a nominated reference speed is given. Each individual pass-by level together with its vehicle speed is recorded, and a regression line of the maximum A-weighted sound pressure level versus the logarithm of speed is calculated for each vehicle category. From this line, the average maximum A-weighted sound pressure level is determined at the reference speed. This level is called the Vehicle Sound Level, L_{veh}. [1]

[1] *low* - 45 to 64 km/h, *medium* - 65 to 99 km/h and *high* - over 100 km/h average speed at which the traffic is operating.

The measurement site needs to meet certain requirements. The measured road must be at least straight and flat for 30 m (50 m for speed category *high*) on each side of the microphone position. The microphone is positioned 7.5 m perpendicular to the center of the test lane, at a height of 1.2 m. The omni-directional microphone is positioned towards the road. Additionally free-field conditions shall be met 25 m around the microphone, i.e. free from reflecting objects except the ground. The surface to test should exceed 3.75 m towards the microphone.

To reduce random errors, certain minimum numbers of vehicles must be measured: minimum 100 cars, minimum 30 dual-axle heavy vehicles and 30 multi-axle heavy vehicles, with a minimum total of 80 vehicles of the heavy vehicles category. As the traffic and vehicle composition is not controllable, the measurement depends on a good mix inside the vehicle categories. The other important aspect, which extends the measurement time, are the rules for the selection of valid measurements:

> Measurements shall only be taken on individual vehicle pass-bys which can be clearly distinguished acoustically from other traffic on the road. [1]

To ensure that no disturbance from other vehicles occurs, the A-weighted sound pressure level just before and after the test vehicle must be lower than 6 dB than the maximum level at the pass-by, see Fig. 2.1. It is easy to see that therefore measurements on a road with much traffic can be difficult, because there must be a significant distance between two vehicles. So to measure these roads, measurements are often performed at night, when it is often difficult to reach the minimum amount of vehicles per category. Furthermore, a significant other traffic composition than during the day may be found. At this point beamforming may present a solution by adaptively suppressing the disturbing vehicles.

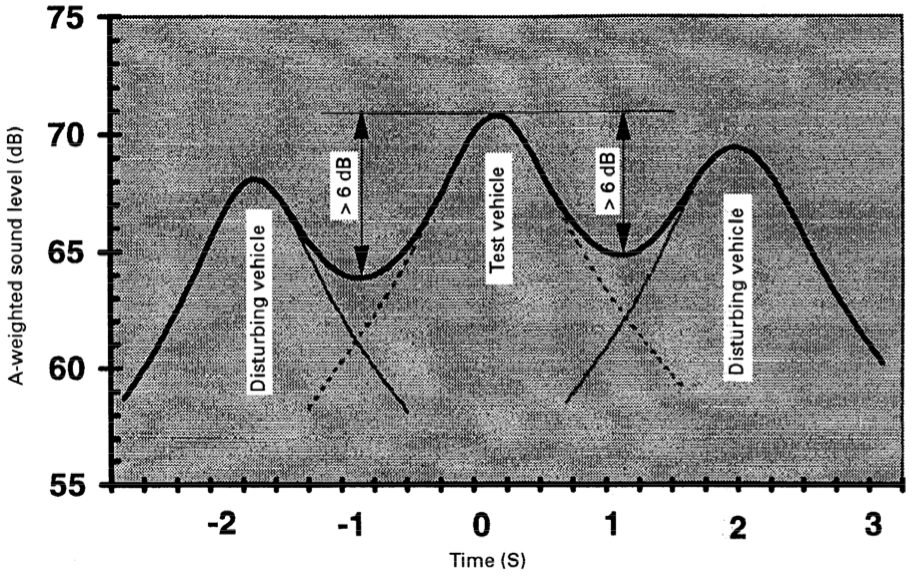

Figure 2.1: Required signal-to-noise ratio for an individual vehicle pass-by [1].

Additionally there should be no other disturbing sound sources, such as vehicles passing-by on the other lanes or extraordinary sound emissions from the test vehicle (fault exhaust system, audible warning signals). Also the test vehicle should drive with a constant speed on a straight line on the test lane.

The maximum sound pressure level of each vehicle is determined by using a frequency weighting A and a time weighting *FAST* ($L_{Af,\max}$), as described in [4]. Together with the measured speed of the vehicle v, a linear regression (least-squares-method) over the pairs of $L_{Af,\max}$ and $\log(v)$ for all vehicles in this category is performed. This regression line is then evaluated at

the reference speed to get the Vehicle Sound Level L_{veh} for this category and furthermore to calculate the Statistical Pass-By Index (SPBI) with a weighted power addition. This SPBI can then be used as a single value for comparing road surfaces.

The coefficient of determination R^2 won't show a strong correlation for the regression, because of the different sound emissions of various vehicles in one category. But it is possible to get an *average* noise level of the combination of surface, speed and vehicle type [5], which can be used for noise emission and propagation simulations.

2.1.2 Controlled Pass-By Method

In the UNECE Regulation 117 [3] the method is referred to as coast-by test method, but the term controlled pass-by method (CPB-method) [5] is also common and used for better distinction to the SPB method. The measurement procedure is nearly the same as the SPB method, including the set-up, measurement of sound and speed, and performing the regression. The difference between these two methods is, that the measurement is performed on a controllable test site without any disturbing sound sources and for one specific vehicle for different speeds. Then it is possible to rate the specific tyre-road noise emission for this vehicle, plus the coefficient of determination R^2 for the regression can reach values above 0.98.

The UNECE Regulation 117 focuses on assessing the emission of tyres. Therefore, the surface of the test site is strictly specified. The measured speed range is from 70 to 90 km/h for car tyres; the numerous measurements are again used for linear regression between $L_{Af,\max}$ and $\log(v)$ to get the sound emission at 80 km/h. At this speed the noise emitted by the tyres is dominant for modern vehicles [5], so no further requirements are given for the test vehicle. The background noise at the test site must be at least 10 dB lower than the maximum sound pressure level.

The controlled pass-by measurement method can be used for various measurement goals, and here another opportunity for beamforming can be found. If the background noise is diffuse, then beamforming can improve the signal-to-noise ratio and measurements with electric vehicles at low speeds are possible. This is because it is very hard to find a test site with a background noise 10 dB below the maximum sound pressure level emitted by a electric car at low speeds, which may be inaudible in urban areas.

2.2 Beamforming

Microphone arrays give the ability to listen to sound coming from one direction and suppress disturbing sounds from other directions. It is also possible to detect the direction of an incoming sound, which is called *source localization*. Generally speaking, sensor arrays are already common in different applications like ballistic missile detection, ground- and ship-based radars or cellphone towers, and were already used in the second World War [6]. But in comparison to the capturing of electromagnetic waves, the capturing of sound reveals some challenges. As it will be discussed later, arrays make extensive use of the propagation of the sound wave to the single sensors or microphones; therefore the performance is critically dependent on wavelength differences and, as a result, on the frequency. Radar signals have in regard to their operation frequency a very small bandwidth (a few percent) and will therefore always work in their best configuration. Sound sources work in a range of at least seven octaves (63 Hz to 8 kHz). Furthermore, different noise sources have to be considered, mainly the equivalent noise level of the microphones and reverberation [6].

2.2.1 Terms and Parameters

In this section the basic principle of beamforming is derived, in order to understand the specific terminology and parameters.

Basic Principle

Beamforming exploits the propagation of sound waves between the microphones to retrieve geometric information about the sound field. Therefore, it is important to use a suitable model to describe the sound field. In this project all the algorithms use the near-field model, i.e. spherical waves, which is more accurate and is also more suitable for numeric simulations [6]. The simpler far-field model uses planar waves, which gives a good approximation for sound sources that are far away, but in the used algorithms the simplification is not needed. The propagation $D_m(f)$ of a spherical wave originating from a point $\boldsymbol{x}$ to each microphone m is for a frequency f

$$D_m(f, \boldsymbol{x}) = \frac{1}{||\boldsymbol{x} - \boldsymbol{p}_m||} \mathrm{e}^{-j2\pi f \frac{||\boldsymbol{x}-\boldsymbol{p}_m||}{c}} , \tag{2.1}$$

where c is the speed of sound and $\boldsymbol{p}_m$ is the position of each microphone m. The air temperature T has a relevant influence on the speed of sound and must therefore be measured at the test site and considered in the algorithm with $c_{air} = 20.1\sqrt{T/\mathrm{K}}\,\mathrm{m/s}$ [7]. If we assume that the microphones have a flat frequency response in the desired frequency range and an omni-directional directivity pattern, the microphone signals are

$$X_m(f, \boldsymbol{x}) = D_m(f, \boldsymbol{x})S(f) + N_m(f) = \frac{1}{||\boldsymbol{x} - \boldsymbol{p}_m||} \mathrm{e}^{-j2\pi f \frac{||\boldsymbol{x}-\boldsymbol{p}_m||}{c}} S(f) + N_m(f)\,. \tag{2.2}$$

The source signal $S(f)$ is delayed and attenuated by the sound propagation model $D_m(f, \boldsymbol{x})$, $N_m(f)$ is the noise on each microphone. Finally all microphone signals are combined to get the output signal $Y(f)$ of the beamformer

$$Y(f) = \sum_{m=1}^{N} W_m(f, \boldsymbol{b}) X_m(f, \boldsymbol{x}) = \boldsymbol{W}(f, \boldsymbol{b})^\top \boldsymbol{X}(f, \boldsymbol{x})\,, \tag{2.3}$$

where N is the number of used microphones; $W_m(f, \boldsymbol{b})$ describes the (linear) filter (*delay*) used on each microphone channel before they are summed. $W_m(f, \boldsymbol{b})$ depends on the *beamsteering*, i.e. the direction or point, represented by vector $\boldsymbol{b}$, focused by the array (the position of the sound sources the array wants to capture). The vectors $\boldsymbol{W}(f, \boldsymbol{b})$ $(N \times 1)$ and $\boldsymbol{X}(f)$ $(N \times 1)$ contain the respective microphone dependent terms. The dependency on $\boldsymbol{b}$ for the filter coefficients $\boldsymbol{W}(f, \boldsymbol{b})$ is from now on omitted for better readability. Different beamforming algorithms use different filter coefficients in $\boldsymbol{W}(f)$, but all of them use eq. (2.3). So the output signal $Y(f)$ is dependent on the beamsteering position $\boldsymbol{b}$, the source position $\boldsymbol{x}$, the microphone positions $\boldsymbol{p}_m$ and the filter coefficients $\boldsymbol{W}(f)$ beside the obvious source signal and noise. Fig. 2.2 shows this basic principle.

Spatial Aliasing

In order to use beamforming, we must have unique phase differences for different directions. Therefore there is a strict relationship between the usable frequency range and the smallest distance between two microphones d_{min}:

$$f < f_{max} = \frac{c}{2d_{min}}\,. \tag{2.4}$$

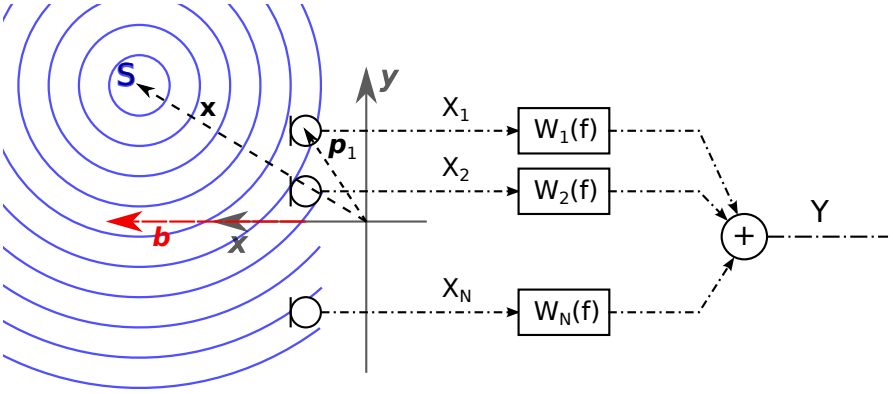

Figure 2.2: Basic principle of beamforming: The spherical wave of the source S is captured by the N microphones. The microphone signals X_m are filtered with the respective coefficients W_m and summed together.

Again, the speed of sound c is used, and therefore the highest usable frequency of an array depends on the air temperature. This relationship is equivalent to the Nyquist-Shannon sampling theorem in signal processing. If only one half wave-length can fit between the smallest distance between two microphones, no aliasing occurs. But it is also important to notice, that close to f_{max} the phase differences are at the maximum (up to $\pm\pi$) and the array will perform best. At lower frequencies the phase difference between the sensors is less due to the larger wavelength. Therefore, we get an omni-directional behaviour at lower frequencies and noise will have a greater impact. So there is not a hard limit for a lower frequency, but an array, which also has microphones far away from each other, will perform better at lower frequencies.

Array Types and Cone of Confusion

Not only the distance between the microphones, but also the structure of the array is important. If we want to distinguish between all solid angles (azimuth & elevation), the array structure needs to be three dimensional, so at least four microphones have to be used. But in many situations it is not necessary to distinguish all solid angles, and one can use much simpler array structures. The most common types are:

Linear Arrays: The microphones are positioned along a line (minimum 2). Sound sources which are rotationally symmetric around this line can not be distinguished. Therefore, they are on the *Cone of Confusion*[2], and the array can not detect whether the sound source is in front, above, back, or below. If the array is parallel to the ground, the array can only be steered in horizontal direction. If the sound source is in line with the array, the term *endfire* mode is used. If the sound source is most of the time perpendicular to the array, it is a *broadside* array [6].

Planar Arrays: All microphones are placed in one plane (minimum 3). If the array is perpendicular to the ground, the array can be steered horizontally and vertically, but it can not distinguish between front and back. A special case of planar arrays are *circular* and *spiral* arrays.

Another important geometry parameter is the spacing between the sensors. This can be equidistant or arbitrary, which will have different effects on the frequency range and directivity of the array. This is further discussed in Chapter 3.

[2] Actually the sound localization with two ears with Interaural Time Differences (ITD) can be seen as beamforming. There is also a Cone of Confusion on the sagital plane.

Directivity Pattern and Directivity Index

The most important performance parameters of a microphone array are the directivity pattern and the directivity index. The directivity pattern is the complex phasor $B(f, \boldsymbol{x})$ per frequency for every relevant source position in $\boldsymbol{x}$, but it is mostly presented only as the magnitude.

$$B(f, \boldsymbol{x}) = \boldsymbol{W}(f)^\top \boldsymbol{D}(f, \boldsymbol{x}) \tag{2.5}$$

The terms of the vector $\boldsymbol{D}$ are the physical model for the propagation from the source positions $\boldsymbol{x}$ (as in eq. (2.1)) to every microphone. If only the direction of the source without the distance (i.e. all source points are on a sphere/circle) is used, it is possible to plot the magnitude of the directivity pattern either in one plane over all frequencies in a color plot, or a three dimensional volumetric plot for one frequency. All directivity pattern plots are for one specific beamsteering position $\boldsymbol{b}$. The directivity index known from microphones can be easily calculated [8]:

$$DI(f) = 10\log_{10} \frac{4\pi}{\int_0^{2\pi} \int_0^{\pi} |B(f, \boldsymbol{x}(\theta, \varphi))|^2 \sin\varphi \, \mathrm{d}\theta \, \mathrm{d}\varphi} \tag{2.6}$$

The directivity index can also be seen as the noise suppression for isotropic noise, also known as the *ambient noise gain*.

Uncorrelated Noise Gain

Beside the ambient noise which is correlated among the microphones, there are other noise sources we have to consider and which are uncorrelated on the single microphones. The microphones self-noise and the quantization noise are the most intuitive sources. If we assume that we have identical microphones, the uncorrelated noise gain $G_I(f)$ is defined as

$$G_I(f) = 10\log_{10} \boldsymbol{W}(f)^H \boldsymbol{W}(f) \,. \tag{2.7}$$

Tashev showed that manufacturing tolerances such as uncertainty in microphone position, phase, magnitude and delay differences in the recording chain from microphones, pre-amplifiers and ADCs can be modeled as uncorrelated noise [6]. For a *robust* beamformer its qualities are independent on these noise sources. On the one hand it is important to minimize the manufacturing tolerances as good as possible, on the other hand the beamformer must not be sensitive to them, i.e. the noise gain has to be small.

2.2.2 Algorithms

The beamforming algorithm calculates the filter coefficients $\boldsymbol{W}(f)$. The first group of algorithms are *time-invariant* beamformers, i.e. the coefficients are independent of the input data. Of course, if the beam is steered towards another direction, the coefficients have to be changed. The second group of algorithms are *adaptive* beamformers, which are data-dependent; these beamformers can, e.g., adapt to a different spectral noise power distribution. For our application time-invariant beamformers are sufficient, which will be discussed later in section 2.3.

Delay-and-sum Beamformer

The most intuitive beamforming algorithm is the delay-and-sum beamformer. The filter coefficients are delays, which compensate the time difference of the propagation to the single microphones in order to create maximal constructive interference for signals coming from the

steered direction. Therefore the coefficients are (compared to eq. (2.1))

$$W_m(f, \boldsymbol{b}) = \frac{1}{N} e^{j2\pi f \frac{||\boldsymbol{x}-\boldsymbol{p}_m||}{c}} . \tag{2.8}$$

The weights are divided by the number of microphones N to get unit gain compared to a single microphone. This beamformer focuses on perfect reconstruction in the steered direction (point source), all other directions have a smaller gain because of interference. The directivity pattern is shown in Fig. 2.3.

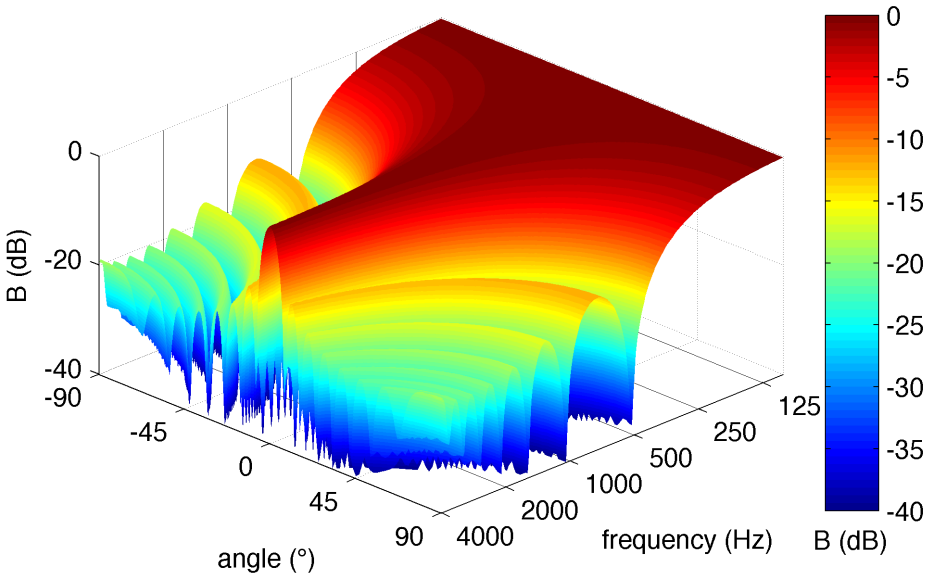

Figure 2.3: Directivity pattern of a delay-and-sum beamformer with 13 microphones (equidistant linear array with 60 cm length). The array is steered at 0°.

Because this beamformer explicitly uses only delays, the magnitude of all coefficients in $\boldsymbol{W}(f)$ is $1/N$, therefore after eq. (2.7) the uncorrelated noise gain is $G_I(f) = -10\log N$. The delay-and-sum beamformer is most robust to manufacturing tolerances [6].

In Chapter 3 approaches to optimize the main lobe of the delay-and-sum beamformer by using weights on the filter coefficients are presented. The general idea is to use the microphones in the center only for higher frequencies to widen the narrower main lobe there.

Directivity Pattern Synthesis

The goal of directivity pattern synthesis, as shown in [6], is to optimize the filter coefficients $\boldsymbol{W}(f)$ for a specified pattern. The delay-and-sum beamformer only tries to provide unit gain in the steering direction, so now we are going to define the gain in all or at least many directions. Therefore, we define a vector $\boldsymbol{\Delta}$, which consists of all gains of the M predefined directions. In the simplest form we use a gain of one for the directions of interest, and a zero for directions to be suppressed. The M directions should be equally distributed over the whole half circle and M should be much higher than the number of microphones N. Then the following equation applies:

$$\boldsymbol{D}\boldsymbol{W} = \boldsymbol{\Delta} \tag{2.9}$$

$\boldsymbol{D}$ is a $M \times N$ matrix, the rows of this matrix are filled with the sound propagation model to the microphones D_m for one direction. The specified gain of this direction is in the corresponding

row of vector $\boldsymbol{\Delta}$ ($M \times 1$). The minimum mean squared error (MMSE) solution of this overdetermined equation system is calculated via the Moore–Penrose pseudoinverse:

$$\hat{\boldsymbol{W}} = (\boldsymbol{D}^H \boldsymbol{D})^{-1} \boldsymbol{D}^H \boldsymbol{\Delta} \tag{2.10}$$

Finally we have to normalize the solution $\hat{\boldsymbol{W}}$, so we get unit gain in the listening direction $\boldsymbol{D_b}$:

$$\boldsymbol{W_{opt}} = \frac{\hat{\boldsymbol{W}}}{\boldsymbol{D_b}\hat{\boldsymbol{W}}} \tag{2.11}$$

This only shows the theoretical background of pattern synthesis. In practice, noise constraints must be considered to design a robust beamformer. A practical implementation will be presented in section 3.3.1.

2.3 Approaching the Combination

The last two sections showed the different fields of pass-by measurements and beamforming. But how can beamforming be helpful in pass-by measurements? The main goal is to capture the sound level of the test vehicle as good as possible, while not be disturbed by other noise sources or background noise. The following aspects have to be considered:

Frequency Range: Due to the A-Weighting and typical spectral distribution of traffic noise [9], a frequency range from 125 Hz up to 8 kHz is more than sufficient, because only the overall level is analyzed.

Noise Spectrum: The noise spectrum of the background noise may be measured, but the spectrum of a disturbing vehicle is unknown.

Vehicles: To show the main principle, this project concentrates on cars with a maximal length of $5 - 6$ m.

Sound Sources: A car at close range can not be modeled as a single, omni-directional point source. At the side of the vehicle the main sound sources are the four tyres, which may not be radiating omni-directional due to the *horn effect* between tyre and street [5]. Therefore, the maximum sound pressure may not occur, when the car is directly in front of the microphone.

Damping: It is enough to reach a damping of 20 dB of a disturbing sound source. If we sum the level of two (incoherent) sound sources with 70 dB and 50 dB, we get an overall level of 70.04 dB[3], so the influence of the second source is negligible.

Because the noise spectrum is unknown, this project will only use time-invariant beamformers. Also the beam should be steerable, to focus the array on the tested vehicle and suppress disturbing vehicles. Because of the directivity, the signal-to-noise ratio in regard to diffuse background noise will also improve. As we are not interested in the vertical plane, a linear array is sufficient. Of course the cone of confusion will give us a directivity to the back, so we have to assume that no loud noise sources are in this particular area. The frequency range, length of the vehicles and required damping have to be considered when designing the array and algorithm.

The next chapter will focus on the effects of array geometry and algorithms on the directivity pattern by simulating the directivity pattern on the street. If we know the shape of the directivity pattern, a suitable analysis procedure can be found.

[3] $L_{\text{all}} = 10 \lg \left\{ \sum_{i=1}^{N} 10^{L_i/10} \right\}$ [7].

3 Effects of Array Geometry and Algorithms on Directivity Patterns (MATLAB GUI)

This chapter focuses on the influence of the array geometry and the beamforming algorithm on the directivity pattern. For the delay-and-sum beamformer the directivity pattern depends on frequency, steering direction and array geometry. To identify the effects of different parameters on the directivity pattern, two MATLAB® R2010b[4] GUIs were created, which make the effects of a change in the parameter set immediately visible. Not only can the mechanisms in beamforming be better understood, ranges for optimal parameter sets can also be found. The GUI *Richt Tool DS* uses a delay-and-sum beamformer algorithm and is described in section 3.2, the *Richt Tool LS* uses a synthesized directivity pattern with a least-squares algorithm (section 3.3). The latter includes also various performance parameters for numerical optimization and evaluation, which is done in section 3.4. In section 3.1 the common basic functionality for both GUIs is described.

At the end of this chapter we are able to design an adapted array for beamforming and have got an algorithm, which can capture a whole car and still provide enough damping for disturbing noise sources. Then we do not need any extra source separation algorithm, because the sound sources are already separated by the beamforming algorithm itself. Instead we can directly analyze the beamformer output signal with the standard analysis procedures, i.e. calculate the $L_{Af,\max}$ for the pass-by.

3.1 Basic Functionality

Although the *Richt Tools LS* is much more complex, both GUIs share basic functionalitiy, which is described here. The main goal of the the MATLAB GUIs is to show the resulting directivity pattern along the street for different algorithms, geometries and other parameters. For performance reasons, the GUIs calculate the directivity pattern for 150 single frequencies logarithmically spaced between 63 Hz and 10 kHz. They use a spherical wave propagation model as described in eq. (2.1) and apply beamforming for each frequency in the frequency domain. Additionally a slider is placed on the GUI to move the steering direction along the street, Fig. 3.3 shows a screenshot of the *Richt Tool DS* GUI on page 16. This position is also shown in the directivity pattern and overall level plots as a horizontal, dotted line. Also the maximum frequency is shown in the directivity pattern as a vertical solid black line. If parameters can be entered, there is no verification if the entered parameters are valid (e.g. a positive integer number of microphones).

Directivity Pattern

Usually the directivity pattern is plotted for source positions on a circle or sphere around the array to show the influence of the angle on the captured level (see eq. (2.5)). In this project the angle around the array is not as important as the corresponding position on the street.

[4] The GUIs have been created with MATLAB R2010b on Mac OS X 10.9 and have been tested also with R2014a on Mac OS X and R2009b on Windows 7. Different versions of MATLAB may scale and render the GUI differently, but the underlying algorithms work with all tested versions.

Therefore the source positions in vector $\boldsymbol{x}$ are positioned along the virtual street. As we are only interested in the magnitude, the level of the directivity pattern can be encoded in a color plot with the frequency and the position on the street as axes.

Overall Level

For the evaluation of the SPB method, only the A-weighted, overall sound pressure level is relevant. Therefore, it is possible to switch the directivity pattern between an A-weighting and Z-weighting (i.e. no weighting). This will be rather helpful, because for the overall level the less directivity at lower frequencies is due to the damping of the A-weighting compensated. This can be easily seen in the plotted overall level in the sreenshot of the GUI *Richt Tool DS* in Fig. 3.3. The A-weighted overall level shows more directivity than the not weighted overall level (Z-weighting). The overall level is calculated as the energetic sum over the single frequencies (Parsevals theorem).

Geometry

Three geometry parameters can be entered. The type of the arrangement of the microphones (for a linear array), the number of microphones N and the total length of the array L. Four types of arrangement are defined (see Fig. 3.1):

equidistant: The distance d between two microphones is constant. As it can be easily seen, $d = \frac{L}{N-1} = d_{min}$ and, with eq. (2.4), $f_{max} = \frac{c(N-1)}{2L}$. So the maximum frequency will increase with more microphones and a smaller array.

Golomb: No pair of marks, which have integer positions, have the same distance to each other on a Golomb ruler. The number of marks defines the order and equals the number of microphones N in our set-up. For a Golomb ruler of order four the positions are $0, 1, 4, 6$, which is considered *perfect* (because all distances up to length six can be measured with this ruler) and *optimal* (because no shorter Golomb ruler with the same order exists). *Perfect* rulers only exist to order 4 [10]. Püschel positioned the microphones at the Golomb ruler marks, which should give an "optimal solution for the correlation of the microphone signals while measuring broadband signals" [2]. The maximum frequency is derived from d_{min} as in eq. (2.4), which depends on the order of the Golomb ruler (equals N) and the length of the array L.

symmetric Golomb: The Golomb ruler is highly asymmetric, which may or may not lead to a dependency on the driving direction. Therefore, a so called *symmetric Golomb* positioning was derived, where two Golomb rulers with the round-up half order $\lceil N/2 \rceil$ are symmetrically combined.

nested: Arrays with more sensors in the center are very common and are able to increase the frequency range of microphone arrays. These are often called *nested* arrays. In this project a function to define the positions of the array was used; a cubic term in combination with a linear term spreads the microphones apart from each other. For $i = 1, .., N$ equidistant positions x_i in the interval $[-10, 10]$ ($x_1 = -10$, $x_N = 10$), the new positions y_i were calculated with

$$y_i = x_i^3 + a x_i \tag{3.1}$$

These positions get normalized to match the total length L. The factor a can be described as *spreading factor*. A small spreading factor will result in a strong concentration of microphones in the center, where for $a \to \infty$ the array will be equidistant, see Fig. 3.2.

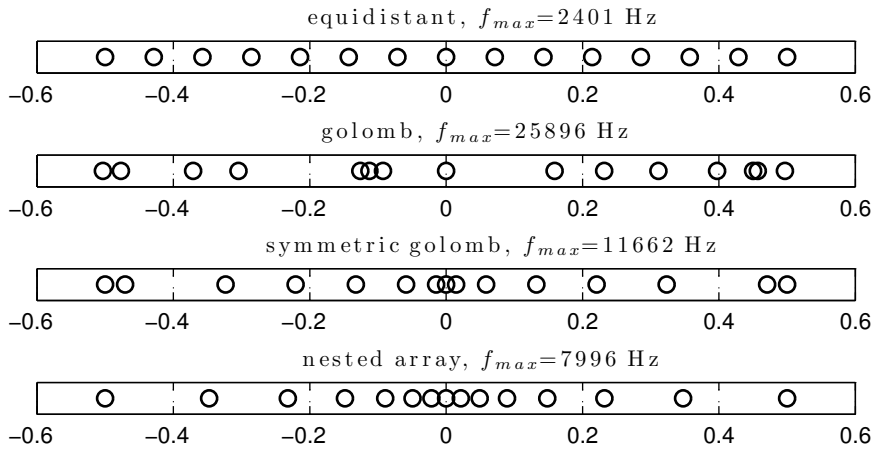

Figure 3.1: Used array geometries for the Richt Tool *MATLAB GUIs with $N = 15$ microphones and a total length of $L = 1\,m$. The nested array uses a factor $a = 40$.*

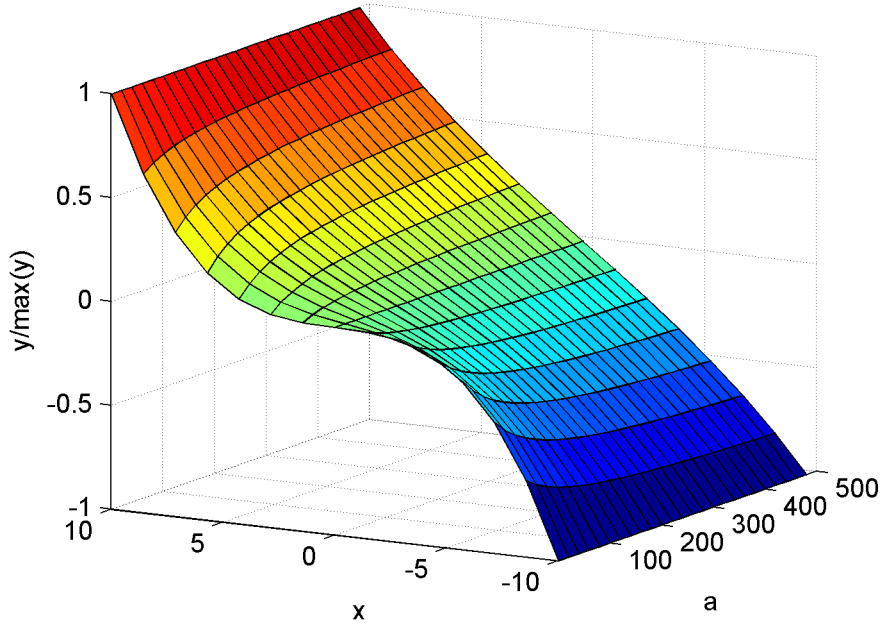

Figure 3.2: Visualization of eq. (3.1) for different spreading factors a. The total length is normalized to be independent from a.

Mismatch in Placement

To simulate a not perfect positioning of the microphones, a *mismatch in placement* can be specified. To each coordinate of each microphone in a three dimensional Cartesian coordinate system a zero-mean, normally distributed random number with a user-specified standard deviation is added. This standard deviation is controlled by the input parameter *mismatch in placement* and is always the same for all three coordinates. With this parameter the robustness against a mismatch in placement for the microphones can be examined.

3.2 Richt Tool DS

The *Richt Tool DS* contains the delay-and-sum beamformer algorithm. Fig. 3.3 shows a screenshot of the GUI. On the left side the input parameters are placed, next to a vertical slider, which controls the steering position on the street. This slider uses the same axis as the two plots next to it, the overall level and the frequency dependent directivity pattern. The frequency weighting of the latter can be controlled via the check-box *A-weighting*. At the bottom a frequency dependent Hann-window can be controlled, which is described in section 3.2.1. The positions along the street are calculated from −30 to 30 m in 20 cm steps; the overall levels are calculated over the whole frequency range, regardless of f_{max}.

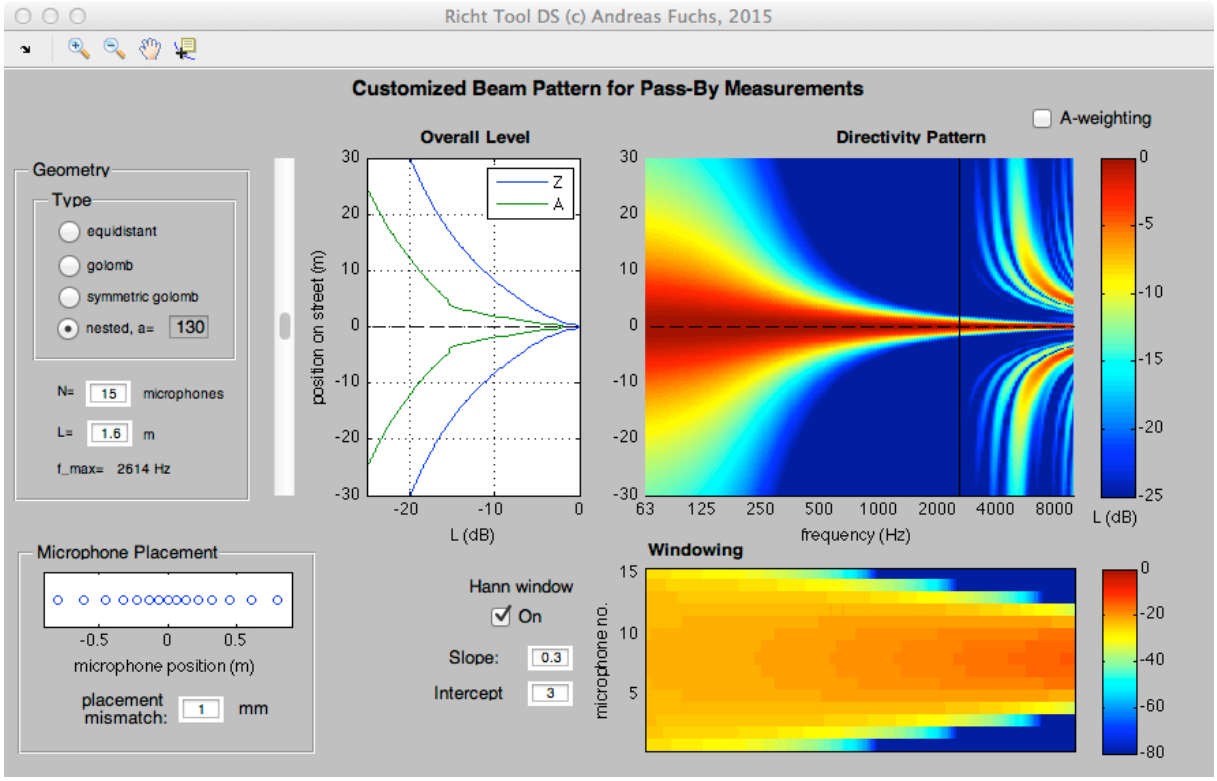

Figure 3.3: Matlab GUI Richt Tool DS.

The *RichtTool DS* simulates sound sources along the virtual street with spherical propagation. As we do not want to compensate the propagation damping for farther away positions and are only interested in the damping of the beamforming, the coefficients for the delay-and-sum beamformer (eq. (2.8)) are compensated for the propagation between the microphone positions and the location on the street in front of the array. The important part of the calculation of the beamforming coefficients is shown in Listing 3.1.

```
1  for mm=1:N
2      s = norm(l(mm,:)-beam_pos);
3      W(mm,:) =1./N*norm([0 7.5 0]+l(mm,:)).*exp(1i.*2*pi.*f.*s/343);
4  end
```

Listing 3.1: Calculation of the beamforming coefficients for the delay-and-sum beamformer. The microphone positions are in matrix l ($N \times 3$) and are centered on the x-axis. beam_pos is a vector (1×3) with the steering position.

The main idea of the MATLAB GUIs is to see quickly the influence of the parameters, therefore only specific examples are given in this report. In Fig. 3.4 the resulting directivity patterns on the street are shown to see the influence of different spacings. The length of the array is adapted, so that all spacings have the same maximum frequency $f_{max} = 8\,\text{kHz}$. The data is calculated by the *Richt Tool DS* with no microphone displacement. The influence of the microphone displacement

is very small, due to the robustness of the delay-and-sum beamformer. All spacings show a very narrow beam at high frequencies. The equidistant array shows a very small useful frequency range; the visible damping below 250 Hz is due to the higher distance of the source position and not caused by beamforming. The spacings based on the Golomb ruler show a much wider frequency range, but with a narrow beam. The nested array is a compromise, which is scalable with the spreading factor a.

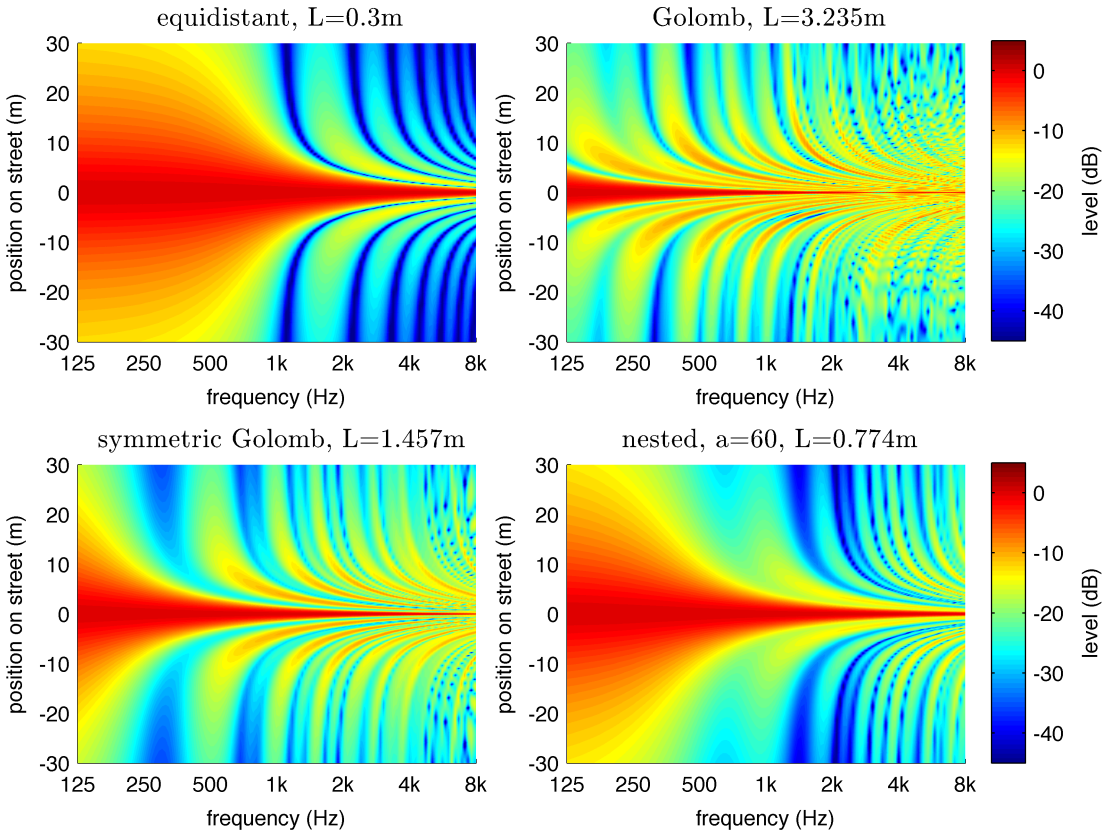

Figure 3.4: Directivity pattern along the street for different spacings of the microphones with the delay-and-sum beamformer. The length of the array for the different spacings is adjusted, so that $f_{max} = 8\,kHz$.

3.2.1 Hann Windowing

With the *Richt Tool DS* it can be easily seen that more microphones will narrow the beam; also the beam in Fig. 3.4 is narrower for higher frequencies. Therefore, it may be possible to equalize the beam width, if we use less microphones for higher frequencies. This is done by applying a Hann window over the microphones. The length of the window depends on the frequency and for low frequencies all microphones should be inside the window; for higher frequencies only the microphones in the center of the array should be inside the window.

The length of the window is calculated with

$$L_w = 2^{-\mathrm{ld}(f)\cdot slope + intercept} L \tag{3.2}$$

with the two adjustable parameters *slope* and *intercept*; afterwards the specific microphone gains are calculated according to the Hann window function. The above function for the length of the Hann window normalized to L is shown in Fig. 3.5 for different *slopes* and *intercepts*. Because

slope and *intercept* are defined inside the power of 2, the influence of these two is clearer in the logarithmic display on the left side. On the right side the ordinate is linear, and the effects are more visible. For example, with a *slope* of 0.5 and an *intercept* of 5 (red line), the Hann window is more than 3 times bigger for 63 Hz than at 8 kHz; the resulting window and therefore the specific gains for the microphones are shown in Fig. 3.6. A similar display with a color-coded plot over the microphones is used in the bottom right corner of the GUI, which shows the resulting damping caused by the window in dB, see Fig. 3.3.

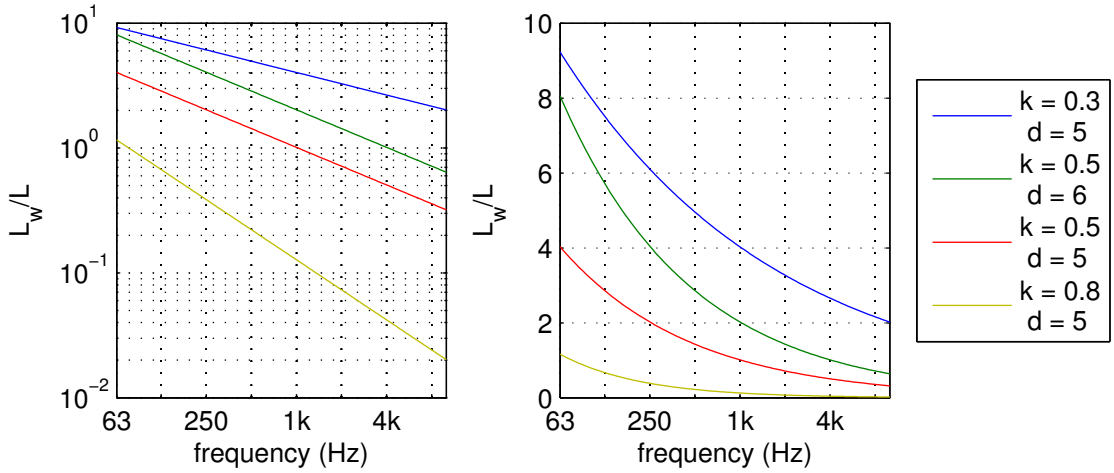

Figure 3.5: Visualization of eq. (3.2)*. For short display, k is used for slope and d for intercept.*

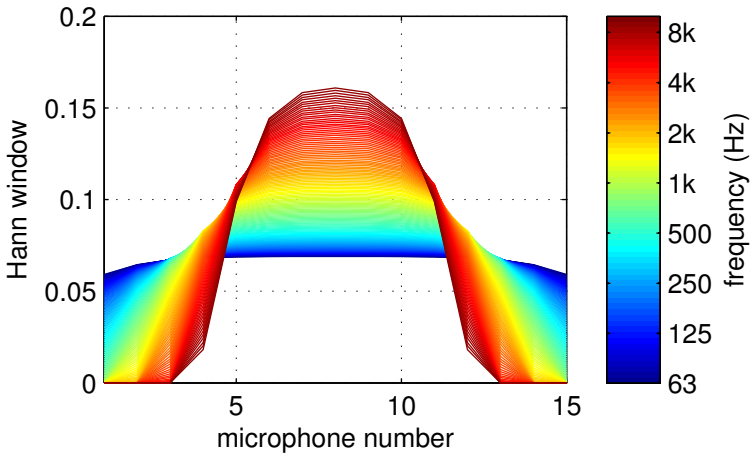

Figure 3.6: Hann windows for the entire frequency range with slope = 0.5 and intercept = 5.

If we use the same array geometries as before, we see the influence of the Hann window in Fig. 3.7. The width of the main lobe changes less over frequencies and is also wider. But nevertheless it is very hard to really control the shape of the main lobe, and we especially don't know if a particular parameter combination is optimal. It is also not possible to steer the beam with the coefficients. Therefore, the further development of the *Richt Tool DS* was ceased and instead a more suitable algorithm, based on least-squares optimization, was approached, where we can actually optimize the main lobe.

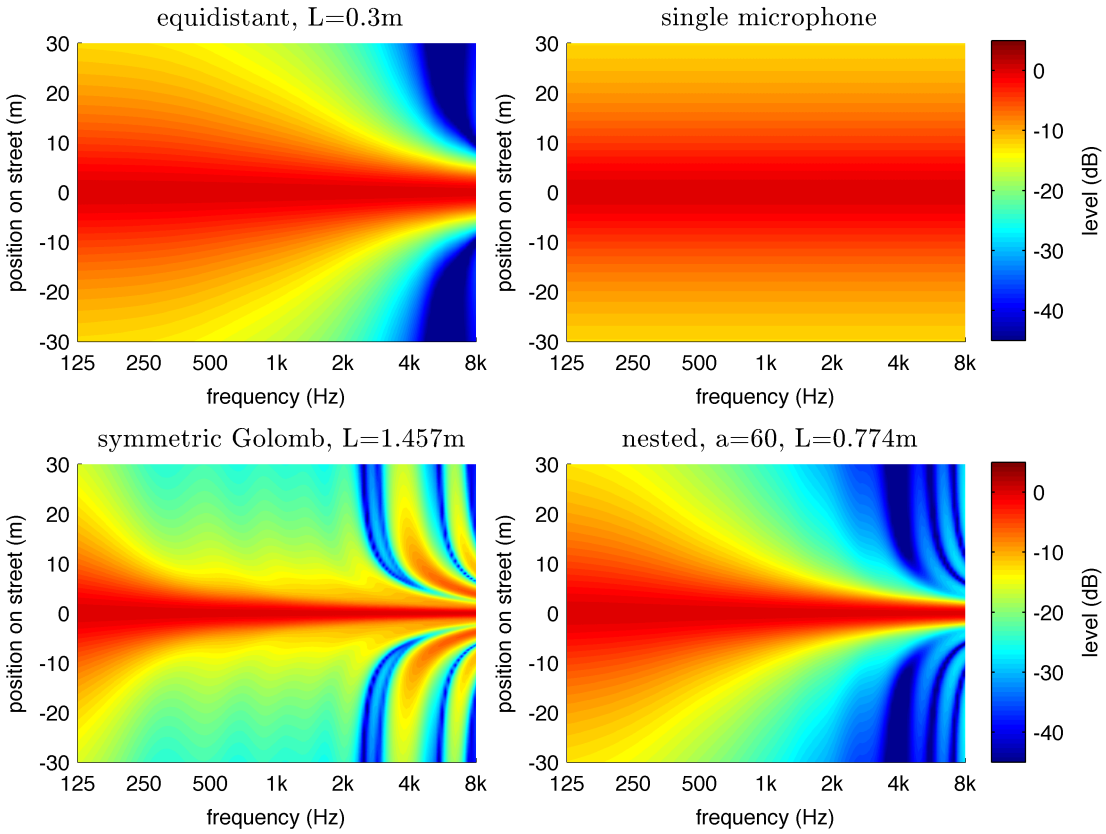

Figure 3.7: Directivity pattern along the street for different spacings of the microphones with the delay-and-sum beamformer with a Hann window (intercept = 5, slope = 0.5). Because of the asymmetry the Hann window for the Golomb spacing does not work properly; instead the "directivity pattern" for a single microphone is plotted. The length of the array for the different spacings is adjusted to a resulting $f_{max} = 8\,kHz$.

3.3 Richt Tool LS

In this section a specialized beamforming algorithm is derived. It uses directivity pattern synthesis [6] to control the width of the main lobe. This algorithm of the *Richt Tool LS* is described in section 3.3.1. But beside the directivity pattern and the overall level along the street, the *Richt Tool LS* also shows various performance parameters, which will be discussed in section 3.3.2. A screen-shot of the GUI is shown in Fig. 3.8. To address the specifications of the least-squares algorithm, some general adjustments in comparison to the *Richt Tool DS* have been made. To ensure the correct behavior of the algorithm, the directivity pattern should be calculated for an angle from 0 to π radians. Furthermore, the beamformer should not compensate the damping of the propagation. So the sources of the directivity pattern with the positions $\boldsymbol{x}$ are placed in a half-circle around the array, but the axis in the directivity pattern shows the corresponding position on the virtual street. The overall level is only calculated up to f_{max}, the level for the frequency range from f_{max} to 10 kHz is shown in a lighter color.

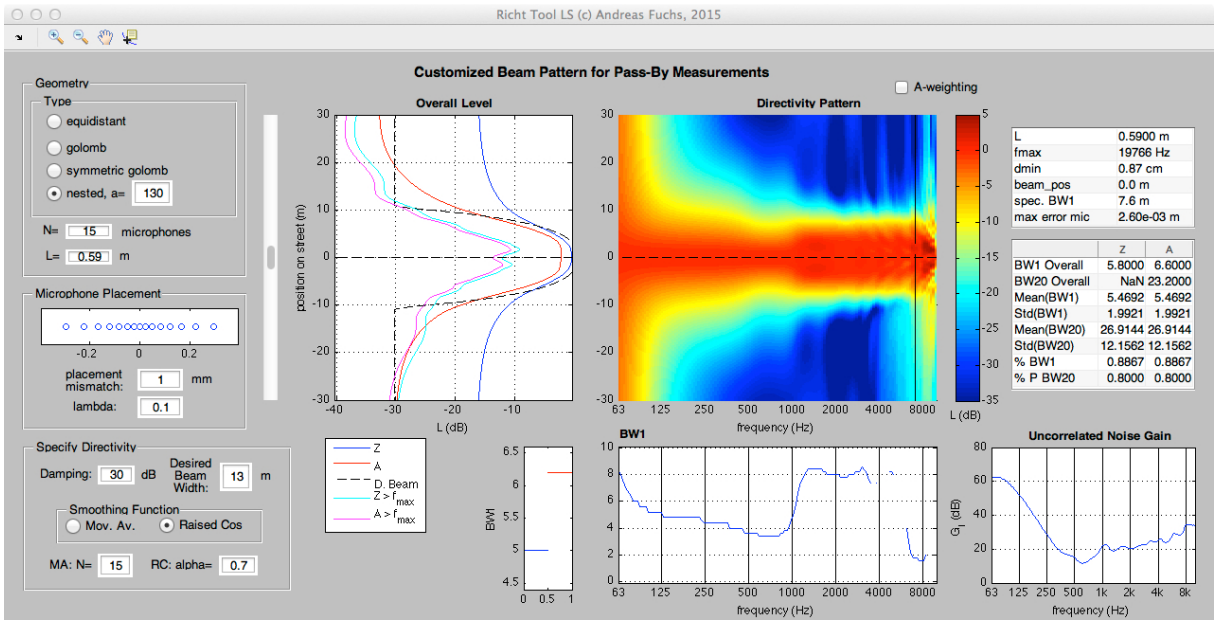

Figure 3.8: Matlab GUI Richt Tool LS.

3.3.1 Algorithm

In section 2.2.2 the fundamentals for the directivity pattern synthesis were given (eq. (2.9) to (2.11)). These can be easily implemented in MATLAB with the function *pinv()* for the pseudo-inverse:

```
1  %% directivity pattern synthesis
2  W = pinv(D)*B;
3  W_opt = W./(D(b,:)*W);
```

Listing 3.2: Implementation of the directivity pattern synthesis without additional constraint. Vector B contains the specified directivity Δ, matrix D the physical model of the propagation in the pre-defined directions.

This implementation results, beside numerical effects, in a perfect directivity pattern, see Fig. 3.9 in the upper left corner. If we use a microphone displacement with 0.1 mm in every direction, which will be very hard to achieve, the directivity pattern below 2 kHz is useless, as depicted in the upper right corner in Fig. 3.9; one has to notice the different colorbar ranging from -90 to 150 dB (!). With this configuration the uncorrelated noise gain, which is shown in the lower right corner in the *Richt Tool LS*, is $300 - 400$ dB for lower frequencies. Therefore, the noise gain is directly added into the optimization criterion as additional constraint:

$$\hat{\boldsymbol{W}} = \arg\min_{\boldsymbol{W}} \mathcal{J}(\boldsymbol{W}) = \arg\min_{\boldsymbol{W}} ||\boldsymbol{\Delta} - \boldsymbol{D}\boldsymbol{W}|| + \lambda ||\boldsymbol{W}|| \tag{3.3}$$

The frequency dependency of $\boldsymbol{W}$, $\boldsymbol{W_c}$ and $\boldsymbol{D}$ is omitted for better readability. The optimal solution $\hat{\boldsymbol{W}}$ is calculated by minimizing the cost function

$$\mathcal{J}(\boldsymbol{W}) = \boldsymbol{\Delta}^H\boldsymbol{\Delta} - \left[\left(\boldsymbol{\Delta}^H\boldsymbol{D}\boldsymbol{W}\right)^H + \boldsymbol{\Delta}^H\boldsymbol{D}\boldsymbol{W}\right] + \boldsymbol{W}^H\boldsymbol{D}^H\boldsymbol{D}\boldsymbol{W} + \lambda\boldsymbol{W}^H\boldsymbol{W}\,. \tag{3.4}$$

Applying the Wirtinger derivates we get

$$\boldsymbol{\nabla}_{\boldsymbol{W}}\mathcal{J}(\boldsymbol{W}) = -2\boldsymbol{D}^H\boldsymbol{\Delta} + 2\boldsymbol{D}^H\boldsymbol{D}\boldsymbol{W} + \lambda\boldsymbol{W} \stackrel{!}{=} \boldsymbol{0}\,. \tag{3.5}$$

The optimal solution after the optimization must also be normalized:

$$\hat{\boldsymbol{W}} = (2\boldsymbol{D}^H\boldsymbol{D} + \lambda\boldsymbol{I})^{-1}(2\boldsymbol{D}^H\boldsymbol{\Delta}) \tag{3.6}$$

$$\boldsymbol{W_{opt}} = \frac{\hat{\boldsymbol{W}}}{\boldsymbol{D_b}\hat{\boldsymbol{W}}} \tag{3.7}$$

The Matlab implementation can effectively be done with *mldivide* (\):

```
1  %% directivity pattern synthesis with noise gain constraint
2  W = (2*(D'*D)+lambda.*I)\( 2*D'*B);
3  W_opt{ff} = W./(D(v,:)*W);
```

Listing 3.3: Implementation of the directivity pattern synthesis with an additional constraint. Vector B contains the specified directivity Δ, matrix D the physical model of the propagation in the specified directions.

It can be easily seen, that for $\lambda = 0$ this solution is the pseudo-inverse (eq. 2.10). For better comparison, there is a distinction in the *Richt Tool LS*. The solution with $\lambda = 0$ is calculated with the code in Listing 3.3, if λ is set to -1 the *pinv* function (Listing 3.2) is used, which is numerically more stable. In the lower half of Fig. 3.9 the effect of this noise constraint is shown. In comparison to the solution without displacement, we lose directivity for low frequencies and also the side lobe damping is less in the lower left figure to the figure above. But if microphone displacement is added, as done in the lower right figure, the main lobe is nearly the same, and there is still enough side lobe damping of about 30 dB at higher frequencies.

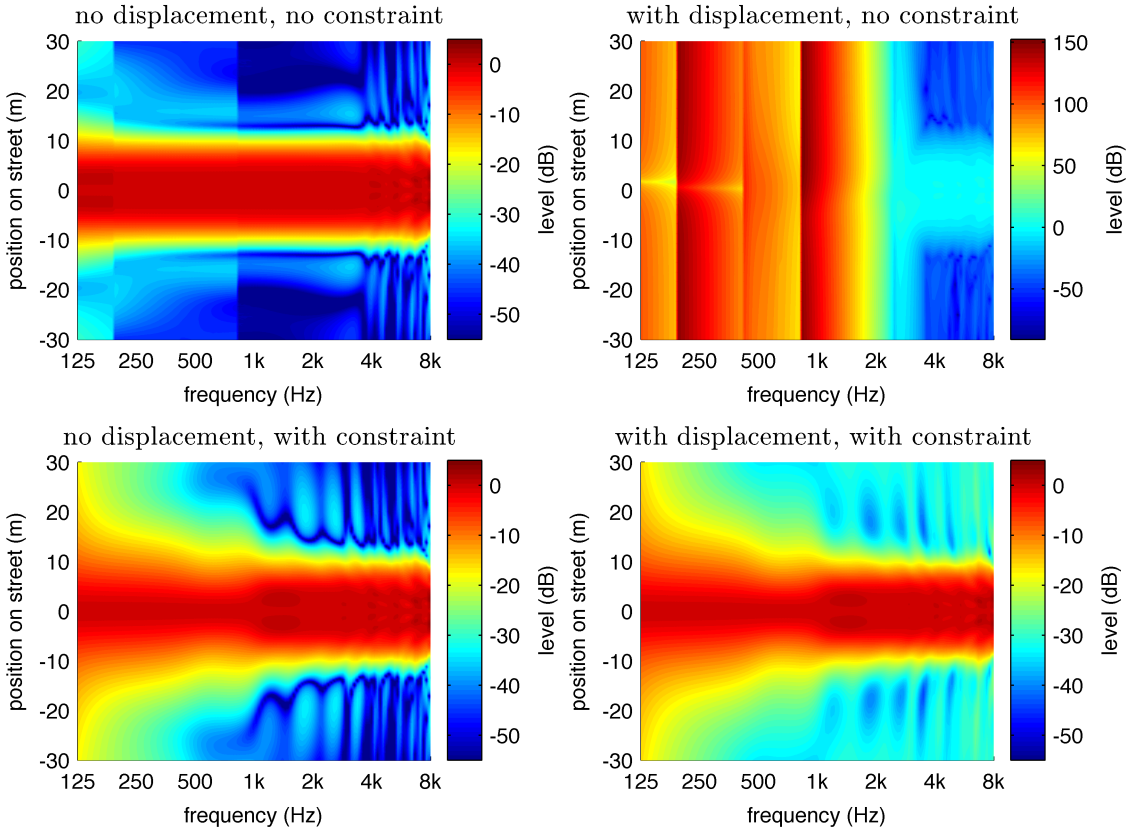

Figure 3.9: Influence of displacement and noise constraint on the directivity pattern for the least-squares beamformer. In the upper half, one can see the implementation with no constraint from Listing 3.2. The implementation with constraint ($\lambda = 0.1$) from Listing 3.3 is visible in the lower half. On the right side a microphone displacement below 0.5 mm is simulated.

Although we have now taken care of the robustness of the algorithm, special considerations must be made in regard to the specified directivity, because the least-squares optimization tries to fit the filter coefficients $\boldsymbol{W}$ to the specified directivity in $\boldsymbol{\Delta}$. The intuitive approach is to put an 1 in every direction we are interested in, and in all other directions a zero. Of course it is impossible to achieve this directivity, so it may be better to specify a more realistic directivity. First of all, we set the coefficients outside the main lobe not to zero, but to a sufficient *damping*, for example $-40\,\mathrm{dB}$ (i.e. 0.01). Then we tried two approaches to smooth the slope of the main lobe:

Moving-Average: A (zero-phase) moving-average filter with variable length N_{MA} is applied in order to smooth the transition.

Raised-Cosine: A raised-cosine filter with a variable roll-off factor α is used, which should give an even more realistic directivity pattern.

The specified directivity is marked as a dotted line in the overall level in the *Richt Tool LS*; Fig. 3.10 shows these smoothing functions. The filter length of the moving average filter N_{MA} and the result of the optimization depend on the spacing of the predefined directions, i.e. for which directions the optimization is calculated. We chose a 20 cm spacing between -50 to 50 m on the street, but it is necessary to define the whole half circle. Therefore, this spacing is converted to an angle ϑ ($\vartheta = 0$ in front), and the missing angle to $-\pi/2$ and $\pi/2$ respectively is filled with 30 equidistant directions. Also the parameter λ depends on this spacing or at least on the number of specified directions, which is important in the analysis of measurements.

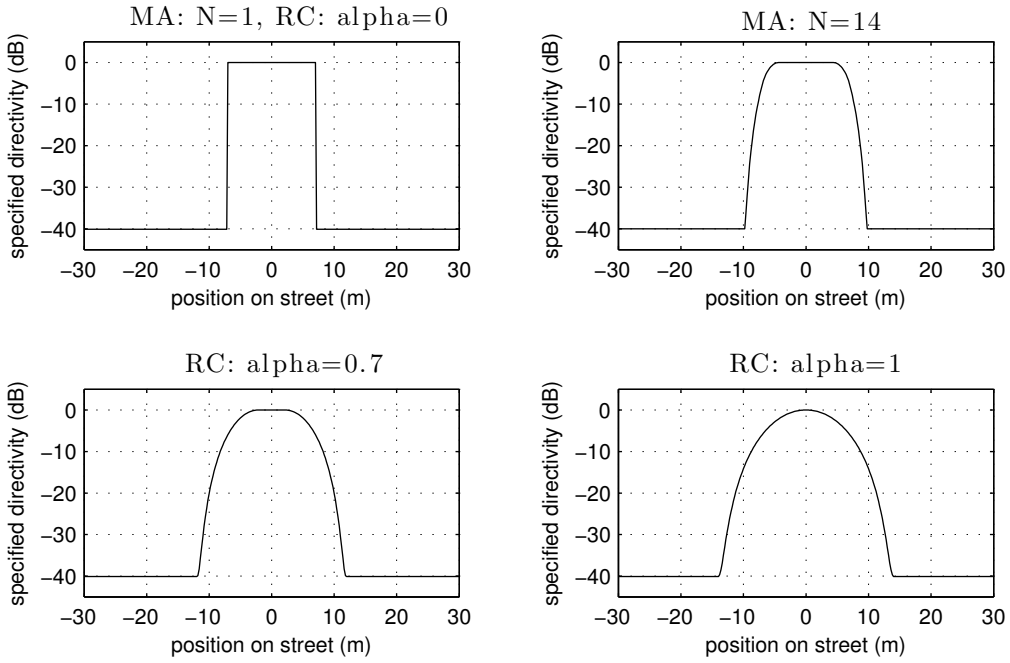

Figure 3.10: Smoothing functions for the specified directivity for the LS algorithm. The figure in the upper left corner uses no smoothing, in the upper right corner is the moving average filter with a filter length $N_{MA} = 14$. In the lower half the slope is smoothed with a raised-cosine filter with a roll-off factor $\alpha = 0.7$ and $\alpha = 1$ respectively.

Without any smoothing, definite ripples can be seen in the directivity pattern (Fig. 3.11 upper left corner). If we use a roll-off factor of $\alpha = 1$ as in the lower right figure, we can get a very good side-lobe damping, but the gain factor in the main lobe is not constant. The two remaining figures show compromises, so an optimal solution exists.

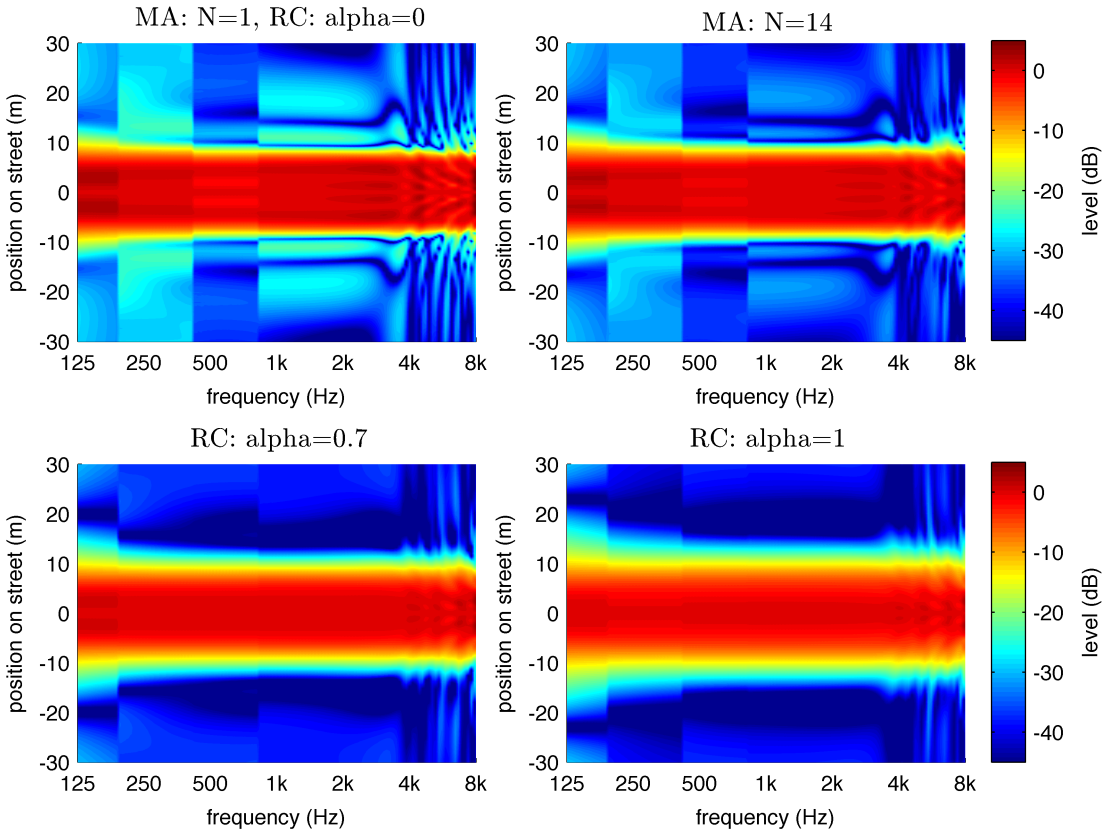

Figure 3.11: Directivity patterns for the specified directivities in Fig. 3.10. To see the isolated effects of the smoothing, these patterns are calculated without any noise constraint and microphone displacements.

3.3.2 Performance Parameters

We have now ten parameters defined, which affect our directivity pattern. The MATLAB GUIs can immediately show the effects of these and one can find useful ranges for the parameters. Even though it is not hard to find a promising combination, as shown in the past figures, it is not possible to know, if this is the optimum. Therefore various performance parameters have been intuitively created to run a computerized optimization. To test these parameters they have also been added to the *Richt Tool LS* on the right side.

BW1 Overall: The main goal is to capture the whole vehicle with sufficient accuracy. Therefore, we are interested in the width of the main lobe (which must occur only one time), where the deviation is less than $\pm 1\,\mathrm{dB}$ and is called **B**eam**W**idth**1**. The deviation of $\pm 1\,\mathrm{dB}$ is based on the accepted deviations in directivity and frequency for sound pressure level meters class 1, as defined in [4]. As the name suggests, the BW1 Overall is calculated for the overall level and should be maximized.

BW20 Overall: Although defined in a similar way (the width of the main lobe where the deviation is less than $20\,\mathrm{dB}$), this parameter can be used to asses the damping. If only the main lobe exceeds a damping of $20\,\mathrm{dB}$, this width shows at which point on the street a sufficient side lobe damping is reached. This parameter should be as small as possible.

mean(BW1)/std(BW1)/mean(BW20)/std(BW20): The BW1 and BW20 can also be calculated for every frequency bin. Statistics help to reduce these values to only a few numbers,

which can then be easily analyzed. The mean value and the standard deviation are calculated, whereas frequencies, where the beamwidths aren't defined due to unacceptable deviations, are omitted.

% P BW1/% P BW20: This percentage is the ratio of the frequency bins, where the beam widths are defined, to the total number of frequency bins. Therefore this percentage shows the reliability of the statistics over the frequency bins.

In the *Richt Tool LS* (see Fig. 3.8) the BW1 is also plotted for the Overall Level and for each frequency bin below the directivity pattern. In the lower right corner the uncorrelated noise gain after eq. (2.7) is plotted. In the upper right corner, the specification of the array and the directivity is summarized: The total length L of the array, the maximum frequency before aliasing f_{max}, the minimum microphone distance d_{min} and the steered position of the beam on the street *beam_pos*. The resulting BW1 for the specification *spec. BW1* and the greatest occurring deviation of a microphone is shown as well.

3.4 Optimization

The previous two sections focused on the description of the MATLAB GUI *Richt Tool LS*, but the underlying algorithms can be used for the optimization of the parameter set. We have a total of ten input parameters, for which it was possible to find plausible ranges with the GUI. Furthermore, we have different performance measures to evaluate a parameter set. Our approach of optimization is a mixture of a Brute-Force method and a Monte-Carlo Simulation. Each parameter has predefined possible values; for some parameters each possible value is picked, for others we pick the value randomly from this predefined set. The chosen predefined set is shown in Tab. 3.1.

Parameter	**Predefined Set**	**Type**
N	{10,15}	brute
a	{(10:5:45), (50:20:230), 250, 400, 500, 800, 1000}	brute
beam_pos	{0}	brute
mismatch in placement	{0.1, 0.5, 1}	random
damping	{100, 40, 30, 25}	random
beamwidth	{(10:2:20)}	random
λ	{0.001, 0.01, 0.1}	random
smoothing function	{MA, RC}	random
N_{MA} (only MA)	{1, (10 : 5 : 50)}	random
α (only RC)	{(0:0.1:0.9)}	random

Table 3.1: Parameter sets for the optimization.

The first two parameters (N & a) are calculated for every possible combination, because they define the geometry of the array and can not be changed after the measurement has been done. The measurement equipment uses units with 5 microphone amplifiers with a total maximum of 3 units ($N = 15$ microphones). Of course more microphones are better, but two units ($N = 10$ microphones) might be enough. The *Richt Tool LS* showed that the microphone spacings based on the Golomb ruler perform poorly with the least-squares algorithm, therefore they are excluded from the simulation. Then the whole spacing can be controlled with the parameter a, because if it is set to 1000, a perfectly equidistant spacing is used, which is more or less equivalent. The length of the array is set, so the maximum frequency is 8 kHz. The

simulation was prepared for various steering positions *beam_pos*, but because of efficiency only the perpendicular direction was calculated.

The brute-force parameters are 46 combinations in total. The random parameters equal 4320 combinations, from which 2241 have been calculated, which is more than 51 %, and equals more than 100 000 input parameter sets.

To find an optimal parameter set, the number of results got reduced by defining thresholds for three performance parameters:

BW1 $> 7\,$**m:** The main lobe must be greater than the measured car plus one additional meter at the front and back for measurement uncertainty.

BW20 $< 35\,$**m:** At least in 17.5 m distance we want to have sufficient side lobe damping of 20 dB.

% P BW1 $> 95\,$**%:** The **BW1** must be defined for more than 95 % of all frequency bins.

Only 239 combinations fulfill these rather strict thresholds, which will guarantee useful solutions. To find an optimal solution, two specific parameters were used: The ratio of BW1 and BW20 and the standard deviation of BW1. If the ratio BW1/BW20 is high, the slope of the main lobe will be very steep. So we want to maximize it. If the standard deviation of the BW1 for different frequencies is low, we have a similar BW1 in all frequencies, which is favorable. Fig. 3.12 shows these two parameters for all remaining combinations. From index 140 to 157 and index 162 to 179 a similar structure with good properties (high BW1/BW20 and low std(BW1)) is visible. In this whole section (index 140-179) only few input parameters vary: In the first half the damping is $-40\,$dB, in the second half $-100\,$dB, but actually there is not much difference. The second varying parameter is the spreading factor a, which changes monotonically from 110 to 250 in each half. The regularization parameter for the least-squares algorithm λ is sometimes either 0.001 or 0.01. The other parameters were constant in these ranges, leading to the following choice of input parameters:

N	a	*damping*	*beamwidth*	λ	*smooth. func.*	α
15	130	40	14	0.01	RC	0.7

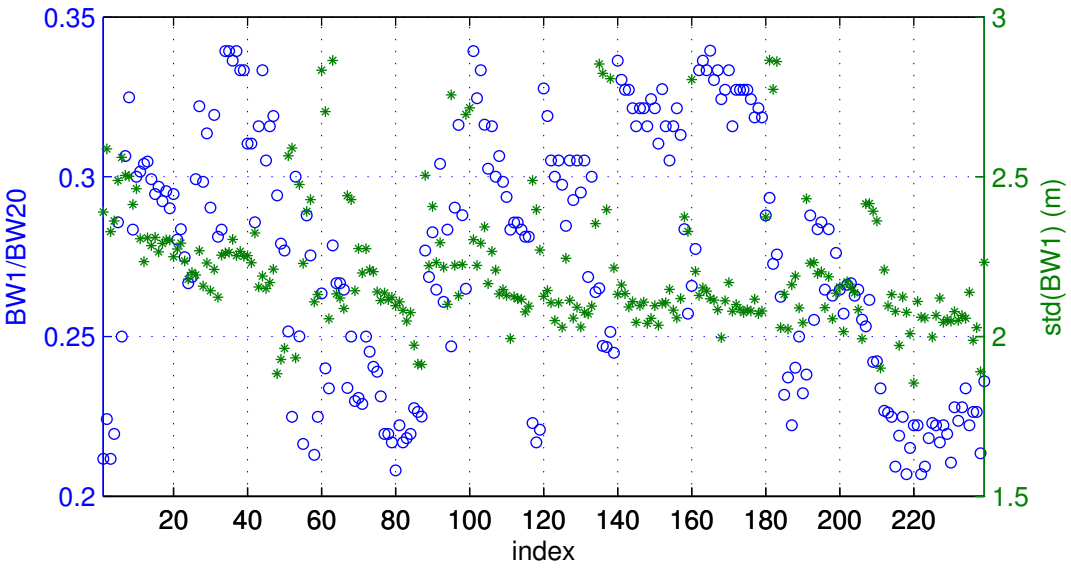

Figure 3.12: Performance parameters BW1/BW20 (blue circles) and std(BW1) (green crosses) for various combinations of input parameters.

Fig. 3.13 shows the resulting directivity pattern and A-weighted overall level along the street with this optimized parameter set. The main lobe is more than 7.6 m wide, and in 12 m from

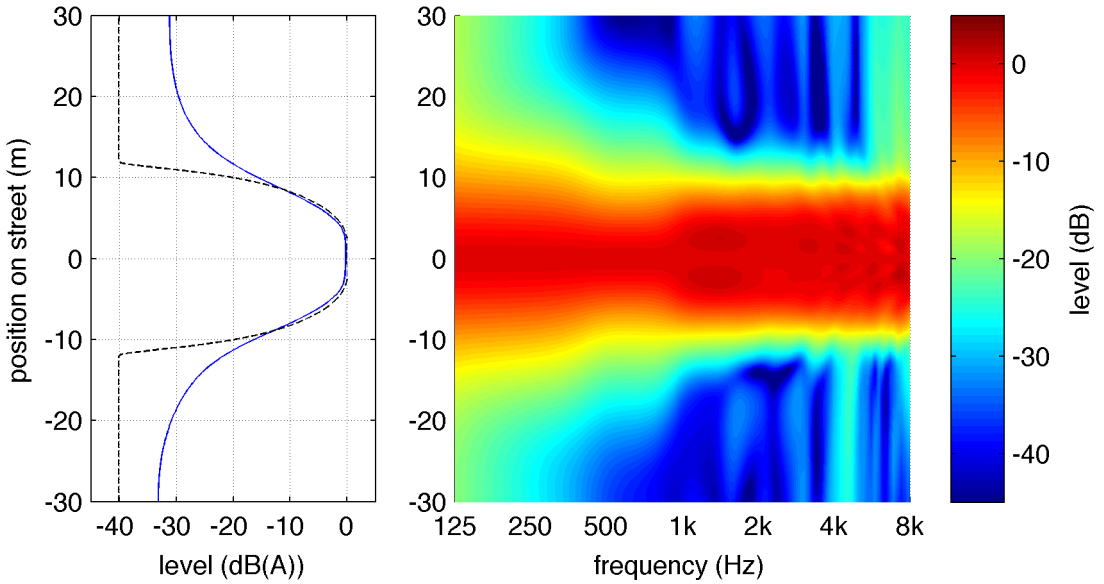

Figure 3.13: Optimized directivity pattern along the street. Left the A-weighted overall level is plotted, in the right picture no frequency weighting is used to show the constant beamwidth over the whole frequency range. The biggest occuring microphone displacement in this simulation is 2.8 mm.

the center of the main lobe a damping of 20 dB is reached. The width of the main lobe is also very constant over the whole frequency range, especially above 1 kHz. This range is particularly important, because due to the A-weighting the overall level is most sensitive there.

3.5 Summary

This chapter showed the evolution of a small GUI, which was created to get a sense for the influence of different parameters, to a complex simulation with numerical optimization, efficient computation and integrated evaluation. The simplicity of a delay-and-sum beamformer was replaced by finding an optimal solution for this application. As shown in Fig. 3.13 we are able to design a main lobe, where a whole car fits in, and we still have sufficient side-lobe damping. By taking special care of the robustness and the specification of the directivity, a both practical and high-performance solution was found. In the next chapter the measurements are presented to see the capabilities of this design.

4 Measurements

The aim of this project is to show the possible application of beamforming for pass-by measurements. In the previous chapter a robust beamforming algorithm and array geometry for exactly this purpose was derived and optimized. Now, this beamformer should be actively used in pass-by measurements. As mentioned in Chapter 2, beamforming may be useful for statistical pass-by measurements (SPB), but for evaluating the performance, controlled pass-by measurements are much more suitable. To test the capabilities of the beamformer (unit gain main lobe and sufficient side-lobe damping), these measurements are performed with two vehicles consecutively passing-by. The measurement set-up is described in section 4.1, whereas the results are presented in section 4.2.

4.1 Measurement Set-up

The measurements have thankfully been realised at the *AIT Austrian Institute of Technology GmbH - Mobility Department* in Vienna, which also owns the two measured vehicles: a Volvo XC60 and a BMW X3 (both compact SUVs).

Tab. 4.1 gives an overview of the used measurement equipment. Except for the Norsonic microphones, the whole system is calibrated in an accredited way. If the three Brüel&Kjaer 3560 front-ends are wired-up, a total of 15 microphone channels are available. Because of this high-quality equipment, we can assume that the microphones have a flat frequency response and a good omni-directivity. The highest achievable frequency f_{max} for the built array is 4.7 kHz, due to the size of the microphones, microphone holders and windshields. If only the total length L of the array of our optimized parameter set is changed, the directivity pattern gets shifted to lower frequencies, which will improve the directivity at lower frequencies. Due to the A-weighting and typical spectral distribution of traffic noise [9], the mandatory low-pass filtering at $f_{max} = 4.7$ kHz to prevent aliasing will not have much effect on the overall level. The resulting geometry for the array is shown in Fig. 4.1.

Number	Manufacturer	Name	Description
3	Brüel&Kjaer	3560 B-130	front-end with five microphone channels
10	Brüel&Kjaer	4189-L-001	1/2 in free-field microphone
10	Brüel&Kjaer	2269-L	microphone pre-amplifier
5	Norsonic	Nor1220	1/2 in free-field microphone
5	Norsonic	Nor1201	microphone pre-amplifier
1	Brüel&Kjaer	4231	sound calibrator
15	Brüel&Kjaer	-	windshields (5 and 9 cm diameter)

Table 4.1: The used acoustic measurement equipment.

Although the least-squares beamformer should be robust, we must still have a good positioning of the microphones. For this purpose a positioning-aid was designed and manufactured at *Graz University of Technology*, which is made of two rectangular acrylic glass plates joined together. One plate has 1/2 in laser-cut holes at the exact positions of the microphones, whereas the other one is solid. With this positioning-aid an accurate positioning should be possible, as one can see in Fig. 4.2.

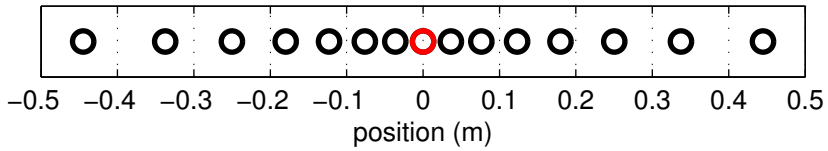

Figure 4.1: Illustration of the finally used array dimensions for the measurements. The maximum frequency is 4.7 kHz. The microphone in the center (red) is used as the reference microphone in the analysis.

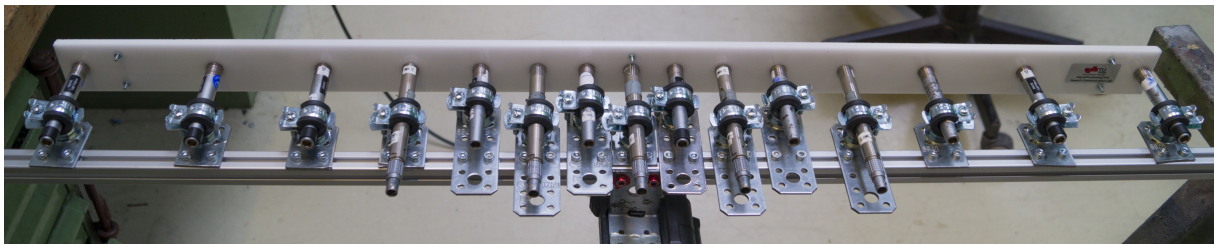

Figure 4.2: The microphone array with positioning-aid.

The following goals were defined for the measurement:

- speed range 20-70 km/h
- for every vehicle a minimum of seven single CPB measurements at different speeds
- 15 pass-bys with two cars with different gaps. *Close* $\sim$ 1 s, *middle* $\sim$2 s, *wide* $\sim$ 3 s.

Additionally three light barriers were used to track the position of the vehicles. Their position can bee seen in Fig. 4.3. With these light barriers we do not need to do acoustic source tracking to steer the beam, because the position of the vehicles can be calculated from the three signals from the light barriers under the already needed condition of constant speed at the pass-by. In Fig. 4.4 a picture of a double pass-by measurement with the array with windshields is shown. Also a light-barrier next to the most right traffic cone is visible.

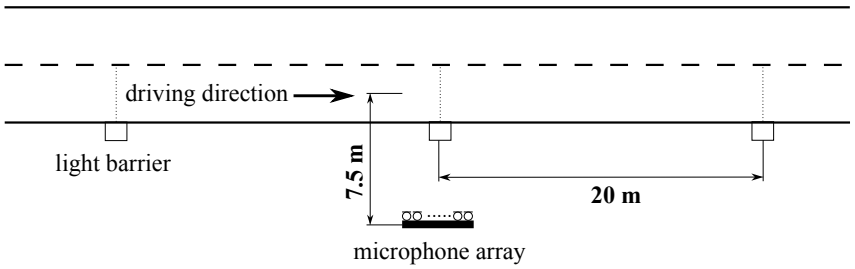

Figure 4.3: Positioning of the measurement equipment on the measurement site.

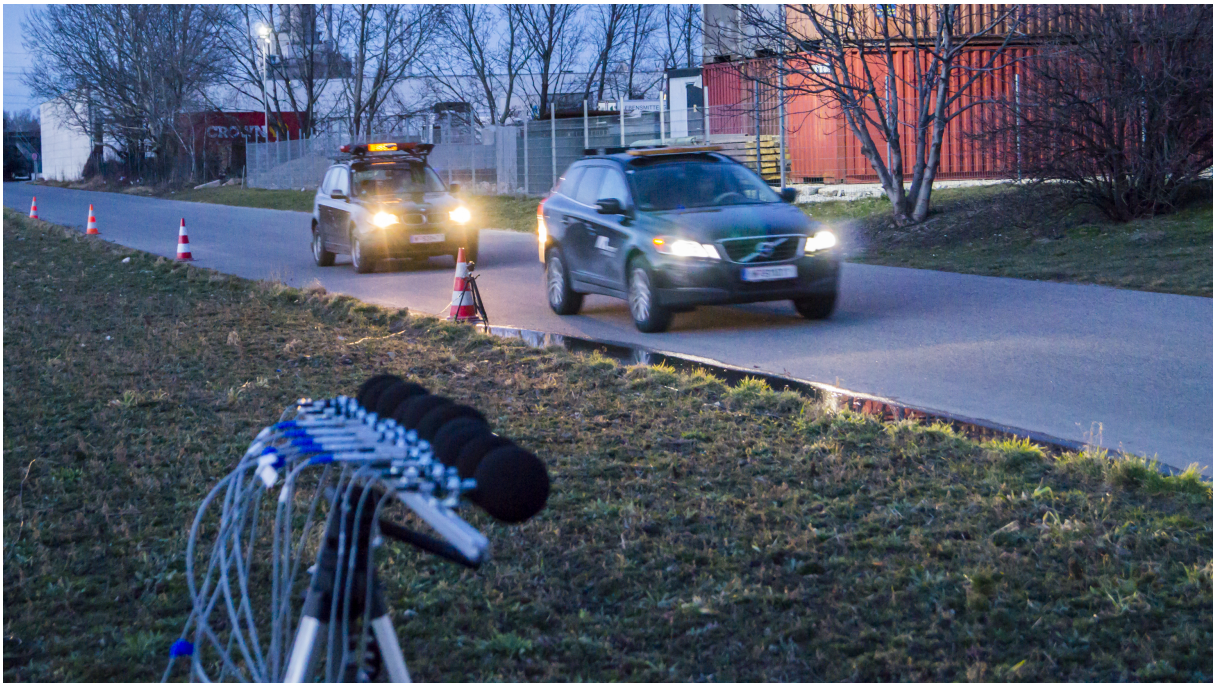

Figure 4.4: Picture of a double pass-by measurement.

4.2 Analysis

The first task in the analysis is to check the recording system. To do so, a clap in one meter distance next to the array was recorded. In Fig. 4.5 this quasi impulse responses of the microphones are shown. The peaks of all measurements are aligned to show the similar phase of the microphones, although the Norsonic microphones need to be phase-shifted by 180 °, which is already done in the figure.

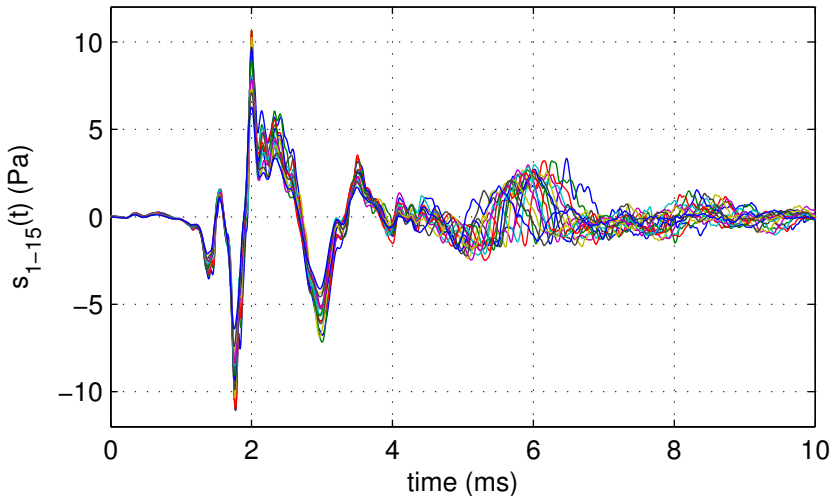

Figure 4.5: Time-aligned recordings of a clap along the array from 15 microphones. At about 6 ms the first reflection from the ground with different propagation delays is visible.

The beamforming coefficients matrix $\boldsymbol{W}$ does not depend on the input signal and can be calculated beforehand. Beside the optimized set of input parameters, the matrix depends on the air temperature (during the measurement 7 °C) to calculate the correct speed of sound. The matrix was computed for a steering position on the street from −30 to 30 m in 10 cm steps.

In the analysis the time signals of the microphones are loaded, and are arranged in blocks with 8192 samples length without any overlap, but zero-padded in the first half. This is done because the beamformer will work as a delay with a negative time delay. Then the Fast-Fourier transform is calculated. For every time block the current position of the car is calculated from the light barrier signals and the closest available coefficient matrix $\boldsymbol{W}$ is used for beamforming in the frequency domain. Therefore the beam is steered with the vehicle. Afterwards the inverse Fourier transform is calculated and with an overlap&add method the output signal y of the beamformer is calculated.

Fig. 4.6 shows why the steering of the beam is mandatory. If the beam is fixed to the perpendicular direction (dashed line), we may get the exact level, when the car is in front of the array. But we apply much damping to the level right before and after this point. At high speeds (lower right picture) this may result in lower time-weighted levels. The position taken from the light barriers (solid lines) correlate very well with the level.

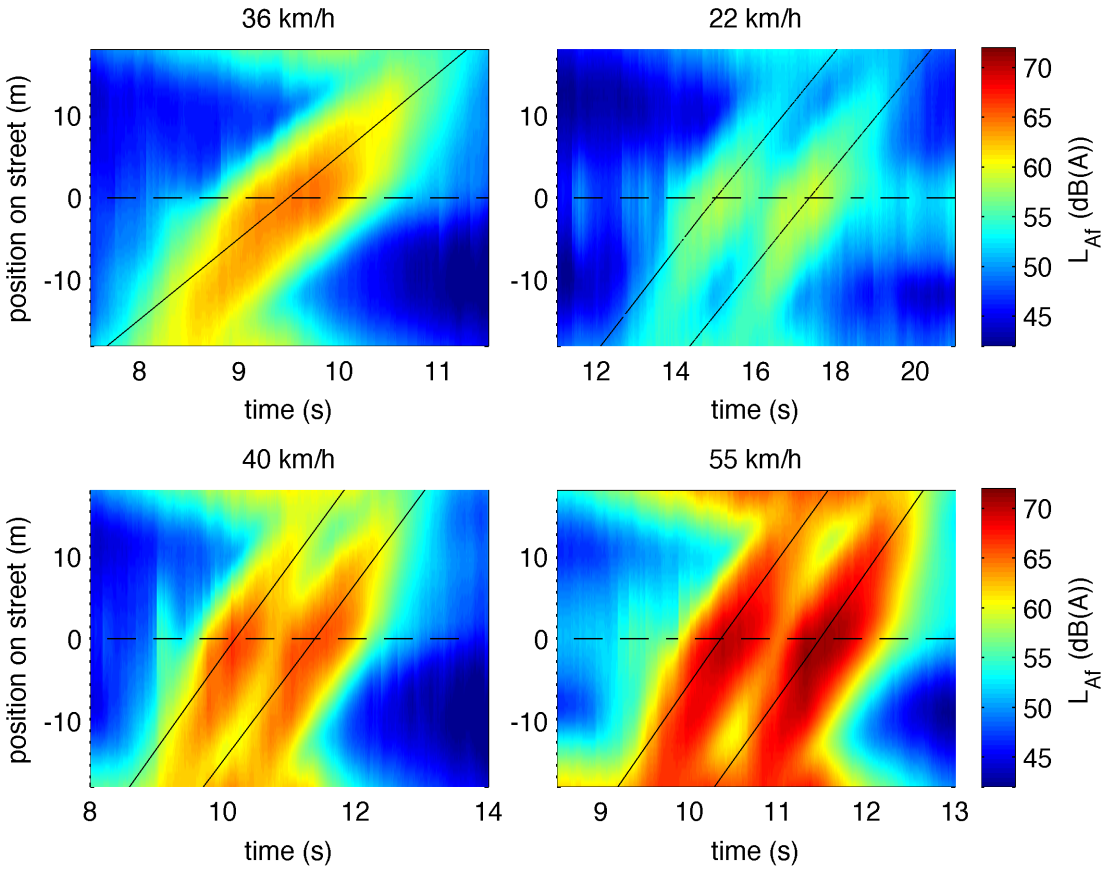

Figure 4.6: Four pass-bys (1 single, 3 double) with fixed steering to the measured area on the street. If no steering is used, the level would get calculated along the dashed, horizontal lines. Instead the level is calculated along the solid, skewed lines, which are the positions of the cars from the light barriers.

Fig. 4.7 shows the level after beamforming with steered beam. For better comparison the figure shows the same pass-bys with the same time-axis as Fig. 4.6. For the single pass-by in the upper left corner the level of the beamformer output follows even small level changes on the reference microphone in the center of the array. The deviation is below 0.2 dB. A negligible time-delay is also visible. This means the level of the beamformer works precisely. The other important thing for the application is the damping of disturbing noises. This is visible in the three other pictures with double pass-bys. The double pass-by in the upper right corner has a gap between the cars of 2.4 s, but because of the low speed the gap is very small in meters (15 m).

On the reference microphone it is hard to see two individual vehicle pass-bys and determine two maxima, whereas with beamforming two distinct maxima are visible. The pass-bys in the lower half of the figure have similar gaps in meters (13.3 m and 16.5 m) and again only with beamforming the individual pass-bys are distinguishable.

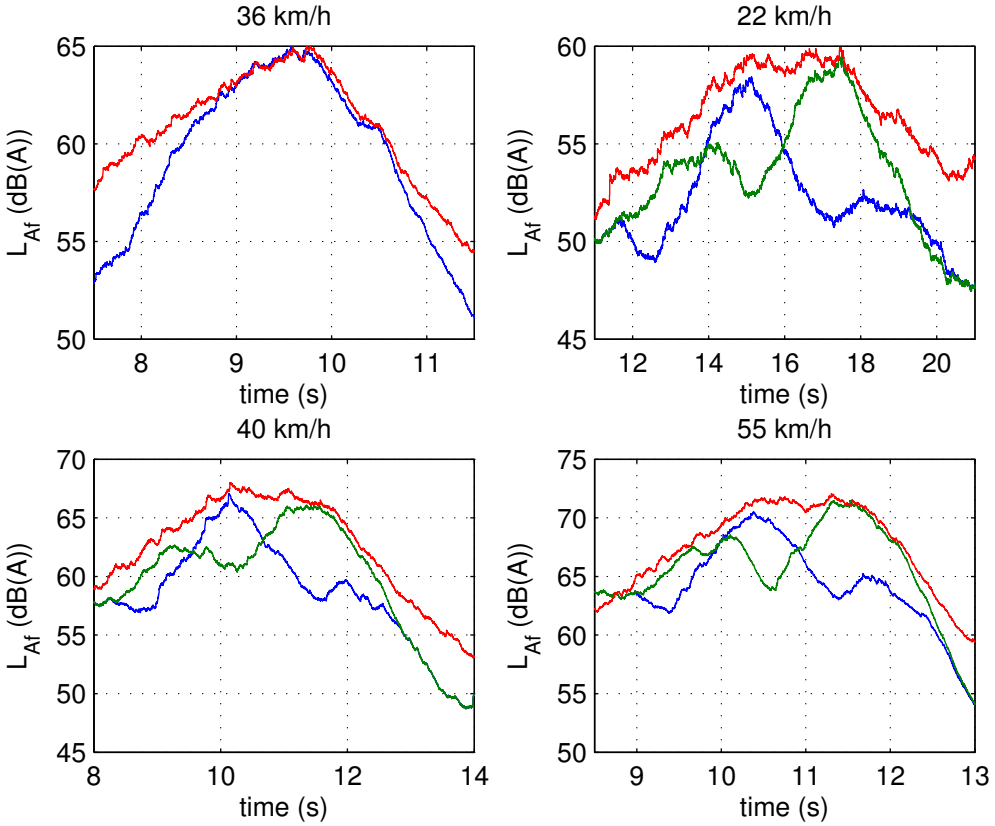

Figure 4.7: Level for the four pass-by measurements (1 single, 3 double). The blue line depicts the level of the first car, the green line the level of the second car. The red line is the level of the reference microphone in the center of the array.

For SPB and CPB measurements the maximum level $L_{Af,\max}$ is important and also used for regression over the logarithm of speed. Fig. 4.8 shows these regressions for the single pass-by and double pass-bys with different gaps for the beamforming level (blue) and the level of the reference microphone (red). For the single pass-by in the upper left picture nearly no deviation is visible. This also matches with Tab. 4.2, which shows the mean value and standard deviation of the error for the single pass-bys. In comparison to [2], where the error of 10 microphones lies between -1.2 dB to -0.4 dB, a much more reliable result was achieved (for the Volvo XC60 the error lies between -0.1 dB and 0.1 dB).

The double pass-bys in Fig. 4.8 show some interesting characteristics. There is nearly no visible difference in the measured maximum levels between *wide* gap and a single pass-by, which is additionally plotted in black in the double pass-bys. Especially for high speeds the distance between the two cars is big enough that no level disturbance occurs. Therefore, beamforming has little effect, but it must be considered that the measured cars are very similar. If the disturbing car is significantly louder than the measured car, the beamformer might improve the result. If the gap is reduced (*middle*), beginning at low speeds the reference microphone shows a too high level, whereas with beamforming the measured level stays nearly the same. This can also be seen at higher speeds for the smallest gaps (*close*). At low speeds, the theoretical +3 dB for two uncorrelated sound sources are visible on the reference microphone, whereas the level with beamforming is still closer to the single pass-by. So with beamforming the measured level

and the resulting regression line stays nearly the same, regardless of the distance to a close-by vehicle.

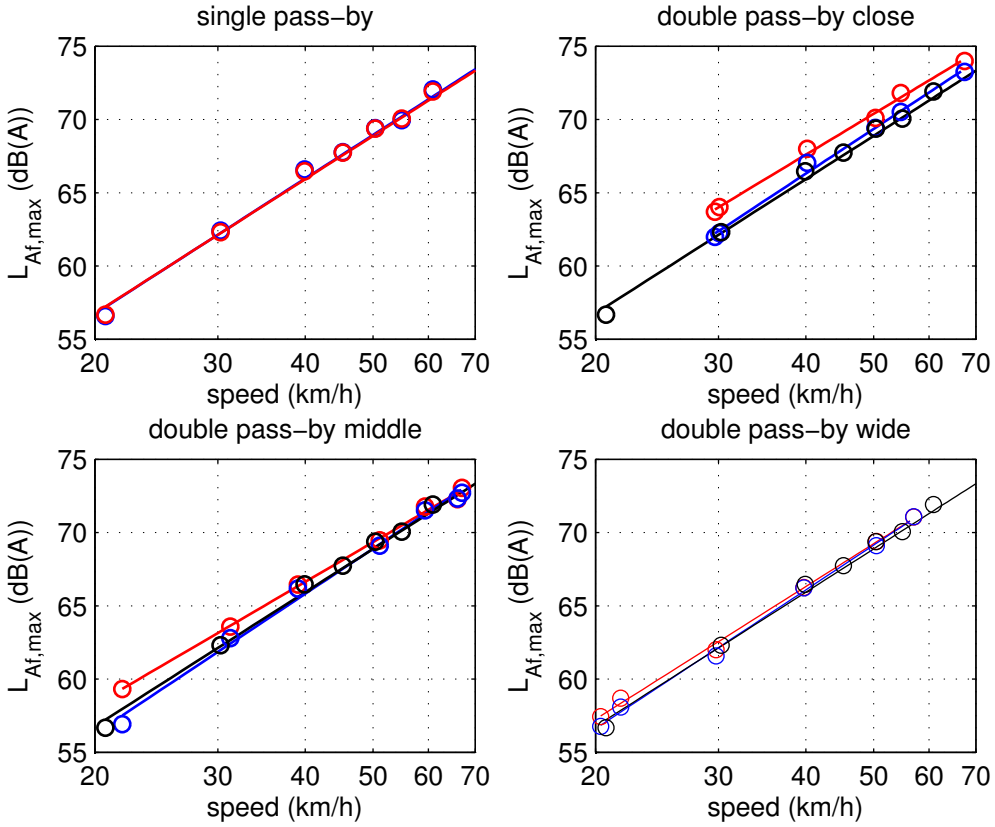

Figure 4.8: Regressions for the CPB measurements with the Volvo XC60. The dots mark the individual maximum levels. In blue is the level calculated with beamforming, red is the level from the reference microphone. For the double pass-bys also the reference level from the single pass-by measurements is plotted in black for better comparision. Close *refers to a gap below* 1.7 *s,* middle *a gap between* 1.7 *s and* 2.6 *s and* wide *a gap over* 2.6 *s between the two cars.*

	BMW	Volvo
μ_X	0.06 dB	0.06 dB
σ_X	0.08 dB	0.10 dB

Table 4.2: Mean μ_X and standard deviation σ_X of the devation of the level calculated via beamforming to the level of the reference microphone: $X = L^{\mathrm{BF}}_{Af,\max} - L^{\mathrm{Ref}}_{Af,\max}$

5 Conclusion

This project tried to combine advanced signal processing techniques and standardized noise measurements. The first results looked promising. The strategy was to capture the whole tested sound source with a beam, which has defined qualities. First the level deviations in the main lobe must be small ($\pm 1\,\mathrm{dB}$), second the width of the main lobe should be constant in the important frequency range, which is highly dependent on the A-weighting. Third we want a steep slope on the edge of the main lobe and a damping of at least 20 dB in all other directions. With simulations in MATLAB R2010b a deeper understanding of the influence of geometry parameters was gained, but a specialized algorithm will have much more effect. In the simulation the derived, least-squares beamformer showed good results and with the right combination of ten input parameters, it was able to fulfill the strict requirements. Special considerations for the robustness must be made in order to design a beamformer which works with real-life imperfections such as displacements of microphones or microphone self-noise and not only with numerical precision.

The measurements validated the theoretical groundwork. The mean deviation of 0.1 dB in the measurements is significantly lower than in the inspired article by Püschel [2], even though or maybe because no extra source separation algorithm was used. By steering the main lobe with the car, the algorithm actually does not calculate any levels, but a new time signal with an improved signal-to-noise ratio in respect to the tested vehicle. Then this time signal can be analyzed with the standard analysis procedures and showed, due to the design of the measurement with two cars, the source separating qualities of the approach. At low speeds the distance between two cars can be very small, therefore the disturbance of the level from a close additional vehicle is very high. At higher speeds the influence is smaller. Although the algorithm may introduce some error due to imperfections in the directivity pattern, the error is still much less than the disturbance from a second vehicle. Most importantly it is independent from the distance between the two cars. This may present the opportunity to perform statistical pass-by measurements in heavier traffic and/or obtain more data in shorter time, because less measurements must be discarded. Due to the directivity the background noise is reduced as well, which may be helpful for controlled pass-by measurements under heavy conditions. The additional error source and the higher effort by measuring with 15 microphones instead of one could be disadvantages.

Additional research should be done in using the algorithm for actual statistical pass-by measurements, to see the actual possible improvements. To use this method in standardized measurements, the additional introduced error should be closer examined, in order to find the exact stage the deviation is introduced (and how constant this deviation is). Although we got nearly the same small error for both cars, this is still no sample size to make any strong statistical assumptions.

Bibliography

[1] *Acoustics - Measurement of the influence of road surfaces on traffic noise - Part 1: Statistical Pass-By method*, ISO 11819-1, 1997.

[2] D. Püschel, M. Auerbach, and W. Bartolomaeus, *Einsatz eines Mikrofon-Arrays für Statistische Vorbeifahrt-Messungen (SPB)*, Forschung Straßenbau und Straßenverkehrsechnik, vol. 1054, 2011.

[3] *Uniform provisions concerning the approval of tyres with regard to rolling sound emissions and to adhesion on wet surfaces and/or to rolling resistance*, UNECE Reg.117, 2011.

[4] *Elektroakustik - Schallpegelmesser Teil 1: Anforderungen*, ÖNORM 61672-1, 2005.

[5] U. Sandberg and J. A. Ejsmont, *Tyre/Road Noise Reference Book*. Harg, Kisa, Sweden: Informex, 2002.

[6] I. Tashev, *Sound Capture and Processing: Practical Approaches*. Wiley, July 2009.

[7] G. Müller, *Taschenbuch der Technischen Akustik*, 3rd ed., M. Möser, Ed. Berlin Heidelberg: Springer-Verlag, 2004.

[8] S. Weinzierl, *Handbuch der Audiotechnik*. Berlin Heidelberg: Springer-Verlag, 2008.

[9] *Road traffic noise reducing devices - Test method for determining the acoustic performance - Part 3: Normalized traffic noise spectrum*, ÖNORM EN 1793-3, 1998.

[10] [Online]. Available: https://en.wikipedia.org/wiki/Golomb_ruler