Faculty of Engineering, Computer Science and Psychology
Institute of Databases and Information Systems (DBIS)

Differential Analysis of Acoustical Smartphone Recording Capabilities - a Contribution towards Smartphone-modulated Perception of Tinnitus

Bachelor Thesis at Ulm University

Presented by:
Johannes Vedder
johannes.vedder@uni-ulm.de
691733

Examiner:
Prof. Dr. Manfred Reichert

Advisor:
Robin Kraft, M.Sc.

2020

Last updated: October 13, 2020

Typesetting: PDF-LaTeX 2_ε

Abstract

Loud noise is a common risk factor for physical and mental health in our industrialized world, which can trigger different sorts of health issues like permanent hearing loss and tinnitus. To mitigate noise-induced problems in daily life, smartphones can be used as an easy way to observe noise levels. As recording quality differs depending on smartphone models and calibration techniques, standardized methods are needed to acquire comparable results. To examine such possibilities in more detail, several acoustical experiments were performed regarding the recording capabilities of in-build smartphone microphones compared to an external microphone to figure out optimal smartphone recording conditions as this further increases measurement accuracy. Additionally, various different calibration approaches differing in effort and accuracy are evaluated. Results show that smartphones are capable of measuring sound pressure levels accurately with only small deviations of about ± 3 dB(A). Moreover, smartphone microphones are heavily frequency dependent, which is why an approach was presented to normalize for these variations. Gathered calibration data was further brought in conjunction with sound perception data of tinnitus probands, to show an application in health issues. The presented methods provide a straightforward approach to measure sound levels with a smartphone and compare them to other device conditions, opening the use of smartphones in the modulation of sound perception in tinnitus and other conditions.

Contents

Acronyms

ADC Analog-to-digital Converter.

AGC Automatic Gain Control.

API Application Programming Interface.

dB Decibel.

Hz Hertz.

NIHL Noise-induced Hearing Loss.

OS Operating System.

Pa Pascal.

PCM Pulse Code Modulation.

SLM Sound Level Meter.

SPL Sound Pressure Level.

STFT Short-time Fourier Transform.

TFFT Time-based fast Fourier Transformation.

ToH Threshold of Hearing.

1 Introduction

Loud noise in daily life can often cause reduced well-being, stress, sleep and concentration problems or in extreme situations even hearing problems, like hearing loss or tinnitus. To quantify the amount of noise people are exposed during the day, an easy and effortless way is needed.

Since measuring the loudness of noise is not an easy task, this can normally only be fulfilled by appropriate hardware, which is capable of measuring sound levels e.g. a Sound Level Meter (SLM). To measure sound levels properly, the device needs to be calibrated beforehand, which can only be achieved by using another very precise reference device. All in all, this is a very cumbersome task. Also, sound measuring devices are not widely used, since they are expensive, large, heavy and only usable for the task of measuring. Besides that, paying attention to high noise levels, e.g. at a concert or a nightclub is not that common and socially accepted. Nevertheless, a frequent exposure to high noise levels has permanent consequences on hearing. To bring a change to this fact, the high prevalence of smartphones can help, since they are equipped with a microphone and a permanent companion. Therefore, adding the possibility to measure noise with a smartphone could increase the awareness for high sound levels and help avoiding them. As a result, this could mitigate the exposure time to high sound levels, thus decreasing the risk for permanent hearing problems and provide a positive impact on general health. Moreover, people with hearing disabilities like tinnitus have the possibility to get acoustical information about their surroundings easily, which might help to better understand their disease.

However, a lot of different smartphones with a varying equipment of hardware exist, and at least for the largest operating mobile systems Android by Google and iOS by Apple, it is not possible to measure the current sound level with a smartphone out of the box. Rather than that, they all differentiate in the way of how they record sound levels since not only the build-in hardware is different, but also can other software lead to different amplitude levels. Additionally, neither the manufacturers of the operating system nor the smartphone producers, provide a calculation to convert the measured amplitudes to a comparable unit. Thus, the question arises, if there is a way to measure the current loudness with a smartphone in a comparable unit, so that measurements can be compared independently of the used device.

To improve the current situation, three calibration approaches are compiled in the following work, which address the problem of the heterogeneous hardware. Two of the approaches are furthermore evaluated and assessed in terms of accuracy, feasibility and convenience.

Additionally, to further review the measuring capabilities of smartphone microphones under different conditions like distance and direction, the acoustical recording capabilities of smartphones as well as the usage of an external microphone, are tested and compared. This shall give information about the accuracy of sound measuring results in certain situations and how reliable they are.

1.1 Motivation

To better analyze the impact of noise in daily life, it is necessary to measure information about surrounding noise multiple times a day. This can help to get a better understanding of the loudness for noise-induced mental health problems and hearing disabilities, such as hearing loss and tinnitus.

Since the analysis of surrounding noise shall be available to as many people as possible, the process of collecting noise levels should rather be uncomplicated to provide an easy use. Therefore, it is not favorable to use any expensive or complicated external hardware. Instead, using a smartphone-based application that makes it easy to collect and save noise data for anyone seems to be a better approach.

Additionally, by offering the possibility to measure noise levels with a smartphone, this opens an uncomplicated way to get quick information about the sound level around oneself to assess the current noise situation. Observing high sound levels in the surrounding with the smartphone, could probably convince a person to leave an area that could possibly lead to hearing damage. Thus, an application could start to raise and increase social awareness for noise-induced health problems since its easy availability.

1.2 Purpose

Gathering a lot of acoustical data for a longer period can not only increase personal well-being and reduce the proneness for noise-induced disabilities, but also benefit scientific medical research. Therefore, this work shall aid the research for the TrackYourTinnitus[1] project, by providing a solution to gather acoustical data with a crowd-based technique to assist the analysis of reasons for tinnitus.

Using the already well working TrackYourTinnitus application, which analyzes tinnitus fluctuation via a survey, it is possible to extend the built-in loudness measuring function. For further analysis, calibration of measurements taken from various smartphones allows a better understanding of how tinnitus is perceived in different noise environments. All in all, this can support further tinnitus research.

1.3 Approach

The following work is based on the already implemented noise measuring function of the TrackYourTinnitus application. Although, the function to record amplitudes is already working, the retrieved data was not satisfactory, due to the hardware differences and no standardized recording methods of the various smartphones. Therefore, several calibration approaches were developed to adjust the measured amplitudes to a common scale. For this purpose, already working approaches were found and verified, however further options which might give the chance for more precise results have been explored in addition to that.

Also, the idea to conduct further experiments regarding acoustical behavior of smartphones has come up, since the quality of the measuring results might change under different conditions. This includes measuring amplitudes in decreasing distances

[1] https://www.trackyourtinnitus.org/

of the sound source and analyzing the fluctuations and changes. Therefore, this should help gaining knowledge about the fact whether the recorded values of smartphones do accurately reflect reality since decrease of sound level follows specific physical laws. Additionally, the impact of measuring the sound level from different directions has been considered as significant, to examine the possible omnidirectional recording capabilities of smartphone microphones in more detail.

1.4 Structure of the Thesis

The thesis is structured in seven chapters. The following chapter deals with the audiological fundamentals about hearing and the technical expertise of sound measuring. The third chapter describes the general concept. Afterwards, in the fourth chapter the experiments which are examined in the course of this thesis are described. The fifth chapter presents the results of the experiments which are then discussed in the sixth chapter. Finally, a conclusion with an outlook is given in the last chapter.

2 Fundamentals

The following chapter explains the audiological fundamentals of hearing and the perception of sound in the brain. Moreover, it is described how sound measuring works in general as well as how it is performed with smartphones. Lastly, additional reference to related studies is given.

2.1 Hearing

The human ear gives us the ability to sense sounds perceived via vibrations, which are released by an audio source. The outer ear collects the waves in the air, which then go through the ear canal until they reach the eardrum, which vibrates. Three bones, malleus, incus and stapes then transfer the sound vibrations further to the snail-shaped cochlea. As the cochlea is filled with fluid, the vibrations cause ripples, which are sensed at the basilar membrane. As the cochlea is not shaped evenly, sound waves of high frequency are sensed at the beginning part, which is narrow and stiff. Lower frequencies are sensed at the more flexible and wider top of the cochlea (apex). At the basilar membrane, the organ of Corti picks up the waves by using about 20.000 small hair cells, which convert the sound waves into nerve impulses. The electrical signal then travels via the auditory nerve to the cerebral cortex of the brain, thus enabling us to hear [31]. Figure 2.1 shows this.

Being exposed to noise over a longer period may decrease the ability to hear. This is referenced to as Noise-induced Hearing Loss (NIHL). NIHL can be caused by damage to the hair cells in the cochlea or by damage to the synapses of the auditory nerve [1]. High sound levels cause destruction to hair cells and degenerated hair cells will not get replaced and therefore die, resulting in a permanent deterioration to hearing. NIHL can be temporary by exposure to sounds of more than 80 dB(A), e.g.

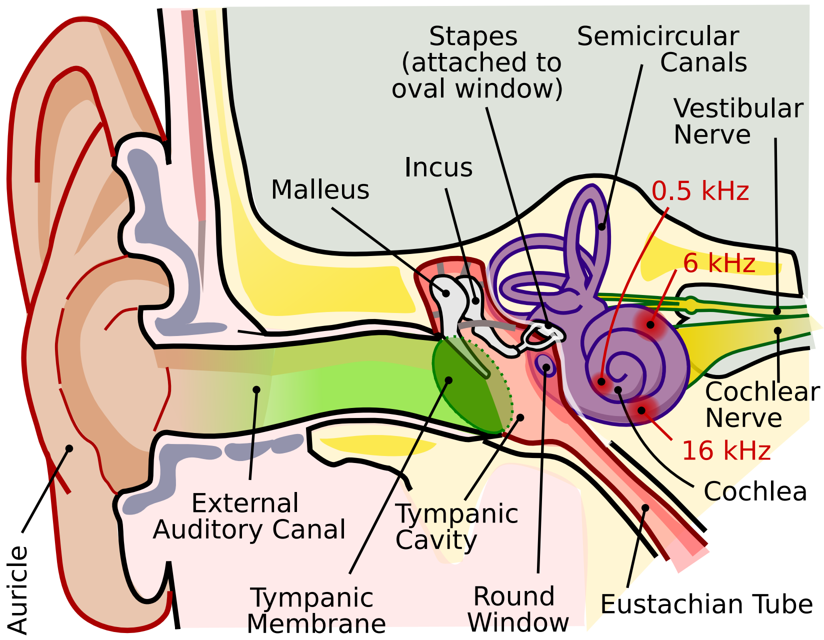


Figure 2.1: Overview of the auditory system [7]

by industrial noise, jet engine, visiting a nightclub. Usually hearing then recovers within 24 hours. But reoccurring exposure and exposure over a longer time to high-intensity sounds (> 120 dB(A)) will cause the hearing loss to be persistent [26].

2.2 Auditory Perception

Hearing a sound and understanding what it means does not mean the same thing. While the pure task of hearing is performed solely by our ears, the actual process of recognizing the heard sound and interpreting it is a complex task for our brain and is called auditory perception [5].

The auditory perception can be divided into different process steps [8].

- Detection
  A sound wave is within the human hearing range and can be heard.
- Discrimination
  The ability to differentiate sounds from each other correctly.

- Identification
  A sound needs to be identified and assigned to where it comes from and what the sound source is.
- Comprehension
  The information of the sound needs to be understood, so that the perceived message can be used.

Further studies, involving research and analysis of the functionality of the auditory system and the auditory cortex during the last years, have shown that the path to perceiving auditory information is quite complex.

Following the explanation of the auditory system in section 2.1, the acoustic stimuli which have been transferred to the brain, are divided into perceptual features. This enables us to split a mixture of auditory sources into distinct auditory objects. Therefore, we can differentiate the incoming sound stream and assign each sound to its corresponding source [5]. The brain's auditory cortex makes this possible by analyzing the perceptual components of sound, such as pitch ("height" of the tone), timbre ("tone color") and loudness [29]. By having defined the auditory object, it can be further converted into an abstract level of perception, which then makes it possible to include information from the brain to combine them with the perceived object. Then, this abstract representation can be interpreted to make use of the information [5].

However, the sound we hear might not be heard by other humans the same way since the level of perception is not the same for everyone. Different intensity levels and noise frequencies might have a stressful or more unwanted effect to some humans more than others. Therefore, also the definition between sound and noise can be of highly subjective nature [10]. Since the exact parameters that lead to noise cannot be clearly defined, a research branch called psychoacoustics deals with the "relationship between physical stimuli and the induced perception" [42]. The most important parameters of psychoacoustics are loudness (critical ranges of human hearing) and sharpness (proportion of high-frequency spectral components to total loudness) as well as roughness, fluctuations and strength (time and level differences in the sound signal, which modifies both amplitude and frequency) [10]. There are several ways to measure each parameter, e.g. by calculating the psychoacoustical perceived loudness with a method by E. Zwicker. Considering psychoacoustical parameters, it is evidently that there is a huge gap between the way sound is measured and how it is perceived by the individual, as other factors such as psychological experience, expectation and attitude have to be included into a measurement which concerns human perception [10]. Therefore, it is not enough to only include A-weighted levels when measuring noise since no adequate answer can be given by that [10].

2.3 Tinnitus

Tinnitus, derived from the Latin word *tinnire* (to ring), describes the perception of sound in absence of an external sound source [16]. Persons affected can sense a lot of different sounds ranging from a typical ringing noise to hissing, whistling, sounds analogous to animals like cicadas or crickets or complex sounds resembling human voices or music [12] [3]. Data from the National Health and Nutrition Examination Survey (NHANES) in the U.S. has shown that more than 50 million US adults with about 15 % between 60 and 69 years old reported having experienced any form of tinnitus [37].

Tinnitus can be classified into objective and subjective tinnitus. The former one is the more seldom occurring form of tinnitus, with only 1% of all tinnitus reported cases in the U.S. affected [2]. This is the case if another person can measure it. The more often occurring form is subjective tinnitus, which is only perceived by the person affected [3]. This makes it very hard for objective measurements, as there are only certain ways to examine the severity of the case and ways for further investigation and comparison rely on subjective impressions.

The main cause for tinnitus is hearing loss, which is often caused by reduced cochlear nerve activity. Beside otological causes, there are numerous risk factors, which include exposure to loud environments, head or neck injuries, acoustic shock traumas and high blood pressure. Since there is no general rule for what causes tinnitus initially, people with limited hearing capabilities can develop it as well as people with normal hearing. Generally, the likelihood for tinnitus increases with age up to 70 years [3].

Symptoms related with tinnitus are sleeping difficulties, emotional stress, physical exhaustion and difficulties staying in too noisy or very quiet environments. Severe degrees of tinnitus can lead to a large impairment to health including working disability, anxiety and depression [12].

Possible treatments for tinnitus can either reduce the intensity of the tinnitus by means of pharmacotherapy or try to weaken the disturbance of subjective perception with cognitive, sound or habituation therapy [12]. Also hearing aids proved to be a supporting factor when tinnitus comes in combination with hearing loss. Not only do hearing aids offer the possibility to increase the audibility of speech, they can also support their wearer by amplifying only specific non-related tinnitus frequencies which helps in decreasing the perception of these frequencies and aiming towards better recognition of speech as well as taking away the stress associated to them [12]. Other approaches using hearing aids include masking the tinnitus by using a sound generator, which usually creates white noise or pleasant ambient sounds based on the specific type of tinnitus to reduce its loudness and thus reducing the awareness [32].

2.4 Measuring Sound Levels

Converting sound intensity and sound pressure into a numerical unit can be a quite confusing task since the logarithmic scale differs to typical measurement scales. However, due to the way sound is sensed by the human ear, the logarithmic scale better depicts changes in loudness. A short introduction on how sound levels are measured and a connection to the human hearing are given in the following chapter.

2.4.1 Decibel

Sound travels in waves and consists of frequency and amplitude. The measurement unit of sound is Decibel (dB) with a logarithmic scale. That is the case because of the way our hearing works. The logarithmic scale starts from 0 and increases in multiples of 10 dB. An increase of 10 dB will sound twice as loud to our ears. However, an increase of 3 dB will already double the sound intensity. This can be explained by looking into what loudness and sound intensity actually means. While sound power is measured by the energy per united time which is emitted by a sound source, the sound intensity is the sound power per area. The intensity measured in dB is defined as $dB(I) := 10 * \log_{10}(\frac{I}{I_0})$ [27]. Loudness on the other hand is subjective and depends on the ability of the listener's hearing and characteristics of the sound like duration and frequency. For example, sounds with the same intensity but different frequencies do not necessarily need to have the same loudness. Figure 2.2 shows the coherence between intensity and loudness to our hearing. The unit phone describes the perceived loudness of pure frequencies. The perception of lower frequencies increases with rising intensity, as higher frequencies around 3 kHz can be perceived with even lower intensity [27].

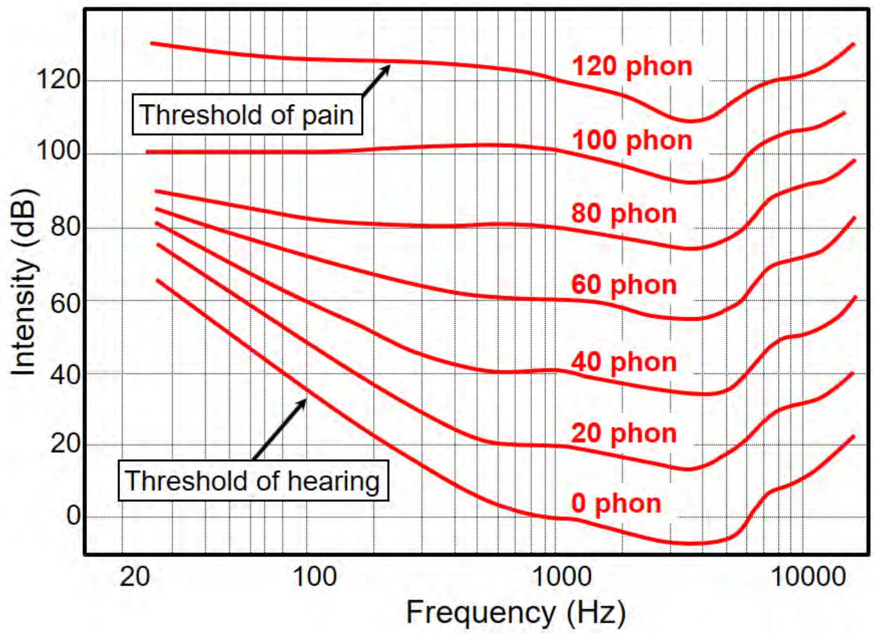


Figure 2.2: Loudness is different for the perception of our hearing depending on the frequency and the intensity [27].

2.4.2 Sound Pressure Level

To effectively use the decibel scale for measuring loudness or intensity, it is necessary to use a reference unit, as decibel itself is a relative scale and does not have a magnitude. To make use of the decibel unit regarding measuring sound levels, the loudness can be measured in dB(SPL). The unit Sound Pressure Level (SPL) is dependent on the atmospheric pressure and is measured in Pascal (Pa). It is referenced as 20 μPa, which was assumed to be the Threshold of Hearing (ToH).

This makes 1 Pa correspond to an SPL of 94 dB. An increase of 6 dB will double the sound pressure [36]. A comparison between the sound level of different sounds can be seen in figure 2.3. In the following work the term *sound pressure level* is abbreviated to *sound level*.

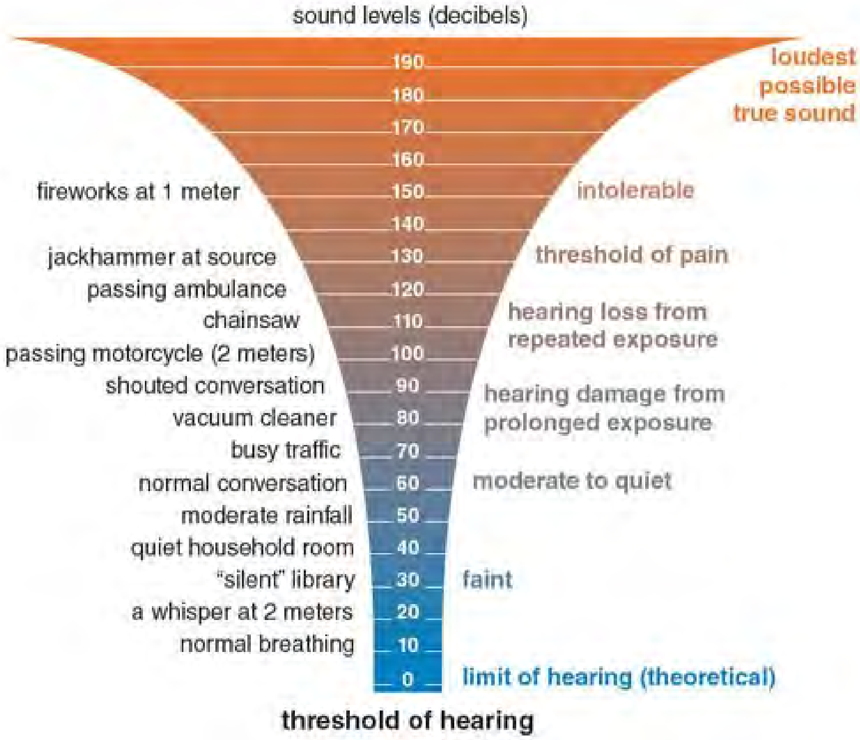


Figure 2.3: Comparison of different decibel values and their effects on hearing [39].

2.4.3 A-weighting

The way of measuring sound can differ according to its use. In the case of measuring for hearing-based applications, it is often measured using the A-weighted sound pressure level dB(A).

This scale focuses more on the way humans hear. As seen in Figure 2.2 frequencies are not perceived equally by the ear, as humans are only able to hear frequencies between 16 Hertz (Hz) and 16 kHz with the highest sensitivity between 2 and 4 kHz. Due to the minor sensitivity for low frequencies, a consistent scale would only provide limited meaning to noise measurements, as non-perceivable, lower frequencies would be given the same significance as higher frequencies which are perceived more. Weighting helps to directly show the effects of loudness concerning hearing damage [30]. The curve of the A-weighting can be seen in figure 2.4. A-weighting comes in use for sound pressure levels below 55 dB. For higher levels, B and C-weighting are chosen.

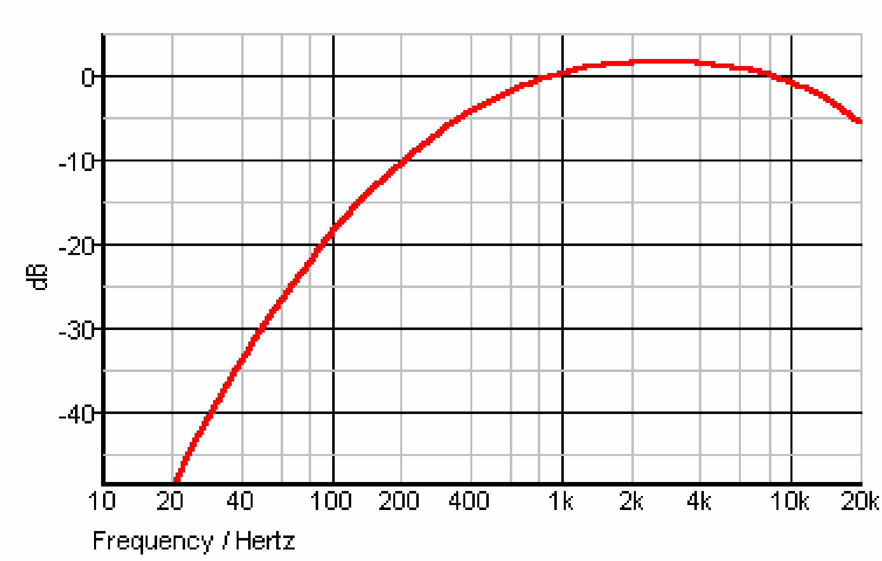


Figure 2.4: The A-weighting filter curve [36].

However, studies have shown, that the role of lower frequencies is much more important to physical and psychological factors than it is depicted with the A-weighting sound pressure level. Although high-frequency hearing loss is much more common, the effect of a vast amount of health factors for lower frequencies has been proven in recent studies, making the general use of the A-weighted sound pressure level questionable [30].

2.4.4 Sound Level Meter

Measuring the sound pressure with a decent degree of precision is achieved by using a SLM e.g. the one in figure 2.5. Basic devices have a range of +30 dB to +130 dB and can measure frequencies from 20 Hz to 8 kHz. Usable devices are low-cost and can be used for simple measurements, however, they have quite a high amount of tolerance range, which must be taken into account if an accurate measurement is needed. Dependent on the standardization class, devices have different tolerance limits for different reference frequencies. In the most-used frequencies between 20 Hz and 10 kHz lower Class 2 devices have an error-rate between ± 1.5 and 5.6 dB [15]. For basic sound measurements though the difference between a Class 2 and a Class 1 SLM is inconsiderable.

Figure 2.5: A sound level meter with a pop filter

The functionality of a SLM is achieved by using a microphone, amplifier, weighting network, rectifier, and a display. In the first step, the signal of the sound wave is converted into an electrical signal, which is then increased by using a preamplifier. Afterwards, a weighting network alters the signal by filtering its frequencies to adapt to a defined weighting standard. Most SLMs use the widespread A-weighting, which is discussed in chapter 2.4.3. Following this, the signal is amplified once more and converted from alternating current to direct current. Finally, the signal can be displayed [24]. This process is visualized in figure 2.6.

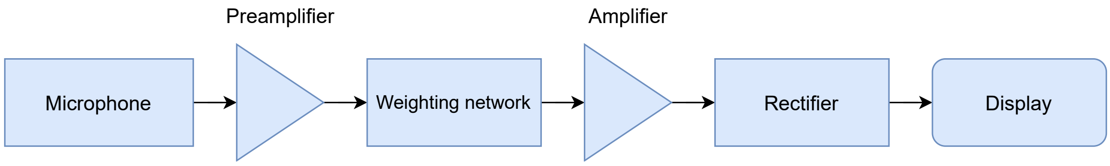


Figure 2.6: The functionality of a sound level meter, based on Kumar (2018)

2.5 Smartphone Sound Processing

As briefly touched in the introduction chapter, collecting acoustic data with a smartphone poses an ambitious challenge to its executor. To understand the difficulties one has to overcome, it is important to get a brief knowledge about how the actual functionality behind recording sound with a smartphone works and what stages a signal has to pass until it can be measured by an application. Faber (2017) has made some important research about the sound measurement possibilities and limitations of smartphones, which are summarized briefly in the following sections [9].

An acoustic signal can be recorded with either the build-in or an external microphone, which can be connected via the audio jack or wireless. Following this, the analog signal is transmitted through an Analog-to-digital Converter (ADC), which creates a numerical representation out of the measured signal. Lastly, it can then be delivered by the Operating System (OS) to requesting applications [9]. The process is visualized in figure 2.7.

All mentioned steps will manipulate the incoming sound signal until it reaches the application which measures it. It is important to state, that a smartphone was initially not designed and built to provide proper sound measuring tools, as it is mainly intent to be used as a communication device. Nevertheless, due to its vast amount of functionality and tools, it can be used for other applications as well, like loudness measuring. Most of the tools, which are build-in to modify the incoming signal are designed to enhance the sound signal to understand someone better while making a call for example. Since there was no need to get a clear amplitude signal, most systems did not provide a method to disable these modifiers in the past or even do

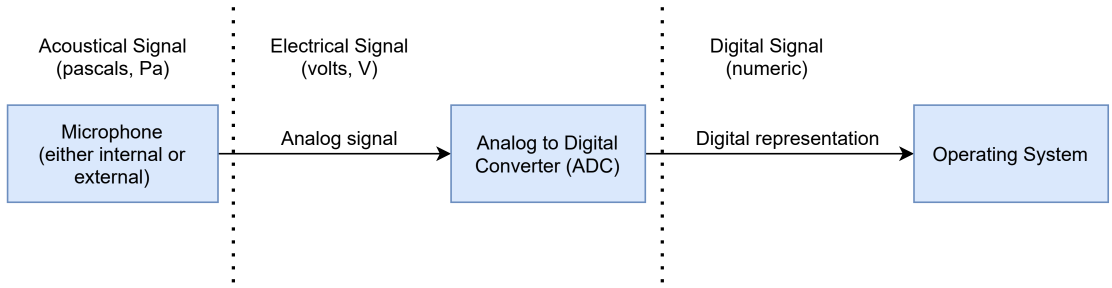


Figure 2.7: The process of converting an acoustical signal into a measurement value that can be used by a smartphone application, based on Faber (2017)

not provide it today. Moreover, the build-in hardware like the microphone does have limited quality as it is designed to fit into the phone which only has a small outlet, thus reducing its capability to record clear signals. This explains, why it is often not that easy to get a good measurement accuracy.

2.5.1 Microphone

In the first step the microphone is a very import factor, when it comes to deviations of precision.

In most cases, the build-in microphone is used for measurement purposes. If there are multiple microphones available, normally only one, which provides the best omnidirectional features is selected. Given the position the microphone is located, it can only record sound properly in a monodirectional way, even though its omnidirectional design [9]. This changes the recorded signal strength depending on where the signal originates, due to the acoustic shadow of the smartphone.

Better results have been achieved by using an external microphone. Possible options to connect can either be wireless (headset) microphones via Bluetooth. Other non-wireless microphones are connected directly wired to the audio jack or data port of the phone. Usually, wired measurements using the audio jack provide a higher quality than wireless [9].

2.5.2 Hardware

The sheer amount of different smartphone models are an important factor to consider when obtaining sound measurements. Different build-in hardware parts such as microphone or ADC, which are involved in signal processing, will manipulate the quality of the outcome. Furthermore, smartphone manufacturers do not rely on the same parts for all their smartphones, so the quality can differ for each smartphone model, too. Also, systems like Automatic Gain Control (AGC) can completely corrupt the signal, as they are normally used to keep the outgoing signal at a constant level independent of the incoming signal's amplitude to enhance the quality of phone calls [9].

2.5.3 Operating System

Just as the hardware does, the OS, which runs on the system, plays an important role when it comes to the quality of the signal a phone can provide. As the OS has direct control to the underlying hardware, it can often control to a certain part the way the hardware operates. Whether this helps to improve the outgoing signal does not only rely on the interface the hardware offers, but also on the Application Programming Interface (API) the OS provides [9]. If one of both is not available, the application on the top of the chain does not have any possibility to get an accurate signal.

For example, Apple's operating system (iOS) did constantly provide a high-pass filter to all microphone inputs, due to their limitation to only partly record frequencies below 200 Hz. Only iOS 6 in 2012 has added a function to disable this behavior and the in-build AGC via the system's API which now makes reliable measurements possible [9].

2.6 Related Work

As of the current state of writing, performing acoustical measurements is a niche topic since smartphones initially are not built for this. However, there is an increased need to measure sound levels of noise. Since it is necessary to measure sound levels as reliable as possible, several smartphone applications are available, which try to achieve a high degree of accuracy. There are a lot of studies which analyze the performance of these applications like Kardous and Shaw (2014) or Faber (2017) [17] [9]. Despite the increasing number of applications, only some of them provide decent results. Since for most of the applications with good results the complete functionality is unclear and most of them are subject to change, additional ways are needed to make sound measuring possible for the average user. To further investigate the acoustical behavior of smartphones in regard to measure sound in changing environments as for directional or omnidirectional recordings, Hawley and McClain (2018) provided valuable findings [13].

Moreover, this work provides a contribution to smartphone-based mobile crowdsensing studies which aim at gathering information about environmental data on a large scale. Additional work in this field has been done by Kraft et al. (2020) who presented an approach on how geospatial data could be helpful for concerned persons and tinnitus patients to track noise levels in their surrounding area [22]. For this purpose, an architecture was developed to process large amounts of data effectively and give the user a visual overview of loudness data that was gathered by other users [21].

3 Concept

The work of this thesis builds up on the already well-established Track Your Tinnitus application, which can be used to easily track tinnitus perception by using a smartphone. At the moment this involves observing the progress of the user's tinnitus measured by questionnaires throughout the day. This enables the affected persons to measure fluctuations and working out daily habits and coherences, that have a positive or negative impact on the perception of tinnitus [20].

To further extend this well-functioning system, improved noise measuring with a smartphone will be introduced. Noise poses an important factor for broader tinnitus evaluation, which will enable the system to gather more precise information about the actual situation, hence using the noise measurements in combination with user given data for extended evaluation. Recorded measurements e.g. environment noise include data such as loudness and frequencies, which can give direct feedback to the user, thus supporting him to gain more knowledge of his specific tinnitus. Consequently, including sound data for tinnitus studies allows analyzing the precise characteristics and effects tinnitus has on the health of the affected person.

For uncomplicated sound measuring purposes the smartphone microphone should be the primary way for measurements, as the aim for further studies is to have a lot of participants who can join the study without the need for additional hardware. However, due to better omnidirectional performance, an external microphone will be tested and compared against the qualities of the internal microphone. This should compensate for the different types of smartphones, the various analysis procedures and the position of the smartphone while recording.

Recording with a smartphone seems like a trivial thing, since every conventional smartphone is equipped with a microphone and a lot of apps make heavily use of this technology to send voice messages or record videos with audio. Therefore, the technological part is already established. However, measuring sound levels with this equipment is a completely new application area for which neither the smartphone nor the hardware was designed for. Hence, a workaround by developing an adequate applicable recording method and building an application around it, that allows to take precise measurements is something completely new. Measuring noise with a smartphone, which would enable almost everyone who has a smartphone to do so, is only a niche sector at the moment and a standardized process to cope with the hardware differences of the smartphones has yet to be developed.

Altogether, the following work will examine the characteristics of different smartphone types, and the use of internal and external microphones for sound measuring purposes by using different frequencies and changing volume levels. Additionally, the results of the measurements will aid to create a method, that ensures comparability between the recordings and to achieve necessary modulations to adjust the measurement abilities of various smartphone types. This includes setting up a standardized concept to guarantee objectivity on one hand, and simultaneously working close to reality to reach a high degree of validity.

One way to achieve this for instance, is the positioning of the smartphone's microphone, as it is mostly located at the bottom of the phone. Most of the time, the microphone is facing the user, which results being in the acoustic shadow without acquiring a direct source of the signal, but an altered and mitigated version as seen in figure 3.1. As a consequence, this does not only influence loudness, but also implies recording a mutated frequency spectrum, which could both interfere with a reliable result. Nonetheless, it has yet to be proven how much influence the limited omnidirectional features of the internal microphone have.

Other important factors are effects of hardware and software differences of smartphones as already discussed in section 2.5. It has to be shown, if and to what extent different microphones and varying processing hardware account for deviating data outcomes. To create a whole spectrum of multiple factors, different frequencies have to be tested together with multiple volume levels to find similarities and distinctions.

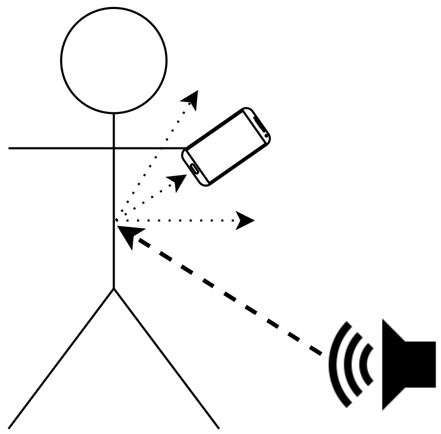


Figure 3.1: Most of the sound waves are reflected and mitigated before they are recorded by the smartphone.

4 Experimental Design

Measuring amplitudes with smartphones poses a challenge to a lot of different tasks. The first issue that needs to be handled are the various amount of smartphone models, which measure amplitudes on a separate scale. That is the case why a calibration of all models is necessary. This calibration topic will be evaluated with different approaches, as there are multiple possible methods to execute a calibration. As part of the methods will be tested and evaluated in the next chapter, the following text should give an introduction on the ideas, the functionality and use-cases of the method. Additionally, experiments are being conducted, that analyze the acoustical modalities of the involved smartphone microphones. This will be done to give an overview of different aspects, that need to be considered when amplitude measurements are performed by a smartphone. This includes acoustical behavior tests, which are performed to evaluate how results vary for different situations. Since the sound environment varies during daily life, some frequencies may be recorded differently than others. This is why varying sound samples are being part of the evaluation and will also be compared against each other.

4.1 Calibration Techniques

As introduced earlier, smartphone models have different characteristics when it comes to recording audio, which has an effect on the measured amplitudes. However, this mostly does not have an impact on the comprehension of the recorded data (e.g. listening to a conversation that was recorded with a low amplitude can be compensated by increasing the volume). On the contrary, when amplitudes are measured this fact does play a huge role, as these amplitudes cannot be converted directly to a standardized scale. Hence, a need for calibration is obviously necessary.

A calibration of a smartphone recording can be achieved in many possible ways. To decide which calibration method should be chosen, it is necessary to assess the objectives of the project. Particularly, the required accuracy of the measurement and the effort willing to apply in order to receive the desired degree of accuracy, should be considered accurately. Evaluation of several ways for calibration has shown that gaining a higher degree of accuracy is not possible without the use of external hardware such as a calibrated SLM or a tone generator, which can either measure the amplitude on a normalized scale or play a tone with a standardized volume. However, as this greatly increases the difficulty for measurements that only require a rough estimation of the measured sound level, other means of calibration are required, that reduce the necessary work until a measurement can be performed. Especially when crowd-sourcing shall be performed and many people with different technical skills are put into place, it is inevitable to keep the process of calibration as uncomplicated as possible (or even perform it on behalf of the user as seen later) to eliminate any high deviation errors in calibration since this would decrease the accuracy of the whole data set. Consequently, having diversified opportunities for calibration depending on purpose of use has its justification.

In the following section two calibration methods will be covered, that have been classified applicable for general use. For every method there will be a way on how the calibration can be performed in detail. This includes not only a detailed description of the functionality, but also associated assumptions, that will be verified to evaluate the further use of the respective method. Additionally, a theoretical method which could serve as a convenient method for future calibration uses will be presented.

4.1.1 Basic Calibration

To perform a straightforward sound level measurement with a smartphone, no external hardware is necessary. Nevertheless, getting an estimate of the sound level requires performing a calibration of the microphone. To keep this as simple as possible, Dr. M. Ziegler outlined an approach which involves an easy-to-use method for a simple calibration in combination with the application "Spaichinger Schallanalysator" [1]. The calibration can either be performed by tearing apart sheets of

[1] https://spaichinger-schallpegelmesser.de/schallanalysator.html

copy paper, or with a tone which has a consistently equal volume that must be verified beforehand. Since the former method is usable without any additional hardware, the main focus will be put on it. The following summary is taken from the instruction document of the application [40].

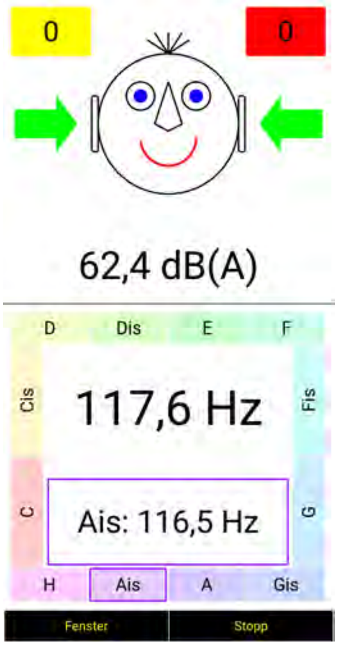


Figure 4.1: The application "Spaichinger Schallanalysator" in measurement mode, displaying sound pressure level, base frequency and corresponding tone [40]

A screenshot of the user interface of the application is given in figure 4.1. The current volume in dB(A) is given as well as the base frequency with its corresponding tone. Additionally, the app offers more features like measuring the sound intensity in W/m^2, the effective sound pressure in Pa and much more.

A calibration with the paper tearing method consists of two steps. First, the smartphone or an external microphone is placed approximately 90 cm (which equals three length of paper sheets) away of the person who will perform the calibration. It is important to state, that the smartphone should be positioned at the same height as the paper that is getting teared apart and that the direction of the smartphone microphone is aimed at the person. The calibration can then be started by letting the smartphone measure the amount of background noise so that a difference calculation can be performed afterwards. Also, a sound level correction value can be given depending on where the calibration was performed. The second step of the calibration process includes tearing apart 10 DIN-A4 sheets of white copy paper ($80\ g/m^2$) one after another. The application measures the sound level of every rip and calculates a standard deviation, which is about 2 dB afterwards. The smartphone is then ready to be used for basic sound level measurements. Figure 4.2 shows the setup that was established for the experiment.

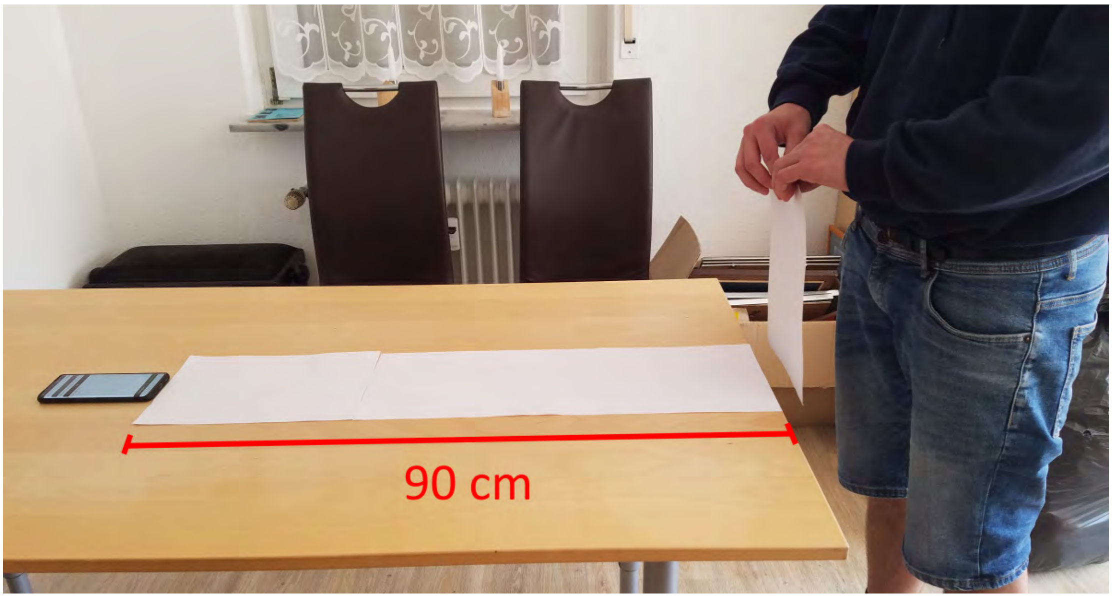


Figure 4.2: Execution of the calibration by tearing apart paper

The method works by the fact, that the noise, which is created by tearing apart paper at a specific distance away from the microphone, has the same loudness for every performed paper tearing. Therefore, the recorded amplitude will serve as a comparison value to calculate amplitude values to dB(A), since the noise of tearing apart paper in dB(A) is known to the application. This enables the application to calculate a compensation value for later measurements, based on the loudness of the performed calibration.

Anyway, depending on where the calibration has been performed, a sound level correction value can be applied at the time of calibration. This is due to the fact, that the sound which is recorded by the microphone consists of primary sound from the sound source, as well as sound that has been reflected by walls and furniture. As diffused sound superimposes the direct sound of the loudspeaker, the correction value needs to be increased for small rooms. An approximate value for different type of rooms would need to be gathered in further experiments.

Moreover, if an external microphone is used, the possibility is given to manually configure the frequency response of the microphone. However, as this is already covered in the next section 4.1.2, no further research will be done in this direction with the application.

Another way to calibrate the microphone with the help of the application is offered by using a tone generator, which creates a 1000 Hz tone between 75 and 95 dB or by using an external sound source and a reference SLM, that can confirm the correct decibel value. Since this is also part of the next chapter 4.1.2, it will not be elaborated further at this place.

The paper tearing technique was then reconstructed and examined for its accuracy. To do so, different frequencies were outputted on a Bluetooth loudspeaker and the sound level in dB(A) was measured by the smartphone using the already mentioned "Spaichinger Schallanalysator" application. Beyond that, an SLM was used next to the smartphone to validate the measurements. Thus, both results can be analyzed and compared. This approach is shown in figure 4.3.

Figure 4.3: Measuring the sound level of different frequencies after calibrating the smartphone with the paper tearing technique

This experiment will check the following hypotheses:

1. Calibration of the microphone will lead to reliable measurements
2. Different frequencies will not exceed a certain deviation level
3. The deviation level varies for different frequencies, so each frequency is recorded with various sound levels

4.1.2 Frequency-based Calibration

Every sound consists of several frequencies, which are all measured with varying intensity by a microphone. This fact is referred to as the *frequency response* of the microphone. Each microphone has a defined frequency operating range, which determines what frequencies can be recorded by it. However, this does not imply that all frequencies in this range are recorded equally. This is why a frequency response curve is necessary to specifically compare the capabilities of microphones in detail. The frequency response is specified by testing the entire frequency spectrum of the microphone and then measuring the amplitude of each tested frequency which yields a microphone-specific frequency response curve. An example for a curve is given in figure 4.4.

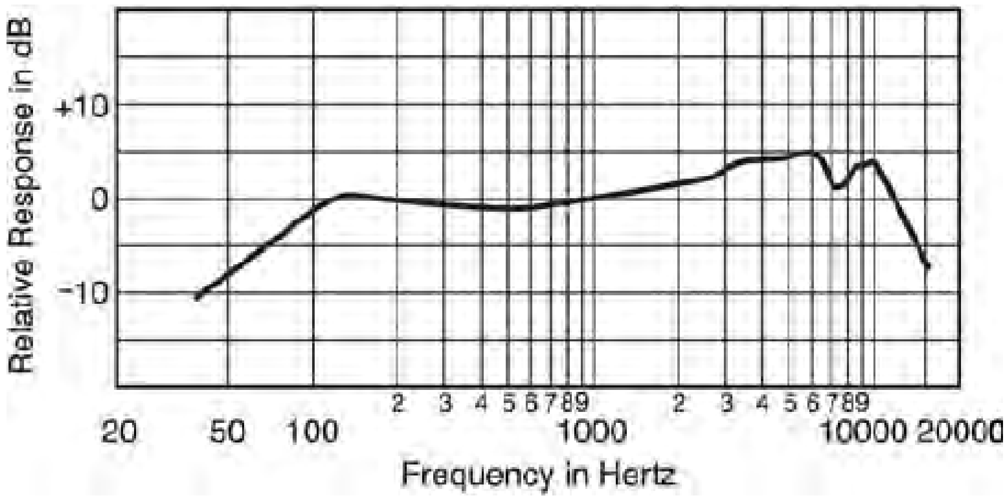


Figure 4.4: A frequency response curve of a microphone. The output varies depending on the frequency. [6]

Since smartphone models are not equipped with the same type of microphone, sound level measuring with a smartphone microphone will undoubtedly face the same issues, meaning they will certainly not only differ in the frequency ranges, which can be sensed, but will also have differences in their frequency responses.

Aside from the deviations of the frequency responses, comparing loudness with smartphones is additionally made complicated by the specific characteristics of the smartphone microphone and the way the sound is processed by the system, which is further discussed in chapter 2.5. This leads to the fact that measured amplitude values cannot be converted to standardized decibel values without previous calibration.

Nevertheless, by creating a frequency response curve, the deviations of every frequency as well as the overall amplitude shift caused by the microphone and the system characteristics can be represented together. Moreover, by comparing frequency response curves of different smartphones with each other, an adjustment can be made for each frequency. This makes it possible to measure how much smartphone microphones deviate from each other independently of the hard- and software that is being used.

A calibration is performed by calculating a specific difference value per frequency. In other words, a specific frequency curve c for smartphone A shall be referenced to as c_A. This curve shall be compared to c_B, the frequency curve of smartphone B. Each frequency curve consists of multiple frequencies n that shall be tested. For the following experiment, n is defined as $n \in \{500, 1000, 2000, 4000, 5000, 7000, 8000, 10000\}$. After the creation of the whole device curve, each frequency will have a corresponding measured amplitude.

$$c_A(n) : Amplitude\ of\ smartphone\ A\ for\ frequency\ n$$

Advantages of this method should be a good precision since the measurement deviation of all frequencies are taken into account for calibration.

However, other than for the basic calibration in 4.1.1, additional hardware is necessary to perform this type of calibration approach. Firstly, to play the different frequencies it is necessary to have a loudspeaker which can play all the frequencies that have been defined. Secondly, a calibrated reference device is necessary.

Figure 4.5: Performing a frequency calibration by the help of a sound level meter

This includes either a SLM as shown in figure 4.5 to measure the sound level for which the calibration should take place, or alternatively by using an already calibrated smartphone. This works by finding out the sound level with the help of the already calibrated smartphone before the start of the calibration.

In both cases a smartphone or another device needs to be paired to the loudspeaker to play all the defined frequencies, but the smartphone that should be calibrated can also take over this role. The same process must be carried out for each smartphone, that shall be calibrated.

Luckily, once a frequency response curve $c(n)$ has been created for a smartphone model, it can be shared with other smartphones of the same model as they are supposed to have the same acoustics. In this case a cloud database could be established to store the model specific $c(n)$ values. For every new non-calibrated smartphone the model identifier could be compared with the database and if available, automatically download the calibration values. Therefore, only one calibration for each smartphone model would be required.

The idea is based on the company Eardial[2], which already uses a similar approach

[2]https://eardial.com/calibration/

to calibrate smartphones, however, without the focus on different frequencies.

Beyond testing for multiple frequencies, the sound level plays an important role. It is necessary to calibrate all smartphones with the same defined sound level, so the sound level of a starting frequency should be constant and verified before the start of a calibration attempt. Therefore, an initial volume calibration at about 80 dB(A) for 1000 Hz must be made initially by using a SLM or an already calibrated smartphone.

However, the extent of the sound level regarding the measured amplitude for different smartphones is unknown. To further analyze the effects of loudness on the measurement accuracy, an Android application has been developed to test the frequency responses of microphones at different volume levels. For this purpose, the smartphone must be paired to a Bluetooth loudspeaker, which is placed at a 20 cm distance away from the smartphone microphone. Then, a specially defined list of frequencies is played one after another for 2 seconds each with a 1 second pause afterwards to measure from silence again. Furthermore, to calculate the deviation level for each frequency, several volume levels are analyzed, too. Therefore, the frequency output starts at the lowest selectable volume level of the smartphone and is then gradually increased until 15, which is the highest sound level for an Android smartphone. During the output of the frequencies, the smartphone measures the amplitude of the frequency that is played. The process of outputting and recording the amplitude has been developed to run fully automatically. This should provide knowledge about the impact of loudness on the overall amplitude shift.

The following hypotheses will be examined in combination with this experiment:

1. Varying frequencies have an impact on the recorded amplitude
2. Different smartphone models do not reflect varying frequencies evenly in amplitude.
3. Smartphones do not represent rising sound levels evenly in amplitude.

4.1.3 Internal Calibration

Another calibration approach is presented by performing a self-calibration with a smartphone. However, as this method deals with the internal sound processing of the OS, it is unknown if it can practically be used at the time of writing, therefore it is only described theoretically. Nevertheless, there might be a chance to develop a calibration method out of this idea later since it should be very precise and would not require additional user effort.

To get a practical use out of the previously described methods, a lot of time-consuming work is necessary to be done by the user. Beyond that, additional items are necessary like the paper sheets for the method presented in 4.1.1, a SLM or a calibrated smartphone shown in 4.1.2. This needs to be done to obtain a reference volume where all executions of the experiments can refer to. However, for a self-calibration this is not necessary, as it is only performed internally by the smartphone itself.

To perform a calibration without relying on a monitored volume, the application needs to have access to the digital sound input interface. This enables the application to pass testing frequencies with a fixed volume directly to the digital input processing of the system without using the loudspeaker and microphone. Previous calibration methods always relied on outputting sound by using a loudspeaker or creating it in other ways and then recording it via the microphone. This always involves an analog to digital conversion, which is prone to measurement errors and involves the use of an ADC, that alters the input unpredictably.

By the method of passing digital sound directly to a digital interface, all previously necessary digital to analog conversion and a re-conversion into a digital signal would thus be skipped. The schematic for this approach can be seen in figure 4.6. Any sound leaving the operating system for calibration purposes would directly be diverted to the digital input line, to avoid any conversation interferences. This pure digital to digital interface would allow identifying differences in signal conversion and internal processing in a more precise way and correct them.

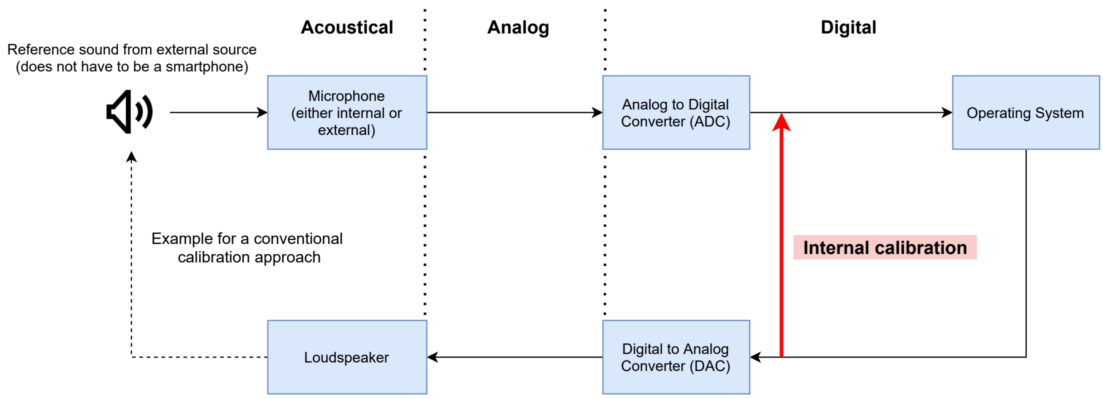


Figure 4.6: Schematic of a calibration approach with a pure digital-to-digital interface

4.2 Acoustical Behavior

To achieve knowledge about the behavior of smartphone microphones, a variety of experiments will be performed. An overview of each experiment is given in the following section.

4.2.1 Noise Samples

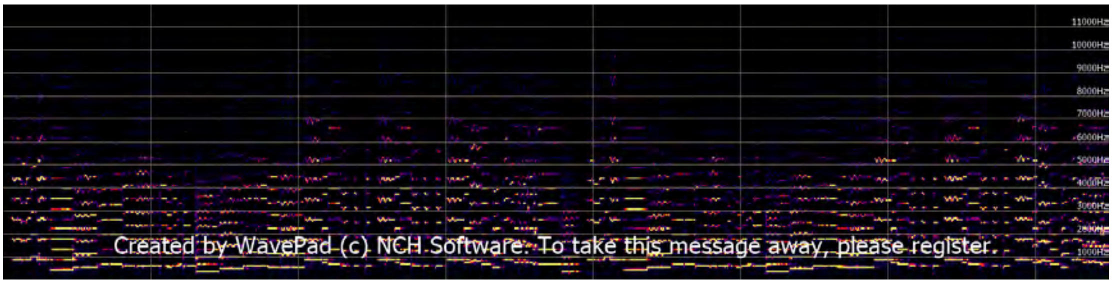
To test the overall microphone recording performance, four different audio samples were selected varying in frequency, tone and base. This is done to figure out if and to what extend microphones record volumes for various frequencies differently. Analyzing the audio samples is performed by utilizing the Time-based fast Fourier Transformation (TFFT), which transforms the input signal of a time domain into a frequency domain. This allows comparing the distinctions of the characteristics of the audio samples. All samples are uncompressed and lossless WAV files, which are shortened to a duration of about half a minute. All samples are free to use and taken from the website soundeffects+ [3]. Every audio sample will be shortly evaluated below. The figures with the time-based fast fourier transformation of the

[3] https://www.soundeffectsplus.com/

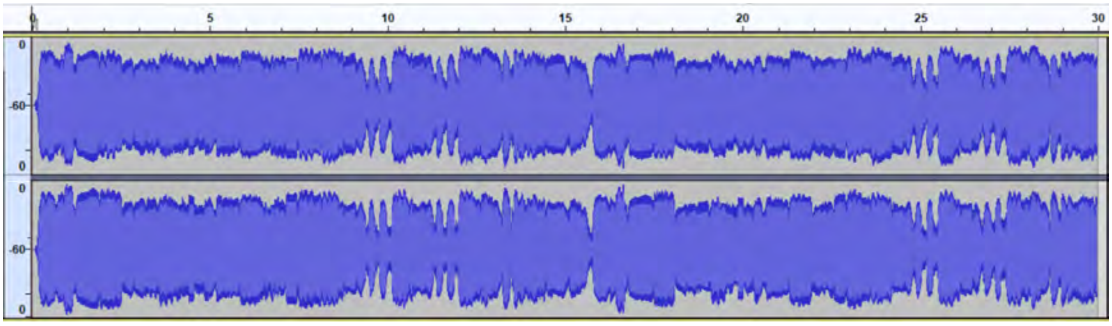
samples were made with the software WavePad[4], the figures with the audio spectrum of the samples are created with the help of the software Audacity[5].

1. Violin

The first sample is 30 seconds long and consists of a violin only, playing a part of the Minuet from String Quintet in E Major, Op. 11, No. 5 by Luigi Boccherini. As seen in figure 4.7a the sample consists of a tone pattern with mostly small, high intensity tones ranging from 0 to 5000 Hz. Figure 4.7b shows large fluctuations of the loudness.



(a) Time-based fast fourier transformation (TFFT) of a violin playing



(b) Audio spectrum of a violin in the relative dB unit

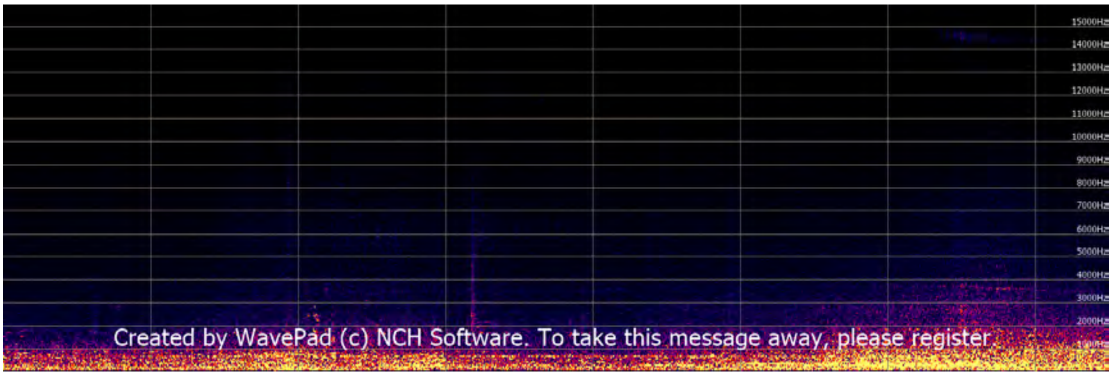
Figure 4.7: Characteristics of the violin sample

2. Street Traffic

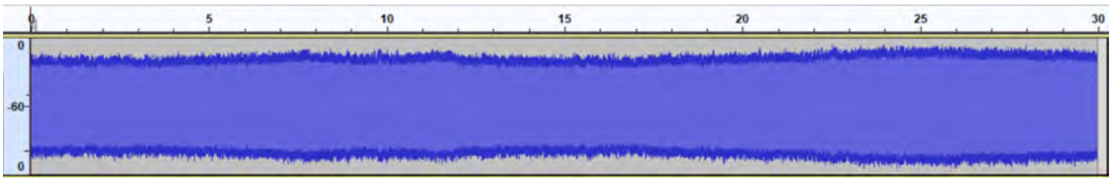
The second sample also has a duration of 30 seconds and includes mostly of street noise caused by traffic. Additionally, some chatter here and there can be noticed. The TFFT graph in figure 4.8a shows a loud and stacked frequency image for smaller frequencies up to 1000 Hz, which represent the motor sounds. Higher frequencies in the range from 1000 to 4000 Hz are

[4]https://www.nch.com.au/wavepad/fft.html
[5]https://www.audacityteam.org/

quieter and less dense. Analyzing the loudness of the traffic noise in figure 4.8b shows a changing volume, however this change is performed steadily and no fluctuations in a short time can be observed.



(a) TFFT of street traffic

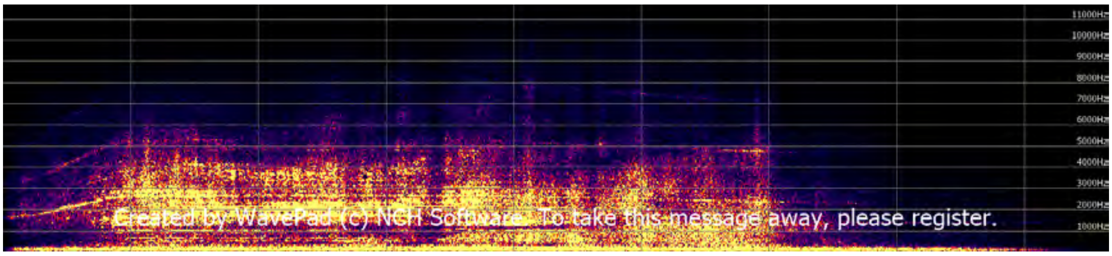


(b) Audio spectrum of street traffic in the relative unit

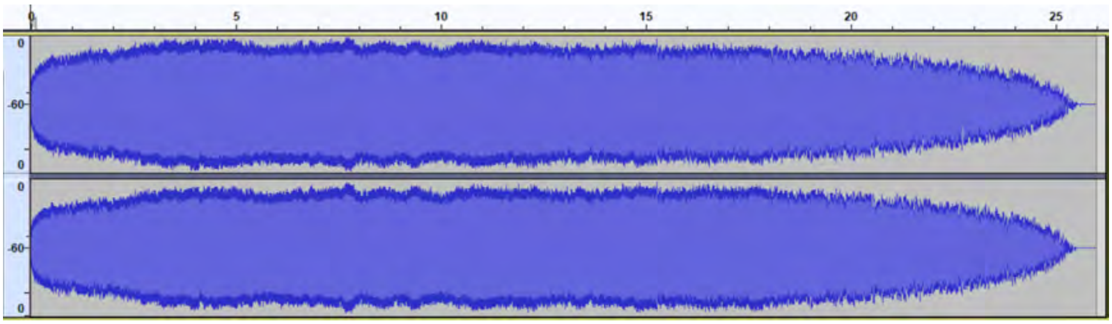
Figure 4.8: Characteristics of the traffic sample

3. Airliner Takeoff

The third sample is 26 seconds long and consists of a fixed microphone recording heavy sound of an aircraft engine spooling up to takeoff thrust with decreasing sound level as the aircraft accelerates away from the microphone. The transformation in figure 4.9a shows a base-heavy and low frequency pattern, which is measured continuously throughout the recording. Two increasing lines can be seen, one starting from 1800 Hz increases up to 2900 Hz and the other higher one starts at 3000 Hz and increases to 6000 Hz. Increasing frequencies up to 10 kHz can be measured at about 7 seconds until 13 seconds, as the aircraft passes by the microphone. Afterwards the distance to the aircraft increases, hence higher frequencies are declining. The loudness of the airliner can be seen in figure 4.9b, it shows a gradually increasing loudness until the turbines are fully ready, then some regularly occurring small fluctuations, until the aircraft takes off. Thereafter, the sound decreases rather fast.



(a) TFFT of a large airliner taking off

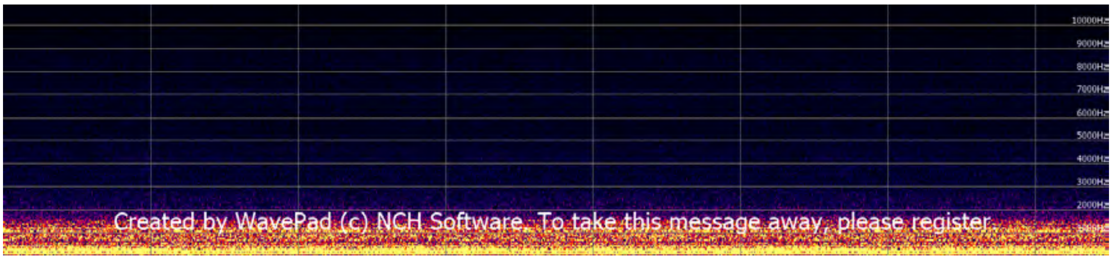


(b) Audio spectrum of a large airliner

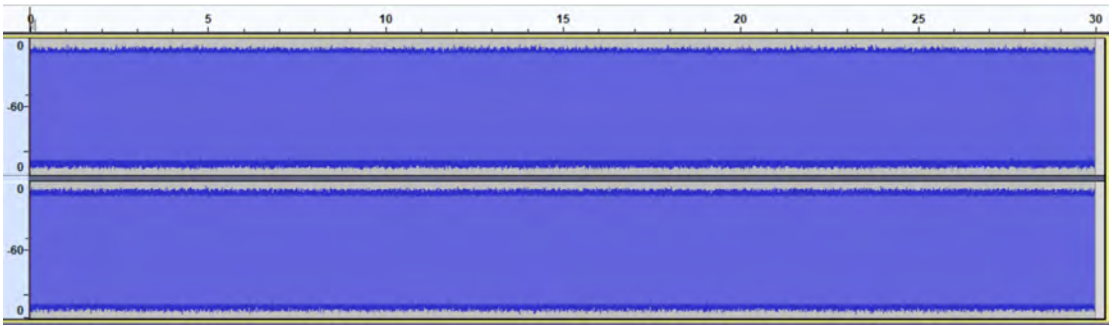
Figure 4.9: Characteristics of the airliner sample

4. Helicopter

The fourth and last sample is a 30 seconds long noise caused by a helicopter. As seen in figure 4.10a, due to the rotation of the rotor blades, it creates a loud and bass-heavy noise between 0 and 2000 Hz. Additionally, a lot of higher and more quiet frequencies up to 9000 Hz can be measured, which is created by the oscillating wind. Other than the first samples, this sample does hardly have any variations, so it provides a good comparability source for some experiments later. The audio spectrum of the helicopter is visualized in figure 4.10b. Here, a generally even loudness can be observed with no visible fluctuations.



(a) TFFT of a helicopter



(b) Audio spectrum of a helicopter in the relative

Figure 4.10: Characteristics of the helicopter sample

4.2.2 Recording Environment

For the recording of the samples a specific experimental set-up was created. This included setting up a sound system in stereo mode with two side and one center speaker to achieve a wider angle of sound source. The center speaker was placed in the middle directly in front of the smartphone. The two side speakers are placed 40 cm to the side and turned 45 degrees inwards. To eliminate possible post processing of the sound system as much as possible, any audio enhancing modes and bass amplifier were disabled. The samples were transferred from a laptop over an AUX IN cable.

To conduct experiments from different distances, markers were installed on the floor. Every meter, stripes of different color were installed. Due to limited room space about 5 meters were covered.

During the experiments, the ambient noise was minimized as much as possible to about 30 dB(A).

4.2.3 Technical Requirements

For the process of measuring sound levels with a smartphone, a custom android application was created.

One feature of the application was a selection of the recording mode the application should use for the upcoming measurement. Although, Android only mentions the `MediaRecorder` class on their official documentation for recording audio[6], there is in fact one more possibility to record audio, which includes the `AudioRecord` class. The differences are rather complex. `MediaRecorder` only offers a straightforward solution for developers who just want to record audio and not process it further in any way. Therefore, data is directly written into a file without any possibilities to intervene during the process of recording. Nevertheless, in terms of measuring sound levels, `MediaRecorder` does include the `getMaxAmplitude()` method, which allows to receive the maximum amplitude recorded since its last call. After some tests, it seems that the return value is converted to signed 16-bit integer values (0 to 32767), which correspond to the sound pressure recorded at the microphone. However, the Android documentation does not explicitly state this[7].

On the other hand, `AudioRecord` offers a greater number of choices, as it provides the raw sound stream in byte format, which needs to be compressed manually. Even though this seems rather obstructive since it makes the whole process of just retrieving the amplitude more complicated, it is actually a huge benefit, as sound compressing can be accomplished by the app directly. Hence, there is no need of being dependent on the unknown underlying algorithm of the `MediaRecorder`. Additionally, small experiments on different devices have shown, that retrieved sound levels are much higher, when using `AudioRecord`, which leads to a higher accuracy when sensing small level changes. A comparison between both modes is shown in figure 4.11.

However, both classes support the deactivation of any possible activated AGC that usually normalizes any incoming signal to the same amplitude, which would make any sound level measurements pointless.

[6]https://developer.android.com/guide/topics/media/index.html
[7]https://developer.android.com/reference/android/media/MediaRecorder

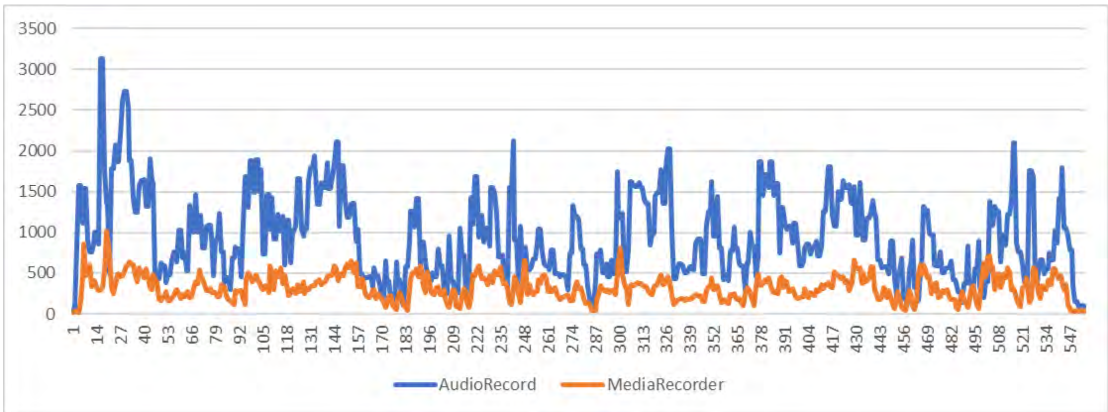
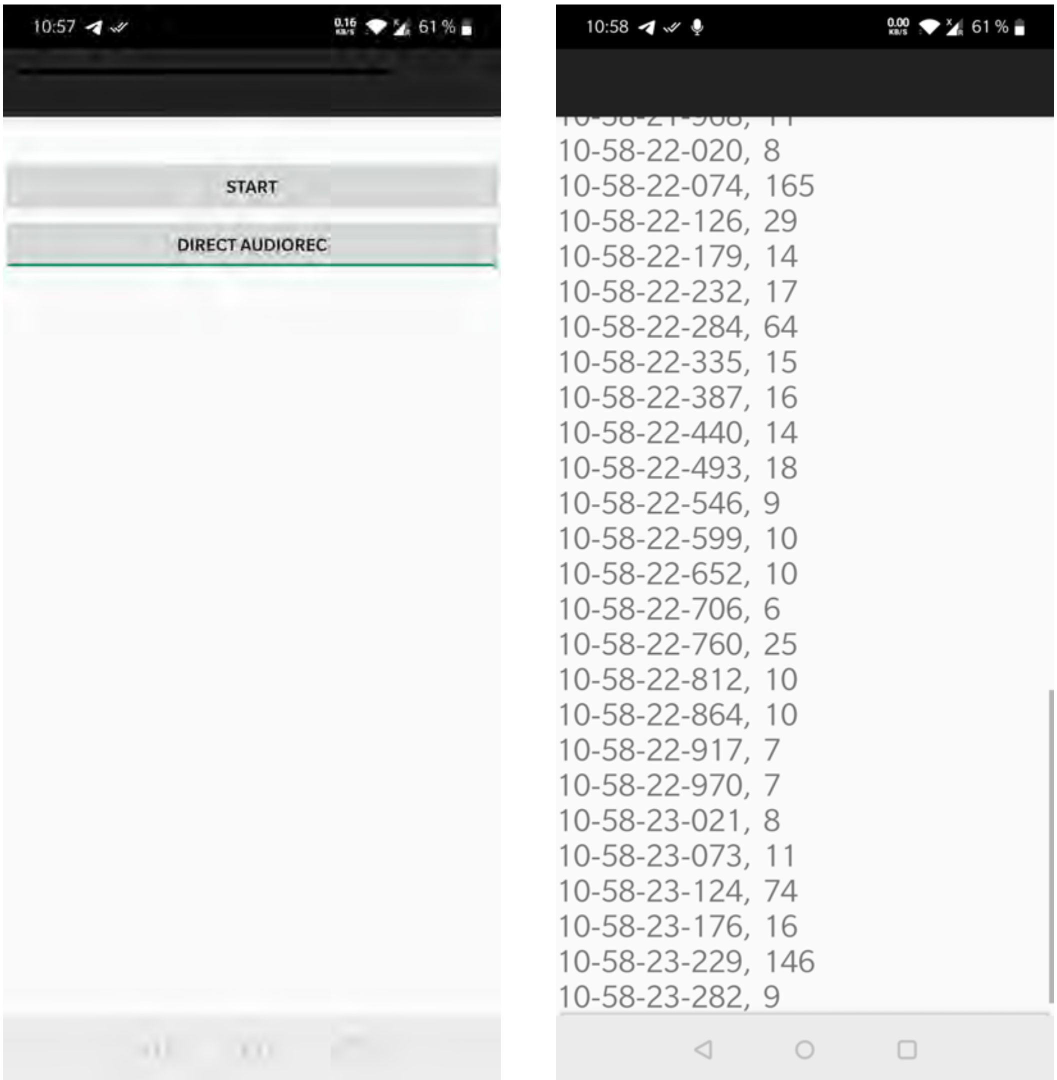


Figure 4.11: Different amplitude levels can be measured, depending on the selected recorder mode.

Before the start of a recording, different parameter need to be set. This includes the audio sampling rate, audio encoding bit rate and the audio channel. The `MediaRecorder` class also includes setting an output format and the audio encoder, which should be used since the recorded data is directly encoded and saved into a file. On the contrary, `AudioRecord` allows setting an audio format, which is set to a linear Pulse Code Modulation (PCM). The application automatically selects a representation of a 16-bit signed integer, or if unavailable, it falls back to 8 bit. Same applies for the sample rate, which has a default value of 48000 and will fall back to smaller values, if higher ones are not available. The number of channels is set to mono on default, to increase comparability, as only some smartphones are capable of stereo recordings.

Furthermore, the application measures the current sound level in both modes every 50 milliseconds for 40 seconds and logs the amplitude including a timestamp in a CSV file. The uncompressed audio data is also recorded in both modes and saved if further analysis needs to be performed. Screenshots of the application can be seen in figure 4.12.



(a) The start screen with a toggle switch to start a direct recording with AudioRecord or the rather unsophisticated MediaRecorder.

(b) The application records the amplitudes

Figure 4.12: Screenshots of the self-developed application to measure amplitudes

4.2.4 Monodirectional Behavior

In the same way loudness does, distance is having an impact on the measured sound level, as sound waves are traveling longer until they are recorded by a microphone. To find out what extent distance has on the measured sound level an experiment will be conducted where the distance from the acoustic source to the microphone will be increased consistently.

The monodirectional behavior experiment will be performed in a way to achieve a high degree of realism to serve as a reference statement for future studies. Therefore, three speakers are positioned in one line in front of the person taking the measurements to get a distributed sound source over a range of about 1 meter. This is due to the fact, that typical noise is mostly not released from a selected point but from a broader range. Additionally, it is assumed that the source of noise originates at or below of the ears of the person and that the smartphone is generally placed lower than the head. As a general rule, it shall be defined, that the most frequently taken posture for recording measurements is a person standing upright, with a smartphone in the (right) hand at the height of the thorax region and with the smartphone microphone facing the body.This posture with its common position of the smartphone was selected because it can be found very frequently in everyday life of a lot of persons. Therefore, having this height of the smartphone might help to receive more comparable results for the analysis of data of future experiments and studies. Since this effect was prioritized, no preference was given to measure the exact loudness coming out of the speaker. In conclusion, this posture can serve as a general standard for future measurement research since it has been defined to best resemble the conditions where future measurements should be taken.

However, due to the individual sound characteristics of the narrow and tall room where the experiment was conducted and due to the fact, that no sound absorber or other precautionary measures were installed to mitigate sound reflection, the outcome will result in a lot of inevitable deviations. Therefore, the results cannot be compared to similar experiments taken in a sound laboratory, since the experiment was not conducted in perfect conditions. Nevertheless, it may serve as a general orientation for future measurements, since the results should still provide significant data.

By conducting the experiment, the following hypotheses shall be validated:

1. The measured sound level will decrease with increasing distance.
2. Decrementation of measured sound levels will occur proportionally with increasing distance. Only minor measurement deviations might occur.

4.2.5 Omnidirectional Behavior

The validity of sound level measurement likely depends heavily on the direction the sound is coming from. This is assumed due to the fact, that smartphones usually only have one primary microphone, which is positioned at one side of the device. Although, often smartphone manufacturers claim their microphone to be omnidirectional, this cannot be achieved reliably because of the construction type of smartphones. This fact has already been investigated in other studies like Faber (2017), therefore the focus shall be put on the posture in which the smartphone is held. This can make a difference, due to coverage of the microphone by parts of the body like the hand or the upper body, when facing away from the sound source.

To investigate this issue further, the following experiment was conducted. One person stands in front of a speaker with a smartphone in his hand. Then he rotates around his own axis to let the sound hit the smartphone from different sides. As the sound is coming from the front most of the time, the smartphone itself deflects part of it and a lot is reflected by the body of the person into the microphone. However, when the person is turning to the side by 90 degrees, sound will pass by the microphone unchanged, as there is only little reflection, but it will not directly hit the smartphone. Turning by 90 degrees further to 180 degrees, the person is standing between microphone and sound source, so amplitude values will most likely be much lower than before. The last 90 degree for a total of 270 degrees is similar to the 90-degree turn, nevertheless as the smartphone is held in the right hand, sound waves could slightly be blocked by the right arm, although it is unknown if this deviation is measurable with the given experimental setup.

Besides testing the results of the internal smartphone microphone, the same experiment shall be conducted with an external microphone to compare the results. It can be assumed, that even a cheap, external microphone with omnidirectional capabilities, is performing better by delivering much more precise results, than the build-in microphone of the smartphone, since it is less likely, that the external microphone gets covered up, when attached correctly.

In summary, the following hypotheses can be made:

1. A smartphone microphone reflects direction differences in the sound level it measures depending on a predefined posture.
2. Sound coming from another direction then the smartphone microphone points to, will be measured with a lower sound level.
3. An external microphone provides steady results, independent on the direction of sound.
4. According to 1., 2. and 3. an external microphone will deliver more precise results than the internal microphone with variable direction of sound.

5 Results

During the process of collecting data for the experiments, various smartphones running the Android operating system were used. To represent a selection of varying hardware, smartphones of the three companies Samsung, Huawei and Oneplus were used. The utilized smartphones are listed below:

- Huawei Honor 6X, running Android Version 7.0
- OnePlus 7 Pro, running Android Version 10.0.7
- Samsung Galaxy S4, running Android Version 6.0.2
- Samsung Galaxy S7, running Android Version 8.0
- Samsung Galaxy S8, running Android Version 9.0

Additionally, an external condenser microphone with an omnidirectional polar pattern is used, which has a frequency range of 20 Hz to 16 kHz and a sensitivity of -30 dB$\pm$2dB.

For a lot of experiments during this study, a SLM of the type Tenmars TM 103 was used which has a frequency range of 31.5 Hz to 8 kHz, a measurement range of 30 to 130 dB(A) and an accuracy of $\pm$1.5dB.

The analysis of gathered data was performed by using python with the mathematical and scientific libraries NumPy and SciPy. To process the gathered measurements, the database format SQLite was used. Other diagrams were created with Microsoft Excel with the CSV file format.

During the analysis of the results an interesting observation was made. The results often showed a significant amount of double measurement values consecutively, indicating a transmission smaller than measured since the polling rate of the app was set to 50 ms. However, these values have not distorted the comparison of multiple data sets, as not the amount of polling values was decisive for the length of analyzed data but the length of the audio sample. Anyhow, all consecutive double values have been filtered out.

The following chapter gives an overview of the results that were gathered while performing different acoustical smartphone experiments. The first section deals with the results of the two experiments that were performed to achieve a reasonable degree of precision among different smartphone models. The section afterwards investigates the acoustical behavior of smartphones, especially how reliable the measurement values are they provide. Therefore, data is gathered under different conditions to obtain a wide range of information, which will allow analyzing gathered data in the future more efficiently. Finally, the third section deals with the evaluation of the already collected data of the Track Your Tinnitus application.

5.1 Calibration Techniques

The acoustical calibration of smartphones is a vital task for a possible model-wide comparison. Since each smartphone manufacturer or even smartphone model measures amplitudes at a varying specific value range, it is hard to calculate a comparable loudness value, e.g. in dB purely based on the given amplitude values.

As an example, the violin sample seen in figure 5.1 shows, that the Samsung smartphone records amplitude values mostly in the range of 6000 and 8000, sometimes tearing out to 5000 and 9000. On the other hand, values of the Huawei are generally lower at around 3000 to 6500. Unlike the previous models, the Oneplus shows lower values around 3000 to 4000 almost continuously. In addition to that, the fluctuations in amplitude on the Oneplus are very low, which could be due to the smartphone characteristics in general or it could be a sign for a built-in, non-deactivatable AGC. Another possible reason for low indicated fluctuations might be a low rate of transmission of the current amplitude, either by the ADC to the system or by the system to its API. Nevertheless, it is shown, that a calibration is necessary before taking any sound level measurements.

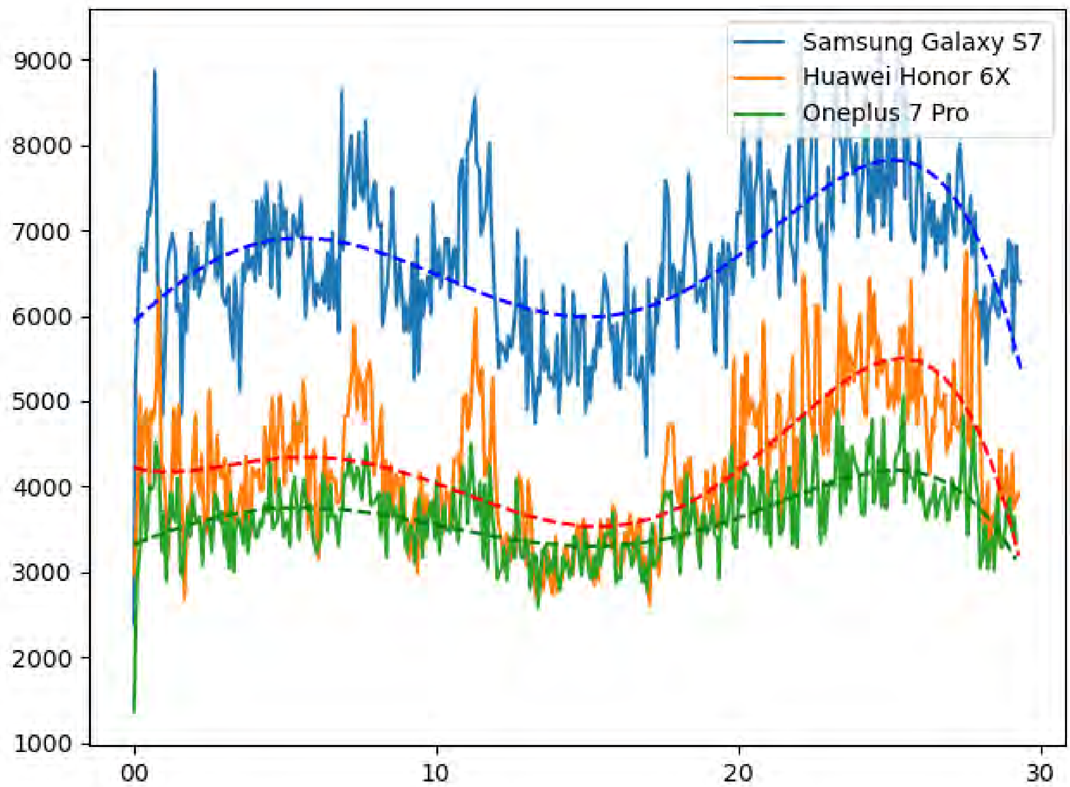


Figure 5.1: Comparison of smartphone models recording at varying amplitude ranges

The following section presents the results of the calibration experiments.

5.1.1 Basic Calibration

The following experiment was performed by using the application "Spaichinger Schallanalysator" for Android with version 2.2 (last update: 05.05.2020)[1].

The experiment is split into two parts. Firstly, conducting the calibration approach of the application, which includes tearing apart 10 sheets of copy paper. Secondly, measuring sound levels with the smartphone and a SLM next to it. For this purpose, multiple frequencies (100 Hz, 500 Hz, 1000 Hz, 2000 Hz, 4000 Hz, 5000 Hz, 7000 Hz, 8000 Hz, 10000 Hz) were played at a consistent volume with a Bluetooth

[1] https://spaichinger-schallpegelmesser.de/schallanalysator.html

speaker. This makes it possible to find out how much the sound levels of the smartphone deviate from the ones measured with the SLM and to find out if the deviation changes for different frequencies. The sound level for each frequency was consecutively measured 12 times simultaneously on the Oneplus smartphone and on the SLM.

The calibration part of the application is performed with the help of an interactive process. Initially, the process starts by measuring the background noise to calculate a threshold value for later measurements. Then, the actual calibration process can be initiated by the user, by pressing a button and starting to tear apart one sheet of paper. Meanwhile, the application measures the amplitude of the tearing and allocates it a decibel value. This process is repeated for 10 times. After that, a standard deviation is calculated and outputted. For this measurement, a deviation value of 2 dB(A) has been given by the application.

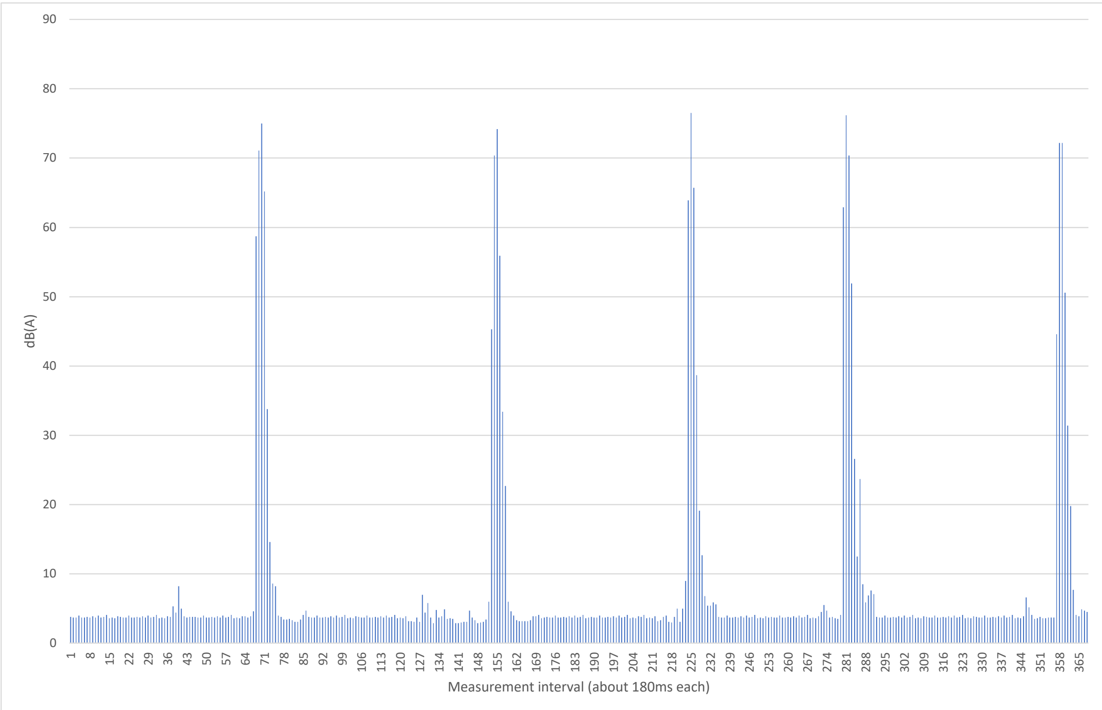


Figure 5.2: Measuring sound levels while tearing apart five sheets of paper to calculate the deviations of the peaks

To prove how the standard deviation of the tearing process can be computed, five sheets of paper were teared apart, and the noise was measured by using the mentioned application in an already calibrated phase. The audio spectrum of the process can be seen in figure 5.2. A standard deviation was then calculated with the maximum values of each tearing approach. By doing this, the standard deviation results in 1.7 dB(A), which is close to the 2 dB(A), that were calculated by the application while performing the calibration with ten sheets. Additionally, the average of all maximum values is calculated, which corresponds to 74.8 dB(A).

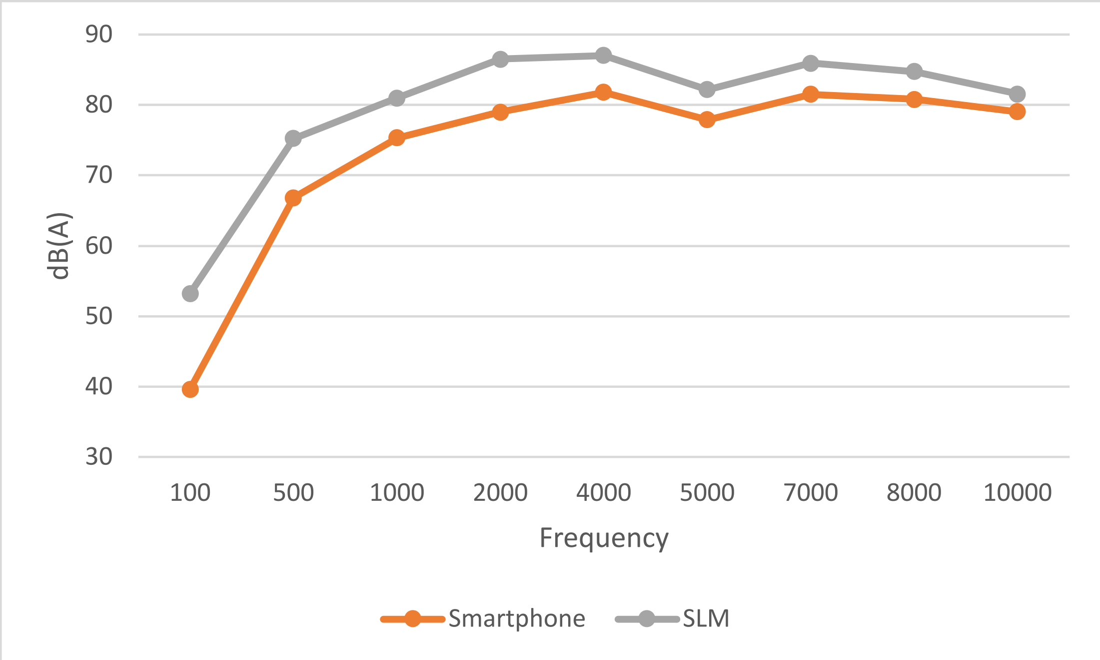


Figure 5.3: Measured sound levels of different frequencies on a smartphone and on the SLM after having performed a smartphone calibration

The second part of the experiment gives insights on the deviations of the sound levels measured by the smartphone application and a SLM. To do this, the sound levels of each frequency were mean averaged for the smartphone and SLM each. Then, both results were compared, which is shown in figure 5.3. Here, the frequencies in Hz are depicted, as well as the recorded sound level in dB(A). Both devices show a high increasing sound level between 100 Hz to 500 Hz and then only a lower increase until 4000 Hz. As it was expected since both SLM and smartphone measure the sound level in dB(A) this does approximately match the A-weighting filter curve. Although, the A-weighting peaks at around 3000 Hz and further shows a slow decrease. Calculating a two-paired, homoscedastic t-test of smartphone and SLM values (without 100 Hz), provides a result of $p = 0.68$. Nevertheless, it can clearly be said, that the measurements of the smartphone are reliable, which proves the first hypothesis.

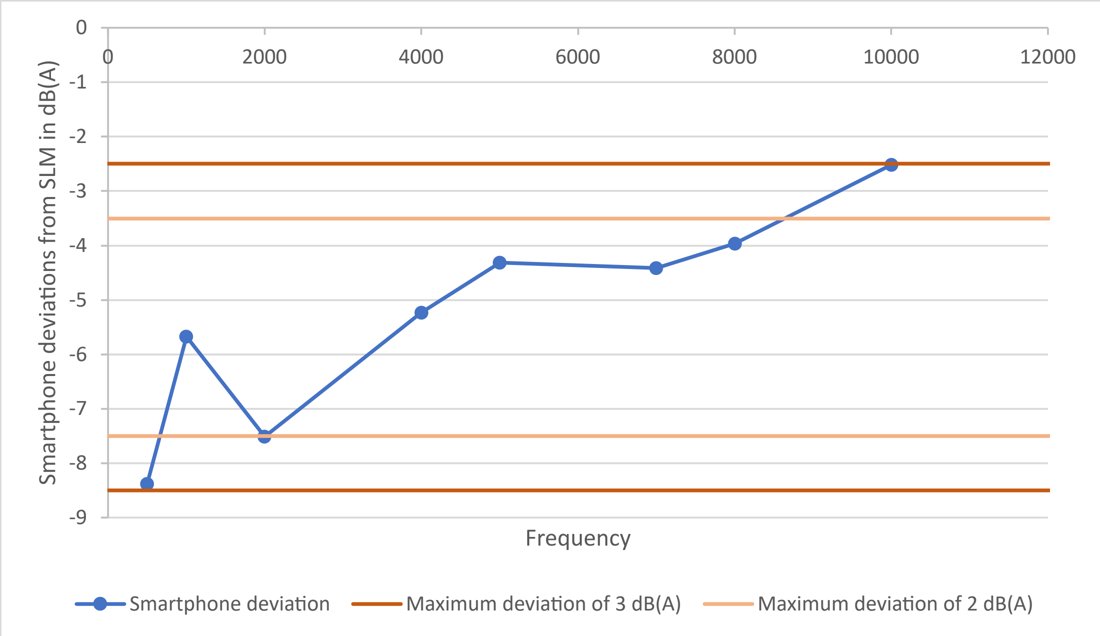


Figure 5.4: Average deviation of sound levels measured with a smartphone in regard to the SLM at different frequencies (without 100 Hz).

To further evaluate the deviations of the smartphone, the values were analyzed in detail. The deviations of the smartphone in regard to the SLM for each measured frequency can be seen in figure 5.4. Although, the deviations of the smartphone seem to be very high with about -14 dB(A) at 100 Hz, the error decreases drastically at 2000 Hz in the range of -8 dB(A) to -9 dB(A) and then further improves for 4000 Hz to around 5 dB(A). After that, the error rate decreases even further to a range of -4 dB(A) to -2 dB(A) for the highest frequencies measured. The reason for these differences could be the result of different frequency response rates of the microphone for each frequency, which will be discussed later. Additionally, some minor measurement errors shall be considered. Nonetheless, it is necessary to say, that no sound level correction value has been added to the measured smartphone values, which would be needed to settle the effects of the room, where the measurement has been taken place. However, by adding a correction value of 5.5 dB(A), the measurement error reduces to $\pm$ 3 dB(A) for frequencies between 500 and 10000 Hz. An even further reduction to a maximum error of $\pm$ 2 dB(A) is also ensured for frequencies between 500 and 8000 Hz. Therefore, the second and the third hypothesis, that different frequencies are recorded with different sound pressure levels and do not exceed a certain deviation level, have also be proven.

Interestingly, different frequencies do not generally show large fluctuations in deviation, which shows that different frequencies only have a limited impact on the accuracy of the measured sound level for the tested smartphone microphone. Thus, applying only a single correction value for all frequencies is feasible. Nevertheless, to achieve even better results with smaller deviations the frequency response values of the microphone should be taken into account if known, to limit the compensation of the frequency difference. Another possibility, if the exact frequency response value is unknown, is to use the frequency-based calibration method, which is explained in the next section 5.1.2.

5.1.2 Frequency-based Calibration

To include the frequency response of the smartphone microphone in the calibration, a more sophisticated approach is necessary. For this purpose, a set of frequencies was outputted with a loudspeaker 20 cm away from a smartphone microphone with an initial calibration value of 80 dB(A) at 1000 Hz. This defined volume level was not changed during the experiment. For each smartphone, every frequency was measured three times and average values were calculated. Calculating a two-paired, homoscedastic t-test of the results, provides significant values with $\alpha = 0.05$ and $p < \alpha$ for $p = 0.016,\ p = 0.005$ and $p = 0.004$ for the Galaxy S4 and $p = 0.049,\ p = 0.0$ and $p = 0.0$ for the Galaxy S7. The Oneplus results were not significant for the second try with $p = 0.0,\ p = 0.787$ and $p = 0.0$.

The results of each measurement are compared in figure 5.5. Both the Oneplus 7 Pro and the Galaxy S7 have consistently higher amplitudes, which are at least twice as large for 5000 Hz and seventeen times as large for 10 kHz than the Galaxy S4. This shows the differences that the smartphones have due to their specific characteristics. Additionally, a distinct frequency response for each smartphone can clearly be seen as well as the results for alternating each frequency. This shows that a single calibration value for every smartphone is not sufficient in every case for a general well-performing calibration. Therefore, the first hypothesis that varying frequencies have an impact on the recorded amplitude is correct.

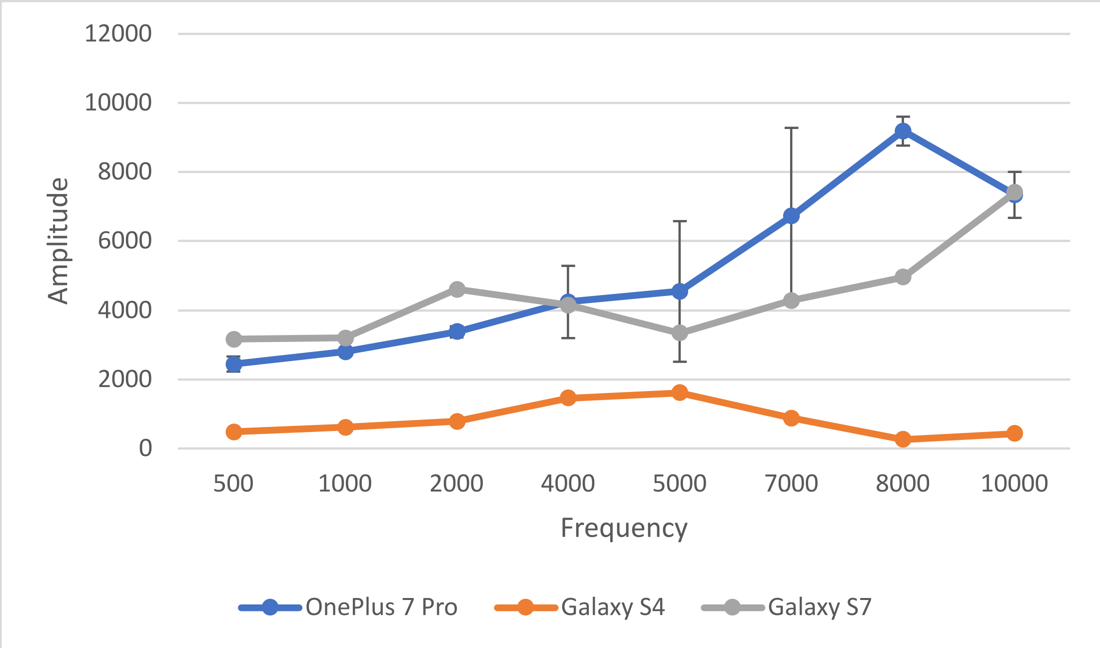


Figure 5.5: Measured amplitudes for the frequency calibration with indicated standard deviations.

The next step was to generate a dependency between amplitude and the measured sound level. Since the sound level of the A-weighting is not constant for all frequencies, a difference value must be added or subtracted to the 80 dB(A) at 1000 Hz. This is further explained in section 2.4.3 with figure 2.4. However, measuring the sound level in dB(A) with a SLM, the recorded values are not the same due to measurement inaccuracies like direction, position or reflection. The maximum deviations were 4.06 % for 20 kHz, which corresponds to +3.3 dB(A) or 4.26 % for 10 kHz, with -3.3 dB(A). For all other frequencies, the deviations were lower or even zero. The full list of deviations is shown in table 5.1.

Frequency in Hz	500	1000	2000	4000	5000	7000	8000	10000
A-weighting in dB(A)	76,8	80	81,2	81	80,5	79,48	78,9	77,5
SLM in dB(A)	74,4	80	84,5	81	81,8	79,6	77,8	74,2
Relative deviations in dB(A)	-2,4	0	3,3	0	1,3	0,12	-1,1	-3,3
Absolute deviations in %	-3,12	0,00	4,06	0,00	1,61	0,15	-1,39	-4,26

Table 5.1: Deviations of the SLM to the A-weighting

All in all, there are two options to calibrate the amplitude values with: the optimum A-weighting decibel values or the recorded experimental ones which include the measurement deviations. For the following calculations, the reference curve was created with the decibel values, which were taken with the sound level meter under experimental conditions. This decision was made, because the specific circumstances under which the data with the sound level meter was gathered are the same as when the amplitudes were recorded with the smartphone. To display the maximum deviations to the optimal A-weighting curve, the standard deviation is indicated in the plot, however as already discussed, it is very minor and not detectable in figure 5.6.

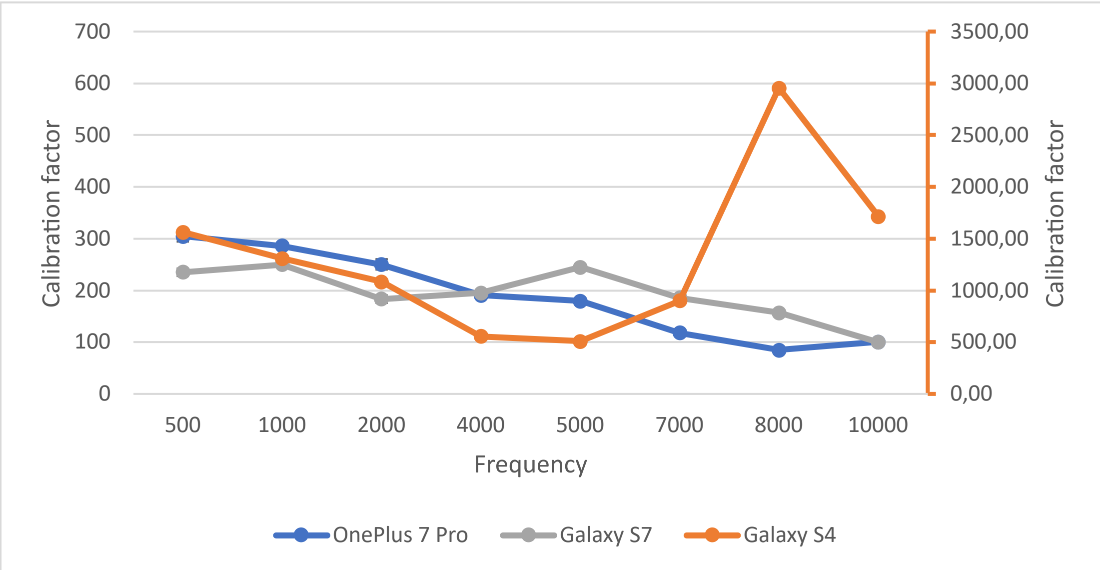
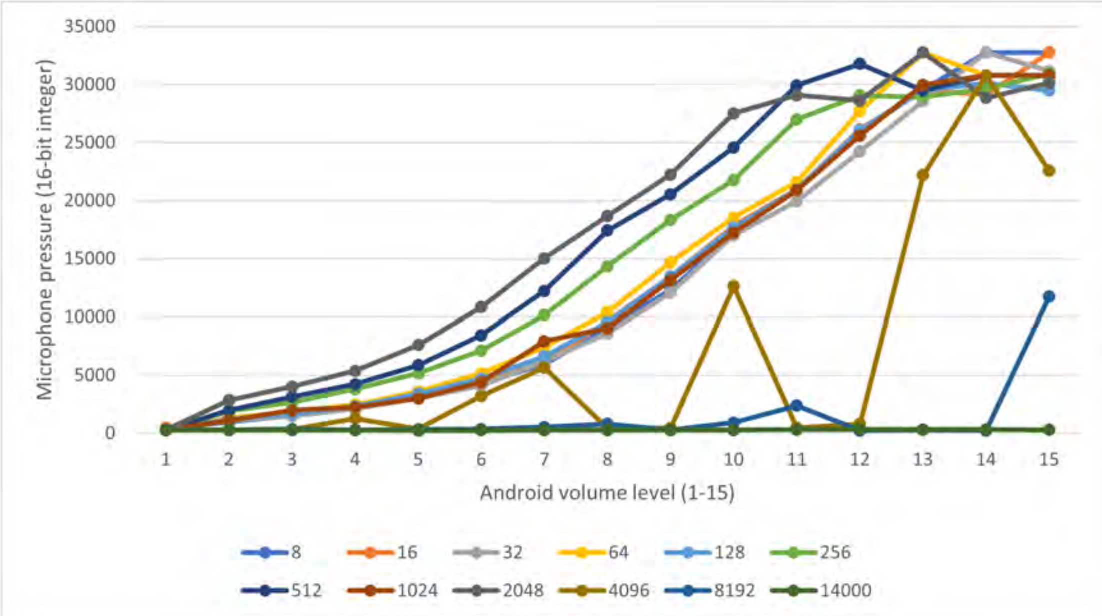


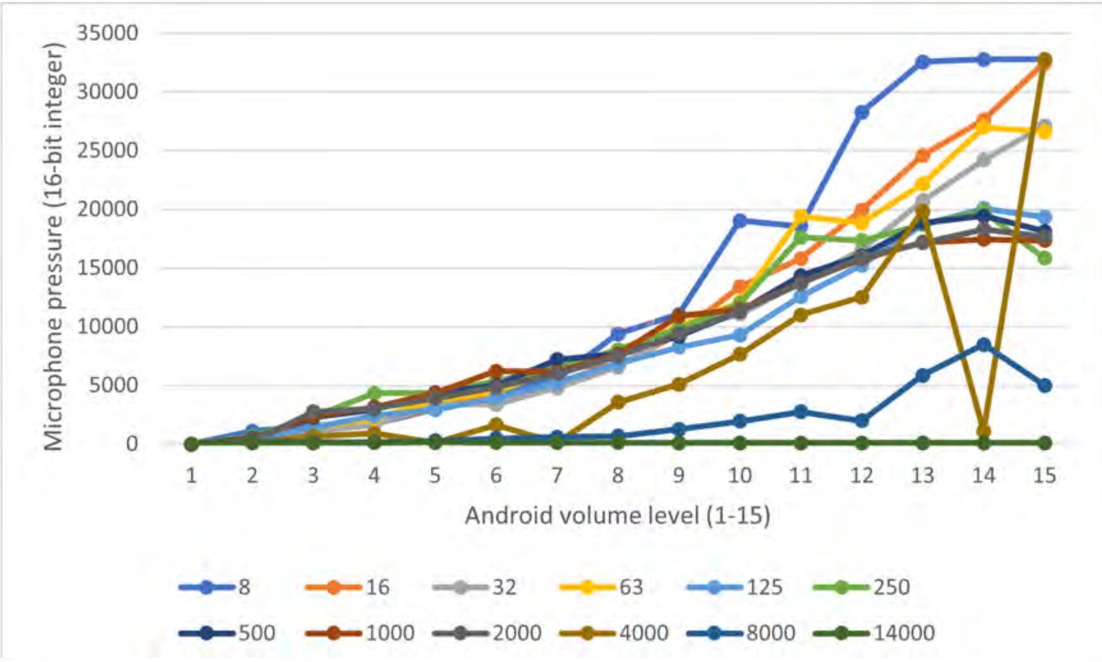
Figure 5.6: The calibration factors for the Galaxy S4 (right axis) are much higher and show a greater deviation rate than the ones of the Oneplus and the Galaxy S7 (left axis).

To achieve a general comparison for each frequency across all devices, the dB(A) value measured with the SLM is divided by the measured amplitude of the smartphone, which gives a frequency- and device-specific calibration factor. Due to norm purposes, it is multiplied by 10000. This factor has been chosen to adapt to the current amplitude values of the smartphones. Nonetheless, should the amplitude values rise for smartphones in the next generations, this factor must be scaled further up or the calibration values will be rather small or result in a high deviation. The current decibel deviation of the calibration factor varies due to the amplitude range of the smartphone and is larger for higher amplitudes. For the Oneplus 7 Pro at 8000 Hz, the highest amplitude was measured in this experiment, which leads to a deviation of 1 calibration factor of 0.9 dB(A). All calibration factors can be seen in figure 5.6. The calibration factors of the Oneplus and the Galaxy S7 are in a similar range, due to their similar operating amplitudes and are denoted on the primary vertical axis. Since the Galaxy S4 records amplitudes with a much lower value, a larger calibration factor is needed, shown on the secondary vertical axis. Another fact is that the graphs of the Oneplus and the Galaxy S7 are very steady and both show low fluctuations, however, the fluctuations in amplitudes are much higher for the Galaxy S4. For example, the calibration factor is six times as high for the Galaxy S4 between 5000 Hz and 8000 Hz due to the high drop in amplitude. For the Oneplus and the Galaxy S7, the deviation is not greater than three times between 500 Hz. and 10000 Hz. This indicates that the deviations between the microphone characteristics of the Galaxy S4 in regard to the A-weighting are much greater than with the other smartphones. Since different smartphones do not record frequencies with the same amplitude, the second statement is proven.

Additionally, the question is raised by how much the accuracy changes if multiple sound levels are included into the calibration. With the use of the self-developed application, many frequencies have been outputted on a Bluetooth speaker with the use of the complete volume spectrum of the smartphone. The results of the self-calibration can be seen in figure 5.7.



(a) Samsung Galaxy S6



(b) Samsung Galaxy S8

Figure 5.7: Measuring volumes of different frequencies on two different Samsung Galaxy models with external speaker.

The graph of figure 5.7a shows multiple frequency lines, which were outputted and measured at different volumes with a Galaxy S6. Generally, frequencies between 8 Hz and 128 Hz have overlapping amplitudes and form a steady increasing line. Interestingly, the same applies for 1024 Hz. Higher frequencies with 256 Hz, 512 Hz and 2048 Hz have continuously much steeper rising curves, due to microphone characteristics. Frequencies of 4096 Hz, 8192 Hz and 14000 Hz have not been generated and outputted correctly by the tone generator used with the application, however they are not relevant for this example, which is why they have been left out. The average background amplitude level was 249 for both measurements.

The same experiment has been conducted with the Galaxy S8 which is another smartphone of the same manufacturer, however in a later version. This was done to see how large the deviations in model variants are. In figure 5.7b, a relatively similar image to the first tested version can be seen. However, the former bundled frequency ranges are now stretched further apart and more fluctuating for higher selected sound levels. In contrast, for lower sound levels the measured frequencies are dense. Moreover, the higher, correctly outputted frequencies, are not measured at higher levels, thus a huge change in microphone characteristics can be detected. Looking at the general changes for increasing volumes, the overall rise is far lower compared to the curves, than what is shown in the curves of the former model.

This fact can obviously be identified, when calculating a moving average curve out of all frequencies as seen in figure 5.8.

This has been done to efficiently compare the average amplitude level with the other smartphone variant. Both curves show a similar behavior for the first five volume levels with deviations below 1000, then the curves split apart and increase with different rates until volume level 12 with a deviation of about 9000. While the measured amplitude levels of the Galaxy S6 rise much higher, the Galaxy S8 does not increase that fast, thus the distance to the S6 is getting bigger. The Galaxy S6 stops its steep rise at a volume level of 13 and only increases marginally thereafter. For the Galaxy S8, the curve peaks at level 14 and stays constant after that. Deviations are between amplitudes of 8000 and 9000. However, as discussed below, the standard deviation of the mean is much higher at the end for the Galaxy S8 and more visible at the middle levels for the Galaxy S6. This shows, that smartphones do not represent rising sound levels evenly in amplitude, which is the answer to the third statement.

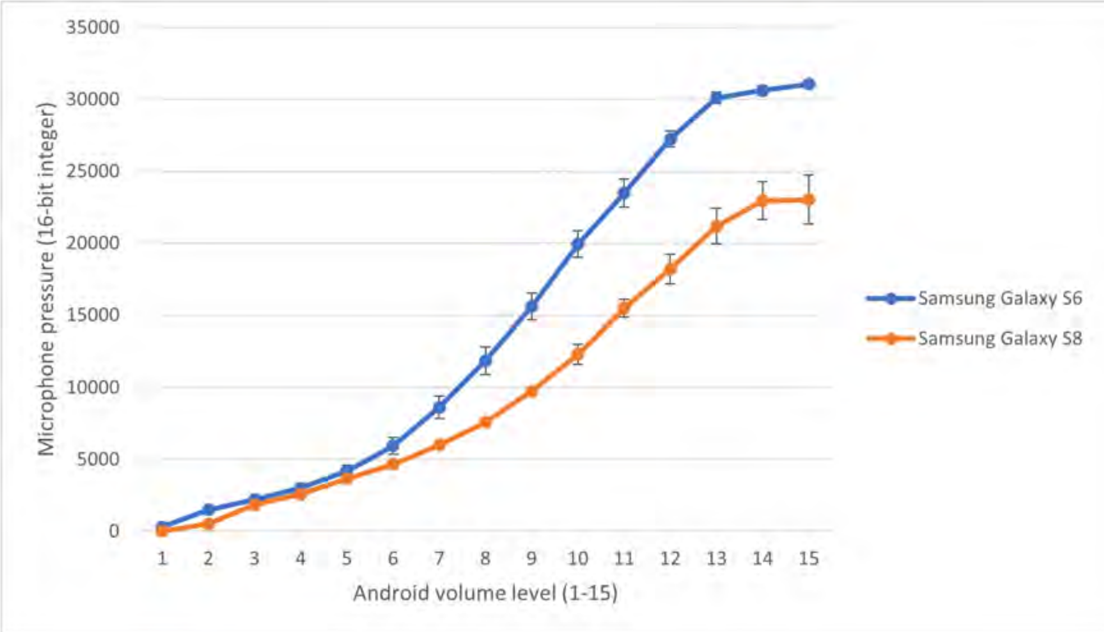


Figure 5.8: Moving average of amplitudes for frequencies 8 hz to 2048 hz of two smartphone variants

5.1.3 Analysis of already gathered data

Before the work for this thesis started, a large data set has already been gathered with the TrackYourTinnitus application for Android and iOS. The functionality included a set of defined questions regarding the current state and extent of the tinnitus perception at the time of answering [20]. For a more detailed analysis, the sound level data should be combined with information about the current noise situation. However, since the involved smartphones have not been calibrated beforehand, it is necessary to find a calibration approach afterwards. The gathered data include the amplitude values, a user id, hard- and software information about the device and the current time.

To determine the actual loudness data for a specific device it is possible to directly convert the amplitude data into sound pressure values. However, this requires to have information at least about the characteristics of the microphone, as already discussed in section 4.1, since a more detailed approximation of the sound level is only possible by having knowledge about the frequencies of the sound sample, as shown in section 4.1.2.

Nevertheless, to make a simplified statement about the approximate sound pressure value for a given amplitude, an average amplitude calculated out of different sound scenes was measured to include a wide range of frequencies. Therefore, the violin and the traffic sound sample, which were already used in section 4.2, were played by a Bluetooth loudspeaker at an average sound level of 80 dB(A) and 60 dB(A) at a distance of about 20 cm. The sound levels have been verified by an SLM, the same way as it is shown in figure 4.5. At the same time, the amplitudes were measured by the Galaxy S7 smartphone. The measured amplitudes of the complete sample duration were averaged for each sample and a whole average value was calculated out of that. The results are shown in table 5.2.

dB(A)	80	60
Amplitudes for the violin sample	3691	409
Amplitudes for the traffic sample	4282	401
Average amplitudes for both samples	3987	405

Table 5.2: Averaged amplitudes for the violin and traffic sound samples played at two different sound pressure levels.

The first average amplitude of 3987 shall be defined as a_0 and was used for the calibration at 80 dB(A). The second average amplitude of 405 shall be defined as a_1 and was used later for validation purposes at 60 dB(A). Before calculating the calibration factor, $b = 100$ amplitude values were subtracted from the amplitude at 80 dB(A), due to static background noise. Thereafter, the resulting amplitude is divided by sound pressure of the corresponding sound level of 80 dB(A), which corresponds to the calibration value c. This is done by converting the sound pressure level L_p, which is measured in dB(SPL) to the sound pressure p, which is measured in Pascal, as introduced in section 2.4. Additionally, p_0 is defined as the reference sound pressure given as $p_0 = 20\ \mu Pa = 2 \times 10^{-5}$. The formula is defined as:

$$\tilde{p} = p_0 * 10^{\frac{L_P}{20}}$$

[36]

For the calibration sound level of $L_p = 80\ dB(A)$, the sound pressure corresponds to $\tilde{p} = 0.2\ Pa$. This results in the following reference value, which serves as a device specific calibration.

$$c = \frac{a_0 - b}{\tilde{p}} = \frac{3987 - 100}{0.2} = 19435$$

For the next step, an amplitude value for which the sound level is calculated is divided by the calibration value. For this example, the amplitude of 405 is taken, which corresponds to the sound pressure of this amplitude.

$$\tilde{p} = \frac{a_1}{c} = \frac{405}{19435} \approx 0.0208\ Pa$$

Lastly, this sound pressure can then be converted back to its corresponding sound pressure level L_p [36]. For the measured amplitude value of 405 with its sound pressure of about $0.0208\ Pa$, the sound pressure level is calculated as:

$$L_p = 20 * \log_{10} \frac{\tilde{p}}{p_0} = 20 * \log_{10} \frac{0.02084}{0.00002} \approx 60.36\ dB(SPL)$$

This corresponds roughly to the average volume of our sound samples which was 60 dB(A). This shows, that a conversion of amplitudes to a sound pressure level is indeed possible. The complete source code is given in listing A.1.

This knowledge can be transferred to the data samples for the Galaxy S7 of the TrackYourTinnitus dataset. For the Galaxy S7 140 data samples exist, which were measured by one user. Calculating the amplitudes to their corresponding dB(SPL) values by using the already explained method, creates the histogram in figure 5.9. This shows, that all samples might be measured between 48 dB and 93 dB, which appear to be reasonable values.

Since this approach is only feasible if loudness information about the smartphone is known, another approach shall be presented if no calibration was done beforehand. Nevertheless, this requires that the data set is large enough and the amplitude range is known.

However, it is necessary to consider, that the sound levels have been calculated by neglecting their corresponding frequencies, which is an important factor as seen in section 5.1.2. Additionally, no defined measurement posture was defined, which could interfere with the results, as seen in section 5.2.2. Nevertheless, the method serves as a fairly accurate value to evaluate part of the already gathered data under

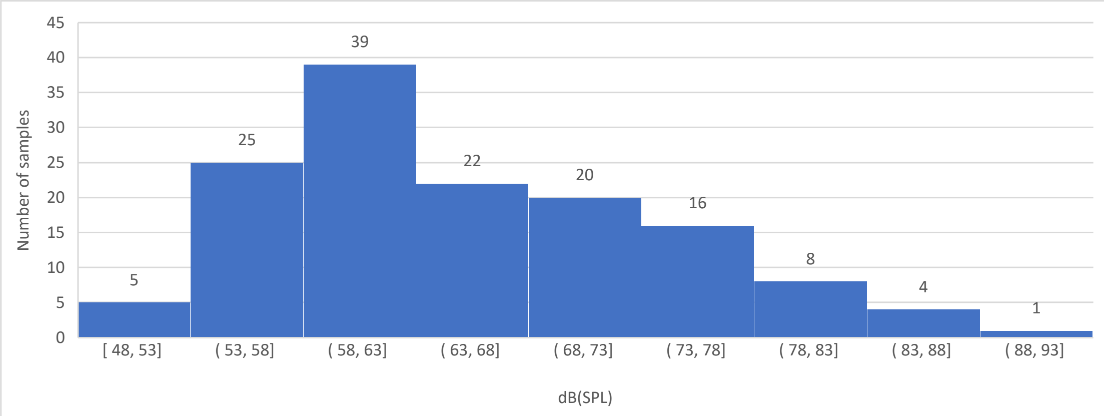


Figure 5.9: A histogram of the converted sound level values for the Galaxy S7 data samples

the condition, that an amplitude value for a defined sound level is known.

Looking at the TrackYourTinnitus data set again, shows that the amount of individual data points is 78767, however, since this thesis only deals with Android smartphones, data set from Apple devices are omitted. The largest amount of measurements (1596) was taken with an LG Optimus smartphone with an amplitude range between 0 and 29050, which roughly corresponds to the default amplitude range on Android. Other Android devices which show a high occurrence in the data set are the Samsung Galaxy S III mini (1030) with a range of 89 to 31157 and the Samsung Galaxy Note 3 (916) with a range of 0 to 4633. Since the maximum amplitude of the Note 3 is much lower, compared to the other ones, this indicates, that the amplitude range is different for varying devices, even from the same manufacturer.

Doing a further analysis with a selected number of devices which supposedly have the same amplitude range of about 0 to 32767 (LG Optimus, Samsung Galaxy S III mini, Samsung Galaxy S4), provides a sample of 33710 data samples by 41 individual users. Falsified values with a sound level of 0 were filtered out. The analysis can be seen in a histogram in figure 5.10. By dividing the amplitude scale into several regions this indicates that with 30% about third of the samples were taken in an amplitude range between 15 and 1515. The next group with amplitudes up to 3015 takes up 21% of all samples. Then amplitudes to a value of 4515, which is about a third of the whole amplitude range, correspond to 11% of all samples. It can clearly be seen that most of the amplitude data is taken in rather quiet surroundings like at home for example in a calm room. This fact can help to further analyze the sound

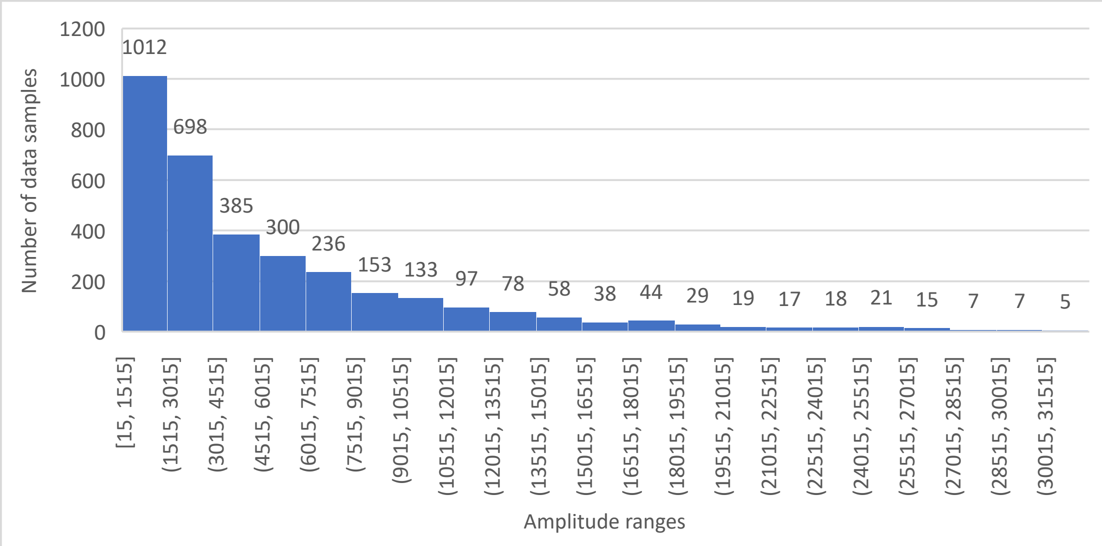


Figure 5.10: A histogram with the number of data samples for multiple Android smartphones of the same amplitude range.

level for example by setting a constant volume level for the most common amplitude values.

For a very quiet room the sound level could be about 20 dB(A). Since the lowest measured amplitude is 15, this will serve as the corresponding calibration amplitude. Calculating the sound pressure levels for every data point as shown before creates the histogram in figure 5.11.

Here, it can easily be seen, that most of the data samples could be taken at around 60 dB(SPL) to 70 dB(SPL), which corresponds to the sound level of a conversation or street traffic. Since these are both frequently occurring occasions, the calibration values seem to be well selected. Nevertheless, once again these values do only represent an approximation of the actual sound level and should therefore not be taken at face value.

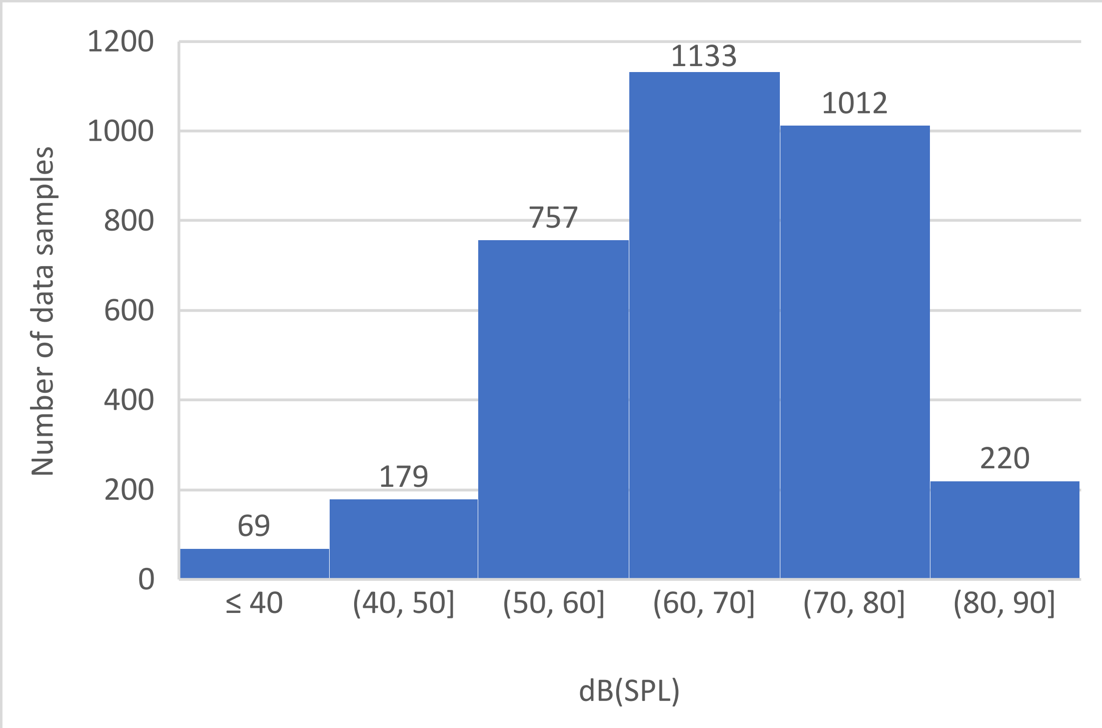


Figure 5.11: A histogram with the number of data samples and their corresponding sound level values for multiple Android smartphones of the same amplitude range.

5.2 Acoustical Behavior

Investigation of the acoustical behavior of smartphones, was performed by using a stereo sound system and the already in section 4.2.1 described samples. All experimental data was taken on a consistent volume level. The number of tested smartphones was $n = 3$ with different manufacturers to cover a diverse spectrum of hardware. Furthermore, the experiments were repeated with an external microphone, connected via an AUX-IN cable instead of the in-built smartphone microphone.

5.2.1 Monodirectional Behavior

To find out if and to what extent distance has an influence on the amplitude and how the quality varies for increasing distance, further experiments have been conducted. As sound only emerges from one direction and directly hits the smartphone, the experiment measures the monodirectional behavior of the smartphone. The experiment is conducted by playing different sound samples on a stereo system and recording the emitted sound waves on different smartphones at three distinct distances from the main speaker (1 meter, 2 meters, 3 meters). After that, the experiment was repeated by using an external microphone as the choice of input source.

During the experiment the smartphone was held in the hand at a height of 1.5 meter in a posture explained in subsection 4.2.4.

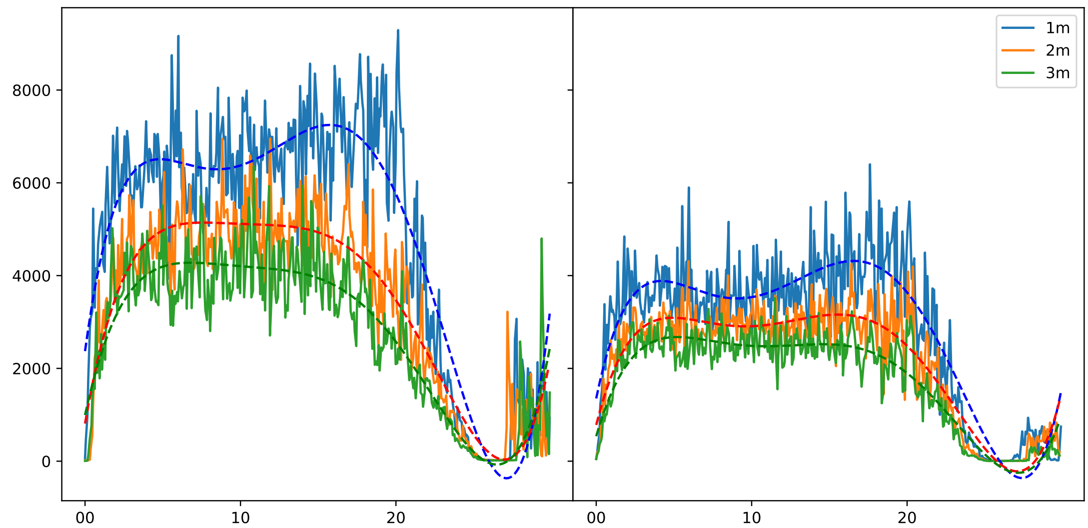


Figure 5.12: Measuring amplitudes with increasing distance with the aircraft takeoff sample on the Galaxy S7 (left) and the Oneplus (right)

The results show that the measured sound level indeed decreases with further distance with every device. The measured amplitudes were consistently lower on all devices with consistent behavior for the aircraft takeoff sample. This can be seen in figure 5.12 for the Galaxy S7 and the Oneplus smartphones. For both smartphones, the measurement provides similar results when comparing both smartphones in terms of their ability to reflect distance correctly in the amplitude values. Noticeably, the Galaxy shows more deviant values for different distances, consequently it might have a better distance measurement than the Oneplus, which values lie more together. Another interesting point is that the course of the polynomial trend line is not the same for all distances. For the first distance on both phones more excessive movements can be observed, thus the sound level characteristics of the sample are described better. For increasing distances, the amplitude movements decrease.

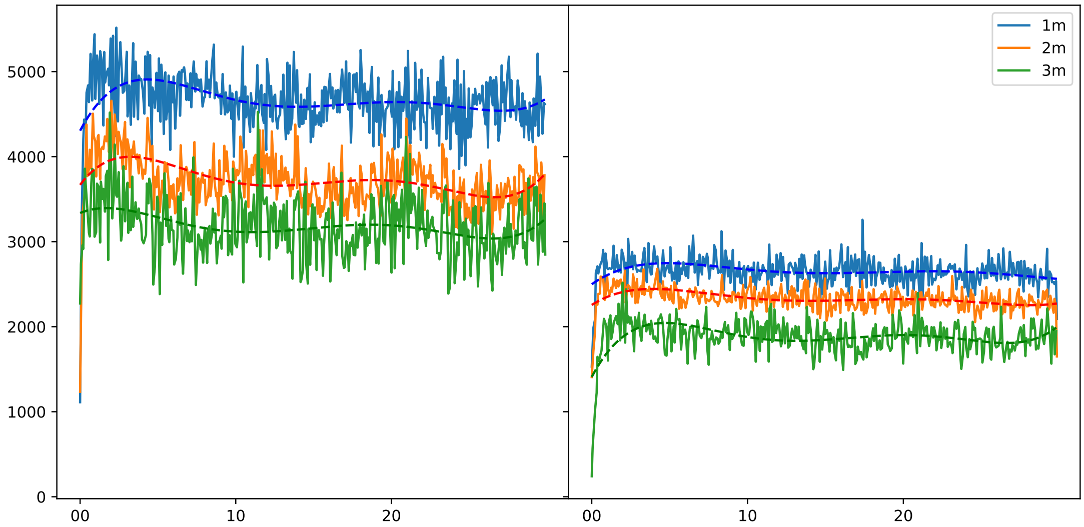


Figure 5.13: Measuring amplitudes with increasing distance with the helicopter sound sample on the Galaxy S7 (left) and the Oneplus (right)

A similar result can be seen with the helicopter sample on both phones in figure 5.13. Again, the Galaxy S7 shows much more split apart values for the first distance, while the amplitude values for the Oneplus seem to have spread more equally. Nonetheless, all smartphones show the ability to reflect distance in their amplitude values for both samples.

However, the ability to distinguish distance well is not consistent with all samples. The violin sample does not give any coherent clues about the difference of 2 meters and 3 meters distance as seen in figure 5.14. While a distance of 1 meter is clearly identifiable with the use of a polynomial trend line, for an increasing distance this is not possible anymore. The reason for this might be the high frequencies of the violin which might be more prone to a higher rate of room reflection. Similar results are measured on all smartphones and even the use of an external microphone did not show any improvement in this regard.

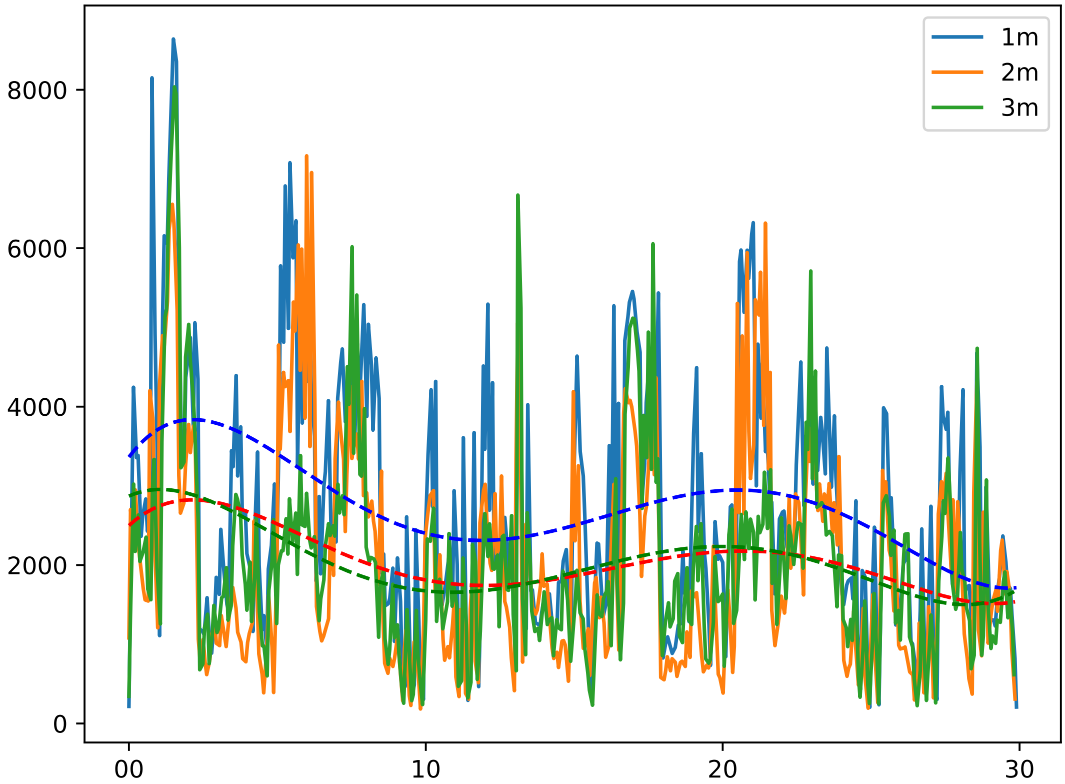


Figure 5.14: No statements about the distance of 2 m and 3 m can be made for the violin sample across all devices

Repeating the whole experiment with the use of an external microphone, rather leads to similar results for the traffic sample with the Galaxy S7 smartphone. A comparison for the internal and external microphone can be seen in figure 5.15.

This similar behavior was also noticed on other devices. Analysis of the raw amplitude data of the devices however shows, that the 1 meter and 2 meters measurement values are split more apart when using an external microphone. This creates a noticeable larger margin between the amplitudes of the first distance and the ones thereafter, which can also be confirmed consistently on all devices. Although, this behavior is not evenly distributed on all smartphones, as intensity varies according to which smartphone is used. For the Samsung smartphone this effect can slightly be noticed at around the 25 second mark, as the curves do not overlap that much. Nevertheless, for the monodirectional measurement, the differences can only be measured slightly.

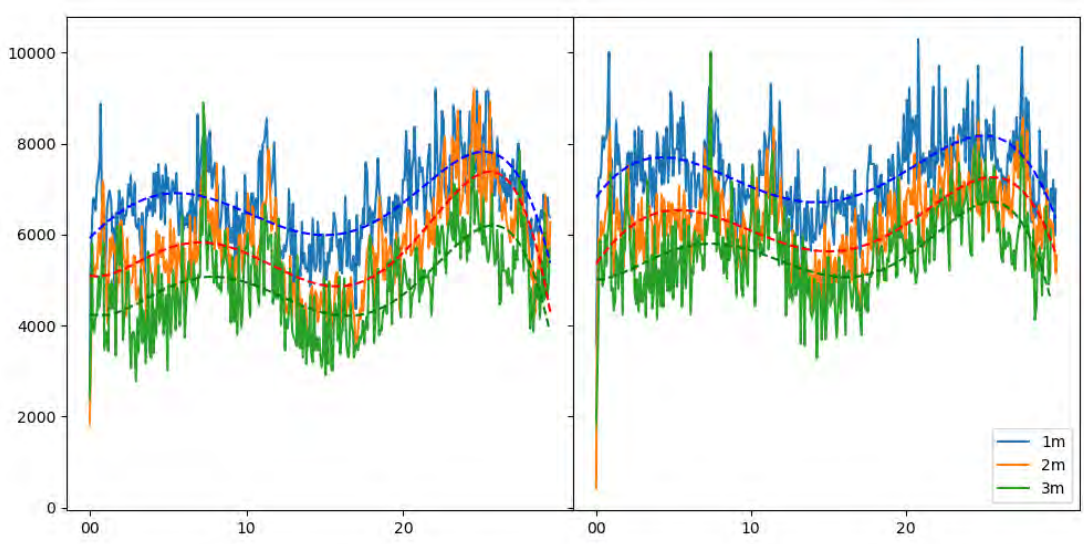


Figure 5.15: Comparison of amplitudes with increasing distance by using a Samsung Smartphone (left) in combination with an external microphone (right)

To find out, whether the obtained results of the monodirectional behavior can be valued, it is necessary to prove them in some way. Following the scientific Inverse-square law, the sound pressure decreases proportionally with distance by 1/r. Since this rule should apply to the executed measurements as well, it should be possible to find out if the law can be transferred to smartphone recordings. Proving this would allow the comparison of two or more values received from a sound source playing sound with a steady loudness. Furthermore, this technique would allow to roughly determine the distance from the sound source to the smartphone. To evaluate if the executed experiment does have any correlation to the Inverse-square law, three distinct amplitude values from each measured distance (1 meter, 2 meters, 3 meters) have been compared to the proportional decrease of the Inverse-square law. However, since only three measurement points exist, just an approximate guess with three data points can be made.

For analysis purposes, the measured amplitude value of each distance is plotted together with the curve of the Inverse-square law 1/r, where r is defined as the distance in meters. This attenuation factor is then applied by multiplying the measured amplitude at 1 meter to it, which allows to adapt the formula to the measurement. The results can be seen in figure 5.16. A clear decrease in amplitude for increas-

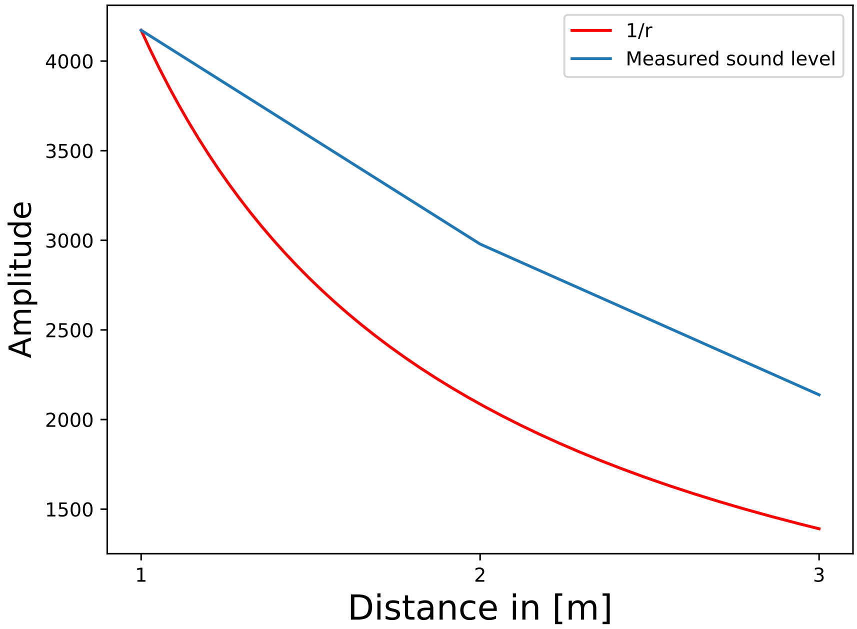


Figure 5.16: Comparison between proportional signal reduction and measured values

ing distance can be analyzed, as it was observed before. However, signal levels do not decrease proportional but proceed more linear. The reason for this might be the acoustical characteristics of the room, where the measurement was taken. Due to sound reflection of the walls, the amplitude degradation is far less than assumed. If the experiment was replicated with less reflection, the derivation of the measured levels with regard to the expected values should be far less. Additionally, it is unknown, if the amplitude of the smartphone directly corresponds to the sound pressure, since it is an arbitrary value of the Android OS. Hence, it might not entirely describe the sound pressure at the smartphone microphone.

Even though this experiment was only performed at three distances and the acoustical characteristics of the room might have modified the measurement values, the results still indicate a clear reduction of amplitude with accumulated distance. Comparing the results with the curve of the Inverse-square law, they are in a proximate range, although in spite of the given limitations a highly certain prove cannot be made.

5.2.2 Omnidirectional Behavior

To figure out the impact of the direction the sound is coming from, the sound level was measured at different directions from the sound source. As sound is not always hitting the smartphone from straight ahead, it must be evaluated how other sound directions influence the measured amplitude of the smartphone.

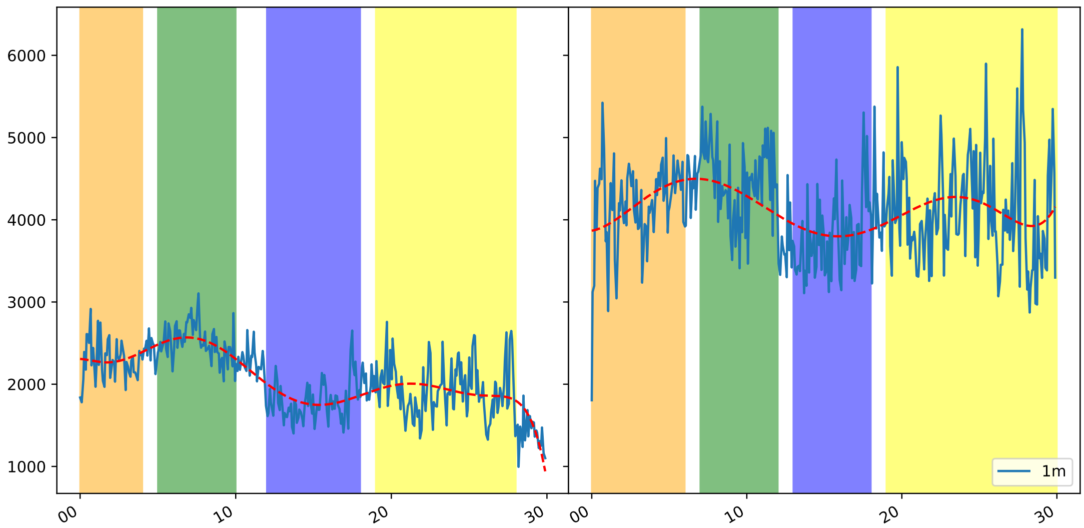


Figure 5.17: Measuring the amplitude with the internal microphone of the Oneplus (left) and an external microphone (right) at different positions to the sound source. The use of an external microphone leads to lower deviations.

The omnidirectional experiment has been performed with the traffic sample. As figure 5.17 shows, variations in loudness can clearly be seen for different positions. In the first orange marked range, the sound is coming from the front (0 degrees). After that, the person holding the smartphone in his right hand turns to the right (90 degrees). Interestingly, an increasing amplitude can be observed. Then, the person is turning right again and stands with his back to the sound source (180 degrees). The amplitude declines a lot in consequence of this and less amplitude spikes are measured. As a last position, the person turns right again (270 degrees), which increases the amplitude again, which can be seen in the yellow range. Nevertheless, a slight less average amplitude value than in the green range can be observed, although the person stands sideways to the sound source during both ranges. This might be the case due to the shielding of the right arm, which blocks the sound waves, that would directly hit the microphone, therefore the amplitude is lower in the yellow range than in the green. On the other hand, the green range shows a large increase, which could be the case of the reflection of the right hand, which reflects the sound waves into the microphone, thus the spike in amplitude is seen.

Repeating the experiment with an external microphone does show some differences as expected. The use of a true omnidirectional microphone does indeed mitigate the position differences since the amplitude does not show a lot of excessive changes as the internal microphone does. Obviously, the amplitude drops significantly, when standing behind the sound source (blue range), since the body blocks a lot of the incoming sound waves. However, this effect is much less, when compared to the internal microphone. Amplitude changes between the other three positions are only slightly visible and could also be part of the sample. Hence, it is clearly deducible, that the use of an external microphone does lead to more accurate results, when measuring sound from different positions.

6 Discussion

The performed experiments have shown that smartphones are indeed capable to measure noise in daily life. Therefore, they can also be used as an easy tool to be aware of the surrounding noise level in real time to lower the risk of health issues like stress, hearing loss or tinnitus caused by high noise levels. Moreover, this type of smartphone application can also be used to mitigate the effects of tinnitus for affected people. This can be done for example by identifying unpleasant frequencies or uncomfortable noise levels (too quiet or too loud) and issuing a warning if a threshold is exceeded. In summary, this research about noise measuring with a smartphone represents only the beginning of a larger field of application, which can be adopted quite easily, after the basic issues are solved.

Basic Calibration Approaches

Several calibration approaches have been introduced to convert smartphone amplitude values to the comparable dB(A) unit. Two methods, the basic and the frequency calibration, have been presented. Although, both methods have only been tested with Android smartphones in this thesis, the theory behind it works for all smartphones the same if their operating system provides an amplitude value that corresponds to the surrounding loudness. This has been proven for iPhone smartphones as well by M. Ziegler [40]. Beyond that, the calibration can be reproduced by anyone, which shows that the external validation can be ensured.

The results of the basic calibration show that it only takes about 5 minutes and some sheets of paper to gain an accuracy of a maximum of 3 dB(A). However, a compensation value of 5.5 dB(A) has to be taken into account to compensate for any room impacts. Since this correction value has only been created by using the help of a SLM, results may change if an estimation of this value must be assumed.

This topic is further discussed in the section The Effects of Reflection. Nonetheless, this approach has proven to work as an uncomplicated method to provide decent results.

The measured results are comparable with the results of other testing measurements by M. Ziegler (Sept. 2019), which have been gained by using this calibration method. First, the standard deviation of the calibration is said to be about 2 dB(A). For the conducted measurement, these 2 dB(A) have exactly been measured during this study. Additionally, the accuracy of the measurement is supposed to be about 2 to 3 dB(A), which has also been shown here with a maximum deviation of 3 dB(A). The compensation value was chosen a bit higher with 5.5 dB(A) than the values with 0 to 3.5 dB(A) in the comparison measurement of Ziegler (2019). However, since the compensation value is dependent on the characteristics of room and microphone, it is hard to compare [41].

Nonetheless, only one compensation value is added for the basic calibration, that corresponds to all frequencies and sound levels. This shows a limitation of the basic calibration. Since the process of tearing a paper covers a specific range of frequencies and only has a defined loudness range, it does not take the specific frequency response and the response for different sound levels of the microphone into account. While a difference in frequency response was suspected before, its existence has already been verified with the measurements of the basic approach.

Despite that, most of these limitations shall be solved by using the frequency-based calibration, which aims at providing a solution for such issues.

Frequency-dependent Calibration

The frequency calibration focuses on two things. Firstly, to compensate any deviations resulting from a different frequency response of the microphone and secondly, provide compensation for deviations in different sound levels.

Additionally, the frequency calibration emphasizes the importance of performing an individual calibration for different frequencies, since the amplitude curve of each smartphone clearly shows an own behavior. By including an individual calibration factor, the frequency differences can indeed be compensated. Nonetheless, it is necessary to apply further research to bring the concept to application. Especially, when measured sound consists of multiple frequencies, a single calibration factor is not available, due to the unique characteristics of each frequency to the recording. Hence, it is necessary to determine a new calibration factor out of all recorded frequencies. One solution could be to use the calibration factor value of the lowest measured frequency of the sample in question, which is referred as the *fundamental frequency*. This frequency has the feature of being the lowest and the loudest perceived frequency of the human ear [4]. However, it must be shown, that the calibration factor identified, results in a reliable sound level measurement.

Another approach to accomplish a valid calibration factor is by performing a time-frequency analysis like the Short-time Fourier Transform (STFT). This is used for continuous signal streams with changing frequencies over time. To determine the corresponding frequencies, the time signal is split into segments, which can then be analyzed further by dividing them into a frequency spectrogram [35]. A graphical representation of the result is given in figure 6.1. Information about specific frequency components of the sound could then be applied to calculate an appropriate calibration factor using the already collected calibration data.

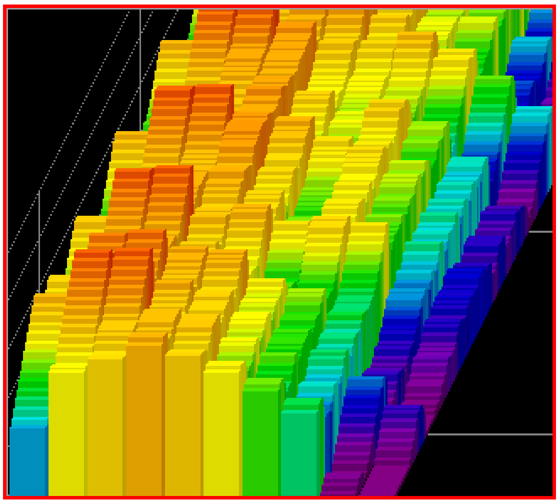


Figure 6.1: A 3D example for the short-time Fourier transform, which includes frequency (x-axis), colored intensity (y-axis) and time (z-axis) [38].

Additional Factor: Smartphone Age

One more issue that is worth mentioning, is that the age of a smartphone seems to have a negative effect on the accuracy. The result of the frequency calibration shows a larger scale of amplitude for the newer Oneplus 7 Pro and Galaxy S7 phones, which indicates a higher accuracy, due to a larger scale. Additionally, looking at the calibration factors for each frequency of the involved smartphones, shows that the frequency fluctuations are much higher for the Galaxy S4, whereas the Oneplus and the Galaxy S7 show an almost constant trend. Hence, the extent of necessary calibration is much higher for the Galaxy S4 since it does not depict the A-weighting accurately. This finding is comparable to the results of Murphy and King (2016), who observed that accuracy increases with newer phones. The reason behind this might be more advanced microphone technology or a deterioration for smartphone microphones with age [28]. Similarities between the Galaxy S4 and the Galaxy S7, which are smartphones by the same manufacturer could not be identified, however due to the age difference this is not surprising.

The Impact of Sound Levels for Calibration

The results of the impact of loudness for the calibration do not give a clear outcome. Resulting from the measurements, a different increasing rate of amplitudes for varying sound levels was shown. This indicates that the amplitude does not increase gradually. However, this was only shown by increasing the internal Android volume level and testing for pure frequencies. It is necessary to state, that the Android volume level generally cannot be used to calibrate a smartphone since it was not proven that the output sound level gradually increases evenly with increasing volume levels. Furthermore, different loudspeakers could possibly output the volume levels at different sound levels, too. Thus, a better procedure for future applications would be to measure the different sound levels in dB(A), as it would increase comparability. Additionally, several loudness values in dB(A) could be defined to use for future frequency measurements. Nevertheless, this includes additional hardware like a SLM or an already calibrated smartphone. However, looking at the results of the monodirectional experiment shows that by decreasing the distance and thus loudness indeed not every sample has shown an equally decreasing amplitude (apart from possible impacts of reflection).

Besides that, it is important to notice, that for this experiment and the behavior measurements, no amplitude to dB(A) conversion was performed to avoid falsifying the measurement results and showing the raw amplitude values. This was done to prevent calculation errors tamper with the validity of the results. Therefore, no conclusions from microphone pressure to dB(A) should be made.

To say if an inclusion of a loudness compensation value in future calibrations can be done, further knowledge about the smartphone behavior for decreasing loudness is needed. However, information about loudness cannot directly be retrieved since this is part of the calibration result and the actual sound level is unknown. But retrieving additional information about the type of noise and its direction could be achieved by detailed sound analysis with the use of machine learning. Thus, the recorded sound could be classified, which on one hand might be beneficial to the calibration and on the other hand could be an advantage in tinnitus research, since the specific type of noise which correlates to the perception of tinnitus at that moment could be identified. Hence, this might serve as a future approach to improve calibration and well-being as well.

Comparison to Related Studies

For a comprehensive comparison of the results, other studies are included, which deal with noise levels measured by a smartphone. Kardous and Shaw 2014 have tested several sound measuring applications for iPhone devices without performing any type of calibration and 3 of 10 applications had deviations below 2 dB(A) [17]. A large experiment has been performed by Murphy and King (2016), who conducted 1472 tests on 100 phones. Interestingly, only one application for Android but four applications for iOS have accurately measured deviations below 2 dB(A). Additionally, it was stated, that Android applications do generally provide "less reliable" results, than iOS applications, which might be the case due to more diverse hardware [28].

All in all, the relatively straightforward and uncomplicated calibration approaches evaluated in the course of this study give promising results, which can be improved even more by some modifications regarding the calibration. Nonetheless, the results show that a comparable level to professional applications can be achieved, although it is necessary to have in mind that the measurements for this work have been gathered under limited conditions.

Additional Factors: Distance to the Microphone and its Surroundings

Results of the data for the acoustical behavior give different results depending on the samples, which have been tested. A general capability to reflect even small distance changes of one meter in sound levels was proven. Despite an even decrease for the traffic sample for all 3 meters, other samples have shown problems to correctly reflect changes between 2 and 3 meters. Possible reasons for this are again room reflection parameters. Interestingly, the impact does change depending on the type of frequency, which is explained in further detail in section 6. Thus, it is no surprise, that for higher frequency samples like the violin tones, a much higher degree of reflection is measured.

The results of the omnidirectional experiment show, that the direction of the microphone to the sound source does indeed make a difference. All in all, this experiment provides knowledge if and to what extent different directions of sound sources have an influence on the amplitudes measured. Obtained results can possibly help to design future experiments which include the originating direction of sound into the calculation of sound level, as sound can be sensed with a higher intensity by the person when it is coming from behind, than it is measured with the smartphone in front. Therefore, a compensation value might be added to account for direction parameters. Having in mind, that the smartphone is not exposed directly to the surrounding sound every time, the same applies, if the smartphone is covert up, e.g. in a pocket. Under these conditions, it is harder to make any reliable assumptions about the results of the internal microphone, as low sound levels might not be recognized correctly anymore.

Additional Facotrs: Use of external Microphones

The performance of the external microphone is varying. While for the Samsung smartphone the use of an external microphone provides similar results as to the internal microphone, the results showed a much lower amplitude for the Oneplus phone.

The reason for this lies in the custom characteristics of the external microphone completely differing from the one of the internal microphone of the smartphone. Here the question arises, whether it is possible to achieve the same acoustical behavior by using the same external microphone on different smartphones. However, when comparing two different smartphones using the same external microphone, the results still differ, as the sound processing modalities of the smartphones are not the same, as it is also discussed in 2.5. For the external microphone in the monodirectional experiment the amplitudes are diverging (about 500 higher with the Samsung, and 2000 lower with the Oneplus). Nevertheless, for another external microphone, this is different again. If that was the case, a re-calibration would have to be performed every time, when an external microphone is used in combination with a new smartphone.

As a general consequence, values of the distance measurements increased at least between the distances of 1 meter and 2 meters for all devices by the use of an external microphone, which certainly improves the measurement quality. With 2 meters and 3 meters, the improvement can still be measured with the Oneplus phone in a mitigated form, the Samsung phone seems to be more prone to sound reflection since the positive effect of the external microphone can no longer be detected. Nonetheless, the positive effect of the external microphone can be seen, that is why the use of an external microphone is mostly better when compared to the internal microphone. However, considering the used external microphone was a low-price model, better results might be even achieved with a higher quality model.

These findings are close to the results of Faber (2017), who compared two inexpensive external microphones and concluded that “even a very inexpensive microphone [...] can be used for reasonably accurate measurements with a smartphone“ [9]. Additionally, a follow-up study of Kardous, Chucri and Shaw (2016) with external microphones has shown an enhancement of the mean deviation from 2 dB(A) with the internal microphone to 1 dB(A) with an external microphone, stating that external microphones “greatly improve the overall accuracy and precision of smartphone sound measurements“ [18].

A Guidance for future Measurements

Following the outcome of the results, the general recommendation to retrieve loudness data as accurate as possible, is to point the smartphone microphone in the direction of the sound to avoid sound reflection as much as possible. For the same reason, a lot of space should be around the microphone, which is the case with wide rooms being more favorable than small rooms. Additionally, nothing should block the path between the sound source and the microphone, so the sound level can be recorded directly in an unmitigated form. This includes removing any smartphone bumpers, cases or other form of protection parts, which could interfere with the measurement. Moreover, the height of the smartphone should be the same as the sound source to achieve an accurate recording. For an unsophisticated measurement, holding the smartphone at the general posture is sufficient. However, depending on the aims of the measurement, a defined height for holding the smartphone could be predefined, e.g. at face level to examine the consequences for hearing.

Since all measurements in this study have only been conducted under limited conditions in a living room, they need to be proven under some real life conditions, which includes a lot more external influences that might have an enormous impact on the accuracy. This includes for example increased background noise like wind or walking sounds, reduced sound reflection, changing distances and directions, a covert smartphone in a pocket or accidentally covering the microphone for a short time, which can lead to high amplitude spikes. Therefore, methods to identify and exclude such errors need to be developed to take correct measurements during large surveys. Some additional factors that are needed to consider when performing complex measurements will be elaborated in the next section.

The Effects of Reflection

Every calibration attempt and behavior experiment was made in a room with no sound absorbing materials, so it is possible that the results heavily depend on the room characteristics of the place where the experiment was conducted. This includes sound waves, which are either absorbed or reflected according to the material they hit. Lower frequencies have a longer wavelength and spread very much, so they are only reflected by larger objects, while higher frequencies are more directional and can even bounce off from smaller objects and thus are more likely to add to the measurement [33]. Thus, the measurement values deviate from exact results, which would have been gained under optimal conditions. A good example for this is demonstrated in figure 6.2, which is taken from a monodirectional measurement, which has been taken during an early experimental stage. Here, the impacts of reflection can be seen very clearly, since the measurement results at 5 meters distance have been recorded with a higher amplitude than the ones at 3 meters. The reason for this abnormality is the wall of the room, since due to the diffused sound reflection of the wall, the recorded amplitude is much higher. Thus, even though the sound at 3 meter distance has a much smaller distance to the sound source, it is measured at a lower amplitude, because of the impact, that reflection has on the 5 meter measurements.

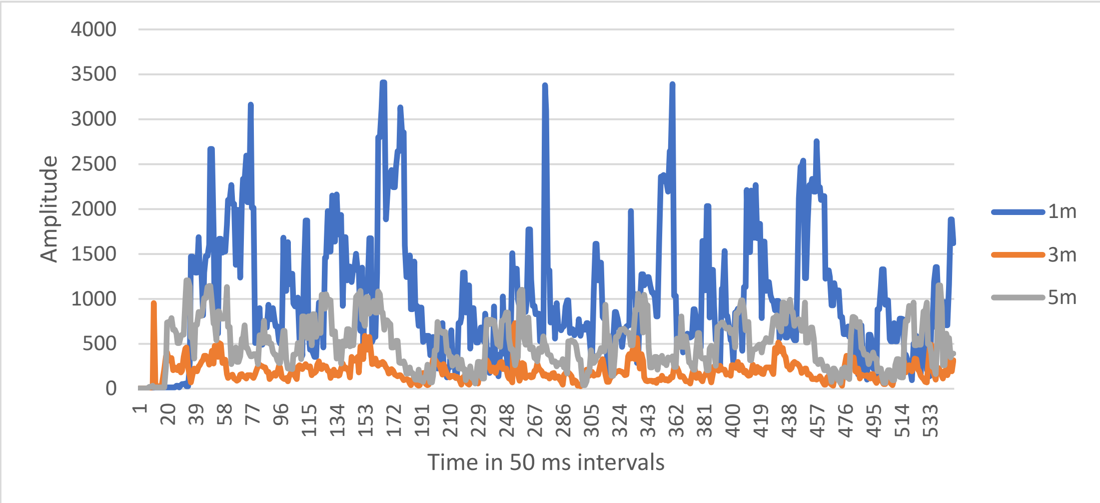


Figure 6.2: The impact of sound reflection is the reason why the sound is recorded at a much higher amplitude than it has.

To counteract this effect in future calibration approaches, it is necessary to choose a balancing value, which need to be further evaluated. This value can be estimated based on experience or calculated as a better alternative. For this purpose, new methods need to be developed to calculate an approximate balancing value depending on the size, equipment, floor and wall materials of the room. Additional help could be given by having the possibility to locate the current position of the user to change the balancing value, depending on the surrounding conditions. For example, choose a lower compensation value for reflection when outdoors and a higher one when in a small room indoors. These approaches can even include real time balancing values, which might help to reduce the impact of reflection on the result.

Considering Accuracy and Use Case

In any case one can assume, that for more precise results also more complex measurement procedures are needed. This can either include more advanced software and experience taken in experiments beforehand, the help of additional hardware or even both. Although, the aim of gathering loudness data with a smartphone is to achieve as much as precise results as possible, this does not mean to make this task too hard to be accomplished by the average user. Thus, a careful consideration between a high accuracy and a straightforward calibration and measuring procedure needs to be found. There are several reasons for this: Firstly, a too advanced approach leaves higher chances for the user to make errors during calibration or measurements, which could result in small deviations of measurements in the best case, or even falsify the complete result in the worst case. Secondly, the user could be unable to carry out the calibration due to missing hardware or additional needed equipment. Another reason for this might be incomprehensible or too complex instructions, which deter the user to even start the calibration. Consequently, the selected approach should be as easy as possible to perform.

Additionally, if microphone characteristics of a specific smartphone are known or have been examined, it is possible to share these over the internet to let other users benefit from it. Therefore, only one calibration per smartphone would be necessary to perform since calibration data is the same for identical smartphone models. This would reduce the difficulty for sound level measurements even further, if a large database is established, since the average user would only need to deal with the calibration, if no one else has already performed it. Hence, this would make it possible to perform a calibration on behalf of the user, which is especially useful for large crowd-based data collections.

Noise Measuring in Context of Tinnitus Perception

When measuring noise to investigate the course of tinnitus perception, it is important to consider the sensation of the individual person, since subjective tinnitus cannot be measured objectively [11]. Studies have already shown that the perception and annoyance of tinnitus fluctuates during the day, however, this varies from patient to patient [14]. Hence, in order to find a way to reduce the distress, further studies to measure the individual fluctuations in the perception of tinnitus are needed, which can be achieved by the use of questionnaires, as they are provided as part of the TrackYourTinnitus application [34]. The advantage of this is to further analyze the health condition of patients by gathering relevant medical data in their daily life, which is also referred to as Ecological Momentary Assessment (EMA). By combining the collected data of a large number of different participants, this helps to gain addition insights [19].

Moreover, including data about noise can additionally provide important information, since exposure to noise can likely cause emotional stress, which in turn has negative effects on tinnitus [25] [23]. Therefore, specific types of noises that can cause stress to the individual person need to be identified. This may include but might not be limited to high frequency sounds, high intensity sound levels, specific noise occurrences, which include a certain type of frequencies that cause annoyance or a combination of all. Thus, it is necessary to examine probable causes further by measuring and analyzing frequencies and noise levels.

The presented frequency calibration approach, in combination with a standardized measuring method, is well suited for this type of investigation since it does not only provide the ability to measure the sound level, but it also assures that the measurement is accurately done for different types of frequency responses of smartphone microphones. The acquired noise data can then further be used by performing a correlation with subjective data of the patient's questionnaire about the current tinnitus condition to obtain a widespread result. This method allows to do a much more differential analysis and utilization of the collected data since the relationship between noise and tinnitus perception can be investigated much easier and in more detail. As a result, the frequency method serves as a valuable approach for measuring noise levels for a general use, as well as to determine and modulate the tinnitus perception with a smartphone.

7 Conclusion

Following the results of the study it can be concluded, that the complex procedure of measuring sound levels with a smartphone is a task that can be accomplished. With the acoustical behavior experiments it has been proven, that smartphones are indeed capable of measuring sound correctly in relation to its distance. However, all results should be considered with caution, due to the impact of sound reflection and other factors.

All in all, the experiments and data do not only make it possible to convert amplitude values into sound pressure levels, but additionally lead to a better understanding in the way of how smartphones record sound and how changing factors like distance or position have an influence on the recorded amplitude. This enables further applications in the field of noise measuring for personal and health usage, as measuring sound levels (in a basic fashion) does not require more additional equipment than a smartphone and a calibration procedure. Therefore, anyone has the possibility to perform sound level measurements in daily life, which can provide approximate answers to basic questions, like: Could the current sound level be harmful for my hearing? Or: How long can I expose myself to this sound level until health risk might occur?

An additional use case is given by collecting and analyzing information about frequency and loudness of tinnitus patients in their daily life, which might result in further knowledge about the impact of tinnitus especially in its subjective occurrence. This can be achieved by putting the gathered data into context with information about the perception of the affected person to provide further improvements in dealing with tinnitus.

A summary of the findings and an outlook to future opportunities is given below.

7.1 Summary

This work deals with different topics, which include the analysis of smartphone amplitude data, collecting and developing own approaches to calibrate a smartphone microphone and performing different experiments regarding the acoustical behavior of an internal as well as an external microphone.

Results from the basic calibration have shown promising results with regard to general application, with a good accuracy to start with and a small effort for the calibration. Additionally, to compensate for the microphone response and to deal with differences in loudness, the frequency-based calibration has been introduced. While it provides a higher degree of accuracy, it requires much more effort to be accomplished. However, it has been shown, that it is worth to further develop the basic calibration technique to include reference values into the calibration that account for changing conditions and therefore achieve more precise results in these cases.

The results of the calibration have consequently led to the analysis of sound data samples of tinnitus probands since the data could be determined by a subsequent calibration approach.

Moreover, additional information when measuring amplitudes at different distances and positions were gathered. This has demonstrated that smartphones are able to show distance differences in amplitude, however, due to measurement limitations, the accuracy could not be proven. Another important finding is that the position to the sound source is important as well as the general posture as the time of measuring since this is highly influential for the result. Therefore, standardization suggestions for future measurements have been made and tested with multiple smartphones to increase the reliability.

Together with the achieved knowledge in the course of this study, a way to efficiently gather sound level data has been presented, which does not only benefit the average user, but could be very beneficial for tinnitus patients to monitor their health issues as well as for other conditions.

7.2 Outlook

The results achieved in this work may serve as a starting point to measure loudness with the help of a smartphone. As discussed before, a calibration is necessary beforehand and can be carried out in different ways, which all pose different difficulty and accuracy levels. Additionally, the number of overall calibrations to be performed can be reduced drastically, if they do not need to be performed by every user, e.g. by sharing them over the internet with regard to the type of smartphone used. This method would need to be developed and tested.

Nevertheless, all calibration measurements have not been in use outside up to now. Thus, more effort will be necessary to achieve good results also in daily life measurements. Although the presented calibration methods are state of the art, it will be necessary to improve and adapt them to changing indoor and outdoor conditions. Beside the introduced and currently working methods, with changing technology and more knowledge on this topic, there is a lot more potential to develop new approaches in the future, as the presented internal calibration shows. Hopefully, this will give everyone in the future the possibility to measure sound levels with a smartphone. Moreover, it might lead to a general awareness to sound as sound levels and sound types are factors for well-being and illness. The further use of the results in tinnitus evaluation and treatment shows a first application in this direction.

A Source Codes

Listing A.1: Conversion of a measured amplitude value to the sound pressure level in dB(SPL), by using a predefined amplitude and its sound pressure level as a calibration value.

```java
package de.uulm.dbis;

// Idea based on https://stackoverflow.com/a/14870458/3964954

// conversion of smartphone amplitudes to dB(SPL) after a calibration was performed
public class Amp2Dec {

    // reference sound pressure
    private final double REFERENCE = 0.00002;

    // ------- defined calibration values ----------------
    private final int BG_AMP = 50;         // measured amplitude of background noise
    private final int X = 4000 - BG_AMP;   // measured amplitude during calibration
    private final double Y = 80.0;         // corresponding dB(SPL)
    // -------------------------------------------

    // calculated dB(SPL)
    private double db;

    public Amp2Dec(int amp) {
        db = amplitude_to_decibel(amp);
```

```
    }

    private double amplitude_to_decibel(int amp) {
        // calibration value, given that amplitude value X
        ↪  corresponds to Y dB(SPL)
        double calibration_value = X / spl_to_pressure(Y);

        // pressure in pascal, given that amplitude behaves
        ↪  relative to the pressure
        double pressure = amp / calibration_value;

        // dB(SPL), using the formula for the sound pressure level
        return 20 * Math.log10(pressure / REFERENCE);
    }

    //sound pressure in pascal for a given sound level
    private double spl_to_pressure(double spl) {
        return Math.round(1000000000 * REFERENCE * Math.pow(10, spl
        ↪  / 20)) / 1000000000.0;
    }

    public double getDb() {
        return db;
    }
}
```

Listing A.2: Measure amplitude levels with an Android smartphone aiming for the highest possible quality.

```java
package de.uulm.dbis;

// Imports are shortened due to their large amount

public class AudioAnalyzer {

    // Adapted from
    // https://gist.github.com/pardom-zz/9644274 audioRecorder implementation
    // https://stackoverflow.com/a/21124025 audioRecorder to wav implementation

    private static int RECORDER_BPP; // bits per sample
    private static int RECORDER_CHANNELS_INT; // 1=MONO, 2=STEREO
    private final int SECONDS_TO_RUN = 40;
    Thread recordingThread = null;
    private boolean USE_DIRECT_REC; // true: audioRecord false: mediaRecorder
    private ScrollView scroller;
    private File filename;
    private TextView myText;
    private Context context;
    private String agc_status = "";
    private String recorder_settings = "";
    private File recording_file;
    private File recording_file_dir;
    private MediaRecorder mediaRecorder = null;
    private AudioRecord audioRecorder = null;
    private int mSampleRate;
    private short mAudioFormat;
    private short mChannelConfig;
```

```java
    private short[] mBuffer;
    private int mBufferSize = AudioRecord.ERROR_BAD_VALUE;
    private int mLocks = 0;
    private ArrayList<Integer> amplitude_list = new ArrayList<>();
    private File recording_file_dir_temp;

    public AudioAnalyzer(TextView con, ScrollView scroller, Context context, boolean dir) {
        this.myText = con;
        this.scroller = scroller;
        this.context = context;
        this.USE_DIRECT_REC = dir;
        analyze();
    }

    public void analyze() {
        Handler handler = new Handler();
        Runnable runnable = () -> {
            Date date = new Date();
            SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy_HH-mm-ss-SSS");
            SimpleDateFormat formatter2 = new SimpleDateFormat("HH-mm-ss-SSS");
            File path = new File(Environment.getExternalStorageDirectory().getAbsolutePath() + "/AudioAnalyzer");
            if (!(path.isDirectory())) {
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                    try {
                        Files.createDirectory(Paths.get(path.getAbsolutePath()));
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
```

```
            } else {
                path.mkdir();
            }
            System.out.println("AudioAnalyzer Directory
            ↪ created");
        } else {
            System.out.println("AudioAnalyzer Directory is not
            ↪ created");
        }
        filename = new File(path, "audioanalyzer_" +
        ↪ formatter.format(date) + ".csv");
        recording_file = new File(path, "recording_" +
        ↪ formatter.format(date) + ".aac");
        recording_file_dir_temp = new File(path, "recording_" +
        ↪ formatter.format(date) + "_temp.raw");
        recording_file_dir = new File(path, "recording_" +
        ↪ formatter.format(date) + ".wav");
        initRecorder();
        try {
            FileWriter writer = new FileWriter(filename);
            writer.append("DEBUG,");
            writer.append(" RECORDING MODE: " + (USE_DIRECT_REC
            ↪ ? "DIRECT (AUDIO REC) " : "MEDIA REC "));
            writer.append(" HARDWARE INFOS: " + "ManuFacturer
            ↪ :" + Build.MANUFACTURER + "Version OS: " +
            ↪ Build.VERSION.BASE_OS +
                    "Model: " + Build.MODEL +
                    "Product: " + Build.PRODUCT + " "
            );
            writer.append(" " + agc_status + " " +
            ↪ recorder_settings + " ");
            writer.append("\n");
            // 20 * seconds
            for (int i = 0; i < 20 * SECONDS_TO_RUN; i++) {
                int amp = getAmp();
```

```
                    String currdate_log = formatter.format(new Date());
                    String currdate_disp = formatter2.format(new Date());
                    String str = currdate_log + "," + amp + "\n";
                    writer.append(str);
                    System.out.println("MAX AMP: " + amp);
                    handler.post(() -> {
                        myText.append(currdate_disp + ",  " + amp + "\n");
                        scroller.fullScroll(ScrollView.FOCUS_DOWN);
                    });
                    try {
                        Thread.sleep(50);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                writer.flush();
                writer.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
            stopRecorder();
        };
        new Thread(runnable).start();
    }

    private void stopRecorder() {
        if (USE_DIRECT_REC)
            stopAudioRecord();
        else {
            mediaRecorder.stop();
            mediaRecorder.reset();
            mediaRecorder.release();
```

```
        }
    }

    private void initRecorder() {
        if (USE_DIRECT_REC) {
            initializeAudioRecord();
            startAudioRecord();
        } else
            initializeMediaRecord();
    }

    private int getAmp() {
        if (USE_DIRECT_REC)
            return getRawAmplitude();
        else
            return mediaRecorder.getMaxAmplitude();
    }

    public void initializeMediaRecord() {
        mediaRecorder = new MediaRecorder();
        // disable any agc
        AudioManager audioManager = (AudioManager) context.getSystemService(Context.AUDIO_SERVICE);
        if (audioManager.getProperty(AudioManager.PROPERTY_SUPPORT_AUDIO_SOURCE_UNPROCESSED) != null) {
            mediaRecorder.setAudioSource(MediaRecorder.AudioSource.UNPROCESSED);
            agc_status += " AGC: UNPROCESSED ";
        } else {
            mediaRecorder.setAudioSource(MediaRecorder.AudioSource.VOICE_RECOGNITION);
            agc_status += " AGC: VOICE RECOGNITION ";
        }
        // testing
```

```
        int sampr = 96000;
        int encbitr = 384000;
        int auch = 1;
        // or 44100?
        mediaRecorder.setAudioSamplingRate(sampr);
        // or 96000?
        mediaRecorder.setAudioEncodingBitRate(encbitr);
        // or AMR_NB?
        mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
        // or AAC or AMR_NB?
        mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.HE_AAC);
        // could also be "/dev/null"
        mediaRecorder.setOutputFile(String.valueOf(recording_file));
        // mono or stereo?
        mediaRecorder.setAudioChannels(auch);
        recorder_settings += "AUDIO RECORDER SAMPLERATE/ENCODINGBITRATE/CHANNELCONFINT/AUDIOFORMAT: "
                + sampr + "/" + encbitr + "/" + auch + "/" + "output: MPEG_4 / encoder: HE_AAC";
        try {
            mediaRecorder.prepare();
        } catch (IOException e) {
            e.printStackTrace();
        }
        mediaRecorder.start();
    }

    public void initializeAudioRecord() {
        if (mSampleRate > 0 && mAudioFormat > 0 && mChannelConfig > 0) {
```

```java
            audioRecorder = new AudioRecord(MediaRecorder.AudioSource.MIC, mSampleRate, mChannelConfig, mAudioFormat, mBufferSize);
            return;
        }

        // Find best/compatible AudioRecord
        for (int sampleRate : new int[]{48000, 44100, 32000, 16000, 8000}) {
            for (short audioFormat : new short[]{AudioFormat.ENCODING_PCM_16BIT, AudioFormat.ENCODING_PCM_8BIT}) {
                for (short channelConfig : new short[]{AudioFormat.CHANNEL_IN_MONO, AudioFormat.CHANNEL_IN_STEREO}) {
                    // Try to initialize
                    try {
                        mBufferSize = AudioRecord.getMinBufferSize(sampleRate, channelConfig, audioFormat);
                        if (mBufferSize < 0) {
                            continue;
                        }
                        mBuffer = new short[mBufferSize];
                        audioRecorder = new AudioRecord(MediaRecorder.AudioSource.MIC, sampleRate, channelConfig, audioFormat, mBufferSize);

                        if (audioRecorder.getState() == AudioRecord.STATE_INITIALIZED) {
                            mSampleRate = sampleRate;
                            mAudioFormat = audioFormat;
                            mChannelConfig = channelConfig;
```

```
                    if (channelConfig ==
                    ↪  AudioFormat.CHANNEL_IN_MONO) {
                        RECORDER_CHANNELS_INT = 1;
                    } else {
                        RECORDER_CHANNELS_INT = 2;
                    }
                    if (audioFormat ==
                    ↪  AudioFormat.ENCODING_PCM_16BIT) {
                        RECORDER_BPP = 16;
                    } else {
                        RECORDER_BPP = 8;
                    }
                    System.out.println("AUDIO RECORDER
                    ↪  INITIALIZED");
                    String rec_set = "AUDIO RECORDER
                    ↪  SAMPLERATE/CHANNELCONFINT/␣
                    ↪  AUDIOFORMAT/mBUFFERSIZE:
                    ↪  "
                            + sampleRate + "/" +
                            ↪  RECORDER_CHANNELS_INT + "/"
                            ↪  + RECORDER_BPP + "/" +
                            ↪  mBufferSize;
                    System.out.println(rec_set);
                    recorder_settings += rec_set;
                    return;
                }
                System.out.println("AUDIO RECORDER ERROR");
                audioRecorder.release();
                audioRecorder = null;
                throw new IllegalStateException("Could not
                ↪  init audioRecorder");
            } catch (Exception ignored) {
            }
        }
    }
```

```
        }
    }

    public synchronized void startAudioRecord() {
        // AGC DISABLE
        if (AutomaticGainControl.isAvailable()) {
            AutomaticGainControl agc = AutomaticGainControl.create(
                    audioRecorder.getAudioSessionId()
            );
            agc_status += " AGC STATUS BEFORE: " + agc.getEnabled() + " ";
            agc.setEnabled(false);
            agc_status += " AGC STATUS AFTER: " + agc.getEnabled() + " ";
        } else {
            agc_status = " NO AGC AVAILABLE ";
        }
        if (audioRecorder == null || audioRecorder.getState() != AudioRecord.STATE_INITIALIZED) {
            throw new IllegalStateException("startRecording() called on an uninitialized AudioRecord.");
        }
        if (mLocks == 0) {
            audioRecorder.startRecording();
            recordingThread = new Thread(this::writeAudioDataToFile, "AudioRecorder Thread");
            recordingThread.start();
        } else {
            throw new IllegalStateException("Could not start audioRecorder");
        }

        mLocks++;
    }
```

```

    // Write the output audio in byte
    private void writeAudioDataToFile() {
        String filename = getTempFilename();
        FileOutputStream os = null;
        try {
            os = new FileOutputStream(filename);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

        while (mLocks != 0) {
            // gets the voice output from microphone to byte format
            int bufferReadSize = audioRecorder.read(mBuffer, 0, mBufferSize);
            calcRawAmplitude(mBuffer, bufferReadSize);
            try {
                // writes the data to file from buffer
                // stores the voice buffer
                // short[] shorts = new short[bytes.length/2];
                // to turn bytes to shorts as either big endian or little
                // endian.
                // ByteBuffer.wrap(bytes).order(ByteOrder.LITTLE_ENDIAN).asShortBuffer().get(shorts);
                // to turn shorts back to bytes.
                byte[] bytes2 = new byte[mBuffer.length * 2];
                ByteBuffer.wrap(bytes2).order(ByteOrder.LITTLE_ENDIAN)
                        .asShortBuffer().put(mBuffer);
                os.write(bytes2);
                //  ServerInteractor.SendAudio(buffer);
            } catch (IOException e) {
                e.printStackTrace();
            }
```

```
        }

        try {
            os.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public synchronized void stopAudioRecord() {
        mLocks--;
        if (mLocks == 0) {
            if (audioRecorder != null) {
                audioRecorder.stop();
                audioRecorder.release();
                audioRecorder = null;
                recordingThread = null;
                // copy the recorded file to original copy & delete
                ↪  the recorded copy
                copyWaveFile(getTempFilename(), getFilename());
                deleteTempFile();
            }
        }
    }

    private String getFilename() {
        return recording_file_dir.getAbsolutePath();
    }

    private void deleteTempFile() {
        File file = new File(getTempFilename());
        file.delete();
    }

    // stores the file into the SDCARD
```

```
    private String getTempFilename() {
        return recording_file_dir_temp.getAbsolutePath();
    }

    private void copyWaveFile(String inFilename, String
     ↪ outFilename) {
        FileInputStream in;
        FileOutputStream out;
        long totalAudioLen;
        long totalDataLen;
        int channels = RECORDER_CHANNELS_INT;
        long byteRate = RECORDER_BPP * mSampleRate * channels / 8;

        try {
            in = new FileInputStream(inFilename);
            out = new FileOutputStream(outFilename);
            totalAudioLen = in.getChannel().size();
            totalDataLen = totalAudioLen + 36;

            WriteWaveFileHeader(out, totalAudioLen, totalDataLen,
                    mSampleRate, channels, byteRate);
            byte[] bytes2 = new byte[mBuffer.length * 2];
            ByteBuffer.wrap(bytes2).order(ByteOrder.LITTLE_ENDIAN)
                    .asShortBuffer().put(mBuffer);
            while (in.read(bytes2) != -1) {
                out.write(bytes2);
            }

            in.close();
            out.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

```

```java

    private void WriteWaveFileHeader(FileOutputStream out, long totalAudioLen,
                                     long totalDataLen, long longSampleRate, int channels, long byteRate)
            throws IOException {
        byte[] header = new byte[4088];

        header[0] = 'R'; // RIFF/WAVE header
        header[1] = 'I';
        header[2] = 'F';
        header[3] = 'F';
        header[4] = (byte) (totalDataLen & 0xff);
        header[5] = (byte) ((totalDataLen >> 8) & 0xff);
        header[6] = (byte) ((totalDataLen >> 16) & 0xff);
        header[7] = (byte) ((totalDataLen >> 24) & 0xff);
        header[8] = 'W';
        header[9] = 'A';
        header[10] = 'V';
        header[11] = 'E';
        header[12] = 'f'; // 'fmt ' chunk
        header[13] = 'm';
        header[14] = 't';
        header[15] = ' ';
        header[16] = 16; // 4 bytes: size of 'fmt ' chunk
        header[17] = 0;
        header[18] = 0;
        header[19] = 0;
        header[20] = 1; // format = 1
        header[21] = 0;
        header[22] = (byte) channels;
        header[23] = 0;
        header[24] = (byte) (longSampleRate & 0xff);
        header[25] = (byte) ((longSampleRate >> 8) & 0xff);
```

```
        header[26] = (byte) ((longSampleRate >> 16) & 0xff);
        header[27] = (byte) ((longSampleRate >> 24) & 0xff);
        header[28] = (byte) (byteRate & 0xff);
        header[29] = (byte) ((byteRate >> 8) & 0xff);
        header[30] = (byte) ((byteRate >> 16) & 0xff);
        header[31] = (byte) ((byteRate >> 24) & 0xff);
        header[32] = (byte) (channels * RECORDER_BPP / 8); // block align
        header[33] = 0;
        header[34] = (byte) RECORDER_BPP; // bits per sample
        header[35] = 0;
        header[36] = 'd';
        header[37] = 'a';
        header[38] = 't';
        header[39] = 'a';
        header[40] = (byte) (totalAudioLen & 0xff);
        header[41] = (byte) ((totalAudioLen >> 8) & 0xff);
        header[42] = (byte) ((totalAudioLen >> 16) & 0xff);
        header[43] = (byte) ((totalAudioLen >> 24) & 0xff);

        out.write(header, 0, 4088);
    }

    private void calcRawAmplitude(short[] mBuffer, int bufferReadSize) {
        if (bufferReadSize < 0) {
            amplitude_list.add(0);
            return;
        }
        int sum = 0;
        for (int i = 0; i < bufferReadSize; i++) {
            sum += Math.abs(mBuffer[i]);
        }
        if (bufferReadSize > 0) {
            amplitude_list.add(sum / bufferReadSize);
```

```
                return;
            }
            amplitude_list.add(0);
        }

    private int getRawAmplitude() {
        if (amplitude_list.size() == 0) {
            return 0;
        } else
            return amplitude_list.get(amplitude_list.size() - 1);
    }
}
```

List of Figures

List of Listings

Bibliography

[1] Peter Alberti. “Noise induced hearing loss”. In: *BMJ (Clinical research ed.)* 304 (Mar. 1992), p. 522. DOI: 10.1136/bmj.304.6826.522.

[2] American Tinnitus Association. *Understanding the Facts*. 2020. URL: https://www.ata.org/understanding-facts.

[3] David Baguley, Don McFerran, and Deborah Hall. “Tinnitus”. In: *The Lancet* 382.9904 (2013), pp. 1600–1607. ISSN: 0140-6736. DOI: 10.1016/S0140-6736(13)60142-7. URL: https://doi.org/10.1016/S0140-6736(13)60142-7.

[4] B. Benward and M.N. Saker. *Music in Theory and Practice*. Music in Theory and Practice Vol. 1. McGraw-Hill, 2003. ISBN: 9780072942620.

[5] Jennifer Bizley and Yale Cohen. “The what, where and how of auditory-object perception”. In: *Nature reviews. Neuroscience* 14 (Oct. 2013), pp. 693–707. DOI: 10.1038/nrn3565.

[6] Hristo Bojinov et al. “Mobile Device Identification via Sensor Fingerprinting”. In: *CoRR* abs/1408.1416 (2014). arXiv: 1408.1416. URL: http://arxiv.org/abs/1408.1416.

[7] Wikimedia Commons. *Anatomy of Human Ear with Cochlear Frequency Mapping*. 2009. URL: https://commons.wikimedia.org/wiki/File:Anatomy_of_Human_Ear_with_Cochlear_Frequency_Mapping.svg.

[8] Norman Erber. *Auditory Training*. Jan. 1982. ISBN: 0-88200-149-3.

[9] Benjamin M. Faber. “Acoustical Measurements with Smartphones : Possibilities and Limitations”. In: 2017, pp. 10–17.

[10] Klaus Genuit. “Status of psychoacoustics in noise analysis”. In: *Journal of the Acoustical Society of America* 137 (Apr. 2015), pp. 2291–2291. DOI: 10.1121/1.4920362.

[11] Haúla Faruk Haider et al. "Pathophysiology of Subjective Tinnitus: Triggers and Maintenance". In: *Frontiers in Neuroscience* 12 (2018), p. 866. ISSN: 1662-453X. DOI: `10.3389/fnins.2018.00866`. URL: `https://www.frontiersin.org/article/10.3389/fnins.2018.00866`.

[12] Byung Han et al. "Tinnitus: Characteristics, Causes, Mechanisms, and Treatments". In: *Journal of clinical neurology (Seoul, Korea)* 5 (Apr. 2009), pp. 11–9. DOI: `10.3988/jcn.2009.5.1.11`.

[13] Scott H. Hawley and Robert E. McClain. "Visualizing Sound Directivity via Smartphone Sensors". In: *The Physics Teacher* 56.2 (2018), pp. 72–74. DOI: `10.1119/1.5021430`. eprint: `https://doi.org/10.1119/1.5021430`. URL: `https://doi.org/10.1119/1.5021430`.

[14] James Henry et al. "Pilot Study to Evaluate Ecological Momentary Assessment of Tinnitus". In: *Ear and hearing* 33 (Sept. 2011), pp. 179–290. DOI: `10.1097/AUD.0b013e31822f6740`.

[15] IEC. *IEC 61672-1:2013 Electroacoustics - Sound level meters - Part 1: Specifications*. 2013.

[16] Pawel Jastreboff. "Phantom Auditory Perception (Tinnitus): Mechanisms of Generation and Perception". In: *Neuroscience research* 8 (Sept. 1990), pp. 221–54. DOI: `10.1016/0168-0102(90)90031-9`.

[17] Chucri A. Kardous and Peter B. Shaw. "Evaluation of smartphone sound measurement applications". In: *The Journal of the Acoustical Society of America* 135.4 (2014), EL186–EL192. DOI: `10.1121/1.4865269`. eprint: `https://doi.org/10.1121/1.4865269`. URL: `https://doi.org/10.1121/1.4865269`.

[18] Chucri Kardous, Peter Shaw, and William Murphy. "Evaluation of smartphone sound measurement applications using external microphones - A follow-up study". In: vol. 139. May 2016, pp. 2036–2036. DOI: `10.1121/1.4950017`.

[19] Robin Kraft et al. "Combining Mobile Crowdsensing and Ecological Momentary Assessments in the Healthcare Domain". In: *Frontiers in Neuroscience* 14 (2020), p. 164. ISSN: 1662-453X. DOI: `10.3389/fnins.2020.00164`. URL: `https://www.frontiersin.org/article/10.3389/fnins.2020.00164`.

[20] Robin Kraft et al. "Comprehensive insights into the TrackYourTinnitus database". In: *Procedia Computer Science* 175 (2020). The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology, pp. 28 –35. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.07.008. URL: http://www.sciencedirect.com/science/article/pii/S1877050920316872.

[21] Robin Kraft et al. "Design and Implementation of a Scalable Crowdsensing Platform for Geospatial Data of Tinnitus Patients". In: *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*. 2019, pp. 294–299.

[22] Robin Kraft et al. "Efficient Processing of Geospatial mHealth Data Using a Scalable Crowdsensing Platform". In: *Sensors* 20.12 (2020). ISSN: 1424-8220. URL: https://www.mdpi.com/1424-8220/20/12/3456.

[23] Kari Suzanne Kraus and Barbara Canlon. "Neuronal connectivity and interactions between the auditory and limbic systems. Effects of noise and tinnitus". In: *Hearing Research* 288.1 (2012). Annual Reviews 2012, pp. 34 –46. ISSN: 0378-5955. DOI: https://doi.org/10.1016/j.heares.2012.02.009. URL: http://www.sciencedirect.com/science/article/pii/S0378595512000457.

[24] Dilip Kumar and Deepak Kumar. "Chapter 13 - Abatement of Noise Pollution". In: *Sustainable Management of Coal Preparation*. Ed. by Dilip Kumar and Deepak Kumar. Woodhead Publishing, 2018, pp. 279 –291. ISBN: 978-0-12-812632-5. DOI: https://doi.org/10.1016/B978-0-12-812632-5.00013-6. URL: http://www.sciencedirect.com/science/article/pii/B9780128126325000136.

[25] Berthold Langguth. "A review of tinnitus symptoms beyond 'ringing in the ears': a call to action". In: *Current Medical Research and Opinion* 27.8 (2011). PMID: 21699365, pp. 1635–1643. DOI: 10.1185/03007995.2011.595781. eprint: https://doi.org/10.1185/03007995.2011.595781. URL: https://doi.org/10.1185/03007995.2011.595781.

[26] Mark Lutman. "What Is the Risk of Noise-Induced Hearing Loss at 80, 85, 90 dB(A) and Above?" In: *Occupational medicine (Oxford, England)* 50 (June 2000), pp. 274–5. DOI: 10.1093/occmed/50.4.274.

[27] Meinard Müller. *Fundamentals of Music Processing*. Springer Verlag, 2015. ISBN: 978-3-319-21944-8.

[28] Enda Murphy and Eoin A. King. "Testing the accuracy of smartphones and sound level meter applications for measuring environmental noise". In: *Applied Acoustics* 106 (2016), pp. 16 –22. ISSN: 0003-682X. DOI: https://doi.org/10.1016/j.apacoust.2015.12.012. URL: http://www.sciencedirect.com/science/article/pii/S0003682X15003667.

[29] Casey O'Callaghan. "Auditory Perception". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2020. Metaphysics Research Lab, Stanford University, 2020.

[30] R. Pierre and D. Maguire. "The Impact of A-weighting Sound Pressure Level Measurements during the Evaluation of Noise Exposure". In: *NOISE-CON*. 2004.

[31] Christopher J. Plack. *The Sense of Hearing - Second Edition*. London: Psychology Press, 2013. ISBN: 978-1-134-71668-5.

[32] Kelly Reavis, Janice Chang, and Fan-Gang Zeng. "Patterned sound therapy for the treatment of tinnitus". In: *The Hearing Journal* 63 (Nov. 2010), pp. 21–22,24. DOI: 10.1097/01.HJ.0000390817.79500.ed.

[33] Jens Rindel. "Book Review: Room Acoustics, sixth edition, by Heinrich Kuttruff". In: Aug. 2017. DOI: 10.13140/RG.2.2.28652.46724.

[34] Winfried Schlee et al. "Measuring the Moment-to-Moment Variability of Tinnitus: The TrackYourTinnitus Smart Phone App". In: *Frontiers in Aging Neuroscience* 8 (2016), p. 294. ISSN: 1663-4365. DOI: 10.3389/fnagi.2016.00294. URL: https://www.frontiersin.org/article/10.3389/fnagi.2016.00294.

[35] Ervin Sejdić, Igor Djurović, and Jin Jiang. "Time–frequency feature representation using energy concentration: An overview of recent advances". In: *Digital Signal Processing* 19.1 (2009), pp. 153 –183. ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2007.12.004. URL: http://www.sciencedirect.com/science/article/pii/S105120040800002X.

[36] Eberhard Sengpiel. *Sengpiel Audio Berlin*. URL: http://www.sengpielaudio.com/.

[37] Josef Shargorodsky, Gary Curhan, and Wildon Farwell. "Prevalence and Characteristics of Tinnitus among US Adults". In: *The American journal of medicine* 123 (Aug. 2010), pp. 711–8. DOI: 10.1016/j.amjmed.2010.02.015.

[38] Ethan Weil. *Short time fourier transform*. 2006. URL: https://upload.wikimedia.org/wikipedia/commons/9/9b/Short_time_fourier_transform.PNG.

[39] Xueqing Zhang. "Noise Prediction of a NACA 0015 Airfoil with Gurney Flap: A study based on Time-Resolved Particle Image Velocimetry". PhD thesis. Jan. 2016.

[40] Dr. Markus Ziegler. *Bedienungsanleitung Schallanalyator*. Available at https://spaichinger-schallpegelmesser.de/Anleitung_Schallanalysator.pdf, version 2.2. Sept. 2019.

[41] Dr. Markus Ziegler. *Spaichinger Schallpegelmesser*. Available at https://spaichinger-schallpegelmesser.de/Anleitung_Spaichinger_Schallpegelmesser.pdf, version 4.3. Dec. 2019.

[42] E. Zwicker. *Psychoakustik*. Hochschultext. Springer Berlin Heidelberg, 2013. ISBN: 9783642685101. URL: https://books.google.de/books?id=xXOiBgAAQBAJ.

Name: Johannes Vedder

Matriculation Number: 691733

Declaration

I hereby confirm that I created this work on my own and that I have not used any other materials than the ones referenced to in this thesis.

Ulm, 13.10.2020 ....

Johannes Vedder