Unterschrift Betreuer

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DIPLOMARBEIT

Microphone Localisation using Sound Waves and Multilateration

ausgeführt unter der Betreuung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Martin Gröschl
Institut für Angewandte Physik

und der Betreuung von Dr. Christoph Reichl, AIT – Austrian Institute of Technology, Center for Energy, Sustainable Thermal Energy Systems

durch

Peter Wimberger, BSc

Speichberggasse 9, 3002 Purkersdorf

October 20, 2020

Unterschrift Student

Abstract

This thesis has been done in the framework of the SilentAirHP research project conducted at AIT – Austrian Institute of Technology, which aims to reduce the sound emission of air-to-water heat pumps. In this framework, an acoustic dome with 64 microphones is used to capture frequency-, time-, and space resolved acoustic data. Exact coordinates of all 64 microphones are needed to ensure reproducibility of measurements, as foundation for other algorithms (e.g. sound power level analysis) and for visualization purposes (plots and augmented reality applications), but are difficult and imprecise to produce by hand.

In order to automatically and efficiently determine the position of these microphones, i.e. their spatial coordinates, an acoustic solution using multilateration and Time Difference of Arrival is being developed. Multilateration calculations use distances between multiple objects to obtain relative coordinates between those objects, similar to (satellite-based) navigation systems, such as GPS. The problem solution should be made modular and usable independent of the hard- and software environment, measurement procedure, and application.

The calculation procedure starts by playing predetermined audio samples on speakers with known coordinates and recording the microphone data. By correlating the sent with the recorded data, it is possible to extract time offsets, which are caused by different distances from each microphone to each speaker. These time offsets (or 'delays'), the precisely known speakers' coordinates, and the temperature-dependent speed of sound are used as input for the multilateration algorithm, which returns the microphones' coordinates.

Tasks therefore included: researching literature, deciding on hardware and software to use, designing the optimal speaker output sample, developing a stable application to be used by people without programming experience, using robust algorithms to record and pre-process audio data, implementing data correlation methods and the multilateration algorithm. Additionally, the procedure had to be tested thoroughly and implemented in the existing workflow, including data acquisition, analysis and visualization.

For measurements in line of sight the system works within the accuracy required for the application, but performance degrades in settings with too many multipath effects (reflection and diffraction), which causes unrecoverable loss of information.

SilentAirHP is supported in the framework of the Energy research program of the Climate and Energy Fund (5148527) initiated by the Austrian Ministry for Transport, Innovation and Technology.

Kurzfassung

Diese Arbeit wurde im Rahmen des am AIT – Austrian Institute of Technology durchgeführten Forschungsprojekts SilentAirHP durchgeführt, das die Reduktion von Schallemissionen von Luft/Wasser-Wärmepumpen zum Ziel hat. In diesem Projekt wird ein akustischer Dom mit 64 Mikrofonen verwendet, um frequenz-, zeit- und raumaufgelöste akustische Daten zu erfassen. Exakte Koordinaten aller 64 Mikrofone werden zur Sicherstellung der Reproduzierbarkeit der Messungen, als Grundlage für andere Algorithmen (z.B. Schallleistungspegelanalyse) und zu Visualisierungszwecken (Plots und Augmented-Reality-Anwendungen) benötigt, sind aber von Hand nur schwer und ungenau zu erhalten.

Um die Position dieser Mikrofone, d.h. ihre Raumkoordinaten, automatisch und effizient zu bestimmen, wird eine akustische Lösung unter Verwendung von Multilateration und Time Difference of Arrival entwickelt. Multilaterationsberechnungen verwenden Entfernungen zwischen mehreren Objekten, um relative Koordinaten zwischen diesen Objekten zu erhalten, ähnlich wie bei (satellitengestützten) Navigationssystemen wie GPS. Die Problemlösung soll modular und unabhängig von der Hard- und Softwareumgebung, dem Messverfahren und der Anwendung nutzbar gemacht werden.

Das Berechnungsverfahren beginnt mit der Wiedergabe von vorgegebenen Samples auf Lautsprechern mit bekannten Koordinaten und der Aufzeichnung der Mikrofondaten. Durch die Korrelation der gesendeten mit den aufgezeichneten Daten ist es möglich, Zeitverschiebungen zu extrahieren, die durch unterschiedliche Abstände von jedem Mikrofon zu jedem Lautsprecher verursacht werden. Diese Zeitverschiebungen, die genau bekannten Lautsprecherkoordinaten und die temperaturabhängige Schallgeschwindigkeit werden als Input für den Multilaterationsalgorithmus verwendet, der die Koordinaten der Mikrofone zurückgibt.

Zu den durchgeführten Aufgaben gehörten daher: Literaturrecherche, Entscheidung über die zu verwendende Hard- und Software, Entwurf des optimalen Lautsprecherausgangssamples, Entwicklung einer stabilen Anwendung, die auch von Personen ohne Programmiererfahrung verwendet werden kann, Verwendung robuster Algorithmen zur Aufzeichnung und Vorverarbeitung von Audiodaten, Implementierung von Datenkorrelationsmethoden und des Multilaterationsal-

gorithmus. Zusätzlich musste das Verfahren gründlich getestet und in den bestehenden Arbeitsablauf implementiert werden, einschließlich Datenerfassung, Analyse und Visualisierung.

Für Messungen mit Sichtverbindung der Lautsprecher zum Mikrofon funktioniert das System innerhalb der geforderten Genauigkeiten, aber die Leistung lässt in Umgebungen mit zu vielen Mehrwegeffekten (Reflexion und Beugung) nach, was zu nicht wiederherstellbaren Informationsverlusten führt.

SilentAirHP wird im Rahmen des Energieforschungsprogramms des Klima- und Energiefonds (5148527) unterstützt, das vom österreichischen Bundesministerium für Verkehr, Innovation und Technologie initiiert wurde.

Contents

Figure 1: MicLocator - Icon of the software developed for this thesis
Made with icons from Flaticon (`https://www.flaticon.com/`) from Authors smashicons and dave-gandy. All licensed under Creative Commons BY 3.0 (`http://creativecommons.org/licenses/by/3.0/`)

Nomenclature

P	Sound power	W
SPL	Sound pressure level	dB
c	Speed of sound	m/s
p	Momentary pressure of sound waves	Pa
W_{wf}	Continuous waveform (of type wf)	
f	Frequency of a sound wave	Hz
ADC	Analog-to-Digital Converter	
DAW	Digital Audio Workstation	
ρ	Density of air	kg/m^3
p_{abs}	Absolute air pressure	Pa
DBL	Device Based Localisation	
FFT	Fast Fourier Transform	
GNSS	Global Navigation Satellite System	
GPS	Global Positioning System	
IoT	Internet of Things	
LORAN	LOng RAnge Navigation	
MBL	Monitor Based Localisation	
RToF	Return Time of Flight	
SONAR	SOund Navigation And Ranging	
TDoA	Time Difference of Arrival	
ToA	Time of Arrival	

1 Introduction

To reach ambitious climate goals, heating technologies with low or even zero emission of greenhouse gases have to become widely used and adapted. One such technology are heat pumps, which effectively use the thermodynamic process of a refrigerator to heat up a medium, commonly water, using electrical power and a (colder) heat reservoir, which, in the case of air-to-water heat pumps, is the outside air. Using a heat pump is more efficient than heating directly with electrical power, measured by the Coefficient of Performance (CoP) with typical

values of 3 to 5 (produced heating power over input power).

The commercial success of air-to-water heat pumps is hindered by their unwanted sound emission. This fact was the reason for our group at the AIT (Austrian Institute of Technology) to start the project 'SilentAirHP' with the goal to investigate existing, and developing novel methods to reduce sound emission of air-to-water heat pumps. Work done included quantitative measurements and analysis of existing special components, such as fans, compressors and evaporators, using a specially built heat pump and the development of acoustic objects in 1-dimensional process models, which allow process control optimizations to pay respect to sound emission and acoustics. Acoustic measurements are done in a climate chamber with controllable air temperature and humidity.

I became part of this project to work on my bachelor thesis [3], which involved tasks such as setting up all necessary hardware, finding suitable software for audio recording and developing analysis scripts and tools.

For analysis and visualization purposes accurate information about the microphones' positions is needed. In detail, automated microphone position determination (localisation and tracking) in our setting is important for:

- Visualization in post-analysis 3D plots and virtual reality
- Sound source localisation
- Real-time visualization in virtual reality, especially relevant in an Industry 4.0 environment
- Subsequent algorithmic analysis, such as finite element methods or the exclusion of unwanted effects such as outside noise and reflection

for settings on a stage:

- Automated acoustic parameter optimization in room acoustics
- Automated light spot tracking

and for:

- Settings where electromagnetic tracking is not possible or not desired

Manual measurement is time intensive and prone to errors, both in regards to inaccuracy and unintended re-positioning after distance measurement (i.e. running into the microphone holders while working in the climate chamber). Tracking systems using optical or other electromagnetic waves exist, but they are very expensive. An automated position tracking system using sound waves could use the measurement system already in place, would be cheap and a nice engineering challenge. Additionally, it would allow recurring checks, that the microphones' positions did not change after heat pump maintenance. It would basically unite the concept of localisation of GPS (namely multilateration) and the Time of Flight principle used by SONAR systems.

The research presented in this thesis has been published and presented in numerous papers and conference talks written or held by team members, such as:

- Frosting and Defrosting Behavior of Evaporators, 13th IEA Heat Pump Conference, Jeju Island, South Korea, April 2021 (accepted). [4]
- Akustik von Wärmepumpen mit speziellem Fokus auf Vereisung, Abtauung und Platzierung, Chillventa Congress 2020, online (accepted), October 2020. [5]
- Acoustic behaviour and placement of heat pumps - The perception of sound and heat pumps, The Essence of Heat Pumps Series, Webinar online, September 2020 [6].
- Akustische Optimierung von Wärmepumpen (IEA HPT Annex 51), 26. Tagung des BFE-Forschungsprogramms „Wärmepumpen und Kälte“, BFH Burgdorf, June 2020. [7]
- Frosting Soft Sensor, IEA HPT – IOT Annex, IoT Heat Pump Conference, AIT Austrian Institute of Technology, Vienna, January 2020. [8]
- SilentAirHP Projekt Endbericht, 2019. [9]
- Annex51 and Experiences, Sound Workshop, Vienna, October 2019. [10]

- Poster: Acoustic Emissions and Noise Abatement of Air to Water Heat-Pumps, ICR 2019 - 25th IIR International congress of refrigeration, Montreal, Canada, August 2019. [11]
- Akustische Emissionen von Wärmepumpen, Chillventa Congress 2018, 5. Innovationstag Kältetechnik, Messe Nürnberg, Germany, October 2018. [12]
- International research: acoustic signatures of heat pumps, 11de Warmtepomp Synposium, Communicatiehuis, Gent, Belgium, October 2018. [13]
- MicLocator - Determine multiple microphones' positions using sound wave delay and trilateration, 68th Annual Meeting of the Austrian Physical Society, TU Graz, September 2018. [14]
- OpenFOAM implementation of algebraic frosting model and its applications on heat pump evaporators, 13th IIR Gustav Lorentzen Conference on Natural Refrigerants (GL2018), Valencia, Spain, June 2018. [15]
- IEA HPT Annex 51: Acoustic Signatures of Heat Pumps Update - Acoustic Transmission Measurements and Sound Source Detection, 26th Ercoftac ADA Pilot Center Meeting, Graz, November 2017. [16]
- Experimental and numerical methods for the fluid dynamic and acoustic characterization of heat exchanger icing, 67th Annual Meeting of the Austrian Physical Society (and Swiss Physical Society), CERN and CICG, Geneva, Switzerland, August 2017. [17]
- Icing of heat exchangers by measurements and simulations on micro- and macroscale, 25th ERCOFTAC ADA PC Meeting, Ercoftac Spring Festival, AIT Austrian Institute of Technology, Vienna, April 2017. [18]
- SilentAirHP - Analyse und Entwicklung von Schallreduktionsverfahren für Luft-Wasser-Wärmepumpen, Fortschritte der Akustik - DAGA2017, Kiel, Germany, March 2017. [19]
- Aktive Störschallunterdrückung für Wärmepumpenanwendungen, Fortschritte der Akustik - DAGA2017, Kiel, Germany, March 2017. [20]

- Charakterisierung der Schallabstrahlung von Luft-Wasser-Wärmepumpen mittels simultaner Hitzdrahtanemometrie, Vibrationsmessung und Schalldruckbestimmung, Fortschritte der Akustik - DAGA2017, Kiel, Germany, March 2017. [21]

- Transient Acoustic Signatures of the GreenHP with special focus on icing and defrosting, Proceedings of 12th IEA Heat Pump Conference, Rotterdam, Netherlands, May 2017. [22]

- GreenHP: Strömungs-Analyse der Verdampfer-Luftseite, DKV-Tagung 2016, Kassel, Germany, November 2016. [23]

- Active Noise Cancelling for Heat Pump Applications, 66th Annual Meeting of the Austrian Physical Society, Universität Wien, September 2016. [24]

- Space-, time- and frequency resolved recording and analysis of sound emissions and sound source localisation using a multichannel measuring system, 66th Annual Meeting of the Austrian Physical Society, Universität Wien, September 2016. [25]

2 Theoretical Background

2.1 Principal procedure

The principal procedure for microphone localisation developed in this thesis is the following: a predefined special audio signal is sent to each of multiple speakers with a known time offset, one after the other. The microphone records all incoming audio signals. The time differences from the mentioned time offsets and measured time offsets arise from the distance traversed by the sound wave from speaker to microphone (time of arrival). The measured time offsets are calculated by peak-picking the cross-correlation of outgoing and incoming signal. These time differences, one for each speaker, together with the pre-determined speaker coordinates are then put into a multilateration algorithm producing the microphone's coordinates with respect to the speakers coordinates system. This process can be parallelized for all microphones in question.

2.2 Localisation methods

The localisation, i.e. position determination, of objects is fundamental to society today. Starting from technology developed mostly for warfare, many technologies including RADAR, LIDAR, SONAR, LORAN and global navigation satellite systems (GNSS) including GPS have been developed and are in widespread and diverse use today. Usage includes applications in:

- airplane, drone and ship navigation systems
- street traffic for navigation and orientation of autonomous cars
- tracking of animals or autonomous machines in remote regions or over large areas
- mapping of large surfaces (oceans and land)
- localisation and tracking of end user devices over cellular and, more recently, WiFi networks [26]
- (involuntary or non-cooperative) localisation and tracking of missiles and harmful objects

A detailed survey of localisation technologies (with a focus on Internet of Things applications) can be found in [27]. Although this paper is primarily focused on end-user devices and advertising opportunities, it includes a general overview of localisation techniques such as RSSI (Received Signal Strength Indicator), CSI (Channel State Information), AoA (Angle of Arrival), ToF/ToA (Time of Flight/Time of Arrival), TDoA (Time Difference of Arrival), RToF (Return Time of Flight) and PoA (Phase of Arrival) and technologies like WiFi, Bluetooth, UWB (Ultra Wideband), Visible Light and sound waves.

Several Big Tech companies have developed applications, protocols and frameworks using primarily Bluetooth, but also WiFi and near-ultra sound signals to detect close proximity and/or localisation of end-user devices, e.g. smartphones in stores. These include Apple's iBeacons and Google's eddystone beacons. Such systems can either function without explicit allowance of the device's user (by using technology aspects with weak privacy) or without deep understanding of the user, by agreeing to lengthy hard-to-understand agreements. Data from beacons can be combined with data sets generated by GNSS (GPS) tracking used in mobile phone apps to enable more meaningful data analysis. GNSS data collection has to be approved by the app user, but this explicit approval can be obfuscated by hard-to-understand technical terms or exploiting technological complexities.

One has to distinguish between two different modes of localisation techniques, as named by [27]:

- Device Based Localisation (DBL), where an entity (device, user, microphone) determines their position in regards to multiple reference nodes (beacons, speakers) and
- Monitor Based Localisation (MBL), where reference nodes are used to track the entity in question

In DBL the reference nodes usually act as transmitter, where as in MBL the entity to track usually is the one to send signals, which are then received by the reference nodes. In the paper mentioned above, a third quasi-localisation mode is introduced, proximity detection, which is the process of 1-dimensional distance estimation between an entity and the reference node (Point of Interest). This

mode of operation can e.g. be used for location-based advertisement campaigns.

A similar technology is currently being used in apps to track exposure to SARS-CoV-2 positive persons (contact-tracing) in the ongoing pandemic. Two smartphones act as both sender and receiver of Bluetooth-Low-Energy messages. Privacy compliance was achieved by a centralized implementation on the operation system level (Apple's iOS and Google's Android) and decentralized, anonymous device identification.

Proximity detection based on sound waves is used in smartphone apps to detect another device nearby, which may both have an internet connection, but are not on the same network or have no Bluetooth capability. This procedure is often used to establish a connection (pairing; via WiFi or Bluetooth) in a privacy compliant way.

2.3 Localisation technologies

The system developed for this thesis uses the concept of TDoA (Time Difference of Arrival), Multilateration (MLAT, section 2.5) and "Device Based Localisation" as the speakers (reference nodes) play (send) a sample and the microphone records it (multiple microphones are analyzed independently of each other.) In this regard, the system functions similarly to the LORAN (Long Range Navigation) system developed during World War II by the USA and used extensively throughout the 20th century, primarily by ships. Nowadays GNSS (Global Navigation Satellite Systems) enable more precise localisation globally. LORAN uses signal emission stations with a fixed position, pulsed signals and a pre-defined emission delay between each reference node. LORAN systems measure the time difference between each incoming signal pulse to calculate the distances of signal travel, the so-called pseudo ranges. Another widely used navigation system in the 20th century is the Decca system, which uses phase differences to generate pseudo ranges. Fundamental differences from the approach discussed in this paper and LORAN are the usage of sound waves over electromagnetic waves and analysis done in 3D space, not on the quasi-2D plane of the earth's surface.

For accurate position determination, the location of all reference nodes needs to be known exactly at the time of their signal emission, either in regards to each other in a local coordinate system or measured in a global coordinate sytem. In

indoor settings reference nodes are usually fixed in place causing no complication in the equations, but in more sophisticated settings, such as GNSS, trajectories have to be controlled and corrected for and trajectory information transmitted to the receiver or known a priori. In GNSS, the satellites' orbits are monitored and controlled from the ground. The satellites carry a high-quality atomic clock to enable signal emission with a detailed time code, which allows signal receiving devices to calculate the accurate satellite's trajectory.

For position determination in 3D space, three independent signal travel path lengths are needed. The path length (often called pseudo range to emphasize its uncertainty) is given by the signal propagation velocity times the time passed since emission. This time difference can only be determined if the absolute emission time is known to the receiver, i.e. both the reference nodes and the receiver are time synchronized. This can be achieved either via radio (e.g. via internet), or, via encoding of the absolute time information in the signal itself. This encoded time information can be recursively interpreted with a valid localisation to eliminate delay from signal pathways. In TDoA applications an additional reference node is needed to provide enough information to eliminate the need for knowledge about absolute time in the equations, because four reference nodes only provide information of three time differences (pseudo ranges). In any case, the reference nodes have to be time synchronized between themselves to obtain valid information. One therefore needs at least three or four reference nodes for localisation in 3D space. Of course, with the use of robust algorithms, an increasing number of reference nodes also increases localisation accuracy.

In most applications, bandwidth is limited and digital signals have to be used. These signals can then encode information about emission times and reference node positions in the signal itself. Global navigation satellite systems use their own time system with a few seconds difference to UTC (Coordinated Universal Time) as they do not undergo leap second adjustments. In GPS for example, the current time can be recovered within an accuracy of 1 μs, because data packets include information about the current time in the GPS time system and its current difference to UTC in seconds. Additionally, GNSS data messages transmit many more parameters, including the orbital flight parameters (ephemerides), ionospheric correction parameters and other clock correction

parameters, and the almanac. The almanac represents a more general information of all system satellites' orbits to be able to localize them quicker after signal loss by estimation of expected time differences and expected doppler frequency shifts.

If all reference nodes use the same frequency band, a unique identifier has to be included, either digital or analog.

The system developed in this thesis does not use digital signals, but fast changing continuous frequency sweeps over a wide frequency range. This approach allows use of unambiguous cross correlation peak picking, even when using multiple signals concurrently.

All localisation methods share the error source of multipath effects, including reflection, diffraction, scattering and refraction, which are often critical, when the direct line of sight is obscured as is commonly the case in indoor environments. All multipath effects cause an increase of the pseudo-range measurement. Without additional knowledge, it cannot be known if a wrong measurement has been made and how big the increase is.

2.4 Acoustic sound source localisation

The use of acoustic waves in localisation environments has advantages and disadvantages compared to the use of electromagnetic waves. Advantages include high accuracy and cheap hardware. Disadvantages can be the influence of sound pollution (noise) and potential disturbance for humans and animals.

Localisation using acoustic waves is often used for (uncooperative) tracking of entities, such as humans, smartphones, guns, etc. and measurements of wildlife, oceanic life and ocean floor elevation using SONAR. Sound wave localisation techniques using the "DBL" (navigational) approach are not widely used and developed yet.

Acoustic source localisation has been used in military context since before the first world war (i.e. artillery sound ranging). Sound source localisation in these times was done by humans exploiting the interaural time difference, i.e. the time difference sound waves travel from one ear to the other, see fig. 2. Nowadays, stationary systems using acoustic scene classification and sound ranging are used in the military context to automatically alarm of and locate the

Figure 2: Sound location equipment "Horchgerät" in Germany, 1939. It consists of four acoustic horns, a horizontal pair and a vertical pair, connected by rubber tubes to stethoscope type earphones worn by the two technicians left and right. The stereo earphones enabled one technician to determine the horizontal direction and the other the elevation of the aircraft.
Bundesarchiv, Bild 183-E12007 / Eisenhardt / CC-BY-SA 3.0, CC BY-SA 3.0 DE (`https://creativecommons.org/licenses/by-sa/3.0/de/deed.en`)

firing of guns. This approach is e.g. used in war zones, big cities and national parks to prevent wildlife poaching. Portable systems, e.g. carried on the body of soldiers, are also in deployment.

2.5 Trilateration and Multilateration - Localisation algorithms

Multilateration (MLAT) is the process of determining an estimate of the receiver's n coordinates from measurements of time differences by signals sent from at least $n + 1$ reference nodes with a precisely known location (TDoA, Time Difference of Arrival). A pair of reference nodes form a hyperbolic curve for the device's possible location, coining the term hyperbolic (navigation) system.

Trilateration is the special case for $n = 3$. Three intersecting spheres (with pseudo-ranges as radii) in 3D generate two (symmetric) points, making a solution ambiguous in principal. However, one solution can often be ruled out based on plausibility. To add information to the system, reference nodes must not be positioned with its center on a co-linear line or co-planar plane.

Murphy, Hermen et al. presented in [28] and [29] the calculation of multilateration solutions with different algorithms. The paper's nonlinear least squares (NLLS) approach is presented in this chapter, as computer hardware is fast enough nowadays to evaluate all factors quickly for low dimensional systems (amount of reference nodes) as it is the case here.

Starting from the equation for an outgoing spherical wave from (x_i, y_i, z_i), where i is the index of each reference node,

$$r_i^2 = (x - x_i)^2 + (y - y_i)^2 + (z - z_1)^2 \tag{1}$$

the aim is to minimize the error d_i of the measured pseudo range s_i

$$d_i = l_i - s_i \tag{2}$$

with

$$l_i = \sqrt{r_i^2} = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}. \tag{3}$$

Because the pseudo range s_i is not directly accessible, as only time differences between each incoming wave front are measured, this equation has to be modified by using one reference node (with index k) as the time zero-point from which

the time differences are calculated from.

$$f_i = l_i - l_k - c_{Air}\Delta t_i \tag{4}$$

The loss function to be minimized can then be defined as the sum of least squares, i.e. the f_i squared,

$$F(x, y, z) = \sum_i f_i(x, y, z)^2. \tag{5}$$

Each measurement (error) is hereby treated equally. The summation index i should be adjusted to omit k ($i \neq k$) to save processing time, but does not contribute anyway. This loss function can be minimized by the Gauss–Newton method, because the first derivative for each coordinate can be computed analytically as following (for y and z analogous)

$$\frac{\partial F}{\partial x} = 2\sum_i f_i \frac{\partial f_i}{\partial x} \tag{6}$$

and

$$\frac{\partial f_i}{\partial x} = \frac{x - x_i}{l_i} - \frac{x - x_k}{l_k} \tag{7}$$

the Gauss-Newton method iteratively finds the zero-point $\vec{R_c}$ of the function f in question with

$$\vec{R_{c+1}} = \vec{R_c} - (\mathbf{J}_c^T \mathbf{J}_c)^{-1} \mathbf{J}_c^T f_{c,i}, \tag{8}$$

where c denotes the c-th iteration cycle and $\vec{R_c} = (x, y, z)$ the current best estimate for the coordinates of the receiver (microphone).

$\mathbf{J}_c$ is the Jacobian matrix with

$$\mathbf{J}_c = \mathbf{J}_{ij,c} = \frac{\partial f_i(\vec{R_c})}{\partial j}, \tag{9}$$

where i iterates over the reference nodes and i over the coordinates (x, y, z), evaluated at $\vec{R_c}$. The factor $(\mathbf{J}_c^T \mathbf{J}_c)^{-1} \mathbf{J}_c^T$ is commonly known as the left generalized (pseudo)inverse of J_c. Since the Gauss–Newton method is iterative, it requires an initial solution estimate $\vec{R_0}$, which needs to be inside the hull spanned by

the reference nodes. The algorithm approaches the zero-point (most likely coordinates from time difference measurements) fairly quickly, if only one global minimum is present.

2.6 Cross-correlation

The cross-correlation of two discrete time-series is a measure for similarity in dependence of displacement of each other:

$$(f \star g)[n] \;=\; \sum_{m=-\infty}^{\infty} \overline{f[m]}\, g[m+n] \tag{10}$$

In signal processing of acoustics, cross-correlation allows the measurement of the time delay of an echo and gives information about similarity of two signals, e.g. sent and received.

As signals are played with a known time offset for each speaker, the measured time differences between the signals give information about the time needed to transverse the medium (Time of Transmission). These time differences are measured by peak-picking the cross-correlation of the signals sent and the one received (recorded). The time differences are then converted to so-called pseudo ranges using the speed of sound $s = c_{Air} \cdot \Delta t$.

Our setup allows synchronous signal sending (sample playing), but the absolute time of signal output is not known for the receiver (microphone). This localisation method is named TDoA (Time difference of Arrival) and described in section 2.5.

Calculation of the cross-correlation is preferably done in the frequency domain, where a multiplication is equivalent to a convolution (and the, mathematically similar, cross-correlation) in the time domain (see section 3.4 for a detailed and continued discussion).

2.7 Sound propagation

Sound travels in air with a, compared to light, slow velocity of 343.2 m/s at a temperature of 20 $°C$.

$$c_{Air} = 331.5 + 0.6T \; m/s, \tag{11}$$

where T is given in $°C$.

Differences in velocities for different sound wave frequencies (dispersion) are extremely small [30], especially in frequency ranges perceived by the human ear.

With a sampling rate of 192 kHz, sound waves at 20 $°C$ therefore cover a distance of $343.5/192000 \approx 1.8$ mm per sample. With sample-accurate analysis, and considering the dimensions of a microphone membrane, the theoretical limit for microphone localisation accuracy is very good. Resolutions can be even further improved by using higher sampling rates at hardware level or interpolation/resampling of signals with frequencies far below the Nyquist-threshold. The Nyquist-threshold is given by halved sampling rate.

2.8 Optimal signal design

In order to efficiently and reliably determine the pseudo-range (effectively the propagation time of a sound signal) from source to receiver, a "good" signal must be found that has a sharp peak in the cross-correlation between the originally transmitted synthetic signal and the recorded signal. The original signal sent to the speakers therefore has to be designed to maximize the similarity under real-world influences, which include the following:

- Diffraction causing longer propagation paths and times and other multi-path effects
- Non-ideal properties of real speakers: Resonances causing different sound levels for different frequencies
- Unwanted artefacts caused by a finite sampling rate and digital to analog conversion
- Unwanted influence to recordings of other speakers

As explained in detail in section 3.3 the designed sample consists of multiple frequency sweeps parameterized by non-repeatable numbers to obtain a sharp

peak. Non-repeatable numbers can be obtained by a concept mainly used in information theory and computer science: the multiplication hashing method.

2.9 Non-repeatable numbers and hash functions

The need for non-repeatable numbers (in contrast to random numbers) arose during tests in the real world of cross-correlation peak-picking, as multiple peaks are generated for random numbers, that are similar to themselves.

After using random numbers to parameterize the generated sample, a notable effect on performance was observed for different randomization seeds. This odd discovery quickly pointed to the fact, that random numbers were not optimal, but non-repeatable numbers should be used instead. This behaviour can be linked to another field of study: algorithms and efficient data storage. When storing data efficiently, which must also be quickly retrievable, hashing functions are often used to map arbitrary data entries ("keys") to a predetermined finite set of hash codes. These keys usually take up less space than the original data and are sortable. Because of the near-infinite possibilities (limited only by memory/storage space) of input data and finiteness of the output for the hash function, so-called collisions can occur. A collision happens, when two different inputs produce the same output. For reasons of efficient data storage it is often beneficial that two similar, but different inputs produce vastly different outputs. Hash functions are used for checksums, fingerprints, compression algorithms, cryptography and much more. Depending on their specific application, hash functions are often expanded to more general hash algorithms.

A simple hash function is the multiplicative hashing method (multiplication-hashing function). It maps the product from the data ("key") k with a specific number A (given by the algorithm) and truncates the integer part to obtain a mapping to the [0,1) range. This decimal number is then multiplied by the hash table size and rounded down to obtain an integer. This hash function can be mathematically expressed as:

$$h(k) = \lfloor m \cdot (kA \bmod 1) \rfloor = \lfloor m \cdot (kA - \lfloor kA \rfloor) \rfloor \,, \qquad 0 < A < 1, \tag{12}$$

where $\lfloor x \rfloor$ is the floor function of x, rounding real numbers down to the next lower integer.

In practice, the optimal value for A for most applications was found to be the inverse golden ratio

$$A = \Phi^{-1} = \Phi - 1 = \frac{\sqrt{5}-1}{2} \sim 0.61803398875, \tag{13}$$

which is also used in this thesis.

To obtain non-repeatable numbers from this algorithm, we simply start with an arbitrary (non-zero positive) integer k and increment it for each value $h(k)$. This approach, in contrast to random numbers, also assures the same number sequence independent of operating systems and randomization seeds.

2.10 Sound power level

Microphones are able to measure changes in the air pressure (measured in Pa or dB), but for analysis of a machine's sound emission it is important to consider the total emitted sound power (in W). In contrast to the sound pressure, the sound power, if measured correctly, is independent of the receiver's distance from the sound source, the room size and wall absorption. This does not mean that the sound pressure is insignificant as it describes the sound field perceived by the human ear, which can cause damage or discomfort.

The sound power describes the amount of energy per time emitted by the sound source and is calculated using the product of the sound intensity and a hull area. Assuming lossless media, the total sound power radiated by a sound source is the same for every closed hull area around it, independent of size and form. For actual measurements it is important, that the sound intensity vectors are parallel to each other and perpendicular to the hull area.

Information on how to obtain a pressure value from a microphone signal can be found in section 3.1.

The sound pressure level (SPL) is given with unit signs of dB and can be calculated from a sound pressure given in Pa by

$$SPL[dB] = 10 \log_{10} \frac{p[Pa]^2}{p_0^2} = 20 \log_{10} \frac{p[Pa]}{p_0}, \tag{14}$$

where p_0 is the reference pressure of $p_0 = 2 \times 10^{-5}\ Pa$. The logarithmic nature of the sound pressure level captures the qualitative meaning of the Weber-Fechner law, which says that human sensory perception is proportional to the logarithmic of the actual physical stimulus. The formula maps pressure levels to a range from 0 to 100 dB for applications in everyday life, the same as many other quantities. 0 dB corresponds to the hearing threshold of the human ear at 1000 Hz and a sound pressure level of 100 dB is often encountered in discos, which most people would consider very loud. Hearing protection is recommended in these settings. Special transparent earplugs can be used for loud music. For construction, aviation or industrial settings multiple layers of hearing protection (plugs and headphones) may be required to protect the ear in a safe manner.

The inverse calculation (from dB to Pa) is given by:

$$p[Pa] = p_0\ 10^{\frac{SPL[dB]}{20}}. \tag{15}$$

Both means of measurement are important as dB gives a better understanding of the human perceived loudness than values given in Pascal, but for actual physical interactions, such as addition of absolute pressures values, only operations on pressure values in Pascal are valid.

The sound power is defined by the product of the sound intensity I and the area of a closed hull around the sound source A:

$$P[W] = I \cdot A = p \cdot v \cdot A, \tag{16}$$

where p is the sound wave's pressure (in Pa) and v the sound particle velocity (in m/s). $p = p(t)$ is of course actually a time series, but can be averaged for pure sine waves by calculating the root mean squared (rms) value. The particle velocity is not measurable with an ordinary microphone, but assumptions can be made to allow calculation of the sound intensity in the far field. The far field describes the region of distances for which the emitted waves can be assumed to be plane waves. In general, this region satisfies $k \cdot r \gg 1$, where k is the wavenumber and r the distance of the source to the receiver. At 20 °C ($c =$

343.2 m/s) k can be evaluated to

$$k = \frac{2\pi}{\lambda} = \frac{2\pi f}{c} = 0.01831 \cdot f \tag{17}$$

with f being the frequency and λ the wavelength. For a distance of 1 m the condition $k \cdot r \gg 1$ is met for frequencies greater than a few hundred Hz. In the far field the pressure and velocity waves are in phase. This means that the sound impedance Z is real and equal to the specific acoustic impedance of air $Z_0 = \rho \cdot c$, where ρ is the air density.

The sound intensity can also be expressed as $I = p^2/Z$. The sound power then becomes

$$P = I \cdot A = \frac{p^2 \cdot A}{Z_0} = \frac{p^2 \cdot A}{\rho \cdot c}. \tag{18}$$

The density ρ of dry air can be calculated using the ideal gas law:

$$\rho = \frac{p_{abs} \cdot M}{R \cdot T} = \frac{p_{abs}}{R_S \cdot T}, \tag{19}$$

where p is the absolute air pressure, M is the molar mass of dry air (0.0289 kg/mol), R the ideal gas constant (8.31), T the absolute temperature (in K) and R_S the specific gas constant given by $R/M = 287.058\ J/(kgK)$.

Equation 19 can be generalized for humid air by introducing a modified specific gas constant:

$$R_S \longrightarrow R_h = \frac{R_S}{1 - \phi \cdot \frac{p_d(T)}{p_{abs}} \cdot (1 - \frac{R_S}{R_v})}, \tag{20}$$

ϕ is the relative humidity, $p_d(T)$ is the temperature-dependent vapor pressure of water (e.g. 2338 Pa at 20 °C and 611 Pa at 0 °C) and R_v is the specific gas constant of water vapor (461.523 $J/(kgK)$). The vapor pressure of water p_d in dependence of temperature can be looked up in tables or calculated with a variety of equations. The table lookup approach was chosen because of its accuracy even for sub-0 °C temperature ranges. Experiments in the climate chamber typically model outside temperatures encountered in the European climate. The chamber is therefore capable of producing temperatures in the range of −18 °C up to 50 °C.

The air density therefore depends on knowledge about the absolute air pressure, temperature and humidity. The speed of sound is also temperature-dependent as given in Equation 11.

The sound power in dependence of absolute air pressure, temperature and humidity can therefore be calculated by combining Equations 18, 19, and 20:

$$P[W] = I \cdot A = \frac{p^2 \cdot A}{Z_0} = \frac{p^2 \cdot A}{\rho \cdot c(T)} = \frac{p^2 \cdot A}{p_{abs} \cdot c(T)} \cdot \frac{T \cdot R_S}{1 - \phi \cdot \frac{p_d(T)}{p} \cdot (1 - \frac{R_S}{R_v})} \tag{21}$$

The sound power is given in W, but similarly to the sound pressure levels, relationships can be better formulated by transforming the Watt value to decibel (dB). The reference sound power $P_0 = 10^{-12}\ W$ is motivated by the approximate lower limit of sound intensity perceptible by the human ear at 1000 Hz. A sound intensity level of 0 dB at 1000 Hz therefore is the hearing threshold, similar to the sound pressure level.

$$P[dB] = 10 \log_{10} \frac{P[W]}{P_0} \tag{22}$$

The calculation of the aforementioned representative areas A appearing in Equation 21 can be done manually by measuring the microphones' position and exploiting symmetry effects, which can be generated while setting up the microphone mounts. It can also be automated, after obtaining the microphones' positions by applying the Voronoi algorithm for each microphone to calculate the region of a plane closest to the microphone head. This algorithm is described in section 2.11.

2.11 Voronoi diagrams

Voronoi diagrams have many different applications in science and technology, but are also a tool commonly used in visual art and design.

The Voronoi algorithm (after Georgy Voronoy; also called natural neighbor interpolation or Dirichlet tessellation after Peter Dirichlet) separates a 2D plane or 3D body and a given set of points into multiple regions (called Voronoi cells or Thiessen polygons), in which each region corresponds to one point in the

sense that this point is closer than any other point. Figure 3 shows a small explanation plot and two examples of artistic usage. All Voronoi cells together extend over the whole plane or body.

This approach can therefore provide polygons representative for each point of a given set along a given (convex) hull in 3D. To obtain representative areas needed for sound power level calculation given the microphone geometry of the so-called dome (as seen in figure 5), the idealized cylinder surface has to be found and the microphones positioned accordingly. This cylinder surface has to be bounded by the floor and is then given to the Voronoi algorithm. This procedure has not yet been implemented.

Figure 3: Top left: A simple 2D visualization illustrating the usage of the Voronoi algorithm (see section 2.11). The red points are given to the algorithm, which outputs the vertices of the blue border polygons. Each polygon edge line lies exactly between two red points.
Top right: A 2D visualization with polygons colored artistically.
Bottom: Three (easter related) 3D printed models using Voronoi cells along a 2D hull (pre-modeled surface of a bunny, bunny head and egg) in 3D. The 3D printed models are obtained from:
virtox - Stanford Easter Bunny - Voronoi `https://www.thingiverse.com/thing:303842`
roman_hegglin - Bunny Head - Voronoi Style `https://www.thingiverse.com/thing:287884`
Antonin_Nosek - Easter Eggs `https://www.thingiverse.com/thing:2829553`

3 Methods & Implementation

The system developed for this thesis uses the concept of TDoA (Time Difference of Arrival), multilateration and "Device based localisation" as the speakers (reference nodes) play (send) a sample and the microphone records it. The procedure is the same for multiple microphones as they are analyzed independently of each other.

3.1 Hardware

3.1.1 Microphones and Clipping

Microphones transform the momentary displacement caused by the pressure differences of sound waves to an electrical voltage signal. These analog signals are amplified and then converted to a digital signal in an ADC (Analog-to-Digital Converter). The ADC is parameterized by the sampling rate and bit depth. The sampling rate describes how often the momentary voltage is measured. The bit depth describes the value of the maximal digital value. The ADC maps the momentary voltage value to the digital value in a (hopefully) linear fashion. In practice this means that there is a voltage limit above which all analog voltages are mapped to the maximal digital value (0 $dBFS$, where FS stands for Full Scale). This effect is called clipping. Clipping of course means a loss of information as it cuts off the highest and lowest parts of the waveform and has to be avoided with highest priority. In live settings, clipping is normally noticed quickly as it is audible, and the solution is to reduce the gain or apply compression. Scientific measurements though, if clipping is noticed during recording, need to be repeated, as the gain needs to be consistent over the measurement's duration. This can be very cumbersome if only noticed afterwards. The simple solution would of course be to have no gain whatsoever, but this is also not recommended, because sound analysis is typically done in the dB domain to reflect the logarithmic human sensory sensation. This means that loud signals near the maximal digital value have a much finer resolution (in the dB domain) than signals with low levels. Therefore the goal is ultimately to tune the gain to be as close as possible to the maximal digital value, but never go above it. In practice this means to excite the microphone with the loudest signal you expect

to happen in the experiment, then set the gain to a value of about $-3\ dB$ to have a safety buffer. This buffer is also called headroom and $3\ dB$ is already a very small headroom. Typically the loudest event in our measurements is the calibrator signals (explained below), so the small headroom is big enough and can be enlarged before a specific measurement, if louder levels are expected. In very loud settings, the physical components of the microphone head or inside the pre-amplifier and ADC can clip. This is only possible in very extraordinary situations and requires specialized hardware to be avoided.

Signal amplification of course needs to be adaptable and its control is often realized with a knob or fader. This makes sense for normal usage of audio equipment in studio or live settings, because its is quickly adjustable. But for scientific measurements a fixed or, better, reproducible amplification gain control is preferred to ensure consistency. (The equipment used in the SilentAirHP project allows digital and therefore reproducible control of the amplification factor. The gain can be controlled in $1\ dB$ steps from 0 to $+40\ dB$ in the RME Octamic XTC pre-amplifier settings. [3])

3.1.2 Sound Pressure Level Calibration

To be able to translate the digital value to a pressure at the microphone head, a calibrator and its careful usage is required. An acoustic calibrator is a special, mobile and battery-powered device which emits a signal with a certified sound pressure level and is put on a measurement microphone head. Typically this level is $94\ dB$ or $114\ dB$. These devices typically only fit on the standardized measurement microphone head sizes of $1''$ and $1/2''$. They send out a signal of $1000\ Hz$, which is the reference point for nearly all acoustic quantities and zero-point of all psychoacoustic weightings. After putting the calibrator on the microphone, the measured level output needs to be checked to ensure no clipping. If clipping occurs, the amplification gain needs to be reduced as described above. Now a short recording needs to be made (we use a duration of $10\ s$). Because of the high sound pressure level and encapsulation of the microphone head, background noise should not falsify the measurement, but it is good practice to ensure a quiet environment nonetheless. This recording can then be analyzed in the data analysis chain.

Recalling equation 14 for sound pressure level conversion (from Pascal to dB):

$$SPL[dB] = 10 \log_{10} \frac{p[Pa]^2}{p_0^2} = 20 \log_{10} \frac{p[Pa]}{p_0}, \tag{23}$$

where p_0 is the reference pressure of $p_0 = 2 \times 10^{-5}\ Pa$.

After trimming off a few samples of the signal to ensure a good signal, the mean is taken of all squared sample values s times a yet-to-be-determined calibration factor γ:

$$SPL_{ref} = 94\ dB \stackrel{!}{=} 10 \log_{10} \frac{\bar{p}[Pa]^2}{p_0^2} = 10 \log_{10} \frac{\gamma^2 \frac{\sum s^2}{n}}{(2 \times 10^{-5})^2}, \tag{24}$$

where n is the amount of elements in the sum (the length of the recording in samples)

The calibration factor γ is then given by

$$\gamma = 10^{\frac{1}{2}(\frac{SPL_{ref}}{10} - \log_{10}(\frac{\sum^n s^2}{n}) + 2\ (\log_{10}(2) - 5))}. \tag{25}$$

The calibration factor γ is of course dependent on the range of digital values, i.e. the bit depth of the ADC. A bit depth of 24 bits for example means a range of 2^{24} digital values going from -2^{23} to 0 up to $2^{23} - 1$.

3.1.3 Recording Setup

The recording setup consists of up to 64 MicW M215 Class 1 Measurement Microphones connected via XRL to 8 RME OctaMic XTC acting as Pre-Amplifier and ADC. The OctaMics are connected via optical MADI cables to 2 (of 3 available) internal RME HSDPe MADI FX audio interfaces. Their driver exposes the audio signals as ASIO channels, usable by a dedicated recording program (DAW) or sound libraries (like `python`'s `sounddevice`). The 64 microphones are normally set up in a dome-like geometry with mounts specifically produced for this setup as seen in figure 5.

The smaller setup used in conjunction of the SoundDAQ system (section

3.6.2) for long-lasting recording measurements uses 5 measurement microphones and a Focusrite Scarlett 18i20 external audio interface connected via USB to the PC. Again, the interface's driver exposes ASIO channels usable for other software. The setup for 5 microphones uses four normal and one overhead microphone stand and can be seen in figure 14.

Of course many more complications need to be considered for a working setup, such as MADI synchronisation in cable loops, correct automated set-up of sampling rates and software workspaces and ensuring correct cable connections. A more detailed description of the hardware used for the recording process can be found in the preceding work (bachelor thesis) [3].

3.1.4 Speaker setup

To perform acoustic microphone localisation, a speaker setup is needed. Our setup uses 6 speakers, as this is a sufficient amount ($n > 3$) and we have 3 audio interfaces readily available with a 6.35 mm stereo jack output each. These outputs are connected to 3 dedicated amplifiers. Amplification can be controlled freely, but should be as high as possible for a good and loud signal, but not too high as the speakers would start to clip physically, distorting the signal and rendering it useless. 6 individual mono speakers are connected to the (stereo) amplifiers. The speakers are mounted on 3 walls of the climate chamber (2 speakers per wall). The fourth wall is completely covered with fabric bags continuously filled with temperature-controlled air from the climate control system above and below the climate chamber and is therefore not suitable for mounting speakers on (as seen on the right side of figure 5). As described in section 2.5, all reference nodes of multilateration are optimally placed in a non-co-planar way. Considering the z-axis pointing upwards, the x and y coordinates of all reference nodes exhibit near-maximal variance. For optimal multilateration performance, the z-axis height also needs to be varied. The limiting factors upwards are struts going across the room at a height of about 3 m, which would obstruct the signal path if in line of sight between the speaker and microphone. The limiting factor to the ground is the occurrence of multipath effects as reflection off the floor (as described in section 3.5), if the speakers are placed too low. Wall reflections are not an issue, because the walls are covered with thermal insulating polystyrene

foam acting as a mediocre sound-proofing foam. The floor is covered with a smooth and flat metal surface. This specific floor surface was installed as a robust ground to place heavy heat pumps on, which also acts as a worst-case situation of audio reflection in the real world and to easily remove standing water (by using a squeegee to move it to the drain).

The speakers' location needs to be known very precisely for an accurate multilateration result. The speakers therefore have to be fixed in place and their location measured by any means, such as a tape measure or laser distance meter before making acoustic measurements. This procedure should ideally be done while the climate chamber is empty. The speakers used are about 2 cm wide and its midpoint is used as reference. The maximum angle of sound emission is near 180° causing no problems. Furthermore their directional characteristics is not important as absolute signal loudness does not matter, just that each microphone can record a good signal for a high-quality cross-correlation.

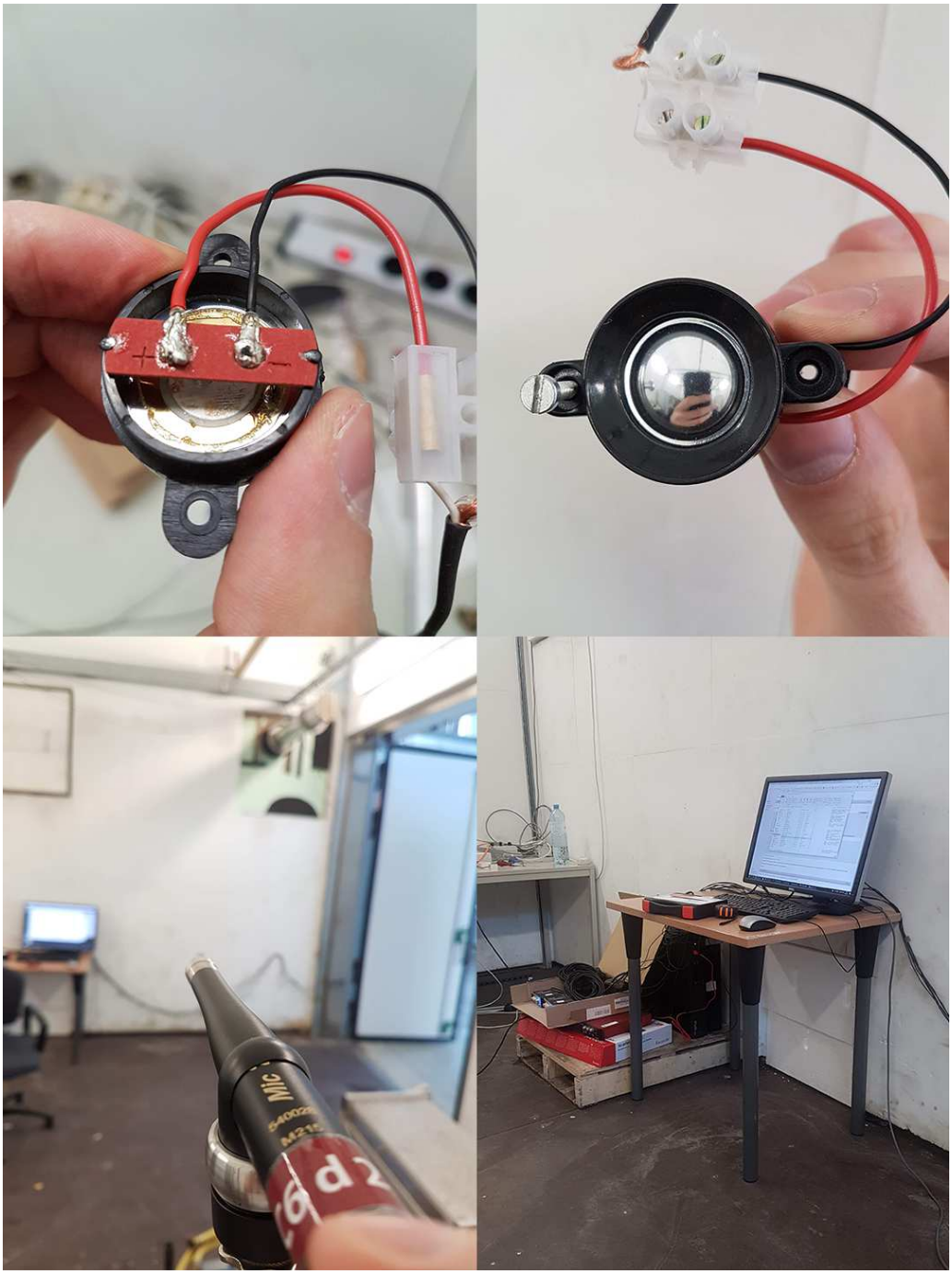

Figure 4: Top images: One of the 6 speakers used for playing the sample/signal for microphone localisation. These speakers are mounted to the walls of foam in the climate chamber with screws and nails. Bottom left: A measurement microphone illustrating the pathway (without obstructions) of sound waves coming from a speaker on the opposing wall (directly behind microphone head; not visible). Bottom right: A temporary setup of the recording station with software running on a Desktop PC with the USB audio interface connected and the amplifiers inside a cardboard box on top.

Figure 5: Photograph of the 64 channel setup in the dome arrangement with a six-sided symmetry. 61 microphones are used in this geometry. 8 arms are mounted to each of the six stands with one metal rod going to the middle. Microphones are positioned on each arm to maximize spread along the cylinder's surface area. Individual adjustments can be made if strong sound emission directivity is suspected. Two microphones are mounted on each top rod and one microphone is placed in the middle. In operation, the heat pump is placed in the middle of the microphone dome. To change hardware parts for different measurements, the arms need to be moved to gain access to the pump. Afterwards the arms need to be readjusted, which can alter the geometry and sound intensity at the microphone head.

3.2 Software - MicLocator

The software developed for this thesis is named MicLocator. MicLocator and all of its preliminary proof-of-concepts and tests are written in `python`, because of its ease of use and multitude of packages available. Packages used (not included in the standard installation) include `numpy`, `scipy`, `sounddevice`, `matplotlib`, `yaml`, `glob` and `datetime`. For development, the IDE Visual Studio Code and the newest version of `python` 3.8 were used .

A graphical user interface (GUI) was developed to ease software usage by providing a convenient way to enter and ensure correct parameters like speaker coordinates. The GUI uses preset YAML files for configuration, `PyQt` (`PySide2`) as GUI backend and can be seen in figure 6. In addition to a GUI version, a CLI (command line interface) version was also developed for automated usage.

The software is responsible for the synthesis of the samples (section 3.3), playing it on the correct speakers and doing all analysis steps including calculation of the cross-correlation of the played and recorded sample, peak-picking to calculate pseudo-ranges used by the multilateration algorithm and visualising the results in 3D interactively.

To test and validate the principal work procedure, a prototype script was developed and used to simulate a real recording before acquiring hardware.

Code snippets can be found in the appendix (section 6) and section 3.4.

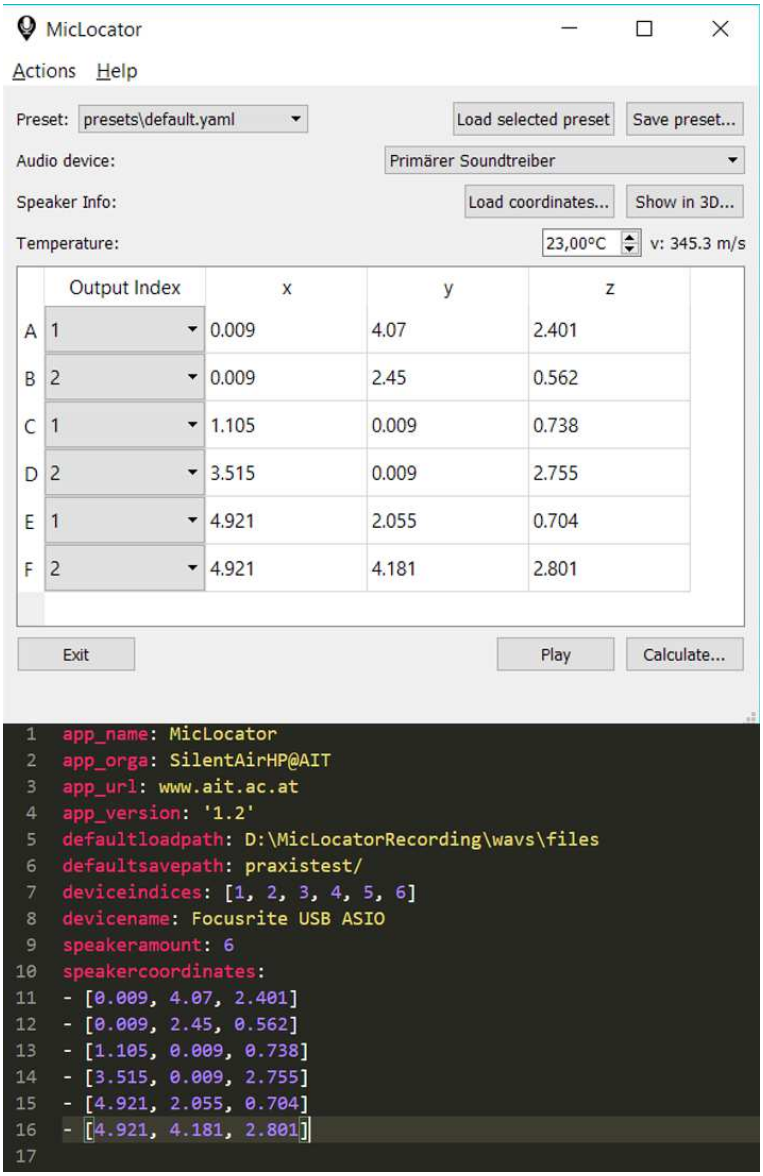

Figure 6: The stand-alone program for microphone localisation in our setup is called MicLocator. This is the version with a graphical user interface (GUI). (There is also a command line based (CLI) version.) The user has the option to load and save a preset with all parameters needed for calculation. The bottom part of the image shows one of these presets, written in the markdown language YAML. The speed of sound is temperature-dependent (section 2.7) and needs to be monitored before each calculation and maybe changed in the GUI. The Output Index column specifies on which output channel the sample shall play (The screenshot was made on a PC with only one stereo output, not the actual recording machine). The GUI MicLocator program is meant to be used manually, so the normal recording program reaper is used for recording data, then the Play button in the GUI is pressed to send out the sample. After stopping the recording again, one should press the button "Calculate..." and select the appropriate WAVE files. The localisation result is then shown in text form and can be visualized with the "Show in 3D..." button.

3.3 Sample design

To obtain pseudo-ranges for the multilateration algorithm, a specific audio signal has to be played from each of the six speakers. The sample played can either be unique for each speaker or the same sample can be time-shifted to ensure a direct correspondence (for the trade-off of a longer total signal duration and a maximal valid distance). For simplicity, the second approach was used, as it can be expanded to the first later on.

This section illustrates the process of finding a sample providing good results. The most simple waveform (W) is the sine wave of a frequency f

$$W_{sine}(t) = A\sin(2\pi \cdot t \cdot f), \tag{26}$$

Because of its perfect periodicity, the auto-correlation yields a constant time series and cannot be used for peak-picking. To embed fingerprints into the signal, one can change the signal's amplitude or the frequency. Continuous changes in the amplitude are difficult to distinguish from background noise in a changing environment and not applicable to our situation. Fast changes in the amplitude behave similarly to switching the signal on and off.

The simplest approach of frequency variation is a frequency sweep (also called chirp), which can be modeled by a linear increase of the instantaneous frequency $f(t)$ from a minimal to a maximal value according to:

$$f(t) = f_{min} + c \cdot t, \quad c = \frac{(f_{max} - f_{min}) \cdot t}{T}, \tag{27}$$

where T is the duration of the sample (in the same units as t).

The waveform should be continuous and the instantaneous phase is given by the time integral of the instantaneous frequency:

$$\phi(t) = 2\pi \int_0^t f(\tau)\, d\tau = 2\pi \cdot \int_0^t f_{min} + c \cdot \tau \; d\tau = 2\pi \cdot (f_{min} \cdot t + \frac{(f_{max} - f_{min})}{T} \frac{t^2}{2}), \tag{28}$$

resulting in the waveform of:

$$W_{sweep}(t) = A\sin(2\pi \cdot t \cdot (f_{min} + \frac{(f_{max} - f_{min}) \cdot t}{2T})). \tag{29}$$

The factor of 2 in the denominator is a common pitfall. Error sources like this can be excluded from the get-go by using library functions of the used programming language or its modules. These functions also often increase performance (sometimes massively). In `python` the submodule `scipy.signal` is commonly used for signal processing and has a dedicated function `chirp()`

```
waveform = scipy.signal.chirp(t: array, f_min, duration_in_s,
    f_max, method="linear")
```

Of course, from a theoretical standpoint, the auto-correlation of any non-repeating time series such as a frequency sweep should exhibit a perfect peak if quantization errors are neglected. Because of the non-ideal frequency response of the speakers and microphones used in the audio signal way and the fact that the cross-correlation of the signal sent out and recorded is dependent of the amplitude of both signals and the recording in a noisy environment, correct peak-picking can be a problem. Sliding filters to block out environmental noise or simply a louder signal emission could be employed to combat these problems, but an easier and more robust approach was found.

After also testing multiple sinus waveforms separated by periods of silence, a combination of fast frequency sweeps and durations of silence was found to be the most robust signal form for good results in the procedure of cross-correlation and peak-picking.

During testing of the naive approach of using random numbers to determine the duration of the sweep and silence parts and the minimal and maximal frequencies, it became apparent, that certain randomization sweeps exhibited sharper peaks, which led to the realization, that not random, but in fact non-repeatable numbers are suited best for sharp peaks in the cross-correlation. As described in section 2.9, non-repeatable numbers can be obtained by the multiplication-hashing function.

The final sample consists of a sequence of frequency sweeps and silence with a duration determined by the non-repeatable numbers in a range of 0 to 0.025 s. The decision, if a sweep or silence is next in the sequence is determined by a coin-flipping (50/50) randomization, which is fixed by choosing a specific `numpy`'s randomization seed. The parameters of the frequency sweep (minimal and maximal frequency) are again determined by another sequence of non-repeatable

numbers and the decision if a sweep is increasing or decreasing in frequency is again determined by a digital coin-toss. The multiplication hashing method is determined by its parameters' starting value. These, along the randomization seed, can be changed to generate unique channel samples. A spectrogram of the final sample can be found in figure 7. The code for this specific sample can be found in the appendix (section 6.2).

As channel sample duration 0.5 s was chosen with a time offset (shift) between each output channel of 0.1 s, resulting in a total sample duration of 1 s. The sample's multi-channel spectrogram can be found in figure 8. The channel time shift/offset limits the distance between the speaker and the microphone (while ensuring a correct speaker correspondence) to $0.1\ s \times 343\ m/s = 34.3\ m$, which greatly exceeds all requirements of our indoor setting. The channel time shift can be reduced to 0 s, if a unique sample is provided to each channel. In our case however, signal duration is not an issue, as we need to do the procedure only once. For continuous tracking of a microphone or other entities equipped with a microphone (human, phone,...) a higher polling rate is needed. In addition to uniquifying a signal to reduce the total sample duration, the sample duration itself can be reduced. A signal of this type is plotted as spectrogram in figure 9. This approach may increase multipath effects, but in continuous tracking applications a false data point may not be a severe issue as the next tracking data point comes very soon. To extract useful information from multiple data points, a plausibility test (maximal speed threshold) and/or a Kalman filter would need to be implemented.

In theory, it should be possible to shift all frequencies occurring in the sample to the supersonic regime, but our equipment is not able to work reliably for this case. Further testing needs to be done to determine if the source of error are the amplifiers or the speakers. The frequency response for the measurement microphones is also unknown above 22 kHz, but it is assumed to be not the primary source of error, as amplifiers and speakers are produced specifically for "normal" audio usage, where high frequency signal content is unwanted and therefore most likely filtered out.

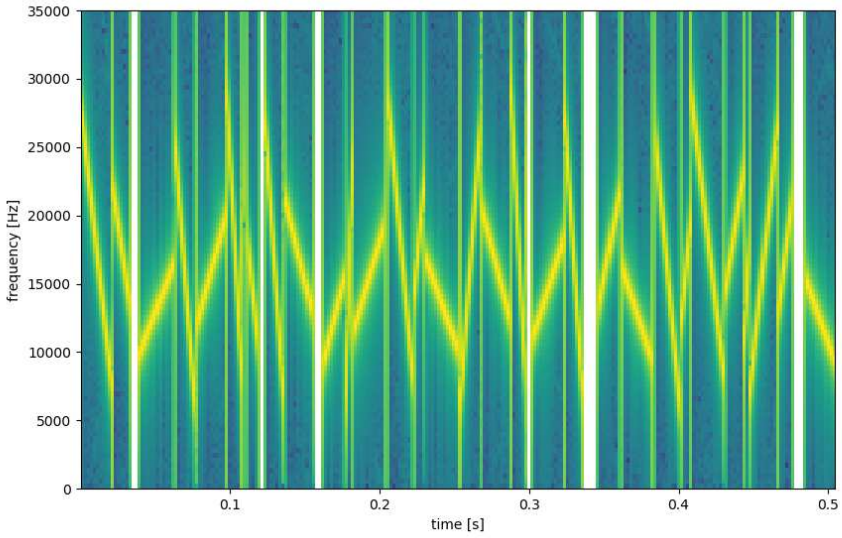

Figure 7: Spectrogram plot of the audio sample generated by the multiplication hashing method, as described in section 3.3. Sequences of frequency sweeps of different duration, minimal and maximal frequencies are separated by silence of varying duration. The total sample duration is just over 0.5 s and can be, on one hand, increased to encompass multipath effects of wrong peak-picking, or, on the other hand, reduced to increase polling frequency and decrease computing time. Frequencies could be shifted to the ultrasonic regime when using appropriate equipment.

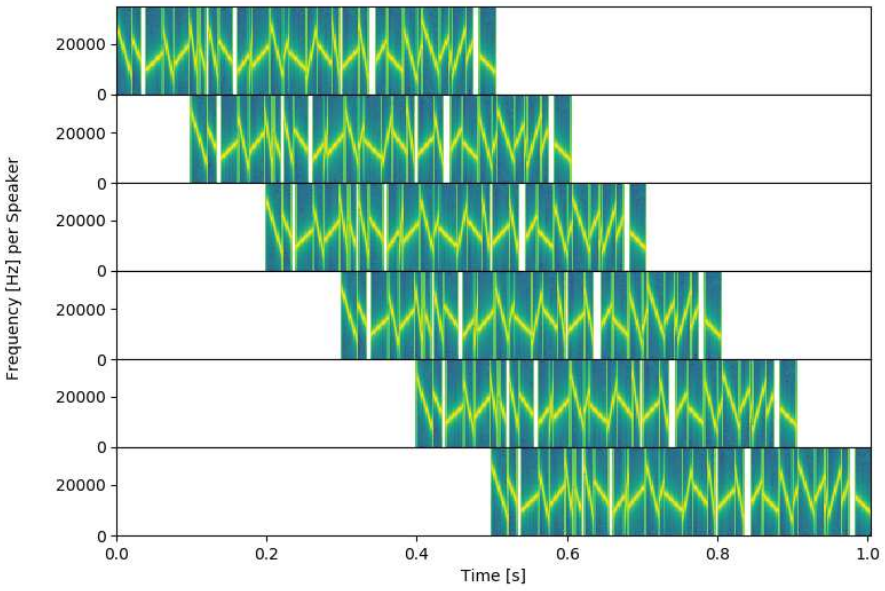

Figure 8:
Multi-Channel Spectrogram of all six speakers. White regions indicate zero signal (silence). The spectrogram for each speaker is the same (as in figure 7), but is time-shifted by 0.1 s. The time shift allows a correct correspondence to the speaker in the recorded signal.

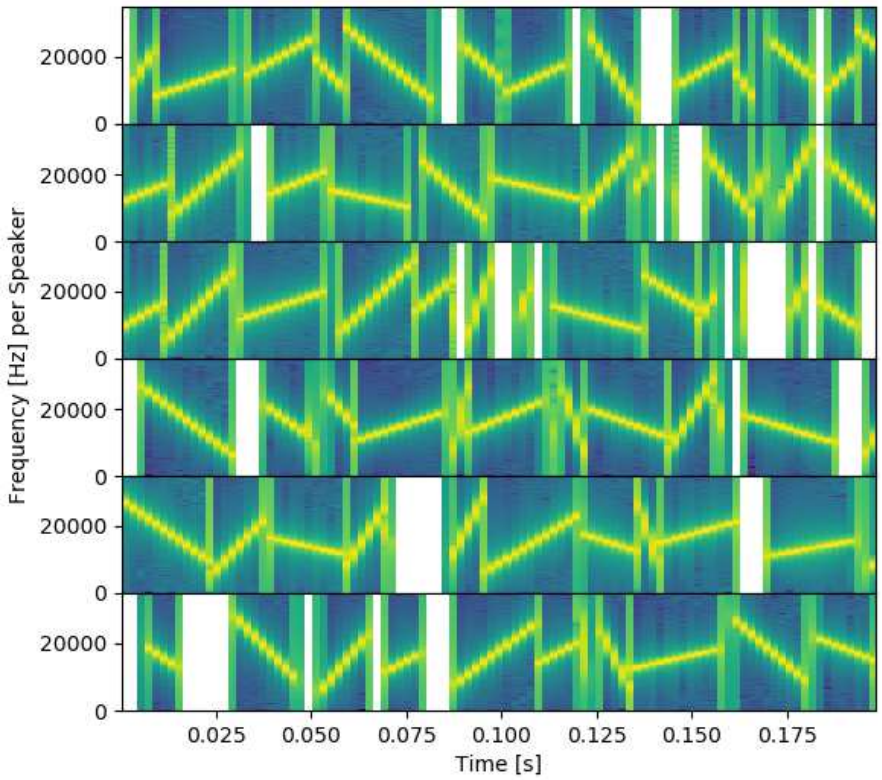

Figure 9:
Multi-Channel Spectrogram with unique signals for each of the six speakers and reduced (total) sample duration (0.2 s). This sample may be used for continuous location tracking, where not every polling result has to be correct necessarily, as it can be filtered out with plausibility tests or Kalman filters. This sample was not tested in the lab.

3.4 Cross-Correlation and Peak-Picking

3.4.1 Cross-Correlation

Different processing procedures for optimal and efficient computation of the cross-correlation of sent and received signal were considered and tested. The correlation function provided by `python`'s numerical package `numpy.correlate()` uses the formal definition (see Eq. 10) and is too slow for this use-case. A big improvement can be made by exploiting the fact, that the cross-correlation is mathematically similar to the convolution which can be calculated in the frequency domain by multiplication of the two Fourier-transformed time series. The cross-correlation can therefore be calculated by:

```
def rfft_corr(x1,x2):
    total_size = len(x1) + len(x2) - 1
    fft_size = 2 ** int(np.ceil(np.log2(total_size)))
    fourier_x1 = np.fft.rfft(x1, fft_size)
    fourier_x2 = np.fft.rfft(x2, fft_size)
    xcorr = np.fft.irfft(fourier_x1 * np.conj(fourier_x2))
```

Listing 1: Code of efficient cross-correlation in `python`

where `fft_size` is a power of two to allow efficient calculation and `rfft()` is the fast Fourier transform for real-valued time-series and `irfft()` the inverse. `np.conj()` returns the complex conjugate.

As it is known that the signal consists only of frequencies in a range we have chosen before, we can apply digital low and high pass filters to isolate this frequency range from unwanted noise picked up during recording. This can be done easily by trimming the Fourier series in the above code example:

```
    freqs = np.fft.rfftfreq(N, d=1./samplerate)
    fourier_x1[freqs < f_low] = 0
    fourier_x1[freqs > f_high] = 0
    fourier_x2[freqs < f_low] = 0
    fourier_x2[freqs > f_high] = 0
    xcorr = np.fft.irfft(fourier_x1 * np.conj(fourier_x2))
```

The Fourier transformation was applied using the Fast Fourier Transform implemented in `python`'s `numpy.fft.rfft()` which itself uses `FFTPACK`'s implementation in `FORTRAN` and `C`.

3.4.2 Peak-picking

Peak-Picking is the process of selecting an extremum in a data sequence and is usually easy for humans, but needs to be automated and therefore formulated in an algorithmic way. The simplest approach would consist of applying a certain threshold, scanning for regions of points where the values are above the threshold and calculating each regions mean. This method is error-prone, as sometimes values that are too high or too low with regards to the threshold are characterized in a wrong way leading to false-positives and false-negative results. A threshold set too high will yield no data requiring readjustment, whereasa threshold set too low will pick up noise or smear out peaks and therefore introduce worse accuracy.

A better approach (for our specific cross-correlation) is the following: The maximal value of the absolute of the cross-correlation is determined, saved and a close region around it is masked (by setting all values of this region to zero). This process is repeated until a peak for each speaker signal is identified. These peaks are then identified with its corresponding speaker (by sorting) and converted to time differences with regards to the peak from the first playing speaker and used for the multilateration algorithm.

```
1 import numpy as np
2 def pick_peaks(correlation_envelope, mask_window_size,
     rec_samplerate, sample_bufferlength, speakeramount)
3     """returns the times [t_1-t_0, t_2-t_0, ..., t_n-t_0] as
         list (in seconds), where n is the speakeramount-1
4     Function parameters:
5     correlation_envelope: the data from cross-correlation (may
         be pre-processed)
6     mask_window_size: size of masking window in samples on one
         of the symmetrical sides; 2500 samples is a good size at
         192 kHz
7     rec_samplerate: samplerate of the recorded data; normally
         either 48000, 96000 or 192000
8     sample_bufferlength: time difference of starttimes for
         different speakers in the multi-channel sample; in
         seconds (would probably be 0 if using unique speaker
         samples); normally 0.1s
9     speakeramount: amount of speakers used, normally 6"""
```

```
    indices_of_maxima = list()
    for speaker in range(speakeramount):
        maximumindex = np.argmax(np.abs(correlation_envelope))
        correlation_envelope[maximumindex -
            mask_window_size:maximumindex + mask_window_size] = 0
        indices_of_maxima.append(maximumindex)
    indices_of_maxima = sorted(indices_of_maxima)

    times_of_maxima = np.array(
        indices_of_maxima, dtype=np.float64)/rec_samplerate

    #times are now in absolute form, but we needed the relative
       times in regards to the first speaker time
    start_time_offset = times_of_maxima[0]
    times = list()
    for timeindex in range(1, speakeramount): #from 1 to
       speakeramount-1
        starttime_of_sample = sample_bufferlength*timeindex
        newtime = times_of_maxima[timeindex] -
            start_time_offset - starttime_of_sample
        times.append(newtime)

    return times
```

Listing 2: Peak-picking code in `python` (compressed)

A typical cross-correlation with all 6 channel signals, picked peaks and masking can be found in figure 10.

Peak-picking can be challenging if the cross-correlation has no perfect peak. This can be traced back to multipath effects or other unwanted influences. A human may be able to understand these effects, its implications on the cross-correlation and can often select the correct pick given the plot (as in figure 11). However, it is difficult to capture this understanding in an algorithmic way to be used by a program. Machine learning concepts could maybe be employed, but only signal processing tools were investigated. Sadly, no significant improvement could be made compared to the procedure outlined above. Multiple methods were considered and tested, including:

- A cubic interpolation of the cross-correlation alongside the application of

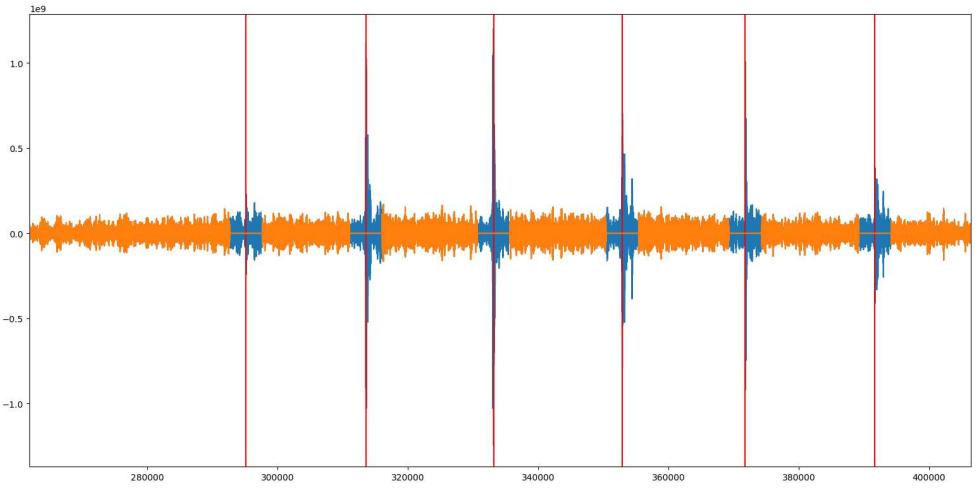

Figure 10: A typical cross correlation of the played and recorded signal. The absolute magnitude is insignificant. Picked peaks are highlighted with a red vertical line. The procedure for peak-picking is described in section 3.4.2. The procedure includes masking of the close region around the detected peak to expose the next highest local maximum as global maximum. This region is colored blue. The spacing of peaks appears very regular, because the distance from the microphone to each speaker is similar (i.e. the microphone is placed in the middle). In this setting, the offset (buffer) between the starting time of each speaker's sample could be reduced dramatically, which would reduce total sample duration, but may increase problems for peak-picking.

a Savitzky-Golay (savgol) filter [31]

- using Hilbert transform (used in signal processing) as smooth envelope
- Resampling of the cross-correlation
- Applying a moving average to the cross-correlation
- Convolution of the cross-correlation with a step-function (100 elements of -1 followed by 100 elements of $+1$)
- Additional filters acting upon the cross-correlation
- Using Finite differences (discrete derivatives) of the cross-correlation
- Calculation of the cross-correlation after applying a mask in the time domain of the original audio data

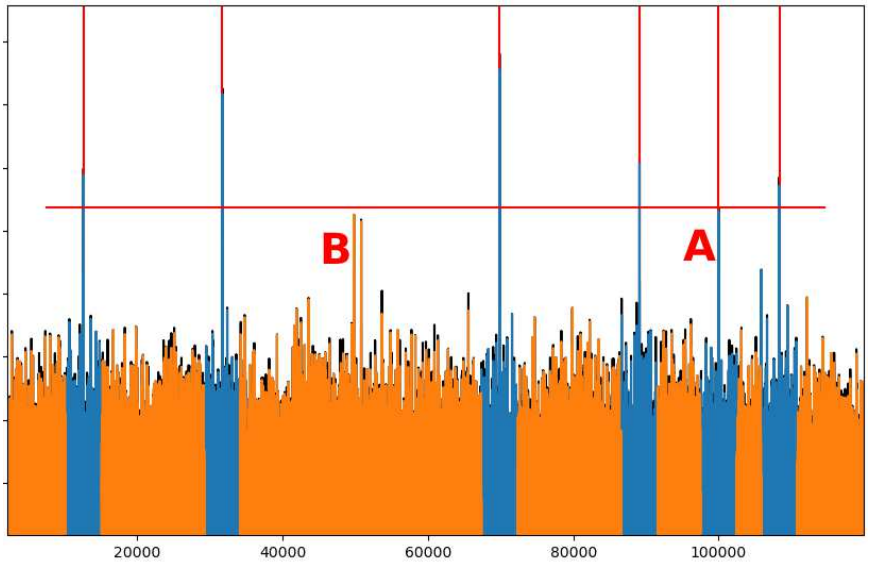

Figure 11: This plot showcases the absolute of a cross correlation with wrong peak-picking. It was generated while testing the limits of correctness (see section 4.1) with a simulated signal with added noise and artificially introduced clipping. Compared to figure 10 it immediately becomes obvious, that the periodicity of peaks is broken. The peak at A is (for a human) obviously wrongly interpreted - a false positive. The peak picking algorithm (outlined in section 3.4.2) only searches for the n largest peaks, where n is the amount of speakers (6 in our case). This means, that a false positive leads to a false negative at B. The red horizontal line shows that the peak at A is higher than B and therefore selected (first).
The peaks at B showcase another problem: two peaks of similar height close to each other. This problem may lead to a wrong peak-picking if a peak generated by noise (as in A) is higher than the true peak. Noise typically does not lead to this behaviour, but floor reflections or other multipath effects may do so (section 3.5).

3.5 Multipath effects

Multipath effects are a common problem in multilateration applications. They occur in presence phenomena such as reflection, diffraction, scattering and refraction. In settings where the receivers or reference nodes are stationary, multipath effects pose an even bigger problem than in non-stationary settings, because the signal pathway will not change and erroneous signals can not be omitted without total loss of information. Non-stationary settings can also use information from multiple measurements to calculate probable trajectories (using advanced algorithms and/or Kalman filters [32]).

Examples of multipath effects include, in GNSS applications, the reflection from walls and houses (particularly in urban environments), diffraction during global data transmissions off the ionosphere or, as is the case in our (in-door) setting, refraction if the direct line of sight is obstructed (e.g. by machines fixtures or other sensor equipment).

Figure 12 shows a cross-correlation, where floor reflections caused a wrong peak-picking. Floor reflection can be mitigated by re-positioning speakers or dampening the floor itself. Other multipath effects could be reduced by optimizing the sample design or improving the peak-picking algorithm.

The problem of line of sight obstructions can be mitigated by modelling the actual pathway or applying other complex procedures [33], but is not within the scope of this thesis.

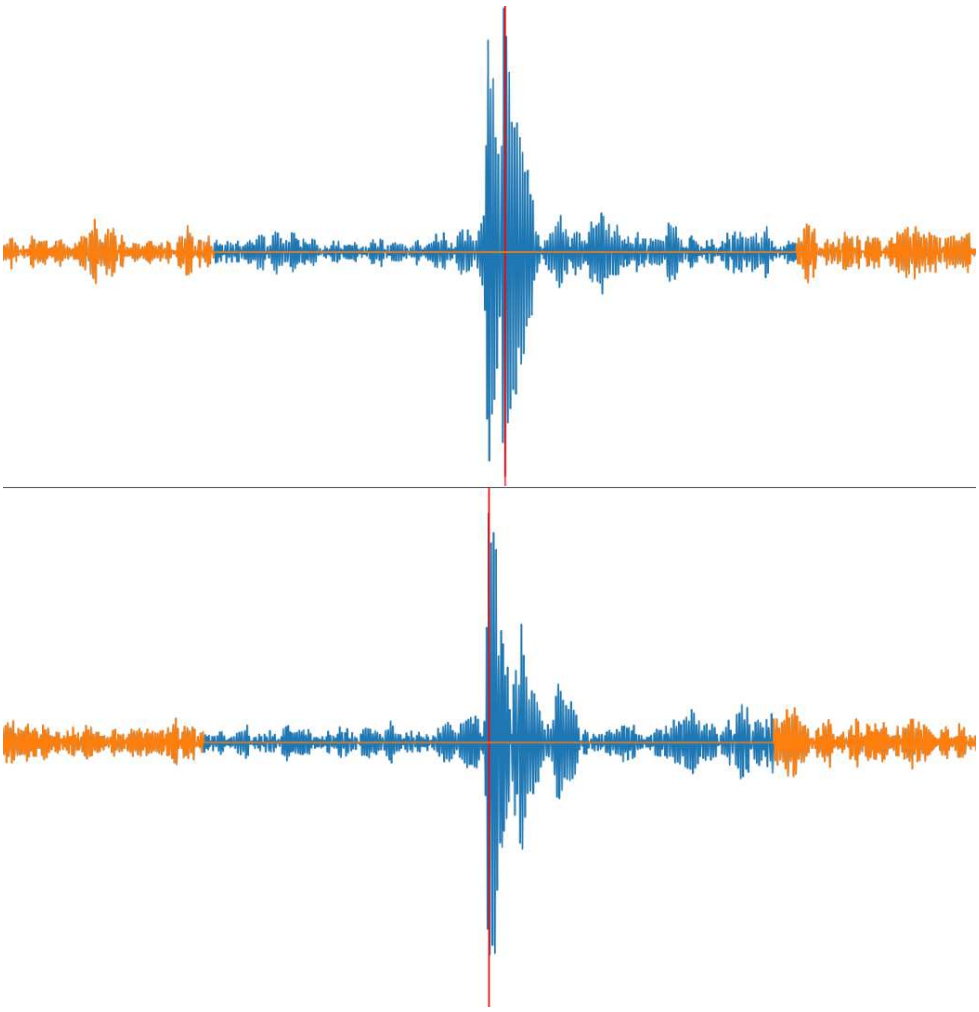

Figure 12: This figure showcases the problem of multipath effects. The bottom plot shows a cross-correlation (zoomed in compared to figure 10) with correct peak-picking, while the top plot exhibits wrong peak-picking caused by floor reflections. In the top plot, two peaks exist and the second is selected by the algorithm (detailed in section 3.4.2), because it has a higher value compared to the first. The first peak would be the correct one as the signal reaches the microphone sooner, because of a shorter signal pathway (no floor reflection). A human can easily identify the first peak as the correct one, but this behavior is difficult to formulate in an algorithmic way for the computer. To mitigate the effect of floor reflections, a couple of measures can be undertaken: further refinement of the peak-picking algorithm, improvement of the speaker placement or dampening the floor itself.

3.6 Automated data logging

3.6.1 AirLogESP - Automated air data logging

Most of our experiments with heat pumps take place inside climate chambers with a controllable air environment. The air's temperature and humidity around the heat pump is therefore known, either from its predefined control value or the multiple temperature sensors of the climate chamber's thermodynamics data recording. For measurements outside the climate chambers, as is the case for already installed heat pumps or other big and loud equipment, such as industrial fans and coolers, a mobile data acquisition is needed. To accurately calculate the sound power level, the air's temperature, and air density or, by easier measurement, the absolute air pressure needs to be known. The air pressure is not measured by the climate chamber's infrastructure and could be inferred from weather data publication and knowledge of the absolute height above sea level of the experiment station, but can be measured easily with IoT (Internet of Things) components.

In order to be able to measure the absolute air pressure and have an independent, mobile source of data of the air's properties, a microchip capable of WiFi connectivity with a connected temperature sensor was introduced to the experimental setup. As hardware the ESP8266 microchip and the BME 820 temperature sensor were chosen, which are readily and cheaply available as they are commonly used in IoT applications. The BME 820 supports measurements of the air's temperature, absolute pressure and relative humidity (as percentage). It is not possible to calibrate the sensor to the same degree of accuracy as for example a PT-100, but a high accuracy is not needed for our application. The chip's WiFi capability allows easy remote readout of the data and data handling on the same machine the audio recording software is running. The experimental setup already included a WiFi router for remote audio recording control, e.g. while doing a calibration procedure, as the climate chamber's door needs to be closed for sound-proofing [3]. The microchip provides values via a simple HTTP server in the JSON format, which are updated each time a website serving request is received.

For protection and mounting capability an enclosure of the microchip and

the temperature sensor was designed and 3D printed. This device was then named AirLogESP.

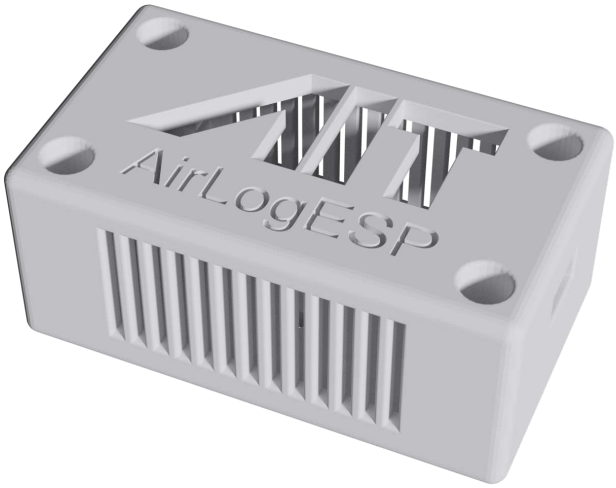

Figure 13: A rendering of the case for the so-called AirLogESP, a microchip (ESP8266) connected to a sensor, reading out data of the air's temperature, pressure and humidity and providing it on a simple website. Data logging happens on the main machine in the same WiFi network. Information about the air's properties is needed for sound power level calculation. The case was 3D printed and the microchip and sensor mounted inside it.

The microchip can be programmed with the Arduino IDE from any PC over USB. The program's counterpart on the recording PC consisted of a python script connecting to the microchip running a simple HTTP server and logging its data values in a format convenient for later acoustic processing in a user-defined frequency of 10 seconds by default.

3.6.2 SoundDAQ - Automated acoustic data logging

Experiments typically consist of analyzing research heat pumps from projects at the research division or from commercial companies, which ship prototypes in for thorough testing in a well equipped environmental chamber or for certification. In the past, these procedures analyzed only thermodynamic properties of the heat pump, but with the project *SilentAirHP* ([9], [3]) acoustic measurements were made possible and gained interest from manufactures. Acoustic analysis

is also incorporated into in-house projects, e.g. the *GreenHP* project, and is a high priority task in the *DigitalTwin* and *IoT-HP* projects, which implement an advanced modular analysis framework in the spirit of the Industry 4.0 revolution. For a detailed digital reconstruction (a digital twin) of the thermodynamic processes, data of temperature, pressure, humidity and more sensors need to be captured and saved in a dedicated data base. Acoustic measurements should be incorporated directly alongside the thermodynamic data into the same data acquisition system to allow efficient data analysis. The immense data amount of acoustic analysis has to be reduced and therefore values are pre-processed live and only sound pressure and power levels are inserted into the shared framework. For later reference and detailed analysis, waveforms are stored in a separate system.

The framework and database used, is made by the Austrian company B&R Industrial Automation and is compliant to the industrial standard OPC-UA (Open Platform Communications - Unified Architecture). This standard offers many benefits for industrial settings, where control parameters and sensor values need to be monitored reliably over extended periods of time. The standard's platform-independence allows development of small proof-of-concept scripts up to large software projects. The approach presented here used the programming language `python` due to its maintainability and ease of development.

Another goal of the shared database is to easily enable concurrent analysis of thermodynamic and acoustic data and quantification of sound emission for different heat pump operation modes for people working in the lab mainly trained on only the thermodynamic parts of heat pump operation and without a broad background in acoustics. Nevertheless, the user needs a basic understanding of how to correctly handle acoustic data. This involves knowledge about logarithmic addition and averaging of multiple values in dB. It is only valid to add and average RMS values of sound pressure in the correct units, such as Pa. The conversion equations can be seen in section 2.10. In the optimal scenario a future version of the database and analysis framework supports built-in logarithmic addition, but potential data export by the user and analysis in other tools (e.g. Excel or self-written scripts), which do not operate in a correct way for acoustic quantities, has to be accounted for. The user also needs to input the represen-

tative areas needed for sound power calculation (see section 2.10). This can be automated, if measurements are taken inside climate chambers equipped with the microphone localisation system presented in the main part of this thesis, but can also be done manually by measuring the 3 distances between opposite microphones for a box-like heat pump (see figure 14) and calculating the surface areas for each side of the box-shaped hull area around the heat pump. In the optimal case, the framework supports common data analysis operations, like calculation of averages and correlations, resampling and plotting features.

As most experiments of the heat pump's thermodynamic properties take multiple hours to a couple of days, the ongoing and continuing data logging produces vast amounts of data that need to be stored long-time for future reference and newly arising research questions. The minimum data retention duration is projected to be 10 years. To not flood the database with acoustic data, but still maintaining good data quality, it was decided to use five microphones (four in the center of each side of the (commonly box-shaped) heat pump and one on top, see figure 14). For measurements, the microphones are placed 50 cm away from each face of the idealized box around the heat pump.

For efficiency in data analysis and data storage, acoustic data is not saved as waveforms (time domain), but spectrograms (frequency over time), where a reduced resolution does not lead to a loss of significant information. Compared to waveforms, the time resolution can be brought down from typical audio sampling rates of 48 kHz or 96 kHz down to data points every 1 second. Data depth from 24 bits in the pressure domain down to one or two significant decimals figures in the dB domain. The frequency dimension is not continuous, but binned. For time to frequency conversion an FFT (Fast Fourier Transformation) implementation was chosen. Frequencies coming directly from the FFT algorithm are linearly spaced, while acoustic data analysis is typically down in a logarithmic frequency domain in accordance with the Weber-Fechner law of human perception. Data is therefore binned logarithmically into frequencies from the 1/3-octave band (according to ISO 266) from 31.5 Hz to 25 kHz resulting in 30 frequency bins.

Of course each microphone needs to be treated independently. It was decided to implement another series of database entries with the logarithmic sum of all

microphones and all sound power levels (for each weighting) and sound pressure levels for each frequency. This was done for convenience of the user, having one microphone representing the total sound emission.

The human ear does not perceive the same pressure difference (sound pressure level) of different frequencies as the same loudness. Low frequencies need a much higher sound pressure to be perceived as loud as high frequencies. (This is also the reason why most of the power in sound systems is needed by the sub-woofers and low-end speakers.) To capture this behavior mathematically, psychoacoustical sound level weightings were introduced. Commonly used are the A and C-weightings. The A-weighting was introduced for medium sound pressure levels, whereas the C-weighting is supposed to be used for loud settings. The zero weighting (no weighting) is abbreviated with Z. Sound pressure level and sound power level values weighted by the A-weighting use $db(A)$ as unit sign.

With modern computers, a perfect weighting is achievable with both pressure level dependence and following the measured equal-loudness contour. Such a weighting is described in ISO 226. Sadly, the A-weighting is still predominantly used. Reasons include better comparisons to old data (only available in A) and usage of small handheld decibel meters. Because of its widespread use in prior heat pump acoustic measurements, our data is stored in the A-weighting. The A, C and Z-weightings used by us are linearly related to each other in the dB domain and values in other weightings can easily be recovered by adding or the subtracting the appropriate weighting values on-the-fly in the subsequent processing tools without saving it to a drive.

In addition to the sound pressure level values, each microphone channel saves a time series of sound power level values for the weightings A, C and Z. As discussed in section 2.10, the sound power level can be derived from the sound pressure level with information about the surrounding air environment and representative areas. It is valid to weight the sound power levels before this calculation. Therefore, sound power levels can also be weighted.

A compromise needs to be made between the frequency of these data points for reduced storage space and to still being able to accurately capture momentary sound phenomena, such as openings of an expansion valve or external sound

sources. These phenomena can also be analyzed in detail with the wave files stored alongside the database. The frequency of data readout can be readjusted, but was decided to be one sample per second by default. The time series of sound pressure levels therefore represents an effective low-resolution spectrogram with a sampling of 1 s and in the 1/3-octave band frequencies of each microphone. Information about the sound power level reconstructed with the microphones' representative areas is also included. The data acquisition system is expandable for new pre-processing approaches, which may include more inputs, inputs from intensity probes, or more meta data.

The total number of data channels supplied to the database are therefore: 30 frequency binned A-weighted sound pressure levels, 3 logarithmic sums of the sound pressure levels over all frequencies (A, C and Z-weighted) and 3 logarithmic sums of the sound power levels over all frequencies (A, C and Z-weighted). These 36 channels are saved for each of the 5 physical microphones plus one virtual microphone, which is the logarithmic sum of all other microphone channels (weighted evenly).

Therefore, in total there are $36 \times 6 = 216$ data channels to be stored in the database each second with a value in dB with one significant decimal point.

(The author is aware, that some information stored in the database could be calculated from other data entries, but decided to write those time series into the database for ease of analysis and lack of feature support of the current framework version. If those features are added in the future, those entries could be dropped in future measurements.)

Additionally, meta data about the configuration and analysis script version is transmitted at the start of recording. The configuration data includes the most important parameters for analysis, such as the FFT block size, recording sampling rate, microphone input channels, calibration factors (for bitstream to pressure conversion), and the values of representative areas used.

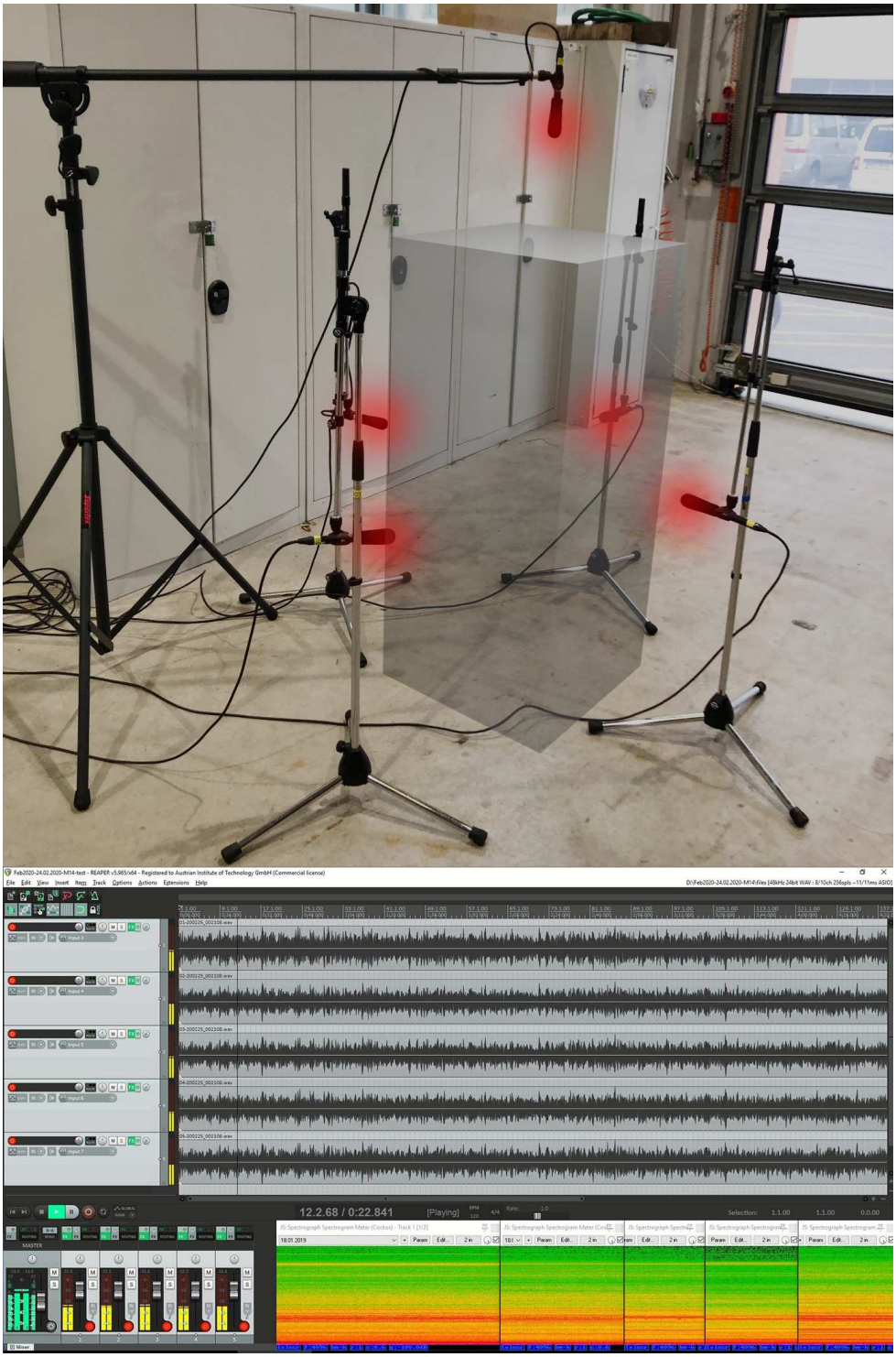

Figure 14: Photograph of the 5 channel setup used for data in the SoundDAQ database (Section 3.6.2). Imprinted is a rendering of a box to illustrate the microphone placements around a box-like heat pump. Microphone heads are marked in red. Included is also a screenshot of Reaper, the audio recording program to visualize a typical waveform of these 5 microphones and a scrolling, live spectrogram view for each microphone channel. The developed SoundDAQ software is command-line-based and currently has no spectrogram or waveform GUI.

4 Simulation & Measurements

4.1 Simulation

Prior to this thesis, and before acquiring any of the additional hardware needed, a number of tests were carried out which served as proof-of-concept. These fully digital tests showed promising results, but were only done in an idealized environment with one sweep as sample and without noise. The occurrence of performance-degrading multipath effects were not anticipated at this point.

These tests and simulations were repeated and improved to test limiting factors and other potential environments and configurations with results presented in this section.

The simulations were done in a completely digital environment using an audio signal of a sequence of sweeps parameterized by non-repeatable numbers as described in the first section of 3.3. Signal transmission was simulated by delaying the onset of each signal channel according to the distance from a virtually placed microphone to the appropriate speaker. These signal channels are then added together.

Six speakers were placed at six edge points of a normalized cube. Figure 16 shows a 3D plot of this speaker configuration (with speakers A through F colored in light gray).

At a sampling rate of 192 kHz and without any artificial error sources added, the algorithm outputs the correct target coordinates within an error of about 0.3 mm for each coordinate. These small errors arise from quantification errors of translating the real-valued distance to discrete signal time delays. At this sampling rate, 1 sample corresponds to 1.8 mm (at 20°C).

The analysis remains stable with small added errors. An artificial enlargement of the distance from each speaker to the microphone in question was investigated. This type of error may be encountered in real measurements, when dealing with multipath effects, especially diffraction. An elongation of a random value between 0 and Δ was added to each of these distances. For microphone A in figure 15 (placed at $(0.32, 0.47, 0.64)$ m) and $\Delta <= 4$ cm the algorithm can still locate the microphone within an accuracy of 1 cm for each coordinate. For microphone B (placed at $(0.12, 0.18, 0.08)$ m), the algorithm loses the accuracy

of 1 cm for $\Delta > 2.5\ cm$. However, in this idealized scenario, the algorithm still converges for both microphones for $\Delta < 2\ m$ (sic!). Obviously these results are not usable as they do not have acceptable accuracy. Results for different Δ can be seen in figure 15. It is noteworthy that although random values are employed as error source, all results are consistent across multiple simulations.

Additional tests were performed with random (white) noise added to the audio sample. This procedure does not impact the performance significantly, even if the noise is many times "louder" than the signal itself, as long as it does not lead to significant clipping. White noise contributes equally across all frequencies and thus may (if it contributes at all) only generate completely wrong peaks in the cross-correlation, leading to either good results or a divergent multilateration algorithm.

Another test was performed to assess the capability for localisation outside of the convex hull spanned by the speakers (reference nodes). Figure 16 shows the results. While results are obtained, they are not accurate enough and thus not usable.

One additional simulation was performed to assess the need for a non-planar speaker configuration. Speakers placed in a circular pattern do not yield consistent results, because the multilateration algorithm diverges, even for signals without any artifical error.

In all simulations outlined above, the multilateration algorithm (explanation in 2.5 and its implementation in section 6.1) either converges or diverges in < 20 iterations.

Without a fully fledged physical sound wave propagation and in idealized scenarios, these simulations cannot capture the multitude of multipath effects, which are the primary source for performance degradation.

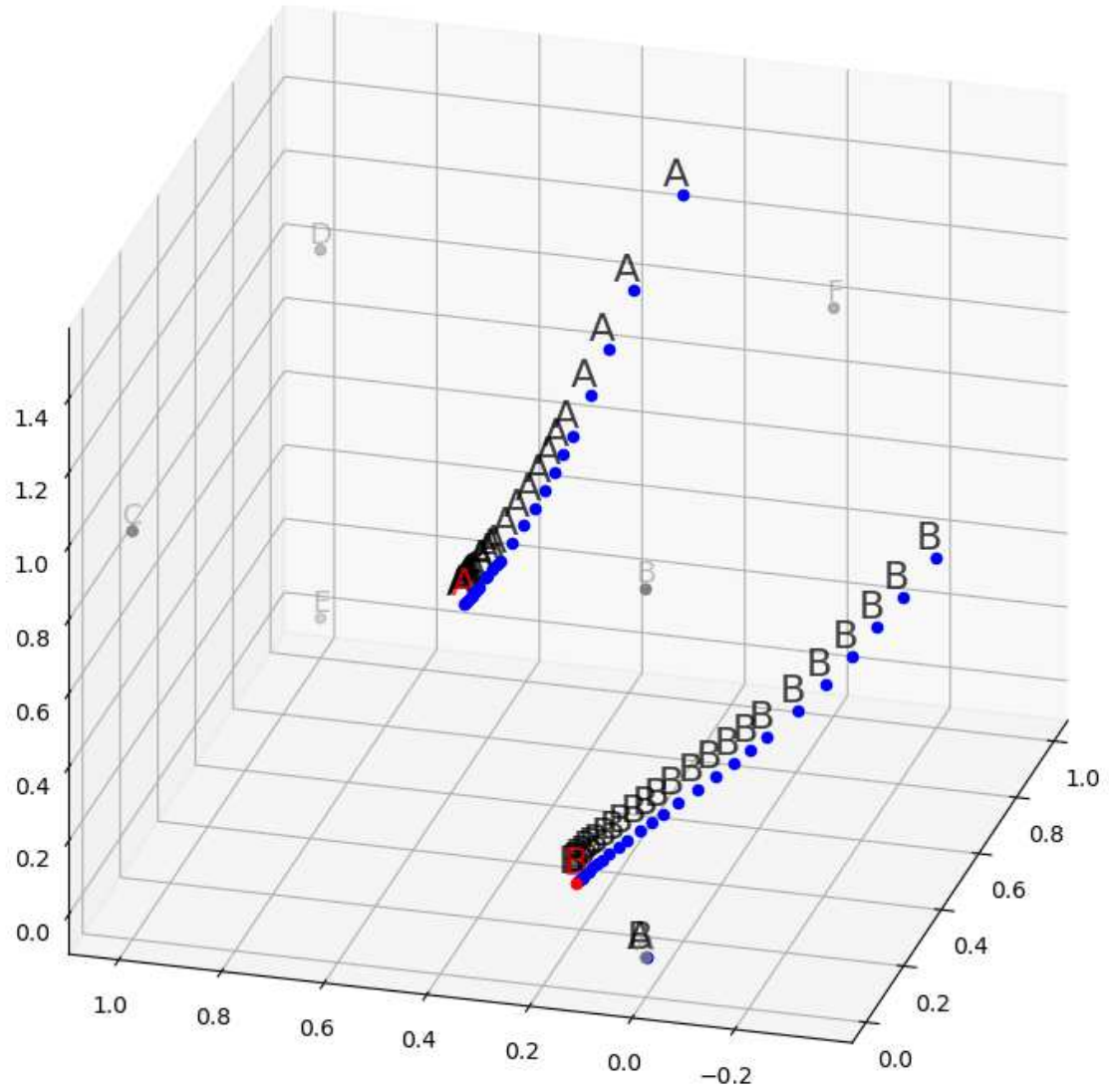

Figure 15: A 3D plot of a simulation with speakers in a cube pattern. Speaker A is located at $(0, 0, 0)$. The other speakers B through F are visible and colored in light gray. Two simulated microphones are placed (A at $(0.32, 0.47, 0.64)$ m and B at $(0.12, 0.18, 0.08)$ m) and their true coordinates colored in red. Tests were performed to investigate how an elongation of the distances to each speaker impacts performance. These elongations were determined randomly in a range of 0 to Δ, where Δ was varied. For $\Delta < 0.025$ $m = 2.5$ cm, the analysis has an accuracy within 1 cm. For increasing Δ the localisation result is drifting away, until the multilateration algorithm finally does not converge anymore. Divergent algorithm results are set to $(0, 0, 0)$. The lowest Δ leading to a divergent algorithm is 2.5 m and 2.25 m, respectively. The spacing of the drifting coordinates was determined by a manually defined set of Δ values.
Of course, these results are only valid for this specific geometry. In real measurements such a defined and 'well-behaving' error source is not to be expected.

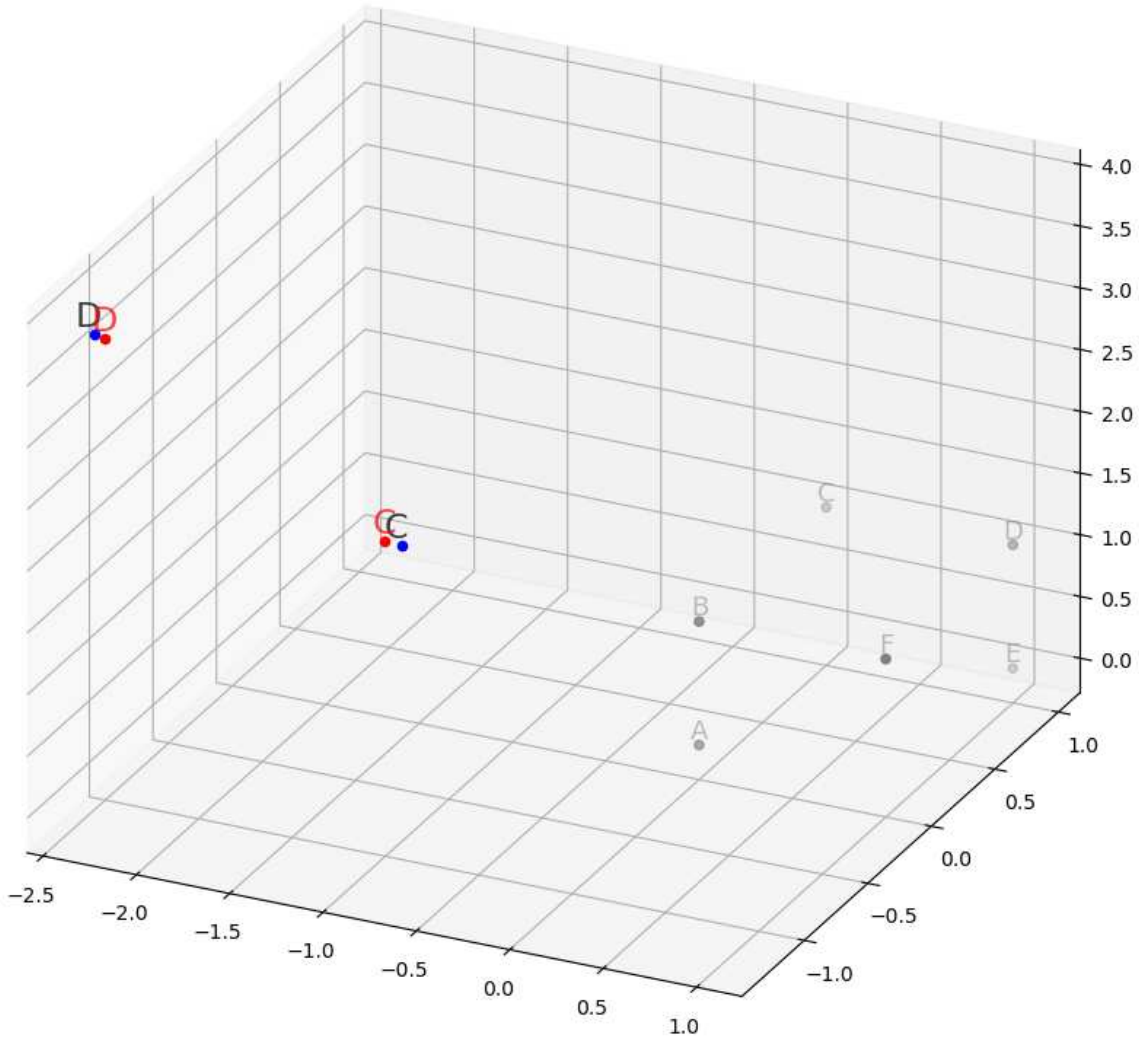

Figure 16: 3D plot of a simulation with speakers placed in the same cube pattern as in figure 15 to investigate the performance of microphones placed outside of the convex hull spanned by the speakers. Microphone C was placed at $(-1.89, 0.30, 0.80)$ and microphone D at $(-2.30, -1.30, 3.80)$ (as seen in red). The algorithm returned $(-1.797, 0.300, 0.796)$ for microphone C and $(-2.341, -1.321, 3.843)$ for microphone D (as seen in blue and black text). The deviation of the obtained and target values is not acceptable and would only worsen in real environments. Microphones should therefore be placed inside the speaker's convex hull.

4.2 Measurements

4.2.1 Setup

For this setup, the 6 speakers and the 3 amplifiers were used in conjunction with the 'smaller' audio recording setup as detailed in 3.1.3. The sampling rate was set to 192 kHz. The signal sample can be played by clicking the 'Play' button in the GUI (Figure 6) of the dedicated program ('MicLocator') during the recording process. For correct routing, the correct audio interface output channels need to be selected from the drop-down menu in the main table. The sample can also be generated and saved as a `.wav` file to be loaded into a DAW (Digital Audio Workstation) that records the microphones' signal. Reaper is used as a DAW. If the audio interface output channels are not configured correctly, the algorithm will not yield good results; e.g. if only 2 speakers emit audio (default stereo output setting), the results will lie on an one-dimensional line and not be spread in 3D space.

To extend the explanation of speaker placement in section 3.1.4, the following guidelines should be applied: The speakers should not be placed too close to the floor, ceiling or perpendicular walls to prevent floor reflections causing wrong peak-picking. Additionally, to combat these effects, damping materials can be laid out, if possible. To ensure a good and loud signal, the speakers should not be placed too far away from the microphones. Furthermore, they should face each other as good as possible. For optimal algorithm performance, the x, y and z coordinate axes should be treated equally by placing the speakers in a way as to maximize the extent of the coordinate span of each axis. In other words, coplanar speaker placement is to be avoided. The amplifiers should be set to a gain that does not distort the speakers by clipping, but as high as possible for good signal pick-up.

For the measurements presented in this chapter, microphone stands were used on which the speakers were mounted. To do this in a robust way, an adapter was designed in Autodesk Fusion 360 and 3D printed for each speaker. Carpets were laid out to prevent the occurrence of floor reflections. The origin of the coordinate system for these measurements was placed in a ceiling corner of the room to be independent of floor height variations (e.g. caused by carpets).

As a consequence, the z-axis was orientated down. The orientation of the x and y axes follow the right-hand rule.

The coordinate system's origin and orientation can be seen in figure 17.

4.2.2 Results

The localisation of 8 different microphone positions was measured and compared to reference coordinates. These reference coordinates were measured manually with a laser distance meter and tape measure by projecting each speaker and microphone position to the ceiling, measuring the distance, and then projecting this point again onto a wall in a right angle. Then, the distance from this point to another wall is measured and thus all 3 coordinates are obtained.

The localisation results are given in table 1 and visualized in figure 18. Microphone 1, 2 and 3 are also visualized in figure 17.

Tests with microphones placed outside the convex hull spanned by the speakers did not yield any usable results, because the speakers have high directivity and are oriented towards the middle, leading to weak signal pick-up and wrong peak-picking.

Variations of the audio sample with different parameters (multiplication hashing method algorithm, and duration) were tested, but did not yield better or worse results than for the sample used to obtain the results presented in this chapter.

mic	reference ±0.01 m			localisation			difference			
	x	y	z	x	y	z	x	y	z	$norm$
1	1.650	1.629	2.243	1.615	1.644	2.254	-0.035	0.015	0.011	0.040
2	1.804	1.334	1.512	1.766	1.293	1.479	-0.038	-0.041	-0.033	0.065
3	2.153	1.638	1.160	2.133	1.630	1.174	-0.020	-0.008	0.014	0.026
4	1.905	2.230	1.858	1.879	2.252	1.888	-0.026	0.022	0.030	0.045
5	2.563	1.929	1.256	2.539	1.984	1.300	-0.024	0.055	0.044	0.074
6	1.307	2.378	1.200	1.341	2.372	1.163	0.034	-0.006	-0.037	0.051
7	1.751	1.908	2.281	1.738	1.921	2.308	-0.013	0.013	0.027	0.033
8	2.129	2.529	1.119	2.154	2.574	1.169	0.025	0.045	0.050	0.072

Table 1: Table of results, given in m. The reference and localisation coordinates are visualized in figure 18. The reference coordinates were obtained by manual measurement with a laser distance meter. The difference columns contain the difference of the coordinates obtained from the localisation system and the reference coordinates. The norm column presents the norm of each difference vector. This error measure is dependent on the geometry of the speaker arrangement and the location of the microphone. It is important to note that the speaker coordinates used for the multilateration algorithm have the same uncertainty as the reference microphone coordinates. Section 4.2.4 shows that this uncertainty severely impacts localisation performance. Nevertheless, localisation can be performed under these conditions with an accuracy of a few centimeters at a typical scale of 1 m.

Figure 17: A photo of the experimental environment used for testing the localisation techniques. Carpets were laid out to prevent floor reflections. The coordinate system's origin and orientation can be seen in the top right room corner. Speakers were mounted on microphone stands with a self-designed 3D printed adapter. Their positions are emphasized with gray circles and labelled A through F. Three microphones were placed and their position manually determined with a laser distance meter. These reference positions are visualized in red and labeled 1, 2 and 3. The microphones' coordinates can be found in table 1 (also labelled as microphone 1, 2 and 3).

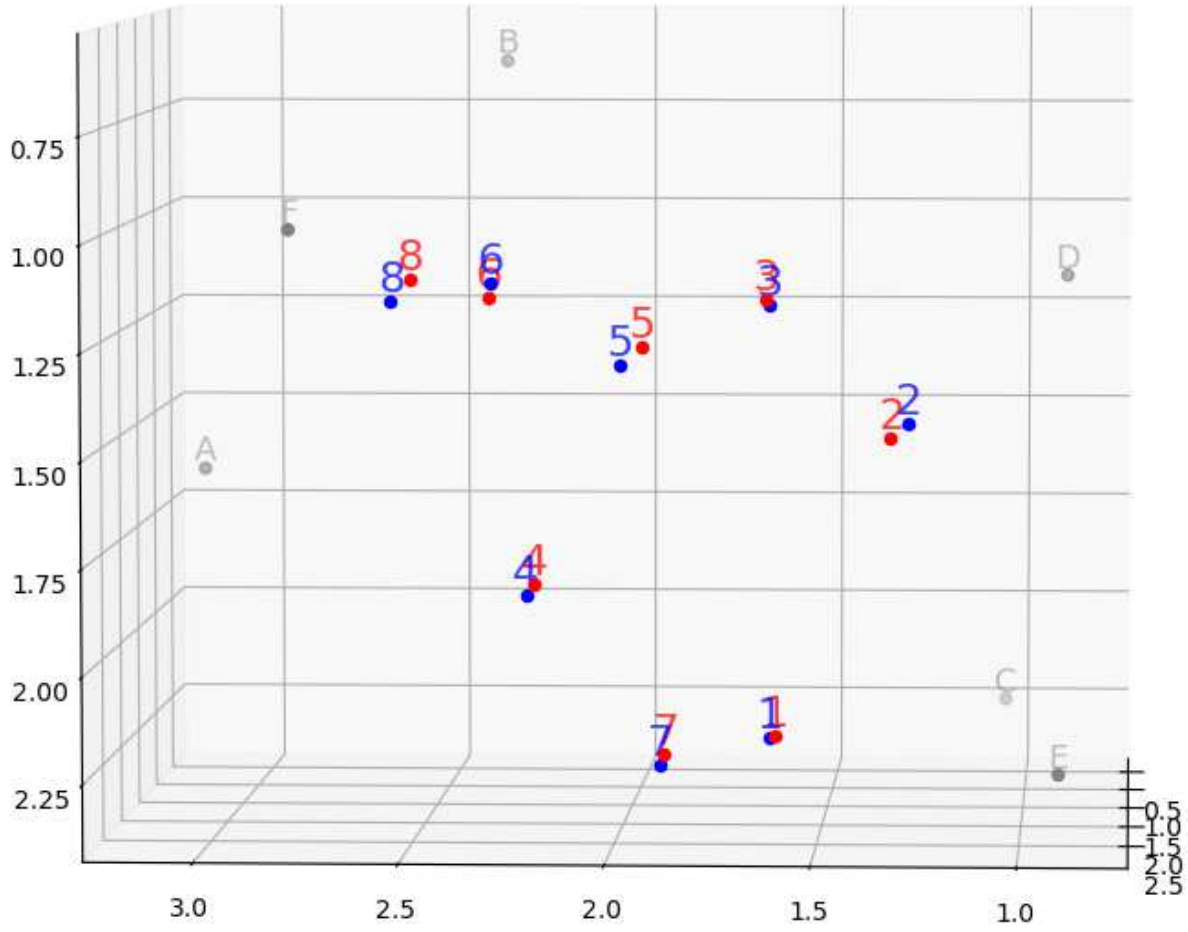

Figure 18: 3D plot of the measurement results as given in table 1 (in m). Reference points are visualized and labelled in red. Results obtained from the localisation system are colored in blue. The speakers are visualized with gray points and labels. The speakers and microphones 1 through 3 can also be seen in figure 17.

4.2.3 Discussion

The measurements were performed in an environment which gives freedom to take measures against multipath effects, such as laying out carpets and positioning the speakers without constraints. Additionally, no obstructions are in the way of any line of sight. This is not always the case, as in typical environments machines, struts and other obstructions are present and often unmovable.

The results in table 1 show that the localisation system works, for the given reference coordinates, within an accuracy of a few centimeters, under the assumption of correctly identified peaks. This is a good result, but could be improved significantly as the following tests show.

4.2.4 Validation using Multilateration Error Estimation

The first test is a simulation that consists of estimating the deviation of the microphone position localised by multilateration with regards to a deviation in the given speaker coordinates. This error estimation of the multilateration algorithm is performed statistically. A vector with a random orientation but of specified length is added to each of the speaker (reference nodes) coordinates. These new coordinates are then fed into the multilateration alongside the simulated travel times obtained from the coordinates without artificial error. This procedure investigates the dependence of an error while measuring the speaker coordinates and its implication on the error of the localisation result under the assumption of perfectly detected peaks and no quantization errors. 3 different microphone positions were used (microphone number 2, 7 and 8 in table 1). Speaker coordinates are the same as in figure 17 and figure 18. Figure 19 shows histograms with 10000 samples each for a random error vector of norm 0.01 m. These plots are done for each of the 3 selected microphones and for each of the x, y and z coordinates and the norm of the difference vector of the "true" microphone position to the localised microphone position. For the specified speaker configuration a near one-to-one relation is given between the deviation in speaker coordinates to the localised microphone coordinates. This can be seen from the mean of the norm distribution of the difference vector as given in figure 19. The near one-to-one relation also holds true for other deviations as can be seen in figure 20. There, the linearity for each coordinate and the norm of the difference

vector for realistic measurement deviations (1 to 15 cm) is shown. As expected, the coordinate deviations are centered around zero (mean). The standard deviation (std) of each coordinate is a little bit different. This is suspected to be dependent on the microphone position and speaker configuration. For an optimized speaker configuration, i.e. more variance in the z speaker coordinates, the standard deviation of the z coordinate gets smaller and the mean of the norm of the difference vector also becomes smaller than in figure 19, in other words, the localisation performs better. This leads to the conclusion that a simulation of different speaker configurations under the constraints of a given environment can improve the localisation result. A high variance in the speaker coordinates may improve the localisation performance.

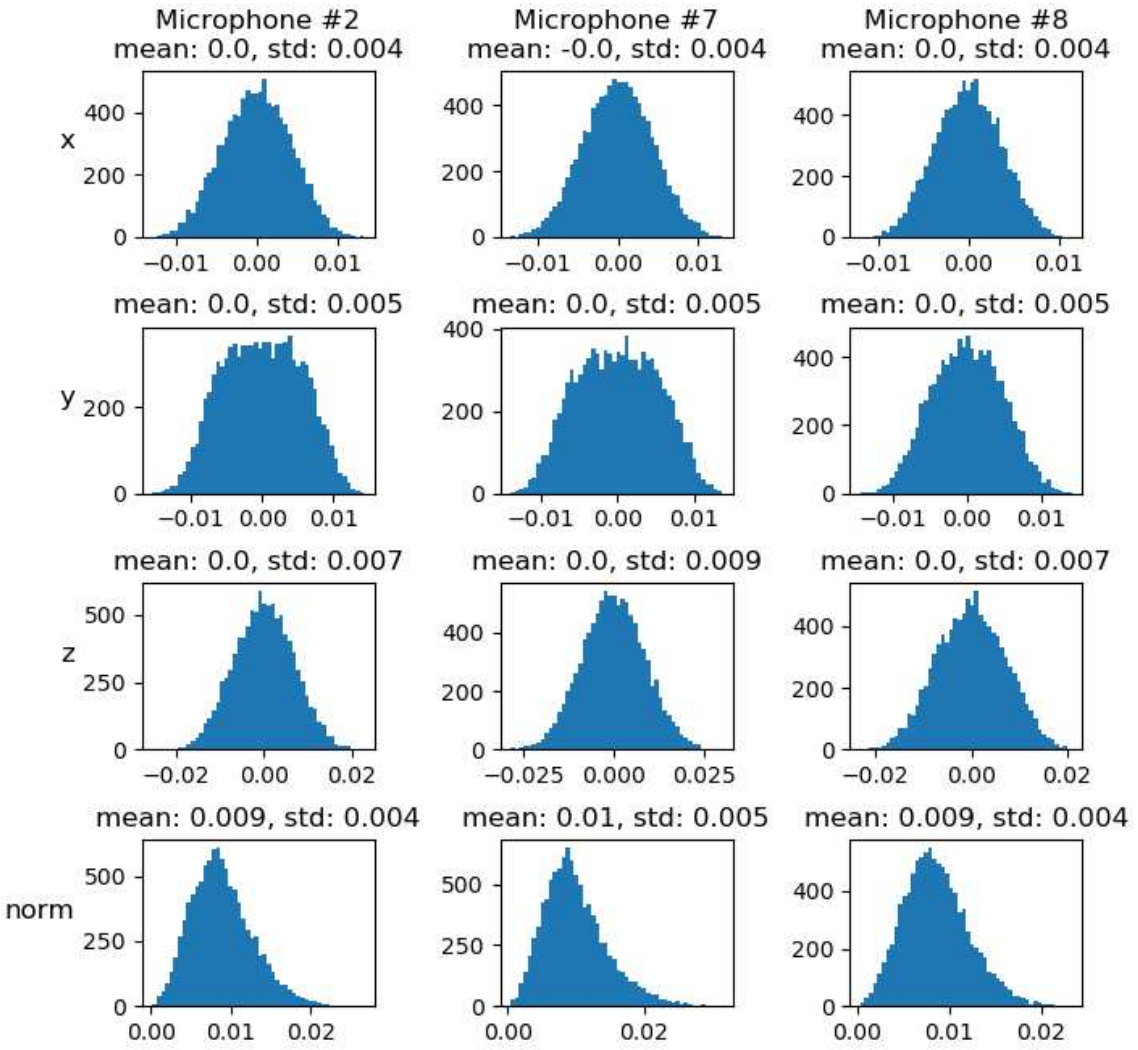

Figure 19: Error estimation for the multilateration algorithm. A random vector of norm 0.01 m is added to each of the speaker coordinates. These new positions are passed to the multilateration algorithm alongside the "true" times. A difference vector of the thereby localised and "true" microphone position is then calculated. Histograms of 10000 samples are plotted for each microphone and coordinate, including the norm of the difference vector (in m). This plot shows that an error of 1 cm in determining the speaker coordinates translates to an accuracy of about 1 cm of the multilateration algorithm. Figure 20 shows the linearity of this dependence, i.e. that this relation holds true for other random vector norms. Microphone coordinates are taken from table 1.

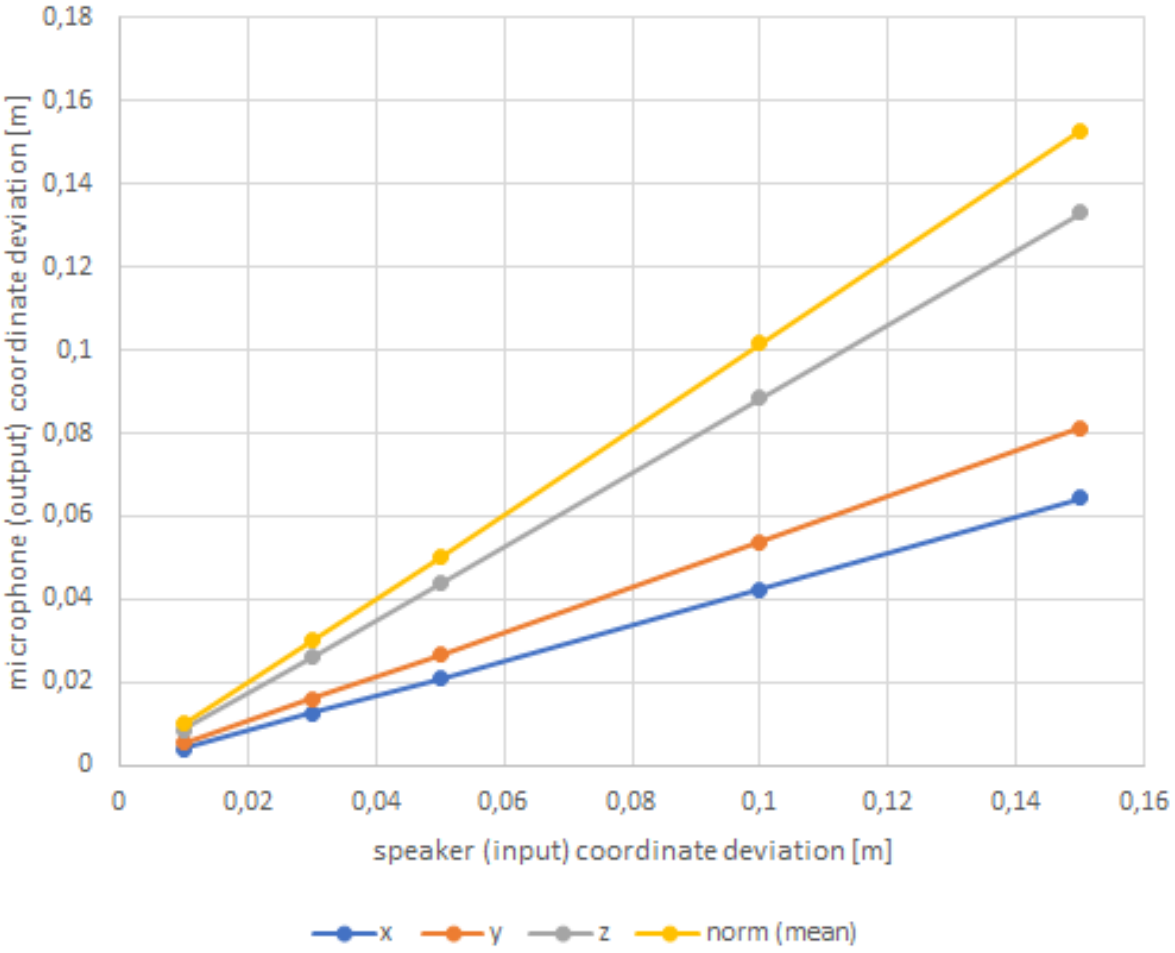

Figure 20: Error estimation for the multilateration algorithm. Figure 19 establishes the near one-to-one relation of deviations in the input coordinates to output coordinates of the multilateration algorithm. This plot shows that this relation holds true for all relevant ranges of 1 to 15 *cm*. This means that the uncertainty of the speaker coordinates directly translates to the localisation performance in a near one-to-one relation (for the given speaker geometry).
Calculations are performed the same way as described in figure 19, only for different deviations (1, 3, 5, 10 and 15 *cm*) and for microphone number 2 (first column).

4.2.5 Validation using Distance Matrices

The second validation test was performed by first measuring all distances from each entity (speakers and microphones) to all other entities manually using an inelastic string and a tape measure. This measurement technique is independent of the selected coordinate system, which may be subject to inaccuracies of the room geometry, i.e. non-right angled walls. Limitations of this measurement technique include the difficulty of holding the string steadily and correctly at the center of the entity without moving the entity. These distances are then saved as a distance matrix holding the experimentally acquired values.

As a next step, using a DAW, simple samples were played at predefined times from each speaker and recorded by each microphone. If the latency of the measurement system is known, the time difference between the beginning of the sent and received signal can be extracted using any means, including manual visual inspection. From this time delay, the distance from the speakers to the microphones can be calculated using the sampling frequency and the speed of sound. These speaker-to-microphone distances are then saved as a second matrix and compared to the previously described distance matrix based on tape measurements.

This second measurement technique uses onset detection of sound waves to calculate distances between speakers and microphones. This technique is only valid if the latency of the measurement equipment is known. Luckily, this latency can be measured similarly to the onset detection itself. While the software buffer used for audio processing can be compensated in the software, the latency of the measurement equipment needs to be measured and corrected for. It consists of the latency introduced by the audio interface, the latency of the amplifiers and response times of the cables, and both the speakers and microphones. These electromagnetic driven response times cannot be measured accurately without additional hardware, but are assumed to be negligible compared to effects based on sound propagation which are governed by the speed of sound. The audio interface latency is measured by using onset detection of a simple signal that gets send out from an output channel directly hooked up to an input channel. The sample used is a simple up and down rectangle of 38 samples length as seen in the top plot of figure 21. The waveform in the middle plot is the sample as it

is recorded from the audio interface. It is shifted by 3 samples into the past with regards to the signal sent out and originates from a latency overcompensation. The audio interface latency is therefore -3 samples. The latency of each amplifier can be measured similarly by connecting its output directly to an audio interface input. A latency of 2 samples is measured, resulting in a total effective latency of -1 samples.

To measure the physical distance from the speaker to a microphone, a sample is played and the time delay of arrival is visually determined with a manual onset detection. This time difference, measured in samples, is then latency compensated (-1 sample) and converted to a physical distance by dividing with the sampling rate and multiplying with the speed of sound. These results can be seen in table 2. A manual onset detection can be performed with an accuracy of about 1 sample as illustrated in figure 21 (bottom waveform). 1 sample at 192 kHz and 23 $°C$ translates to a sound propagation distance of 1.8 mm (as described in section 2.7).

While an onset detection technique may ultimately serve better results than a cross-correlation for calculating time delays from sound propagation, an accurate software implementation is expected to be difficult. One such implementation needs to be sample-accurate to provide the best possible results. Additionally, quiet environments are needed, when using the currently used hardware or new hardware needs to be acquired.

The distances from onset detection and from manual mechanical measurement are compared in table 2. It can be seen, that the absolute difference of these distances is smaller than 5 mm for each speaker to microphone combination. Considering the aforementioned limitations, this can be seen as a remarkable result and shows that both distance measurement techniques produce very good results.

Both distance matrices are now expected to exhibit a deviation of only a few millimetres from the true value. The distance matrix obtained by acoustic means only contains the distances between speakers and microphones and therefore the distances matrix obtained by manual mechanical measurement is used henceforth. This distance matrix can now be used to test the validity of the speaker coordinates used for the multilateration algorithm in section 4.2.2.

These coordinates are now used to calculate the distance matrix. This coordinate distance matrix can then be compared to the high-quality distance matrix by subtracting one from the other. Values range from -4 to 4 cm. These high inaccuracies of the speaker coordinates directly translate to a worse localisation performance, as discussed in section 4.2.4.

These results show that coordinates obtained from the distance matrix (be it from mechanical or acoustic measurements) could improve the localisation performance significantly.

Mic	Speaker	send	receive	range	distance	difference
		audio			tape measure	
	Label	samples	samples	m	m	mm
1	A	97919	98390	0,845266	0,850	4,7
	B	193919	194363	0,796708	0,797	0,3
	C	289919	291013	1,965692	1,966	0,3
	D	385919	386750	1,492703	1,496	3,3
	E	481919	483200	2,302000	2,304	2,0
	F	577919	578766	1,521478	1,525	3,5
2	A	97919	98714	1,427959	1,424	-4,0
	B	193919	195016	1,971088	1,974	2,9
	C	289919	290841	1,656361	1,654	-2,4
	D	385919	386826	1,629384	1,627	-2,4
	E	481919	482719	1,436952	1,436	-1,0
	F	577919	578973	1,893755	1,897	3,2
3	A	97919	98520	1,079063	1,077	-2,1
	B	193919	194734	1,463928	1,466	2,1
	C	289919	291365	2,598742	2,599	0,3
	D	385919	386964	1,877569	1,879	1,4
	E	481919	483077	2,080792	2,078	-2,8
	F	577919	578309	0,699592	0,701	1,4

Table 2: For each microphone/speaker combination a distance is measured via two different techniques as described in section 4.2.5. The difference of these two distances is within $\pm 4\ mm$, establishing the fact that both techniques produce good measurement results.
The distances obtained via audio are produced by playing a signal and detecting its onset in the recording. This time delay can only be measured in samples (column 3 and 4). The physical distance (column 5) can be recovered using the speed of sound for 23 $°C$ and the sampling rate of 192 kHz and by compensating for the sample latency. The difference in mm is then obtained by subtracting values in column 5 from column 6.

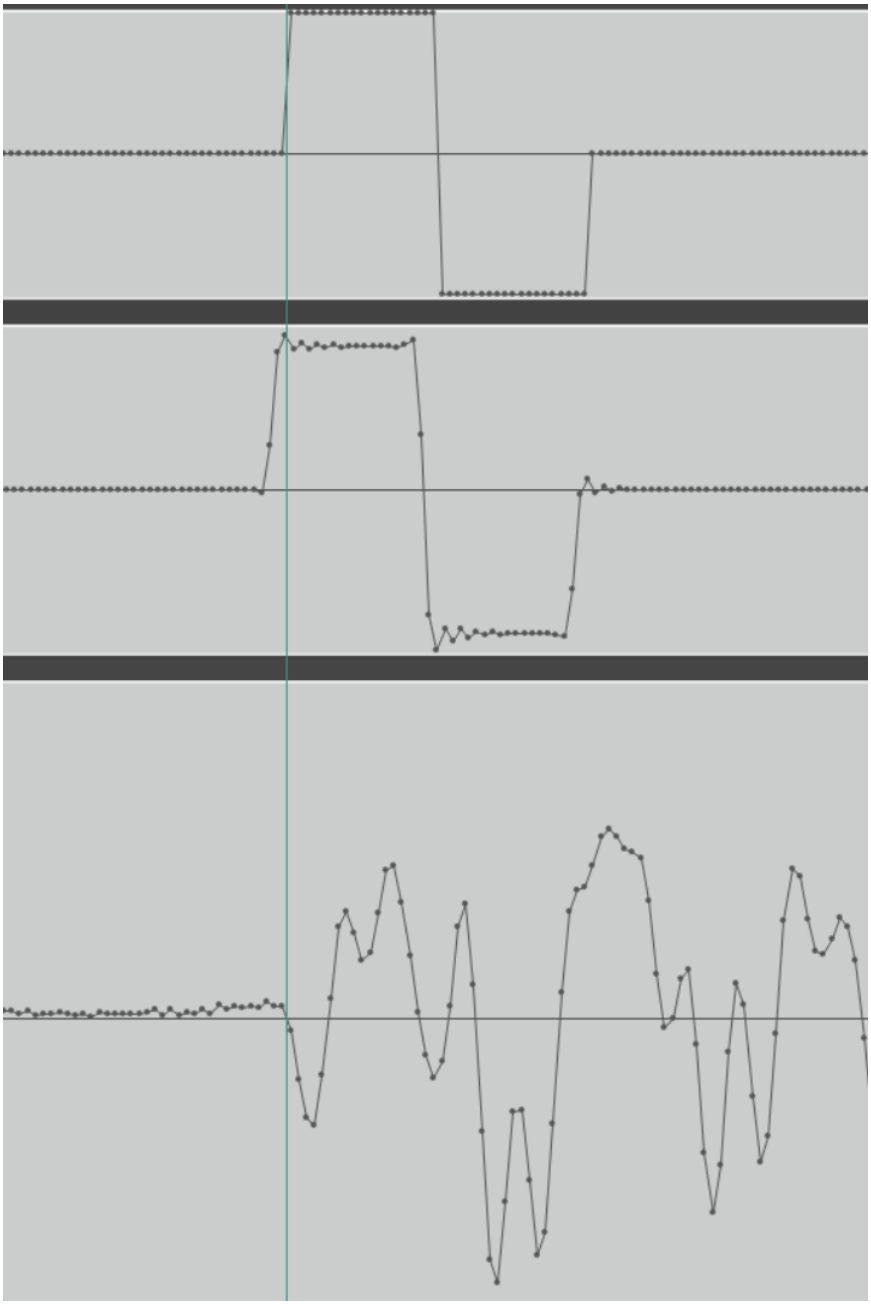

Figure 21: The top waveform is the sample as it was generated: A rectangle signal of 38 samples length. The middle waveform is the sample as it is recorded for the latency measurement of the audio interface. The latency of −3 samples is visible alongside a manifestation of the Gibbs phenomenon (overshooting by Fourier components of high frequency). The bottom waveform shows the sample as it arrives at the microphone. This recording has been realigned, as it would be delayed by a few hundred samples, rendering it invisible.
Each dot represents one sample. The horizontal distance between each dot is equivalent to 1.8 mm of sound propagation for the given environment (sampling rate of 192 kHz and 23 $°C$. An incorrect onset detection may therefore lead to a significant inaccuracy of the distance measurement.

4.3 Point to Point Procedure

After employing the principal microphone localisation procedure using multilateration another kind of procedure was tested. This method was named "Point to Point". This method uses only 1 instead of 6 speakers, which is manually put in front of each microphone head (using an adapter). Then, an audio sample is played and the signals of all microphones are recorded. As the speaker has a known distance to the microphone it's mounted on, the emission time is precisely known. If all microphones are recorded in sync, the distance from this microphone to all others can be recovered. These square matrix of distances (so-called ranges) can then be used to calculate the 3D geometry.

This procedure is quite labor-intensive, but was nevertheless investigated, partly as a proof-of-concept and partly as a backup solution. The premise was that this procedure is not as sensitive to multipath effects as the principal procedure due to the over-constraintness owing to multiple measurements of each microphones distance.

For this method, one speaker is positioned directly in front of the microphone head using a 3D printed adapter. This adapter ensures a consistent distance from the speaker to the microphone head to which it is attached. The microphones used have a head with a diameter of $1/2\ '' = 1.27\ cm$. Although the speaker is emitting in the opposite direction of the microphone, the signal is still loud enough to be easily detected by the microphone. Without considering the orientation of the microphones, a systemic error is introduced as the speaker is of course not at the exact same point as the microphone head. This distance changes depending on the position of the other microphone in question. The error becomes negligible for microphones perpendicular to the microphone-speaker axis, but maximal if the other microphone lies on this axis.

4.3.1 Measurement Setup and Geometry

As seen in figure 22, the geometry used for testing this procedure consists of 55 microphones mounted to the so-called dome structure. 48 of these microphones are arranged in groups of 4 with elements in a group sharing their x and y coordinates, i.e. they are arranged on top of each other. Another set of 6 microphones is mounted on top and one more microphone in the middle of the

structure's top surface.

The dome was placed around a tall air-to-water heat pump indoor unit. This heat pump water heater was analyzed for Annex 51 of the IEA (International Energy Agency) HPT (Heat Pumping Technologies) programme [34]. Air ducts are connected to the top part of the unit, which act as in and outtake of (outside) air. Noise emission of these outlets was recorded in the adjacent room with a separate set of microphones. Sound insulation was placed between the in- and outlet.

The line of sight to microphones on the opposite side of the dome is obstructed by the heat pump, leading to signal distortion and an increase in signal travel paths and therefore increasing the time of sound arrival. These wrong measurements should therefore by excluded from further calculation. For this specific setup, all microphones mounted on the dome's top are heavily obstructed by the aforementioned air ducts and cannot deliver usable localisation data.

Under the assumption that all microphones are correctly synchronized, the placement of the speaker directly in front of the microphone head allows a direct detection of the signal emission time. This is done similarly to the principal procedure by peak-picking the cross-correlation of the reference sample and the recording (as described in section 3.4). Similarly all other signal transmission times are detected. As the signal emission time is known, this procedure falls into the category of ToA/ToF (Time of Arrival, Time of Flight). From the transmission time of each microphone minus the emission at the first microphone, the signal travel time can be recovered and consequently, by multiplication with the speed of sound, the physical distance between these microphones.

As the distance from each microphone head to each other is known, all recovered distances can be handled in a 2D matrix (array). Obviously the diagonal elements are zero and ideally the matrix is symmetric. In reality though, this is not to be expected as a multitude of factors, primarily the occurrence of multipath effects of diffraction, but also the angle of speaker positioning and correct peak-picking, can lead to a false measurement.

Figure 23 shows a plot of the matrix of distances (ranges) with further explanation.

4.3.2 Energy minimization algorithm

To obtain the actual 3D geometry of the microphones from the matrix of distances (ranges) measured, an algorithm was developed, similar to minimizing the energy in a quadratic potential.

This algorithm starts with a random placement of microphones and repositions them in an iterative manner by minimizing the difference of distances to a target value. This approach was inspired by the self-positioning behaviour of atoms repelling and attracting each other under the electromagnetic interaction. The magnitude of the corresponding force is dependent on the distance from the zero-point in the energy potential. The Lennard-Jones potential is often used to model this interaction, but can be approximated by a parabolic potential (near the zero-point), yielding a potential similar to that of a harmonic oscillator. The atoms position themselves in a way that minimizes the total potential energy. For the algorithm, the microphones are treated the same as these atoms getting repositioned and the zero-point in the potential is given by the measured range from one microphone to another.

As a first step in the algorithm, all coordinate vectors $\vec{r_i}$ of each microphone i are distributed randomly around the origin. Then, for each iteration and for each microphone pair i and j, the following calculation is performed. An update factor u is calculated:

$$u = \frac{s}{2} \cdot (|\vec{r_i} - \vec{r_j}| - d(i,j)), \tag{30}$$

where $\vec{r_i}$ and $\vec{r_j}$ are the current microphone's coordinate vectors and $|\vec{r_i} - \vec{r_j}|$ gives the momentary distance between these microphones. $d(i,j)$ is a function taking the microphone numbers i and j as inputs to output a target distance. In an ideal scenario, the matrix of measured ranges is symmetric and $d(i,j)$ can just lookup element (i,j) of the ranges matrix. For real measurements the matrix is not symmetric and more advanced functions need to be employed. Further research is needed, but the `mean()` or `min()` of the matrix elements (i,j) and (j,i) appear as a good starting point for further refinement.

$(|\vec{r_i} - \vec{r_j}| - d(i,j))$ is a measure of distance to the optimal solution and acts on the coordinate vectors r_i attractive and repulsive, depending on the sign. In this regard, this measure can be seen as a linear force, similar to a spring

under load $F = k(x - x_0)$ (Hooke's law). This force corresponds to the potential energy of a harmonic oscillator.

s in Equation 30 is the step size or damping factor, in the range of $(0, 1)$ and used to prevent so-called overshooting of the solution and ensuring convergence of the iterative algorithm. The factor $1/2$ accounts for the fact that both microphone coordinates are updated:

$$\vec{r_i} \to \vec{r_i} + u \cdot (\vec{r_j} - \vec{r_i}) \tag{31}$$

$$\vec{r_j} \to \vec{r_j} - u \cdot (\vec{r_j} - \vec{r_i}) \tag{32}$$

The current implementation of this algorithm is very slow, but acts only as proof of concept. The algorithm's speed could be dramatically increased by:

- parallelizing calculations (creating a thread for each microphone, i.e. range matrix row)
- adaptation for calculation on a GPU (where vector and matrix multiplications are generally faster)
- using adaptive step sizes (damping factors) and other more sophisticated minimization algorithms
- starting from an already known geometry to converge faster ("educated guess")

The algorithm's accuracy can be improved by:

- assessing the systemic error that consists of the small distance between the speaker and microphone head (the microphone's orientation needs to be respected, but may be known from other constraints, i.e. pointing to the middle)
- penalizing or dropping wrong entries not passing plausibility tests
- adding additional constraints, such as the fact that certain microphones are grouped together and share the same x and y coordinates

- employing different weightings for the energy minimization (e.g. least squares minimization or dropping outliers)

Figure 24 shows a 3D plot of the results with a basic algorithm implementation.

4.3.3 Potential future usage - A Locator Device

The algorithm used to generate the coordinates from the matrix of distances could also be used in other localisation applications that need to be more flexible. One possible scenario being the following:

Each of the speakers used in the principal procedure could be mounted together with a microphone at a fixed distance. If > 3 of these combinations are installed such that all have an unobstructed line of sight to each other, the speakers could self-reference and calibrate themselves. Then no manual location measurement of the speakers (reference nodes) would be needed. Of course, the speaker to microphone orientation needs to be consistent across all speakers, but this can be respected while mounting with the help of a spirit (bubble) level.

Additionally, this combination could be fitted inside an enclosure in a small form factor. This device could also contain a microcontroller and an audio amplifier. For sharing of measurement data and synchronization, all of these devices (nodes) would need a network connection. Power could be transmitted via a dedicated cable or, if the network connection is wired, via Power over Ethernet (PoE). If the device should be completely wireless, WiFi and battery power could be used. Additionally, the need for a correct device orientation could be compensated by incorporating a built-in gyroscope sensor.

Such a system could be operated automatically, with the restriction of line of sights between the devices and under the assumption of suitable room acoustics needed for a good cross-correlation peak picking.

As already mentioned, the point to point procedure presented in this subsection is labor-intensive and therefore does not satisfy our need of automated microphone localisation. Nevertheless, the principal algorithm and the hypothetical device outlined above may have other applications and may lead to further research.

Figure 22: Photo of the setup used in the Point to Point procedure. The dome (with 55 microphones) is built around the indoor unit of a heat pump water heater, analyzed in the framework of the RRT4 (Round Robin Test 4) in Annex 51 of the IEA (International Energy Agency) HPT (Heat Pumping Technologies) programme [34]. The heat pump unit is obstructing the line of sight of microphones with with microphones on the opposite site leading to longer sound travel paths and wrong results. The air ducts for outside air in- and outlet are also obstructing the line of sight for microphones placed above them.

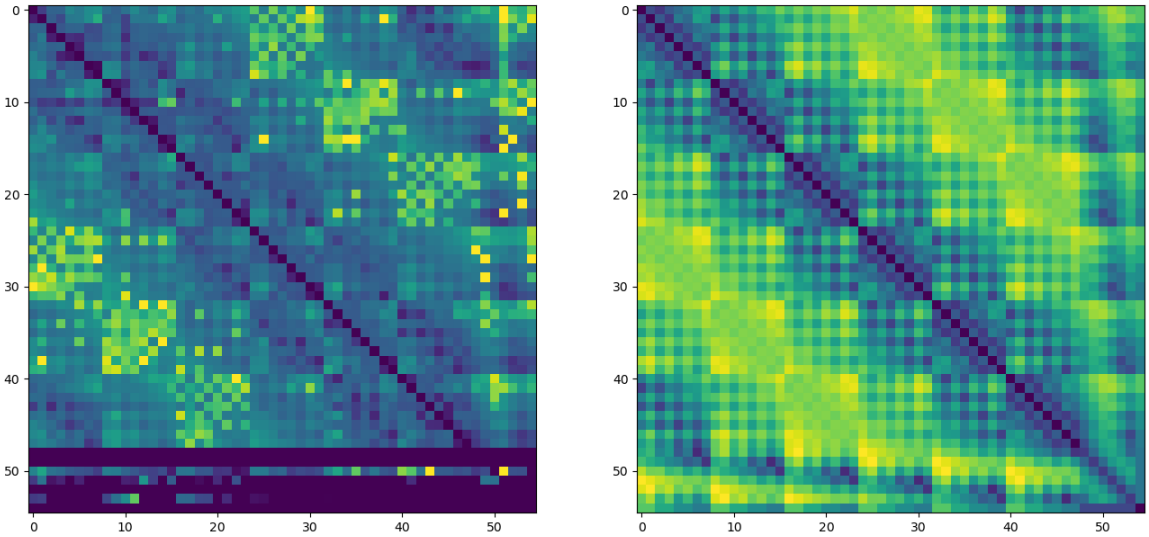

Figure 23: Left plot: Matrix visualization with cells in dark blue corresponding to elements of value zero. Yellow corresponds to a value of 4 m. Values above or below these limits, respectively, have been cut off. Clearly visible is the grouping in batches of 4 and 8 (as depicted in figure 22). Microphones in rows and column of 48 through 54 are placed on the top surface and cannot be identified correctly, because of line of sight obstruction. The matrix is not symmetric, because of multiple wrong measurements that need to be accounted for and corrected in the geometry generating algorithm.
Right plot: For comparison, a plot of the distance matrix of an ideal (target) geometry for another experiment, but also using the acoustic dome. The symmetric nature of all microphone groups is clearly visible.

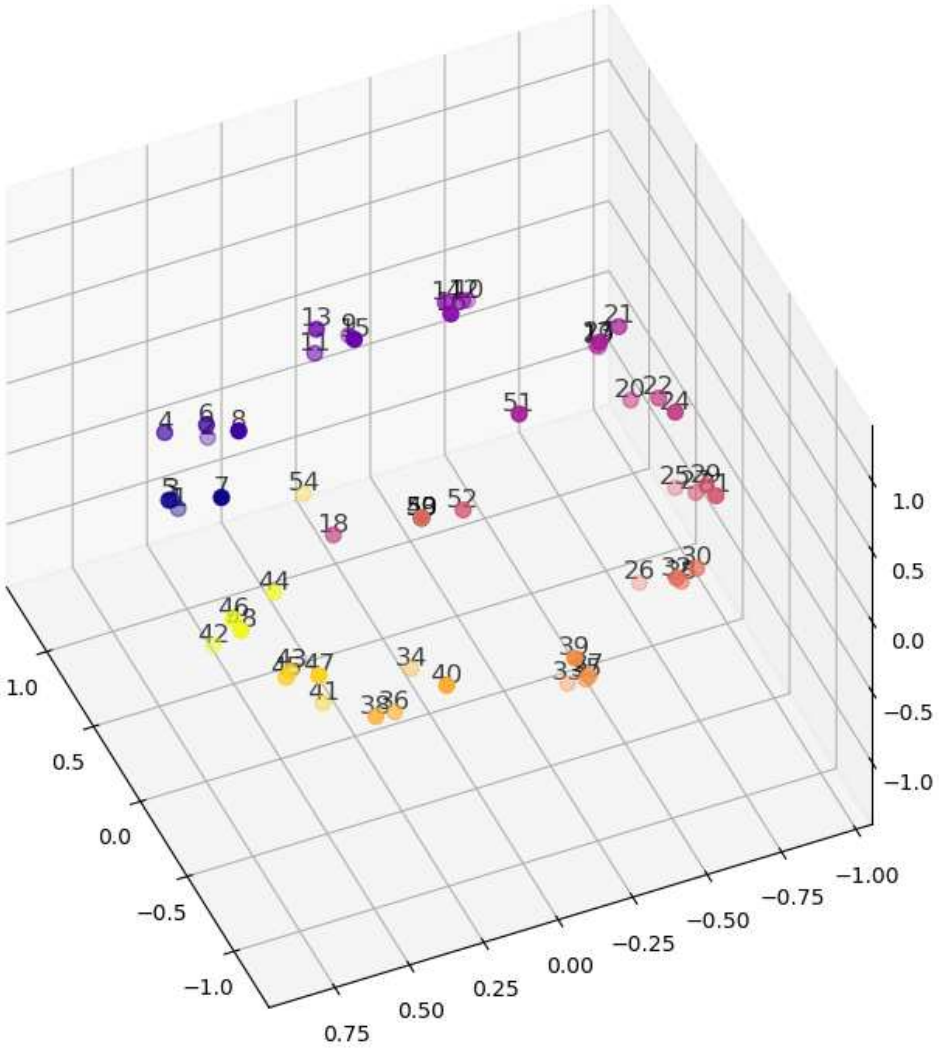

Figure 24: 3D plot of the results obtained with the Point to Point procedure viewed from the top. The orientation of the coordinate system does not reflect the intuitive system from reality. Instead, this coordinate system is rotated differently due to distributing all microphones randomly at the beginning of the solution algorithm. 48 of the 55 microphones are arranged in groups of 4 (plotted in the same color). Each group shares the same x and y coordinates and should therefore be plotted in a line, when viewed from top (as this is an orthographic view). This is not the case, because the algorithm is not optimized and erroneous measurements are still included. The top microphones are far away from a plausible position. Nevertheless, the fundamental geometry can be identified confirming the principal concept.

This implementation used the `mean()` of the two applicable elements of the ranges matrix, excluded outliers (with ranges above 4 m and below 0 m). Furthermore it excluded contributions from ranges which are greater than 50% of the maximum range for one of these microphones. This was done based on the assumption that longer signal travel paths are more likely to be elongated by diffraction from obstructions than shorter ones.

5 Conclusion & Outlook

This thesis presents the design and construction of a localisation system using multilateration and sound waves to localise multiple microphones. A self-designed sample consisting of a sequence of chirps is sent from speakers of a known location. The microphone signal can then be cross-correlated with the sample and the time delays identified and fed into the multilateration algorithm. The coordinates obtained this way have an uncertainty of a few centimeters at room scale.

Using simulations applying statistical methods it had been shown that the accuracy of the speakers' coordinates directly translates to the accuracy of the coordinates of the microphones when using the multilateration technique. Thus, if the coordinates of the speakers are known with an accuracy of e.g. 5 mm, the positions of the microphones could also be determined with an accuracy of 5 mm. Theoretically, the applied detection methods (onset, cross-correlation, ...) are capable of bringing the error down to the size of one sample. Using a sample rate of 192 kHz, this corresponds to an error in distance of around 1.7 mm. In reality, the signals recorded by the microphones will not exhibit an ideal shape, thus the theoretical limit of one sample will not always be achievable.

The cross-correlation could be supplemented with an onset detection algorithm given the appropriate hardware.

In complex scenarios, a lot of attention has to be paid to multipath effects. One possible solution may be the exclusion of wrongly characterized speaker information during the multilateration. The localisation performance can be compared for each combination of excluded speakers, allowing detection of wrong measurements.

Measuring coordinates of arbitrarily placed speakers in a new environment might be a very complicated task, if an accuracy in the range of a few millimeters is required. Therefore it is suggested to first place the speakers as desired followed by a measurement of all distances between these speakers. These distances can then be used to calculate the speaker coordinates using an (optimization) algorithm. These generated coordinates might require a translation and rotation operation to align them with a local reference coordinate system for easy handling.

The use of a sampling rate of 192 kHz is required for a localisation accuracy of a few millimeters. While signals with a lower sampling rate could be interpolated, a sampling rate of 192 kHz is standard for current audio hardware.

The employed algorithms can be improved and optimized by the following suggestions: The Peak-Picking algorithm can be further optimized to exclude wrong measurements (e.g. from multipath effects) as described in section 3.4. An improvement can be made for the multilateration algorithm by using other algorithms than the Newton-Raphson method detailed in section 2.5, such as a Weighted Least Squares approach [35]. Further research is needed to correctly implement the Voronoi algorithm used to obtain representative areas as described in 2.11.

The system could be expanded to allow continuous tracking with a higher polling rate than once every few seconds by implementing the changes mentioned in 3.3. The software could be adapted to "listen" continually and to perform analysis automatically.

The acquisition of higher quality hardware such as new amplifiers and smaller, but louder speakers (similar to those used in smartphones today) may improve performance and allow ultrasonic signals to be used.

While the system presented in this thesis was only used indoors, the principal procedure can be adapted for outdoor usage. Outdoor applications, however, may introduce new problems: louder environmental noise, wind noise, greater environmental influence variances (temperature, humidity, wind speed) and typically greater distances to cover, which may require louder sample playback.

Acknowledgments

I want to thank Professor Martin Gröschl [1] for giving me the opportunity to conduct this work.
I want to thank the whole SilentAirHP team for providing help, whenever needed.
I want to thank Christoph Reichl [2] for his ongoing advice, never-ending support and flexibility and for giving me the chance to prove myself in new projects, such as setting up a computing cluster.
I want to thank my partner, Julia Jaklin, for her support and help throughout this thesis.
Finally, I want to thank my family and friends for encouraging me to continue and finish my studies.

List of Figures

References

[1] Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Martin Gröschl, Technische Universität Wien. URL: `https://tiss.tuwien.ac.at/adressbuch/adressbuch/person/149208`.

[2] Dipl.-Ing. Dr.techn. Christoph Reichl, AIT – Austrian Institute of Technology, Energy Department, Sustainable Thermal Energy Systems. URL: `https://www.ait.ac.at/ueber-das-ait/researcher-profiles/?tx_aitprofile_pi1%5Bname%5D=Reichl%20Christoph`.

[3] Peter Wimberger. *Sound source localisation and space-, time- and frequency resolved analysis of sound emissions using a multichannel measuring system*, 2016. Bakkalaureatsarbeit, TU Wien, Institut für Angewandte Physik.

[4] Christoph Reichl, Mirza Popovac, Felix Hochwallner, Peter Wimberger, Felix Linhardt, Thomas Fleckl, and Johann Emhofer. Frosting and Defrosting Behavior of Evaporators. In *13th IEA Heat Pump Conference*, Jeju Island, South Korea, April 2021. (accepted).

[5] Christoph Reichl, Brigitte Blank-Landeshammer, Andreas Sporr, Gerwin Drexler-Schmid, Johann Emhofer, Mirza Popovac, Peter Wimberger, Christian Köfinger, Andreas Zottl, and Thomas Fleckl. Akustik von Wärmepumpen mit speziellem Fokus auf Vereisung, Abtauung und Platzierung. In *Chillventa Congress 2020*, online (accepted), October 2020.

[6] Christoph Reichl, Johann Emhofer, Gerwin Drexler-Schmid, Peter Wimberger, Felix Linhardt, Brigitte Blank-Landeshammer, Andreas Sporr, and Thomas Fleckl. Acoustic behaviour and placement of heat pumps - The perception of sound and heat pumps. In *The Essence of Heat Pumps Series*, Webinar online, September 2020.

[7] Christoph Reichl, Johann Emhofer, Peter Wimberger, Felix Linhardt, Norbert Schmiedbauer, Gerwin Drexler-Schmid, Brigitte Blank-Landeshammer, Andreas Sporr, Christian Köfinger, and Thomas Fleckl. Akustische Optimierung von Wärmepumpen (IEA HPT Annex 51). In *26. Tagung des BFE-Forschungsprogramms „Wärmepumpen und Kälte“*, BFH Burgdorf, June 2020.

[8] Christoph Reichl, Johann Emhofer, Peter Wimberger, Felix Linhardt, Norbert Schmiedbauer, Gerwin Drexler-Schmid, Brigitte Blank-Landeshammer, Andreas Sporr, Christian Köfinger, and Thomas Fleckl. Frosting Soft Sensor. In *IEA HPT – IOT Annex, IoT Heat Pump Conference*, AIT Austrian Institute of Technology, Vienna, January 2020.

[9] Christoph Reichl and Johann Emhofer. SilentAirHP Projekt Endbericht. 2019. URL: `https://www.energieforschung.at/assets/project/final-report/SilentAirHP-Publizierbarer-Endbericht-final.pdf`.

[10] Christoph Reichl, Johann Emhofer, Mirza Popovac, Gerwin Drexler-Schmid, Peter Wimberger, Felix Linhardt, Karoline Alten, and Thomas Fleckl. Annex51 and Experiences. In *Sound Workshop*, Vienna, October 2019.

[11] Christoph Reichl, Peter Wimberger, Felix Linhardt, and Johann Emhofer. Poster: Acoustic Emissions and Noise Abatement of Air to Water Heat Pumps. In *ICR 2019 - 25th IIR International congress of refrigeration*, Montreal, Canada, August 2019.

[12] Christoph Reichl, Johann Emhofer, Mirza Popovac, Gerwin Drexler-Schmid, Peter Wimberger, Felix Linhardt, Karoline Alten, and Thomas Fleckl. Akustische Emissionen von Wärmepumpen. In *Chillventa Congress 2018, 5. Innovationstag Kältetechnik*, Messe Nürnberg, Germany, October 2018.

[13] Christoph Reichl, Johann Emhofer, Mirza Popovac, Gerwin Drexler-Schmid, Peter Wimberger, Felix Linhardt, Karoline Alten, and Thomas Fleckl. International research: acoustic signatures of heat pumps. In *11de Warmtepomp Synposium, Communicatiehuis*, Gent, Belgium, October 2018.

[14] Peter Wimberger, Johann Emhofer, and Christoph Reichl. MicLocator - Determine multiple microphones' positions using sound wave delay and trilateration. In *68th Annual Meeting of the Austrian Physical Society*, TU Graz, September 2018.

[15] Mirza Popovac, Johann Emhofer, Elisabeth Wasinger, Peter Wimberger, Raimund Zitzenbacher, David Meisl, Felix Linhardt, Norbert Schmiedbauer, and Christoph Reichl. OpenFOAM implementation of algebraic frosting model and its applications on heat pump evaporators. In *13th IIR Gustav Lorentzen Conference on Natural Refrigerants (GL2018). Proceedings*, Valencia, Spain. IIF/IIR, June 2018. DOI: 10.18462/iir.gl.2018.1203.

[16] Christoph Reichl, Johann Emhofer, Mirza Popovac, Peter Wimberger, Felix Linhardt, Karoline Alten, and Thomas Fleckl. IEA HPT Annex 51: Acoustic Signatures of Heat Pumps Update - Acoustic Transmission Measurements and Sound Source Detection. In *26th Ercoftac ADA Pilot Center Meeting*, Graz, November 2017.

[17] Christoph Reichl, Mirza Popovac, Elisabeth Wasinger, David Meisl, Raimund Zitzenbacher, Felix Linhardt, Peter Wimberger, Norbert Schmiedbauer, Johann Emhofer, and Martin Gröschl. Experimental and numerical methods for the fluid dynamic and acoustic characterization of heat exchanger icing. In *67th Annual Meeting of the Austrian Physical Society (and Swiss Physical Society)*, CERN and CICG, Geneva, Switzerland, August 2017.

[18] Christoph Reichl, Mirza Popovac, Elisabeth Wasinger, David Meisl, Raimund Zitzenbacher, Felix Linhardt, Norbert Schmiedbauer, Peter Wimberger, and Johann Emhofer. Icing of heat exchangers by measurements and simulations on micro- and macroscale. In *25th ERCOFTAC ADA PC Meeting, Ercoftac Spring Festival*, AIT Austrian Institute of Technology, Vienna, April 2017.

[19] Christoph Reichl, Johann Emhofer, Peter Wimberger, Norbert Schmiedbauer, Felix Linhardt, Elisabeth Wasinger, Christian Köfinger, and Thomas Fleckl. SilentAirHP - Analyse und Entwicklung von Schallreduktionsverfahren für Luft-Wasser-Wärmepumpen. In *Fortschritte der Akustik - DAGA 2017*, pages 1246–1249, Kiel, Germany. Deutsche Gesellschaft für Akustik e.V. (DEGA), March 2017.

[20] Norbert Schmiedbauer, Johann Emhofer, Christian Köfinger, Peter Wimberger, Thomas Fleckl, Martin Gröschl, and Christoph Reichl. Aktive Störschallunterdrückung für Wärmepumpenanwendungen. In *Fortschritte der Akustik - DAGA 2017*, Kiel, Germany. Deutsche Gesellschaft für Akustik e.V. (DEGA), March 2017.

[21] Felix Linhardt, Karoline Alten, Johann Emhofer, Christian Köfinger, Thomas Fleckl, Peter Wimberger, Martin Gröschl, and Christoph Reichl. Charakterisierung der Schallabstrahlung von Luft-Wasser-Wärmepumpen mittels simultaner Hitzdrahtanemometrie, Vibrationsmessung und Schalldruckbestimmung. In *Fortschritte der Akustik - DAGA 2017*, pages 1238–1241, Kiel, Germany. Deutsche Gesellschaft für Akustik e.V. (DEGA), March 2017.

[22] Christoph Reichl, Johann Emhofer, Frieder Lörcher, Andreas Strehlow, Mirza Popovac, Peter Wimberger, Christian Köfinger, Andreas Zottl, and Thomas Fleckl. Transient Acoustic Signatures of the GreenHP with special focus on icing and defrosting. In *Proceedings of 12th IEA Heat Pump Conference*, Rotterdam, Netherlands, May 2017. ISBN: 978-90-9030412-0.

[23] Christoph Reichl, Johann Emhofer, Frieder Lörcher, Andreas Strehlow, Mirza Popovac, Peter Wimberger, Raimund Zitzenbacher, Christian Köfinger, Andreas Zottl, and Thomas Fleckl. GreenHP: Strömungs-Analyse der Verdampfer-Luftseite. In *DKV-Tagung 2016*, Kassel, Germany, November 2016.

[24] Norbert Schmiedbauer, Johann Emhofer, Christian Köfinger, Peter Wimberger, Thomas Fleckl, Martin Gröschl, and Christoph Reichl. Active Noise Cancelling for Heat Pump Applications. In *66th Annual Meeting of the Austrian Physical Society*, Universität Wien, September 2016.

[25] Peter Wimberger, Johann Emhofer, Christian Köfinger, Thomas Fleckl, Martin Gröschl, and Christoph Reichl. Space-, time- and frequency resolved recording and analysis of sound emissions and sound source localisation using a multichannel measuring system. In *66th Annual Meeting of the Austrian Physical Society*, Universität Wien, September 2016.

[26] Wi-fi certified location™ brings wi-fi indoor positioning capabilities. URL: https://www.wi-fi.org/news-events/newsroom/wi-fi-certified-location-brings-wi-fi-indoor-positioning-capabilities.

[27] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A survey of indoor localization systems and technologies. *CoRR*, abs/1709.01015, 2017. arXiv: 1709.01015. URL: http://arxiv.org/abs/1709.01015.

[28] William Navidi, William Murphy, and Willy Hereman. Statistical methods in surveying by trilateration. *Computational Statistics and Data Analysis*, 27:209–227, April 1998. DOI: 10.1016/S0167-9473(97)00053-4.

[29] William S. Jr. Murphy and Willy Hereman. Determination of a Position in Three Dimensions Using Trilateration and Approximate Distances. Technical report, Colorado School of Mines, 1995.

[30] Fernando J. Álvarez and Roman Kuc. Dispersion relation for air via kramers-kronig analysis. *The Journal of the Acoustical Society of America*, 124(2):EL57–EL61, 2008. DOI: 10.1121/1.2947631. eprint: https://doi.org/10.1121/1.2947631. URL: https://doi.org/10.1121/1.2947631.

[31] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, January 1964.

[32] Alberto Fornaser, Luca Maule, Alessandro Luchetti, Paolo Bosetti, and Mariolino Cecco. Self-weighted multilateration for indoor positioning systems. *Sensors*, 19:872, February 2019. DOI: 10.3390/s19040872.

[33] E. García, P. Poudereux, Á. Hernández, J. Ureña, and D. Gualda. A robust uwb indoor positioning system for highly complex environments. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 3386–3391, 2015.

[34] Annex 51 - Acoustic Signatures of Heat Pumps. URL: https://heatpumpingtechnologies.org/annex51/.

[35] Peng Wu, Shaojing Su, Zhen Zuo, Xiaojun Guo, Bei Sun, and Xudong Wen. Time difference of arrival (tdoa) localization combining weighted least squares and firefly algorithm. *Sensors*, 19:2554, June 2019. DOI: 10.3390/s19112554.

6 Appendix

6.1 Code of Multilateration algorithm

```
import numpy as np

def multilateration_algorithm(refpoints: Vectorlist,
        times: List[float], soundspeed) -> Vector:
    """This algorithm is an application of the procedure
       outlined in Murphy's Thesis from 1995 in which he
       describes a Newton minimization of the (nonlinear) sum
       of squares of errors for trilateration/multilateration.
    The first parameter takes the 3D coordinates of the
       speakers as array or tuple of triplets,
    the second parameter takes all time shifts in regards to
       the first microphone (e.g. [t2-t1,t3-t1,...] as a list
       of floats and
    the third parameter the speed of sound, which is dependent
       on the temperature in the environment"""
    amountrefpoints, checkis3d = np.shape(refpoints)
    #pivotindex sets the speaker, which was used to calculate
       the relative time offset from
    # needs to be 0
    # if you change this value, you need to adapt this
       algorithm (as it implicitly uses the fact, that
       pivotindex is 0) and change start_time_offset in the
       function, which calculates the times
       get_time_differences()
    pivotindex: Final = 0 #use typing's Final to indicate a
       constant for type checkers (like mypy)

    def rootofsumofsquares(x, y, z, i):
        xi, yi, zi = refpoints[i]
        return np.sqrt((x-xi)**2+(y-yi)**2+(z-zi)**2)

    def f(x, y, z, i):
        return rootofsumofsquares(x, y, z,
           i+1)-rootofsumofsquares(x, y, z,
           pivotindex)-soundspeed*times[i]

```

```
    # R is the starting point for the converging algorithm and
      can be chosen very generically.
    # We just use the most probable location to converge a
      fraction of a second faster
    R = [1.5, 3.5, 1.5]
    # Implement J and fvector according to Murphy and the
      special equations for our problem (see pdf)
    J = np.zeros((np.shape(refpoints)[0]-1,
      np.shape(refpoints)[1]))
    fvector = np.zeros(len(refpoints)-1)

    #*R unwraps R and is aquivalent to [R[0],R[1],R[2]]
    def fdiff(R, i, d):
        coordR = R[d]
        coordi = refpoints[i][d]
        coordstart = refpoints[pivotindex][d]
        return (coordR-coordi)/rootofsumofsquares(*R,
          i)-(coordR-coordstart)/rootofsumofsquares(*R,
          pivotindex)
    error = 1000 #arbitrary big number
    from numpy.linalg import inv, norm
    count = 0
    while(error > 1E-12):
        if (count > 1E6):
            print("stopped by reaching maximum iterations")
            break
        for i in range(J.shape[0]):
            for j in range(J.shape[1]):
                J[i][j] = fdiff(R, i+1, j)
            fvector[i] = f(*R, i)
        Rneu = R-np.dot(inv(np.matmul(J.T, J)), np.dot(J.T,
          fvector))
        error = norm(Rneu-R)
        R = Rneu
        count += 1

    return R
```

Listing 3: Code of Multilateration algorithm in python

6.2 Code of Sample Synthesis

```
# import golden ratio, definded by golden = (1 + 5 ** 0.5) / 2
from scipy.constants import golden
seednumber = 7
np.random.seed(seednumber)

self.data = np.zeros(10, dtype=np.float32)
totaltime = 0.
# we use the multiplication method, used in hashing in computer science,
# involving the golden ratio to get non-repeating (in difference to random) parameters
# duration_k is the increment used in the hashing multiplication method for each data subentry
# we reuse duration_k for sweeps and silence, starting from 0 is not permitted and one only yields the maximum duration
duration_k = 3
# start with arbitrary values
f_low_k = 78
f_high_k = 3
while(totaltime < allowed_max_duration):
    # decide if a sweep or a silence should be appended randomly
    if np.random.random() < 0.5:
        # append sweep:
        # possible durations (0..25ms) in samplerate
        duration_m = 0.025*samplerate
        # check https://de.wikipedia.org/wiki/Multiplikative_Methode for the formula
        # int() discards decimals, whereas () % 1 discards everything before the decimal point.
        duration_in_samples = int(
            duration_m*((duration_k*golden) % 1))
        duration_in_time = duration_in_samples/samplerate
        # same formula as for the duration, but this time frequencies, which do not need to be ints
        # low frequncies should between 5k..15k
        # high between 15k..30k
        f_low_m = 10000
```

```
31          f_high_m = 15000
32          f_low = f_low_m*((f_low_k*golden) % 1)+5000
33          f_high = f_high_m*((f_high_k*golden) % 1)+15000
34          f_low_k += 1
35          f_high_k += 1
36          # decide if the sweep should move up or down
37          if np.random.random() < 0.5:
38              up = False
39              f_start = f_low
40              f_end = f_high
41          else:
42              f_start = f_high
43              f_end = f_low
44          t = np.linspace(0., duration_in_time,
                duration_in_samples, dtype=np.float32) # 32 bit is
                the maximum sounddevice and thus portaudio can handle
45          # a linear frequency sweep/chirp as defined in
                https://de.wikipedia.org/wiki/Chirp#Linearer_Chirp
46          data = chirp(t, f_start, duration_in_time, f_end,
                method="linear")
47      else:
48          # append silence:
49          # possible durations (0..20ms) in samplerate
50          duration_m = 0.005*samplerate
51          # check out
                https://de.wikipedia.org/wiki/Multiplikative_Methode
                for the formula
52          duration_in_samples = int(
53              duration_m*((duration_k*golden) % 1))
54          data = np.zeros(duration_in_samples, dtype=np.float32)
55      duration_k += 1
56      self.data = np.concatenate((self.data, data))
57      totaltime += (duration_in_samples/samplerate)
```

Listing 4: Code of Sample Synthesis in `python` as described in section 3.3