Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

Niklas Strømsnes

Touch Localization by Estimating the Direction of Arrivals of Plate Edge Reflections

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Hefeng Dong
Co-supervisor: Tonni Franke Johansen
June 2023

NTNU
Norwegian University of
Science and Technology

Niklas Strømsnes

Touch Localization by Estimating the Direction of Arrivals of Plate Edge Reflections

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Hefeng Dong
Co-supervisor: Tonni Franke Johansen
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Abstract

This project explores the feasibility of touch localization using direction of arrival (DOA) estimation methods on Lamb waves reflected from plate edges. A trigonometric touch localization method is proposed for the feasibility study, utilizing the first three wave arrivals at a uniform linear array (ULA) of accelerometers located in one corner of the plate. By comparing the received signals at the array with the expected arrival times of the waves, we see that the first four wave arrivals can be separated from the rest of the reflections when the signal source is located approximately in the middle of the plate.

To gain a better understanding of the signal generated by a finger touch, we inspect the characteristics of a touch signal in a homogeneous $1000 \times 700 \times 10'$, mm Teflon plate and find that there is significant attenuation of the wave especially above 15 kHz. We find that a notable amount of wideband energy is contained in a touch swipe, and the release tension of the finger is enough to send a detectable pulse through the plate.

We consider four DOA estimation methods: MUSIC, Root-MUSIC, ESPRIT and Root-WSF, of which we propose Root-WSF to be the best suited for the problem. We use signals obtained from COMSOL Multiphysics simulations and real measurements to evaluate these methods and the proposed touch localization approach. The key findings from the simulations are that 1) these methods have problems estimating the DOA of the wave reflecting from both edges in the corner of which the array is located so that higher accuracy is achieved by only estimating three DOAs, and 2) Root-WSF seems to be the most accurate and robust method for these signals, being able to estimate all these three angles to below 0.5° error yielding a touch localization error of 0.8 cm.

The measurements performed are very limited in equipment and construction accuracy, and the results are therefore not very conclusive. However, we see indications that only two DOAs are found by the estimation methods, suggesting that the wave passing through the array before reflecting back is too attenuated or distorted. From the estimated touch locations, we see that the trigonometric touch localization method is not very robust with regards to errors in the estimated DOAs. Although Root-WSF is again able to locate a source in the middle of the plate within 0.8 cm using two DOAs at errors below 1°, this accuracy does not follow for other tested source locations.

Sammendrag

Dette prosjektet utforsker muligheten for berøringslokalisering ved bruk av metoder for estimering av ankomstvinkler (direction of arrival (DOA)) på Lamb-bølger som reflekteres fra kantene på en plate. Det foreslås en trigonometrisk metode for berøringslokaliseringen ved å utnytte de tre første ankomstene av bølger til et uniformt lineært array (uniform linear array (ULA)) av akselerometere plassert i et hjørne av platen. Ved å sammenligne mottatte signaler i arrayet med forventede ankomsttider for bølgene, ser vi at de fire første ankomstene av bølger kan skilles fra resten av refleksjonene når signalkilden befinner seg omtrent midt på platen.

For å oppnå bedre forståelse av signalet generert av en fingerberøring, undersøker vi egenskapene til et berøringssignal i en homogen Teflonplate med dimensjonene $1000 \times 700 \times 10\,\text{mm}$, og ser at det er betydelig demping av bølgen spesielt over 15 kHz. Fra undersøkelsen ser vi at en betydelig mengde bredbåndet energi er tilstede i en sveipende berøring, og at når fingeren forlater overflaten sendes en svak men detekterbar puls gjennom platen.

Vi vurderer fire metoder for DOA-estimering: MUSIC, Root-MUSIC, ESPRIT og Root-WSF, hvor vi foreslår at Root-WSF er den best egnede for problemet. For å evaluere disse metodene og den foreslåtte berøringslokaliseringsmetoden, bruker vi signaler som er hentet fra COMSOL Multiphysics-simuleringer og virkelige målinger. De viktigste funnene fra simuleringene er at 1) disse metodene har problemer med å estimere DOA for bølgen som reflekteres fra begge kantene i hjørnet hvor arrayet befinner seg slik at høyere nøyaktighet oppnås ved bare å estimere tre DOA-er, og 2) Root-WSF synes å være den mest nøyaktige og robuste metoden for disse signalene, da den er i stand til å estimere alle de tre vinklene med en feil på mindre enn $0.5°$, noe som resulterer i en berøringslokalisering med en feil på 0.8 cm.

Målingene som ble utført hadde begrenset utstyr og konstruksjonsnøyaktighet, og resultatene er derfor ikke veldig konkluderende. Imidlertid ser vi indikasjoner på at bare to DOA-er blir funnet av estimeringsmetodene, noe som antyder at bølgen som passerer gjennom arrayet før refleksjonen er for dempet eller forvrengt. Fra de estimerte berøringsstedene ser vi at den trigonometriske metoden for berøringslokalisering ikke er veldig robust med tanke på feil i de estimerte DOA-ene. Selv om Root-WSF igjen er i stand til å lokalisere en kilde i midten av platen med en feil på 0.8 cm ved hjelp av to DOA-er med feil under $1°$, finner vi ikke like lovende resultater for andre testede kildeplasseringer.

Preface

This thesis is written for the Department of Electronic Systems at the Norwegian University of Science and Technology (NTNU) in collaboration with SINTEF Digital in Trondheim, Norway. The idea for the project was proposed by SINTEF, where the starting point was that they wanted to develop a portable, single-enclosure device for touch-enabling surfaces. SINTEF is developing new accelerometers that they believe can enable new possibilities for touch localization. Although the intent of this project was a more direct evaluation of these accelerometers for the touch-sensing purpose, they did not become available in time.

The parts of the project related to the inspection of the touch signal were performed in collaboration with Vegard Iversen, whose thesis is also associated with the project.

Special thanks to Hefeng Dong from NTNU and Tonni F. Johansen from SINTEF Digital for supervising the project and for their helpful guidance on the acoustics part of the thesis. Thanks to Tim Cato Netland from NTNU for his help obtaining a plate for the measurements. Also, thanks to Camilla for her motivational words and great help bringing the badger home.

Contents

Acronyms

ADC analog-to-digital converter

AE acoustic emission

APR acoustic pulse recognition

B&K Brüel & Kjær

DML Deterministic Maximum Likelihood

DOA direction of arrival

ESPRIT Estimation of Signal Parameters via Rotational Invariance Techniques

MEMS microelectromechanical systems

MUSIC Multiple Signal Classification

SAW surface acoustic wave

SML Stochastic Maximum Likelihood

SNR signal-to-noise ratio

UCA uniform circular array

ULA uniform linear array

WSF Weighted Subspace Fitting

Chapter 1

Introduction

Since its introduction in the 1960s and its subsequent commercialization, touch-sensing surfaces have become increasingly important in industries and personal lives. A wide range of different touch-sensing technologies have been developed to meet the demands of a great variety of applications, with capacitive touch sensing being most significant in the current market [1].

While capacitive touch technologies are great in applications such as smartphones and tablets, they possess downsides that make their use in certain applications impractical, expensive or even impossible. A different approach, which has also found its place in the market, is resistive touch sensing. This cheaper and simpler alternative utilizes the change in resistance between two layers of a material upon a touch, but is limited in its transparency and multi-touch support. Other solutions exist with their respective advantages and disadvantages, such as various optical methods.

As an alternative to these touch localization methods, designs have been made using acoustic waves in a medium as the source of information. Acoustic touch sensing is often cheaper in comparison to capacitive sensing, and the touch input can in theory be anything as long as it generates a sufficiently strong acoustic signal. Other benefits include good transparency, which is desirable in applications such as touch screens or even windows or mirrors, and potentially simpler and more physically robust systems.

An early idea was developed during the 1980s when Dr. Robert Adler came up with a way to transmit and measure surface acoustic waves (SAWs) in the x- and y-directions of a plate [2]. By transmitting a pulse along the edges from two opposite corners of a plate, the waves could evenly be reflected perpendicularly inwards towards the plate before being directed towards two sensors in a corner on the other side of the plate.

Another example of an acoustic touch sensing technology is the acoustic pulse recognition (APR) technology developed by Elo TouchSystems [3]. Like the SAW technology described above, this technology also relies on acoustic waves in the medium, but with a different philosophy. Instead of transmitting SAWs, APR measures the Lamb waves that are excited by a touch on the surface of a plate. These methods are built on the concept of recognizing a signal "fingerprint", similar to solutions with time-reversal acoustics ([4]; [5]). An important implication of time-reversal acoustics is that the complex reverberation in the plate after an impact will be unique to that source location, and the accuracy increases as the wave propagation gets more

complicated. This fundamental robustness to plate inhomogeneities is a key reason why pulse recognition is a popular approach and could theoretically be done using as little as one sensor.

A pretty successful project was carried out at Stanford University in 2016 where a model was trained to recognize the pattern of how transmitted Lamb waves are attenuated when a touch is applied to a plate [6]. Somewhat reminiscent of the SAW method, their setup consisted of a transmitting and a receiving transducer on each end of the plate, but the difference was that they applied a pattern-recognizing model, pre-trained, to recognize many different touch locations. Not only did they achieve a promising accuracy, but the model can also recognize multiple touches simultaneously, which has historically been a challenge for most acoustic touch sensing technologies.

Localization of so-called acoustic emission (AE) sources is a somewhat related field of study with a different motivation. This is a relatively well-researched field, with useful applications within non-destructive testing of structures and impact localization on a chassis ([7]; [8]). One example of an interesting technique is using one or two Z-shaped sensor clusters and time-of-arrival correlation between the sensor elements in each cluster to estimate up to four angles of arrival [9]. The resolution of this system is naturally limited by the distance between the two clusters or the distance between elements if only one cluster is used.

Some apparent undesired features of the methods mentioned above are that they depend on additional traits from the sensors and transmitting actuators. They either have to be built into the plate, to be spread out over a larger area of the plate, to be trained by a large number of sample touches, or a combination of these. This is where we see a potential for improvement. What if we could have a portable device that can be placed on any surface and thereby touch-enable that surface?

One such concept was attempted in 2014 with a product called Toffee [10], where Xiao *et al.* wanted to place four sensors on the underside of everyday products such as a laptop or a phone, and then be able to provide some interactive touch-zones around that device. Using a relatively simple time-of-arrival approach for the direct wave between the sensors, they were able to reach a mean error of 4.3° in the estimated direction of arrival (DOA), and an indication of the distance to the touch by calculating a set of hyperbolas with possible touch locations on their intersections.

In the context of this project, Dahl *et al.* from SINTEF Digital argue in an unpublished white paper [11] that their new optical microelectromechanical systems (MEMS) accelerometer technology is expected to allow for inexpensive sensors matching or even outperforming its current alternatives. They also benefit from high manufacturing accuracy, producing highly consistent sensor properties. This raises the question of whether sensor arrays consisting of MEMS accelerometers can be used to accurately estimate a touch location on a plate, or even any arbitrary surface, to a level of confidence that can match the performance of capacitive touch devices. According to the paper, the new optical MEMS accelerometers can have specifications as shown in Table 1.1.

This report will start with an introduction to wave propagation in plates in Chapter 2 to better understand what signals we are working with, along with the basics of array processing that is central to this project. Chapter 3 describes how different estimation methods can be used to estimate the DOAs of multiple correlated signals, and a novel method for estimating the touch location based on the DOAs of the direct wave and two reflections will be presen-

Table 1.1: Estimated specifications of the new optical MEMS accelerometer from SINTEF [11].

Parameter	Estimated Value
Acceleration resolution	$\gtrapprox 10\,\mathrm{ng/Hz^{1/2}}$
Sensor size	$3 \times 3 \times 2\,\mathrm{mm}$
Sensitivity (SNR)	More than 15 dB better than competing solutions

ted. In Chapter 4, we will present the details of the experiments and an explanation of the implementation of relevant preprocessing. Finally, results and discussions of the experiments are presented in Chapter 5, including an inspection of measured touch signals and first results from testing the suggested touch localization method.

1.1 Project Scope and Limitations

Both plate acoustics and array processing are complex topics that quickly get too advanced to be fully covered in this project. Especially since a central part of the project has been researching the possibilities of the proposed single-enclosure touch localization concept. The project is intended as a feasibility study of using an accelerometer array close to a plate's corner to estimate the DOAs of the direct wave and edge reflections. There is likely room for improvement by better understanding the physics of the systems at hand and further adjusting the implemented methods.

The results of the physical tests in Section 5.2.2 should be interpreted with caution, as many factors could have affected the results. Only three non-identical accelerometers were available in experiments that require a minimum of five, preferably identical, accelerometers. One of the accelerometers also had the wrong datasheet attached, causing some uncertainty in its actual response. The workaround is described in Section 4.2.2. Some results are obtained from COMSOL Multiphysics simulations, which are limited in their accessibility due to the long simulation times for the desired simulated environment. Consequently, little quantification of the accuracy of the methods is done, moving the focus over to a more qualitative evaluation of the results.

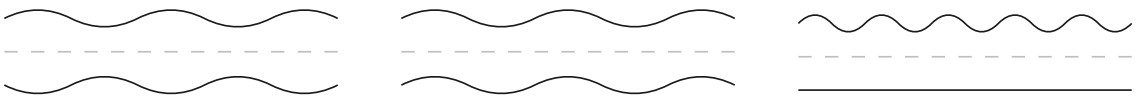
Chapter 2

Theoretical Background

Before diving into the theory behind the methods used to estimate the DOAs and touch locations in Chapter 3, we will first present some of the theoretical background for the problem. In Section 2.1 a relatively surface-level description of the waves expected to propagate in a plate is given. The mathematical descriptions of these waves are outside the scope of the project, but should not be necessary for understanding the rest of the report. Section 2.2 gives an overview of using sensor arrays for signal processing, and describes the mathematical assumptions made for the received signals and array output. Afterward, the relevant concept of analytic signals is presented briefly, as this is the form we represent the signals in for the DOA estimation.

2.1 Wave Propagation in Plates

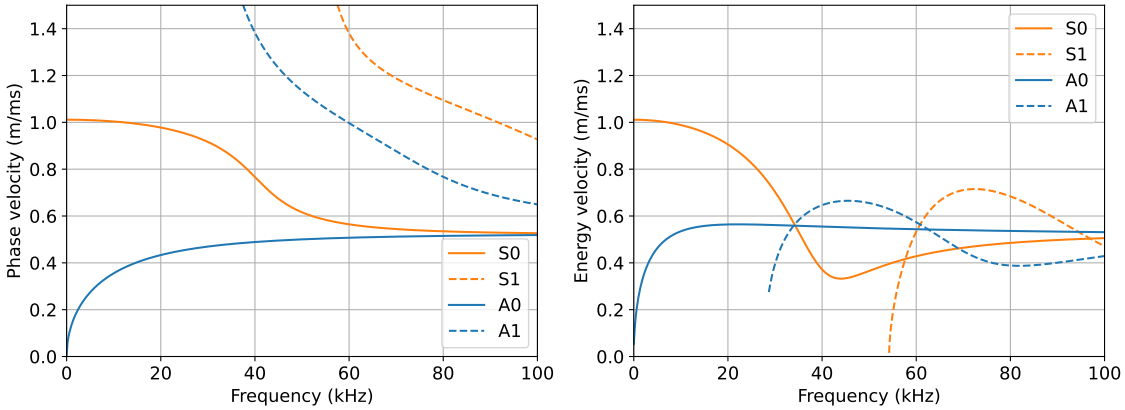
When energy is transferred to a medium, it can excite acoustic waves that propagate through the medium. The nature of these waves can be complex and depend on various properties such as the material, thickness, and geometry of the medium, as well as the excitation method used. For a point force impact on the surface of a thin plate, i.e. the wavelength is long compared to the thickness of the plate, the waves are mainly expected to propagate as Lamb waves [4].



(a) Symmetric mode. **(b)** Antisymmetric mode. **(c)** Rayleigh wave.

Figure 2.1: Diferent wave types commonly found in plates, and their characteristic displacement of the plate. The dashed center line represents the mid-plane of the plate.

Lamb waves can be categorized into symmetric and antisymmetric modes. The symmetric modes cause the motion on both sides to be symmetric with respect to the mid-plane of the plate, and the antisymmetric modes cause the motion to be antisymmetric, as illustrated in Figure 2.1a and Figure 2.1b, respectively. Figure 2.1c illustrates the surface wave, also known as the Rayleigh wave, that is more commonly found in thicker plates or shorter wavelengths [12].



(a) Phase velocities. **(b)** Group velocities.

Figure 2.2: An example of what the phase and energy velocities could look like in a plate [13].

Figure 2.2 shows an example of possible propagation velocities for Lamb waves in a plate, as a function of the frequency-thickness product calculated numerically [13]. As illustrated, the first-order modes and higher will be non-existent up to a certain frequency-thickness product, and will start at a high velocity before converging towards the shear wave velocity. The zero-order asymmetrical mode, also known as the A0 mode, bending wave or flexural wave, starts at zero velocity and increases before converging towards the Rayleigh wave velocity, making it highly dispersive for lower frequencies. The zero-order symmetrical mode, also called the S0 mode or the extensional wave, starts with an almost constant phase velocity, and converges down towards the shear wave velocity.

The physical process of a Lamb wave reflecting off the edge of a plate is complex and differs depnding on the frequency-thickness product and the Lamb wave mode [14]. Considering the energy of an incident wave Φ_{inc}, and the energy of the reflected wave Φ_{ref} to be a sum of n Lamb waves $\Phi_{\text{ref}} = \sum_n \Phi_{\text{n}}$, the reflection wave energy coefficient R_{n} of the mode can be defined as

$$R_{\text{n}} = \frac{\Phi_{\text{n}}}{\Phi_{\text{inc}}}, \tag{2.1}$$

and assuming that the wave energy is maintained during the reflection, we have

$$\sum_n R_{\text{n}} = 1. \tag{2.2}$$

According to Cho and Rose in [15], mode conversions upon free edge reflections can only occur between modes with the same symmetry. They also find that an incident zero-order Lamb wave will mainly reflect as the same wave mode for values of the frequency-thickness product below the cutoff frequencies of the higher modes.

Under the assumption that Lamb waves will reflect from the plate edges with the same angle as it was incident, the use of *mirrored sources* can be used to calculate the direction of arrival (DOA) at a point on the plate given the point of the source ([16], p. 135). The travel path of a wave from the source to the receiver when it is reflecting from the right edge and

then the bottom edge is illustrated in Figure 2.3. For a wave reflecting from two edges, the second mirrored source will have the position of the first mirrored source mirrored across the second edge. Seen from the receiver, the wave will appear to arrive from the last mirrored source.

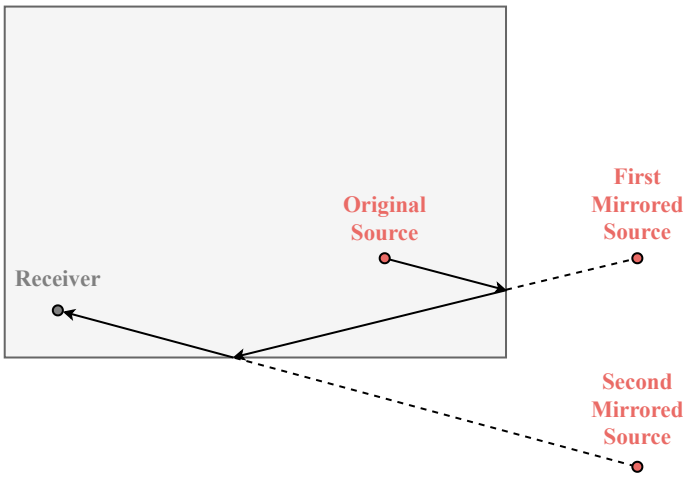


Figure 2.3: An illustration of how the travel path of reflecting waves can be equivalently modeled by mirrored sources. Based on a figure from ([16], p. 135).

2.2 Array Processing

Sensor and antenna array processing is a well-established field of research, with many applications due to its ability to enable high directivity when transmitting and receiving wave signals. A common application of array processing is to utilize the spatial information of the array to form a desired response pattern, which is often used to enhance or suppress signals arriving from different directions. This field is called *beamforming*.

Figure 2.4 illustrates a plane wave arriving at a line of uniformly spaced sensor elements, a so-called uniform linear array (ULA), where the broadside angle of arrival is denoted θ. It is evident that a plane wave arriving at an angle θ will have a different time of arrival at each sensor element and a different phase at each sensor element given by

$$\tau = \frac{d}{v} \sin\theta \tag{2.3}$$

when the distance between elements is d and the wave propagates at a velocity v.

The rest of the theory presented in Section 2.2 is based on [17]. Given that an arriving narrowband signal $s(t)$ will not change significantly between the L sensor elements spaced at a distance d in a ULA, the signals received at each sensor can be described through multiplying $s(t)$ with the steering vector

$$\mathbf{a}_{\text{ULA}}(\theta) = g(\theta)\left[1, \quad e^{-jkd\cos\theta}, \quad \ldots, \quad e^{-jkd(L-1)\cos\theta}\right]^T \tag{2.4}$$

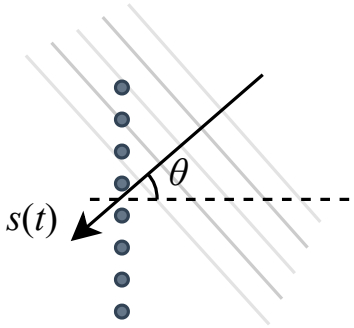


Figure 2.4: A plane wave $s(t)$ arriving at an eight-element ULA where θ illustrates the broadside angle of arrival.

where $g(\theta)$ is the directivity of the sensor elements. Then the output of the array can be described as

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t), & x_2(t), & \ldots, & x_{\mathrm{L}}(t) \end{bmatrix}^T = \mathbf{a}_{\mathrm{ULA}}(\theta)s(t) + \mathbf{n}(t) \tag{2.5}$$

where $\mathbf{n}(t)$ contains the noise of each sensor element

$$\mathbf{n}(t) = \begin{bmatrix} n_1(t), & n_2(t), & \ldots, & n_{\mathrm{L}}(t) \end{bmatrix}^T \tag{2.6}$$

In the case of M signals

$$\mathbf{s}(t) = \begin{bmatrix} s_1(t), & s_2(t), & \ldots, & s_{\mathrm{M}}(t) \end{bmatrix}^T \tag{2.7}$$

arriving at the array from different DOAs

$$\boldsymbol{\theta} = [\theta_1, \quad \theta_2, \quad \ldots, \quad \theta_{\mathrm{M}}]^T, \tag{2.8}$$

the output of the l-th sensor $x_{\mathrm{l}}(t)$ will be a superposition of the individual signals

$$x_{\mathrm{l}}(t) = \sum_{m=1}^{M} s_{\mathrm{m}}(t) + n_{\mathrm{l}}(t) \tag{2.9}$$

The array output vector can then be formulated in the same fashion as (2.5) as

$$\mathbf{x}(t) = \mathbf{A}_{\mathrm{ULA}}(\theta)\mathbf{s}(t) + \mathbf{n}(t) \tag{2.10}$$

where $\mathbf{A}_{\mathrm{ULA}}(\boldsymbol{\theta})$ is the $L \times M$ steering matrix of the array, defined for a ULA as

$$\mathbf{A}_{\mathrm{ULA}}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{a}(\theta_1), & \mathbf{a}(\theta_2), & \ldots, & \mathbf{a}(\theta_{\mathrm{M}}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ e^{j\theta_1} & e^{j\theta_2} & \ldots & e^{j\theta_{\mathrm{M}}} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j(L-1)\theta_1} & e^{j(L-1)\theta_2} & \ldots & e^{j(L-1)\theta_{\mathrm{M}}} \end{bmatrix} \tag{2.11}$$

A useful property of the ULA steering matrix is that it takes the form of a Vandermonde matrix, which is a matrix with the elements of a geometric progression in each row. Since the use

of the steering matrix also works for other array geometries, it will hereby be denoted as $\mathbf{A}(\boldsymbol{\theta}) = \mathbf{A}_{\text{ULA}}(\boldsymbol{\theta})$ for the simplicity of the notation.

A lot of information is contained in the spatial covariance matrix $\mathbf{R}$ of the sensor array, defined as

$$\mathbf{R} = \mathbb{E}\left\{\mathbf{x}(t)\mathbf{x}^H(t)\right\} \tag{2.12}$$

where $\mathbb{E}\{\cdot\}$ denotes the expectation operator and $(\cdot)^H$ denotes the Hermitian transpose. Under the assumption of white noise with variance σ^2 independent between sensors, the noise covariance matrix can be expressed using the identity matrix $\mathbf{I}$ so that

$$\mathbb{E}\left\{\mathbf{n}(t)\mathbf{n}^H(t)\right\} = \sigma^2\mathbf{I} \tag{2.13}$$

From the linear property of the $\mathbb{E}\{\cdot\}$ operator, it follows that (2.12) is equivalent to the expectated value of (2.10), which yields

$$\mathbf{R} = \mathbf{A}\mathbf{P}\mathbf{A}^H + \sigma^2\mathbf{I} \tag{2.14}$$

where $\mathbf{P}$ is the signal covariance matrix

$$\mathbf{P} = \mathbb{E}\left\{\mathbf{s}(t)\mathbf{s}^H(t)\right\} \tag{2.15}$$

The spatial covariance matrix can alternatively be expressed in terms of the eigenvalues sorted from largest to smallest on the diagonal of Λ as

$$\mathbf{R} = \mathbf{U}\Lambda\mathbf{U}^H \tag{2.16}$$

where $\mathbf{U}$ is a unitary matrix of eigenvectors.

A practial approximation of $\mathbf{R}$ can be obtained from the array output $\mathbf{x}(t)$ as the sample covariance matrix

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}(t)\mathbf{x}^H(t) \tag{2.17}$$

2.2.1 Analytic Signals

A more efficient way of representing the information in a real signal is to transform it to an *analytic signal*. An analytic signal is a complex representation of a signal, defined as having no negative frequency components [18]. This means that whereas a real signal $x(t)$ has a spectrum $\mathcal{F}\{x(t)\} = X(f) = \bar{X}(-f)$, the analytic signal has a frequency spectrum $X(f) = 0$ for $f < 0$. By turning $x(t)$ into an analytical signal, the new complex signal

$$x_{\text{a}}(t) = a(t)e^{j\phi(t)} \tag{2.18}$$

obtains useful properties such as instantaneous amplitude $a(t)$, instantaneous phase $\phi(t)$ and instantanesous frequency $\omega(t) = \frac{d\phi}{dt}$ [19]. The individual samples of the time signal now contain information about the phase and power of the signal.

Analytic signals are related to the Hilbert transform as

$$x_{\text{a}}(t) = x(t) + j\mathcal{H}\{x(t)\} \tag{2.19}$$

where $\mathcal{H}\{\cdot\}$ denotes the Hilbert transform. From $x_{\text{a}}(t)$, the original signal can be obtained as the real part $x(t) = \mathcal{R}\{x_{\text{a}}(t)\}$ and the envelope of the signal can be obtained from the absolute value $x_{\text{envelope}} = |x_{\text{a}}(t)|$.

Chapter 3

Methods

This chapter first provides a mathematical description of four DOA estimation methods and how the estimation of coherent sources can be improved through spatial smoothing. Then follows a short discussion of array geometries, before presenting a description of a suggested method to localize a touch using three DOAs. Many techniques exist for estimating the DOAs of multiple sources. However, the ones presented here are chosen based on their relevancy to the problem and ease of implementation through MATLAB System objects as described in Section 4.4.

3.1 Direction-of-Arrival Estimation Methods

The theory behind the following DOA estimation methods in Section 3.1 will be primarily based on a paper from the University of Oslo summarizing and comparing several different array signal processing research [17]. In their paper, Krim and Viberg present some interesting qualities of different estimation methods, shown in Table 3.1. The table summarizes some of the key properties of the methods, such as the consistency, meaning that it will converge towards the true value for infinite samples; the ability to work on coherent signals, meaning that the signals are a scaled or delayed version of each other; the statistical performance, indicating how well it asymptotically reach the Cramér-Rao Bound.

For a problem such as estimating the DOAs of reflections of a signal, some limitations on the choice of method are imposed. First of all, we expect the reflections to be correlated with each other or even coherent. This means that methods such as Capon and MUSIC should not be suitable. Deterministic Maximum Likelihood (DML) and Stochastic Maximum Likelihood (SML) are both potential candidates but are not considered due to the time constraints for the project.

Previous research has suggested that Weighted Subspace Fitting (WSF) and Root-WSF are suitable for the use of signal reflections. In a paper from 2015, Tervo and Politis compared DML, SML and WSF to MUSIC when applied a spherical array of microphones measuring reflections of an impulse transmitted in a room [20]. They conclude that DML and WSF perform the best and that WSF is the most computationally attractive. Similar results are found in [21], showing great estimation performance for WSF and DML. Viberg *et al.* find in [22] that WSF has some benefits over DML, such as being "limited only by the estimation accuracy and not by the

initialization or detection" and that there is a "vast improvement" in detection when correlated or coherent signals are present. In [23], Boustani *et al.* compare the performance of Root-WSF to Root-MUSIC under hard conditions. They find that Root-WSF gives the most accurate results when the signals are under "severe conditions responsible for signal deterioration", considering correlated sources and closely spaced sources among their criteria. Naturally, we expect WSF and Root-WSF to be the most promising techniques for the problem at hand.

To limit the scope of the project, we will test Root-WSF, ESPRIT, Root-MUSIC and, despite its issues with coherent sources, MUSIC as it is one of the most popular methods for multiple signal DOA estimations. Note that the time complexity of the methods does not fall within the scope of this project.

While methods such as Root-MUSIC and ESPRIT are designed for ULAs, methods have been developed to transform measurements from a UCA into a similar structure as a ULA using phase excitation [24], thus allowing for ULA methods to be used on UCAs.

Table 3.1: An overview of a selection of popular DOA estimation methods, from [17].

Method	Consistency	Coherent Signals	Statistical Performance	Array Geometry
Capon	No	No	Poor	Arbitrary
MUSIC	Yes	No	Good	Arbitrary
DML	Yes	Yes	Good	Arbitrary
SML	Yes	Yes	Efficient	Arbitrary
WSF	Yes	Yes	Efficient	Arbitrary
Root-MUSIC	Yes	Yes	Good	ULA
ESPRIT	Yes	Yes	Good	ULA
Root-WSF	Yes	Yes	Efficient	ULA

3.1.1 MUSIC

The Multiple Signal Classification (MUSIC) algorithm has become a popular method for estimating the DOA for multiple sources since its introduction [25]. The algorithm is based on the concept of subspace decomposition, which involves decomposing the covariance matrix of the received signals into two subspaces: the signal subspace and the noise subspace. Starting with the estimated covariance matrix $\hat{\mathbf{R}}$ as in (2.17), the eigendecomposition of $\mathbf{R}$ in (2.16) can be separated into a signal subspace and a noise subspace so that

$$\hat{\mathbf{R}} = \hat{\mathbf{U}}_\mathrm{s}\hat{\Lambda}_\mathrm{s}\hat{\mathbf{U}}_\mathrm{s}^H + \hat{\mathbf{U}}_\mathrm{n}\hat{\Lambda}_\mathrm{n}\hat{\mathbf{U}}_\mathrm{n}^H \tag{3.1}$$

where $\hat{\mathbf{U}}_\mathrm{s}$ and $\hat{\mathbf{U}}_\mathrm{n}$ are the signal and noise eigenvectors, respectively, $\hat{\Lambda}_\mathrm{s}$ contains the largest eigenvalues corresponding to each of the M present signals, and $\hat{\Lambda}_\mathrm{n}$ contains the remaining eigenvalues corresponding to the noise subspace.

Due to the noise eigenvectors $\mathbf{U}_\mathrm{n}$ being orthogonal to $\mathbf{A}$, we have the useful property that

$$\mathbf{U}_\mathrm{n}^H\mathbf{a}(\boldsymbol{\theta}) = 0 \tag{3.2}$$

Finally, by estimating the orthogonal projection vector onto the noise subspace

$$\hat{\Pi}^\perp = \hat{\mathbf{U}}_\mathrm{n}\hat{\mathbf{U}}_\mathrm{n}^H, \tag{3.3}$$

the result in (3.2) allows for the definition of the MUSIC *spatial spectrum* as

$$P_{\mathrm{M}}(\boldsymbol{\theta}) = \frac{\mathbf{a}^H(\boldsymbol{\theta})\mathbf{a}(\boldsymbol{\theta})}{\mathbf{a}^H(\boldsymbol{\theta})\hat{\Pi}^{\perp}\mathbf{a}(\boldsymbol{\theta})} \tag{3.4}$$

that should have sharp peaks at the $\boldsymbol{\theta}$ values corresponding to the DOA of the sources.

3.1.2 Root-MUSIC

MUSIC has the benefit of being applicable to an arbitrary array geometry, but when applied to a ULA, the Root-MUSIC algorithm can perform better for small samples.

From the geometric structure of the Vandermonde matrix in (2.11), we can define polynomials in terms of the l-th eigenvector of the spatial covariance matrix as

$$p_{\mathrm{l}}(z) = \mathbf{u}_{\mathrm{l}}^H \mathbf{p}(z), \quad l = M+1, M+2, \ldots, L \tag{3.5}$$

where $\mathbf{p}(z)$ is a vector containing the roots of the polynomial,

$$\mathbf{p}(z) = \begin{bmatrix} 1, & z, & \ldots, & z^{L-1} \end{bmatrix}^T, \tag{3.6}$$

the polynomials $p_{\mathrm{l}}(z)$ will have zeros at $z = e^{j\theta_{\mathrm{m}}}$, where θ_{m} is the m-th DOA. So to find the DOAs, we can find the zeros of all polynomials and eigenvectors expressed as

$$\mathbf{p}^H(z)\hat{\mathbf{U}}_{\mathrm{n}}\hat{\mathbf{U}}_{\mathrm{n}}^H\mathbf{p}(z), \tag{3.7}$$

or limiting the search to poles on the unit circle, we have the 2(L - 1) degree polynomial

$$p(z) = z^{L-1}\mathbf{p}^T(z^{-1})\hat{\mathbf{U}}_{\mathrm{n}}\hat{\mathbf{U}}_{\mathrm{n}}^H\mathbf{p}(z) \tag{3.8}$$

Now, for the roots inside the unit circle, the M roots of largest magnitude will correspond to the DOAs of the present sources related by their argument as

$$\hat{\theta}_{\mathrm{m}} = \arccos\left(\frac{1}{kd}\arg\{\hat{z}_{\mathrm{m}}\}\right), \quad m = 1, 2, \ldots, M \tag{3.9}$$

3.1.3 ESPRIT

Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) is another subspace-based method that estimates the DOAs of the sources by exploiting the rotational invariance property of $\mathbf{A}_{\mathrm{ULA}}(\boldsymbol{\theta})$.

ESPRIT starts with splitting the steering vector into two sub-matrices $\mathbf{A_1}$ and $\mathbf{A_2}$, where the last and first rows are omitted, respectively. The two sub-matrices are then related by the diagonal matrix Φ with roots $e^{j\theta_{\mathrm{m}}}$ corresponding to the M DOAs on the diagonal,

$$\mathbf{A}_2 = \mathbf{A}_1\Phi \tag{3.10}$$

First, by expressing the identity matrix in (2.14) as $I = \mathbf{U}_{\mathrm{s}}\mathbf{U}_{\mathrm{s}}^H + \mathbf{U}_{\mathrm{n}}\mathbf{U}_{\mathrm{n}}^H$, the expressions for $\mathbf{R}$ in (2.14) and (3.1) can be combined to

$$\mathbf{A}\mathbf{P}\mathbf{A}^H + \sigma^2\mathbf{U}_{\mathrm{s}}\mathbf{U}_{\mathrm{s}}^H = \mathbf{U}_{\mathrm{s}}\Lambda_{\mathrm{s}}\mathbf{U}_{\mathrm{s}}^H \tag{3.11}$$

Multiplying both sides by $\mathbf{U}_s$ from the right and rearranging provides the representation of the signal subspace as

$$\mathbf{U}_s = \mathbf{A}\mathbf{P}\mathbf{A}^H\mathbf{U}_s(\Lambda_s - \sigma^2\mathbf{I})^{-1}, \tag{3.12}$$

or simplified to what will be the basis for ESPRIT and later WSF in Section 3.1.4,

$$\mathbf{U}_s = \mathbf{A}\mathbf{T} \tag{3.13}$$

where

$$\mathbf{T} = \mathbf{P}\mathbf{A}^H\mathbf{U}_s(\Lambda_s - \sigma^2\mathbf{I})^{-1} \tag{3.14}$$

With the two new sub-matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, (3.13) combined with (3.10) can be expressed as

$$\mathbf{U}_1 = \mathbf{A}_1\mathbf{T} = \mathbf{T}^{-1}\Phi\mathbf{T} \tag{3.15}$$

and

$$\mathbf{U}_2 = \mathbf{A}_2\mathbf{T} = \mathbf{A}_1\Phi\mathbf{T}, \tag{3.16}$$

ultimately providing the relation between $\mathbf{U}_1$ and $\mathbf{U}_2$ as

$$\mathbf{U}_2 = \mathbf{U}_1\Psi \tag{3.17}$$

where $\Psi = \Phi\mathbf{T}$ have the same eigenvalues as Φ and contain the DOA information.

The DOAs can now be found by first solving the relation between $\mathbf{U}_1$ and $\mathbf{U}_2$ as a least squares problem using Least Squares or Total Least Squares, and finally apply the same inversion function (3.9) as for Root-MUSIC to obtain the DOAs.

3.1.4 Root-WSF

Weighted Subspace Fitting (WSF) approaches the DOA estimation problem with a search to minimize the distance between an estimated signal subspace $\hat{\mathbf{U}}_s$ and $\mathbf{A}\mathbf{T}$. Using the sum of the squared Euclidian norms estimates for θ and $\mathbf{T}$ can be found by solving the separable linear squares problem

$$\{\hat{\theta}, \hat{\mathbf{T}}\} = \arg\min_{\theta,\mathbf{T}} \left\|\hat{\mathbf{U}}_s - \mathbf{A}\mathbf{T}\right\|^2, \tag{3.18}$$

or by expressing the parameter $\mathbf{T}$ as

$$\mathbf{T} = \mathbf{A}^H\hat{\mathbf{U}}_s \tag{3.19}$$

we can estimate the DOAs by finding

$$\hat{\theta} = \operatorname*{argmin}_{\theta} \operatorname{Tr}\left\{\Pi_A^{\perp}\hat{\mathbf{U}}_s\hat{\Lambda}_s\hat{\mathbf{U}}_s^H\right\} \tag{3.20}$$

where

$$\Pi_A^{\perp} = \mathbf{I} - \Pi_A = \mathbf{I} - \mathbf{A}\mathbf{A}^{\dagger} = \mathbf{I} - \mathbf{A}\left(\mathbf{A}^H\mathbf{A}\right)^{-1}\mathbf{A}^H \tag{3.21}$$

is the orthogonal projection onto the subspace of $\mathbf{A}^H$ and Tr is the trace operator summing the diagonal elements of the matrix. The expression in (3.20) is what the WSF algorithm aims to minimize by weighting the eigenvalues of $\hat{\Lambda}_s$ with the weights $\mathbf{W}$ so that

$$\hat{\theta} = \operatorname*{argmin}_{\theta} \operatorname{Tr}\left\{\Pi_A^{\perp}\hat{\mathbf{U}}_s\mathbf{W}\hat{\mathbf{U}}_s^H\right\} \tag{3.22}$$

The weights can be chosen so that $\mathbf{W}$ is a diagonal matrix containing the inverse of the covariance matrix of the asymptotically independent projected eigenvectors $\Pi_{\mathrm{A}}^{\perp}(\theta_0)\hat{\mathbf{u}}_{\mathrm{k}}$ for $k = 1, 2, \ldots, M$. By estimating values for $\hat{\Lambda}_{\mathrm{s}}$ and $\hat{\sigma}^2$, we define the weights as

$$\hat{\mathbf{W}} = \left(\hat{\Lambda}_{\mathrm{s}} - \hat{\sigma}^2\mathbf{I}\right)^2 \hat{\Lambda}_{\mathrm{s}}^{-1} \tag{3.23}$$

The root-variant of WSF takes a different approach to solving (3.22), utilizing the properties of the ULA structure characterized by (2.11). By defining a polynomial

$$b(z) = \prod_{m=1}^{M} \left(z - e^{j\theta_{\mathrm{m}}}\right) \tag{3.24}$$

of which the roots are at $e^{j\theta_{\mathrm{m}}}$ for $m = 1, 2, \ldots, M$, we can express the polynomial as

$$\mathbf{B}^H \mathbf{A}_{\mathrm{ULA}} = 0 \tag{3.25}$$

where

$$\mathbf{B}^H = \begin{bmatrix} b_{\mathrm{M}} & b_{\mathrm{M}-1} & \cdots & 1 & \cdots & 0 \\ & \ddots & \ddots & \ddots & \ddots & \\ 0 & & b_{\mathrm{M}} & b_{\mathrm{M}-1} & \cdots & 1 \end{bmatrix} \tag{3.26}$$

From this we can rewrite the projection matrix $\Pi_{\mathrm{A}}^{\perp}$ in (3.21) equivalently in terms of $\mathbf{B}$ as

$$\Pi_{\mathrm{A}}^{\perp} = \mathbf{B}(\mathbf{B}^H\mathbf{B})^{-1}\mathbf{B}^H \tag{3.27}$$

Building on the minimization problem central to DML [26] formulated as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \operatorname{Tr}\left\{\Pi_{\mathrm{A}}^{\perp}\hat{\mathbf{R}}\right\} \tag{3.28}$$

where $\hat{\mathbf{R}}$ is the sample covariance matrix, we can instead estimate the polynomial coefficients of (3.24) through $\mathbf{B}$ so that

$$\hat{\mathbf{b}} = \underset{\mathrm{b}}{\operatorname{argmin}} \operatorname{Tr}\left\{\mathbf{B}(\mathbf{B}^H\mathbf{B})^{-1}\mathbf{B}^H\hat{\mathbf{R}}\right\} \tag{3.29}$$

which then can be used to estimate $\boldsymbol{\theta}$ using the same method as for Root-MUSIC in (3.9). Root-WSF limits the search for $\hat{\mathbf{b}}$ through using $\hat{\mathbf{U}}_{\mathrm{s}}\mathbf{W}\hat{\mathbf{U}}_{\mathrm{s}}^H$ instead of $\hat{\mathbf{R}}$ as in (3.22) to two iterations by first solving

$$\hat{\mathbf{b}} = \underset{\mathrm{b}}{\operatorname{argmin}} \operatorname{Tr}\left\{\mathbf{B}\mathbf{B}^H\hat{\mathbf{U}}_{\mathrm{s}}\mathbf{W}\hat{\mathbf{U}}_{\mathrm{s}}^H\right\} \tag{3.30}$$

followed by solving

$$\hat{\mathbf{b}} = \underset{\mathrm{b}}{\operatorname{argmin}} \operatorname{Tr}\left\{\mathbf{B}(\hat{\mathbf{B}}^H\hat{\mathbf{B}})^{-1}\mathbf{B}\hat{\mathbf{U}}_{\mathrm{s}}\mathbf{W}\hat{\mathbf{U}}_{\mathrm{s}}^H\right\} \tag{3.31}$$

and finally etimate the DOAs using the roots of $\hat{b}(z)$ in (3.9).

3.2 Forward-Backward Spatial Smoothing

In situations where the sources of which the DOA is to be estimated are correlated or coherent, a method termed *spatial smoothing* can be used to improve the estimation accuracy. This well described by Pillai and Kwon in [27]. Coherent signals can be represented as scaled and time-shifted versions of each other, which can be described as

$$s_{\mathrm{m}}(t) = \alpha_{\mathrm{m}} s_1(t), \quad m = 1, 2, \ldots, M \tag{3.32}$$

where α_{l} is the complex factor containing the attenuation and phase shift of the m-th signal relative to the first signal $s_1(t)$. This relationship between signals reduces the source covariance matrix $\mathbf{P}$ to

$$\mathbf{P} = \alpha\alpha^H, \quad \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_{\mathrm{M}}]^T \tag{3.33}$$

where α is a vector containing the complex factors α_{m}. Subsequently, the rank of the array output vector $\mathbf{R}$ is also reduced, leading to the fact that (3.2) no longer holds.

By applying forward/backward averaging techniques, a ULA of $3M/2$ sensors is sufficient to estimate the DOAs for M coherent sources [27]. For the forward spatial smoothing, the original array of length L is first divided into K subarrays of length L' where the l-th subarray is given by

$$\mathbf{x}_{\mathrm{k}}^{\mathrm{f}}(t) = [x_{\mathrm{k}}(t), x_{\mathrm{k+1}}(t), \ldots, x_{\mathrm{k+L'-1}}(t)]^T \tag{3.34}$$

The spatial covariance matrix of the k-th forward subarray is then given by

$$\mathbf{R}_{\mathrm{k}}^{\mathrm{f}} = \mathbb{E}\left\{\mathbf{x}_{\mathrm{k}}^{\mathrm{f}}(t)\mathbf{x}_{\mathrm{k}}^{\mathrm{f}H}(t)\right\} \tag{3.35}$$

and the forward spatially smoothed covariance matrix is then defined as

$$\mathbf{R}^{\mathrm{f}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{R}_{\mathrm{k}}^{\mathrm{f}} \tag{3.36}$$

For the backward spatial smoothing, the original array is instead divided into K subarrays from the back, where the k-th backward subarray is given by the complex conjugate of the elements so that

$$\mathbf{x}_{\mathrm{k}}^{\mathrm{b}}(t) = [\bar{x}_{\mathrm{L-k+1}}(t), \bar{x}_{\mathrm{L-k}}(t), \ldots, \bar{x}_{\mathrm{K-k+L'}}(t)]^T \tag{3.37}$$

and then, as for the forward spatial smoothing, the backward spatial covariance matrix for the k-th subarray is

$$\mathbf{R}_{\mathrm{k}}^{\mathrm{b}} = \mathbb{E}\left\{\mathbf{x}_{\mathrm{k}}^{\mathrm{b}}(t)\mathbf{x}_{\mathrm{k}}^{bH}(t)\right\} \tag{3.38}$$

leading to the backward spatially smoothed covariance matrix

$$\mathbf{R}^{\mathrm{b}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{R}_{\mathrm{k}}^{\mathrm{b}} \tag{3.39}$$

Finally, (3.36) and (3.39) are combined to form the forward/backward spatially smoothed covariance matrix

$$\mathbf{R}^{\mathrm{fb}} = \frac{1}{2}\left(\mathbf{R}^{\mathrm{f}} + \mathbf{R}^{\mathrm{b}}\right) \tag{3.40}$$

3.3 Array Configurations

The first design choice to be made should be the geometry of the sensor array. Although array processing methods can be applied to any structure of sensors, the ones considered for this project are the uniform linear array (ULA) and the uniform circular array (UCA). A ULA is a linear array of sensors, i.e. sensors placed in a straight line, distributed with a uniform spacing between them. A UCA is a circular array of sensors, also distributed with a uniform spacing between each sensor element.

The ULA is perhaps the simplest form of array geometry and has many benefits with regards to implementation and computation time, as discussed in Section 3.1. While UCA has the benefit of being able to estimate the DOA of a signal from any direction in a 360° field of view, the ULA is limited to only 180°, and waves arriving from opposite directions of the array broadside will be indistinguishable. According to research done by Vesa and Simu, the performance of a ULA is significantly better than that of a UCA in similar conditions and with the same number of sensors when applying the MUSIC algorithm [28]. They also show that the performance of the UCA increases significantly with the circle's radius.

Although a comparison between a ULA and a UCA would be highly interesting for this project, the UCA is not prioritized due to its increased complexity and the time constraints of the project. Both array geometries have advantages and disadvantages and will ultimately also depend on the desired shape of the device.

All methods considered in this project require the arriving waves at the array to be approximated as plane waves. To ensure this, the array needs to be further away than the far-field limit of the array. For ULA, this limit is often approximated as

$$r_{\text{farfield}} \geq \frac{2D^2}{\lambda} \tag{3.41}$$

where D is the array aperture and λ is the wavelength of the wave [29]. Assuming that the ULA design meets the requirement of $d \leq \lambda/2$, the far-field limit becomes a function depending on the number of sensors and the wavelength of the signal so that

$$r_{\text{farfield}} \geq \frac{2((L-1)(\lambda/2))^2}{\lambda} = \frac{(L-1)^2\lambda}{2} \tag{3.42}$$

where L is the number of sensor elements. This relation is illustrated in Figure 3.1. By choosing a minimum distance of which the wave source can be from the array, the maximum number of sensors spaced at $\lambda/2$ can be calculated. This means that if we can place sensors accurately 1 cm apart and it should be possible to detect a touch at a distance of 40 cm, we can measure wavelengths down to 20 mm and then use no more than seven sensors.

3.4 Touch Localization

In this section, we will first present a relatively simple design of a trigonometry-based touch localization method in Section 3.4.1. The design is intended to be easy to implement and understand and will be used as a starting point for further development. This is the design that will be used for the tests in Section 5.2. Then a different but interesting design will be

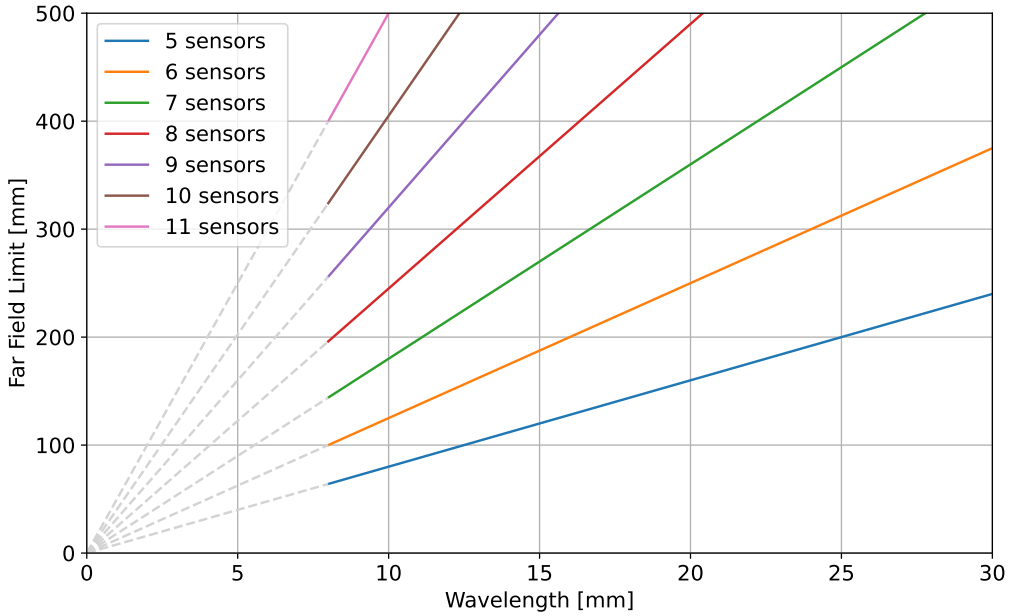


Figure 3.1: Deciding on a required far-field distance allows for choosing the desired number of sensors, given that the spacing between sensors is half the wavelength of the measured signal. Wavelengths below 8 mm are not considered to be relevant due to the physical size of the sensors.

discussed briefly in Section 3.4.2, which is based on using a transmitted pulse to locate a touch. Such a design would be inspired by typical radar systems, using the scattered wave from a finger held to the plate surface for localization.

3.4.1 Trigonometric Localization

Focusing on the purpose of testing the DOA estimation methods described in Section 3.1, we want to design a touch localization method that

- relies on the estimated DOAs for the localization,
- allows for identifying which waves the estimated DOAs correspond to.

The proposed method is illustrated in Figure 3.2, where the ULA is assumed to be placed a known distance from the corner in the bottom left of the plate. Ideally, any touch within some area close to the middle of the plate and downwards should ensure that the first four wave arrivals are the ones shown with the mirrored sources.

Note that a ULA is limited to the range $\{-90°, 90°\}$ and thus can not distinguish signals coming from the left or right. Therefore, the ULA will see the arriving signals as illustrated in Figure 3.3, all coming from the same side. For a touch above the center of the ULA, the geometry of the setups makes sure that the order of the angles is preserved as

$$\theta_2 < \theta_4 < \theta_3 < \theta_1 \tag{3.43}$$

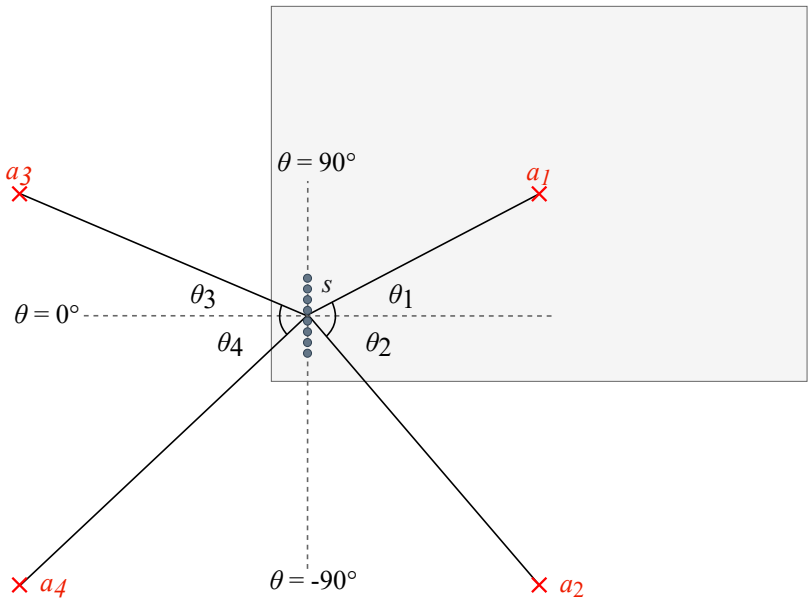


Figure 3.2: The expected DOAs of the first four arrivals at the array from a source in the middle of the plate. The source location a_1 is now estimated using the angles θ_1, θ_2, and θ_3.

By defining the distance vector from the middle of the sensors s to the source location a as $\vec{r}_{\mathrm{s,a}} = [x_{\mathrm{s,a}}, y_{\mathrm{s,a}}]$ and the distance vector from s to the closest corner c as $\vec{r}_{\mathrm{c,s}} = [x_{\mathrm{c,s}}, y_{\mathrm{c,s}}]$, the properties of the tangent function can be used to express the first three angles as

$$\tan\theta_1 = \frac{y_{\mathrm{s,a}}}{x_{\mathrm{s,a}}} \tag{3.44a}$$

$$\tan\theta_2 = \frac{-y_{\mathrm{s,a}} - 2y_{\mathrm{c,s}}}{x_{\mathrm{s,a}}} \tag{3.44b}$$

$$\tan\theta_3 = \frac{y_{\mathrm{s,a}}}{x_{\mathrm{s,a}} + 2x_{\mathrm{c,s}}} \tag{3.44c}$$

Splitting up (3.44b) and inserting (3.44a), we get

$$\begin{aligned} \tan\theta_2 &= \frac{-y_{\mathrm{s,a}} - 2y_{\mathrm{c,s}}}{x_{\mathrm{s,a}}} \\ &= \frac{-y_{\mathrm{s,a}}}{x_{\mathrm{s,a}}} - \frac{2y_{\mathrm{c,s}}}{x_{\mathrm{s,a}}} \\ &= -\tan\theta_1 - \frac{2y_{\mathrm{c,s}}}{x_{\mathrm{s,a}}} \end{aligned} \tag{3.45}$$

which yields

$$x_{\mathrm{s,a}} = -\frac{2y_{\mathrm{c,s}}}{\tan\theta_1 + \tan\theta_2} \tag{3.46}$$

By rewriting (3.44c) as a function of $y_{\mathrm{s,a}}$ so that

$$y_{\mathrm{s,a}} = \tan\theta_3(x_{\mathrm{s,a}} + 2x_{\mathrm{c,s}}) \tag{3.47}$$

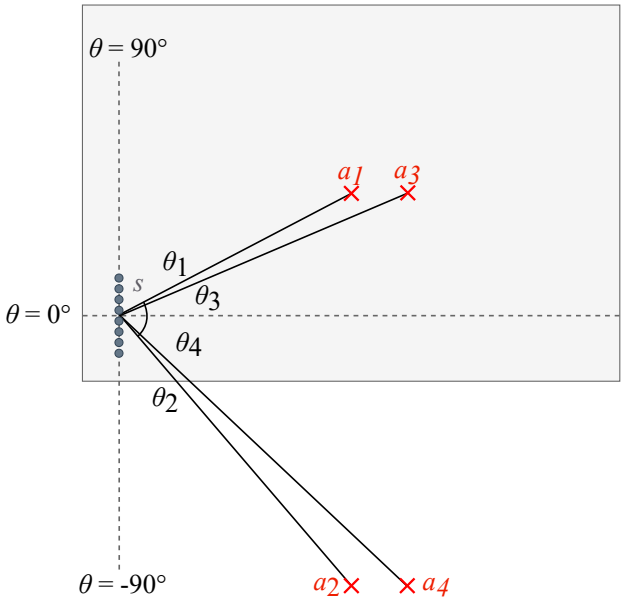


Figure 3.3: The ULA will not be able to distinguish angles arriving from left or right, meaning that Figure 3.2 can equivalently be described with all sources to the right.

and inserting (3.46), the distance $x_{s,a}$ can be expressed as

$$y_{s,a} = \tan\theta_3 \left(-\frac{2y_{c,s}}{\tan\theta_1 + \tan\theta_2} + 2x_{c,s} \right) \tag{3.48}$$

resulting in the $\vec{r}_{s,a}$ to be defined using the estimated DOAs and $\vec{r}_{c,s}$, ultimately allowing the touch location to be expressed as

$$\vec{r}_{s,a} = \left[-\frac{2y_{c,s}}{\tan\theta_1 + \tan\theta_2}, \quad -\tan\theta_3 \left(\frac{2y_{c,s}}{\tan\theta_1 + \tan\theta_2} - 2x_{c,s} \right) \right], \quad \{|\theta_1|, |\theta_2|\} \neq 90° \tag{3.49}$$

Under the assumption that the $r_{c,s}$ is known, the absolute coordinates of the touch on a plate

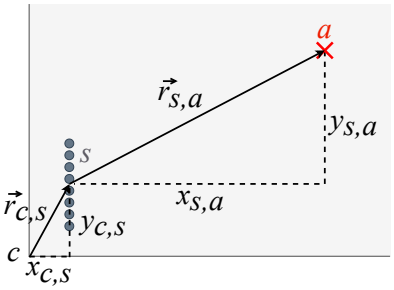


Figure 3.4: The definition of the distance vectors used. The distance vector from the middle of the sensors s to the source location a is defined as $\vec{r}_{s,a} = [x_{s,a}, y_{s,a}]$ and the distance vector from the closest corner $c = [0,0]$ to s is defined as $\vec{r}_{c,s} = [x_{c,s}, y_{c,s}]$.

with $[x, y] = [0, 0]$ in the bottom left corner can be expressed as

$$\vec{r}_{\mathrm{a}} = \vec{r}_{\mathrm{c,s}} + \vec{r}_{\mathrm{s,a}} \tag{3.50}$$

An important exception is the case where a is located directly above or below the ULA, i.e. $|\theta_1| = |\theta_2| = 90°$ so that the tangent function is undefined. Then, both fractions in (3.49) should be set to 0 and the distance vector should be expressed as

$$\vec{r}_{\mathrm{s,a}} = \begin{bmatrix} 0, & 2x_{\mathrm{c,s}} \tan\theta_3 \end{bmatrix}, \quad \{|\theta_1|, |\theta_2|\} = 90° \tag{3.51}$$

From the nature of mirrored sources and the geometry of the setup, the physical limitations can be summed up as

$$\begin{aligned} -90° \leq &\theta_1 \leq 90° \\ 0° < &\theta_2 \leq 90° \\ -90° < &\theta_3 < 90° \\ 0° < &\theta_4 < 90° \end{aligned} \tag{3.52}$$

It is worth noting that having the ULA placed in the bottom left corner, it is, in theory, possible to have a useful touch area down to $y = 0$ while still being able to separate the four signals from the rest of the reflections. In this case, i.e. $y_{\mathrm{s,a}} < 0$, an assumption can still be made that the angles will be in a distinct order when sorted, but the order will be different from the one presented in (3.43). This scenario can conveniently be detected by the fact that all angles will be negative, and the order will be

$$\theta_2 < \theta_4 < \theta_1 < \theta_3 < 0° \tag{3.53}$$

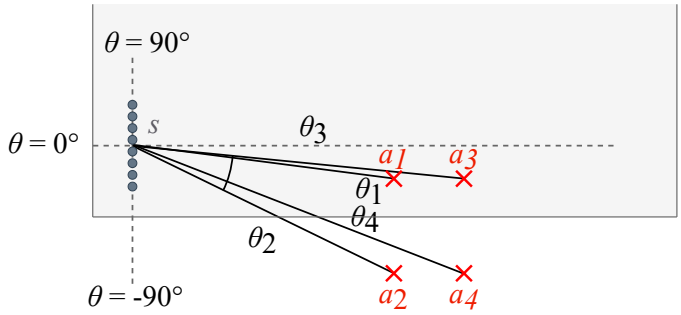


Figure 3.5: If the touch location is below the center of the ULA, all four angles will be negative and the order will be $\theta_2 < \theta_4 < \theta_1 < \theta_3$.

As θ_4 is also estimated, it should be possible to incorporate it into an expression for $\vec{r}_{\mathrm{s,a}}$. However, as the angle estimations are independent in the sense that an error in one estimated angle will not necessarily be corrected by another estimated angle, it will not be investigated further how this can be done to increase the accuracy of the touch location estimate.

If it should be the case that only θ_1 and θ_2 are detected, the touch location can still be estimated by a slight modification to $r_{\mathrm{s,a}}$. By inserting (3.46) into (3.44a) instead of (3.44c), we get

$$y_{\mathrm{s,a}} = -\frac{2y_{\mathrm{c,s}} \tan\theta_1}{\tan\theta_1 + \tan\theta_2} \tag{3.54}$$

and calculate the touch location to be

$$\vec{r}_{\mathrm{s,a}} = \left[-\frac{2y_{\mathrm{c,s}}}{\tan\theta_1 + \tan\theta_2}, \quad -\frac{2y_{\mathrm{c,s}} \tan\theta_1}{\tan\theta_1 + \tan\theta_2} \right], \quad \{|\theta_1|, |\theta_2|\} \neq 90^\circ \tag{3.55}$$

3.4.2 Active Transmission of a Pulse

An alternative to the method described in Section 3.4.1 is to use an active element to transmit a pulse from the touch sensing device and use the time delay and the arrival angle of the reflected pulse from the finger to localize the touch. The concept of using surface acoustic waves (SAWs) was considered by Adler before his final SAW-absorption design, but it was discarded due to difficulties in detecting the reflected waves from a finger [2].

However, a promising feasibility study was conducted by Kang *et al.* in [30] as recent as in 2022, where Lamb waves were transmitted from sensors placed on in the corners and middle of the edges around an $800 \times 600 \times 3$ mm glass plate, and then received by a sensor close by each actuator. They report that from a transmitted pulse at 380 kHz, the fingertip reflection amplitude is 1.23% and 4.81% compared to the short-side and long-side reflections, respectively.

This suggests that a viable approach using the reflected Lamb waves from a finger may be used for touch localization, perhaps even in a single-enclosure design. By transmitting a pulse from within the device, for instance, at the end of a ULA or in the middle of a UCA, the round-trip time and the DOA of the direct way should contain enough information for the touch localization.

As discussed by Kang *et al.*, the main challenge is to detect the finger-reflected pulse between the edge reflections and reverberations. Similar but less thorough measurements done in an unpublished prestudy to this thesis [31] concluded that the reflected signal of interest was too weak to be reliably detected, which is why this design was not chosen for this project.

Some ideas for improvements have been considered during the prestudy and during this project. Using a sensor array placed along one edge of a plate and transmitting a pulse from nearby the array, the measurements will contain a combination of the immediately transmitted pulse, the reflected pulse from the edges and the finger, and the waves generated by the touch itself. If the array can be directed towards the touch location using the direct wave of the touch impact through beamforming techniques, the following pulse reflection from the finger could appear stronger compared to the clutter. Another technique that can be worth looking into is the focused MUSIC algorithm researched by Xu *et al.* in [32] from 2022, which is originally intended for use on AE Lamb waves. This method includes a transmitting transducer and accounts for effects such as dispersion and near-field, both of which are highly relevant. However, the MUSIC may have issues with the correlated signals arriving from edge reflections.

Regardless of the final touch localization design, including an active element in the device can also be beneficial for other reasons. Depending on the design, it may be useful or even essential to have a good model of the plate dimensions, which a transmitted pulse could help do in a more controlled manner than relying on an external source. The transmitted pulse could also help estimate properties such as the phase and group velocity of the waves, for instance, by placing the actuator at the end of a ULA.

3.4.3 Plate Assumptions

In order for the suggested method to work, some assumptions are necessary to make about the system. Firstly, it is assumed that the plate is homogeneous and isotropic, meaning that the material properties are the same throughout the plate and that the material properties are the same in all directions. The plate is chosen to be close to ideal for the tests in this project. In a practical application, however, the materials of the plate can vary greatly, and plates are often made of a composite of materials, e.g. a chipboard core with a laminate surface. For the design suggested in Section 3.3, the plate geometry is not of much relevance other than having one 90° corner and enough surface area to enable a valid touch area in the far-field of the array while still staying clear of the non-interesting edge reflections.

As for the type of waves propagating in the plate, the tests are performed with the assumption that in a time window where multiple Lamb wave modes are potentially present, all other modes are negligible compared to the A0 mode with regards to the DOA estimation. On the topic of the acoustics of the plate, neither the effect that an edge reflection has on a wave, the effect of different edge geometries, nor the effect of plate mountings are considered. An implementation on a table would likely have some interference from table legs in each corner and potential other mounting points on the plate bottom. We assume total reflection, and that there are no significant phase disturbances introduced by the reflection.

Chapter 4

Experiments and Implementation

The methods presented in Chapter 3 are evaluated on two sets of data: simulations generated in COMSOL Multiphysics described in Section 4.1 and measurements performed on a test setup described in Section 4.2. For the measurements, one setup is described in Section 4.2.1 used for inspecting a touch signal at its source, and another setup is described in Section 4.2.2 used for evaluating the touch localization method. Relevant code implementations are presented in Section 4.3 to Section 4.5, with code implementations attached in Appendix C.

4.1 COMSOL Simulations

Starting out, we want to test the methods from Chapter 3 on a COMSOL Multiphysics simulation of a plate where we know the properties of the plate and the measurements are performed by ideal sensors. To match the plate used in the experiments described in Section 4.2, we use a $1000 \times 700 \times 10$ mm plate made of Polytetrafluoroethylene, more commonly known as Teflon, modeled with properties as listed in Table 4.1 [33].

Table 4.1: Properties of the 10 mm Teflon plate used in the COMSOL simulations [33].

Property	**Theoretical Value**
Density (kg/m^3)	2200
Longitudinal wave speed (m/s)	1400
Shear wave speed (m/s)	550
Poisson's ratio (-)	0.4
Young's modulus (GPa)	1.89
Plate wave speed (m/s)	1011

To simulate the touch impact, a short Gaussian modulated pulse with center frequency $f_0 = 25$ kHz is transmitted from the center of the plate at $[x, y] = [50\,\text{cm}, 35\,\text{cm}]$, and the symmetry of the touch location on the plate is used to reduce the simulation time. The touch

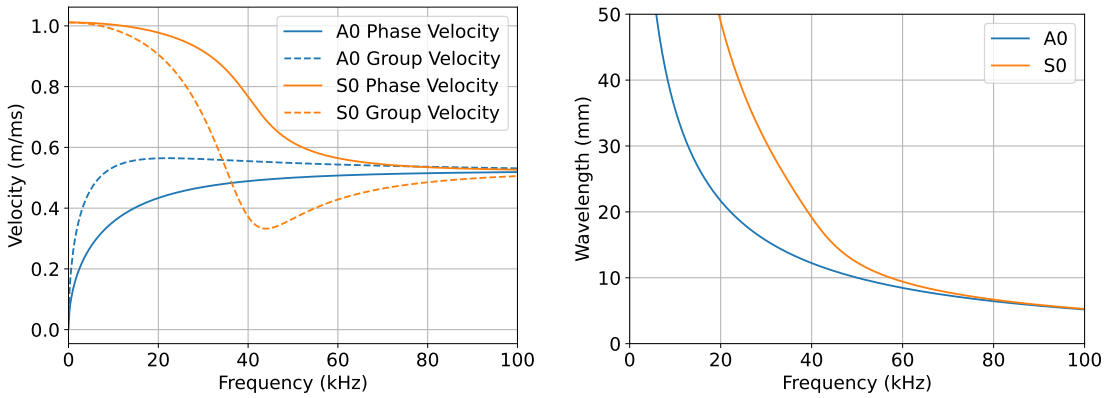
signal is modeled as two periods of the center frequency, expressed as

$$-\exp\left(-\frac{(t-t_{\text{pulse}}/2)^2}{2t_{\text{var}}}\right)\sin(2\pi f_0(t-t_{\text{pulse}}/2)) \tag{4.1}$$

with pulse length $t_{\text{pulse}} = 4/f_0 = 1.6 \cdot 10^{-4}\,\text{s}$, bandwidth $B = 0.7$, frequency variance $f_{\text{var}} = -(Bf_0)^2/(8\ln(0.5)) = 5.5228{\cdot}10^7\,\text{s}^{-2}$ and time variance $t_{\text{var}} = 1/(4\pi^2 f_{\text{var}}) = 4.5865{\cdot}10^{-10}\,\text{s}^2$. The simulation is sampled at 500 kHz. Figure A.3 in Appendix A shows the acceleration of the plate in the z-direction at $t = 600\,\mu s$. Due to long processing times, the simulation is only run for 2 ms

The measuring points, which can be seen as ideal sensors, are placed in a line along $x = 0.05\,\text{m}$ from $y = 0.05\,\text{m}$ to $y = 0.11\,\text{m}$, with a spacing of 10 mm providing a total of seven sensors as is estimated in Section 3.3 to be appropriate.

Since we know the plate properties in the simulations, we can also calculate the phase and group velocities of the Lamb waves using the Dispersion Calculator software from enter for Lightweight Production Technology [13]. The phase and group velocities for the A0 and S0 wave modes are shown in Figure 4.1a, and their corresponding wavelengths are shown in Figure 4.1b.



(a) Simulated phase and group velocities.

(b) Simulated wavelengths.

Figure 4.1: Simulated phase and group velocities for the A0 and S0 modes for a 10 mm Teflon plate, as well as the wavelengths.

As we want the wavelength to be $\lambda \geq 2d = 2\,\text{cm}$, we expect $f_c = 22\,\text{kHz}$ to be the optimal center frequency with $\lambda = 2.01\,\text{cm}$ and phase velocity $v_{\text{p,A0}} = 442.7\,\text{m/s}$. At this frequency, the group velocity is expected to be $v_{\text{e,A0}} = 564.4\,\text{m/s}$, so that the first S0 waves propagating at a group velocity of $v_{\text{e,S0}} = 877.8\,\text{m/s}$ should be somewhat separable. An interesting aspect that will not be investigated further is how the gap in phase velocities affects the accuracy of the DOA estimation, as $v_{\text{p,S0}} = 969.3\,\text{m/s}$ for the same frequency.

In Figure 4.2, the reasoning for choosing to estimate the DOA for the direct A0 wave and the first three A0 reflections is illustrated. Using $v_{\text{e,A0}} = 564.4\text{m/s}$ and the calculated mirrored sources described in Section 4.5, the expected arrival times for all waves that reflect once and

twice from the edges before reaching the seven sensors in the array are calculated. The blue line at 0 s is when the touch impact occurs, first followed by three groupings of waves around 1 ms where the middle group consists of waves arriving at two different angles. These are the four waves of which we want to focus on in this project, as they are clearly separable from the next group of waves arriving around 2 ms. Although the shown example is for a touch in the middle of the plate, this should provide a decent usable area of the plate, limited only by how close the touch is to the upper and right edges.

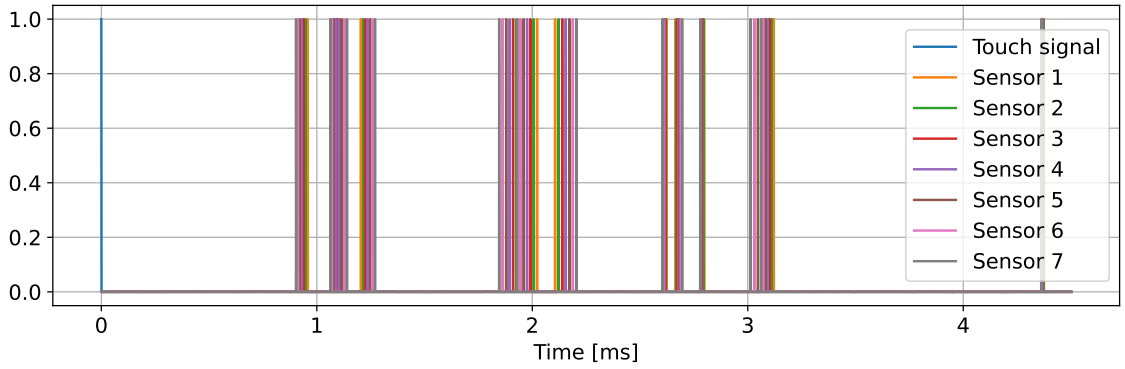


Figure 4.2: Expected arrival times for the direct wave and first and second reflections in the COMSOL simulations when a touch is located in the middle fo the plate. Calculated using mirrored sources described in Section 4.5 and group velocity $v_{e,A0} = 564.4\,\mathrm{m/s}$.

For comparison with the COMSOL-simulated or the physical measurements, the same procedure is performed but with modeling the waves as a short Gaussian pulse instead of the Dirac delta function. This generates a simple "ideal" signal that does not take into account the phase velocity and dispersion, or any other properties beyond shifting a pulse by a calculated amount of time the wave have traveled. By aligning and normalizing the ideal signals to match the first pulse of "Sensor 1" and empirically fitting the attenuation per meter, this provides a helpful comparison with measurements when overlaying the Hilbert envelopes of the modeled and measured signals.

4.2 Experiment Setup and Measurements

For the tests performed using an electronic actuator as the source, a signal is generated by an Agilent 33500B Series Waveform Generator, shown in Figure 4.3a, and transmitted into a $1000 \times 700 \times 10\,\mathrm{mm}$ Teflon plate using a $3 \times 3 \times 2\,\mathrm{mm}$ PA3JEAW piezo chip for the setup in Section 4.2.1, and a circular $\varnothing 5 \times 2\,\mathrm{mm}$ PA25FEW piezo chip for the setup in Section 4.2.2. Both the actuator and all sensors are attached to the plate using beeswax, as shown in Figure 4.4b.

The accelerometers used are from Brüel & Kjær (B&K), which have properties as summarized in Table 4.2 and the full specifications are available in Appendix A. The serial number for each sensor is kept consistent throughout the project, and for setups with more than three sensors, "Sensor 4" is the same model as "Sensor 1", and so on. A notable discrepancy is "Sensor 3", where the serial number on the sensor does not match the one in the attached documenta-

(a) The signal generator. **(b)** The amplifier. **(c)** The computer and ADC.

Figure 4.3: The signal generator, amplifier and the combined computer and ADC used for the physical tests.

tion. However, the sensor is of the same type as "Sensor 1" and the one in the documentation, which have approximately the same specifications and will therefore be assumed to be close enough for the purpose of this project.

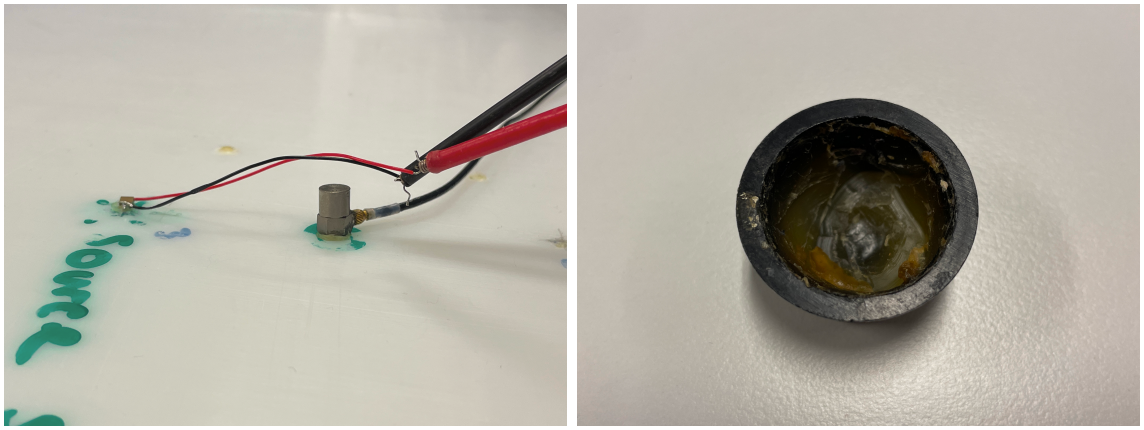
Table 4.2: The specifications of the sensors used for the physical tests. The full specifications are available in Appendix A.

	Sensor 1	**Sensor 2**	**Sensor 3**
Accelerometer (B&K Type)	4344	4393	4344
Serial number	447546	1434348	610404
Flat frequency response (kHz)	≤ 45	≤ 15	≤ 45
Sensitivity $\left(\mathrm{mV/ms^{-2}}\right)$	0.293 at 160 Hz	0.489 at 159.2 Hz	0.305 at 160 Hz

The sensors signals are amplified by a B&K NEXUS Conditioning Amplifier [1], shown in Figure 4.3b. In all measurements, the voltage sensitivity is set to $100\,\mathrm{mV/ms^{-2}}$ with the purpose of increasing the signal-to-noise ratio (SNR) with regards to the noise originating between the amplifier and the analog-to-digital converter (ADC). To acheive correct values for the acceleration, the amplitudes are corrected for in the software according to their specifications. Additionally, when bandpassing a signal to a frequency above 15 kHz, the amplitude of the signal on Sensor 2 is reduced according to the frequency response in the specifications.

The ADC is a National Instruments NI PXI-6733 set to a sampling rate of 150 kHz, which is controlled by a National Instruments NI PXIe-8135 Embedded Controller, both shown in Figure 4.3c. When using an actuator for the tests, the generated signal is split so that it is both transmitted into the plate and sampled by the ADC, and a sync signal is sent on a separate channel directly to the ADC.

[1] The power supply for the amplifier stopped working during the project, and is therefore powered by a Mascot Type 710 power supply. See Appendix A for a picture.



(a) The actuator and a sensor. **(b)** The beeswax used for mounting.

Figure 4.4: The actuator and a sensor mounted on the plate, in this case for the setup described in Section 4.2.1, and the beeswax used to attach them.

4.2.1 Setup for Measuring a Touch Signal at the Source

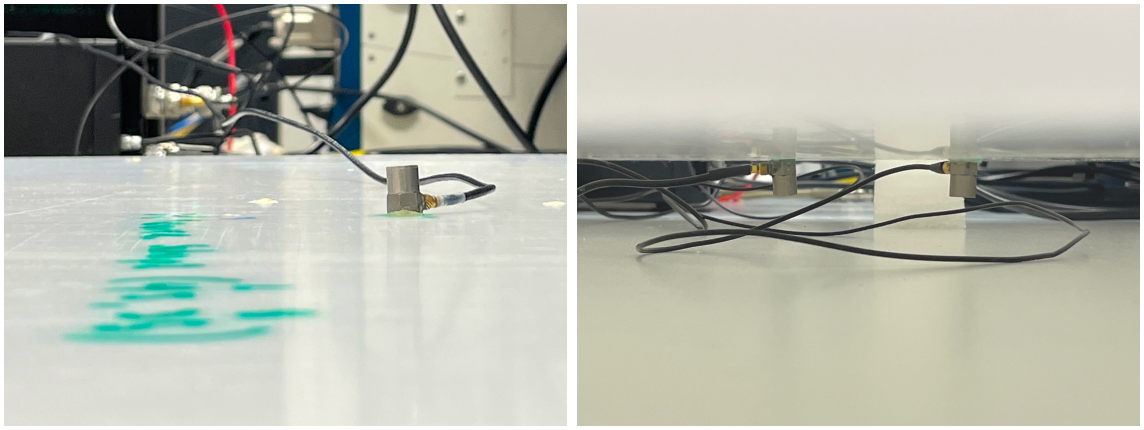
In order to better understand the signals produced by a touch on the plate, we use a setup as shown in Figure 4.5. The setup consists of three sensors, where "Sensor 1" is placed directly under the touch location on the bottom side of the plate, and "Sensor 2" and "Sensor 3" are placed 5 cm from the source on both the top and bottom side of the plate, respectively. All elements are placed close to the middle of the plate to best separate the direct wave from reflections.

The intention of the placement of "Sensor 2" and "Sensor 3" is two-fold. First, we want to see how the plate affects the propagating waves in terms of attenuation, dispersion and reflections compared to the signal at the source. Then we want to exploit the symmetric and antisymmetric properties of the S0 and A0 wave modes and highlight them individually by adding and subtracting the two signals.

4.2.2 Setup with a ULA in the Corner

For testing the DOA experiments tested in the simulations described in Section 4.1, we use an experiment setup as in Figure 4.7. The sensors are placed in the bottom left corner of the plate, starting with "Sensor 1" at $[x, y] = [5\,\text{cm}, 5\,\text{cm}]$ and ending with "Sensor 7" at $[x, y] = [5\,\text{cm}, 11\,\text{cm}]$. This gives a distance between the array elements of $d = 1\,\text{cm}$ chosen because of physical limitations, and a total length of $L = 6\,\text{cm}$. Recall from Section 3.3 that seven sensor elements are the highest possible with $d = 1\,\text{cm}$ to stay outside the far-field limit when the source is around the middle of the plate.

For verification of the results and to test the robustness of the design, four different source locations are used: $[x, y] = [50\,\text{cm}, 35\,\text{cm}]$, $[45\,\text{cm}, 40\,\text{cm}]$, $[55\,\text{cm}, 40\,\text{cm}]$, $[50\,\text{cm}, 20\,\text{cm}]$, illustrated in Figure 4.6. The motivation for these locations is to have one group of relatively closely spaced source locations close to the middle of the plate, and to see the performance of the design when the sources move closer to the bottom edge and closer to $\theta_1 = \theta_3 = 0$ and θ_2



(a) The touch location and “Sensor 2”. **(b)** “Sensor 1” and “Sensor 3”.

Figure 4.5: The setup used to measure properties of the touch signal. The touch is located at a green mark 5 cm to left of “Sensor 2”.

gets closer to θ_1.

Since the experiments are limited to three accelerometers, the accelerometers and the actuator are moved around in order to “simulate” a larger array. This is done by having the three accelerometers first placed at the positions of “Sensor 1”, “Sensor 2”, and “Sensor 3”, then transmitting the pulse from all nine actuator positions, then repeating with the sensors placed in the positions of “Sensor 4”, “Sensor 5”, and “Sensor 6”, and finally “Sensor 7” and “Sensor 8”. “Sensor 8” is omitted for all the results in Chapter 5 due to the theoretical far-field limit discussed in Section 3.3.

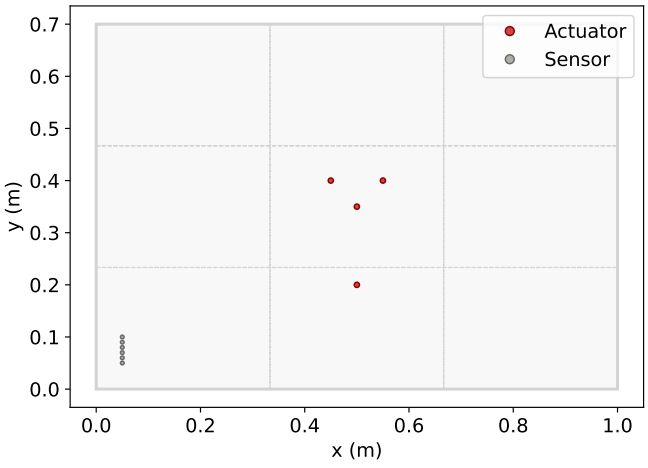


Figure 4.6: An illustration of the sensors and the different actuator placements used for the ULA in the experiments.

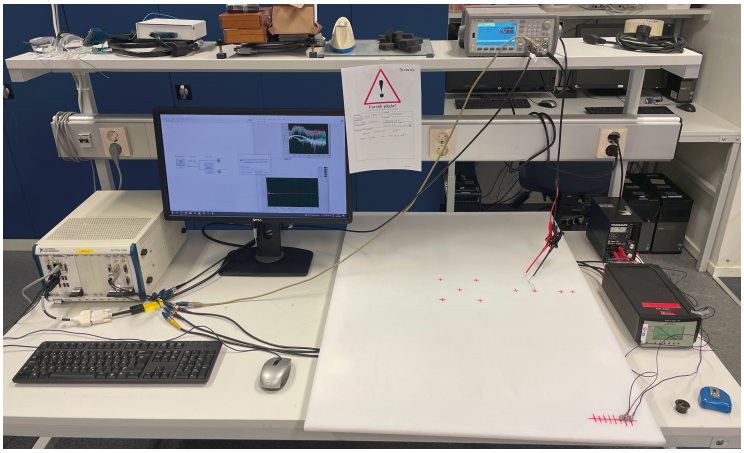


Figure 4.7: The full setup when testing DOA estimation methods on a ULA of up to 8 sensors.

4.3 Signal Preprocessing

Some preprocessing steps are required to prepare the received signals for the DOA estimation methods. The signals are first cropped to the desired time window, which is straightforward and thus not described in any detail here. Next, the signals are filtered as described in Section 4.3.1 to achieve the necessary narrow bandwidth. As the measurements require some time shifting to achieve their intended relative time delays, the signals are first interpolated as described in Section 4.3.2 for higher time resolution before being aligned as described in Section 4.3.3.

4.3.1 Filtering

As all DOA estimation methods require a narrowband signal and a constant phase velocity to work with, all input signals are filtered before any array processing is done. The implementation of the highpass, lowpass and bandpass filters is shown in Code listing C.1 in Appendix C. Unless stated otherwise, the filters are implemented as second-order filters with a quality factor of $q = 0.05$ for the bandpass filters, meaning that the passband is $f_c \pm 0.05 f_c$, where f_c is the critical frequency.

All filters are implemented as Butterworth filters due to their flat frequency response in the passband, using the `scipy.signal.butter` function in Python. The output of the filter is set to second-order sections, as this is recommended for general-purpose filtering [34]. As the measurements from the accelerometers contain a relatively high amount of noise around 50 Hz and we are not interested in frequencies in this range, a highpass filter is first applied with a cutoff frequency of 250 Hz. A lowpass filter is also applied with cutoff at 50 kHz as this is the upper limit of the accelerometers used for the measurements. In order to avoid phase shifting, the signal is filtered in first forwards then backward using the `scipy.signal.sosfiltfilt` function [35].

4.3.2 Interpolation

Although the sampling rate used in the ADC is high enough to avoid aliasing for the frequency range of the accelerometers according to the Nyquist-Shannon sampling theorem, we would still prefer to have it higher. This allows for higher precision when shifting signals as described in Section 4.3.3, makes the signals easier to interpret visually, and should make sure that the DOA estimation methods are not limited by the sampling rate.

The interpolation is done using the `scipy.interpolate.interp1d` function in Python, using a cubic interpolation function [36] as described in Code listing C.2 in Appendix C. The interpolation factor is set by the global variable `INTERPOLATION_FACTOR`, which is adjusted manually depending on available computation resources.

4.3.3 Aligning Signals

Since we don't have the number of accelerometers needed for the desired array geometry, the array measurements are constructed by moving the available sensors to the locations needed while keeping the actuator in the same location and transmitting identical signals.

Even though the initiation of a measurement is triggered by a pulse sent from the signal generator directly to the ADC, the relatively low sample rate causes notable differences in actual start timings. To make up for these small time differences, the measurements are aligned by the highest values in the transmitted signals, as described in Code listing C.3 in Appendix C.

4.4 Estimating Direction of Arrival

The DOA estimation methods are implemented as MATLAB System objects, listed in Table 4.3. In order to call the System objects on analytic signals input, the `scipy.signal.hilbert` is used for the transformation [37]. Where the signal envelopes are calculated, these are also done by taking the absolute value of the analytic signals retrieved from `scipy.signal.hilbert`.

Table 4.3: The MATLAB System objects for the DOA estimation methods.

DOA Estimation Method	MATLAB System object
MUSIC	`phased.MUSICEstimator` [38]
Root-MUSIC	`phased.RootMUSICEstimator` [39]
ESPRIT	`phased.ESPRITEstimator` [40]
Root-WSF	`phased.RootWSFEstimator` [41]

The `ScanAngles` property of `phased.MUSICEstimator` is set to search for angles in the range $[-90°, 90°]$ with a resolution of $1°$.

Although not within this project's scope, it is worth noting that the MATLAB System object of Root-MUSIC can optionally be applied to a UCA. The UCA version of Root-MUSIC expects a different orientation than the one used in the test setup, it is necessary to rotate the estimated

angles. From the documentation [42], the angle pointing towards the l-th element is given by

$$\theta_{\mathrm{m}} = \left(-\frac{L-1}{2} + l - 1\right)\frac{360^\circ}{L} \tag{4.2}$$

where L is the number of elements in the array, and m is the index of the element. Thus, the angle pointing towards the first element is 157.5°, and the estimated angles should be adjusted accordingly as described in Code listing C.4.

4.5 Calculating Wave Travel Times

The calculation of the time and distance traveled by a wave is based on the principle of mirrored sources, as presented in Section 2.1. By knowing the coordinates of the actuator and a sensor, the travel distances of the direct wave, first edge reflections and second edge reflections are calculated as described in Code listing C.5. This implementation iterates through combinations of edges and stores returns a list of travel times, while ignoring combinations of edges that are not physically possible or duplicate combinations. The coordinates of the mirrored source are found in `find_mirrored_source()` in Code listing C.6 by doubling the distance to the edges the source is being mirrored from.

Chapter 5

Results and Discussions

For this chapter, the results of the experiments of Chapter 4 will be presented and discussed consecutively, followed by a more general discussion of the design in Section 5.3. The first results to be presented are the observations made regarding the touch signal from accelerometers close to the source in Section 5.1, where a tap, a swipe and the release of the touch are considered.

Then follow the results of the direction of arrival (DOA) estimation and touch localization in Section 5.2, tested both on simulated signals and on real measurements. Both the simulations and the measurements start with comparing the array output signals with signals generated based on a group velocity and calculated arrival times for each wave. These comparisons are only intended to help better understand what is present in the signals when we know the source location and the sensor locations and are not used directly for the DOA estimation. The presented estimations are the results of applying the methods of Section 3.1 to different interesting configurations of the preprocessing, estimation parameters and source locations.

5.1 Touch Signals

To better understand what signals are available to work within the DOA estimation, we will first look at the signals generated by a touch during the initial impact, a swipe and the touch release. The analysis is relatively superficial and is intended to give an idea of what is going on in the plate upon a touch event rather than an extensive analysis. An attempt is made to highlight the S0 and A0 Lamb modes by placing an accelerometer directly above and below each other close to the touch location.

5.1.1 Touch Tapping

Figure 5.1 shows the measured signals from the sensors placed as described in Section 4.2.1, where “Sensor 1” is located directly under the touch and “Sensor 2” and “Sensor 3” are located on the top and bottom 5 cm away. The signal on “Sensor 1” is inverted to make it clearer what the real touch signal looks like instead of the signal from the underside of the plate. As we are, for the purpose of this project, not interested in the lower frequencies as discussed in Chapter 3, the signals are highpassed at 3 kHz to make them easier to interpret.

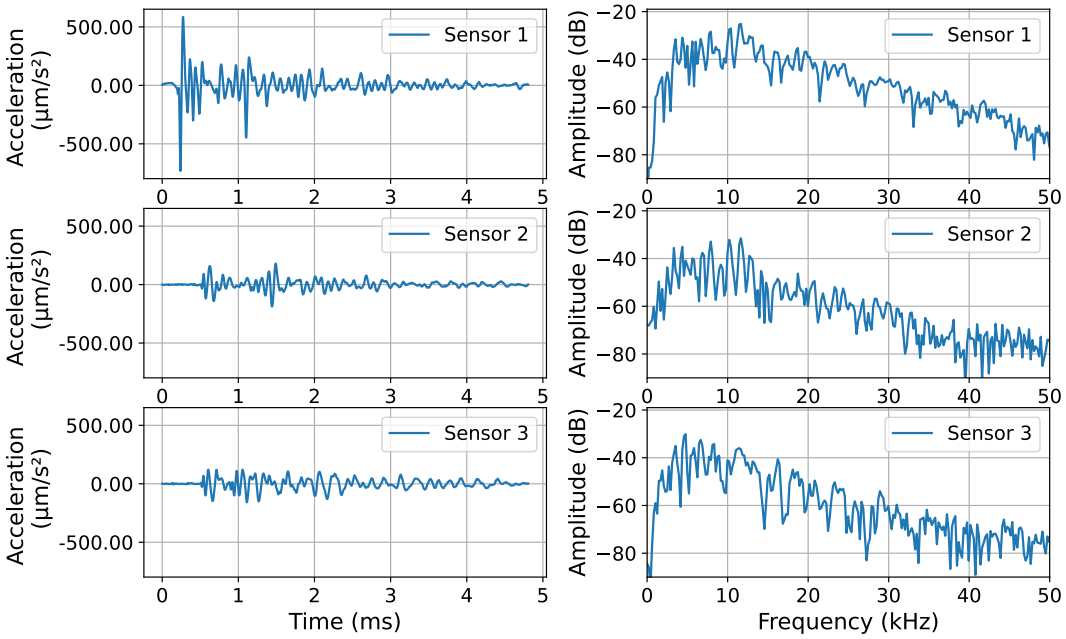


Figure 5.1: Touch signals measured directly under the touch location, above and under the plate 5 cm from the touch location. Highpassed at 3 kHz, and "Sensor 3" signal shifted by $-94.4\,\mu s$.

Under the assumption that the stronger motion in the plate should arrive simultaneously on "Sensor 2" and "Sensor 3", and that a deviation in arrival time is due to inaccurate placement of sensors, the signal on "Sensor 3" has been shifted by $-94.4\,\mu s$ to align the signals.

From "Sensor 1", we see that the touch signal, as one may expect, begins with a prominent first pulse lasting for approximately 0.1 ms. Then we may expect the plate to behave like a damped oscillator and keep vibrating after the touch. In this period, we also see a second pulse reminiscent of the initial one but with a lower amplitude. From other measurements, it is apparent that these peaks appear a short time after the touch and sometimes more than just once. Due to these variations in the same experiments, due to their relatively non-dispersive appearance, and their absence on "Sensor 2" and "Sensor 3", we do not believe that these peaks are not due to the edge reflections. As will be discussed in Section 5.1.2, the pulse resulting from the touch release is visible later in the signal than what is visible in Figure 5.1. With these possibilities ruled out, we believe that these peaks are either multiple contacts during the touch, such as the nail tapping the plate after the fingertip, although we do not want to exclude the possibility that it is an artifact of the complex plate vibrations. A final possibility is that it is generated by a contact between the wire cable and the plate. Unless the cause is the latter explanation, it is an interesting phenomenon when considering the transmitted signal from a touch tap.

As shown in the second column of Figure 5.1, most of the energy is contained below around 15 kHz, with an approximately linear drop-off in energy as the frequency increases. From "Sensor 2" and "Sensor 3", we see that the touch signal is highly attenuated already

after 5 cm, with a similar amplitude level on both sides of the plate. The amplitude has decreased from a peak at $-730\,\mu m/s^2$ on "Sensor 1" to a peak of $-148\,\mu m/s^2$ on "Sensor 3", a decrease of -6.93 dB in peak amplitude. Especially the higher frequencies are approaching the noise floor, starting at around 40 kHz. The rapid attenuation of higher frequencies is an important observation when considering the application of array processing. As discussed in Section 3.3, shorter wavelengths allow for higher spatial resolution given a fixed array aperture. When the signal available for array processing is fully controlled by the tap mechanism and the plate characteristics, there are inherent limitations to the spatial resolution that can be achieved. Note that accelerometers with a higher SNR, such as the new optical MEMS accelerometers, will have the possibility to utilize higher frequencies that would otherwise be too weak compared to the noise.

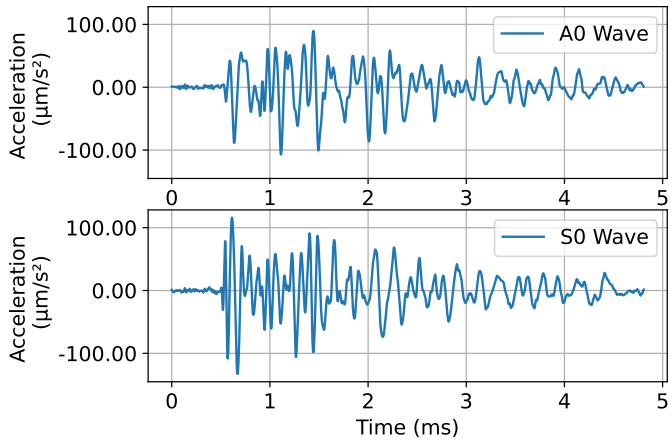


Figure 5.2: Filtering out S0 and A0 wave modes by subtracting and adding the signals measured at "Sensor 2" and "Sensor 3" in Figure 5.1, respectively, and dividing by two.

The results of the attempt to highlight the S0 and A0 wave modes as described in Section 4.2.1 is shown in Figure 5.2. For the initial pulse, there seems to be some correlation between the sides, as the constructed S0 signal seems to be somewhat augmented. Otherwise, there does not seem to be a good match between the sides, as would be expected in the case of Lamb waves. One explanation could be that other wave types are more dominant than the Lamb waves, such as surface or shear waves. Possible explanations for this could be that the sensors are placed too close to the source so that there are some proximity effects present that are not considered.

5.1.2 Touch Swiping and Release

Figure 5.3 shows what a typical touch, including a swipe, could look like from the array placed in the corner, as described in Section 4.2.2. The finger tap seen at 0.15 s is shortly followed by a swipe from a finger moving between the coordinates $[x, y] = [47\,\text{cm}, 0.40\,\text{cm}]$ and $[x, y] = [47\,\text{cm}, 30\,\text{cm}]$, starting at around 0.40 s. While a similar signal is seen at all sensors in the array, the specific measurement in Figure 5.3 is from "Sensor 7" in the array placed at $[x, y] = [5\,\text{cm}, 11\,\text{cm}]$. As we can see from the approximately 1 s long swipe, it is almost as wideband

as the touch signal itself, although with less energy at the lower frequencies with the energy mainly concentrated between 10 kHz and 15 kHz. The swipe also stretches a bit higher in frequency but is visibly closer to the noise floor than the rest of the bandwidth.

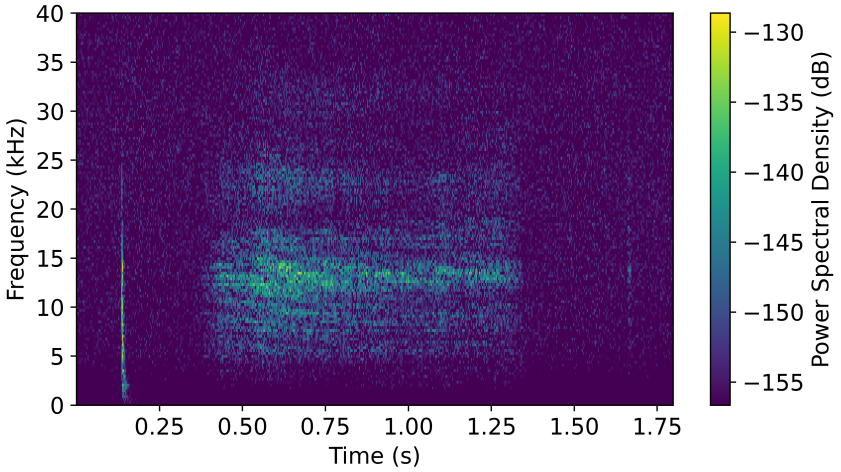


Figure 5.3: A spectrogram of the signal measured at "Sensor 7" in the setup described in Section 4.2.2. Plotted with 28 dB dynamic range.

So while there is less energy in the swipe than in the touch, more information reaches the sensors over this distance than one may expect. This is an important observation when considering the application of a touch-sensing control surface, where a user would often expect from previous experience that swiping functionality is available. However, the distribution of direct signal and edge reflections are not identified for the swipe, so the applicability of DOA methods is unclear. Perhaps especially for a swipe, where the applied force is smaller but more sustained than during a tap, we expect surface waves to be of higher importance.

A weak but highly interesting event in Figure 5.3 is the pulse at 1.65 s, where a pulse approximately at the length of the initial tap appears. This is the wave generated from the touch release due to the slight adhesion between the finger and the plate surface. As with the swipe, the SNR is presumably too low with the current accelerometers to be used reliably.

5.2 Estimated Angles and Touch Locations

The results of the DOA estimations are presented in this section. Section 5.2.1 shows the results of the applying MUSIC, Root-MUSIC, ESPRIT and Root-WSF to the signals simulated in COMSOL Multiphysics by transmitting a pulse in a plate, and registering the signal at ideal measuring points on the top surface at the ULA.

Results of the measurements performed on a real test setup are presented in Section 5.2.2. When interpreting these results, it should be remembered that the test setup has many flaws. The accelerometers must be moved around between transmitted pulses to achieve something close to the equivalent of the entire array. Some of the exact sensor characteristics of the accelerometer used for the channels "Sensor 2", "Sensor 5" and "Sensor 8". The accuracy to

which the actuator and sensors are mounted is limited to what is achievable with a marker pen and a measuring tape.

5.2.1 COMSOL Simulations

Inspired by the procedure described in Section 4.1, Figure 5.4 shows the envelopes of the COMSOL-simulated signals and the generated ideal signals for the ULA setup. To match the simulated signals, the Gaussian pulse in the ideal signal is modeled with (4.1) and set to an attenuation of 20 dB/m. Although not a perfect match, the similarities make it possible to confirm the expected arrival times of Figure 4.2 and that the group of waves between 0.75 ms and 1.50 ms are the indeed a combination of the direct wave and the first three arriving reflections, clearly separable from the next arrivals after 1.75 ms.

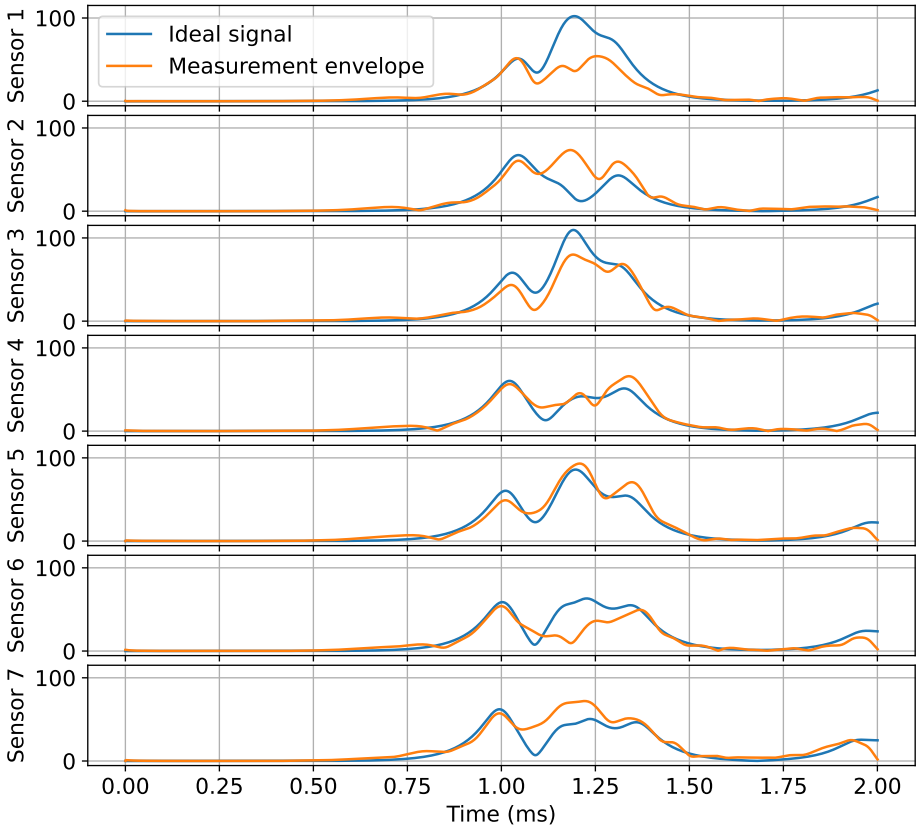


Figure 5.4: COMSOL simulations: Comparison of expected arrival times of the reflections with the measurements done in the ULA setup described in Section 4.1 with the actuator in the center of the plate. The signals are bandpass filtered to 22 kHz and modeled in the ideal signal as a Gaussian-modulated pulse with 20 dB/m attenuation.

Since the ideal signal is modeled only with respect to the A0 mode, an interesting observation can be made regarding the presence of the S0 mode. Recall from Section 4.1 that the A0 mode has an energy velocity at $v_{e,A0}/v_{e,S0} \approx 64\%$ of the S0 mode, meaning that the first

S0 wave should become apparent around 0.7 ms in Figure 5.4. A slight deviation between the ideal, A0-exclusive signals and the COMSOL-simulated signals can be seen, indicating some degree of S0 mode present but at a significantly lower energy level than the A0 mode.

Table 5.1: COMSOL simulations: Results for a source at $[x, y] = [50\,\text{cm}, 35\,\text{cm}]$, and bandpass filtered at $f_c = 22\,\text{kHz}$ with a phase velocity of $v_{\text{p,A0}} = 442.7$ m/s. Resulting in touch location estimates as shown in Figure 5.5a.

	True Angles	MUSIC (Error)	Root-MUSIC (Error)	ESPRIT (Error)	Root-WSF (Error)
θ_1	30.96°	33° (2.04°)	33.22° (2.26°)	34.03° (3.07°)	29.77° (1.19°)
θ_2	−43.70°	−37° (6.70°)	−51.98° (8.28°)	−51.64° (7.95°)	−45.39° (1.69°)
θ_3	26.15°	16° (10.15°)	15.02° (11.13°)	11.25° (14.89°)	29.77° (3.63°)
θ_4	−38.02°	−9° (29.02°)	−36.31° (1.70°)	−35.66° (2.35°)	−30.28° (7.74°)

Building on the observations made in Figure 5.4, we crop the signals to between 0.8 ms and 1.7 ms for the following DOA estimations. By starting with the simplest case, where a bandpass filter is applied at $f_c = 22\,\text{kHz}$ and no spatial smoothing or forward-backward averaging, the results listed in Table 5.1 are obtained. The estimated angles are sorted from lowest to highest and assigned to θ_1, θ_2, θ_3, and θ_4 as in (3.43). Except for θ_1 and θ_2 estimated by Root-WSF, no DOAs are within 2° of the true values, and in total, only four out of the 16 angles are within 4° with errors up to 29° for MUSIC on θ_4 and 14.89° for ESPRIT on θ_3. So far, Root-WSF seems to be handling the signals with the least amount of processing the best with a highest error of 7.74° on θ_4. The estimated touch locations can be seen in Figure 5.5a, where Root-WSF is the closest to the true location at an error of 8.8 cm.

Table 5.2: COMSOL simulations: Results for the same setup as for Table 5.1, but with one level of spatial smoothing and forward-backward averaging. Resulting in touch location estimates as shown in Figure 5.5b.

	True Angles	MUSIC (Error)	Root-MUSIC (Error)	ESPRIT (Error)	Root-WSF (Error)
θ_1	30.96°	31° (0.04°)	32.11° (1.15°)	30.35° (0.61°)	29.77° (1.19°)
θ_2	−43.70°	−44° (0.30°)	−44.37° (0.67°)	−44.27° (0.57°)	−45.39° (1.69°)
θ_3	26.15°	24° (2.15°)	23.87° (2.28°)	24.28° (1.87°)	29.77° (3.63°)
θ_4	−38.02°	−27° (11.02°)	−25.98° (12.03°)	−23.52° (14.50°)	−30.28° (7.74°)

We started the experiments assuming that the correlated sources would be an issue for most methods. Table 5.2 shows the results of the same parameters as in Table 5.1, but with one level of spatial smoothing and forward-backward averaging applied to all methods except Root-WSF. A clear improvement can be seen as errors are reduced to below 4° for θ_1, θ_2, and θ_3 for all methods and below 1° for some estimates of θ_1 and θ_2. The estimated touch locations are shown in Figure 5.5b, where clear improvement is made by all methods, all of which are within Root-WSF's error of 8.8 cm and Root-MUSIC being as close as 2.4 cm.

Since θ_4 is estimated poorly across all methods, it suggests that a property of the wave may make it difficult to estimate. Recall that θ_4 is the arriving angle of the wave that has been reflected twice from the plate edges and thus has traveled the longest distance. This also

Table 5.3: COMSOL simulations: Results for the same setup as for Table 5.2, but estimating three sources instead of four. Resulting in touch location estimates as shown in Figure 5.5c.

	True Angles	**MUSIC (Error)**	**Root-MUSIC (Error)**	**ESPRIT (Error)**	**Root-WSF (Error)**
θ_1	30.96°	27° (3.96°)	31.08° (0.12°)	27.01° (3.96°)	30.58° (0.39°)
θ_2	−43.70°	−43° (0.70°)	−43.03° (0.67°)	−42.94° (0.76°)	−43.24° (0.46°)
θ_3	26.15°	−10° (36.15°)	24.80° (1.34°)	27.01° (0.86°)	25.88° (0.27°)

raises the question of whether the errors in θ_2 in Table 5.1 are actually the results of an error in θ_4 being large enough for the angle assignments to be swapped, although the estimated θ_4 values are far off θ_2 as well. Motivated by this, Table 5.3 shows the result of running the same parameters as in Table 5.2, but with only trying to estimate the DOAs of three sources. Root-WSF now estimates all three DOAs with an error of less than 0.5°, a slight improvement for Root-MUSIC, no apparent improvement in ESPRIT, and worse results for MUSIC.

Table 5.4: COMSOL simulations: Results for the same setup as for Table 5.3, but with a second-order bandpass filter instead of a sixth-order bandpass filter. Resulting in touch location estimates as shown in Figure 5.5d.

	True Angles	**MUSIC (Error)**	**Root-MUSIC (Error)**	**ESPRIT (Error)**	**Root-WSF (Error)**
θ_1	30.96°	31° (0.04°)	30.72° (0.25°)	28.70° (2.27°)	36.67° (5.71°)
θ_2	−43.70°	−43° (0.70°)	−43.13° (0.57°)	−43.11° (0.59°)	−43.61° (0.09°)
θ_3	26.15°	−8° (34.15°)	24.96° (1.18°)	28.70° (2.55°)	28.36° (2.21°)

The good result for Root-WSF is noteworthy, although it should be emphasized that this is only from a single experiment and is only an indication of the method's performance in a zero-noise ideal environment. Figure 5.5c shows the resulting estimated touch locations, where Root-WSF is only 0.8 cm away from the true location with Root-MUSIC and ESPRIT at 3.4 cm and 7.43 cm, respectively. MUSIC misses the target by a large margin, as would be expected, as the method is not generally considered to perform reliably on correlated sources.

Table 5.5: COMSOL simulations: Results for the same setup as for Table 5.3, but with the phase velocity set to $v_{\mathrm{p,A0}} = 492.7\,\mathrm{m/s}$ instead of $v_{\mathrm{p,A0}} = 442.7\,\mathrm{m/s}$. Resulting in touch location estimates as shown in Figure 5.5e.

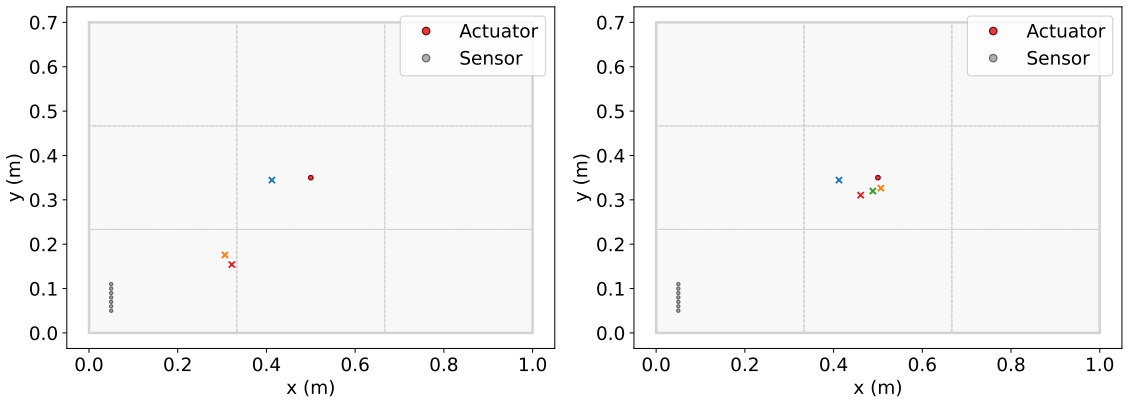
	True Angles	**MUSIC (Error)**	**Root-MUSIC (Error)**	**ESPRIT (Error)**	**Root-WSF (Error)**
θ_1	30.96°	30° (0.96°)	35.07° (4.11°)	30.36° (0.61°)	34.48° (3.52°)
θ_2	−43.70°	−49° (5.30°)	−49.41° (5.71°)	−49.30° (5.61°)	−49.68° (5.98°)
θ_3	26.15°	−11° (37.15°)	27.83° (1.68°)	30.36° (4.21°)	29.06° (2.91°)

To highlight the importance of the bandpass filtering parameters used, Table 5.4 shows the same setup as in Table 5.3, but with the Butterworth filter order decreased from 3 to 1, still at 2.2 kHz passband. The performance of Root-WSF is now significantly worse, while the estimated touch locations by Root-MUSIC and ESPRIT in Figure 5.5d are slightly better despite no real improvement in the DOA estimates.

Table 5.6: COMSOL simulations: Results for the same setup as for Table 5.3, but with the phase velocity set to $v_{\mathrm{p,A0}} = 392.7\,\mathrm{m/s}$ instead of $v_{\mathrm{p,A0}} = 442.7\,\mathrm{m/s}$. Resulting in touch location estimates as shown in Figure 5.5f.

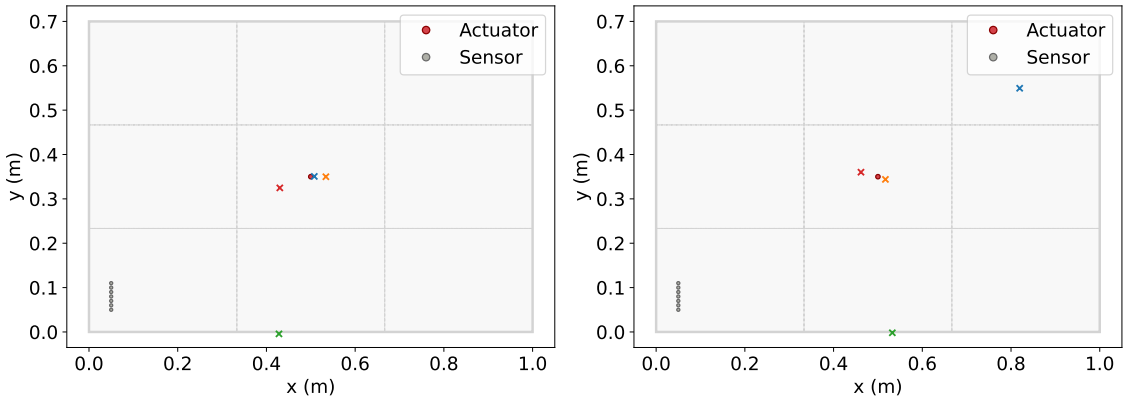
	True Angles	**MUSIC (Error)**	**Root-MUSIC (Error)**	**ESPRIT (Error)**	**Root-WSF (Error)**
θ_1	30.96°	24° (6.96°)	27.26° (3.71°)	23.75° (7.21°)	26.82° (4.14°)
θ_2	−43.70°	−37° (6.70°)	−37.25° (6.45°)	−37.18° (6.52°)	−37.42° (6.28°)
θ_3	26.15°	−9° (35.15°)	21.85° (4.30°)	23.75° (2.39°)	22.78° (3.37°)

To demonstrate the effect that an error in the propagation velocity has on the estimates, Table 5.5 and Figure 5.5e show the same setup as in Table 5.3, but with the phase velocity set to 50 m/s higher and lower, respectively, than the true value of $v_{\mathrm{p,A0}} = 442.7\,\mathrm{m/s}$. Within some proximity to the true phase velocity, the estimated DOAs will increase with increasing phase velocity as shown in Figure 5.5e, and decrease with decreasing phase velocity as shown in Figure 5.5f. This could imply that although the phase velocity is estimated incorrectly by the touch-sensing device itself, there is potential for a user interface where the user can manually adjust the phase velocity to improve its correct value. For instance, by knowing that they are touching the center of the table while seeing the estimated touch location on a display. If the purpose of the device is to recognize gestures, i.e. only the relative position of the touch is of interest, this could indicate that the exact value of $v_{\mathrm{p,A0}}$ may not be necessary to know.



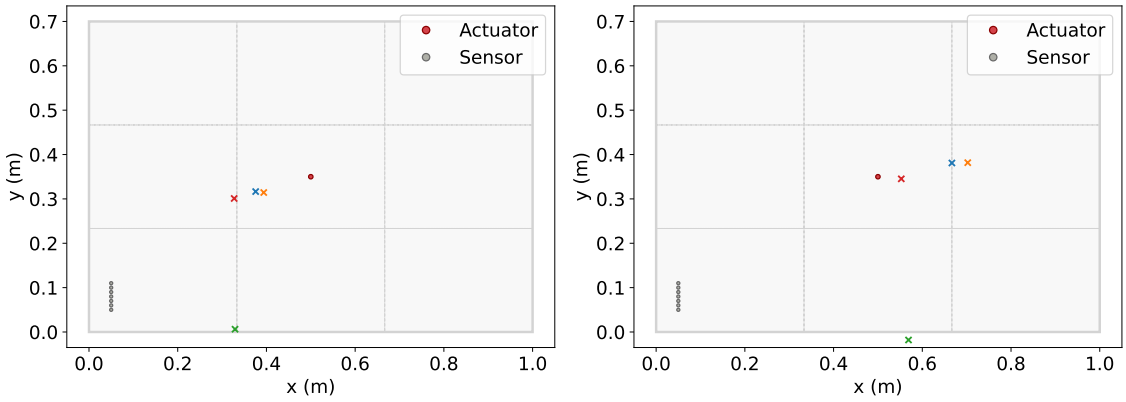
(a) No spatial smoothing.

(b) Spatial smoothing and FB averaging.



(c) Estimating three DOAs.

(d) Less aggressive filtering.



(e) Too high $v_{p,A0}$.

(f) Too low $v_{p,A0}$.

Figure 5.5: COMSOL simulations: The crosses mark the results of the touch localization using MUSIC (**green**), Root-MUSIC (**orange**), ESPRIT (**red**) and Root-WSF (**blue**), based on the estimated angles in (a) Table 5.1, (b) Table 5.2, (c) Table 5.3, (d) Table 5.4, (e) Table 5.5 and (f) Table 5.6.

5.2.2 Measurements

In the same fashion as for the simulations in Section 5.2.1, Figure 5.6 shows a comparison between the envelopes of the measured signals and a generated "ideal signal". The pulse in the ideal signal is modeled as the same single-period 25 kHz pulse transmitted into the plate, again bandpass filtered to 22 kHz and attenuated at 20 dB/m. After first inspecting the envelope of the measurements, it can be seen that the wave groups are arriving much earlier than the waves in the simulations. A certain mismatch between the simulated plate and the real plate is to be expected, although the real waves arrive almost twice as fast as the simulated waves. Although the phase and group velocities ideally should be found using a more precise and rigorous method, the real velocities are estimated to be approximately 1.8 times the simulated velocities, assuming that the phase velocity is scaled by the same factor as the group velocity. Using this new group velocity of $v'_{\mathrm{e,A0}} = 1.8 \cdot v_{\mathrm{e,A0}} = 1015.9\,\mathrm{m/s}$, the envelope of the ideal signal in Figure 5.6 is generated.

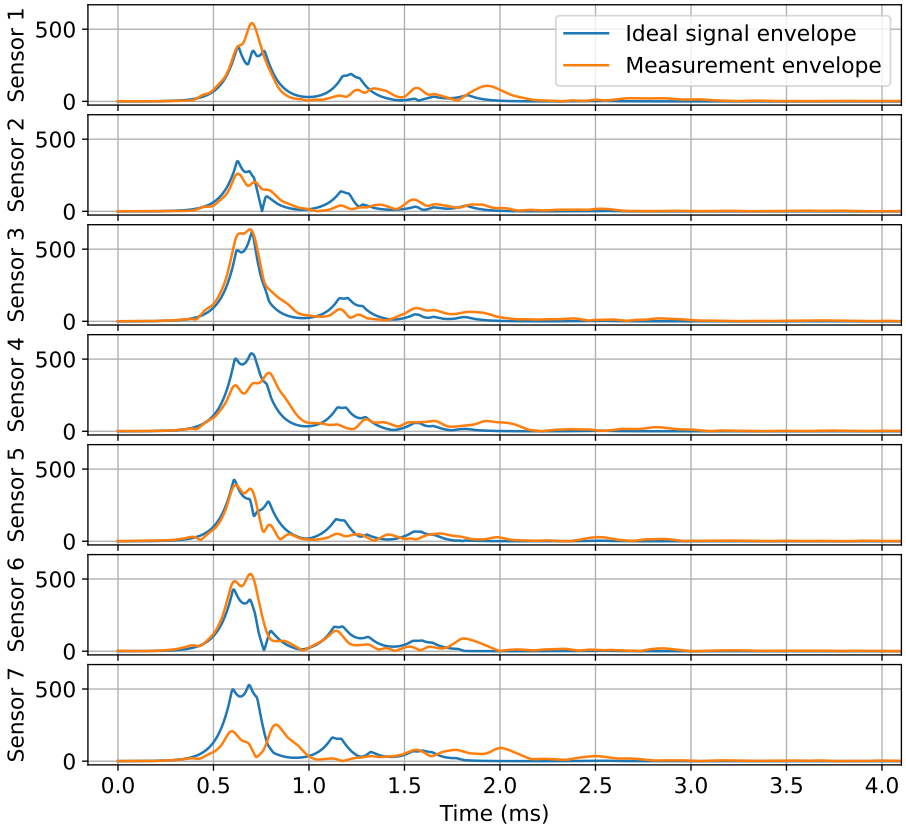


Figure 5.6: Measurements: Comparison of expected arrival times of the reflections with the measurements done in the ULA setup described in Section 4.2.2 with the actuator in the center of the plate. The pulse is modeled as the actually transmitted pulse with 20 dB/m attenuation.

The match between the two signals is not perfect, but some similarities can be seen. With the exception of "Sensor 7", the main group of wave arrivals between 0.5 ms and 1 ms is rel-

atively close for all sensors. A more notable deviation is seen for the arrivals after 1 ms, where the ideal signal would expect a more present group of waves around 1.2 ms, which does not seem to be as evident in the measurements. An exception to this is the measurement of "Sensor 6" which follows the ideal signal to an almost surprising degree.

The measurements appear to have more energy in the later arrivals after 1.5 ms. Considering that the ideal signal only models the A0 mode and that the S0 mode is typically sustained for longer in the plate, these later arrivals could be attributed to the reverberating S0 mode. Before the main group of waves arrives at 0.5 ms, a slight increase in energy reminiscent of the expected S0 mode can be seen, most visibly at "Sensor 5", "Sensor 6" and "Sensor 7". The higher velocities compared to the simulations would, however, make it more likely that the two modes overlap over the same distance traveled.

What is causing the unexpected behavior of the measurements at "Sensor 7", both in amplitude and the arriving groups of waves, is not clear. There is a possibility that the sensor is either not mounted properly or that there is a problem somewhere in the signal chain. This should, either way, be kept in mind when considering the results of the DOA estimations.

Table 5.7: Measurements: Results for a source at $[x, y] = [50\,\text{cm}, 35\,\text{cm}]$ using the same parameters as was used for the results in Table 5.3.

	True Angles	**MUSIC (Error)**	**Root-MUSIC (Error)**	**ESPRIT (Error)**	**Root-WSF (Error)**
θ_1	30.96°	65.00° (34.04°)	64.80° (33.84°)	64.29° (33.33°)	74.10° (43.14°)
θ_2	−43.70°	−22.00° (21.70°)	−21.17° (22.52°)	−23.73° (19.97°)	−22.43° (21.27°)
θ_3	26.15°	15.00° (11.15°)	12.95° (13.20°)	14.16° (11.98°)	16.29° (9.85°)

From the measured envelopes in Figure 5.6, we crop the signals to be between 0.4 ms and 1 ms to isolate the first four wave arrivals as previously discussed. Using the same parameters as for Table 5.3, the resulting DOAs are listed in Table 5.7, with the less significant estimated touch locations shown in Figure 5.7a. With errors between 9.85° and 43.14°, it may suggest the simulation phase velocity of $v_{\text{p,A0}} = 442.7\,\text{m/s}$ is not a good fit for the real plate, as would be expected from the higher group velocity.

Table 5.8: Measurements: The result of the same setup as for Table 5.7, but with the phase velocity adjusted to $v'_{\text{p,A0}} = 1.79 \cdot v_{\text{p,A0}} = 792.4\,\text{m/s}$.

	True Angles	**MUSIC (Error)**	**Root-MUSIC (Error)**	**ESPRIT (Error)**	**Root-WSF (Error)**
θ_1	30.96°	27.00° (3.96°)	23.64° (7.32°)	25.98° (4.99°)	30.15° (0.82°)
θ_2	−43.70°	−41.00° (2.70°)	−40.28° (3.42°)	−46.08° (2.38°)	−43.07° (0.62°)

For the following DOA estimation results, the real phase velocity is first scaled by the same factor as the group velocity so that $v'_{\text{p,A0}} = 1.8 \cdot v_{\text{p,A0}} = 796.9\,\text{m/s}$, and then corrected using the results discussed regarding Figure 5.5e and Figure 5.5f. The results of adjusting the phase velocity to be $v'_{\text{p,A0}} = 1.79 \cdot v_{\text{p,A0}} = 792.4\,\text{m/s}$ are shown in Table 5.8. While the errors for θ_1 and θ_2 are reduced to below 8°, and even below 1° for Root-WSF, θ_3 is not detected by any of the estimators. The reason for this is not apparent, but since θ_3 reaches the array after passing the sensor array first, it could be that the sensors cause a significant amount of attenuation

to the wave. Using only θ_1 and θ_2 for the two-angle touch localization method described in Section 3.4.1 given by (3.55), reasonable touch location estimates are obtained as shown in Figure 5.7b. Root-WSF is only 0.8 cm from the true location with this configuration.

Table 5.9: Measurements: Results for the same setup as Table 5.8, but with only six sensors.

	True Angles	MUSIC (Error)	Root-MUSIC (Error)	ESPRIT (Error)	Root-WSF (Error)
θ_1	31.43°	32.00° (0.57°)	32.45° (1.02°)	32.59° (1.16°)	30.01° (1.42°)
θ_2	−43.36°	−42.00° (1.36°)	−42.48° (0.89°)	−42.49° (0.88°)	−43.74° (0.37°)

Omitting "Sensor 7" as we suspect there may be something wrong or unusual with it, the results of the otherwise same setup as for Table 5.8 are shown in Table 5.9. We can see that the errors for the methods are overall lower with a maximum error of 1.42°, although the Root-WSF performs slightly worse. The touch location estimates are shown in Figure 5.7c, where Root-WSF is still the closest estimate at 7.2 cm from the true location, which gives an indication of the sensitivity of the trigonometric two-angle touch localization approach.

Table 5.10: Measurements: Results for the same setup as Table 5.9, but with the source located at $[x, y] = [45\,\text{cm}, 40\,\text{cm}]$.

	True Angles	MUSIC (Error)	Root-MUSIC (Error)	ESPRIT (Error)	Root-WSF (Error)
θ_1	39.09°	45.00° (5.91°)	45.01° (5.92°)	66.48° (27.39°)	35.29° (3.80°)
θ_2	−49.90°	−24.00° (25.90°)	−23.74° (26.16°)	−24.21° (25.69°)	−30.79° (19.11°)

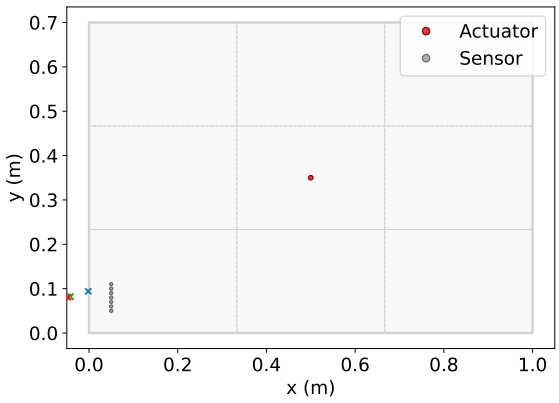
Further demonstrating the fragility of the proposed touch localization design, Table 5.10, Table 5.11 and Table 5.12 show the results of the same setup as for Table 5.9, but on sources located at $[x, y] = [45\,\text{cm}, 40\,\text{cm}]$, $[x, y] = [55\,\text{cm}, 40\,\text{cm}]$ and $[x, y] = [50\,\text{cm}, 20\,\text{cm}]$, respectively. So far the source has been located at $[x, y] = [50\,\text{cm}, 35\,\text{cm}]$. The errors are significantly higher for all these locations, although many somewhat reasonable estimates within 5° are still found. For $[x, y] = [45\,\text{cm}, 40\,\text{cm}]$, the estimates are no closer than 3.80° for Root-WSF on θ_1 and the resulting estimated touch locations are not even on the plate as seen in Figure 5.7d. For $[x, y] = [55\,\text{cm}, 40\,\text{cm}]$, the estimated DOAs are generally better and mostly within 4°, with the exception of Root-MUSIC with an error of 8.28° on θ_2. The estimated touch location from the Root-WSF is the closest at 4.9 cm from the true location, but it should be noted that small errors cause large differences in the estimated touch location like this. Moving downwards on the plate, all methods seem to struggle with the source at $[x, y] = [50\,\text{cm}, 20\,\text{cm}]$.

Table 5.11: Measurements: Results for the same setup as Table 5.9, but with the source located at $[x, y] = [55\,\text{cm}, 40\,\text{cm}]$.

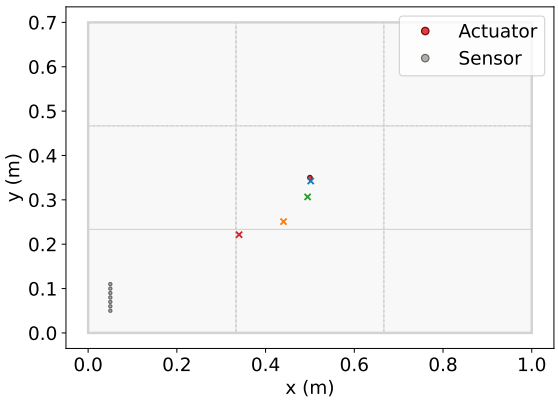
	True Angles	MUSIC (Error)	Root-MUSIC (Error)	ESPRIT (Error)	Root-WSF (Error)
θ_1	33.02°	34.00° (0.98°)	33.22° (2.26°)	34.31° (3.07°)	29.70° (3.32°)
θ_2	−43.53°	−40.00° (3.53°)	−51.98° (8.28°)	−40.46° (3.07°)	−41.29° (2.24°)

Table 5.12: Measurements: Results for the same setup as Table 5.9, but with the source located at $[x, y] = [50\,\text{cm}, 20\,\text{cm}]$.

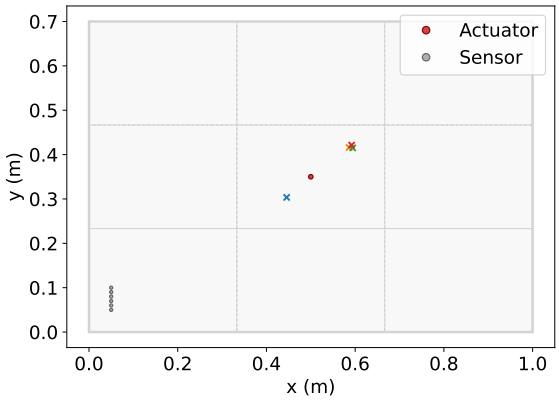
	True Angles	**MUSIC (Error)**	**Root-MUSIC (Error)**	**ESPRIT (Error)**	**Root-WSF (Error)**
θ_1	15.52°	52.00° (36.48°)	52.31° (36.79°)	46.59° (31.07°)	47.11° (31.59°)
θ_2	−31.43°	−80.00° (48.57°)	−71.87° (40.44°)	−22.20° (9.23°)	−63.26° (31.83°)



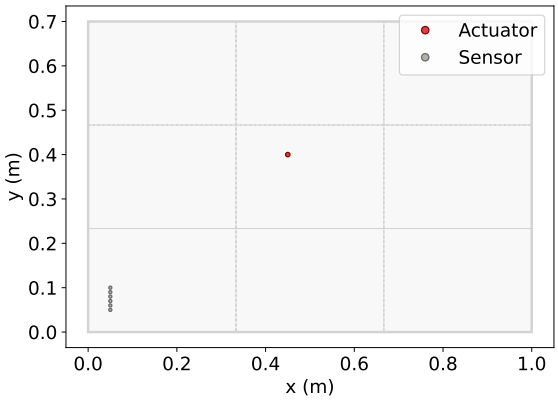
(a) Same parameters as in Figure 5.5c.



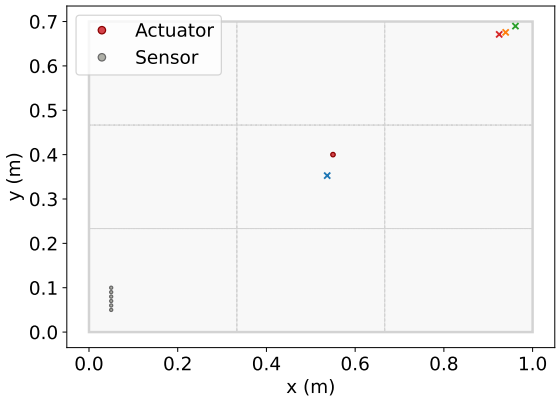
(b) Phase velocity $v_{\mathrm{p,A0}} = 792.4\,\mathrm{m/s}$.



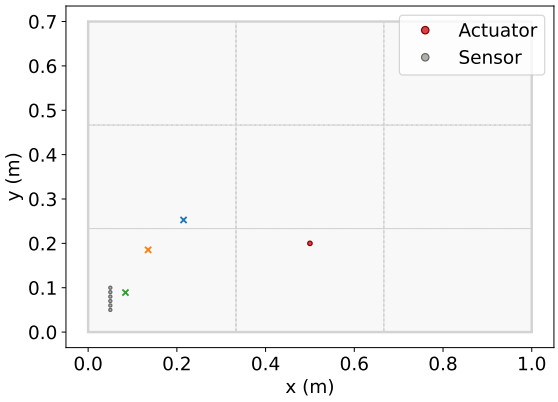
(c) Using only six sensors.



(d) Source at $[x, y] = [45\,\mathrm{cm}, 40\,\mathrm{cm}]$.



(e) Source at $[x, y] = [55\,\mathrm{cm}, 40\,\mathrm{cm}]$.



(f) Source at $[x, y] = [50\,\mathrm{cm}, 20\,\mathrm{cm}]$.

Figure 5.7: Measurements: The crosses mark results of the touch localization using MUSIC (**green**), Root-MUSIC (**orange**), ESPRIT (**red**) and Root-WSF (**blue**), based on the estimated angles in (a) Table 5.7, (b) Table 5.8, (c) Table 5.9, (d) Table 5.10, (e) Table 5.11 and (f) Table 5.12.

5.3 Discussion of the Design and Implementation

In the results from the COMSOL simulation, for instance, in Table 5.2 and Table 5.3, we see examples of why a UCA may be beneficial. When the true values of θ_1 and θ_3 are spaced closely together, in this case 4.81° apart, we see ESPRIT start to struggle separating the closely spaced DOAs. The resolution of the estimator is not high enough to separate the two angles, thus classifying them both as one angle. By using a UCA with 360° coverage, $\theta_3 = 26.15°$ would instead have the true value $\theta_3 = 153.85°$ while still having $\theta_1 = 30.96°$, avoiding the problem at least for the first four wave arrivals at the array. That this would be a common problem for many touch locations is clear from Figure 3.3 as presented in Section 3.4, and using a ULA would likely make expanding to using more than four DOAs more difficult.

The results presented in this chapter are of varying accuracy. Although the estimated DOAs are often relatively close to the true values, the touch localization results are insufficient for a practical application. A more efficient touch localization method should likely be less directly dependent on the accuracies of the DOA estimates and should implement as many signals as possible. Perhaps after estimating the plate dimensions and the phase velocity, a model can be trained that can estimate the touch location from simulated DOA combinations. Alternatively, in a more direct sense, use the plate model to trace each signal back to a certain number of reflections and estimate where these best intersect. It is important to keep in mind when developing a new method that the touch location should be repeatedly estimated to allow for swiping gestures, and so the number of sources present will vary depending on the specific window of the time signal.

We expect considerable improvements to be possible by better understanding the physics behind the wave propagation in plates, and more carefully applying the DOA estimation methods. Being able to place sensors closer together and thus including more sensors is also expected to help with the accuracy of the estimates. Regarding the measurements, the results are hard to evaluate fairly, considering the lack of available accelerometers and the limited achievable accuracy of the placements.

Chapter 6

Conclusions

In this project, a brief inspection of a finger touch signal propagating in a $1000 \times 700 \times 10\,\text{mm}$ Teflon plate was performed. When highpass filtered at 3 kHz, the pulse starts with a single-period pulse of 0.1 ms followed by what is assumed to be a combination of reflections, plate ringing and other non-identified pulses. From measurements at the source of the touch and 5 cm away, it was apparent that there is a significant decrease in amplitude during the wave propagation, showing a $-6.93\,\text{dB}$ decrease in peak amplitude over 5 cm measured on the underside of the plate. The most significant drop in energy was above 15 kHz for this specific plate over this distance. An attempt was made to highlight the S0 and A0 Lamb wave modes, although there seemed to be no clear correlation between the top and bottom sides of the plate. From a touch swipe close to the center of the plate measured in the corner of the plate, it was shown that the swipe contains a wide band of frequencies similar to the touch impact but at lower power and less frequency content below 5 kHz. A weak but evident event was seen when the finger was released from the plate after the swipe.

A trigonometric touch localization method has been presented and tested using four different DOA methods: MUSIC, Root-MUSIC, ESPRIT and Root-WSF. The goal was to do a feasibility study on accurately estimating the DOAs of waves reflected from the plate edges at an accelerometer array located in the corner of the plate.

First, tests were performed on simulations made in COMSOL Multiphysics for a pulse source in the middle of the plate and seven sensors between $[x, y] = [0.05\,\text{cm}, 0.05\,\text{cm}]$ and $[x, y] = [0.05\,\text{cm}, 0.11\,\text{cm}]$ spaced at 1 cm. The tested DOA estimation methods seemed to struggle to accurately estimate θ_4, which is the arrival angle of the wave reflected from both the edges of the corner at which the array is placed. This has notable implications in that only first reflections from the plate edges may be used for multiple-reflection DOA-based touch localization. By applying spatial smoothing and forward-backward averaging to the measurements and only estimating the DOAs of three sources, Root-WSF was able to accurately estimate the angles within a 0.5° error. An interesting relationship between the phase velocity and the estimated angles showed that the estimated angles would increase with increasing phase velocity.

Measurements on a real test-setup were performed, although under very limited conditions. We only had access to three accelerometers, meaning that the experiments using seven sensors needed to be done with a pulse transmitted from a piezo actuator instead of a finger

touch, and the sensors had to be moved around between measurements of the same pulse. After adjusting the phase velocity from $v_{\mathrm{p,A0}} = 442.7\,\mathrm{m/s}$ to $v_{\mathrm{p,A0}} = 792.4\,\mathrm{m/s}$, Root-WSF was able to estimate the DOA θ_1 of the direct wave and θ_2 of one reflection with less than 1° error. None of the methods were able to estimate θ_3 at all, consistently only returning two DOAs. This is suspected to be because of the attenuation applied to a wave traveling through the sensors. If this is the case, it is an interesting find for future DOA-based touch localization method.

Overall, it seemed that the Root-WSF method was the most promising for estimating the DOAs of the reflections from the plate edges. This is in agreement with the expectations given the properties of the considered methods. However, more quantitative tests should be done to determine the most suitable method.

Ideal signals were generated only considering the group velocity, the wave travel distances calculated using mirrored sources, and the originally transmitted pulse. These signals were used to compare the simulated and measured signals with their expected envelopes. The comparisons showed that the expected arrival times of the first four waves, being the direct wave and the first three reflections, were in good agreement with the expectations. Thus, by placing the sensor array in the corner of the plate, there should be an area around the middle of the plate where the first four wave arrivals are easily separable from the rest of the signal. This allows for identifying which DOAs belong to which signal and knowing the exact number of signals to expect.

It is the impression of the author that using DOA estimation on reflected Lamb waves for touch localization is a challenging problem, and there is much development to be done for a reliable and robust solution.

6.1 Further Work

A critical challenge of the DOA estimation is to determine the phase velocity, and it is essential to find a practical way to do this in a real-world application. Whether this could be done using the wave originating from a touch impact or an internal source in the touch-sensing device should be investigated. For a commercial product, it will also be important to deal with a wide range of possible phase velocities due to different plate materials, inhomogeneities in the plate, and potential variations due to temperature and humidity. Generally, designing a product that is robust to many variations in plate geometries, material composites, mechanical mounting points and plate edge shapes is expected to be a considerable effort after a concept has been developed. This problem will be novel to a product meant as a portable device, in contrast to most of the previous work presented in Chapter 1.

Different touch localization concepts should be researched alongside further development of the one presented here. Perhaps most interesting is incorporating the scattered wave from a finger held on the plate, as discussed in Section 3.4.2. Concepts where a model is trained to recognize, for instance, a combination of DOAs should also be looked into further. This or other ways to use the estimated DOAs in a less direct way should help reduce the error of a single estimated angle. If the use of DOA estimation will be central in further work, methods not considered for this project, such as DML, SML and the non-root version of WSF, could be applied and evaluated when applied to a UCA or other geometries.

In order to separate a touch event from other impacts on the plate during use, a method to recognize the touch should be developed. This could possibly be done by training a model to recognize the touch signal by having a user perform several touches on the plate before use. Succeeding in this will be necessary for a commercial product, as false positives would likely become a nuisance for the user.

Another important feature of a touch-sensing device is the ability to detect multiple touches and swipes in order to fulfill the expectations of a modern touch device. The lack of multi-touch features is one of the main drawbacks of previous acoustic touch-sensing solutions, and further development choices should be made with this in mind. For the proposed touch localization method in this project, adding this functionality would require finding a way to identify which DOAs belong to which touch localizations. This could possibly be done by finding which combinations of DOAs provide valid solutions of the touch location, although the feasibility of this is unclear. For swipes, the main challenge is to slice the measured signal into shorter time windows and then find the DOAs for each time window. To what extent the waves produced by a swipe are reflected from the edges should also be investigated.

Bibliography

[1] G. Walker, 'A review of technologies for sensing contact location on the surface of a display,' *Journal of the Society for Information Display*, vol. 20, no. 8, pp. 413–440, 2012. DOI: 10.1002/jsid.100.

[2] J. Kent, M. Takeuchi and G. Laux, '2B-1 Robert Adler's Touchscreen Inventions,' in *2007 IEEE Ultrasonics Symposium Proceedings*, Oct. 2007, pp. 9–20. DOI: 10.1109/ULTSYM.2007.18.

[3] K. North and H. D'Souza, 'Acoustic Pulse Recognition Enters Touch-Screen Market,' *Information Display*, pp. 22–25, Jun. 2006.

[4] R. K. Ing, N. Quieffin, S. Catheline and M. Fink, 'In solid localization of finger impacts using acoustic time-reversal process,' *Applied Physics Letters*, vol. 87, no. 20, p. 204 104, Nov. 2005, ISSN: 0003-6951. DOI: 10.1063/1.2130720. (visited on 30/01/2023).

[5] M. Fink, 'Time-reversal acoustics,' *Journal of Physics: Conference Series*, vol. 118, no. 1, p. 012 001, Oct. 2008, ISSN: 1742-6596. DOI: 10.1088/1742-6596/118/1/012001. (visited on 25/01/2023).

[6] K. Firouzi, A. Nikoozadeh, T. E. Carver and B. P. T. Khuri-Yakub, 'Lamb Wave Multitouch Ultrasonic Touchscreen,' *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 12, pp. 2174–2186, Dec. 2016, ISSN: 1525-8955. DOI: 10.1109/TUFFC.2016.2608781.

[7] M. Ge, 'Analysis of Source Location Algorithms, Part I: Overview and Non-Iterative Methods,' *Journal of Acoustic Emission*, vol. 21, 2003.

[8] M. Ge, 'Analysis of Source Location Algorithms, Part II. Iterative methods,' *Journal of Acoustic Emission*, vol. 21, no. 1, pp. 29–51, 2003.

[9] S. Yin, Z. Cui and T. Kundu, 'Acoustic source localization in anisotropic plates with "Z" shaped sensor clusters,' *Ultrasonics*, vol. 84, pp. 34–37, Mar. 2018, ISSN: 0041-624X. DOI: 10.1016/j.ultras.2017.10.007.

[10] R. Xiao, G. Lew, J. Marsanico, D. Hariharan, S. Hudson and C. Harrison, 'Toffee: Enabling ad hoc, around-Device interaction with acoustic time-of-arrival correlation,' in *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*, ser. MobileHCI '14, New York, NY, USA: Association for Computing Machinery, 2014, pp. 67–76, ISBN: 978-1-4503-3004-6. DOI: 10.1145/2628363.2628383.

[11] T. Dahl, A. Vogl, M. Bjerkeng and O. K. Pettersen, 'Off-screen Touch Detection using an Array of Optical MEMS Accelerometers,' SINTEF DIGITAL, Tech. Rep., Unpublished whitepaper.

[12] N. F. Declercq, A. Teklu, M. A. Breazeale, R. Briers, O. Leroy, J. Degrieck and G. N. Shkerdin, 'Study of the scattering of leaky Rayleigh waves at the extremity of a fluid-loaded thick plate,' *Journal of Applied Physics*, vol. 96, no. 10, pp. 5836–5840, Nov. 2004, ISSN: 0021-8979. DOI: 10.1063/1.1804618. eprint: https://pubs.aip.org/aip/jap/article-pdf/96/10/5836/13394262/5836_1_online.pdf. URL: https://doi.org/10.1063/1.1804618.

[13] A. Huber, 'Numerical Modeling of Guided Waves in Anisotropic Composites with Application to Air-coupled Ultrasonic Inspection,' Doctoralthesis, Universität Augsburg, 2021.

[14] B. Morvan, N. Wilkie-Chancellier, H. Duflo, A. Tinel and J. Duclos, 'Lamb wave reflection at the free edge of a plate,' *The Journal of the Acoustical Society of America*, vol. 113, no. 3, pp. 1417–1425, Feb. 2003, ISSN: 0001-4966. DOI: 10.1121/1.1539521.

[15] Y. Cho and J. L. Rose, 'A boundary element solution for a mode conversion study on the edge reflection of Lamb waves,' *The Journal of the Acoustical Society of America*, vol. 99, no. 4, pp. 2097–2109, Apr. 1996, ISSN: 0001-4966. DOI: 10.1121/1.415396.

[16] T. E. Vigran, *Building Acoustics*, 1st edition. CRC Press, Jul. 2008, ISBN: 978-0-415-42853-8.

[17] H. Krim and M. Viberg, 'Two decades of array signal processing research: The parametric approach,' *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, Jul. 1996, ISSN: 1558-0792. DOI: 10.1109/79.526899.

[18] A. Reilly, G. Frazer and B. Boashash, 'Analytic signal generation-tips and traps,' *IEEE Transactions on Signal Processing*, vol. 42, no. 11, pp. 3241–3245, Nov. 1994, ISSN: 1941-0476. DOI: 10.1109/78.330385.

[19] B. Picinbono, 'On instantaneous amplitude and phase of signals,' *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 552–560, Mar. 1997, ISSN: 1941-0476. DOI: 10.1109/78.558469.

[20] S. Tervo and A. Politis, 'Direction of Arrival Estimation of Reflections from Room Impulse Responses Using a Spherical Microphone Array,' *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 10, pp. 1539–1551, Oct. 2015, ISSN: 2329-9304. DOI: 10.1109/TASLP.2015.2439573.

[21] O. Famoriji and T. Shongwe, 'Critical Review of Basic Methods on DoA Estimation of EM Waves Impinging a Spherical Antenna Array,' *Electronics*, vol. 11, p. 208, Jan. 2022. DOI: 10.3390/electronics11020208.

[22] M. Viberg, B. Ottersten and T. Kailath, 'Detection and estimation in sensor arrays using weighted subspace fitting,' *IEEE Transactions on Signal Processing*, vol. 39, no. 11, pp. 2436–2449, Nov. 1991, ISSN: 1941-0476. DOI: 10.1109/78.97999.

[23] B. Boustani, B. Abdennaceur, A. Sahel, A. Ballouk and A. Badr, 'Performance analysis of direction of arrival estimation under hard condition,' Apr. 2018, pp. 1–5. DOI: 10.1109/ICOA.2018.8370544.

[24] C. Mathews and M. Zoltowski, 'Eigenstructure techniques for 2-D angle estimation with uniform circular arrays,' *IEEE Transactions on Signal Processing*, vol. 42, no. 9, pp. 2395–2407, Sep. 1994, ISSN: 1941-0476. DOI: 10.1109/78.317861.

[25] R. O. Schmidt, 'A signal subspace approach to multiple emitter location and spectral estimation,' Ph.D. dissertation, 1981.

[26] M. Pesavento and A. Gershman, 'Maximum-likelihood direction-of-arrival estimation in the presence of unknown nonuniform noise,' *IEEE Transactions on Signal Processing*, vol. 49, no. 7, pp. 1310–1324, 2001. DOI: 10.1109/78.928686.

[27] S. Pillai and B. Kwon, 'Forward/backward spatial smoothing techniques for coherent signal identification,' *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, pp. 8–15, 1989. DOI: 10.1109/29.17496.

[28] A. Vesa and C. Simu, 'Performances of uniform sensor array antenna in case of doa estimation using the music algorithm,' in *2022 International Symposium on Electronics and Telecommunications (ISETC)*, IEEE, 2022, pp. 1–4.

[29] R. Kennedy, T. Abhayapala and D. Ward, 'Broadband nearfield beamforming using a radial beampattern transformation,' *IEEE Transactions on Signal Processing*, vol. 46, no. 8, pp. 2147–2156, 1998. DOI: 10.1109/78.705426.

[30] K. C. Kang, Y. H. Kim, J. Y. Pyun and K. K. Park, 'Feasibility study on multi-touch ultrasound large-panel touchscreen using guided lamb waves,' *Measurement*, vol. 190, p. 110 755, 2022, ISSN: 0263-2241. DOI: 10.1016/j.measurement.2022.110755.

[31] N. Strømsnes, 'Analysis of flexural waves in a table and experiments for touch sensing purposes,' Norwegian University of Science and Technology, 9th Jan. 2023.

[32] C. Xu, J. Wang, S. Yin and M. Deng, 'A focusing MUSIC algorithm for baseline-free Lamb wave damage localization,' *Mechanical Systems and Signal Processing*, vol. 164, p. 108 242, Feb. 2022, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2021.108242.

[33] T. Johansen, 'Parameters gotten from personal communication with SINTEF.,' 15th Feb. 2023.

[34] 'Scipy.signal.butter — SciPy v1.10.1 Manual,' URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html (visited on 23/05/2023).

[35] 'Scipy.signal.sosfiltfilt — SciPy v1.10.1 Manual,' URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.sosfiltfilt.html (visited on 23/05/2023).

[36] 'Scipy.interpolate.interp1d — SciPy v1.10.1 Manual,' URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html (visited on 23/05/2023).

[37] 'Scipy.signal.hilbert — SciPy v1.10.1 Manual,' URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.hilbert.html (visited on 04/06/2023).

[38] 'Music estimator - MATLAB - MathWorks Nordic,' URL: https://se.mathworks.com/help/phased/ref/phased.musicestimator-system-object.html (visited on 17/06/2023).

[39] 'Root-music estimator - MATLAB - MathWorks Nordic,' URL: https://se.mathworks.com/help/phased/ref/phased.rootmusicestimator-system-object.html (visited on 17/06/2023).

[40] 'Esprit estimator - MATLAB - MathWorks Nordic,' URL: https://se.mathworks.com/help/phased/ref/phased.espritestimator-system-object.html (visited on 17/06/2023).

[41] 'Root-wsf estimator - MATLAB - MathWorks Nordic,' URL: https://se.mathworks.com/help/phased/ref/phased.rootwsfestimator-system-object.html (visited on 17/06/2023).

[42] 'Uniform circular array - MATLAB - MathWorks Nordic,' URL: https://se.mathworks.com/help/phased/ref/phased.uca-system-object.html (visited on 23/05/2023).

Appendix A

Additional Material

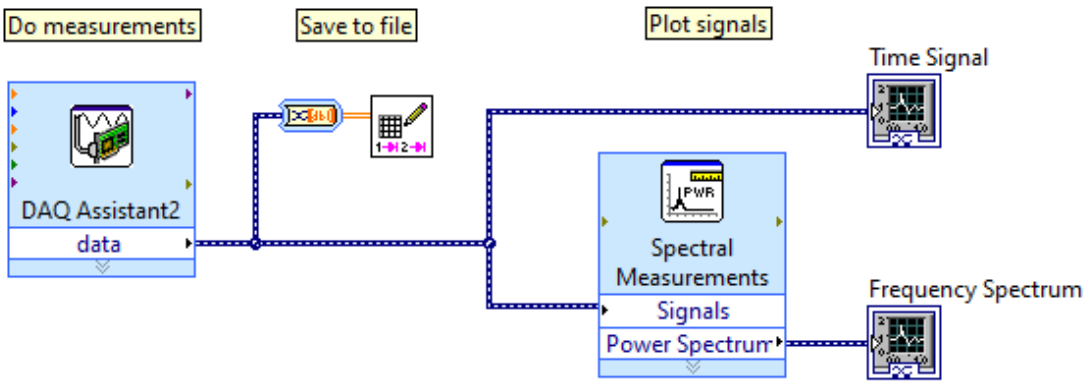


Figure A.1: The LabVIEW setup used for the measurements.

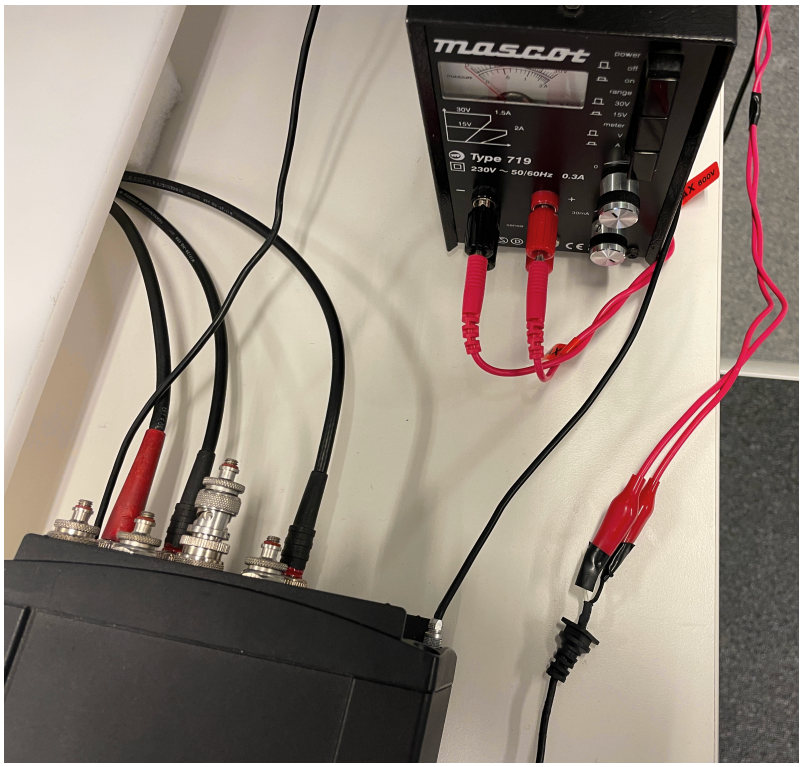


Figure A.2: The fix for the amplifier power supply.

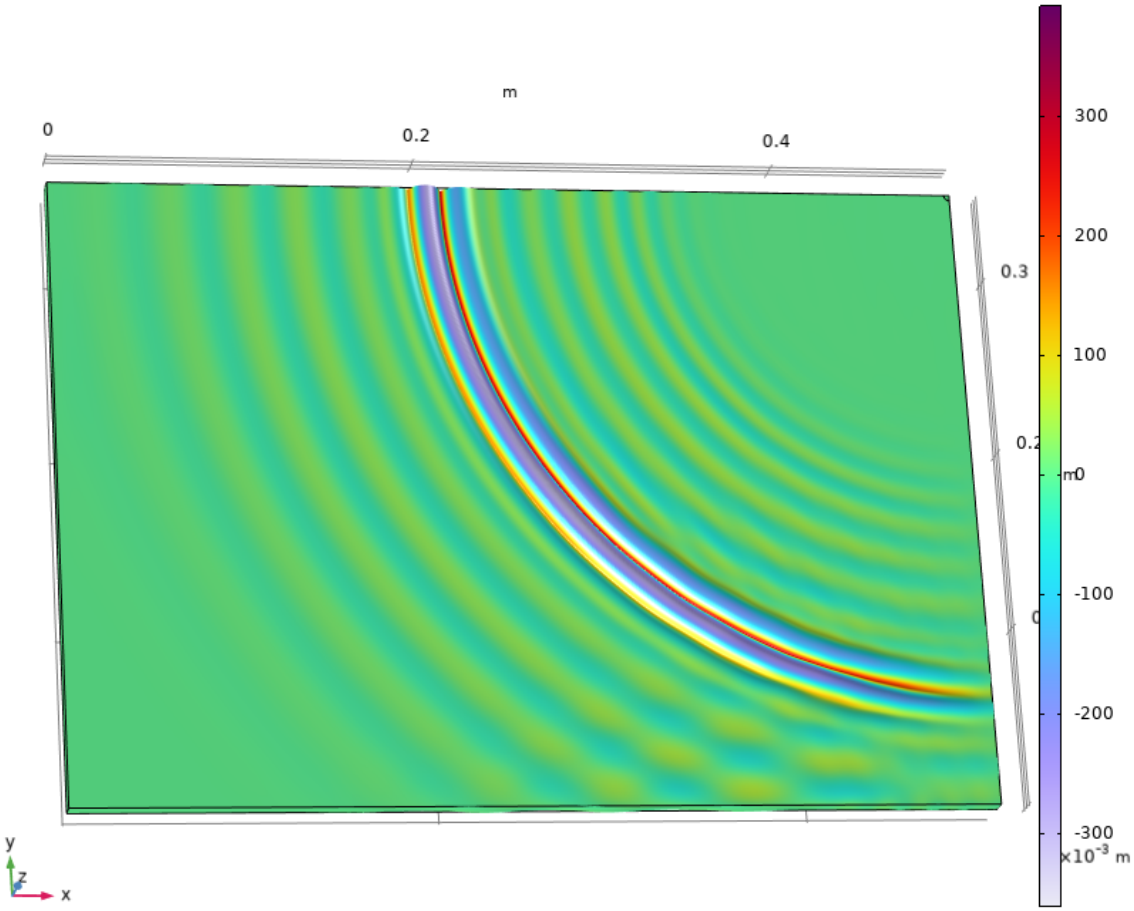


Figure A.3: A snapshot of the COMSOL simulation at $t = 600\,\mu s$, showing the acceleration in the z-direction of the plate.

Appendix B

Sensor Datasheets

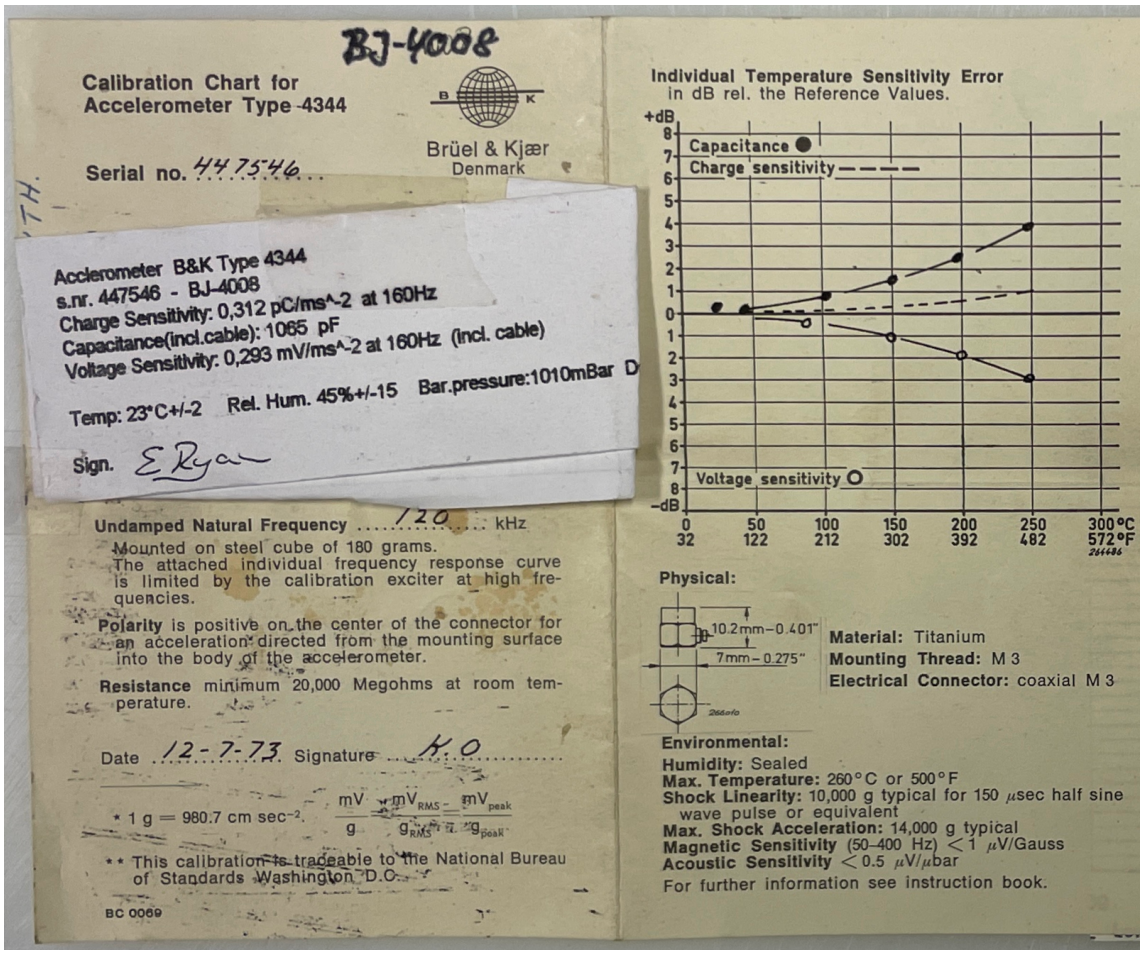


Figure B.1: The specifications for “Sensor 1".

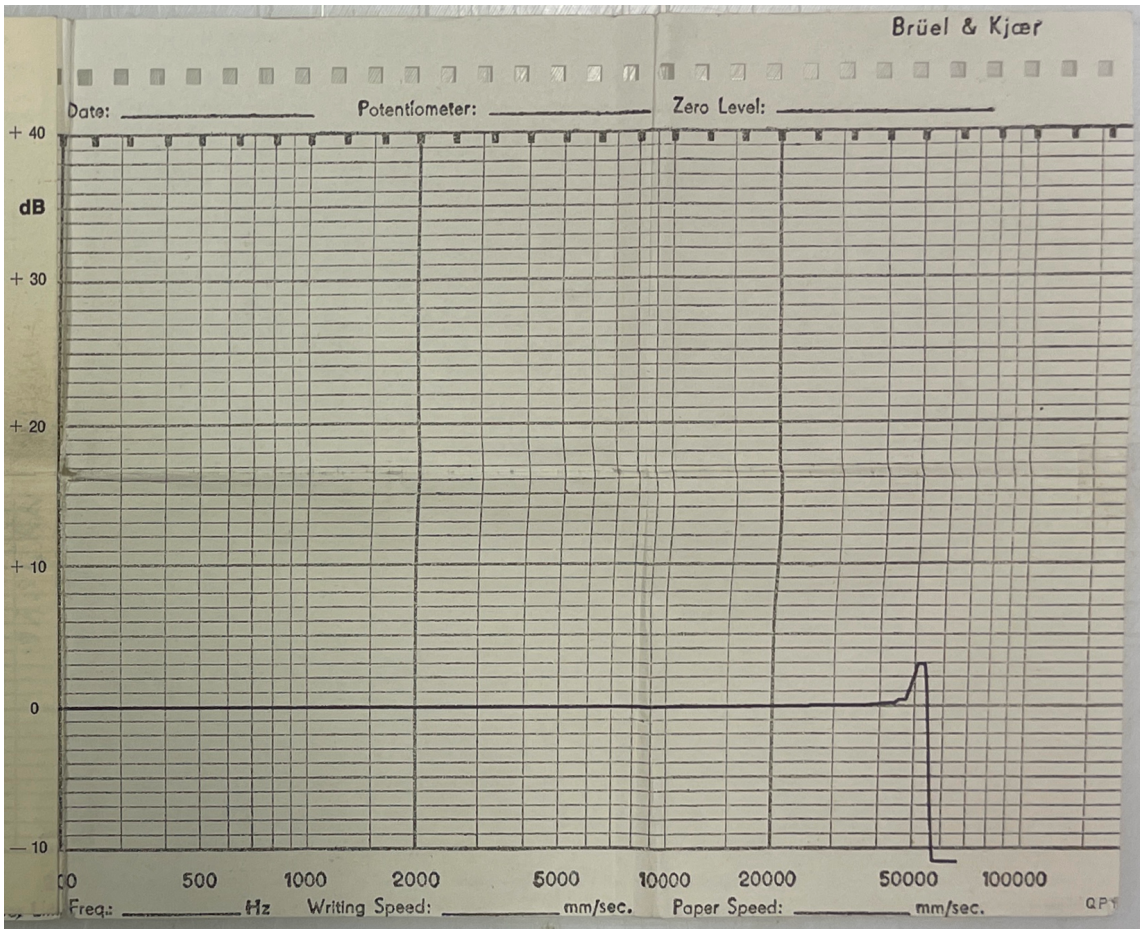
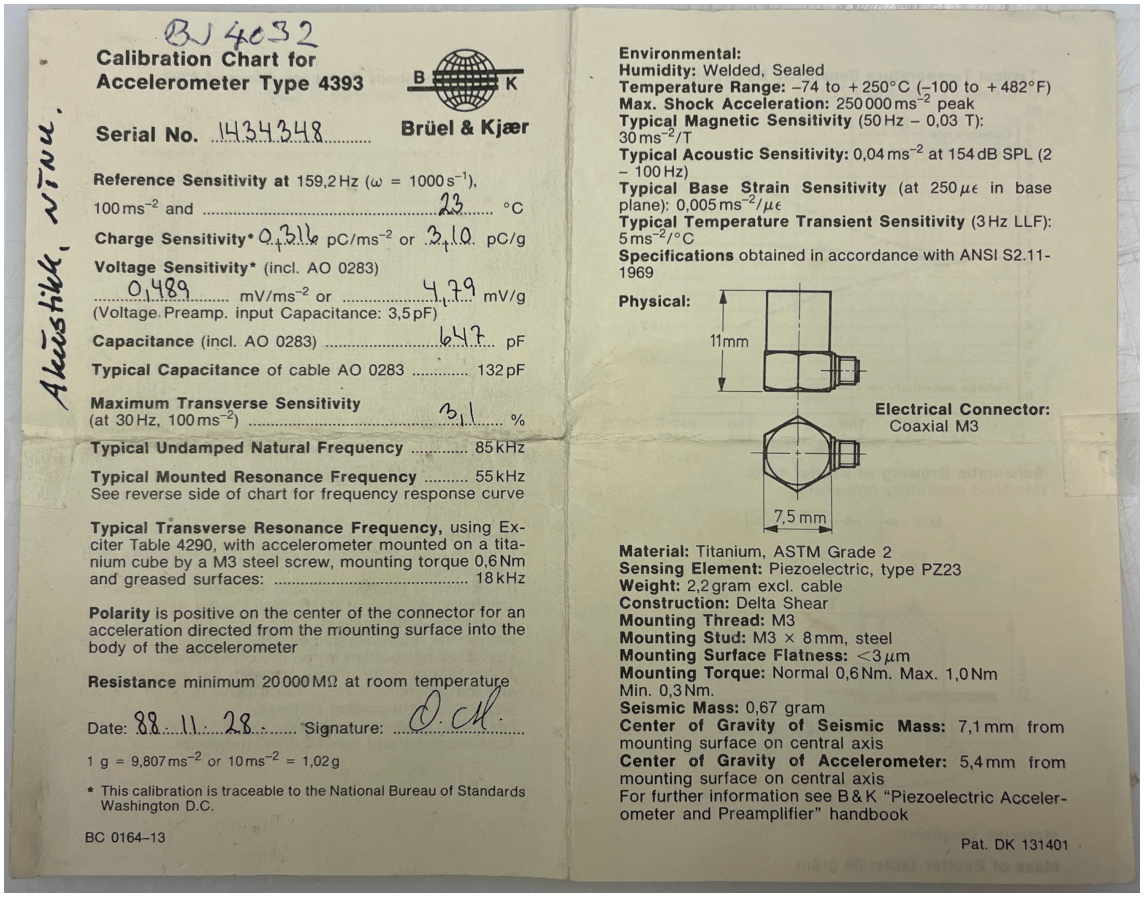


Figure B.2: The back side of the specifications for “Sensor 1".



BJ 4032

Calibration Chart for Accelerometer Type 4393

Serial No. 1434348

Brüel & Kjær

Akustik, NTNU.

Reference Sensitivity at 159,2 Hz (ω = 1000 s^{-1}), 100 ms^{-2} and 23 °C

Charge Sensitivity* 0,311 pC/ms^{-2} or 3,10 pC/g

Voltage Sensitivity* (incl. AO 0283) 0,489 mV/ms^{-2} or 4,79 mV/g (Voltage Preamp. input Capacitance: 3,5 pF)

Capacitance (incl. AO 0283) 647 pF

Typical Capacitance of cable AO 0283 ..... 132 pF

Maximum Transverse Sensitivity (at 30 Hz, 100 ms^{-2}) 3,1 %

Typical Undamped Natural Frequency ..... 85 kHz

Typical Mounted Resonance Frequency ..... 55 kHz
See reverse side of chart for frequency response curve

Typical Transverse Resonance Frequency, using Exciter Table 4290, with accelerometer mounted on a titanium cube by a M3 steel screw, mounting torque 0,6 Nm and greased surfaces: ..... 18 kHz

Polarity is positive on the center of the connector for an acceleration directed from the mounting surface into the body of the accelerometer

Resistance minimum 20 000 MΩ at room temperature

Date: 88.11.28 Signature: O. Ch.

1 g = 9,807 ms^{-2} or 10 ms^{-2} = 1,02 g

* This calibration is traceable to the National Bureau of Standards Washington D.C.

BC 0164–13

Environmental:
Humidity: Welded, Sealed
Temperature Range: –74 to + 250°C (–100 to + 482°F)
Max. Shock Acceleration: 250 000 ms^{-2} peak
Typical Magnetic Sensitivity (50 Hz – 0,03 T): 30 ms^{-2}/T
Typical Acoustic Sensitivity: 0,04 ms^{-2} at 154 dB SPL (2 – 100 Hz)
Typical Base Strain Sensitivity (at 250 $\mu\epsilon$ in base plane): 0,005 $ms^{-2}/\mu\epsilon$
Typical Temperature Transient Sensitivity (3 Hz LLF): 5 ms^{-2}/°C
Specifications obtained in accordance with ANSI S2.11-1969

Physical:

11mm

Electrical Connector: Coaxial M3

7,5 mm

Material: Titanium, ASTM Grade 2
Sensing Element: Piezoelectric, type PZ23
Weight: 2,2 gram excl. cable
Construction: Delta Shear
Mounting Thread: M3
Mounting Stud: M3 × 8 mm, steel
Mounting Surface Flatness: <3 μm
Mounting Torque: Normal 0,6 Nm. Max. 1,0 Nm Min. 0,3 Nm.
Seismic Mass: 0,67 gram
Center of Gravity of Seismic Mass: 7,1 mm from mounting surface on central axis
Center of Gravity of Accelerometer: 5,4 mm from mounting surface on central axis
For further information see B & K "Piezoelectric Accelerometer and Preamplifier" handbook

Pat. DK 131401

Figure B.3: The front side of the specifications for "Sensor 2".

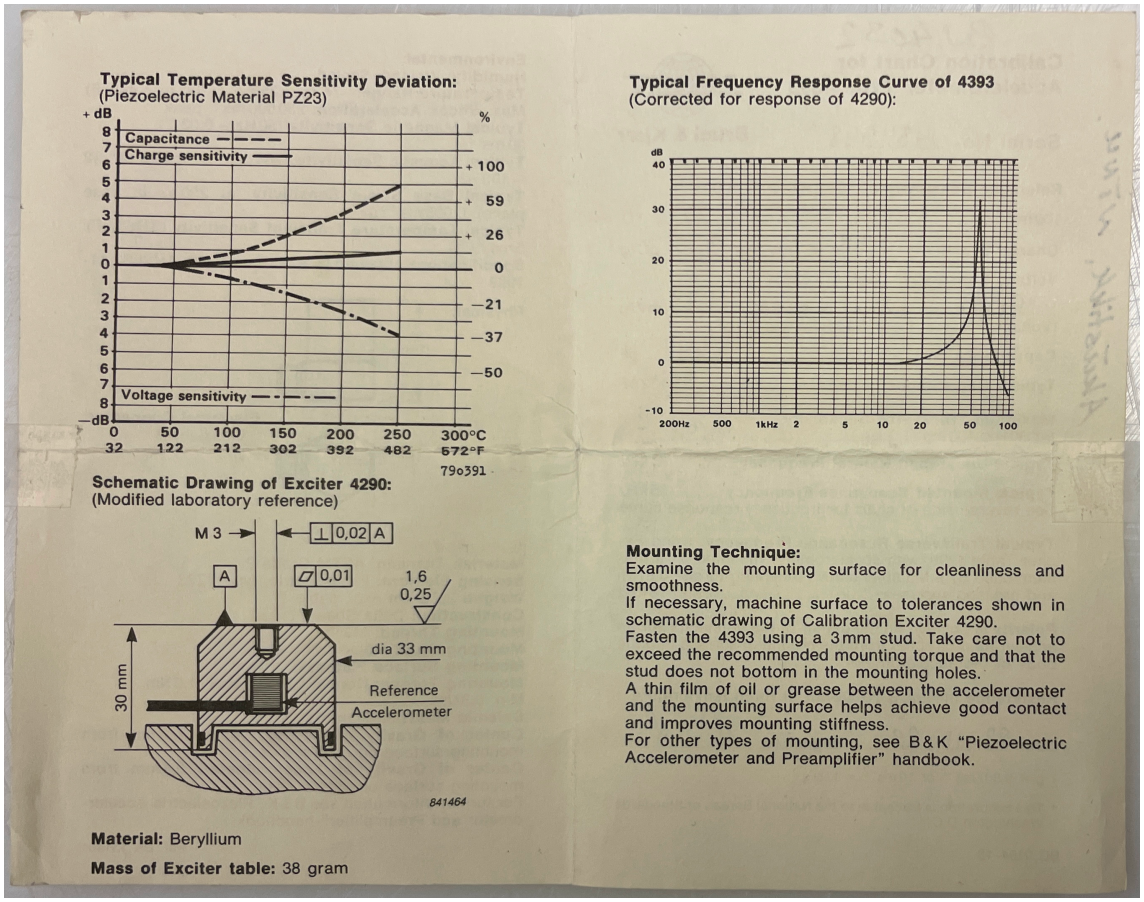


Figure B.4: The back side of the specifications for "Sensor 2".

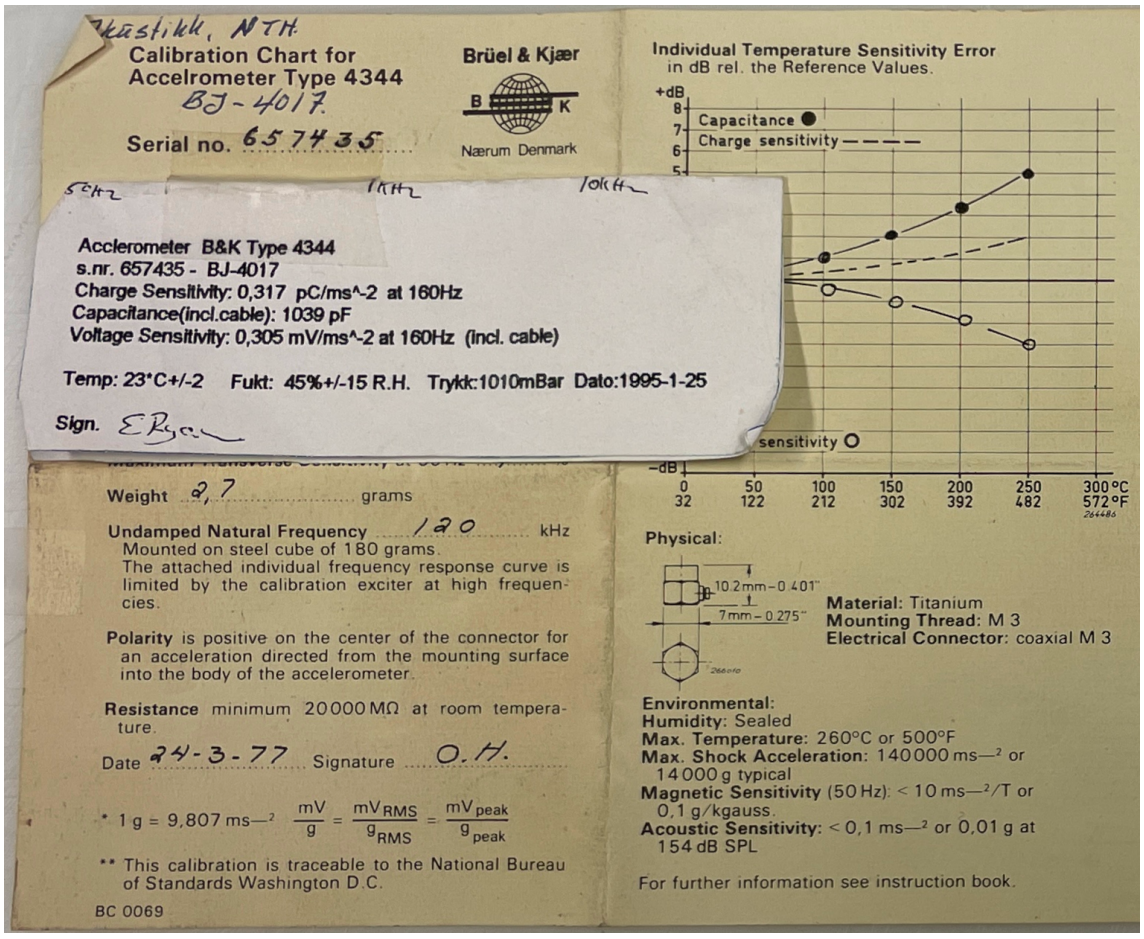


Figure B.5: The back side of the specifications for “Sensor 3".

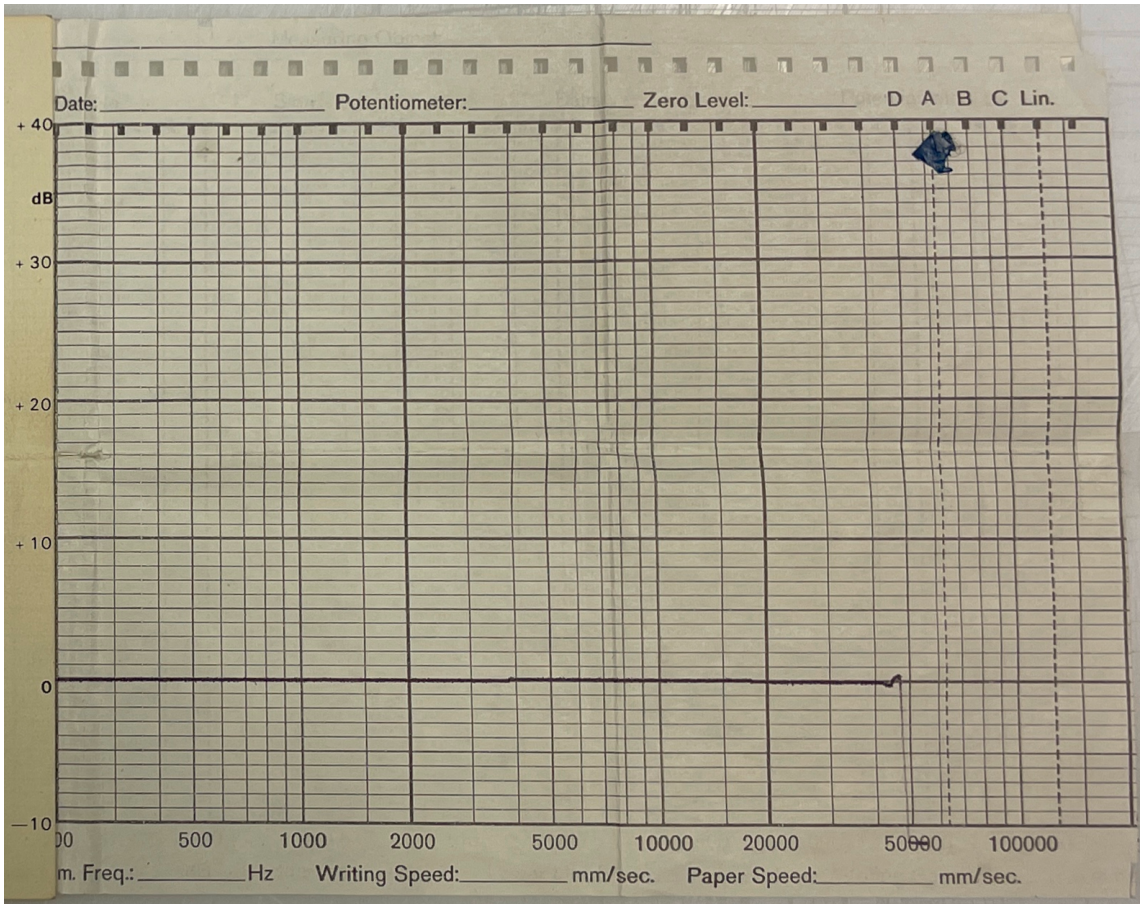


Figure B.6: The back side of the specifications for “Sensor 3".

Appendix C

Code Implementations of Preprocessing

Code listing C.1: Python function to filter a signal, either as highpass, lowpass or bandpass.

```python
def filter_signal(
    signal: np.ndarray,
    filtertype: str,
    critical_frequency: int,
    q: float,
    order: int,
) -> np.ndarray:
    if filtertype == "highpass" or filtertype == "lowpass":
        sos = signal.butter(
            N=order,
            Wn=critical_frequency / (0.5 * SAMPLE_RATE),
            btype=filtertype,
            output="sos",
        )
    elif filtertype == "bandpass":
        sos = signal.butter(
            N=order,
            Wn=[
                critical_frequency * (1 - q) / (0.5 * SAMPLE_RATE),
                critical_frequency * (1 + q) / (0.5 * SAMPLE_RATE),
            ],
            btype="bandpass",
            output="sos",
        )
    signal_filtered = signal.sosfiltfilt(sos, signal, axis=0)
    return signal_filtered
```

Code listing C.2: Python function to interpolate a signal.

```python
def interpolate_signal(signal: np.ndarray) -> np.ndarray:
    new_length = signals.shape[0] * INTERPOLATION_FACTOR
    sample_axis_original = np.linspace(
        start=0,
        stop=signals.size,
        num=signals.size,
    )
    interpolation_function = interpolate.interp1d(
        x=sample_axis_original,
        y=signals,
        kind="cubic",
    )
    sample_axis_interpolated = np.linspace(
        start=0,
        stop=signals.size,
        num=new_length,
    )
    return interpolation_function(sample_axis_interpolated)
```

Code listing C.3: Python function to align the measurements by the transmitted signals.

```python
def align_transmitted_signal(measurements: pd.DataFrame) -> None:
    # Find the delay of the sync signals using the maximum of the "Actuator" signal
    delay456 = np.argmax(measurements["Actuator_456"]) - np.argmax(measurements["Actuator_123"])
    delay78 = np.argmax(measurements["Actuator_78"]) - np.argmax(measurements["Actuator_123"])

    # Shift sensors 4, 5, 6, 7 and 8 by the delay of their actuator signals
    measurements["Sensor_4"] = np.roll(measurements["Sensor_4"], -delay456)
    measurements["Sensor_5"] = np.roll(measurements["Sensor_5"], -delay456)
    measurements["Sensor_6"] = np.roll(measurements["Sensor_6"], -delay456)
    measurements["Sensor_7"] = np.roll(measurements["Sensor_7"], -delay78)
    measurements["Sensor_8"] = np.roll(measurements["Sensor_8"], -delay78)

    # Shift the sync signals by the delay of their sync signals
    measurements["Sync_Signal_456"] = np.roll(measurements["Sync_Signal_456"], -delay456)
    measurements["Sync_Signal_78"] = np.roll(measurements["Sync_Signal_78"], -delay78)

    # Shift the actuators by the delay of their sync signals
    measurements["Actuator_456"] = np.roll(measurements["Actuator_456"], -delay456)
    measurements["Actuator_78"] = np.roll(measurements["Actuator_78"], -delay78)
```

Code listing C.4: Matlab code for rotating the estimated angles from a UCA to match the test setup.

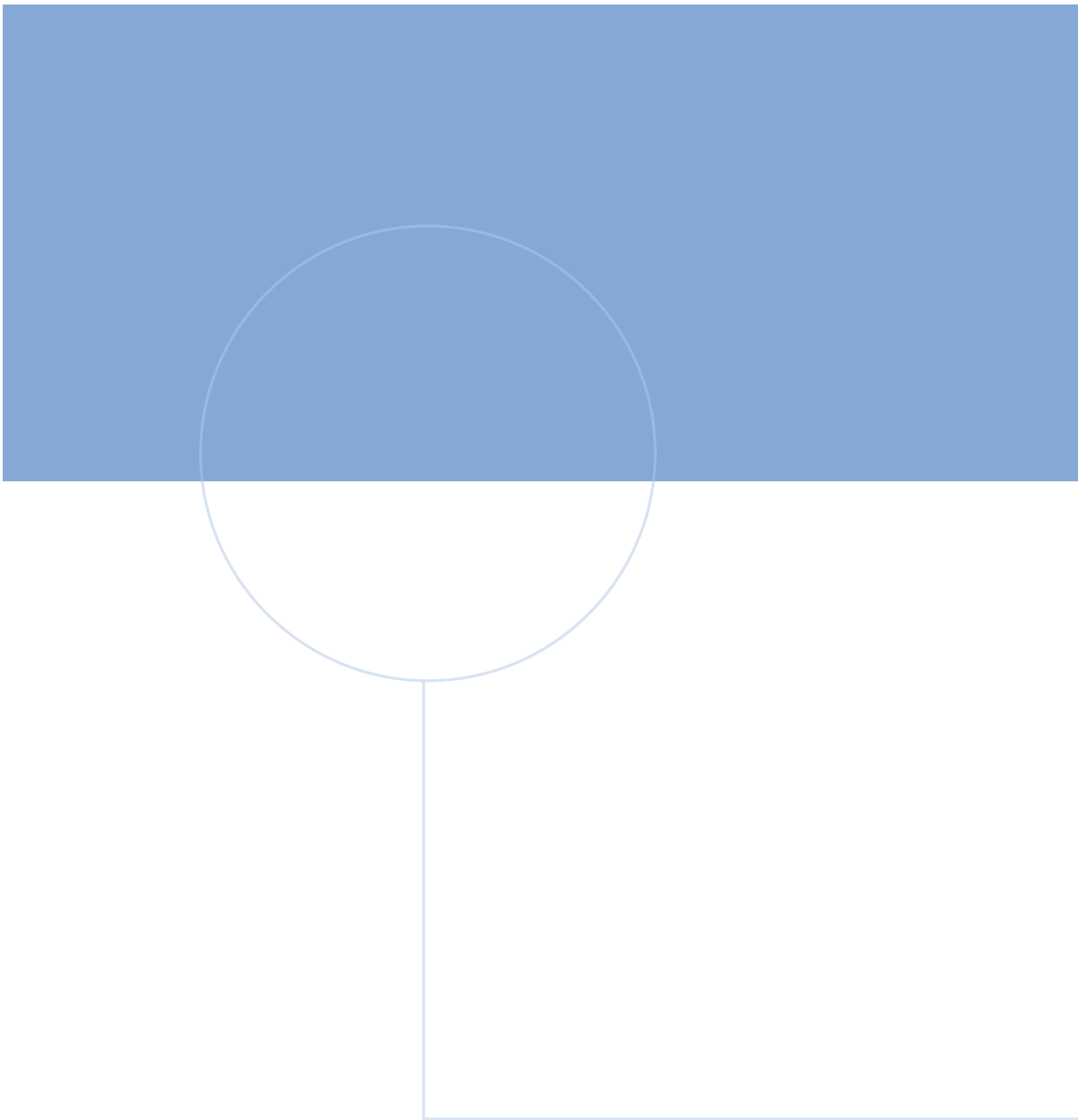
```
function rotated_angles = rotate_angles(angle_list, number_of_sensors)
    rotation_amount = ((number_of_sensors - 1) / 2) * (360 / number_of_sensors);
    rotated_angles = mod(angle_list + rotation_amount + 180, 360) - 180;
end
```

Code listing C.5: Python code for calculating the travel times for waves reflected up to to times from the plate edges.

```python
def get_travel_times(
    actuator: Actuator,
    sensor: Sensor,
    propagation_speed: float,
    surface: Plate,
) -> Tuple[np.ndarray, np.ndarray]:

    arrival_times = np.array([])
    travel_distances = np.array([])

    """Calculate the direct wave travel time"""
    direct_travel_distance = np.linalg.norm(actuator.coordinates - sensor.coordinates)
    travel_distances = np.append(travel_distances, direct_travel_distance)
    direct_travel_time = direct_travel_distance / propagation_speed
    arrival_times = np.append(arrival_times, direct_travel_time)

    EDGES = np.array(
        [
            surface.TOP_EDGE,
            surface.RIGHT_EDGE,
            surface.BOTTOM_EDGE,
            surface.LEFT_EDGE,
        ]
    )
    # Iterate thorugh all combinations of edges to reflect from
    for edge_1 in range(0, EDGES.size + 1):
        for edge_2 in range(0, EDGES.size + 1):
            if ignore_edge_combination(edge_1, edge_2):
                continue
            mirrored_source = find_mirrored_source(
                actuator,
                np.array([edge_1, edge_2]),
                surface
            )
            distance_to_sensor = np.linalg.norm(
                mirrored_source.coordinates - sensor.coordinates
            )
            time_to_sensors = distance_to_sensor / propagation_speed
            travel_distances = np.append(travel_distances, distance_to_sensor)
            arrival_times = np.append(arr ival_times, time_to_sensors)

    return arrival_times, travel_distances
```

Code listing C.6: Python code for finding the coordinates of mirrored sources.

```
def find_mirrored_source(
    actuator: Actuator,
    edges_to_reflect_from: np.ndarray,
    surface: Table or Plate,
) -> MirroredSource:

NO_REFLECTION = 0
mirrored_source = MirroredSource(actuator.coordinates)

for edge in edges_to_reflect_from:
    if edge == NO_REFLECTION:
        continue
    elif edge == surface.TOP_EDGE:
        mirrored_source_offset = np.array([0, 2 * (surface.WIDTH - mirrored_source.y)])
        mirrored_source.set_coordinates(mirrored_source.coordinates + mirrored_source_offset)
    elif edge == surface.RIGHT_EDGE:
        mirrored_source_offset = np.array([2 * (surface.LENGTH - mirrored_source.x), 0])
        mirrored_source.set_coordinates(mirrored_source.coordinates + mirrored_source_offset)
    elif edge == surface.BOTTOM_EDGE:
        mirrored_source_offset = np.array([0, -2 * mirrored_source.y])
        mirrored_source.set_coordinates(mirrored_source.coordinates + mirrored_source_offset)
    elif edge == surface.LEFT_EDGE:
        mirrored_source_offset = np.array([-2 * mirrored_source.x, 0])
        mirrored_source.set_coordinates(mirrored_source.coordinates + mirrored_source_offset)
return mirrored_source
```



NTNU

Norwegian University of
Science and Technology