MASTER THESIS IN ACOUSTICS

Characterization and finite element modelling of piezoelectric ceramic discs vibrating in air, for a frequency range including the first two radial modes

SVERRE KONGSRO FINSTAD

UNIVERSITY OF BERGEN

Department of Physics and Technology

August 2021

Preface

This work was done using an experimental setup based on work by Espen Storheim [1], and Ørnulf Svan Amundsen. The setup has been improved upon by other students at UiB, like Eivind Mosland [2], Rune Hauge [3], and Renate Grindheim [4].

I am very grateful for the help and guidance given by Renate Grindheim and Eivind Mosland. When learning how to use the measurement setup, and when it came time to post process the data gathered, their help was much appreciated.

I want to give my biggest thanks to my supervisors Per Lunde, Magne Vestrheim, and Mathias Sæther. Their guidance and assistance throughout the time put into this work has been great, and the feedback to my work has been really useful. Thank you for the weekly guidance meetings, even though most of them had to be done over Zoom.

I also want to thank the rest of the acoustics group at UiB. Their precence and motivation has helped a lot during long days at the lab. The help with soldering and machining from the UiB workshop is also greatly appreciated.

Finally, I want to thank my family and friends here in Bergen and back home in Porsgrunn. Their constant support has helped me keep both my motivation and mood up, during a year where the COVID-19 pandemic caused people to feel more isolated than ever before.

Sverre Kongsro Finstad, 13. August 2021, Bergen

Contents

Chapter 1

Introduction

1.1 Background and motivation

Ultrasonic measurement technology can be found on many places and industries, such as maritime work, building engineering, and medical technologies. In many of these industries, tools are used which relies on piezoelectric materials to be used as transmitters and receivers of ultrasonic waves. A good understanding of the characteristics and behaviours of these piezoelectric materials is essential when it comes to reduce uncertainty and increase accuracy of measurements, especially those used for fiscal measurements of resources like oil and gas [5, 6, 7]. Work has been done to map and simulate the behaviours and effects relevant for ultrasonic flow meters, for applications where the speed of sound in a gas needs to be known to a high degree of accuracy [8, 9]. The sound speed in gas is used, together with measurements such as pressure and temperature, to find the density and flow rate, which is important when calculating prices of the commodities transferred through the pipes.

The ultrasonic flow meter has been show to be a good alternative to other methods of acquiring the speed of flowing fluids in pipes, because of the high accuracy, low price, and non-invasiveness of the clamp-on method [10]. Other methods include turbine meters, orifice plate meters, and coriolis meters, but all these need to be placed in the flow loop, and therefore interfere with the flow, unlike the ultrasonic meter [11].

The application of ultrasonic measurement methods in industrial applications require more than a simple model of the transmitting and receiving instruments used [12]. A full model of the entire measurement system, called a system model, is required to achieve a greater understanding of the processes used. The ability to simulate the whole system, as well as individual parts, can lead to greater optimization and quality control of the equipment, more accurate measurements, and better correction of unwanted effects in your setup.

The system model is usually divided into several "modules", usually corresponding to the different equipment used, i.e. signal generators, amplifiers, mediums, or transmitting and receiving transducers [12]. The system model makes it so that the signal path through the equipment can be accurately modeled, and the effects of each "module" on the signal can be quantified [13].

Piezoelectric elements are often used as the main sensing component in transmitting or receiving transducers, but is usually only one part of the larger transducer construction. For example, matching layers and backing materials are used to increase the efficiency of the transducer [14], or change the matching characteristics to enable a better coupling to different fluid mediums. To achieve a good model for the whole transducer, a good understanding of the piezoelectric element characteristics and behaviours is important. Using models developed throughout the years, a good approximation of the characteristics of a piezoelectric element can be calculated, and parameters such as admittance, directivity beam pattern, and transmitter and receiver sensitivity can be simulated and determined before any construction is done.

Modelling and simulation of piezoelectric elements and transducers requires a high degree of control over the parameters of the element structure, as well as the surrounding materials and mediums. This can be achieved through the use of a finite element (FE) model. This method has been used for modelling and research of all kinds of structures for many years, and is well suited for work with ultrasonic waves, where parameters like displacement of a vibrating surface over time is particularly relevant [15].

1.2 Previous work

A good model for accurately predicting the behaviour of a transmit-receive ultrasonic measurement system has been sought after for many years. In 1977, a model was presented by Papadakis using transmission line theory, aiming to create a model that would compare well to measurements preformed using ultrasonic transducers [16]. Here, each part of the transducer is treated as a separate module of a larger 1-dimensional system model.

Later, the modelling software FLOSIM started seeing use to model measurement systems. This 1-D model can simulate signal transmission of a signal acoustic beam, with uniform flow in the medium. This method of simulation is presented in work by Lygre et al. [17].

In 1988, a paper by Collie et al. decribes a method for computing the transient voltage response of compressional wave ultrasonic transducers [18]. This method uses 1-D Mason model equivalent circuits, where the effects of the circuitry in the transmitter and receiver can be separated from the acoustical effects.

In 2018, a work by Sanabria et al. presented methods and results for calculations of sound fields produced by air-coupled ultrasound transducers, based on single-plane measurements [19]. By measuring the sound pressure in a plane parallel to the transducer surface, preferrably in the near field, the entire sound field could be calculated, using a closed form reradiation method combining the Rayleigh–Sommerfeld integral and time-reversal acoustics. This method worked well for both 2-D, and 3-D transducers, with a root mean square deviation for the sound field at around 2.5 % in the far field, and 5.5 % in the near field, for the frequency range 50-230 kHz.

Earlier works has been focused on modelling and mapping the directivity beam pattern of a transducer, and how it can affect measurements, for example how the sound generated off axis induce lamb waves in materials when the transducers are used for through-transmission measurements [20] [21]. The work by Benny et al. [20] uses a laser vibrometer to map the vibrating surface of the transducers, and predict the beam profile based on the measured data. This is compared with the measured pressure using an ultrasonic detector, and FEM simulations. These measurements are done in a 4 by 30 cm field in front of the radiating transducer, with a focus on the near field behaviours.

The work of Bashford et al. also use methods for measuring the sound field produced by transducers [21], but in this case the transducer is a capacitance transducer, and the focus of the work is to compare the sound field produced to a theoretical piston model of a radiating plane. These papers show a method for measuring the characteristics of the vibration pattern of piezoelectric transducers, which can be used to calculate the produced sound field. This is an indirect method of measurement of sound pressure, but has the benefit of not having to correct for a receiver in the near field, which causes problems with standing waves between the transmitter and receiver, and the fact that the receiver has a surface that is not point-like, so the received sound waves will not be plane in the near field.

Chillara et al. present measurements of directivity beam pattern and sound field of PZT piezoelectric transducers at the first four radial modes, in water [22]. In addition, the vibrational pattern of the transducer surface is examined, and compared with theoretical models.

The article also explores the application of clamping on piezoelectric transducers, to reduce side-lobe radiation of the transducer, and approach the behaviour of an ideal Bessel-beam pattern, which is non-diffractive, meaning the amplitude does not decrease with distance. This effect can be used to achieve a higher accuracy of measurements using transducers vibrating in the radial modes.

Another article by Chillara et al. shows the measurements and simulations of piezoelectric elements vibrating in the first four radial modes [23]. The vibrational surface pattern is measured using a laser doppler vibrometer, while the directivity, axial amplitude, and sound field pattern was measured using a hydrophone. These experiments show a good compliance between the analytical and experimental results, and use many of the measurement methods used in this work.

Note that most of the work talked about in this section is preformed on assembled piezoelectric transducers, or other types of transducers entirely. Although many of the methods and models used in these earlier works are similar to what is done in this work, here the focus is on a piezoelectric element, which is only a part of a larger piezoelectric transducer. During this work, few papers has been found that study and characterize piezoelectric elements on their own.

1.2.1 Previous work in-house at UiB

A lot of the work done in house has been focusing on mapping the behaviours of piezoelectric ceramic discs and transducers, including several master theses and PhDs. In [1], the diffraction effects and diffraction correction needed when working with non-uniformly vibrating transducers, and how it compares with the widely used baffled piston diffraction correction, was investigated. Measurements of the pressure produced by piezoelectric transmitters were compared with FE simulations, using both diffraction correction methods, and the results were compared.

The FE simulations were preformed using a program for finite element modelling of piezoelectric transducers, (FEMP), developed by Kocbach at the University of Bergen in collaboration with the Christian Michelsen Research Institute [24].

Through his work, Mosland developed and implemented a modified three-transducer reciprocity calibration method, where correction factors for absorption in air, diffraction effects, and transmitting and receiving electronics were included [2]. This method was tested through the use of piezoelectric element transmit-receive pairs, as well as piezoelectric transducers constructed in-house, yielding values for transmitting voltage responses S_V, and free-field open-circuit receiving voltage sensitivities M_V. The frequency range studied was 50 kHz to 300 kHz.

In collaboration with Mosland [2], Hauge did work to develop and integrate a finite element based linear system model for the measurement setup used by both authors [3]. The system model allowed for simulation and analysis of individual components of the measurement system, covered in this work in Section. 2.3. Modeling of transmitting cables was also implemented, through the use of transmission line models. The results were verified through measurements compared with corresponding finite element simulations.

In the work by Søvik, the recorded phase response of the transmit-receive pair was examined, and new corrections based on the non-uniformity of the vibrations of the transducer was included [25]. Problems arose when comparing phase data with simulations, due to the phase being very sensitive to changes in position along the axis of sound travel.

In the work by Grindheim, the transfer functions used to calculate signal amplitude at different points in the signal path was examined, and the results compared to finite element simulations [4]. Comparisons between calculated and simulated transfer functions gave good results, with the largest deviations being seen around the second resonance peak at

the first radial mode (R1). Difficulties in positioning of the element was observed, with the misalignment causing uncertainties in the amplitude of the transfer function $H^{VV}_{15_{open}}$ around the second radial mode (R2).

1.3 Objectives

The objectives of this work is to measure and characterize the electrical and acoustical characteristics of a piezoelectric ceramic disk of type Pz27, across a frequency spectrum covering the first two radial modes, and to compare these measurements to FEM simulations. The electrical parameters such as admittance, conductance and susceptance of a sample of elements are measured, in a frequency range from 1 kHz to 300 kHz, encompassing the two first radial modes of the elements. The acoustic characteristics of one of these elements will also be studied and compared to simulations. Here, parameters such as the source sensitivity, directivity, on axis pressure, and sound pressure field will be measured using a microphone as a receiver.

Finite element modelling (FEM) simulations of the piezoelectric ceramic discs radiating in air are preformed. This data is compared to the measurements of the element admittance, directivity beam pattern, transmitter sensitivity, on-axis pressure, and sound field. The FEM software used is FEMP 6.1, developed by Jan Kocbach [24].

Methods of converting signal pulses to sound pressure values are examined in this work. This includes methods for finding peak to peak voltage of the received signals, as well as corrections for signal losses, either due to losses in air, or losses due to impedances in wires and instruments used. A method for calculating microphone receiver sensitivity is also examined, based on the use of a pistonphone calibration and correction data.

This work is meant to be a part of a combined effort in house at UiB, to characterize and map the electrical and acoustical behaviours of different piezoelectric materials, and different methods for transducer construction.

1.4 Thesis outline

Chapter 2 covers the theoretical material and background needed to understand and perform the measurements in this work. Chapter 3 covers the electrical and acoustical experimental setup, and methods of collecting and processing data. Chapter 4 covers the simulation methods used in this work, and how the FE simulations are set up. Chapter 5 covers the results from the simulations and measurements preformed, and present discussions based on the data gathered. Chapter 6 covers the conclusions that can be pulled from the results in chapter 5, and show suggestions for further work related to the experiments performed in this work.

Chapter 2

Theory

In this chapter, the theoretical basis and equations used in this work is presented. A short summary of the theory behind modes and resonance frequencies is presented in Section 2.1. Section 2.2 goes over the coordinate system used for measurement- and simulation data presented. Section 2.3 covers the theory behind the system model used in this work, highlighting the different modules in the measurement setup. The theory behind transmitting properties of the element, and receiving properties of the receiving microphone system is covered in Sections 2.4 and 2.5 respectively. The different corrections applied to the measured and simulated results are shown in Section 2.6. The theory behind the transmission line models used to calculate the electrical corrections are shown in Section 2.7. Finally, a short summary of the theory behind the finite element modelling used in this work is presented in Section 2.8.

2.1 Modes in piezoelectric ceramic discs

The modes of a radiating piezoelectric element are defined as the frequencies at which standing waves are induced in the element, that match the dimensions of the element [14]. For a piezoelectric disk with a diameter several times the thickness, several types of resonances can be observed, but two types of resonances are often focused on, these being the thickness extensional mode (TE-mode), and radial extensional mode (R-mode) [26]. The different types of modes are expanded upon by Kocback [24].

At these frequencies, the element power output is at its maximum amplitude within the frequency range around the resonance. The radial extensional R-mode, which is focused on in this work, corresponds to the frequencies at which standing waves are induced in the radial direction in the element. The thickness extensional mode is the equivalent, but the standing waves are induced in the direction of the thickness of the element [27][14].

When describing resonance modes of piezoelectric elements, two frequencies are defined for each mode. These are the series resonance frequency, f_s, and the parallel resonance frequency, f_p. At the series resonance, the maximum surface displacement, and therefore the optimal transmitting performance, is observed, while at the parallel resonance, the maximum receiving voltage sensitivity of a piezoelectric element is observed, for a particular vibrational mode [28]. These resonances corresponds with the minimum and maximum impedance of the element, for the series and parallel resonance frequency respectively [27]. The admittance, which is the inverse of the impedance, is defined as

$$Y(f) = G(f) + iB(f) = \frac{1}{Z(f)}, \tag{2.1}$$

where Y(f) is the admittance in Siemens, G(f) is the conductance in Siemens, B(f) is the susceptance in Siemens, and Z(f) is the impedance in Ohms.

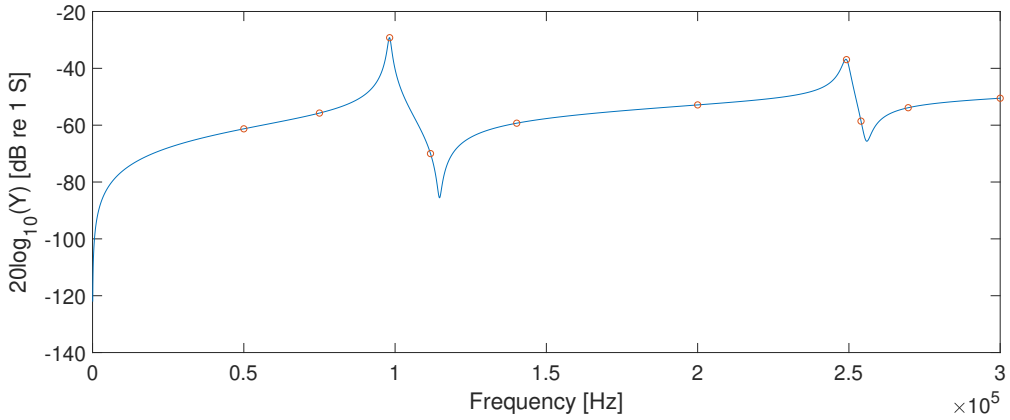


FIGURE 2.1: Simulated admittance of a standard 20x2 element, with markings at frequencies corresponding to frequencies shown in Figs. 2.2 and 2.3.

At the series resonance, the conductance is at its maximum value, the susceptance goes to zero, and the maximum amount of mechanical energy is transmitted to the radiating medium [28]. In this work, the piezoelectric element characteristics are examined in a frequency range covering the two first R-mode resonances, i.e. from 0 to 300 kHz. In Figs. 2.2 and 2.3 displacement patterns and directivities for piezoelectric discs at multiple frequencies in the range covering the first two radial modes are shown. Figures sourced from supplementary material of [29]. The corresponding frequencies are marked in Fig. 2.1 to show the connection between vibrational pattern and admittance. Note that the frequencies in Figs. 2.2 and 2.3 were chosen based on a simulated transfer function H_{06}^{VV} of a transmit-receive system with piezoelectric elements as both transmitter and receiver. In this work, a piezoelectric element is only used as a transmitter, so the admittance is plotted instead. This means that the frequencies shown does not correspond to the exact series and parallel resonances of the piezoelectric element used in this work.

2.2 Coordinate system

In this work, two different coordinate systems are used when talking about position relative to the centre of the transmitting element surface. The first is a Cartesian coordinate system, where the x- and y-axis form the plane corresponding to the plane of the transmitter surface. The x-axis is vertical, and the y-axis is horizontal, see Fig. 2.4. The z-axis represents the sound axis, and the value z is used in this work as separation between transmitter and receiver. The Cartesian coordinates are used in the positioning of the element, and when the on-axis pressure is discussed.

The second coordinate system is a spherical coordinate system, used when discussing the directivity beam pattern of the element. Here, pressure $p(r, \theta)$ at a point is written as a function of the distance r from the source centre to the point, and the angle θ off the sound axis (z-axis), see Fig. 2.4. Normally, a spherical coordinate system also defines ψ as the rotational angle around the z-axis, but in this work, the pressure field produced by the piezoelectric element is assumed to be symmetrical around the z-axis, so the coordinate ψ is omitted. It can be stated that $\psi = 0$ degrees for all measurements, since the measurements in later sections are preformed in the yz-plane.

The origin of both coordinate systems used is set at the centre of the transmitter front surface.

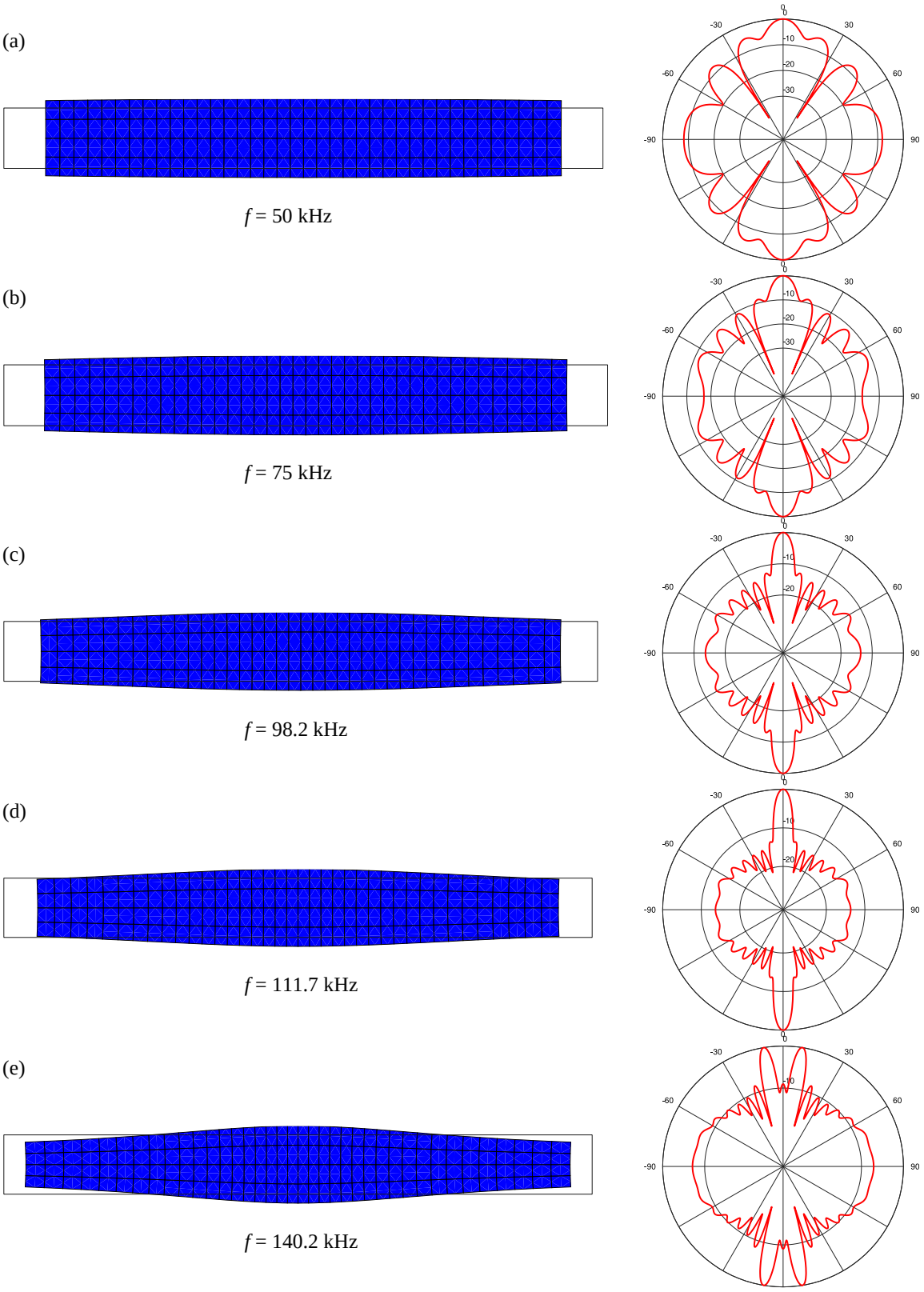


FIGURE 2.2: Exaggerated displacement pattern, and the corresponding directivity beam pattern, calculated using FEMP, from 50 to 140.2 kHz. Figure from supplementary material of [29].

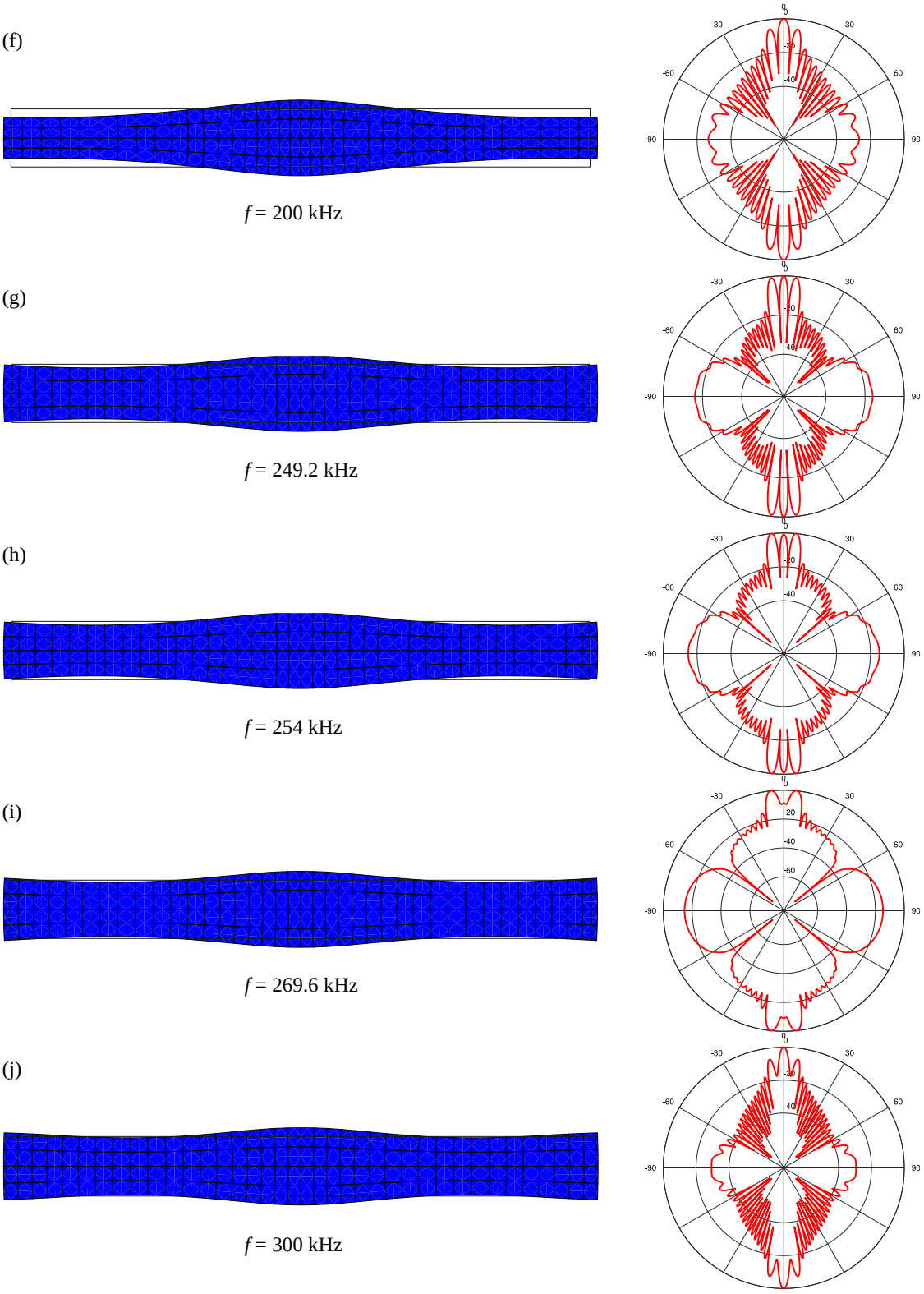


FIGURE 2.3: Exaggerated displacement pattern, and the corresponding directivity beam pattern, calculated using FEMP, from 200 to 300 kHz. Figure from supplementary material of [29].

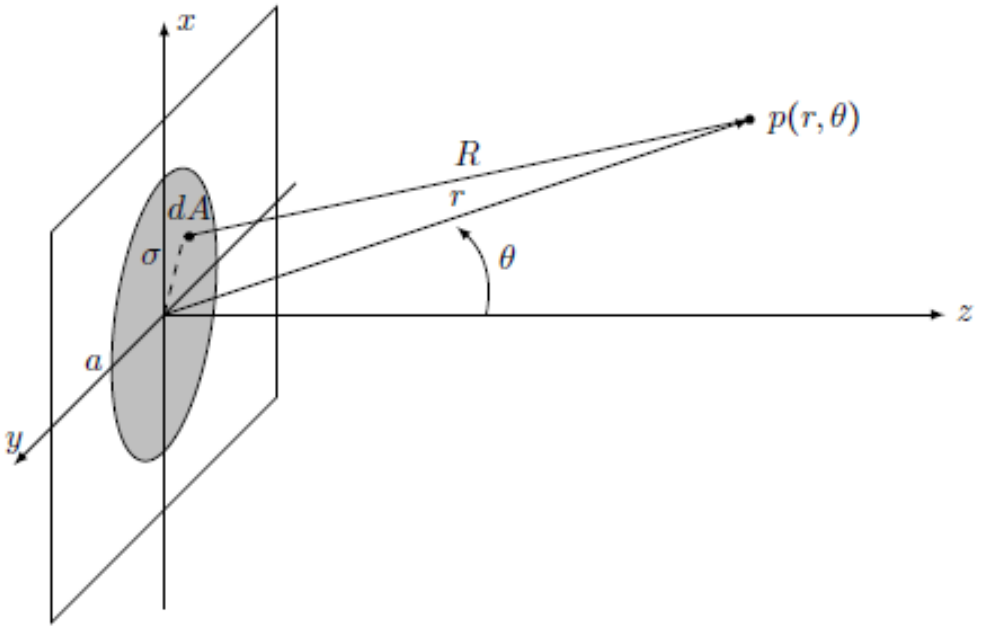


FIGURE 2.4: Illustration of a circular plane transmitter mounted in a rigid baffle. The coordinate values defined here is used in this work. ψ is not shown, as the source is assumed rotationally symmetrical around the z-axis.

2.3 System model

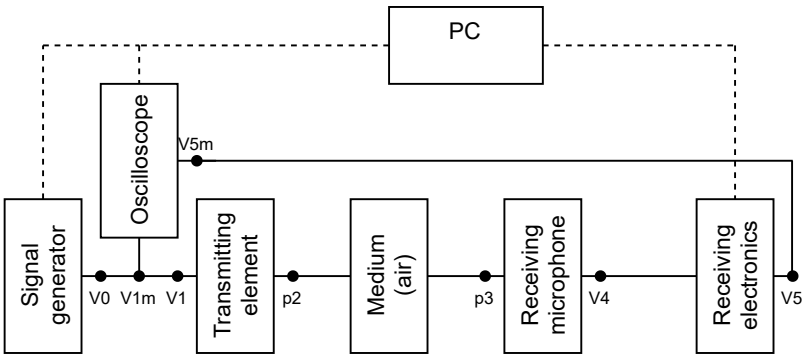


FIGURE 2.5: System model diagram of setup used in this work. Solid lines show signal path, while dotted line show connections to PC.

The measurement setup used can be represented using a system model [12]. The different components are represented by blocks, or "modules", connected by nodes, and the values in each node can be modeled using transfer functions for the different inputs and outputs, see Fig. 2.5.

2.3.1 Signal generator

The signal generator generates the sinusoidal bursts that is sent to the transmitting element, with a peak-to-peak signal voltage V_0 as its output.

2.3.2 Oscilloscope

The oscilloscope is connected in parallel with the transmitting element to the signal generator. The peak-to-peak voltage V_{1m} is the voltage measured at the oscilloscope input, which has a termination resistance of 1 $M\Omega$. The oscilloscope is also connected to the receiving electronics. V_{5m} is the recorded peak-to-peak output voltage of the system, and is used to calculate the sensitivity of the entire receiver electronics system.

2.3.3 Transmitting element

The peak-to-peak voltage V_1 is defined in this work as the generated voltage V_0 after the signal has been transmitted through the coaxial cables connecting the generator to the element. V_1 is the actual voltage over the electrodes of the element, and induces the vibrations in the element, with a surface displacement determined by the amplitude and frequency of the signal. This surface displacement then produces a pressure p_2 at the transducer surface, dependant on the displacement of the element surface [27].

2.3.4 Medium

The transmitting medium in this work is air, and here the pressure signal p_2 is affected by spherical attenuation and losses due to sound attenuation, covered in Section 2.6. The pressure p_2 is at the transmitting element surface, while the pressure p_3 is the pressure calculated to be at the microphone receiver surface. Normally, the free-field pressure over the receiver surface would be calculated by applying a diffraction correction to the recorded value, but in this work this is not done, due to using a microphone instead of an equivalent piezoelectric element as receiver. From now on, the value p_3 is presented as the pressure calculated at the microphone front surface, using the recorded signal voltage from the microphone.

2.3.5 Receiver

The receiver is in this work a Bruel & Kjær 4138-A15 microphone. The microphone records the pressure p_3, and outputs a voltage V_4. This voltage is not recorded in this work, but is defined as the input voltage at the first instrument in the receiver electronics, which in this case is the amplifier.

2.3.6 Receiving electronics

The receiving electronics in this work consists of an amplifier and a band-pass filter, connected by coax-cables. The receiving electronics takes the input voltage from the microphone V_4, and converts it into the output voltage V_5. In this work, this value is defined as the output voltage of the filter. The value V_{5m} is the signal voltage recorded by the oscilloscope, and is defined as the value of V_5 after the losses of the cable connecting the filter and oscilloscope is applied. In this setup, the oscilloscope also acts as the circuit terminal, with a resistance of 1 $M\Omega$ at the receiver input.

2.4 Transmitting properties of piezoelectric element

The transmitting sensitivity of the piezoelectric element describes the relation between the input voltage over the element, and the generated sound pressure. This value gives the on-axis pressure amplitude at a reference distance in the far field, usually 1 meter. For a medium with no loss, the transmitting sensitivity is given by the equation

$$S_V(f) = \frac{p(z = d_0, \theta = 0, f)}{V(f)} = |S_V(f)|e^{i\phi S_v} \tag{2.2}$$

Here, f is the frequency of the input signal, $p((z = d_0, \theta = 0, f)$ is the free-field on-axis sound pressure produced at the distance d_0, d_0 is the reference distance, and $V(f)$ is the voltage amplitude of the signal applied to the transmitting element. $|S_V(f)|$ is the magnitude of the transmitting sensitivity, and $e^{i\phi S_v}$ is the phase.

2.5 Receiving properties of microphone

2.5.1 Calibration of microphone using pistonphone

The receiver sensitivity of the microphone used in this work is already given by the manufacturer, but earlier works has shown that this value is not applicable to all circumstances. Things like air pressure and temperature of the room affects the sensitivity, so the value needs calibration to this specific setup. The calibration is performed by using a Bruel & Kjær Type 4228 pistonphone. The pistonphone produces a known SPL, and includes a pressure gauge for adjusting the recorded SPL. The temperature of the room is also taken, and another correction is applied to the pistonphones SPL. The microphone is inserted into the pistonphone, and the peak-to-peak output voltage of the microphone is recorded using the oscilloscope. The resulting receiver sensitivity can then be calculated as

$$M_V(f) = \frac{V_5 m(f)}{p_3(f)} = |M_V(f)|e^{i\phi M_V} \tag{2.3}$$

where

$$|M_V| = \frac{\frac{V_{pp}^{rec}}{2\sqrt{2}\cdot 10^{\frac{G}{20}}}}{10^{\frac{SPL_\alpha}{20}} \cdot p_{ref}} = \frac{V_{pp}^{rec}}{2\sqrt{2}p_{ref}10^{\frac{G+SPL_\alpha}{20}}} \tag{2.4}$$

Since the phase of the sound pressure is not considered in later sections, the phase value $e^{i\phi M_V}$ is omitted from the receiver sensitivity, and the modulus is defined as the sensitivity $|M_V| = M_V$. $V_5 m$ is measured output voltage of the receiver electronics, which in this case is the microphone, amplifier, band-pass filter, and cables used to connect the instruments together. Ideally, the free-field pressure over the receiver surface would be used to calculate M_V, but due to the inability to use diffraction correction with the microphone, p_3 is defined as the actual pressure recorded by the microphone. $V_{pp}^{rec} = V_{5m}$ is the peak-to-peak voltage recorded off the oscilloscope, when the microphone is recording the pistonphone signal, and the signal is passed through the receiver electronics. SPL_α is the calculated sound pressure level of the pistonphone, adjusted for temperature and pressure. G is the dB gain of the amplifier. p_{ref} is the reference pressure used to calculate the SPL. M_V is then defined as the receiver sensitivity for the whole receiver system, including the microphone, amplifier, filter, and all cables used between. The voltage value is the peak-to-peak value of the signal, while the SPL produced by the pistonphone is given as

$$SPL = 20log_{10}(\frac{p_{rms}}{p_{ref}}) \tag{2.5}$$

To calculate the receiver sensitivity M_V, the measured V_{pp} is converted to the effective value V_{rms}, by dividing the value by $2\sqrt{2}$. The M_V calculated can be used to calculate p_{rms} based on the signal V_{rms}, but in this work, the peak to peak signal voltage V_{pp} is more commonly used. The same value of M_V can be used to calculate pressure based on V_{pp}, but the

resulting pressure values are the peak to peak pressure p_{pp} of the wave, and will for example need to be converted to p_{rms} when calculating SPL for comparison, using the equation

$$p_{rms} = \frac{p_{pp}}{2\sqrt{2}} \tag{2.6}$$

This method of calibration is taken from E. Storheim [1].

2.6 Corrections

Two types of corrections are applied in this work, one to account for losses in cables, and one to correct for losses in pressure due to attenuation in the air [30]. This is done by finding the values of H_{1m1}^{VV} and C_α respectively. C_α is the correction for sound absorption in air, and H_{1m1}^{VV} describes the difference between the signal voltage amplitude recorded by the oscilloscope V_{1m}, and the actual signal voltage applied to the piezoelectic element electrodes V_1. This transfer functions will be explored more in Section 2.7.

The theoretical lossless pressure value p_i, used for comparison with simulations in later sections, is defined as

$$p_i = p_t C_\alpha \tag{2.7}$$

where p_t is the calculated pressure p_3, received by the microphone, and C_α is the sound absorption correction factor.

The transfer function H_{1m1}^{VV} is defined as

$$H_{1m1}^{VV} = \frac{V_1}{V_{1m}}, V_1 = V_{1m} H_{1m1}^{VV} \tag{2.8}$$

where V_1 is the actual input voltage to the transmitting element, and V_{1m} is the input voltage measured at the oscilloscope, see Section 2.3.

2.6.1 Absorption in air

The medium, in this case air, is assumed to be lossless in the theoretical calculations to simulate sound propagation using FEMP, so the absorption seen in the physical experiments needs to be accounted for. Here, the measured output voltage of the microphone can be adjusted to correspond with a lossless scenario. Many factors contribute to the sound waves loss of energy through air, but in this work, the atmospheric absorption will be focused on. Factors like refraction, non-linear propagation effects, and scattering by movements and turbulence in the air is not accounted for [30]. The absorption used in this work is calculated in accordance to the "American National Standard" [30], assuming the sound waves are plane-wave and single frequency. The pressure wave propagating in air is attenuated by the absorption in air, and the pressure amplitude factoring in absorption is calculated as

$$p_t = p_i e^{-0.1151 \alpha_{dB/m} s} \tag{2.9}$$

where p_t is the attenuated pressure amplitude at distance s meter, and p_i is the original pressure amplitude. $\alpha_{dB/m}$ is the pure-tone sound-attenuation coefficient in decibels per meter. The attenuation coefficient $\alpha_{dB/m}$ is a constant found using the temperature T, the ambient air pressure p, the relative humidity h_{rel}, and the relaxation frequencies of oxygen and nitrogen molecules f_{rO} and f_{rN}.

Equations to calculate these absorption coefficients are given in [30]. Due to equipment failure, the relative humidity and pressure could not be recorded for each measurement, so a standard value for both will be used to calculate $\alpha_{dB/m}$.

The term for the absorption correction factor needed is given as

$$C_\alpha = 10^{\alpha_{dB/m}s/20} \tag{2.10}$$

and will act as a transfer function from lossless measured voltage to lossy measured voltage. This is shown as

$$\frac{V_i}{V_t} = \frac{p_i}{p_t} = e^{\alpha_{dB/m}s/20log_{10}(e)} = 10^{\alpha_{dB/m}s/20} \tag{2.11}$$

where V_i is the theoretical output voltage generated by the microphone in a lossless system, and V_t is the actual generated output voltage in the real lossy system. p_i and p_t are the corresponding pressures at the microphone recorder surface inducing the voltages. p_i is used for comparison with simulations, so it is calculated as

$$p_i = p_t \cdot C_\alpha \tag{2.12}$$

where p_i is the theoretical pressure in a lossless system, p_t is the actual recorded pressure, and C_α is the absorption correction factor.

2.7 Electronics

2.7.1 Cables

It is assumed in this work that the impedance of the cables used in this setup can have significant effects on the transmitted signals, and has to be accounted for when calculating the signal voltage at the transmitter V_1, which is used later to scale the simulations. The coaxial cables used in the setup can be modeled as ideal uniform transmission lines, see Fig. 2.6.

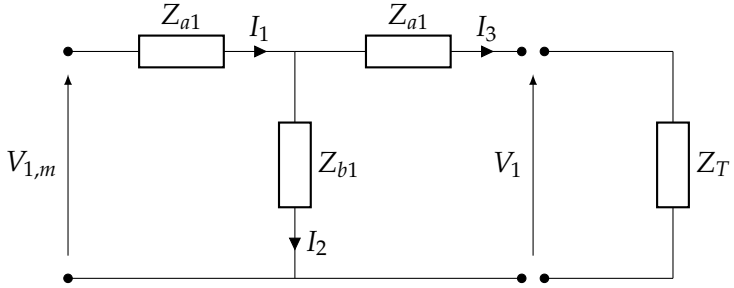


FIGURE 2.6: A circuit description of the coaxial cable connecting the oscilloscope to the transmitting transducer.

The impedances determining the cables behaviour can be expressed as [14]

$$Z_a = iZ_0 \tan\left(k_{em}\frac{l}{2}\right) \tag{2.13}$$

$$Z_b = \frac{Z_0}{i\sin\left(k_{em}l\right)} \tag{2.14}$$

where Z_0 is the characteristic impedance of the cable, k_{em} is the electromagnetic wavenumber, and l is the length of the coaxial cable in meters. The characteristic impedance and the wavenumber can be calculated as

$$Z_0 = \sqrt{\frac{L_x}{C_x}} \tag{2.15}$$

$$k_{em} = \omega\sqrt{L_x C_x} \tag{2.16}$$

where ω is the angular frequency of the signal, L_x is the inductance per meter of the cable, and C_x is the capacitance per meter.

2.7.2 Transmitting electronics

The transmitting electronics in this setup (see Section 2.3) consists of a signal generator, an oscilloscope, and the transmitting piezoelectric element. These are connected through coaxial cables, which are explained in Section 2.7.1. The signal voltage measured at the oscilloscope is denoted $V_{1,m}$, and the transfer function describing the transition from measured voltage to actual voltage over the piezoelectric element is defined as

$$H_{1m1}^{VV} = \frac{V_1}{V_{1m}} \tag{2.17}$$

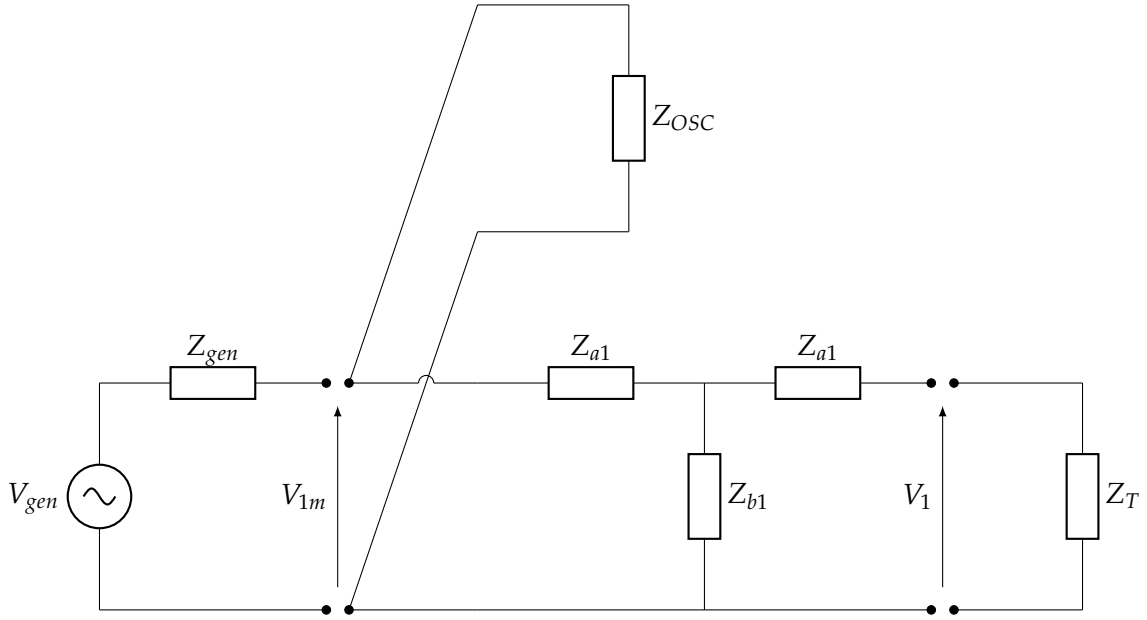


FIGURE 2.7: A circuit description of the transmitting electronics. The signal generator is described as a Thévenin generator, connected to the oscilloscope, with a branch connected to the transmitting element. The connecting cables are described as ideal transmission lines.

To find the voltage values to define H_{1m1}^{VV}, Kirchhoffs voltage and current laws are used.

$$V_{1m} = I_1 Z_{a1} + I_2 Z_{b1} \tag{2.18}$$

$$V_1 = -I_3 Z_{a1} + I_2 Z_{b1} \tag{2.19}$$

$$V_1 = I_3 Z_T \tag{2.20}$$

$$I_1 = I_2 + I_3 \tag{2.21}$$

I_1, I_2 and I_3 are the currents labeled in Fig. 2.6. The Kirchhoff equations above can be rearranged to give equations for I_2 and I_3.

$$I_2 = \frac{V_1}{Z_{b1}} + I_3 \frac{Z_{a1}}{Z_{b1}} \tag{2.22}$$

$$I_3 = \frac{V_1}{Z_T} \tag{2.23}$$

To calculate H_{1m1}^{VV}, we need expressions for the voltages in Fig. 2.7 dependant only on the impedances in the system. The equation for the relation V_1/V_{1m}, where the currents are eliminated, can be found using the equations above.

$$V_{1m} = I_2(Z_{a1} + Z_{b2}) + I_3 Z_{a1} \tag{2.24}$$

$$V_{1m} = \left(\frac{V_1}{Z_{b1}} + I_3 \frac{Z_{a1}}{Z_{b1}}\right)(Z_{a1} + Z_{b1}) + I_3 Z_{a1} = V_1 \frac{Z_{a1} + Z_{b1}}{Z_{b1}} + I_3 \frac{Z_{a1}^2 + 2Z_{a1}Z_{b1}}{Z_{b1}} \tag{2.25}$$

The resulting ratio V_1/V_{1m} can then be found by replacing I_3, and rearranging.

$$\frac{V_1}{V_{1m}} = \frac{Z_{b1}Z_T}{Z_T(Z_{a1} + Z_{b1}) + (Z_{a1} + Z_{b1})^2 - Z_{b1}^2} \tag{2.26}$$

The transfer function converting the measured signal voltage to the actual signal voltage over the element electrodes is defined as the ratio V_1/V_{1m}

$$H_{1m1}^{VV} = \frac{V_1}{V_{1m}} = \frac{Z_{b1}Z_T}{Z_T(Z_{a1} + Z_{b1}) + (Z_{a1} + Z_{b1})^2 - Z_{b1}^2} \tag{2.27}$$

The impedance of the transmitting element Z_T is dependant on frequency, and can be either measured or simulated using FEMP. In this work, both is done, but the physical measurements are used for calculating the transfer function H_{1m1}^{VV}.This transfer function can now be used to calculate the signal voltage at the piezoelectric element V_1, see Eq. 2.17.

2.7.3 Receiving electronics

In this setup, the receiving electronics consists of a microphone with an integrated preamplifier, connected an external amplifier, a frequency filter, and then to the oscilloscope for measurement. The microphone connected to the amplifier is a Bruel & Kjær 4138-A-015, and has an incorporated preamplifier. The receiver sensitivity of the microphone is provided in the documentation, but in this work, the microphone sensitivity is found using calibration methods using a pistonphone, see Section 2.5.1. This receiver sensitivity value M_V takes into account all the components listed above. Both the amplifier and filter has a flat frequency response in the frequency range used in this work, so only the correction of the microphone frequency dependency needs to be applied to M_V, to get the receiver sensitivity for all frequencies in the range 50 kHz to 180 kHz, $M_V(f)$. Even though it would be possible to calculate the correction needed to find the microphone output voltage V_5, it is not necessary in this work as the oscilloscope output voltage V_{5m} is used to calculate pressure.

2.8 Finite element modeling

In this work, the electrical and acoustical properties of a piezoelectric element is simulated and analyzed using finite element modeling. The simulation software used is the FE model

Finite Element Modeling of Piezoelectric structures 6.1 (FEMP 6.1). What follows is a short summary of finite element theory, which is expanded upon in [24].

In this work, the structure being worked with is a piezoelectric disk with electrodes on the flat sides, polarized in the thickness direction, and radiating in a fluid, in this case air. In the simulated structure, the electrodes are omitted, and a specified voltage is applied directly to the element surface. The element is assumed to have cylindrical symmetry, so the simulation models a 2D slice of half of the element, from the center point to the edge, and a half circle of the surrounding fluid (see Section 4.4). This simulation is then mirrored along the z-axis, through the center of the disk in the thickness direction, to create the full 2D simulation. The simulation can also be mirrored 360 degrees around the z-axis, to show the whole 3D setup. The element and fluid is partitioned into a desired number of smaller elements. The fluid is split into finite and infinite elements, so the sound pressure fields can be extrapolated as far out as desired. Nodes along these elements are used as points for calculating the different parameters in the simulation, such as the voltage, pressure, and shift in position over time. These parameters can then be interpolated to any point inside each element, based on the node values. The finite element equations for a piezoelectric disk radiating in an infinite fluid is given as

$$-\omega^2 \begin{bmatrix} M_{uu} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -M_{\psi\psi} \end{bmatrix} \begin{Bmatrix} \hat{u} \\ \hat{\phi} \\ \hat{\psi} \end{Bmatrix} + i\omega \begin{bmatrix} 0 & 0 & C_{u\psi} \\ 0 & 0 & 0 \\ C_{\psi u} & 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{u} \\ \hat{\phi} \\ \hat{\psi} \end{Bmatrix} + \begin{bmatrix} K_{uu} & K_{u\phi} & 0 \\ K_{\phi u} & K_{\phi\phi} & 0 \\ 0 & 0 & -K_{\psi\psi} \end{bmatrix} \begin{Bmatrix} \hat{u} \\ \hat{\phi} \\ \hat{\psi} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -Q \\ 0 \end{Bmatrix} \tag{2.28}$$

The definition of the variables in the FE equation is found in [24], but a short description can be found in Table. 2.1.

TABLE 2.1: Description of the variable matrices used in Eqs. 2.28 and 2.29.

Matrix variable	Description
$[M_{uu}]$	Global Mass Matrix
$[M_{\psi\psi}]$	Global fluid mass matrix
$[C_{u\psi}]$	Global fluid/structure coupling matrix
$[C_{\psi u}]$	Global fluid/structure coupling matrix
$[K_{uu}]$	Global stiffness matrix
$[K_{u\phi}]$	Global piezoelectric stiffness matrix
$[K_{\phi u}]$	Global piezoelectric stiffness matrix
$[K_{\phi\phi}]$	Global dielectric stiffness matrix
$[K_{\psi\psi}]$	Global fluid stiffness matrix
$\{Q\}$	Global charge vector
$\{\hat{u}\}$	Global displacement vector
$\{\hat{\phi}\}$	Global electric potential vector
$\{\hat{\psi}\}$	Global fluid velocity potential vector
ω	Angular frequency

The FE equations (2.28) is transformed to so-called H-form, where the matrices are rearranged to H-form, to lessen the computational load needed to solve the equations.

$$-\omega^2 \begin{bmatrix} M_{uu} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -M_{\psi\psi} \end{bmatrix} \begin{Bmatrix} \hat{u} \\ V \\ \hat{\psi} \end{Bmatrix} + i\omega \begin{bmatrix} 0 & 0 & C_{u\psi} \\ 0 & 0 & 0 \\ C_{\psi u} & 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{u} \\ V \\ \hat{\psi} \end{Bmatrix} + \begin{bmatrix} H_{uu} & H_{u\phi} & 0 \\ H_{\phi u} & H_{\phi\phi} & 0 \\ 0 & 0 & -K_{\psi\psi} \end{bmatrix} \begin{Bmatrix} \hat{u} \\ V \\ \hat{\psi} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -I/(i\omega) \\ 0 \end{Bmatrix} \tag{2.29}$$

H_{uu}, $H_{u\phi}$, $H_{\phi u} = H_{u\phi}^T$, and $H_{\phi\phi}$ are given as Eqs. (3.190) - (3.192) in [24], respectively. The functions for admittance of a piezoelectric element and pressure produced in the fluid can be calculated by manipulating the Eq. (2.29), see [24] for detailed calculation. The third equation in Eq. (2.29) can be written as

$$\omega^2[M_{\psi\psi}]\{\hat{\psi}\} + i\omega[C_{\psi u}]\{\hat{u}\} - [K_{\psi\psi}]\{\hat{\psi}\} = 0 \tag{2.30}$$

where the global fluid velocity potential vector is given as

$$\{\hat{\psi}\} = -i\omega(-[K_{\psi\psi}] + \omega^2[M_{\psi\psi}])^{-1}[C_{\psi u}]\{\hat{u}\}. \tag{2.31}$$

The electrical admittance of the element can then be expressed as [24]

$$Y = i\omega[\{H_{u\phi}\}^T[D]^{-1}\{H_{u\phi}\} - H_{\phi\phi}] \tag{2.32}$$

where the matrix [D] is expressed as [24]

$$[D] = [H_{uu}] - \omega^2[M_{uu}] + \omega^2[C_{u\psi}](-[K_{\psi\psi}] + \omega^2[M_{\psi\psi}])^{-1}[C_{\psi u}]. \tag{2.33}$$

The impedance of the element is found through the relation

$$Z_T = \frac{1}{Y} \tag{2.34}$$

For a time harmonic scenario where the time dependency is assumed to be $e^{i\omega t}$, acoustic pressure in the fluid can be found using the equation

$$p = -i\omega\rho_f\psi \tag{2.35}$$

where ψ is the velocity potential expressed in Eq. (2.31), and ρ_f is the density of the fluid. The pressure values are given as complex numbers, and by taking the absolute value of the pressure matrices given by FEMP, the pressure amplitudes in the fluid is found. This pressure amplitude is used directly to show on-axis pressure, near- and far-field pressure on and off axis, and directivity, which is given as

$$D(\theta, f) = |p(r = 1\ m, \theta, f)| \tag{2.36}$$

where r is the distance from the centre of the piezoelectric element surface to the measurement point, f is the frequency of the simulated signal, and θ is the angle off the sound axis, which is defined as $\theta = 0$. Directivity in FEMP can be calculated at any value of r, but the pressure value is automatically interpolated to what the value would be at r = 1 m.

The transmitter sensitivity of the element is also calculated by FEMP using the pressure values found in Eq. (2.35), expressed as

$$S_V(f) = \frac{p(z = d_0, \theta = 0, f)}{V}, \tag{2.37}$$

where d_0 is the desired calculation distance, which should be in the far field, θ is the angle off the sound axis, f is the frequency of the simulated signal, and V is the voltage applied to the element in the simulation.

To reduce the time and memory needed for some FE simulations, the structure is set up without fluid elements, when the admittance of the element is to be calculated. This can be done without decreasing the accuracy of the simulation, as the fluid load of air is so small that it is almost equivalent to a setup with an element vibrating in vacuum [24].

Chapter 3

Experimental setup and method

In this chapter, the electrical and acoustical measurement setups and methods are outlined. A list and overview of the equipment used can be found in Section 3.1. The setup and method for the electrical measurements is described in Section 3.2. The setup and methods for the acoustical measurements are found in Section 3.3, covering the use of equipment, the methods for positioning the transmitter and receiver, and the various corrections and sources of uncertainty needed to be considered. The details of the microphone measurement system is covered in Section 3.6, covering the calculation of the receiver sensitivity, and the corrections dependant on frequency and angle of the received signals. The setup for the signals used in the acoustical experiments, and the post processing methods used can be found in Section 3.7.

3.1 Equipment used

A list of the equipment used in this work is presented in Table. 3.1.

TABLE 3.1: Equipment used in this work.

Equipment type	Model	Serial number	Doc.
Impedance analyzer	HP 4192A	23423	[31]
Oscilloscope	Tektronix DPO3012	C010426	[32]
Signal generator	Agilent 33220A	MY44023589	[33]
Measurement amplifier	Bruel & Kjær 2636	1815638	[34]
Filter	Krohn-Hite 3940	AM2626	[35]
Measurement Microphone	Bruel & Kjær 4138	2784915	[36]
Microphone preamplifier	Bruel & Kjær 2670	2799662	[36]
Pistonphone	Bruel & Kjær 4228	1918465	[37]
Barometer	Bruel & Kjær UZ0004	1918465	
Thermometer	ASA F250 MkII	2265026992	[38]
Rotational stage	Physik Instrumente M-037	0912A	
Linear stage	Physik Instrumente M-535	1760497	[39]
Linear stage	Physik Instrumente M-531	1460497	[40]
Laser Displacement sensor	Keyence LK-G Series	–	[41][42][43]
Piezoelectric elements	Meggitt A/S Pz27 Ceramic discs	405550/406243	[44]

3.2 Electrical measurement setup

3.2.1 Piezoelectric ceramic elements

The piezoelectric elements used in this work are Soft Relaxor Type PZT, Type Pz27, produced by Meggitt A/S [44]. The given dimensions of the elements are a thickness of 2 mm, and a diameter of 2. The actual dimensions of the elements used are presented in Section 5.1.

The electrical characteristics of a batch of seven elements is studied during this work. One of these elements is mounted in the measurement setup, and used for measurements of the acoustical characteristics. The elements are labeled with the numbers 1,3,8,9,17,18, and 19, which is used as names throughout this work to specify which element is being tested.

3.2.2 Conductance and susceptance measurements

The measurements of the piezoelectric element's electrical characteristics are done by using an impedance analyzer. Model 4192A LF, made by Hewlett Packard, shown in Fig. 3.2. The conductance and susceptance of the element is measured, and the admittance is found using the equation

$$Y_T(f) = G_T(f) + iB_T(f) = \frac{1}{Z_T(f)} \tag{3.1}$$

where $Y_T(f)$, $G_T(f)$, and $B_T(f)$ is the admittance, conductance, and susceptance of the element respectively. f is the frequency of the oscillating input signal used for measurements.

The impedance analyzer is given a warmup time of at least 30 minutes. Before each measurement, a zero offset calibration is preformed, to account for the impedances of the wires connecting the element to the analyzer. The method is described in the manual of the impedance analyzer [31]. The element is put in a holder, with two wires connected to the impedance analyzer's terminals touching the electrodes of the element, see Fig. 3.1. This setup is to minimize the mechanical load on the element, allowing it to vibrate more freely. This gives a more accurate measurement of the electrical characteristics of the element, and gives a better comparison with the simulated admittance. The simulation setup calculates the admittance using a freely vibrating element in vacuum, with no mechanical loads on the element.

The impedance analyzer outputs an oscillating root mean square voltage signal, V_{osc}, to induce vibrations in the element, so the electrical characteristics can be measured. Multiple values of V_{osc} is tested, (see Section 5.2), but a value of $V_{osc} = 0.3$ V is used for admittance measurements. This level is chosen as a compromise to avoid the non-linear effects expected at higher voltages, while still having a good enough resolution for the measured admittance. Any value below $V_{osc} = 0.1$ V is stated to increase the uncertainty of the measurements [31]. Earlier work by Mosland [2], show a slight repeatability issue, especially when the element is removed from the setup and inserted again, showing that the positioning and contact of the wires on the electrodes can affect the measured values.

The measurements are preformed using a computer connected to the impedance analyzer using a GPIB to USB adapter. The MATLAB-script impanal.m is used to specify measurement parameters and to run and save the measurements. In this work, the conductance and susceptance is measured from 1 to 300 kHz, covering the two first radial modes of the element. The frequency resolution is variable across the frequency range, with a higher resolution around the resonance peaks. The frequencies of the resonances are determined first

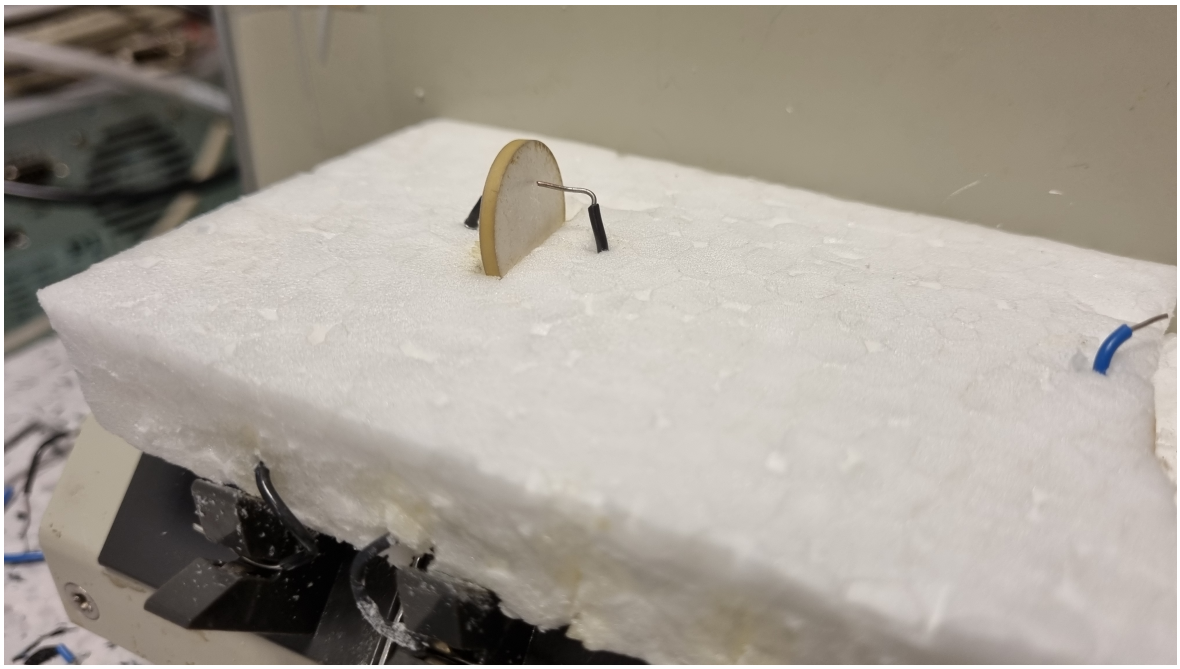


FIGURE 3.1: Piezoelectric ceramic disk is held in a slot in the styrofoam, and the wires are bent so that they get good contact with the electrodes of the element.

FIGURE 3.2: HP 4192A impedance analyzer used to measure the admittance of the element.

by a low resolution measurement of the element, so a higher frequency resolution could be specified for the measurements around the resonances.

3.3 Acoustical measurement setup

The acoustical measurements done in this work is preformed using the setup shown in Figs. 3.3 and 3.4, in house at the acoustics lab at UiB. The signal path in this setup starts at the signal generator (1), producing a sinusoidal pulse of length and frequency determined by the MATLAB-script *measurement_parameters.m*, shown in Appendix A. The pulse is sent to the transmitting element (2), travelling through air to the receiving microphone (3). From there, the signal is sent to the amplifier (4), where the signal is amplified before being sent through the band-pass filter (5). The signal terminates at the oscilloscope (6). The output signal of the signal generator is also measured by the oscilloscope.

The setup has the hardware to measure atmospheric pressure, relative humidity, and temperature during each measurement, but in this work, the atmospheric pressure and humidity is not measured, due to problems with the equipment. A plastic sheet is lowered down in front of the metal cage during measurements, to reduce effects from moving air currents.

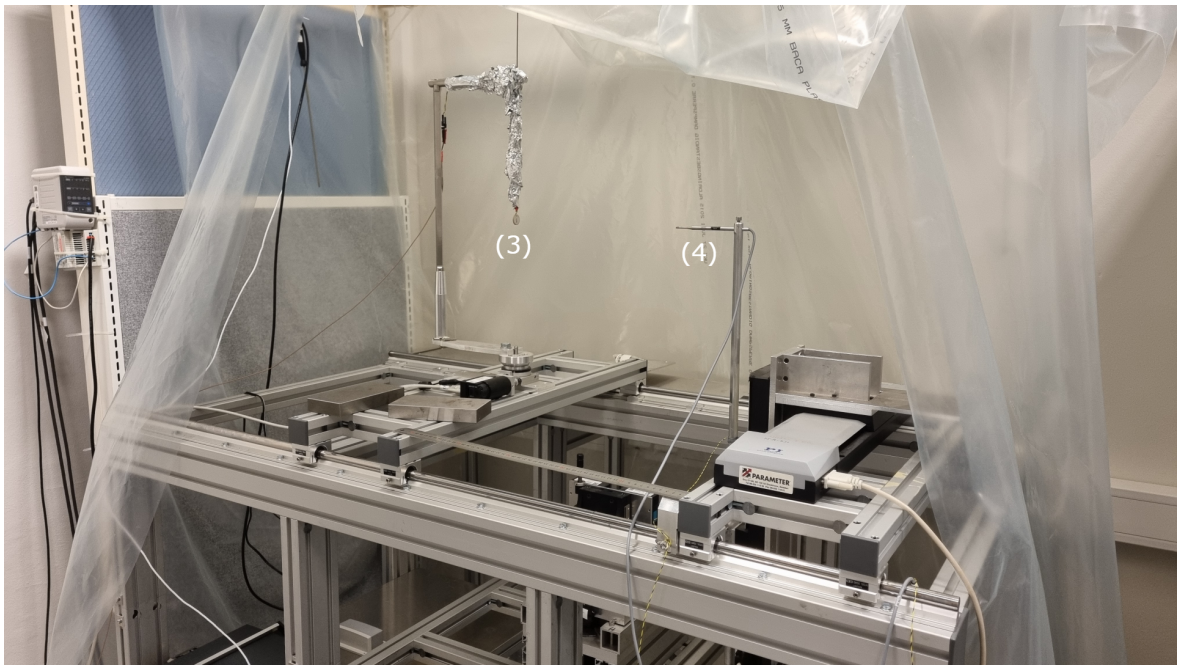


FIGURE 3.3: Picture of acoustic measurement cage, with mounted transmitter element and receiver microphone.

3.3.1 Signal generator

The signal generator used in this work is an *Agilent 33220A 20 MHz Function/Arbitrary Waveform Generator* [33]. The signal generator outputs a pulse of frequency, amplitude, and temporal length specified in the MATLAB script. The peak to peak amplitude of the sine wave is set to $V_{pp} = 1V$, and is equivalent to V_0 in the system diagram. The voltage amplitude set for the signal generator is the output voltage generated towards a theoretical 50Ω load, and in a theoretical open circuit setup, the measured output voltage is doubled. At resonance, the piezoelectric element applies a low resistance, and the measured V_{pp} of the pulse approaches the chosen output voltage of the signal generator, $V_{pp} = 1V$. When measurements are done outside the resonance of the element, the perceived resistance is much higher, and the circuit approaches an open-circuit scenario, so the measured V_{1m} of the pulse approaches 2 volts [33].

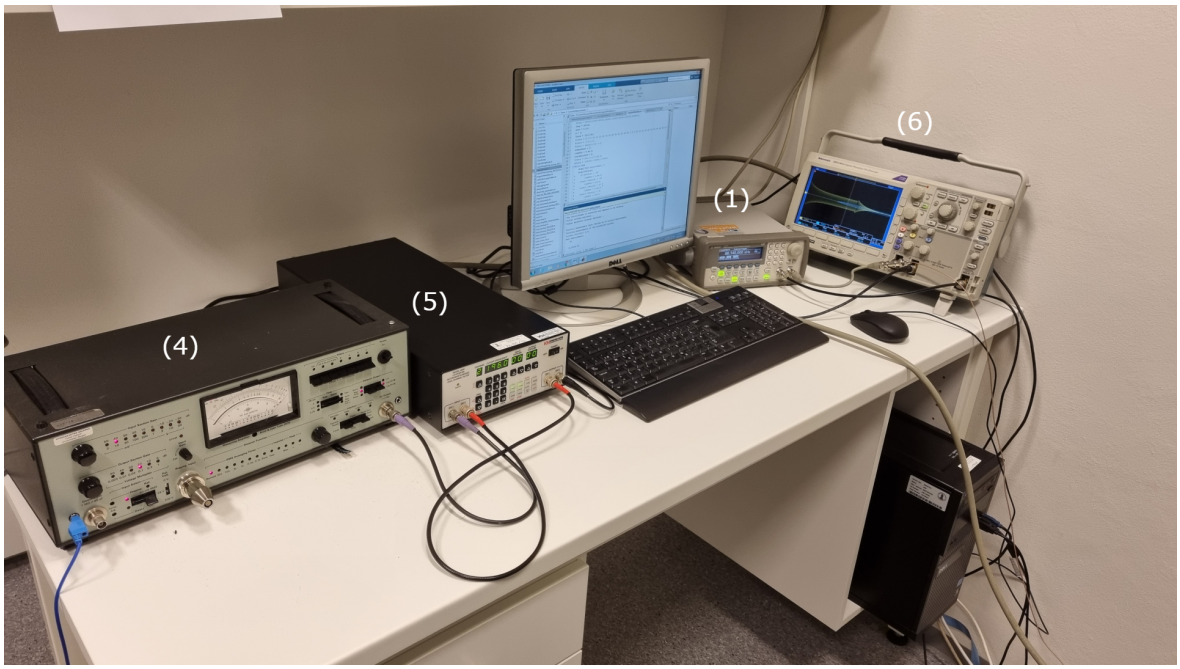


FIGURE 3.4: Picture of acoustic measurement cage, with mounted transmitter element and receiver microphone.

The period between signal bursts is set to 40 ms, which corresponds to a burst rate frequency of 25 Hz. This is to make sure that any reverberations in the element has stopped before the next pulse is sent, and to make sure all reflections in the setup has died off, so the reflections of the previous pulse does not overlap the next pulse.

3.3.2 Oscilloscope

The oscilloscope used in this work is a *Tektronix DPO 3012 Mixed Signal Oscilloscope* [32]. The oscilloscope has two channels, with separate inputs. The first is connected to the signal generator, with a branch that leads to the transmitting piezoelectric element, that records the generated pulse waveforms sent by the signal generator, V_{1m}. The second input is connected to the receiving electronics, and this channel records the received voltage amplitude V_{5m}. The oscilloscope is also connected to the PC. The PC sends commands to adjust the horizontal and vertical scale and position of the recording area on the oscilloscope, so that the entire waveform is recorded.

The oscilloscope displays a rolling average of 128 pulses. This averaging is used to reduce random white noise picked up and generated by the transmitting and receiving technologies, while leaving the regular pulses unchanged. The vertical scaling is automatically changed so the signal waveform fills as much of the screen as possible without clipping. This is to maximize the voltage resolution of the recorded signal. Fig. 3.5 shows an example of the transmitted and received signals as they are displayed and recorded by the oscilloscope, with a rolling average of 128 pulses. Figs. 3.6 and 3.7 show the received pulse at a minimum in the directivity beam pattern of the element, with and without averaging, to show the effects on the noise. As the element is moved away from the receiver, and the frequency is changed to something other than a resonance frequency, the signal to noise ratio increases dramatically, so the averaging effect becomes that much more important.

Where it is possible, the length of the signals are tweaked so that the transmitted and received signals do not overlap in time. This is so that the electric field generated by the voltage running through the transmitting element do not disturb the microphone as the

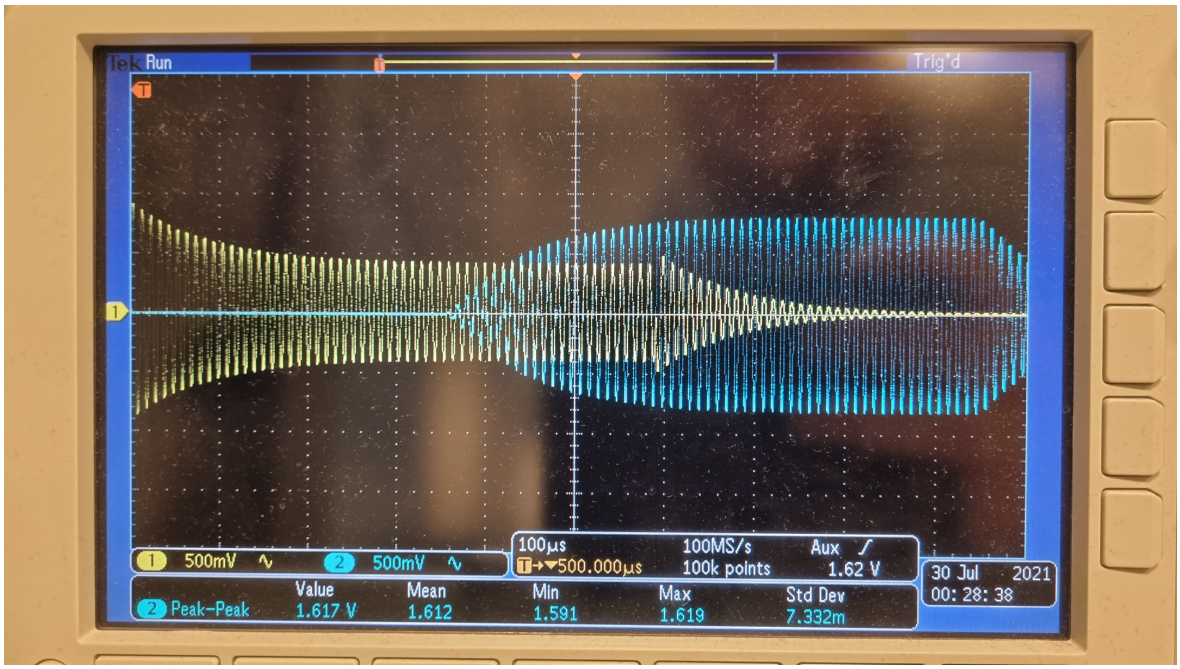


FIGURE 3.5: Transmitted and received pulse, 0.6 ms pulse at 98.14 kHz, z = 10 cm, and $\theta = 0$ *degrees*, with a rolling averaging of 128 pulses.

pulse is received. This overlap can be seen in Fig. 3.5, where the first section of the received pulse is possibly affected by the electric signal. At the elements resonance frequencies, and when the separation is low, the amplitude of the recorded signal is so high that the electric interference is negligible, but with measurements where the amplitude is much lower, the interference needs to be taken into account. In this work, to avoid the effects of the interference, the pulse length has been chosen to avoid overlap. At low separations, where this is not possible due to the low transmission time, the interval of the recorded pulse used for the calculation of V_{pp} can be chosen as to not include the period where the transmitted signal can influence the received signal.

The oscilloscope uses an 8-bit system, so the resolution is limited. Earlier work by Mosland has shown that the lowest viable vertical scaling of the oscilloscope is 10 mV/-div [2]. Below this scaling, the random noise extends above and below the recording area, and the averaging of the pulses doesn't reduce the noise properly, affecting the desired signal. The oscilloscope is set to record 100 000 samples over the recording window, which is usually 1 or 2 ms, giving a sample rate of 100 MHz or 50 MHz respectively. The recording window size is rounded up to the nearest millisecond above the pulse length, which in this work varies between 0.6 ms and 1.4 ms.

3.3.3 Amplifier

The signal amplifier used in this work is a *Bruel & Kjær 2636 Measuring Amplifier*. [34] The measuring Bruel & Kjær microphone is connected to the amplifier via a proprietary connector. The amplifier can provide up to 100 dB gain in 10 ± 0.05 dB steps. This amplification is split over input gain and output gain [34]. In this work, a gain of 60 dB is used, split over 40 dB at the input, and 20 dB at the output. The amplifier has a flat 0 dB frequency response from 1 Hz to 200 kHz, as seen in Fig. 3.8, which covers the frequency range used in this work. Therefore, the signal needs no adjusting based on the data given. The amplifier also has built in frequency filters, but an external filter is used in this work instead.

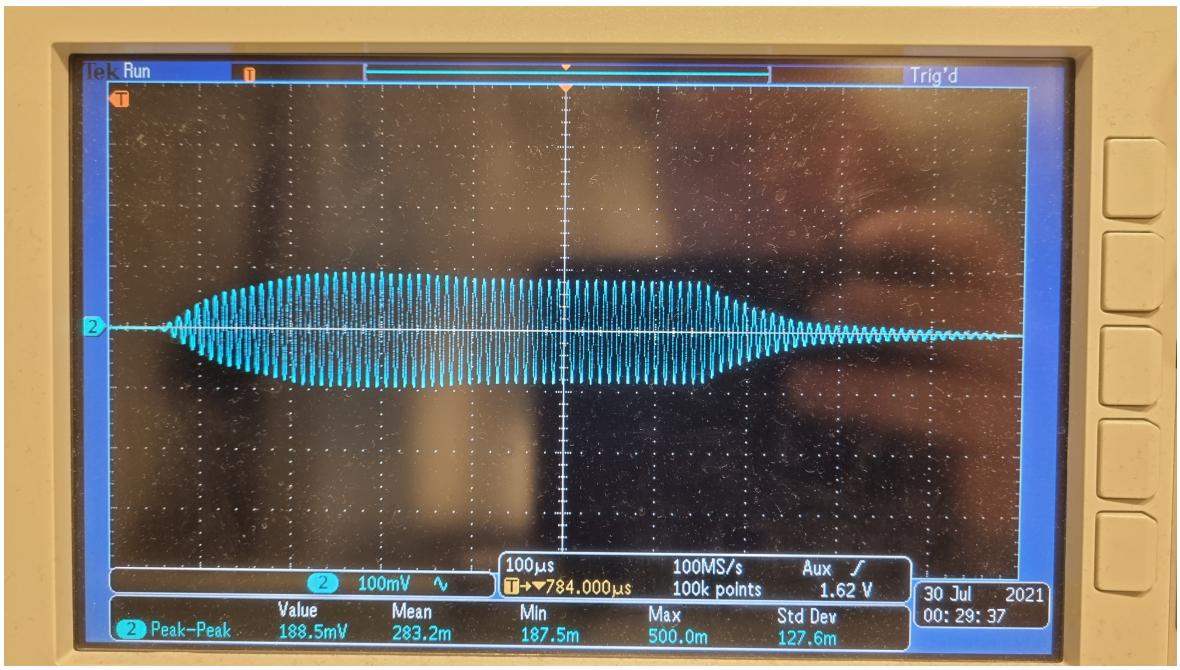


FIGURE 3.6: Received 0.6 ms pulse at 98.14 kHz, z = 10 cm, and $\theta = 42$ *degrees*, with averaging.

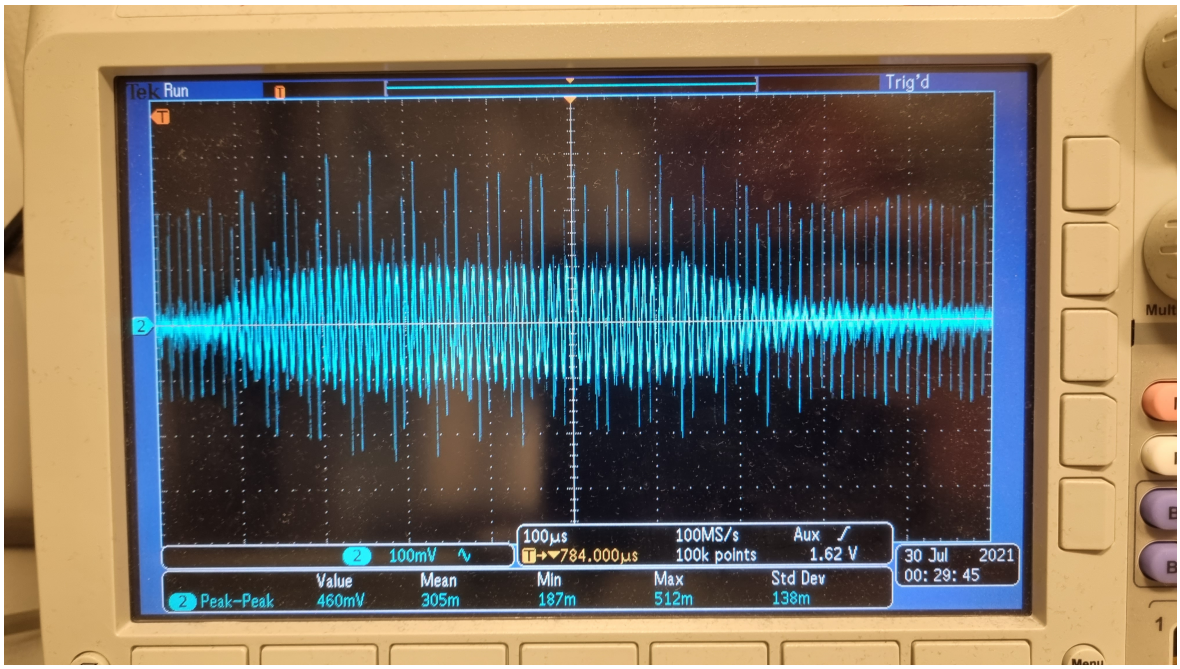


FIGURE 3.7: Received 0.6 ms pulse at 98.14 kHz, z = 10 cm, and $\theta = 42$ *degrees*, showing a single sample, without averaging.

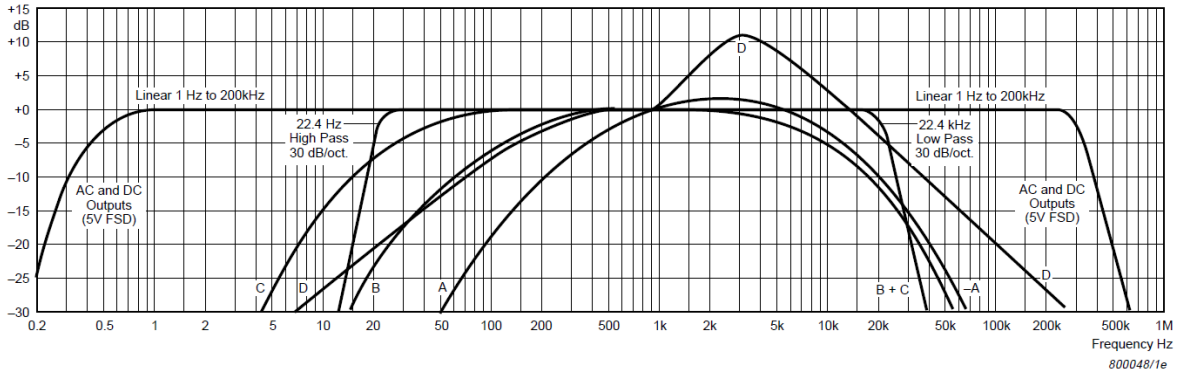


FIGURE 3.8: Typical overall frequency response of Measuring Amplifier Type 2636, with and without internal filter and weighting networks selected [34].

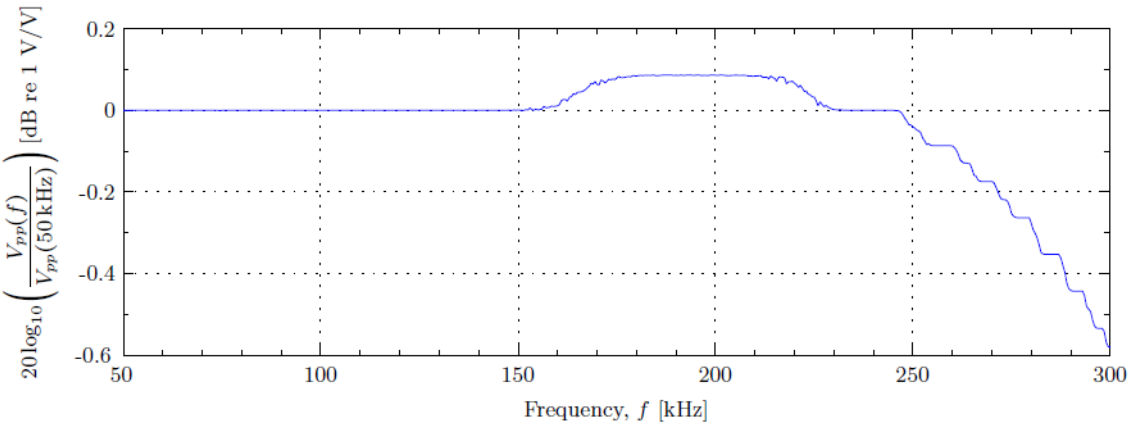


FIGURE 3.9: Ratio of recorded voltage over the frequency span and the voltage recorded at 50 kHz. Taken from [2].

Earlier work by Mosland [2] has shown that the frequency response of the amplifier is not completely flat between 0 and 200 kHz, with a small additional amplification of 0.1 dB from around 175 kHz to around 225 kHz. This correction data is used on all voltage data collected, where the amplifier is used. The data recorded by Mosland is shown in Fig. 3.9.

3.3.4 Bandpass filter

The filter used in this work is a *Programmable (IEEE-488) Filter, Model 3940*, produced by Krohn-Hite [35]. An effective bandpass filter is achieved by running a high-pass and low-pass filter in the two respective channels, and having the signal travel through both channels. The high-pass filter cutoff frequency is set to an octave below the current signal frequency, while the low-pass filter cutoff frequency is set to an octave above. Each channel of the filter has a 24 dB per octave attenuation outside the bandpass frequency range.

3.3.5 Cables

There are two different types of cable used in this setup. The first is used to connect the instruments together, and the second is used to connect the transmitting element to the signal generator. The cables used for connecting the instruments are coaxial cables of type RG58 with a characteristic impedance of 50 Ω. The cable used between the signal generator and the transmitting element does not have any identifying marks on it, but product data has

TABLE 3.2: Specifications for typical coaxial cable, type RG58 [14].

Unit	Value
Inductance per meter	L = 250 nH/m
Capacitance per meter	C = 100 pF/m

TABLE 3.3: Length of cables used in measurement setup.

Cable connection	Length
Signal generator to oscilloscope	0.25 m
Oscilloscope to transmitting element	3 m
Amplifier to filter (ch. 1)	0.5 m
Filter (ch. 1) to filter (ch. 2)	0.8 m
Filter (ch. 2) to oscilloscope	1.5 m

been found for what is assumed to be the same type of cable, based on looks and characteristics. This cable shares the characteristic impedance of the RG58 cable, so it is assumed in this work that the values for inductance and capacitance is the same. This is not necessarily correct, but due to missing data on the cable used, the assumption has to be made. This adds an uncertainty to the calculated values of V_1. The capacitance and inductance per meter of the cable used are shown in Table. 3.2. The length of the different coax cables used in this work is shown in Table. 3.3.

3.4 Transmitter and receiver mounting and positioning

The stages that control the transmitter and receiver positions are mounted on two platforms, that run along rails mounted in the steel cage of the setup. The platform with the receiver mounting are controlled by stages moving the microphone along the x- and y-axis [40][39]. The microphone is mounted through a hole in an aluminium rod, with a tightening screw at the top, to ensure stable mounting. A closeup of the microphone mounting is shown in Fig. 3.10. The transmitting element is mounted hanging from an L-shaped rod mounted to the movable platform. The element hangs from the wires soldered to it, and the wires are shrink-wrapped to a thin metal rod that hangs from the mounting rod, shown in Fig. 3.11. The element mounting is wrapped in aluminium foil to isolate the wires from electrical interference. The horizontal top part of the mounting rod can be rotated in relation to the vertical part, so that the element can be positioned directly over the rotating stage. This is to make sure the position of the element in the xz-plane stays the same as the whole mount rotates, changing the angle of the element. If the element is not exactly centered at the rotational axis, the separation between the element and microphone changes as the element rotates.

Both vertical mounting rods are cut at an angle to reduce the reflections back at the transmitting element and receiving microphone. This works well when the measuring angle is 0 degrees, but when the transmitter setup is rotated to change the element angle, the flat sides of the cut rods begin reflecting the produced sound towards the microphone. This effect is expanded upon in Section 3.5.

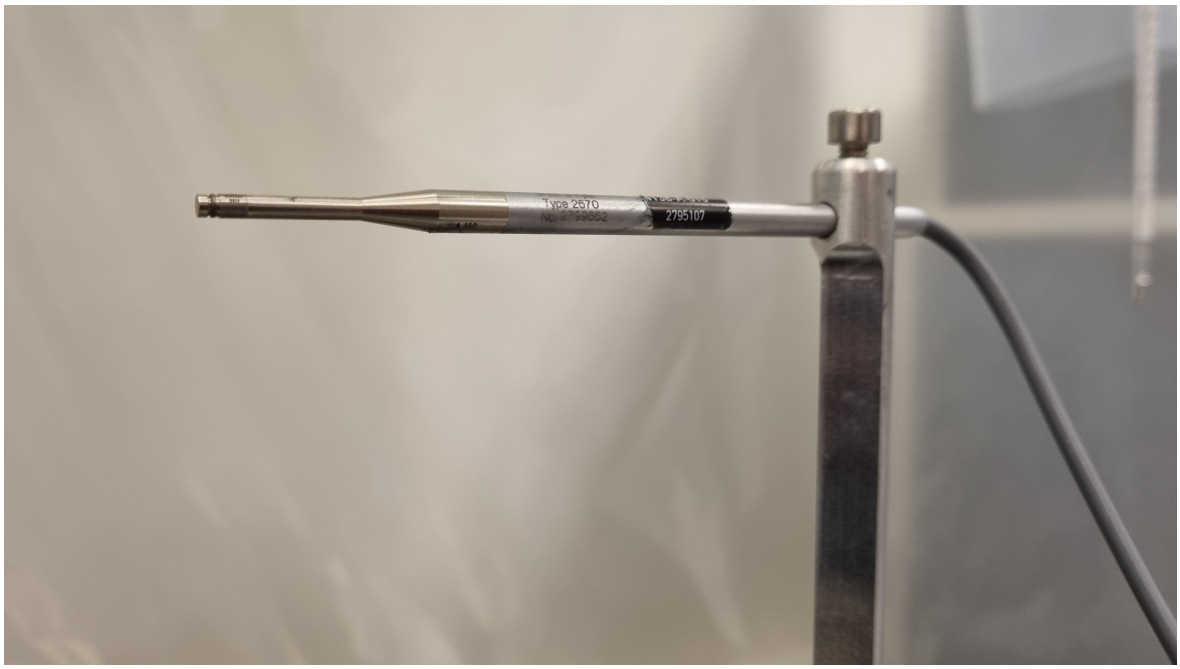


FIGURE 3.10: Bruel & Kjær 4138-A-015 microphone mounted in the measurement setup.

3.4.1 Positioning of the element and microphone

The transmitting element and receiving microphone need to be positioned so that the sound axis of the transmitter and the microphone is lined up. The sound axis of the element is defined as the axis that runs through the main lobe of the element beam pattern, and therefore where the element produces the greatest sound pressure amplitude. The microphone does not have a well defined beam pattern, but the receiver sensitivity is dependant on the incoming angle of the sound waves. This effect is covered in Section 3.6.3. The sound axis of the microphone is defined as the axis running through the middle of the microphone receiver surface and body. Lining up these axes makes sure that the sound pressure recorded is that of the main lobe of the element, which is the most important measure for most uses of piezoelectric elements. It also makes sure that the value for the receiver sensitivity of the microphone holds, since it is defined as the sensitivity when receiving plane waves at 0 degrees incidence. The position where the sound axes are aligned is also defined as the 0 degrees angle point for the element, and is used as a reference when rotating the transmitting element.

3.4.2 Lining up in xy-plane

For this adjustment, a Keyence LK-G series laser displacement sensor setup is used [41]. The setup uses a LK-G 3001 controller, with two LK-G32 sensor heads mounted against each other to measure separation between two objects. The sensor mounting is shown in Fig. 3.12. The laser sensors can only measure the distance to objects within a small range of a couple centimeters, so the microphone and element is positioned so a measurement can be preformed. First, the element and microphone is aligned in the xy-plane by centering the laser dot on the centre of the transmitting element, and moving the microphone in the xy-plane using the motorized stages, until the other laser is aligned with the centre of the microphone receiving surface. Once the element and microphone are aligned, the distance between them are measured. The laser meter displays the distance from each of the

FIGURE 3.11: Transmitting element mounted in measurement setup.

laser sensors, to the element and microphone respectively. The distance between the laser sensors are known, so the three values added together is the total separation between the microphone and element.

3.4.3 Vertical angle

The element is suspended by the wires soldered on to carry the signal to the electrodes, and these wires are then shrink-wrapped to a thin rod to hang the element from the mounting setup. Since the element can't be precisely adjusted while hanging from these wires, it is difficult to make sure the element is completely vertical. This means the sound axis may point slightly up or down, compared to the horizontal xz-plane.

Therefore, after the the element and microphone and element is lined up in the xy-plane, the element is moved to different separations, and multiple measurements are done, while moving the microphone up and down in small steps. The output V_{pp} is measured at each point, and is plotted depending on the position of the microphone in the y-axis. The maximum is then found at different separations, and from there an equation can be found by extrapolation giving the required microphone y-position given a separation distance on the z-axis. This ensures that the default positions of the microphone and element always means that the sound axis of the element passes through the microphone. In this work, after adjusting the element, the sound axis was found to point slightly down, so the microphone had to be lowered by a small amount as the separation increased. At the maximum separation of 70 cm, the microphone was lowered by around 2 cm to compensate for the difference in main lobe position of the element.

3.4.4 Horizontal angle

In this work, the sound pressure will be measured at points spanning from -90 to 90 degrees, mapping the directivity beam pattern of the element at different separations. These directivity measurements at many different separations can then be combined to display a full mapping of the sound pressure field in the xz-plane.

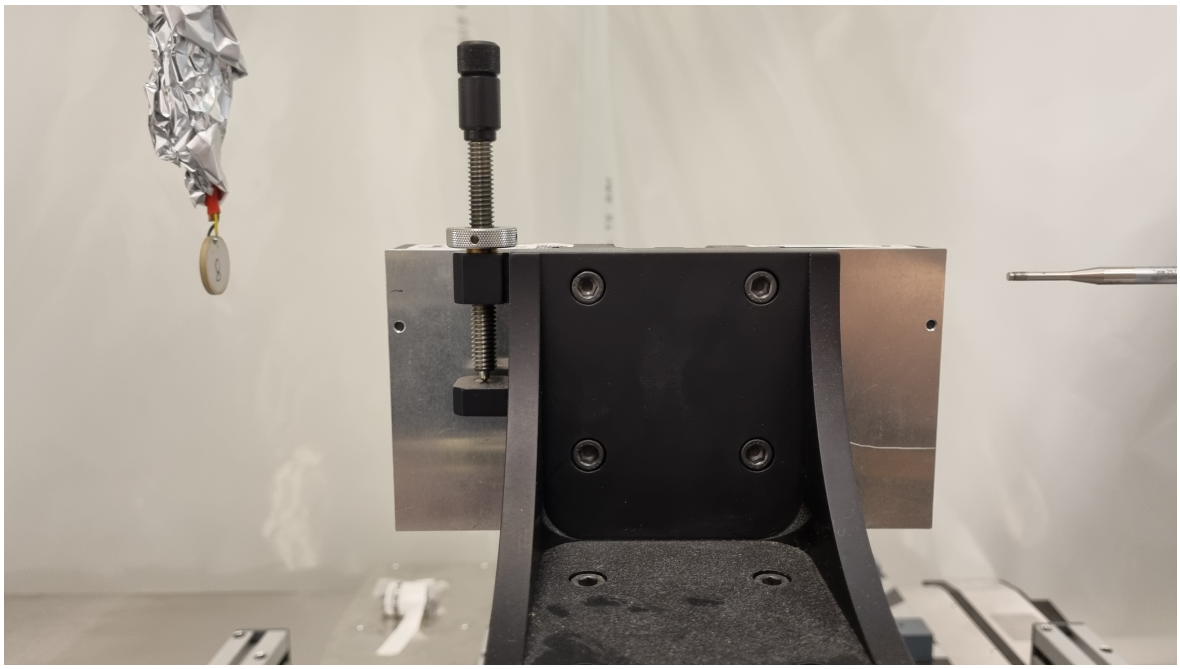


FIGURE 3.12: Laser sensor mounting of the displacement sensor setup used to measure separation. The sensor heads can be raised or lowered to align it, and to move it out of the way for acoustical measurements.

It is therefore required to find the sound axis of the element, and define it as 0 degrees in the beam pattern. Here, the element is roughly rotated, so the transmitting face is facing the microphone. The element is then rotated around this angle, with a couple of degrees on each side. Again, the V_{pp} is recorded and plotted based on the angle. The maximum value is then found, and the 0 degrees position is defined as the position of the maximum. An example of these measurements is shown in Fig. 3.13, where the maximum is found to be at -4 degrees. The element is then rotated to -4 degrees, and the position is defined as the new 0 degrees position in the controller software run on the lab PC. Over the period that the experiments were run, it was noticed that the alignment of the sound axis seemed to worsen over time, either due to accidental touches to the element, or that the element position and angle gradually shifted due to the forces applied to it as it was rotated, as it was only held in place with two hanging wires. The alignment process described earlier was therefore preformed with regular intervals, and before important measurements. The error measured was never more than 1 degree after the first alignment.

3.4.5 Distance between the element and microphone

Due to problems with the linear stage controller controlling the elements position in the z-axis, the positioning of the element had to be done manually. The laser measurement system is used to measure the distance between the element and microphone at a reference distance, and a metal ruler is then used to determine the relative distance between the structures holding the element and the microphone, as shown in Fig. 3.14. When the element is brought right up to the microphone surface, the ruler gives a measurement of 14.4 cm between the measurement points chosen on the two movable platforms. The real separation is therefore given as

$$d_{real} = d_{ruler} - 14.4cm \tag{3.2}$$

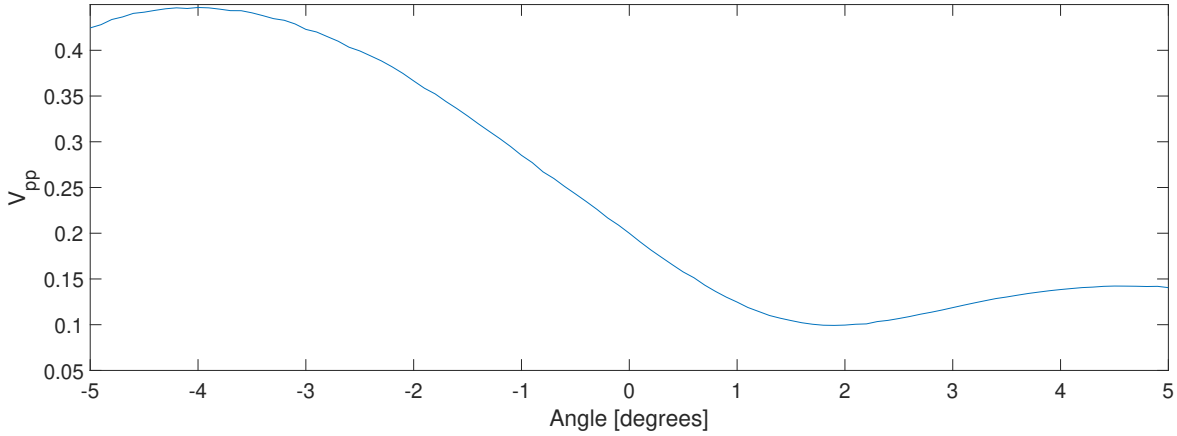


FIGURE 3.13: Measurement peak to peak voltage from -5 to 5 degrees, at 40 cm separation.

The real separation can also be found using the laser displacement sensors, as shown earlier, but only within a small separation range. The differential between the ruler and the real separation is verified by the laser measurement method, that also shows a differential of 14.4 $\pm$0.05 cm. The positioning is done by hand, by moving the structure holding the element on rails. Since the distance is measured using a ruler, the uncertainty of the position is $\pm$ 0.5 mm.

3.5 Reflections

3.5.1 Reflections off the microphone mounting rod

The microphone is mounted through a hole in an aluminium rod, connected to the motorized stages. The rod has been cut so that the side facing the piezoelectric element is wedge-shaped, to reduce reflections off the rod in the direction of the microphone.

3.5.2 Reflections off the piezoelectric element mounting rod

The piezoelectric element is mounted to a rod that has been cut at an angle to reduce reflections back at the element. This should reduce reflections toward the microphone when the element is pointed towards the microphone (0 degrees), but in this work, directivity measurements are done where the element and the rod it is mounted to rotates from -90 to 90 degrees. When the structure rotates, the rod moves off the axis of the microphone, and the cut flat faces of the rod begins to point towards the microphone. This causes the sound that is projected from the backside of the element to hit the rod, and reflect towards the microphone. An example is illustrated in Fig. 3.15, where the element is positioned at 90 degrees. These reflected waves can be seen in the signal as interference that arrives later than the direct signal. The interference can either be constructive or destructive, either raising or lowering the signal amplitude, depending on the phase difference between the main signal and the reflected signal. See Figs. 3.16 and 3.17, where the interfering signal arrives around 0.85 ms after the main signal. The reflected signal can be seen in Fig. 3.16, in the last part of the graph, where the main signal has died off. The reflected signal is shown here to have a much lower amplitude than the main signal, most likely caused by the reflection and scattering off the reflecting rod. The main signal reaches steady-state after around 0.3 ms, so an interference-free section between 0.3 and 0.85 ms can be used for calculating V_{pp}. When the path from the mounting rod to the microphone is physically blocked by a sound

FIGURE 3.14: Ruler used to determine the separation between the movable platforms, and from there, the separation between the element and microphone.

TABLE 3.4: Calculations of the extremes of possible reflection paths, and the corresponding delays in reflected signal arrival at the microphone.

Element angle	Separation	Path length difference	Reflection delay
0 degrees	20 cm	$\Delta d \approx 39$ cm	$\Delta t \approx 1.13$ ms
$\pm$ 90 degrees	20 cm	$\Delta d \approx 27.5$ cm	$\Delta t \approx 0.81$ ms
0 degrees	80 cm	$\Delta d \approx 39$ cm	$\Delta t \approx 1.13$ ms
$\pm$ 90 degrees	80 cm	$\Delta d \approx 21.5$ cm	$\Delta t \approx 0.63$ ms

absorbing material, the interference disappears, so it is confirmed that the reflections are the cause of the interference. Some examples of path length and reflection travel time is shown in Table. 3.4. Trigonometry is used to calculate the path length difference Δd at any angle.

$$\Delta d = d_{refl} - d_{dir} \tag{3.3}$$

where d_{dir} is the direct path distance from the transmitter to the receiver, and d_{refl} is the path distance of the signal reflected off the mounting rod

$$d_{refl}(\theta) = \sqrt{d_{rod}^2 + d_{dir}^2 - 2\, d_{rod}\, d_{dir}\, cos(180 - \theta)} \tag{3.4}$$

where d_{rod} is the distance from the transmitting element to the mounting rod, in this case d_{rod} = 19.5 cm. d_{dir} is the direct path length, and θ is the rotational angle of the element, which is subtracted from 180 degrees, since the reflections are transmitted from the back of the element. The maximum and minimum path length difference is seen at θ = 0 degrees, and $\theta 0$ = 90 degrees respectively. The FFT calculation interval, expanded upon in Section 3.7, is chosen so as to not overlap with any possible reflections from the transmitter mounting rod.

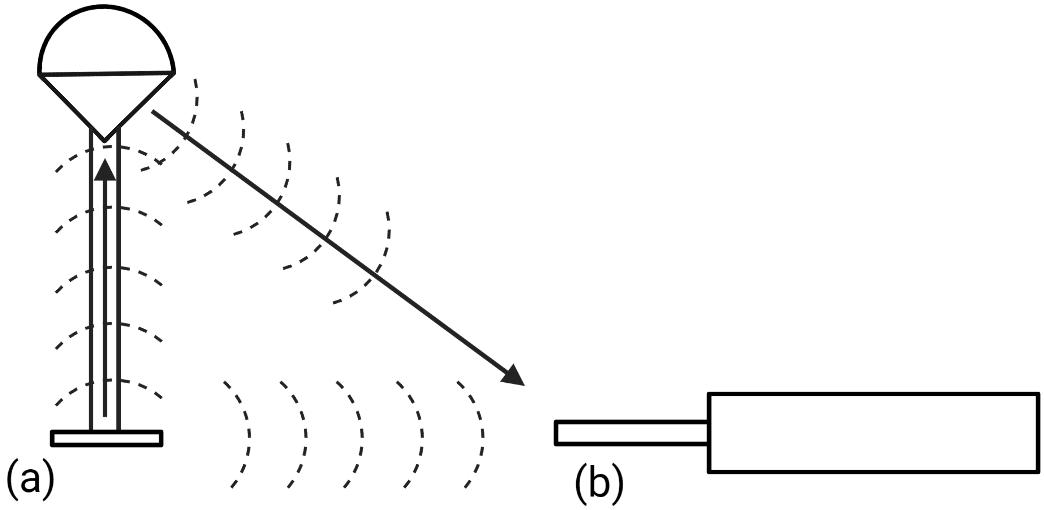


FIGURE 3.15: Drawing showing how the sound waves from the transmitting element, (a), travel when reflected from the rod to the microphone, (b). Element angle in this drawing is θ = 90 degrees, but measurements are done for a range of -90 to 90 degrees.

3.5.3 Reflections off the wall and ceiling

The reflections off the wall and ceiling could interfere with the sound pulses used in this work, but with the short pulse length used, the travel time of these reflections are long enough so that they reach the microphone after the original pulse has passed. The period between pulses has also been chosen, so that reflections from a previous pulse can't interfere with the next.

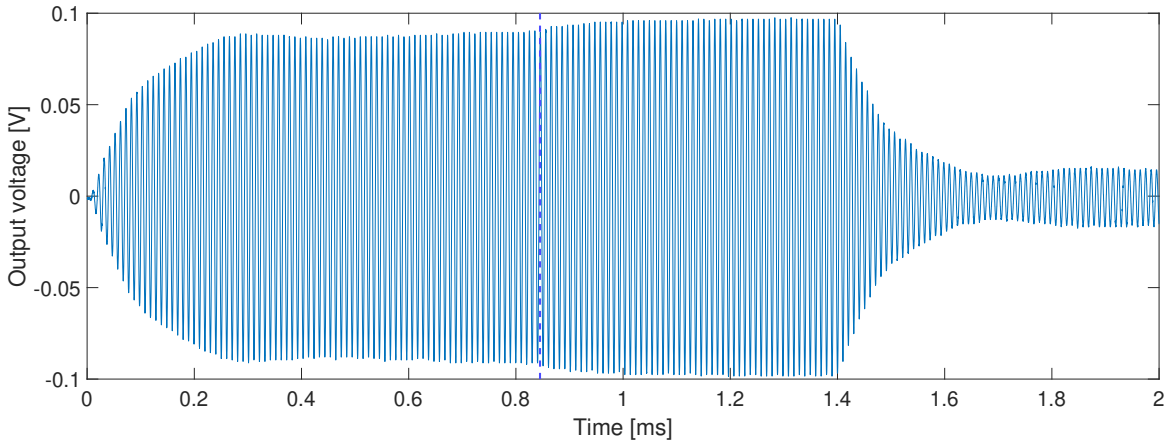


FIGURE 3.16: Measurement at -81 degrees, 20 cm separation, 98.14 kHz, showing constructive interference. Line marks when interference signal arrives.

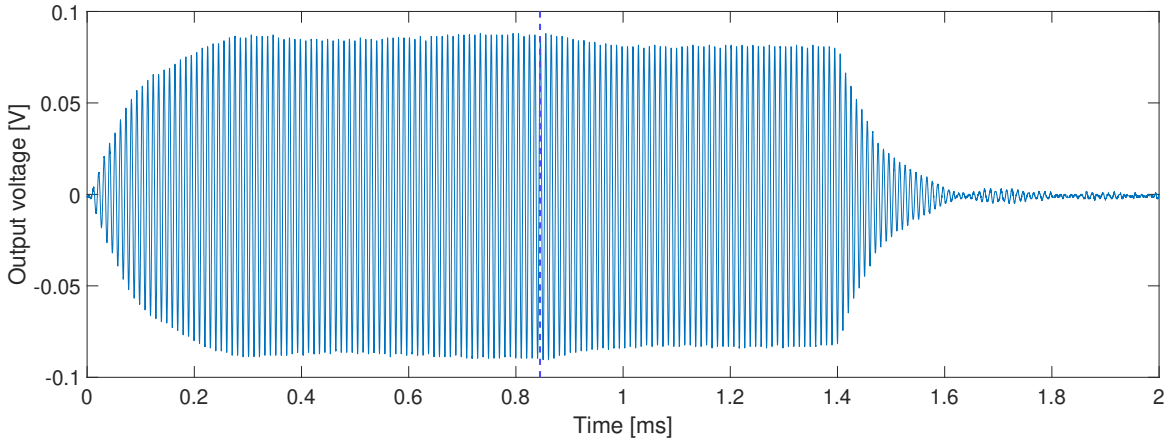


FIGURE 3.17: Measurement at -80 degrees, 20 cm separation, 98.14 kHz, showing destructive interference.

3.6 Microphone measurement system

3.6.1 Receiver sensitivity calibration

A calibration of the microphone is preformed to establish a new receiver sensitivity, accounting for any changes in the measurement equipment or atmospheric conditions in the laboratory that can affect the measurements. The theory and method used for calibration and calculation of $M_V(f)$ is shown in Section 2.5.1. The receiving sensitivity is defined as [14].

$$|M_V| = \frac{V_{eff}}{p_{eff}} \tag{3.5}$$

where V_{eff} is the effective measured open-circuit voltage of the receiver, and p_{eff} is the free field effective pressure at the receiver surface [14]. In order to find the receiver sensitivity, a known sound pressure level must be applied. This is achieved with the Bruel and Kjær 4228 pistonphone, that generates a known SPL for a measured temperature and atmospheric pressure. The SPL is given as [27]

$$SPL = 20log_{10}(\frac{p_{eff}}{p_{ref}}) \tag{3.6}$$

with a p_{ref} given as 20 μPa for the pistonphone. For this particular pistonphone, the SPL is given as 124.11 $\pm$ 0.09 dB re 20 μPa at 251.2 Hz $\pm$ 0.1 %, with a correction for load volume and ambient pressure [37]. The actual SPL is given as [37]

$$SPL_\alpha = SPL + \Delta L_V + \Delta L_p = 124.11 + 0 - 0.05 = 124.06\ dB\ re\ 20\ \mu Pa \tag{3.7}$$

where SPL is the stated sound pressure level of the pistonphone in dB re 20 μPa, ΔL_V is the correction due to load volume, and ΔL_p is the correction due to the difference in ambient pressure from the reference value. The reference value is 1013 hPa, and the correction can be read directly from the barometer included with the pistonphone, which in this case is -0.05 dB, shown in Fig. 3.18. The load volume for this setup is zero, when using an 1/8-inch microphone with a protection grid.

The output voltage of the microphone is read using the DPO3012 oscilloscope. The mean peak-to-peak value V_{rec}^{pp} is read from the oscilloscope measurement function. The magnitude of the receiver sensitivity is the given as [1]

$$|M_V(251.2Hz)| = \frac{\frac{V_{pp}^{rec}}{2\sqrt{2}\cdot 10^{\frac{G}{20}}}}{10^{\frac{SPL_\alpha}{20}}\cdot p_{ref}} = \frac{V_{pp}^{rec}}{2\sqrt{2}p_{ref}10^{\frac{G+SPL_\alpha}{20}}} = \frac{V_{pp}^{rec}}{2\sqrt{2}\cdot 20\cdot 10^{-6}Pa\cdot 10^{\frac{20+124.06}{20}}} = \frac{V_{eff}}{p_{eff}} \tag{3.8}$$

where $|M_V|$ is the modulus of the receiver sensitivity, which in this work is treated as the receiver sensitivity M_V, since only amplitude, not phase, is discussed when talking about pressure. SPL_α is the calculated sound pressure level in dB re 20 μPa, p_{ref} is the reference pressure 20 μPa, V_{pp}^{rec} is the recorded peak-to-peak voltage V_{5m} read off the oscilloscope, and G is the amplifier gain in dB, in this case +20 dB.

The effective output voltage amplitude is found using the equation

$$V_{eff} = \frac{V_{pp}^{rec}}{10\cdot 2\sqrt{2}} \tag{3.9}$$

where the 20 dB gain used is compensated for by dividing the recorded peak-to-peak output voltage by $10^{20/20}$.

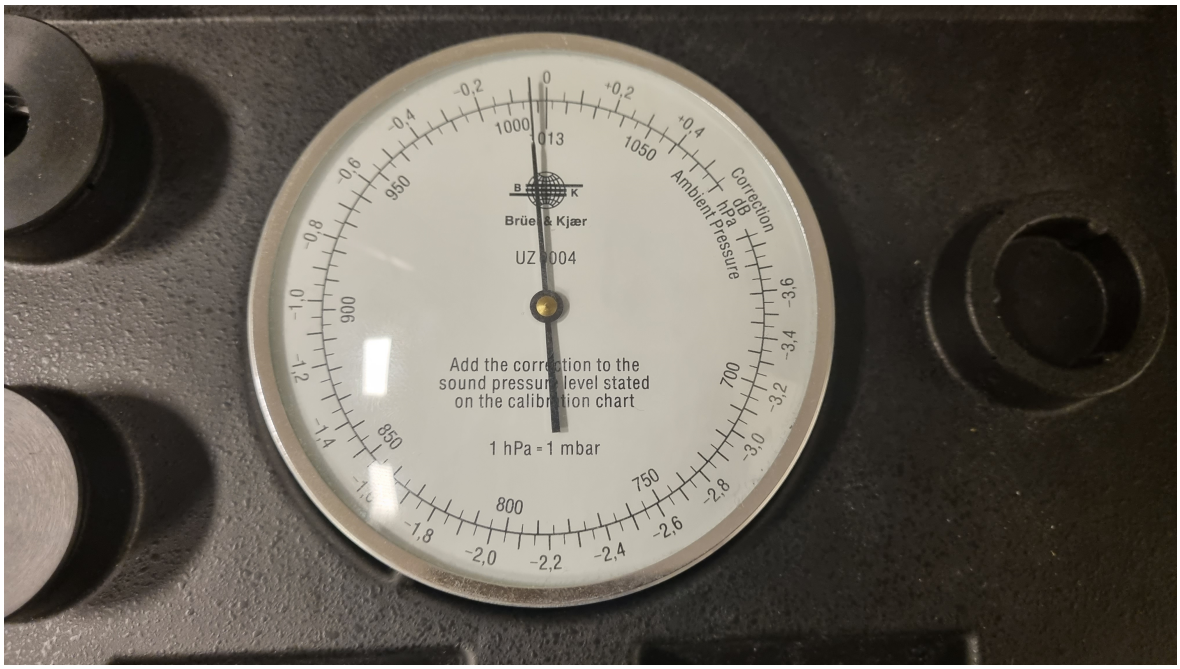


FIGURE 3.18: Bruel & Kjær UZ0004 barometer used to find correction due to ambient pressure.

The effective pressure amplitude p_{eff}, also called the root mean square pressure p_{rms}, is given by the equation

$$p_{eff} = 10^{\frac{SPL+\Delta p}{20}} \cdot p_{ref} = 10^{\frac{124.11+\Delta p}{20}} 20 \cdot 10^{-6} \tag{3.10}$$

where $SPL + \Delta_p$ is the corrected sound pressure level of the pistonphone, and p_{ref} is the reference pressure of the SPL, in this case 20 μPa.

The measured mean peak-to-peak output voltage recorded at 251.2 Hz is V_{pp}^{rec} = 481.2 mV with a standard deviation of 0.1 mV. The oscilloscope mean is calculated using a rolling average of 128 samples, over 10 signal periods, see Fig. 3.19. Ambient pressure correction is at -0.05 dB. The calculated receiver sensitivity is calculated to be

$$|M_V(251.2Hz)| = \frac{V_{eff}}{p_{eff}} = \frac{17mV}{31.918Pa} = 0.533\ mv/Pa = 5.33 \cdot 10^{-4}\ V/Pa \tag{3.11}$$

This same calibration has been preformed earlier by Mosland. In his work, the sensitivity of the microphone system is calculated to a value of $|M_V(251.2Hz)|$ = 0.522 mV/Pa, which is closer to the stated value of $|M_V(251.2Hz)|$ = 0.524 mV/Pa given by the calibration certificate for the microphone system used [36]. The receiver sensitivity provided by the manufacturer is given with an uncertainty of $\pm$ 0.2 dB, or approximately $\pm$ 2.3 %. The value calulated in this work, $|M_V(251.2Hz)|$ = 0.533 mV/Pa, falls within this uncertainty, so while the value calculated by Mosland is closer to the value stated by the manufacturer, the value of M_V calculated here is deemed good enough to use in this work.

Note that the receiver sensitivity is calculated using the effective values of the voltage and pressure, but the same value of M_V can be applied to peak-to-peak voltage, although what you get out is the peak-to-peak pressure, instead of the rms-pressure. In this work, the peak-to-peak voltage is mostly used, since it is easiest to measure directly. The simulated values are however given as the pressure amplitude, which is half of the peak-to-peak pressure, so the calculated pressure is halved before comparison.

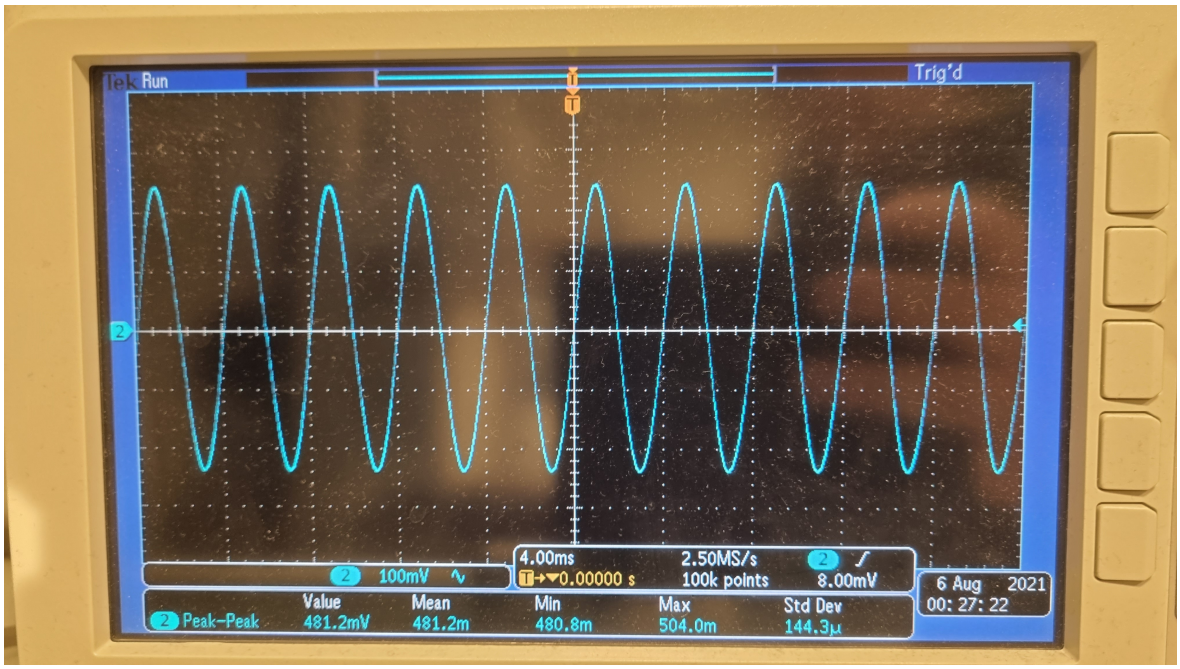


FIGURE 3.19: Signal recorded from Bruel & Kjær 4228 pistonphone with a DP-0774 adaptor, using Bruel & Kjær 4138-A-015 microphone system. Signal averaged over 128 samples.

3.6.2 Actuator response of the microphone over the whole frequency spectrum

The receiver sensitivity of the microphone is found to be

$$M_V(251.2Hz) \approx 5.4 \cdot 10^{-4}\ V/Pa \tag{3.12}$$

but this is only the case at 251.2 Hz. The microphone manufacturer provides the actuator response for a frequency spectrum from 20 Hz to 200 kHz in the form of a graph, shown in Fig. 3.20. Note that the actuator response is only for the microphone, and that the receiver used in this work is a combined 4138 microphone and 2670 preamplifier setup, but the manufacturer states that the frequency response of the preamplifier is flat, so no correction is needed. This graph is digitized using online software [45], shown in Fig. 3.21, and the data is stored. Note that the data points shown in Fig. 3.21 are manually placed, so some error is expected. The thickness of the line is also a factor, as the exact values of the actuator response fall somewhere within the thickness of the drawn line at the graph. The error here is estimated to be $\pm$ 0.1 dB.

The actuator response is then applied to the calculated receiver sensitivity at 251.2 Hz, to get the receiver sensitivity for the whole frequency spectrum used, shown in Fig 3.22. Note that the manufacturer also provides the actuator response numerically, but only up to 100 kHz. For the sake of consistency, the values pulled from the digitized graph is used throughout this work, even of the accuracy may be lower, due to the low resolution of the graph.

3.6.3 Free field correction for incident angle of sound waves

The receiver sensitivity of the microphone is also dependant on the angle of incidence of the incoming sound waves. As the angle increases, the area of the receiving surface "seen" by the incoming sound waves decreases, and the receiver sensitivity decreases with it. The manufacturer provides some data on this angle dependency, but only for some select angles,

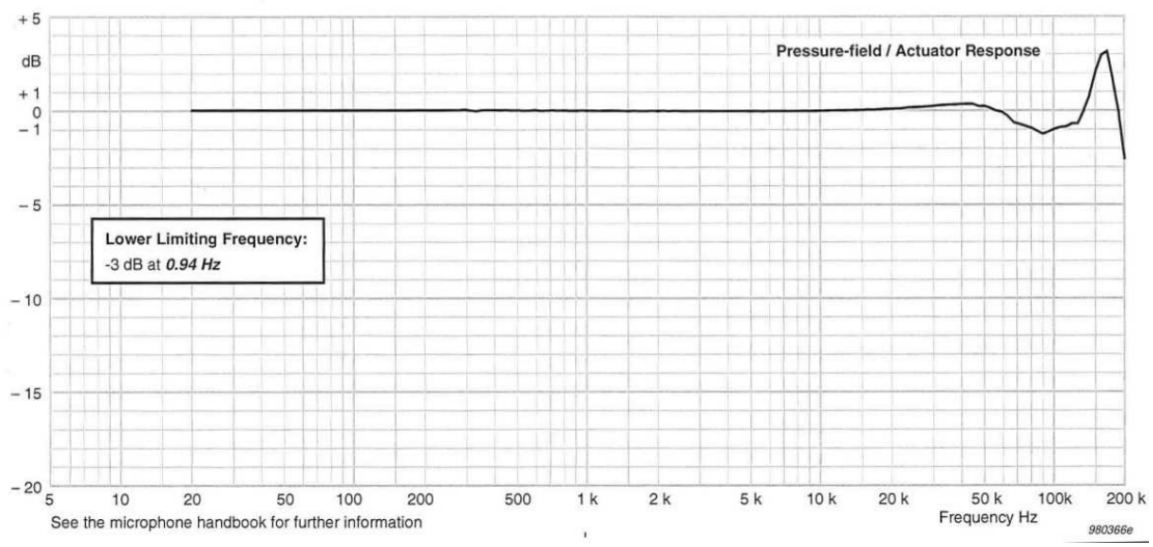


FIGURE 3.20: Actuator response of a 4138 Bruel & Kjær microphone.

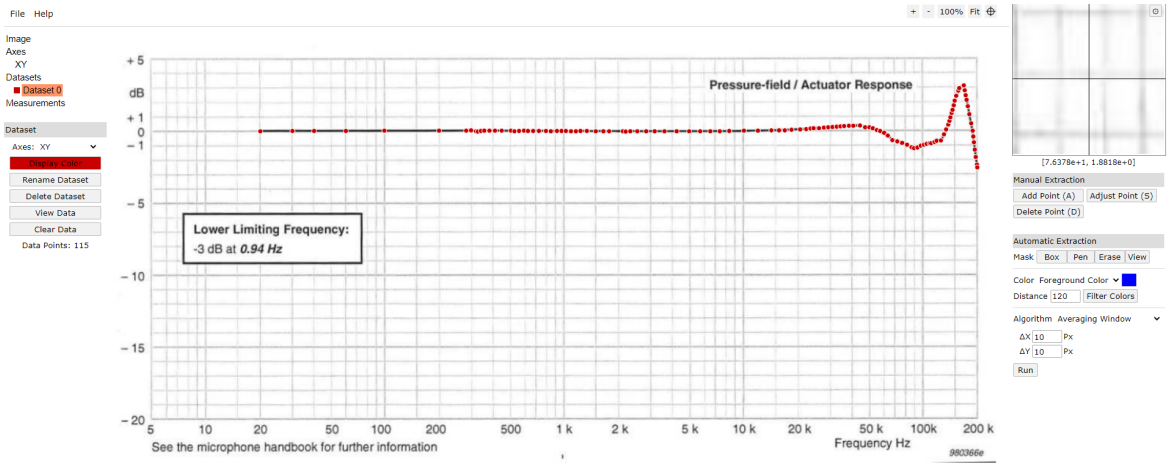


FIGURE 3.21: Actuator response of a 4138 Bruel & Kjær microphone, with manually placed data points, using online software [45].

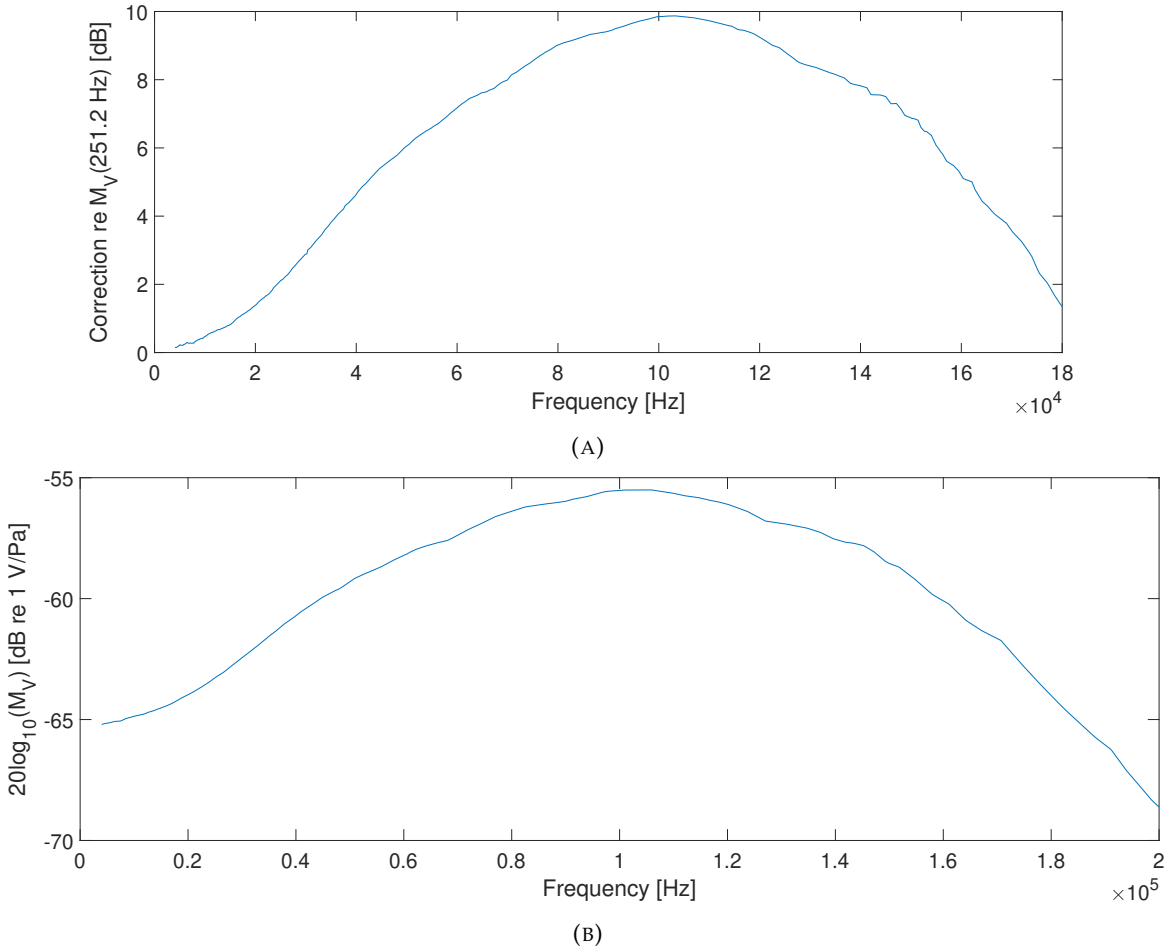


FIGURE 3.22: Calculated actuator response of microphone relative to sensitivity at 251.2 Hz (A), and full receiver sensitivity from 0 to 180 kHz.

see Fig. 3.23. The graph for 0 degree free field correction is digitized and added to the actuator response. The full receiver sensitivity $M_V(f)$ is calculated by adding the full actuator response to the $M_V(251.2Hz)$, and is shown in Fig. 3.22

To avoid having to deal with the angular dependency of the receiver sensitivity, the microphone is kept at the same position for all experiments, instead measuring the element directivity by rotating the element. However, when moving the element closer to the microphone, the sound emitted at different points on the element surface begin to arrive at the microphone at different angles. In the far field, it can be assumed that the sound waves travel together, as the equivalent of a plane wave, but in the near field, the pressure experienced by the microphone surface is the sum of many waves from different angles. It therefore becomes impossible to determine the correction that needs to be applied, to calculate an accurate value of the pressure in the near field of the transmitting element. For the measurements done within the near field of the element, it is more interesting to see if the shape of the pressure field is similar to the simulations, with the same peaks and valleys in the pressure field.

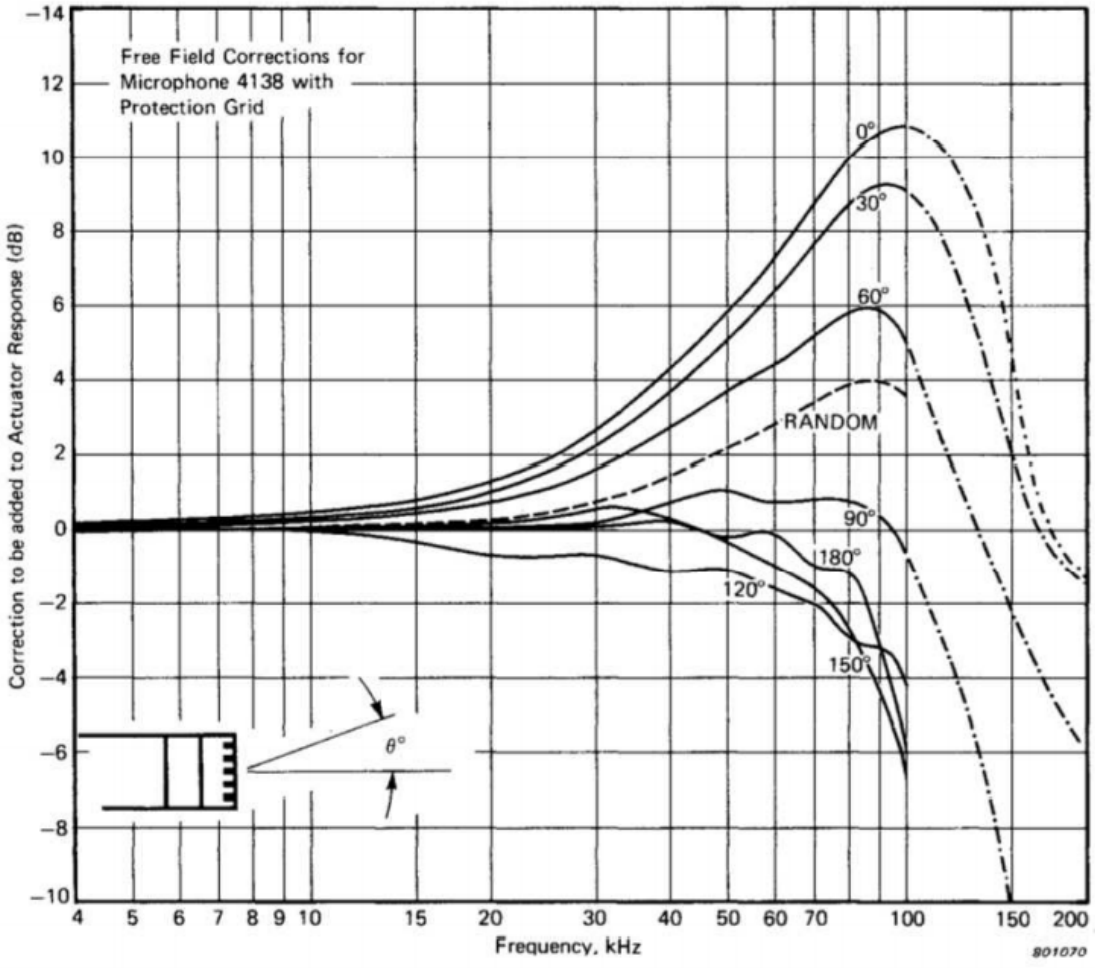


FIGURE 3.23: Correction applied to the actuator response of the microphone, at different angles of incidence of the sound waves.

3.7 Signal setup and processing

3.7.1 Signal setup

The signals used during this work is of variable length, depending on the position of the transmitter, as well as what is being measured, but in general the signal length is at least long enough so that the transmitting element reaches a steady-state, where the true V_{pp} can be determined. The maximum signal length is determined by trying to reduce reflections, and reduce the chance of standing waves between the element and microphone. This is achievable in most of the far field. The spacial length of the signals can be found using the equation

$$s_{sig} = c_{air} \cdot t_{sig} = 343\frac{m}{s} \cdot t_{sig} \tag{3.13}$$

where s_{sig} is the spacial length of the signal, c_{air} is the sound speed in air, and t_{sig} is the temporal length of the signal.

The burst length used in this work can be found in Table 3.5.

TABLE 3.5: Pulse lengths used during measurements, with spacial length calculated.

Type of measurement	Pulse length
2D Sound Field measurements 40 cm to 5 cm 98100 Hz (farfield)	1.5 ms ($\approx 51cm$)
2D Sound Field measurements 5 cm to 1.5 cm 98100 Hz (nearfield)	0.75 ms ($\approx 26cm$)
2D Sound Field measurements 1 cm to 0.1 cm 98100 Hz (nearfield)	0.75 ms ($\approx 26cm$)
On-axis measurements 50 kHz - 180 kHz	1.4 ms ($\approx 48cm$)
Directivity measurements at > 20 cm	1.4 ms ($\approx 48cm$)
Directivity measurements at 20 cm	0.6 ms ($\approx 20cm$)

In the far field outside approximately 50 cm, the pulses are short enough to not fill the entire space between the transmitter and receiver, so there is no danger of standing waves developing. As the transmitter is moved closer to the receiver, the sound pulses start filling the space between, and below approximately 17 cm, a 1.5 ms pulse has time to travel the separation distance three times during the time span of the signal, causing possible standing waves. However, the receiving microphone has a relatively small receiver surface, so the reflections off the microphone back towards the transmitting element is assumed to be minimal in the far field. Standing waves seem not to cause any significant problems until the microphone is placed inside the near field of the element, where the waves has the capacity to bounce back and forth several times within the original pulse time length. This causes the amplitude measurements to have significant uncertainties when the separation distance is reduced.

An example of a transmitted pulse is shown in Fig. 3.24. Examples of received pulses are shown in Figs. 3.25 and 3.26, showing pulses of 98.1 and 180 kHz respectively.

3.7.2 Postprocessing of sinusoidal wave pulse

The oscilloscope records an average of 128 pulses, and the resulting signal is saved on the PC. The horizontal scale on the oscilloscope depends on the signal length, but with the signals being used here, the horizontal scale is either 1 or 2 ms. The oscilloscope is set to

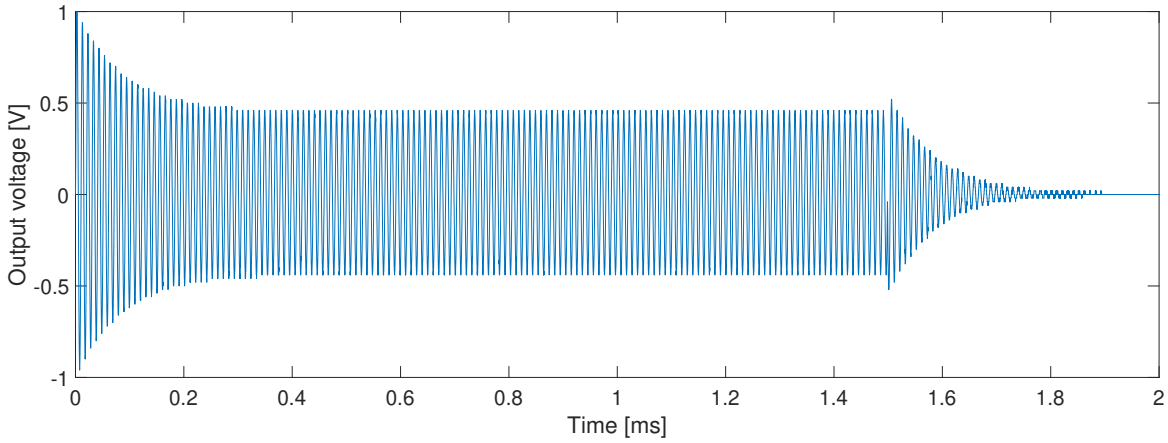


FIGURE 3.24: Example of recorded pulse from signal generator, V_{1m}, with a frequency of 98.1 kHz.

record 100000 samples, but in reality the data vectors saved on the PC is 99991 samples long. This corresponds to a sample rate of

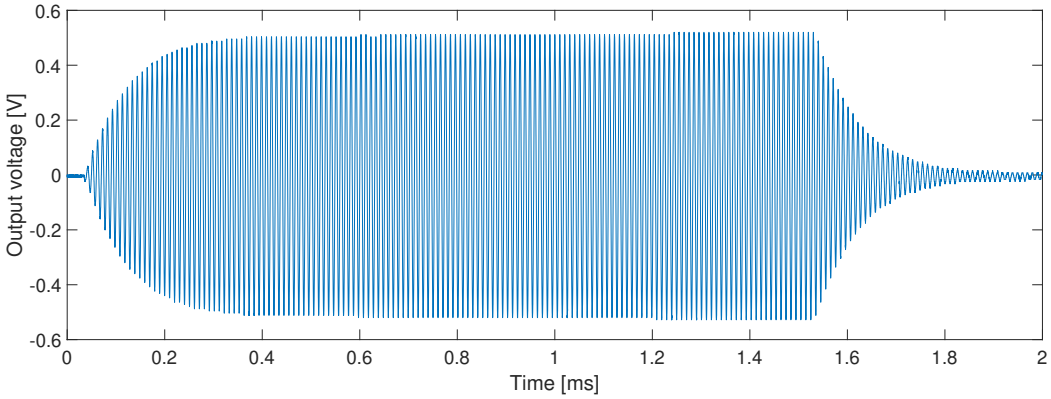
$$f_s = \frac{N}{T} \tag{3.14}$$

where N is the number of samples, and T is the time length of the recorded window of the oscilloscope. For the two different recording windows used, 1 ms and 2 ms, the theoretical sample rate should equal 100 MHz and 50 MHz respectively, but the data vectors saved from the oscilloscope are 99991 samples long, equalling a sample rate of 99991000 Hz and 49995500 Hz respectively. The reason for the difference in sample number is not explored further.

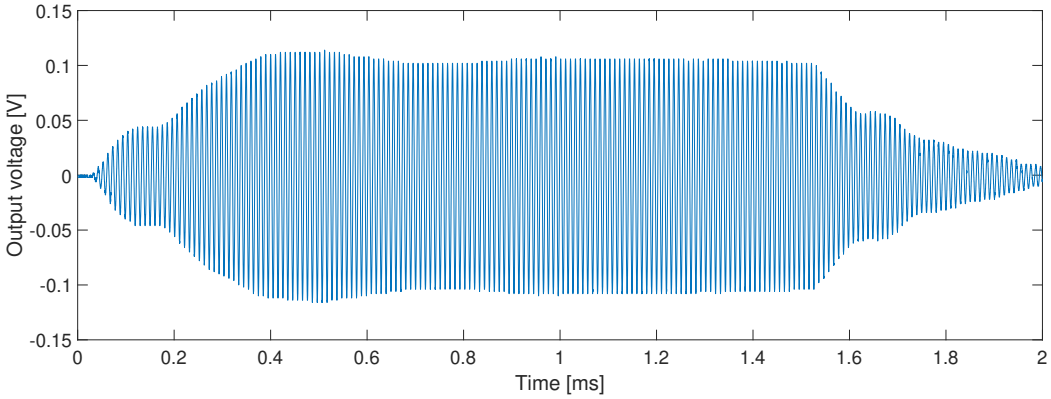
The vertical resolution of the recorded wave forms is dependant on the vertical scaling chosen, and the 8-bit resolution of the oscilloscope. It can be seen that with the 8-bit architecture limiting the resolution, the wave form data comes out looking rather jagged and stairlike. It is preferable to smooth out these wave forms before further post-processing is applied. This is done using a Savitzky-Golay filter. [46]. Summed up, the Savitzky-Golay filter tries to fit a polynomial of a chosen degree to the data set, using a sample window of a chosen size. In this work, a Savitzky-Golay filter function from MATLABs signal processing toolbox is used. For the wave forms being worked with here, a polynomial of fifth degree, and a sample window of 201 samples is used. The effect of this filter can be seen in Fig 3.27. For this particular waveform, the signal recorded has a V_{pp} of approximately $V_{pp} \approx 1$ V. When the oscilloscope scaling is adjusted to fit this waveform, the 8-bit nature of the oscilloscope causes the voltage to be recorded in steps of $0.02V$, causing the stairlike shape. If the waveform peaks were determined by the unfiltered waveform, it would have an uncertainty of ± 0.02 V. The filtering smooths out the curve, so the peaks are closer to the actual value.

3.7.3 Choice of calculation interval

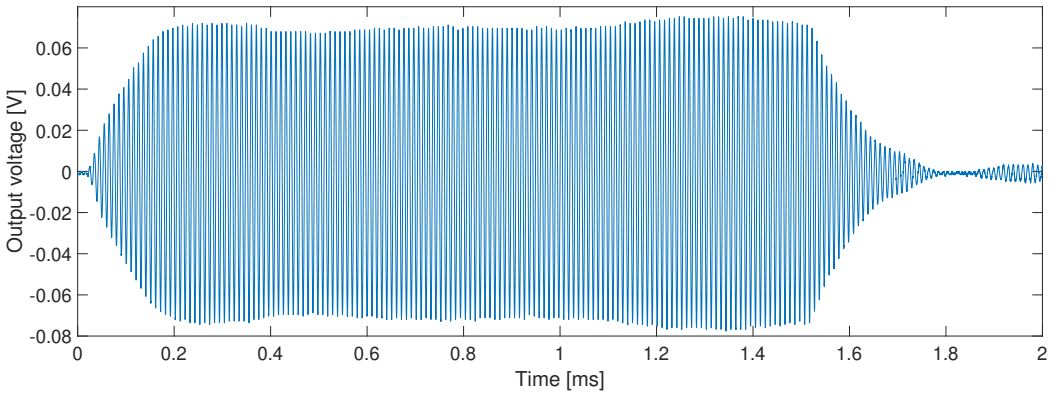
The interval of the pulse where the V_{pp} will be calculated will be chosen by eye. The assumption is that the pulses reach steady-state after approximately the same time period, but as shown in Fig. 3.25, the shape of the pulses can vary depending on the angle of the element. Visual inspection with extended signals show that in some angle ranges, the signal never reaches steady-state, but remains unstable and chaotic. This effect is most prevalent in the dips of the element directivity. Still, a calculation interval has to be determined, and for



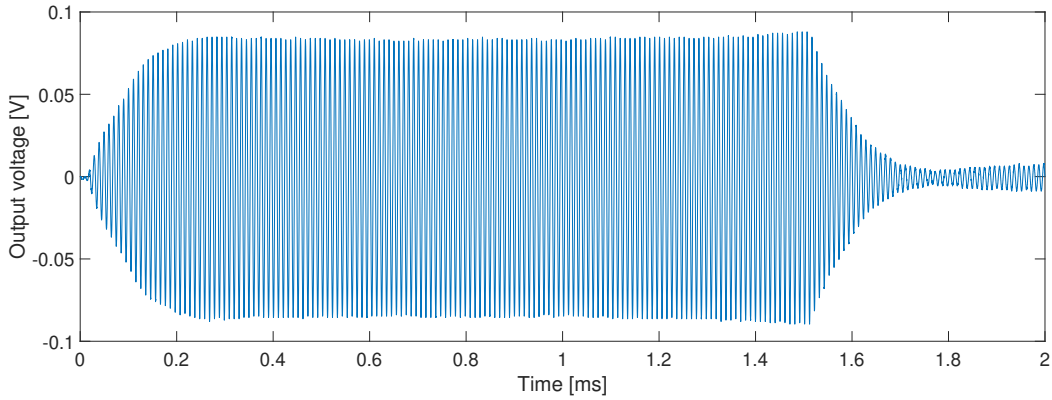
(A)



(B)

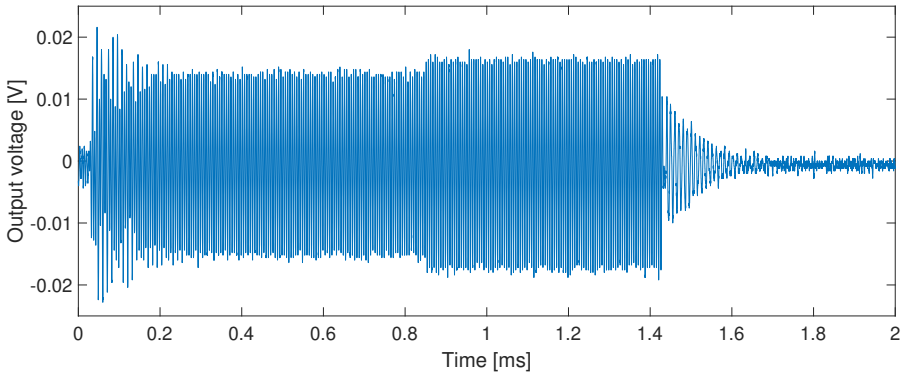


(C)

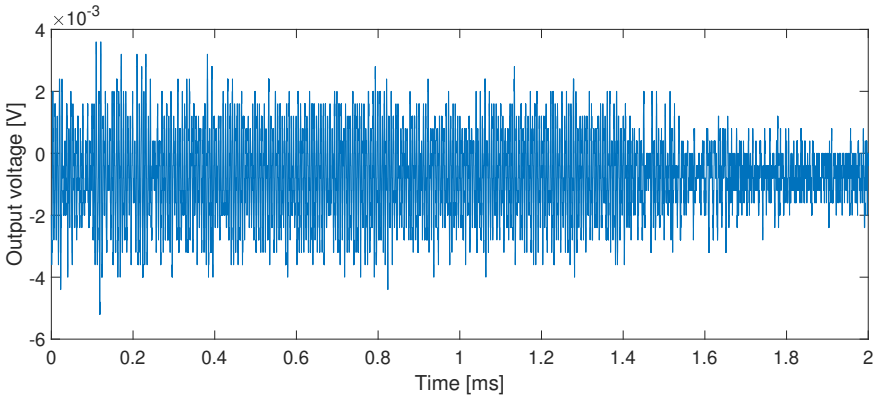


(D)

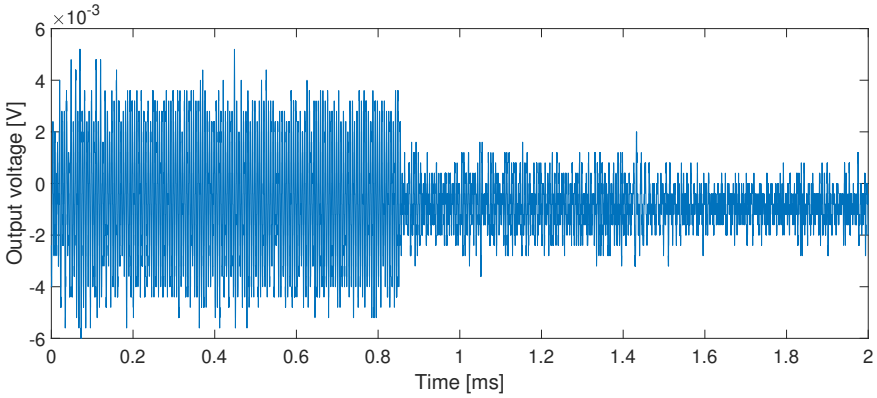
FIGURE 3.25: Example of recorded pulse from receiver electronics, V_{5m}, with a frequency of 98.1 kHz. Element/microphone separation at 20 cm, and element angle at a) 0 b) 20 c) 40 d) 60 degrees.



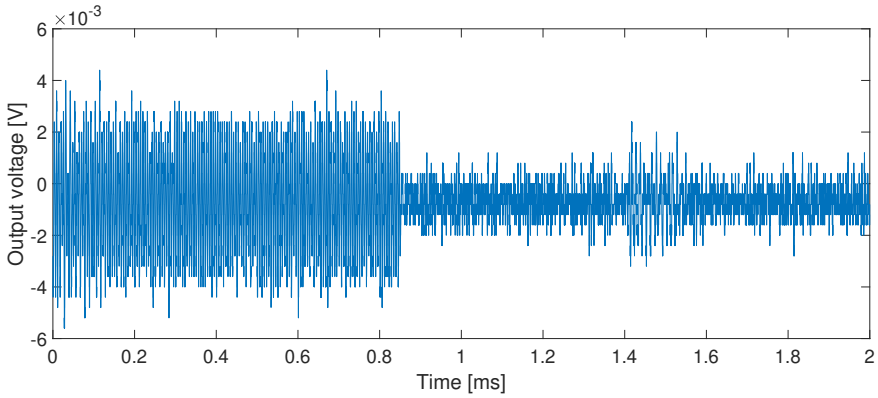
(A)



(B)



(C)



(D)

FIGURE 3.26: Example of recorded pulse from receiver electronics, V_{5m}, with a frequency of 180 kHz. Element/microphone separation at 20 cm, and element angle at a) 0 b) 20 c) 40 d) 60 degrees.

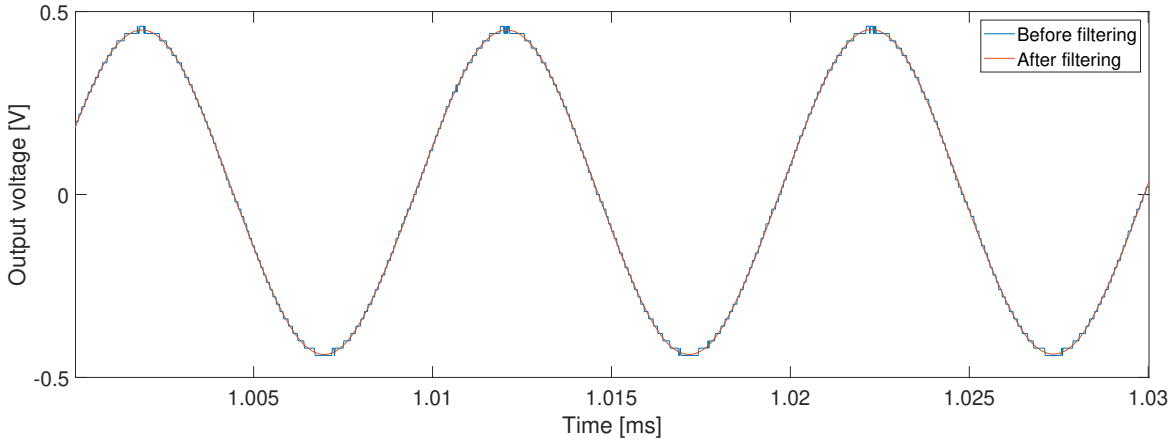


FIGURE 3.27: Demonstration of Savitzky-Golay filter on a recorded waveform, f = 98.1 kHz.

consistency and ease of work, the same interval is chosen for all angles in a set of measurements with the same frequency. Preferably, the interval chosen is as early in the waveform as possible, to reduce the chances that reflections are included. An example is shown in Fig. 3.28, where the reflections off the element mounting rod reaches the microphone at around 1 ms. This causes constructive interference, and the V_{pp} increases. This type of reflection is talked more about in section 3.5.2. For the measurements of the transfer function and on-axis pressure, all measurements are done on the sound axis of the element, and therefore within the main lobe, so there is no danger of unstable wave forms.

An effect that was unfortunately not considered until late in this work, was the effect the electric field produced by the transmitting element would have on the receiver. In Fig. 3.26, the pulses show a change in amplitude at around 0.85 ms. For most of the time working with this data, this change was believed to be because of the constructive or destructive interference expected from reflections off the transmitter mounting rod, discussed more in Section 3.5.2. As it turns out, the effect is caused by interference from the electrical signal transmitted to the element. As the signal pass through the element, a magnetic field is created, overlapping the microphone electronics. This field induces a voltage in the microphone, matching the signal transmitted to the element. The graphs in Figs. 3.25 and 3.26 only shows the received pulse, but the oscilloscope shows both the transmitted and received pulses together on the screen, see Fig. 3.5. The interference occurs where the transmitted and received pulses overlap. When working with pulses at the element resonance, the received pulses had a high enough amplitude so that the interference was undetectable, but when measurements were done with non-resonance frequencies, the amplitudes of the received signals were much lower, so the effect became much more pronounced. As new measurements could not be done, due to time constraints, a workaround had to me found. By choosing a new calculation interval for the FFT-transforms that did not include the areas of interference from the electrical signal, the effects could be avoided. This new calculation interval, however, falls within the time range where interference from reflections could have an effect, so the values of V_{5m}^{pp} calculated from this interval would have a higher uncertainty than necessary. The interference from the transmitted signal could have been avoided, if either the transmitted signal was shorter, or the measurement separation was larger. This would make it so the transmitted and received signals did not overlap in time. Some measurements were done after this effect was discovered, and there the signal parameters were tweaked so that no interference was present.

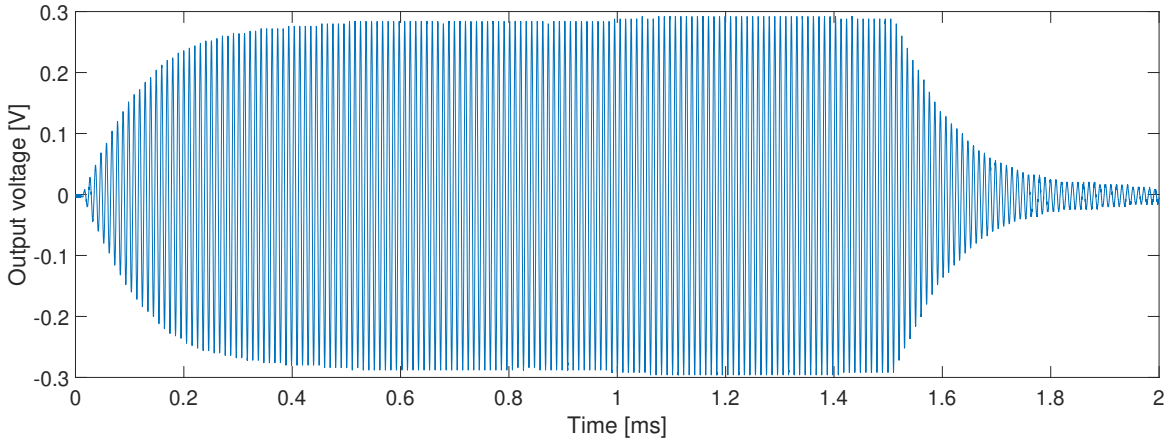


FIGURE 3.28: Recorded waveform, with element at 90 degrees and 10 cm separation.

3.7.4 Calculation of peak to peak voltage, Fourier method

Two methods of calculating V_{pp} exist and has been used in earlier works. The first is the direct method, where the peak values of the wave forms are measured directly. The second method, and the method used in this work, is the Fourier transform method. Here the calculation interval of the waveform is converted from being time dependant, to frequency dependant, using the FFT algorithm built into MATLAB. The full peak to peak FFT calculation is done using the MATLAB script $findPeakToPeak\_FFT\_k.m$, see Appendix A. The waveform within the calculation interval is isolated, and the interval is adjusted so the start and end point of the new signal fall on zero points, so the signal interval evaluated contains a whole number of periods. The new signal is then zero padded up to a length determined by taking $20 \cdot signal\ length$, and then finding the next value equal to a power of 2, which is required to do a fast Fourier transform, or FFT. The zero padding also increases the frequency resolution of the following $V_{pp}(f)$ vector.

$$V_{sig}(t) \xrightarrow{FFT} V_{sig}(f) \tag{3.15}$$

To find the actual V_{pp} of the recorded signal, we need to multiply the FFT signal vector by a factor of 4, and divide it by the number of samples in the signal. The factor of 4 comes from the fact that the "energy" of a Fourier transformed signal is split between positive and negative frequencies, so each side is half of the real value. Also, the FFT graph shows the peak amplitude, when we want to find the peak to peak amplitude, so we multiply the signal by $2 \cdot 2 = 4$.

Figs. 3.29 and 3.30 show that the method gives good results, with the peak of the frequency domain graph matching the frequency and peak to peak amplitude of the signal waveform. Sometimes, the peak frequency doesn't exactly line up with the chosen signal frequency, so the FFT vector is interpolated to the desired frequency.

$$V_{pp}(f_{sig}) = 2 \cdot 2 \cdot \frac{V_{sig}(f_{sig})}{N_{sig}} \tag{3.16}$$

$V_{sig}(f_{sig})$ is the output vector of the FFT transform, which describes the voltage of the signal over the whole frequency spectrum. N_{sig} is the number of sample points in $V_{sig}(f_{sig})$

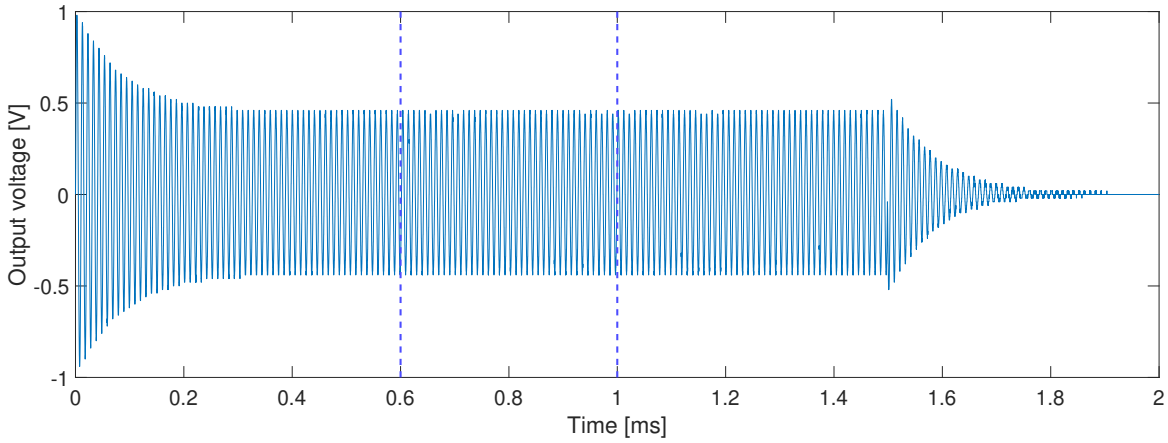


FIGURE 3.29: Example of a waveform recorded off the signal generator, with calculation interval marked by blue lines.

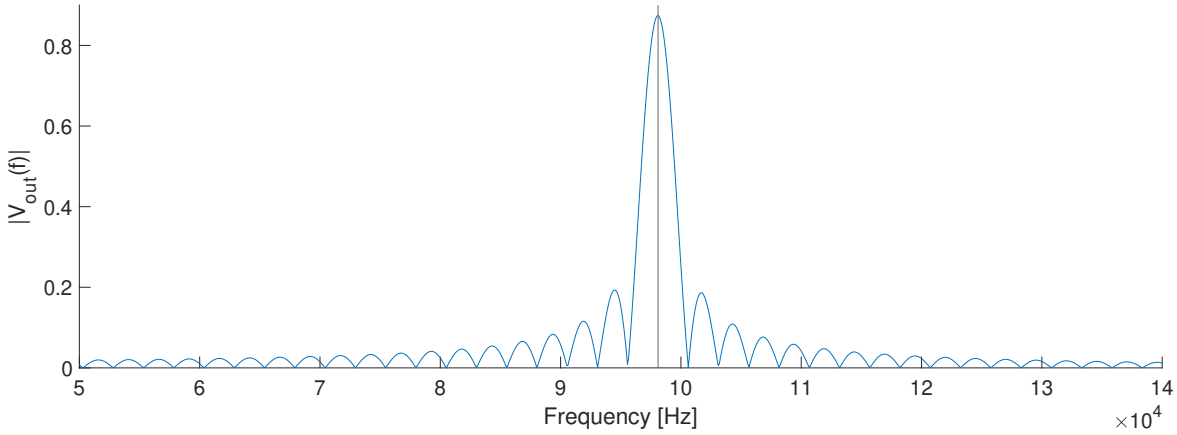


FIGURE 3.30: $V_{pp}(f)$ of waveform above, with signal frequency, 98.14 kHz, marked.

3.7.5 Calculation of pressure by using receiver sensitivity

To calculate the on-axis pressure and 2D pressure field, the recorded V_{pp} of the microphone needs to be converted to pressure. For this, the calculated receiver sensitivity of the microphone is used. The equation for recorded pressure is

$$p_{pp}^{rec}(f) = \frac{V_{pp}(f)^{rec}}{M_V(f)} \tag{3.17}$$

where $p_{pp}^{rec}(f)$ is the calculated peak-to-peak pressure, $M_V(f)$ is the frequency dependent receiver sensitivity of the microphone and receiver electronics, and $V_{pp}(f)^{rec}$ is the recorded peak to peak voltage V_{5m}.

The calculation of pressure is done using the MATLAB script V_to_Pa.m, see Appendix A. Note that 60 dB is subtracted from the recorded V_{pp}, to compensate for the gain from the amplifier. Also note that the pressure values given by the simulation program FEMP is the pressure amplitude, so to directly compare the measured and simulated pressure, the calculated peak-to-peak pressure is divided by 2.

3.7.6 Calculation of SNR

The noise measurements are recorded as its own measurement, recording the time window right before the arrival of the transmitted signal. A time window of 2 ms, and a vertical scaling of 10 mV/div is used. The recorded noise is used to calculate SNR (signal-to-noise ratio), for the measurements performed in this work.

The root-mean-square amplitude of the noise is calculated as [11]

$$V_{rms}^{noise} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(V_i - \bar{V})^2} \tag{3.18}$$

where N is the number of recorded samples, V_i is the voltage of sample i, and $\bar{V}$ is the mean voltage of the noise. The SNR is calculated as [11]

$$SNR = 20log_{10}\left(\frac{V_{rms}^{rec}}{V_{rms}^{noise}}\right) \tag{3.19}$$

where V_{rms}^{rec} is the RMS amplitude of the recorded signal voltage of the corresponding acoustic measurement, calculated as

$$V_{rms}^{rec} = \frac{V_{pp}^{rec}}{2\sqrt{2}} \tag{3.20}$$

where V_{pp}^{rec} is V_{5m}^{pp}, the peak-to-peak amplitude of the recorded signal voltage of the acoustic measurement.

3.8 Measurement routines and data acquisition

In this section, the practical method and programs used for measurements are outlined. For all measurements performed, the instrument addresses and settings are contained in the MATLAB script instruments.m, and are initialized by the script init_instruments, see Appendix A.

3.8.1 Directivity measurement method

Before each batch of measurements, the alignment of the transmitting element and microphone is redone, as described in Section 3.4. To start a directivity measurement, the element angle is set to -92 degrees. Although the data shown spans from -90 to 90 degrees, two extra degrees on each side is recorded to enable adjustment of the zero-degree point in the data vector if the angular alignment alignment was not done properly. If that is the case, the maximum value of the directivity measurement is found, and set as the new zero-degree point. The desired separation between transmitter and receiver is set, and the measurements are performed, using angularMeasMain.m and main.m MATLAB script, see Appendix A. The element angle is controlled using the software PI mikromove [47]. Due to the lack of automation scripts for the stages used, the angle has to be changed manually. The MATLAB script pauses after each measurement, so the angle can be manually adjusted. After measurements for all angels are performed, the MATLAB script is commanded to stop, and the measured signal data and measurement parameter data for all angles are saved. This routine is done for one frequency at a time.

3.8.2 On-axis pressure measurement method

For the measurements of on-axis pressure, the alignment of the transmitter and receiver is done as in the previous section. The measurements are performed using the MATLAB script main.m, see Appendix A. The measurements are done at one separation distance at a time, across many frequencies, chosen in MATLAB. The transmitting element is placed first at its closest separation distance, and the mainAlt.m script is run, automatically running the measurements for all frequencies specified. After the script finishes, the data is saved, and the element moved out to the next separation distance. The separation value is manually updated in the measurement_parameters.m MATLAB script, to make sure the time window of the oscilloscope is adjusted to record the right timeframe, as the signal arrives later as the element is moved away from the microphone. The measured signal data and measurement parameter data is saved in separate files for each separation distance, and is put in a folder together for later post processing.

3.8.3 Transmitter sensitivity measurement method

The transmitter sensitivity measurements are less complicated than the measurements in the earlier sections. Since the transmitter sensitivity is defined at one position in space, there is no need to move the transmitting element between measurements. The element position is adjusted like before, and is moved to a separation distance of z = 0.5 m. When the transmitting element is positioned, the mainAlt.m MATLAB script is run. Measurements are performed automatically for all frequencies specified in measurement_parameters.m. The measured signal data and measurement parameter data is saved in a single file.

3.8.4 2D sound pressure field measurement method

The method for measuring the pressure across the rθ-plane is identical to the method for directivity measurements in Section 3.8.1. To achieve a full mapping of the desired points in space, the directivity beam pattern measurements are performed at many different points of separation. The measured signal data and measurement parameter data for each value of separation is saved on its own file, similar to the on-axis pressure measurements in Section 3.8.2. These files are later combined into a full 2D pressure map in post processing.

3.8.5 Instrument adjustment

As the measurements are performed, the signal received by the oscilloscope may change. To make sure all the data is recorded, the recording window of the oscilloscpe is adjusted automatically before each measurement, using the scripts adjustAmplitude.m, and adjustTime.m. In addition, the temperature is recorded for each measurement, using the script ASL_250.m. See Appendix A for all MATLAB scripts used.

Chapter 4

FE simulation setup

In this work, the finite element method is used to model and simulate the behaviour of a vibrating piezoelectric element radiating in air, as well as the surrounding fluid. A short descriptin of the current version of FEMP is given in Section 4.1. Section 4.2, gives an overview of the important simulation parameters specified for this work. and what they mean. The material parameters of the piezoelectric material and fluid used is given in Section 4.3. The method and setup of the simulation construction is presented in Section 4.4. A description and important information around the simulation of pressure is given in Section 4.5.

4.1 FEMP 6.1

The simulations in this work is performed using the latest version of FEMP available when the work started [24]. A brief description of the theory behind the software can be found in section 2.8. The software is written and run in MATLAB. The input parameters of the FEMP-simulation is distributed across several MATLAB-files. A text file with the extention .inn contains the information about the parameters that will be simulated, like admittance, on-axis pressure, and so on, as well as the desired spacial and frequency resolutions. Another text file with the extention .mat contains a selection of sets of material parameters, for many different piezoelectric materials, as well as some medium fluid parameters. MATLAB construction files read_inn_project.m and init_const_project.m takes the chosen parameters from these text files, and combines them with the simulation construct defined in this file. Here, the dimensions of the element, and the surrounding medium is chosen, as well as the position of these different parts, and the boundary conditions.

4.2 Simulation parameters

The parameters used in the FEMP simulations in this work is presented here. For the simulations done in this work, a direct time-harmonic analysis is used [24]. Complex loss is modeled inside the piezoelectric element, while the air is modeled with no loss. For the analysis of the electric behaviours of the element, a frequency range of 0-300kHz is used. For the acoustic characteristics, a frequency range of 50-180kHz is used in general, to match the frequency range of the microphone in the physical measurements. Some exceptions to these frequency ranges are found in the work, where specific frequencies of greater interest is studied.

4.2.1 Element and fluid dimensions

The dimensions of the element, and fluid radius is set in the .inn files. For the element, the radius and thickness is chosen depending on which element in the batch is being simulated. The radius chosen for the fluid describes the radial distance from the center point of the

simulation mesh, to the edge of the finite element area of the fluid. The infinite elements start at the edge of the finite element, and in this work it it set so that the infinite elements stops at a distance twice that of the radius of the finite elements.

4.2.2 Mesh resolution

The number and size of the elements used in the simulation is dependant on two factors, chosen in the .inn file. The first is the meshing frequency, and the second is the number of elements per wavelength, N_{ele}/λ. The meshing frequency dictates the wavelength used in the elements per wavelength parameter, and is usually set to or slightly over the maximum frequency that is used in the simulation. In this work the meshing frequency is set to 20 kHz above the highest frequency used.

The N_{ele}/λ parameter in the element is split into radius and thickness parameters, elr and elt, so the density in each direction can be set separately. In this work, these are kept equal. The N_{ele}/λ in the fluid is defined by the radial density, and angular density, elfluid and theta. The regular N_{ele}/λ parameters are usually set between 3 and 10, where a higher number equates a higher density of elements. The number can be increased arbitrarily, but above 10 elements per wavelength, the computing time increases drastically, and the computers available during this work does not have enough RAM to store the simulated mesh. For this work, a value of N_{ele}/λ = 7 is used for all parameters, and a theta = $\pi/4$. A convergence test run by Aanes show a maximum deviation of 0.15 % when comparing the pressure values of a simulation using an N_{ele}/λ value of 7, when compared to a simulation using a N_{ele}/λ value of 14 [48]. The finite elements can be set til 1st order, or 2nd order, corresponding to 4 or 8 nodes around each element. in this work, 2nd order elements are used. The infinite elements are set to 10th order.

4.2.3 Choice of finite element radius

The areas of finite and infinite elements in the fluid must be determined by the choice of the parameter R_{inf}, which defines the radius of the finite element area from the origin of the mesh. For 10th order infinite element, it is recommended that R_{inf} at a minimum should equal $R_{inf} = 0.32a^2/\lambda$, where a is the radius of the element, and λ is the wavelength in air of the highest frequency used in the simulations [24]. A small R_{inf} means less elements to simulate, but may affect the accuracy of the simulation. For this work, where the maximum frequency used is 300 kHz, the minimum R_{inf} is calculated to

$$R_{inf} = \frac{0.25 \cdot 0.01m^2}{\frac{343m/s}{300000Hz}} \approx 0.028m \tag{4.1}$$

To ensure the accuracy of the simulations, while keeping the computational time reasonable, R_{inf} is chosen to be $R_{inf} = 0.04$ m in the beginning of this work, when simulating admittance, and $R_{inf} = 0.025$ m, when simulating pressure. Towards the end of this work, some weird behaviour was spotted in the simulations, with on-axis pressure fluctuating randomly, when it should have been smooth. The actual cause was not determined during this work, but when the R_{inf} was reduced to 0.025 m, the fluctuations disappeared. $R_{inf} = 0.025$ was therefore used for the pressure simulations, where the maximum frequency used was 180 kHz, with a corresponding minimum R_{inf} of 0.017 m, below the 0.025 m value used.

4.3 Material parameters

The material data describes the electric, electromechanic and mechanic behaviours of the material, through a set of constants. The constants used by FEMP can be found in Table 4.1.

4.3.1 Piezoelectric element Pz27

The piezoelectric material used is a piezoelectric ceramic, Pz27, produced by Ferroperm. The producer provides provides a set of material constants [49][44], but earlier work has shown that these can be inaccurate, and that finding the constants through measurements can be preferable. The values used here were calculated in-house, by earlier master students [50][51]. The structure of the constant matrices can be seen in [24], and the values used in this work is in Table 4.1.

TABLE 4.1: Material parameters for Pz27. The adjusted data set is used in this work.

Parameter	Unit	Ferroperm data	Adjusted data set (Lohne/Knappskog)
c_{11}^E	$[10^{10}Pa]$	14.70	$11.1875(1+i\frac{1}{95.75})$
c_{12}^E	$[10^{10}Pa]$	10.50	$7.430(1+i\frac{1}{71.24})$
c_{13}^E	$[10^{10}Pa]$	9.37	$7.425(1+i\frac{1}{120.19})$
c_{33}^E	$[10^{10}Pa]$	11.30	$11.205(1+i\frac{1}{177.99})$
c_{44}^E	$[10^{10}Pa]$	2.30	$2.110(1+i\frac{1}{75.00})$
c_{66}^E	$[10^{10}Pa]$	–	$2.223(1+i\frac{1}{225.34})$
e_{31}	$[C\cdot m^{-2}]$	-3.09	$-5.400(1+i\frac{1}{-166.00})$
e_{33}	$[C\cdot m^{-2}]$	16.00	$16.0389(1+i\frac{1}{-323.77})$
e_{15}	$[C\cdot m^{-2}]$	11.60	$11.200(1+i\frac{1}{-200.00})$
ϵ_{11}^S	$[10^{-9F/m}]$	10.0005	$8.110044(1+i\frac{1}{50})$
ϵ_{33}^S	$[10^{-9F/m}]$	8.0927	$8.14585(1+i\frac{1}{86.28})$
ρ	$[kg/m^3]$	7700	7700
Q_m	–	80	–
$tan\delta$	–	0.017	–

4.3.2 Air parameters

For this work, a set of material parameters for air at 20 degress celsius and 1 atmosphere is chosen from the included material sets in FEMP. The parameters is given in Table 4.2.

TABLE 4.2: Material parameters of fluid used in this work, air.

Parameter	Unit	Value
Density $[\rho]$	$[kg/m^3]$	1.21
Sound speed [c]	[m/s]	343

4.4 Structure setup

For this work, the FEMP structure file *piezodiskfluid* is used, with some modifications. The construction files can be found in appendix A. Here, the origin of the mesh is chosen as the centre point on the front radiating surface of the element.

An example of the structure mesh is shown in Figs. 4.1 and 4.2. Here, the elements per wavelength has been reduced from 7 to 3, so the individual elements can be seen clearly. The cyan line marks the area of applied voltage, and the dark blue line represents the are of zero voltage restraint. These two lines represent the electrodes on the element in real life.

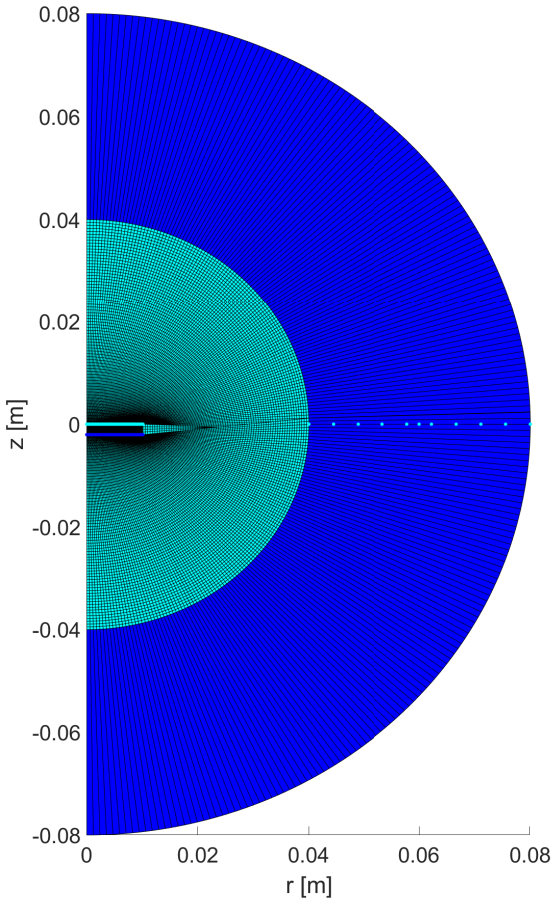


FIGURE 4.1: Structure mesh, with 3 elements per wavelength to show individual elements, and a meshing frequency of 200 kHz.

4.5 General pressure simulations

Note that the voltage on the real piezoelectric element is not constant based on the frequency of the induced pulse waveform. The real V_{pp} of the waveform is dependant on the perceived resistance of the element, which is low at the resonance frequency of the element, but increases dramatically as the pulse frequency moves away from the resonance, (see Section 2.7.2). To get an accurate comparison between the simulated pressure amplitudes, and the calculated measured pressure amplitudes, the simulation needs to be adjusted. Luckily,

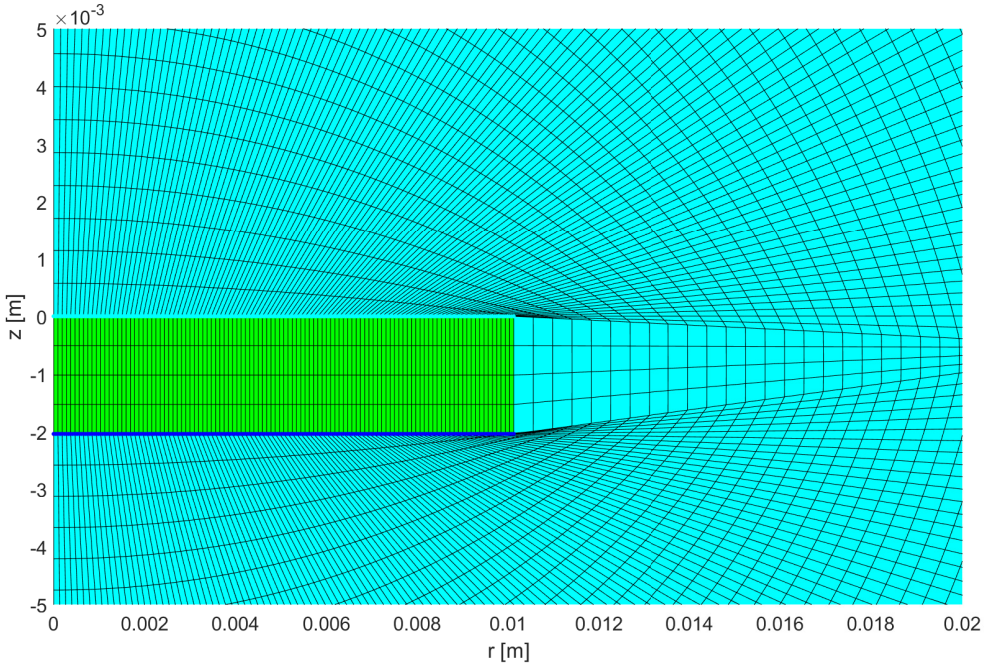


FIGURE 4.2: Structure mesh, with a reduced value of 3 elements per wavelength to show individual elements, and a meshing frequency of 200 kHz. Zoomed in to the region around the piezoelectric element.

the pressure amplitudes given by the simulation is linearly dependant on the input voltage, so to make the simulation equal to the experimental setup, you only have to scale the simulated amplitudes so that the simulated V_{pp} is equal to the measured one. The standard voltage amplitude used in this work that is induced on the element in FEMP is $V_p = 1$, so the peak-to-peak value is $V_{pp} = 2$. The sound field pressure amplitudes is therefore given by the equation [4.2]

$$[p] = [p_{sim}]\frac{V_1}{2} \tag{4.2}$$

Where p is the simulated pressure value generated from an equivalent voltage used in the measurements, p_{sim} is the pressure value given by the simulation, using a signal voltage of V_{pp} = 2 V, and V_1 is the actual peak-to-peak voltage across the transmitter element in the setup, calculated using the measured voltage V_{1m}.

Chapter 5

Results and Discussion

In this chapter, the measurements performed during this work is presented, and in some instances compared with corresponding FEMP simulations. In Section 5.1, the measurements of the piezoelectric element dimensions are presented. Section 5.2 present the measurements and simulations of the electrical parameters of the piezoelectric elements used. In Section 5.3, the rationale of choosing which frequencies to use when comparing measurements and simulations is discussed. Section 5.4 present the measurements and simulations of the on-axis pressure, as well as discussions around the data gathered.

5.1 Measurements of element dimensions

For this work, a batch of seven Pz27 piezoelectric elements was provided. These elements were produced by Meggitt A/S, and has a standard diameter of 20 mm, and a standard thickness of 2 mm. To make the simulations of each element more accurate, the dimensions of each element has been measured, and these parameters are used in the FE simulation. The measurements were done using a Digimatic Micrometer, model MDH-25M, produced by Mitutoyo Corporation, with a resolution of 0.0001 mm. The uncertainty given by the producer is 0.35 micrometers, or 0.00035 mm.

The diameter measurements were done by taking ten measurements at various points along the circumference. The element may have different diameters at different points around the edge, so an average of the measurements is used. Since the FEMP simulation software can only work with one diameter value, the mean of these measurements is used, and the uncertainty of the measurements are calculated, see Table 5.1. The thickness measurements are done in a similar manner to the diameter measurements. The thickness is measured ten times, at different points on the element surface. Again, slight differences in measurement is observed, so an average of the measured values is used for the simulation. The dimensional values for all the piezoelectric elements used in this work is shown in Table 5.1.

The uncertainties of the thickness and diameter values are calculated by combining the uncertainty of the micrometer, and the uncertainty of the repeated measurements.

The standard deviation of the measured values is given as [52]

$$s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{5.1}$$

where N is the number of measurements performed, x_i is the individual measurement value, and $\bar{x}$ is the mean value of all measurements.

The standard uncertainty is given as [52]

$$s_{\bar{x}} = \frac{s}{\sqrt{N}} \tag{5.2}$$

where s is the standard deviation, and N is the number of measurements.
The combined uncertainty is then given as [52]

$$s_{total} = \sqrt{s_{\bar{x}}^2 + \left(\frac{u}{\sqrt{3}}\right)^2} \tag{5.3}$$

assuming rectangular distribution of the instrument uncertainty u = 0.00035 mm. This combined uncertainty is shown in Table 5.1.

TABLE 5.1: Dimensional measurements and uncertainties of element batch. Values are mean of ten measurements.

Element number in batch	Thickness [mm]	Diameter [mm]
1	2.0435 ± 0.0004	20.249 ± 0.004
3	2.0275 ± 0.0005	20.255 ± 0.005
8	2.0407 ± 0.0005	20.239 ± 0.003
9	2.0023 ± 0.0008	20.219 ± 0.003
17	2.0207 ± 0.0004	20.238 ± 0.003
18	2.0402 ± 0.0005	20.243 ± 0.004
19	2.0268 ± 0.0005	20.241 ± 0.004

5.2 Electric properties of piezoelectric element

In this section, the electric properties of the piezoelectric elements will be studied and compared, both to each other, and to the simulations. The measurements are performed as described in Section. 3.2.

5.2.1 Comparison of admittance measurements

The measurements of the electric properties of the elements are done using a Hewlett Packard 4192A LF impedance analyzer. The frequency range used is 1-300 kHz, covering the two first radial modes of the elements. An OSC voltage of $V_{OSC} = 0.3V$ is used during these measurements. Measurements are performed on all elements specified in Section 5.1. The conductance, susceptance, and admittance magnitude and phase of all the elements are shown in Figs. 5.2 - 5.6.

Based on the resonance peaks of the admittance graphs 5.4 and 5.6, the first radial modes of the elements used falls within the frequency range of f_{r1} = 98650 to 99400 Hz, and the second radial modes are within the range $f_{r2} = 249950\ to\ 251650Hz$. The maximum difference in magnitude observed is 0.2 dB at R1, and 0.25 dB at R2. The admittance of Element 9 and 17 are shown to be outliers, with higher frequency resonance peaks than the rest of the batch. According to the diameter measurements in Table 5.1, these two elements are the ones with lowest diameter, which can be correlated with a shorter wavelength for the standing waves in the diameter direction, and therefore a higher resonance frequency. Although the dimensions probably has an effect on the resonance, other factors like differences in material properties within a batch, and internal structural differences, can also have an effect on the measured values.

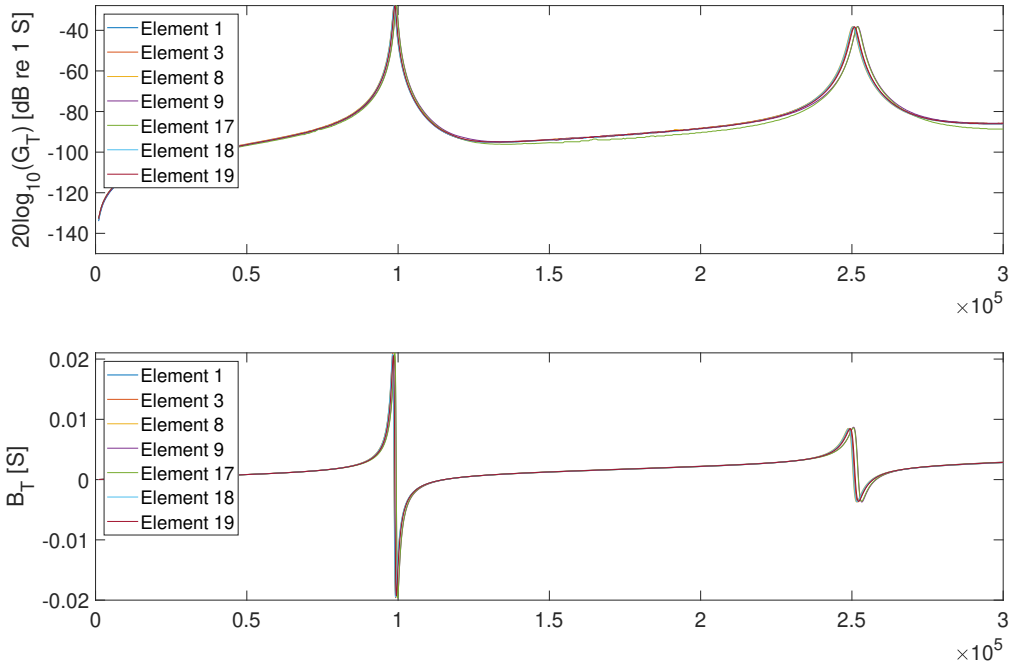


FIGURE 5.1: Conductance and susceptance for all elements, from 1 to 300 kHz.

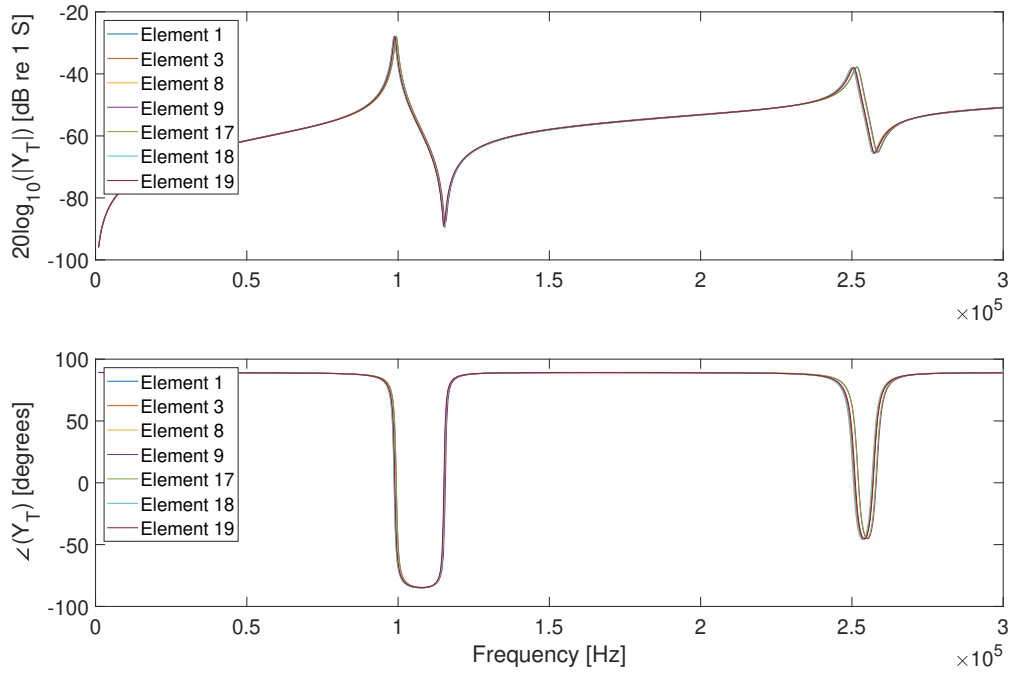


FIGURE 5.2: Admittance magnitude and phase for all elements, from 1 to 300 kHz.

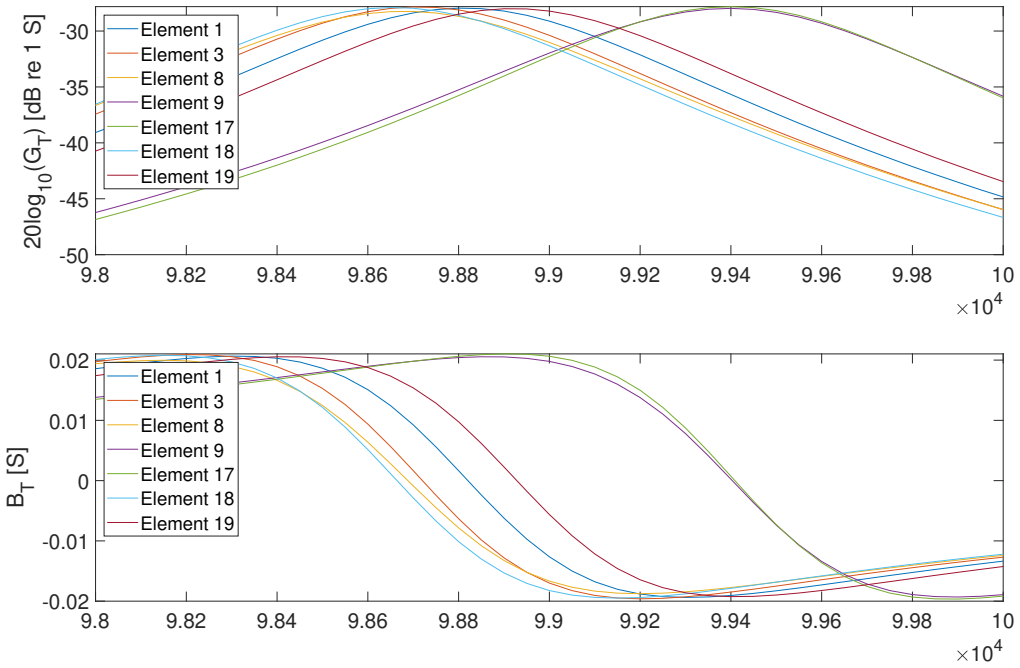


FIGURE 5.3: Conductance and susceptance for all elements, for the frequency range around R1.

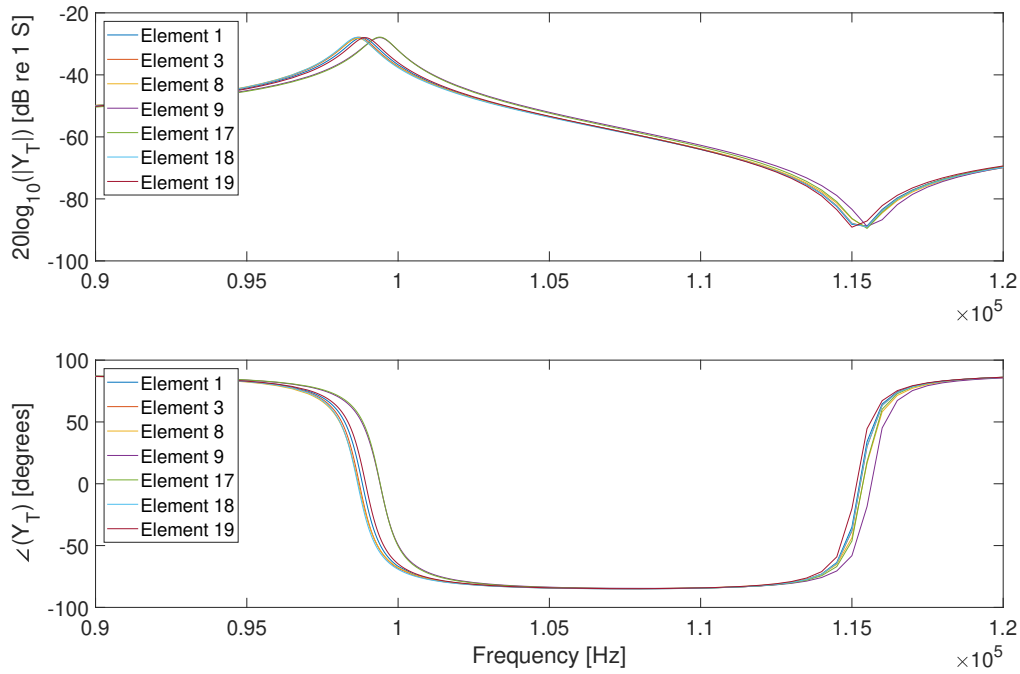


FIGURE 5.4: Admittance magnitude and phase for all elements, for the frequency range around R1.

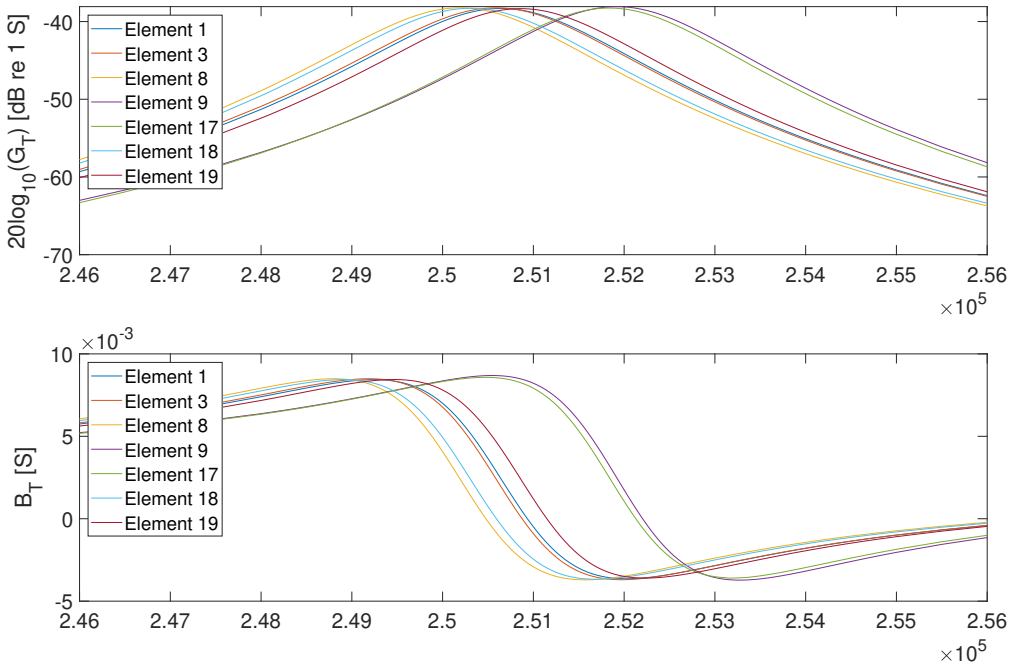


FIGURE 5.5: Conductance and susceptance for all elements, for the frequency range around R2.

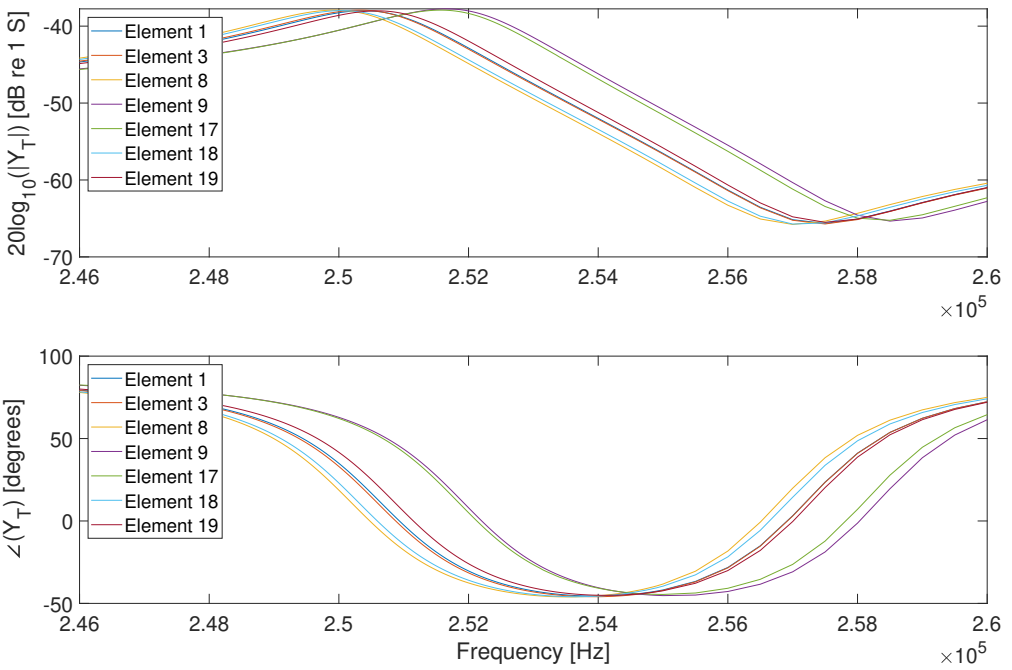


FIGURE 5.6: Admittance magnitude and phase for all elements, for the frequency range around R2.

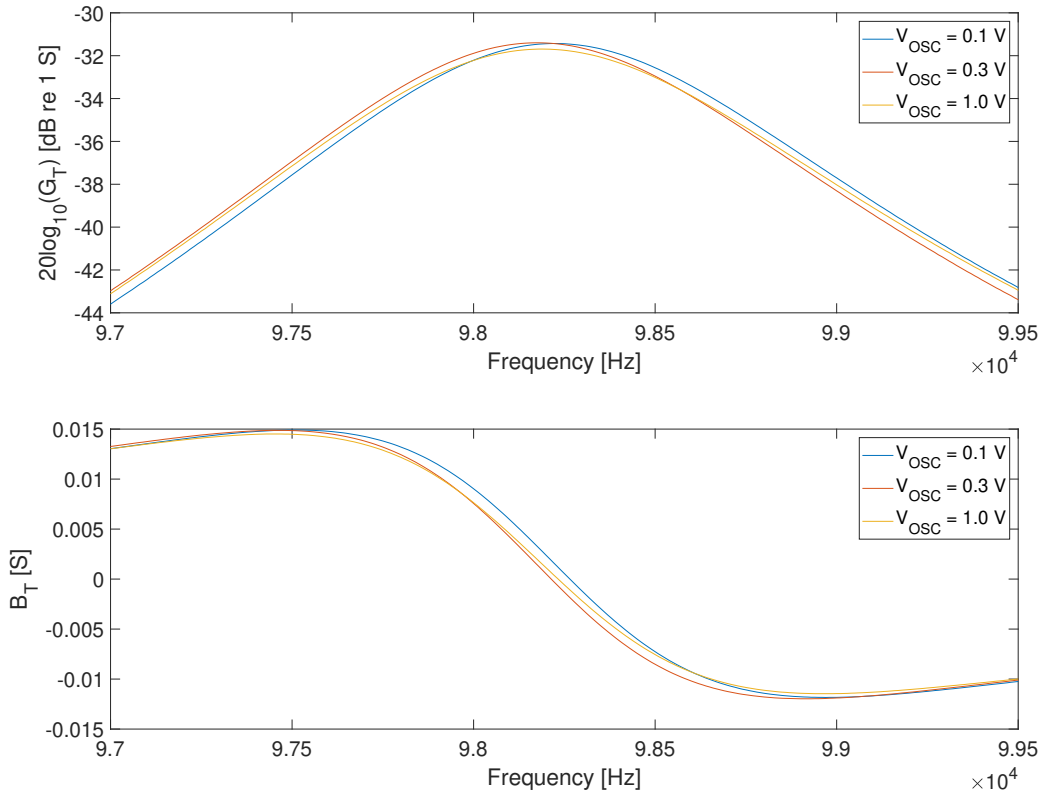


FIGURE 5.7: Conductance and susceptance of element #8. Comparison of measurements with different drive voltages. For the frequency range around the first radial mode (R1) of the piezoelectric element.

5.2.2 Effect of OSC voltage

The OSC voltage amplitude applied to the elements during the measurements can have en effect on the element characteristics. When the voltage becomes too high, the element can start to show non-linear effects, which can distort the desired measurement parameters. However, a lower amplitude increases the uncertainty of the measurement, as was shown by Mosland [2]. To check if the OSC voltage has a significant effect on the result, measurements of the same element were done with different voltages, in this case $V_{OSC} = 0.1$, $V_{OSC} = 0.3$ and $V_{OSC} = 1$.

It can be seen in Figs. 5.7 and 5.8, that the change in OSC voltage causes a slight shift in resonance peak frequency and magnitude. For this particular measurement, the maximum difference is around 100 Hz, and 0.2 dB re 1 S. As a compromise between the higher inaccuracy at $V_{OSC} = 0.1$, and the higher chance for non-linear effects at $V_{OSC} = 1$, a drive voltage of $V_{OSC} = 0.3$ is used for the admittance measurements of the other elements.

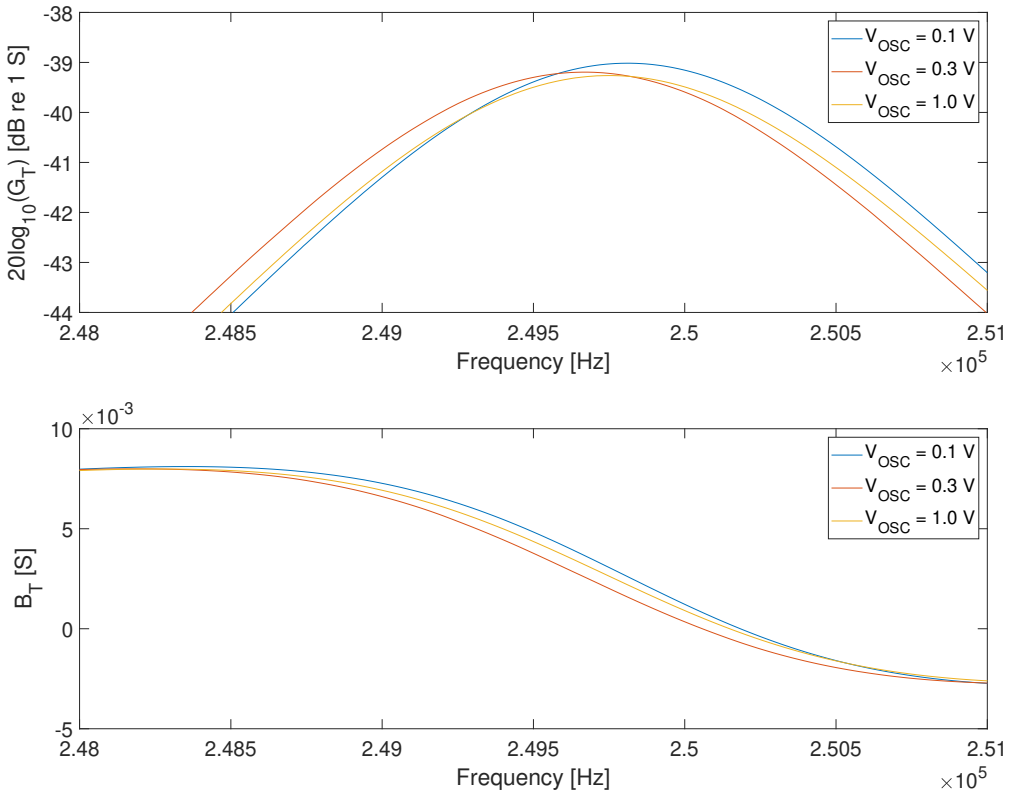


FIGURE 5.8: Conductance and susceptance of element #8. Comparison of measurements with different drive voltages. For the frequency range around the second radial mode (R2) of the piezoelectric element.

5.2.3 Comparison between measurements and simulations

Here, the admittance measurements and simulations of the elements are compared, see Figs. 5.9 - 5.15. A new simulation is run for each element, with the dimensions measured in section 5.1. Note that the frequency resolution of the measured and simulated admittance can differ from each other. This is because of compromises when it comes to the the time it takes to measure or simulate a frequency range.

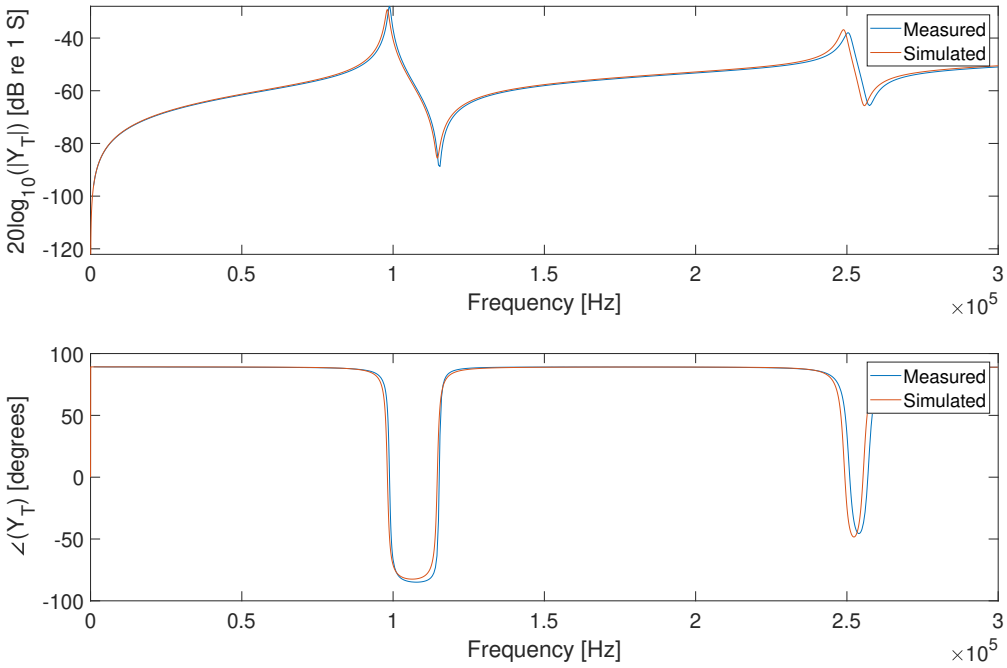


FIGURE 5.9: Measured and simulated admittance magnitude and phase of element 1.

For the piezoelectric elements studied in this work a general pattern can be seen in the comparisons between measured and simulated values. In general, the resonance peaks of the elements fall higher than what the simulations would suggest. Also, the magnitude of the measured admittance is higher than the simulations at the first radial mode, and lower at the second radial mode. The actual differentials between the measured and simulated admittance peaks at R1 and R2 are shown in Table 5.2. The element that showed the closest correlation with the simulations was element 8, so this element was chosen as the element used as transmitter in the pressure measurements to come.

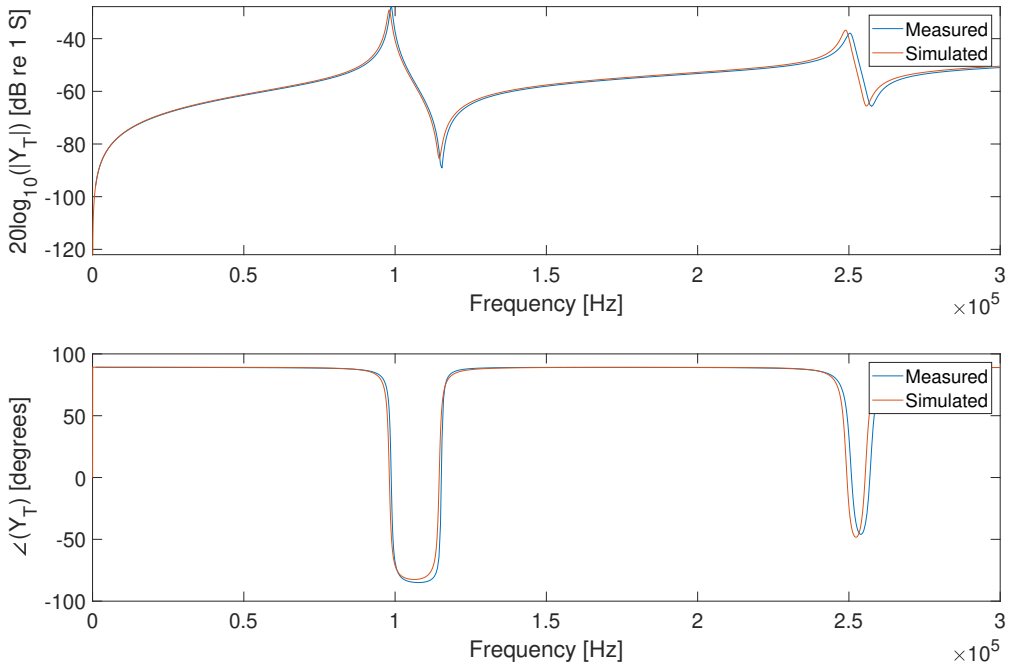


FIGURE 5.10: Measured and simulated admittance magnitude and phase of element 3.

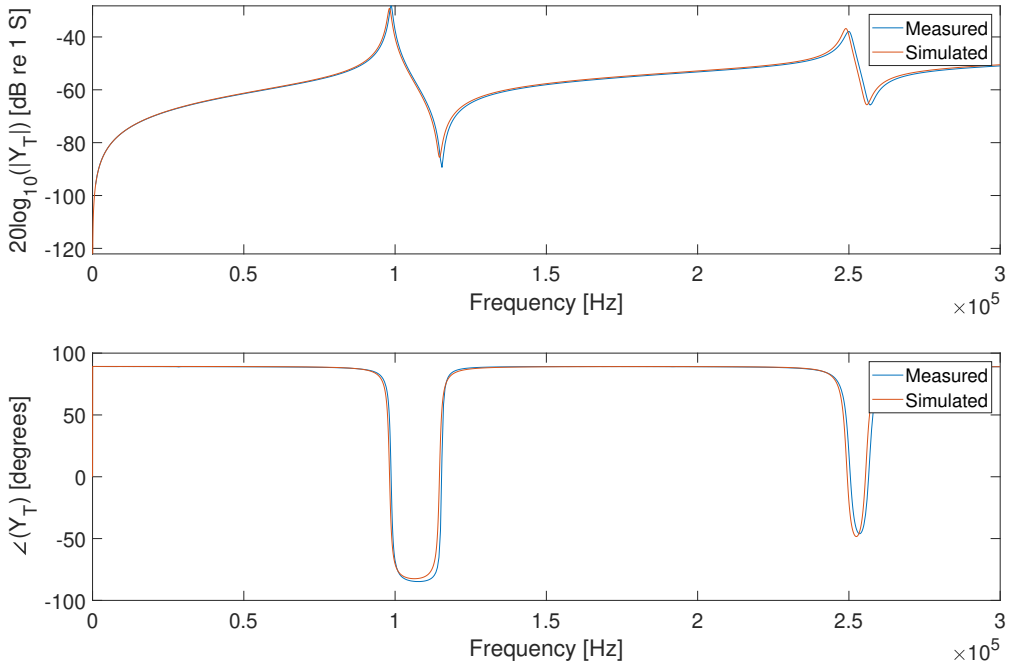


FIGURE 5.11: Measured and simulated admittance magnitude and phase of element 8.

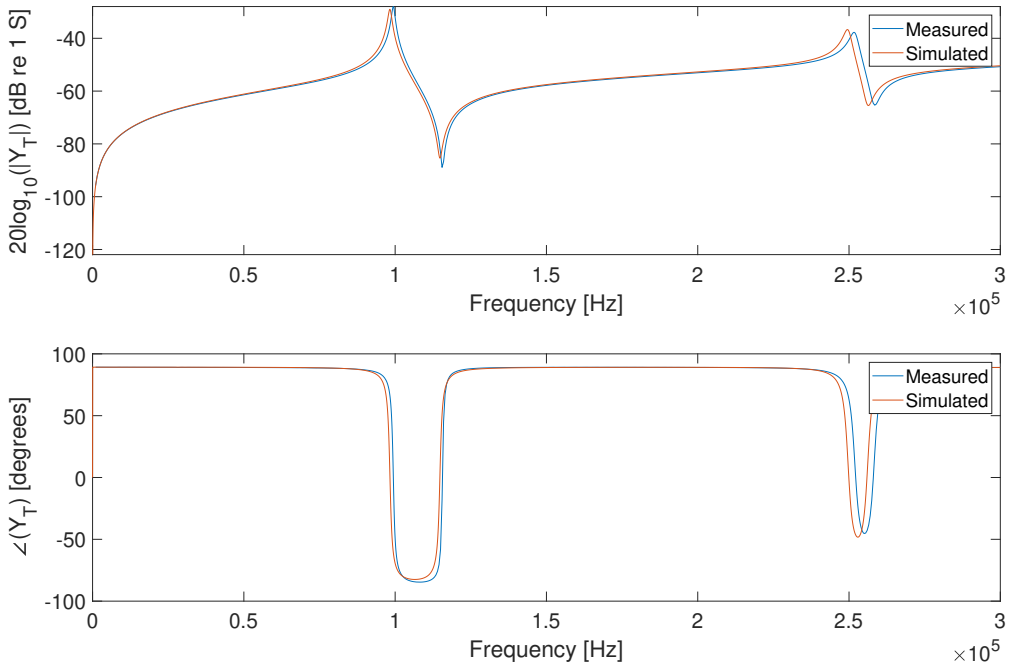


FIGURE 5.12: Measured and simulated admittance magnitude and phase of element 9.

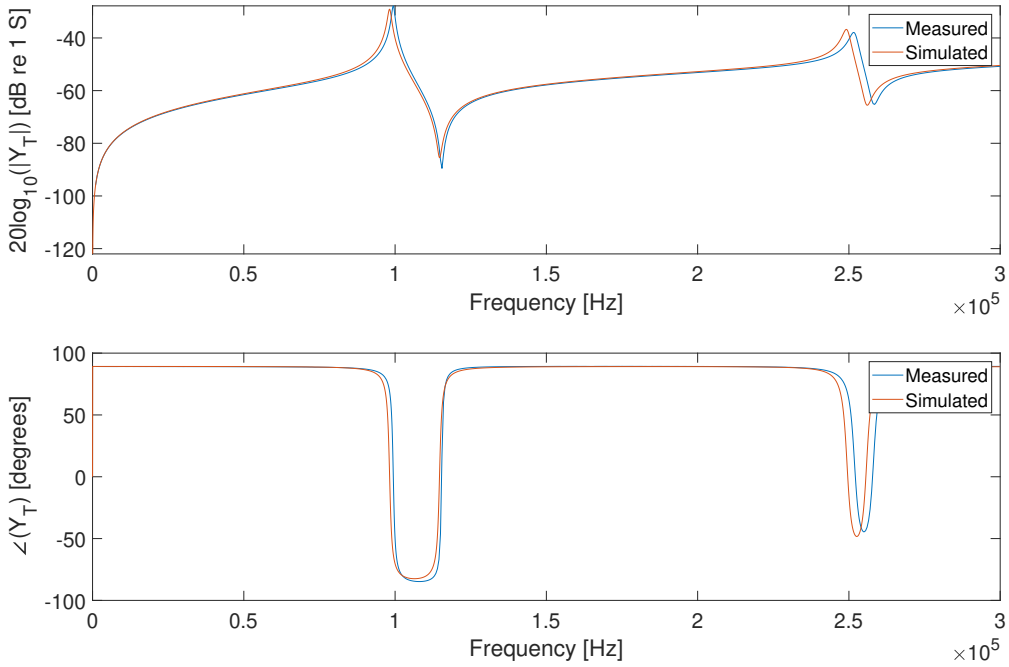


FIGURE 5.13: Measured and simulated admittance magnitude and phase of element 17.

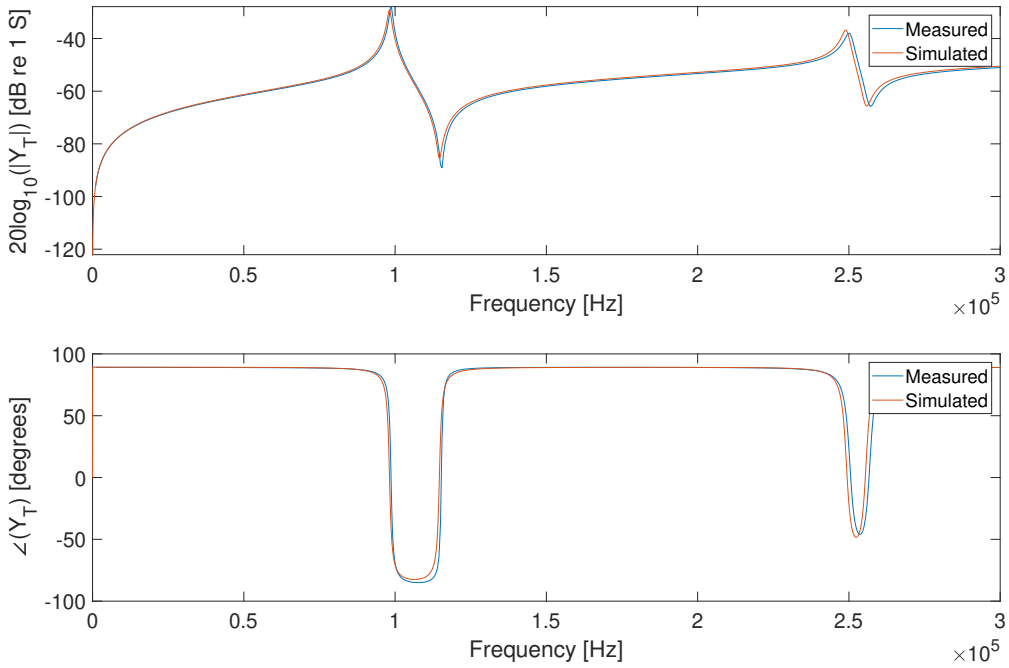


FIGURE 5.14: Measured and simulated admittance magnitude and phase of element 18.

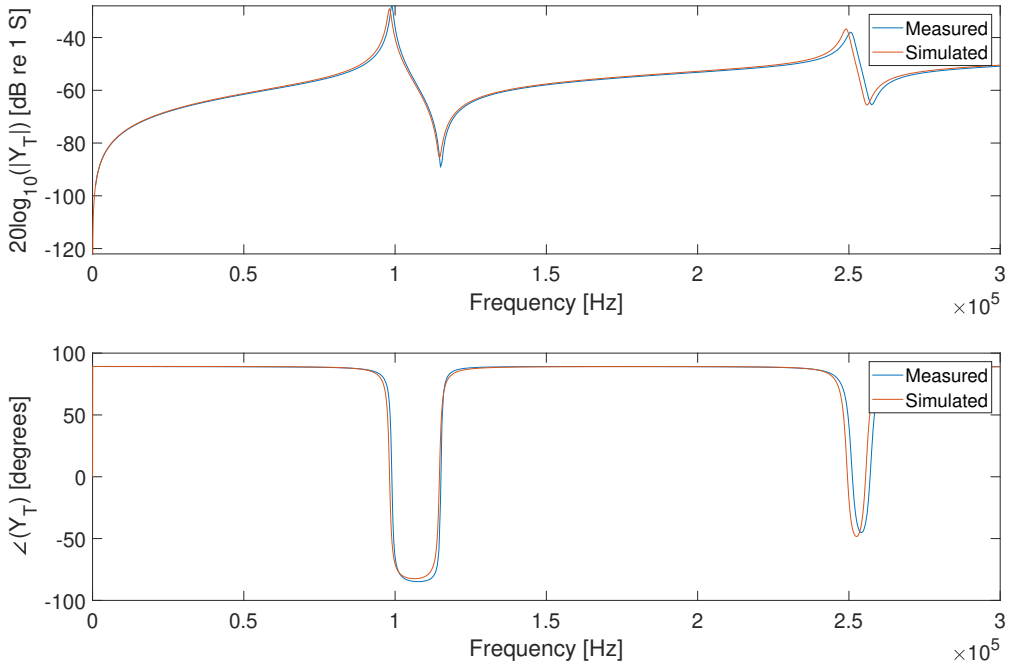


FIGURE 5.15: Measured and simulated admittance magnitude and phase of element 19.

TABLE 5.2: Difference in admittance peak amplitude and frequency at R1 and R2. Positive number means measured peak is higher.

Element Nr.	Magnitude difference at R1 and R2	Frequency difference at R1 and R2
1	1.2 dB and -1.4 dB	700 Hz and 1550 Hz
3	1.3 dB and -1.1 dB	650 Hz and 1400 Hz
8	1 dB and -1.1 dB	500 Hz and 950 Hz
9	1 dB and -1.1 dB	1150 Hz and 2200 Hz
17	1.3 dB and -1.2 dB	1250 Hz and 2450 Hz
18	1.3 dB and -1.2 dB	550 Hz and 1050 Hz
19	1.1 dB and -1.2 dB	750 Hz and 1500 Hz

5.2.4 Effect of soldering on element admittance

To mount the transmitting element in the measurement setup, two wires are soldered to the electrodes of the element. Due to the heat applied during soldering, and the extra weight of the solder on the element, some of the electrical characteristics may be affected. During the soldering process, the temperature was held way below the piezoelectric material curie temperature of 350 C, so as to not affect the polarization [44]. In Figs. 5.16 - 5.18, the measured admittance of element 8 is shown before and after soldering.

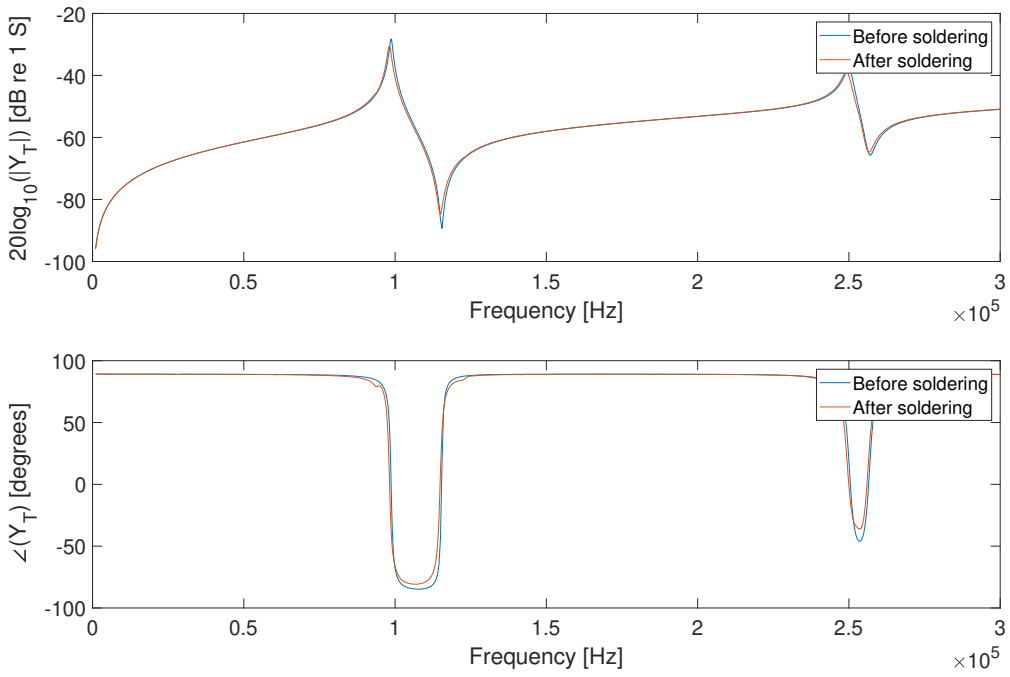


FIGURE 5.16: Measured admittance before and after soldering, from 1 kHz to 300 kHz.

The soldering has caused a general drop in the magnitude of the admittance, with a reduction of 2.25 dB after soldering at the first resonance R1, and a reduction of 2.1 dB at the second resonance R2. A downward shift in resonance frequency can also be seen, with a drop of 500 Hz at the first resonance R1, and a drop of 800 Hz at the second resonance R2. This may be caused by slight changes in the element material characteristics due to the temperature of the solder, or the extra weight if the solder impeding vibration.

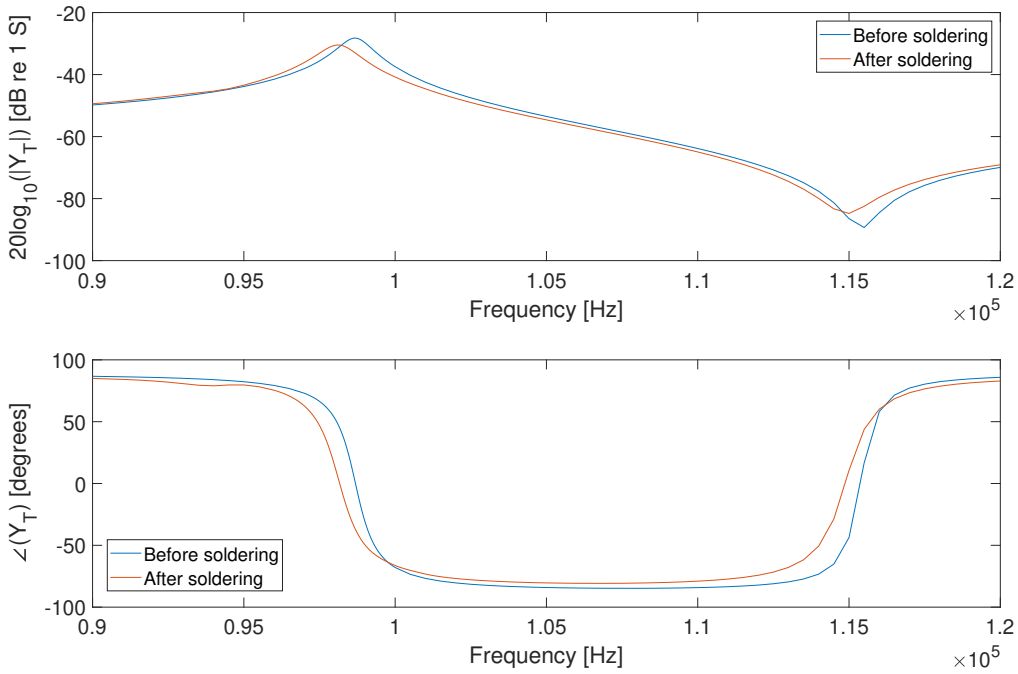


FIGURE 5.17: Measured admittance before and after soldering, from 1 kHz to 300 kHz.

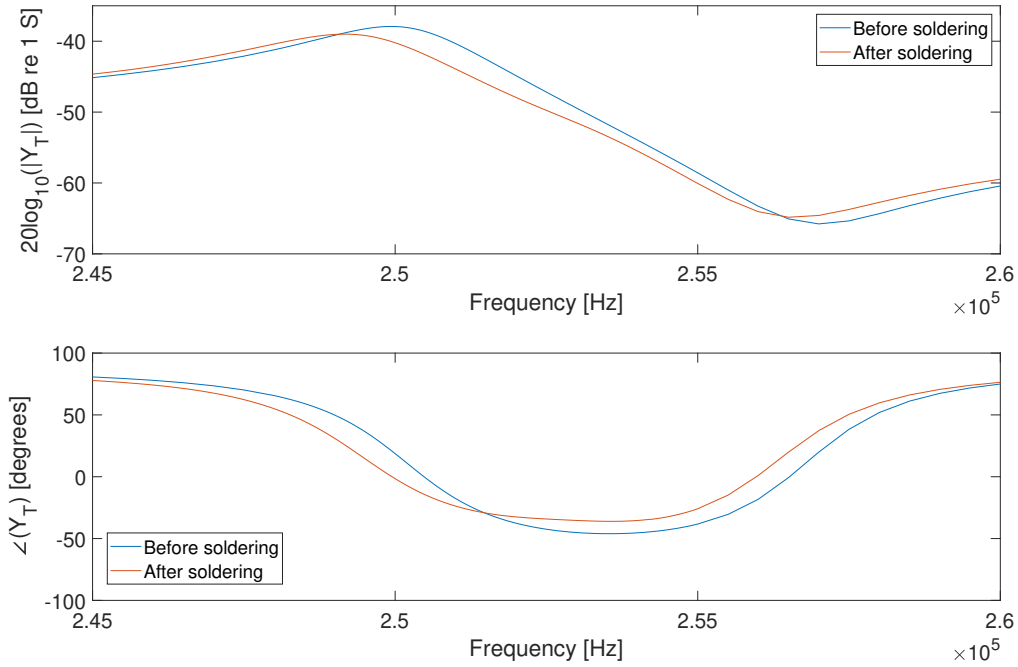


FIGURE 5.18: Measured admittance before and after soldering, from 1 kHz to 300 kHz.

TABLE 5.3: Resonance frequencies of the first two series resonance radial modes, for the simulated element, and the element used in measurement.

Mode	Simulation	Measurement	Δf
First radial mode	98160 Hz	98140 Hz	-20 Hz
Second radial mode	249230 Hz	249520 Hz	290 Hz

5.3 Choice of frequencies for further comparisons

In the next sections, measurements and simulations at different frequencies is compared. Because of the slight difference in resonance frequency between the simulated and measured element, the choice of what frequencies to compare is not as simple as picking the same frequency from both datasets. When looking at the sound pressure and directivity beam pattern produced by the element, it is more accurate to compare the measured and simulated elements at the frequencies where the radial modes are found. The standing wave pattern present in the measured and simulated element will then be more similar, even though the frequencies are similar. The frequencies used for comparison is found in Table 5.3. The resonance frequencies of the simulated and measured element can be found using different methods, but in this work, the series resonance of the elements is found by finding the resonance frequency peak of the conductance of the element. For the frequencies off the resonance peak, linear interpolation between the differences Δf at the radial modes is used to find the frequency difference for each measurement frequency presented. The difference between the radial modes, Δf, is defined as

$$\Delta f = f_m - f_s \tag{5.4}$$

where f_m is the measured frequency at the first and second series resonance modes, and f_s is the simulated frequency at the first and second series resonance modes. These values are used to find the line equation to define Δf at all frequencies. This assumes that the behaviour of the standing waves in the element changes linearly as the frequency changes.

The equation for the frequency difference and corresponding simulation frequency is then calculated using the values in Table 5.3.

$$\Delta f(f_m) = \frac{31}{15138} f_m - \frac{1672550}{7569} \tag{5.5}$$

$$f_s = f_m - \Delta f \tag{5.6}$$

where Δf is the difference between the measurement frequency, and the corresponding simulation frequency, f_m is the measurement frequency used in the comparison, and f_s is the simulation frequency to be compared to f_m. The frequencies used in the comparisons of directivity can be found in Table 5.5, while the frequencies used in the comparisons of on-axis pressure can be found in Table 5.4.

5.4 On-axis pressure

The on axis pressure is measured from 0.1 cm to 40 cm, over a range of frequencies from 60 to 180 kHz. The resolution in the z-axis varies as the distance increases, with lower resolution in the far field. Exact measurement resolution can be found in Table 5.6. The height of the microphone is adjusted for each measurement, to ensure the recording surface is placed at the main lobe of the element. The measured pressure values are corrected to be lossless, so

TABLE 5.4: Calculated equivalent frequencies for directivity comparisons. Simulation frequency rounded to nearest whole Hz.

Simulation	Measurement	Δf
60098 Hz	60000 Hz	-98 Hz
98160 Hz	98140 Hz	-20 Hz
139934 Hz	140000 Hz	66 Hz
179852 Hz	180000 Hz	148 Hz
219770 Hz	220000 Hz	230 Hz
249230 Hz	249520 Hz	290 Hz

TABLE 5.5: Calculated equivalent frequencies for on-axis pressure comparisons. Simulation frequency rounded to nearest whole Hz.

Simulation	Measurement	Δf
60098 Hz	60000 Hz	-98 Hz
80057 Hz	80000 Hz	-57 Hz
98160 Hz	98140 Hz	-20 Hz
119975 Hz	120000 Hz	25 Hz
139934 Hz	140000 Hz	66 Hz
159893 Hz	160000 Hz	107 Hz
179852 Hz	180000 Hz	148 Hz

it matches the lossless simulations. The measured and simulated on-axis pressure is shown in Figs. 5.19 - 5.25

TABLE 5.6: Measurement z-axial resolution at different measurement intervals.

Measurement interval	Spacial resolution
0.1 - 1 cm	0.1 cm
1 - 2 cm	0.2 cm
2 - 5 cm	0.5 cm
5 - 10 cm	1 cm
10 - 40 cm	2 cm

In the near field, a discrepancy in the pressure amplitude can be seen across all frequencies. This is most likely caused by the combined effects of standing waves between transmitter and receiver, and the fact that the waves in the near-field are not plane, so the pressure over the microphone receiver surface is not constant. This could possibly be corrected for by diffraction correction, but it has been difficult to find a method to calculate the correction value for a microphone with unknown receiver radius. This is easier when using the same size piezoelectric element as transmitter and receiver, but this is not done in this work.

Another factor that can cause the discrepancies between the measured and simulated data, is the calibrated values for the receiver sensitivity. The method for calculating this value is shown in Section 3.6. To calculate the pressure using the recorded signal voltage,

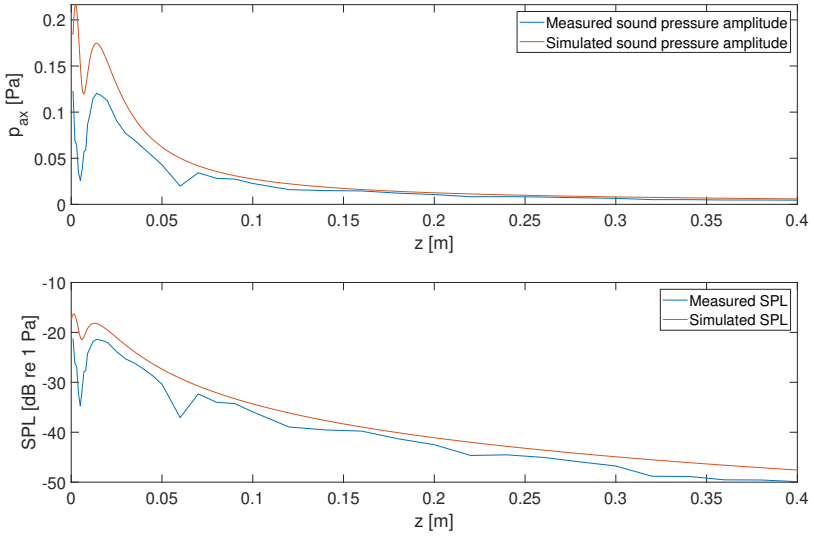


FIGURE 5.19: Simulated (60098 Hz) and measured (60000 Hz) on-axis pressure from 0.1 to 40 cm, pressure amplitude and SPL.

a value for the receiver sensitivity $M_v(f)$ for all frequencies need to be found. The documentation supplied with the microphone only gives reliable correction values from 0 to 100 kHz. Above 100 kHz, the correction values applied to the calibrated $M_V(251.2Hz)$ can only be found in a graph, shown in Fig. 3.22. This graph has been manually digitized, so the values have a high uncertainty, because of the low resolution of the graph, and the possibility of human error when mapping the graph.

The measured on-axis pressure amplitude is in general lower than the simulated on-axis pressure amplitude for the lower frequencies, but when the frequency moves above 140 kHz, the opposite is observed, with the measurements having higher amplitudes than the simulations. The largest difference between measurements and simulations can be seen at the R1 series resonance on-axis pressure, where the difference reaches a value of around 8 dB at the furthest distance measured, 0.4 m.

All the on-axis graphs, except the resonance on-axis pressure, show a dip in the pressure at 6 cm, see Figs. 5.19 - 5.25. This is most likely due to a positioning error during measurement. As the pressure was measured across all non-resonance frequencies in one batch, the misplacement of the element could cause the error seen at the same point for all frequencies, except for the resonance measurements, which were performed separately. 6 cm is beyond the point of the last maximum in the near-field, which falls at around 3 cm, so the behaviour of the on-axis pressure should approach a 1/r-dependency, meaning it should drop in amplitude by the inverse of the separation distance. This can be seen at all frequencies, if the value at 6 cm is ignored.

The SNR can also cause uncertainties with the measured voltages. The calculated SNR for all measurement frequencies is shown in Fig. 5.26. The SNR here stays above 20 dB for all except the measurements at 140 kHz. Since the measurements are performed within the main lobe of the transmitting element beam pattern, the signal strength is expected to be high enough to give good measurements, at the separation distances used in this work.

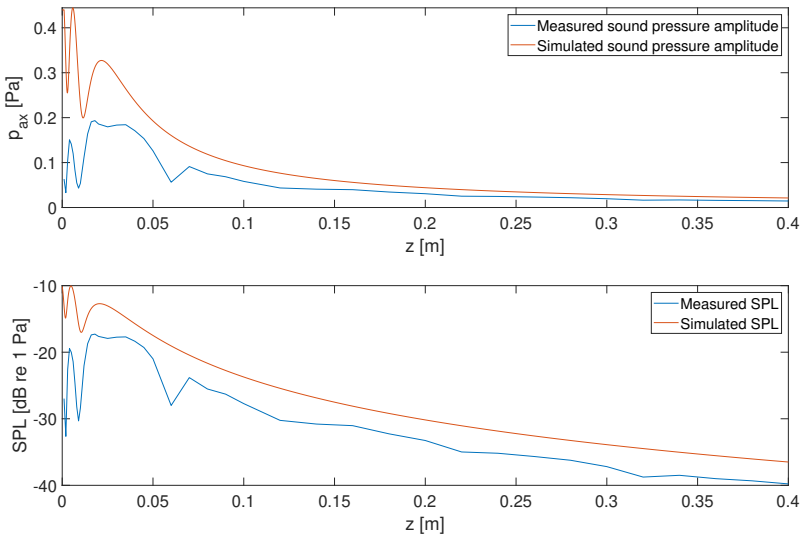


FIGURE 5.20: Simulated (80057 Hz) and measured (80000 Hz) on-axis pressure from 0.1 to 40 cm, pressure amplitude and SPL.

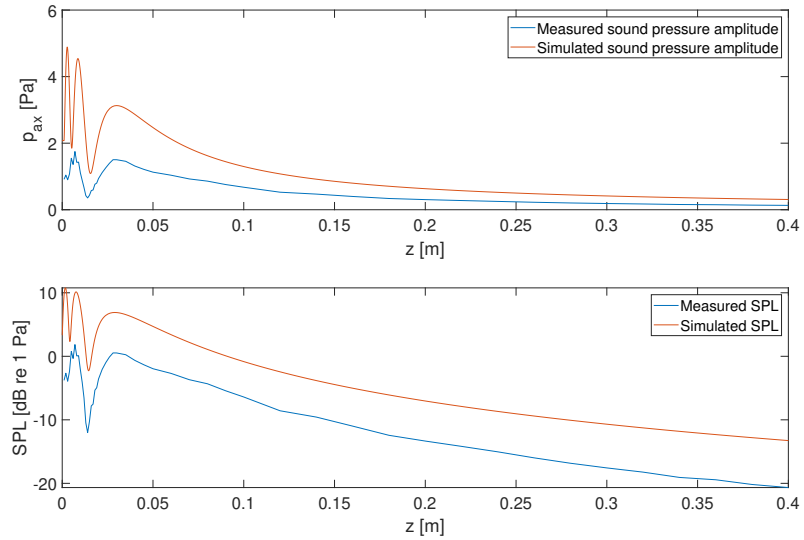


FIGURE 5.21: Simulated (98160 Hz) and measured (98140 Hz) on-axis pressure from 0.1 to 40 cm, pressure amplitude and SPL.

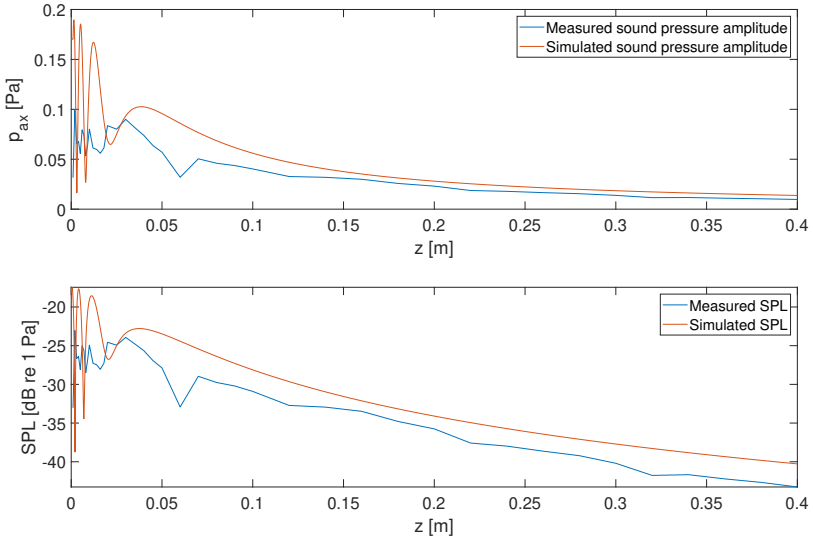
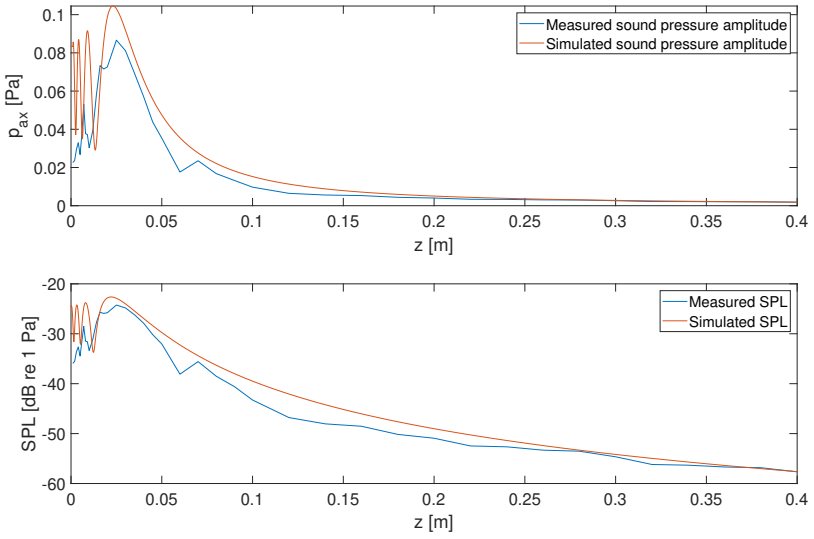


FIGURE 5.22: Simulated (119975 Hz) and measured (120000 Hz) on-axis pressure from 0.1 to 40 cm, pressure amplitude and SPL.



[h]

FIGURE 5.23: Simulated (139934 Hz) and measured (140000 Hz) on-axis pressure from 0.1 to 40 cm, pressure amplitude and SPL.

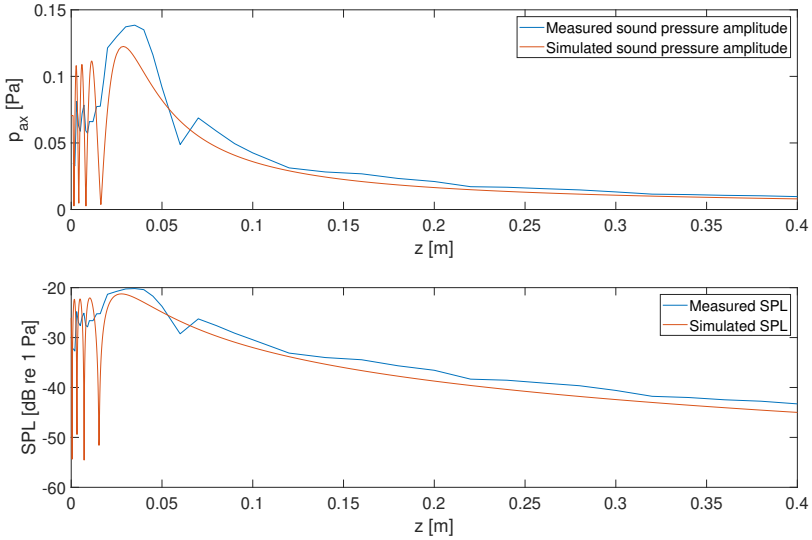


FIGURE 5.24: Simulated (159893 Hz) and measured (160000 Hz) on-axis pressure from 0.1 to 40 cm, pressure amplitude and SPL.

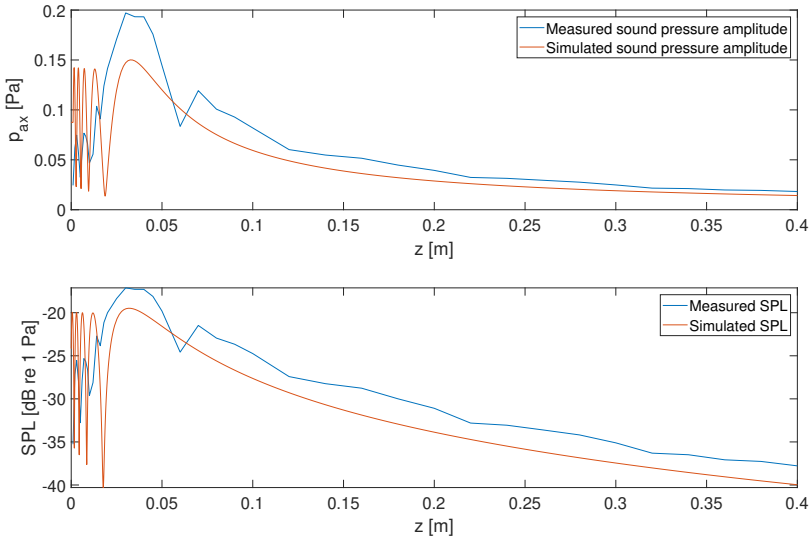


FIGURE 5.25: Simulated (179852 Hz) and measured (180000 Hz) on-axis pressure from 0.1 to 40 cm, pressure amplitude and SPL.

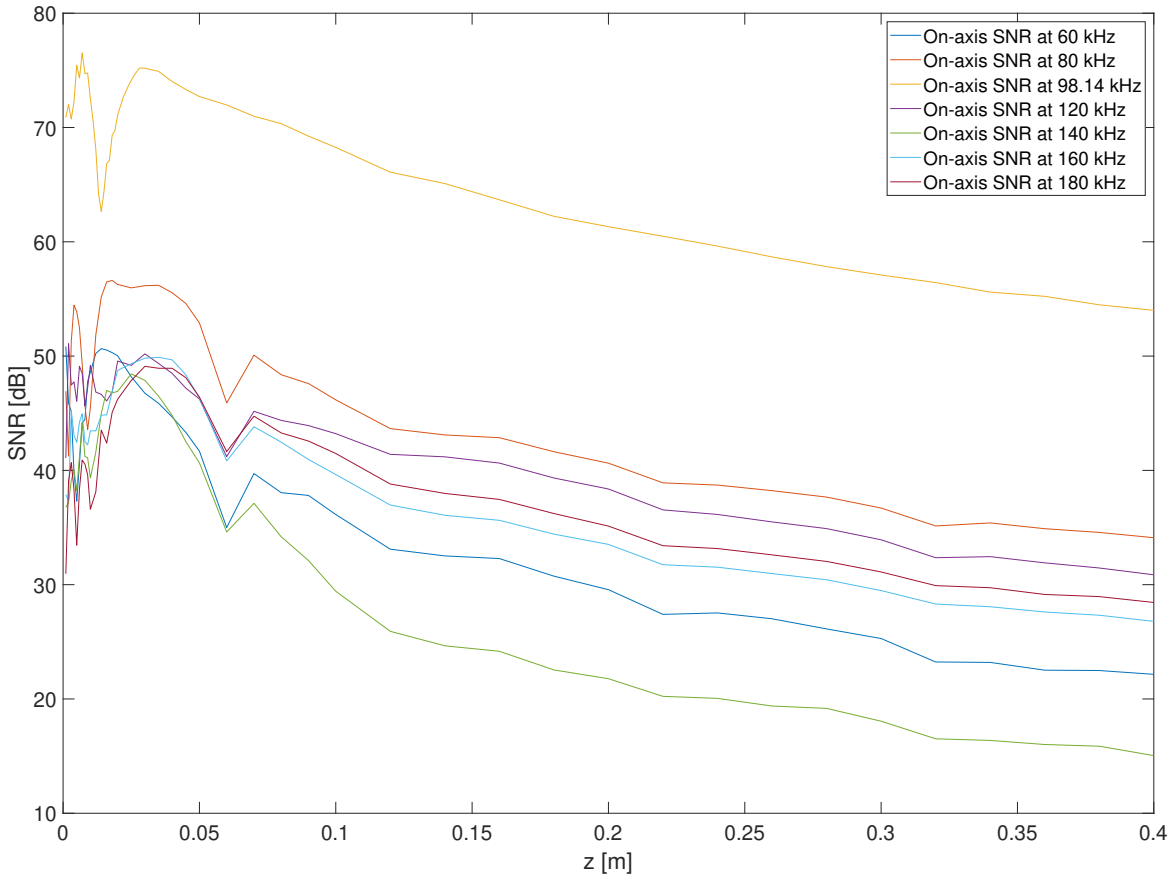


FIGURE 5.26: Calculated SNR for on-axis measurements, at all measurement frequencies.

5.5 Transmitter sensitivity

The transmitter sensitivity describes the relation between the input voltage of the transmitter, and the free-field on axis pressure produced [12]. The equation for the transmitter sensitivity is given as [14]

$$S_V = \frac{p(z = d_0, \theta = 0)}{V_1} \tag{5.7}$$

where S_V is the transmitter sensitivity in Pa/V, p is the free-field on axis pressure amplitude at a reference point d_0 in the far field, and V_1 is the input voltage at the transmitter. Usually, the reference distance d_0 is set to 1 meter, to make sure the point is well into the far field, so the incoming pressure wave can be assumed to be flat.

In this work, the measurements of the sound pressure need to be done at a distance so that the point is well into the far field, while still being close enough so that the received signals has a good signal-to-noise ratio. Normally, the transmitter sensitivity is given as a function of pressure amplitude at 1 meter from the transmitter, but in the setup used in this work, the maximum separation possible is around 80 cm, so measurements at 1 m is impossible. To achieve a good balance between distance and signal strength, a distance of 0.5 m is chosen.

The measurements are done over a frequency spectrum from 50 to 180 kHz, once with an input voltage V_1 equal to 1 V, and once with V_1 equal to 10 V, to observe the changes in S_V caused by non-linear effects in the element at higher voltages, see Fig. 5.27.

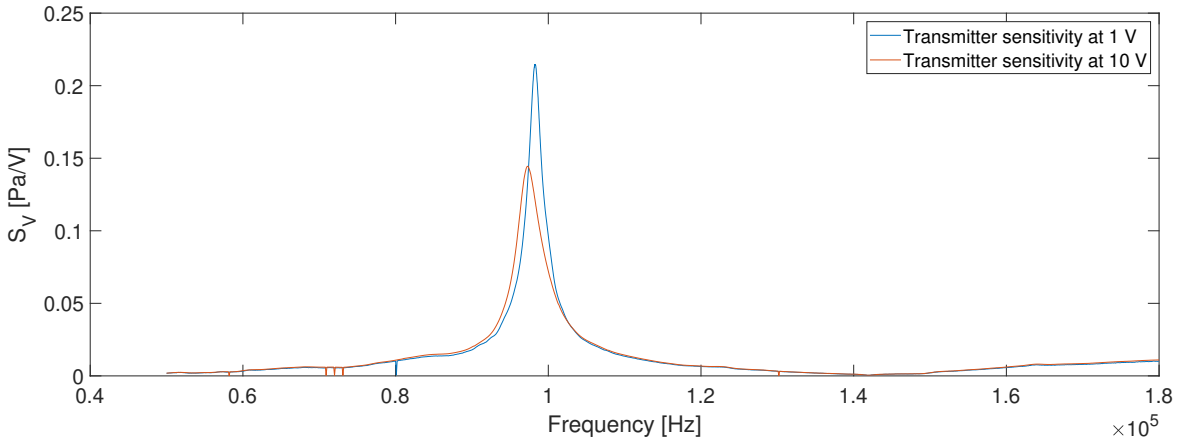
The calculated S_V at $V_1 = 10\ V$ show a reduction at the maximum of 3.44 dB, and a frequency shift of the maximum amplitude from 98.14 kHz to 97.3 kHz.

5.5.1 Comparison with simulations

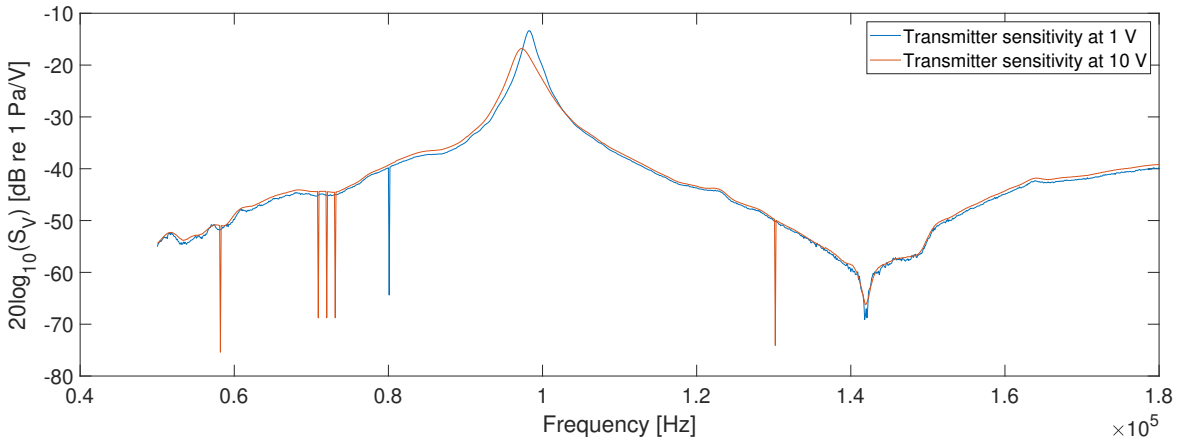
Here, the measured and simulated transmitter sensitivity is presented. As the measured receiver sensitivity is calculated using the same method as the on-axis pressure in Section. 5.4, the same discrepancy between measured and simulated pressure amplitude is expected to be observed. As can be seen in Fig. 5.28, the simulated transmitter sensitivity S_V is significantly higher than the measured S_V. The largest deviation is at and around the first resonance frequency R1, and is around 8 dB. This is close to the discrepancy seen in the on-axis pressure at resonance (see Fig. 5.21). It can also be seen, in Fig. 5.28, that the simulated sensitivity has an uneven amplitude in the low amplitude areas from around 110 kHz and up. This is not observed in any other simulations, and the source of the problem could not be identified during this work.

5.5.2 Measured V_{1m} voltage of the frequency spectrum

The oscilloscope measures the voltage sent to the transmitting element, and the voltage amplitude is dependant on the frequency of the signal, as it affects the resistance of the element "seen" by the signal generator. When the measured V_{1m} is plotted over frequency, some odd behaviour emerges. As can be seen in Fig. 5.29, the peak-to-peak voltage stays around 2 V at non-resonance frequencies, and drop to around 1 V at the resonance, as expected, but in addition, there is a regular small oscillation everywhere off the resonance dip. The voltage transmitted to the element goes up and down by about 0.006 volts, every 25 Hz or so. When the peak-to-peak voltage of the signal is increased to 10 V, a similar pattern can be seen, see Fig. 5.30. There is also a smaller regular oscillation within each period of around 25 Hz. During this work, an explanation of this behaviour could not be found.



(A)



(B)

FIGURE 5.27: Linear (A) and logarithmic (B) transmitter sensitivity, calculated at a distance of 0.5 m. Spikes due to measurement error, where no signal was received.

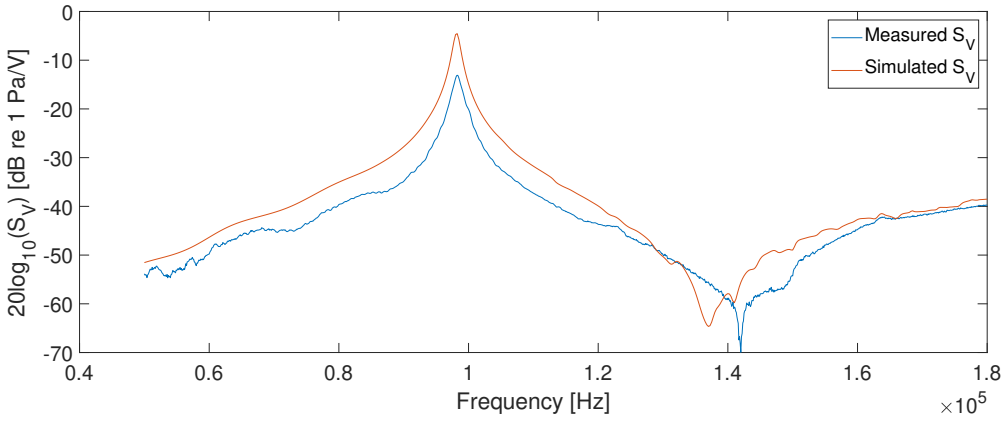


FIGURE 5.28: Measured S_V using input voltage V_0 = 1 V, and simulated sensitivity, both at 0.5 m. Measurement mistakes removed.

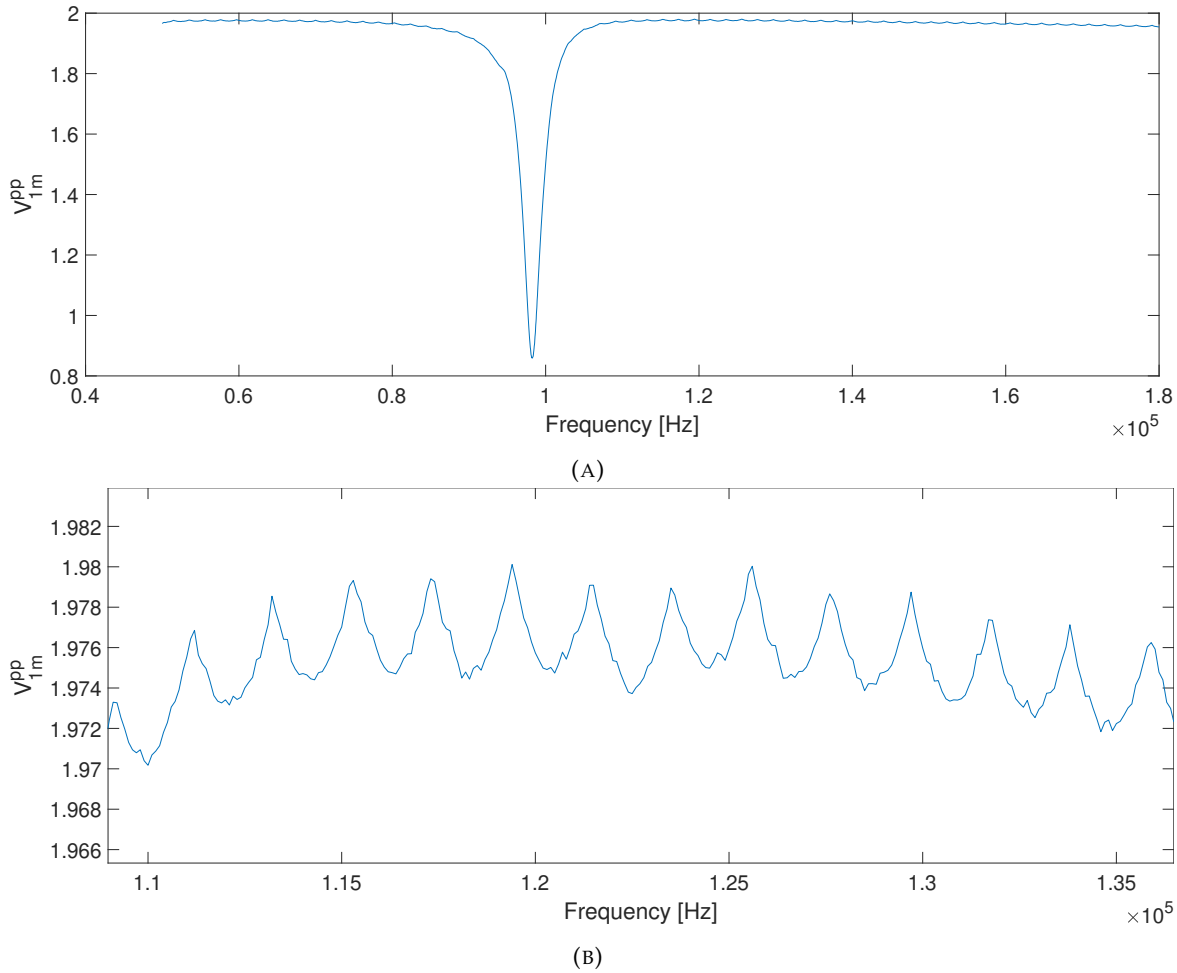
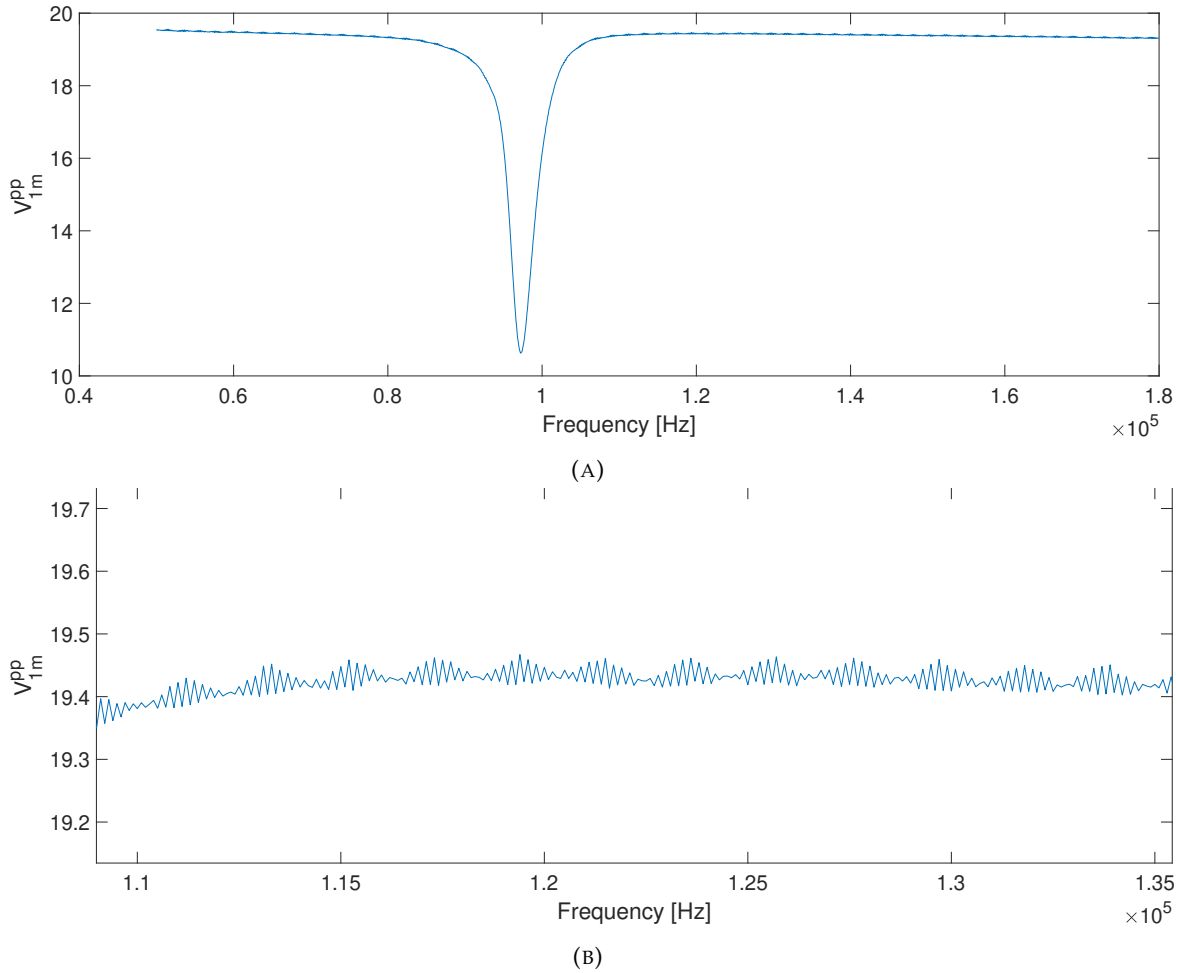


FIGURE 5.29: V_{1m} measured over the frequency span, with a V_0 of 1 V. (A) shows full frequency spectrum, while (B) shows a zoomed in section of the non-resonance area.



(A)

(B)

FIGURE 5.30: V_{1m} measured over the frequency span, with a V_0 of 10 V. (A) shows full frequency spectrum, while (B) shows a zoomed in section of the non-resonance area.

5.6 Directivity

In this section, the measurements of the directivity beam pattern of the element is presented.

5.6.1 Directivity measurement pulses

In this section, some example pluses from the different directivity measurements will be shown, see Figs. 5.31 - 5.36. This is to show the variation in quality of the pulses as the element rotates, and demonstrate that the uncertainty of the measured V_{pp} increases in the dips of the beam pattern. The figures showing the pulses recorded at -40 degrees generally shows a change in V_{pp} at around 0.85 ms. This has been identified to be caused by electrical interference from the transmitting element magnetic field, mentioned in Section 3.3. This electrical interference cause either constructive or destructive interference, increasing or decreasing the signal amplitude. To eliminate this disturbance from the processing, the FFT area is chosen as the interval from 1 to 1.4 ms, when using a pulse length of 1.4 ms, marked in Figs. 5.31 - 5.36 as blue stippled lines. After the effect of the interference was discovered, the pulse length was changed to 0.6 ms, where the FFT calculation interval is set from 0.3 ms to 0.5 ms, see for example Fig. 5.32. More examples of pulses are presented in Appendix B.

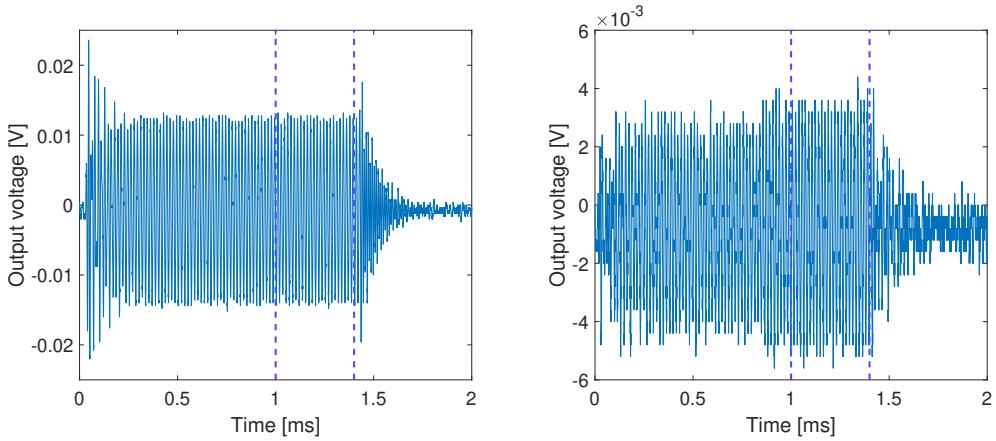


FIGURE 5.31: Recorded pulses at 0 degrees (left), and -40 degrees (right), at 60 kHz, pulse length of 1.4 ms. Blue lines mark FFT calculation interval.

5.6.2 Comparison between simulations and measurements

The measured and simulated directivities are shown in Figs. 5.37 - 5.42. Curve fitting is used on the measurement datasets. For the simulated frequencies, the standard method in FEMP is to simulate the directivity at a distance guaranteed to be in the far field [24]. For regular simulations, a distance of 1000 m can be chosen, and FEMP automatically interpolates the pressure values back to 1 m. In this work, the measurements of the directivity beam pattern were done primarily at 20 cm. The sound pressure decreases as the distance from the element increases, so measurements of directivity in the distant far field becomes difficult. This effect is especially prevalent when measuring sound pressure with a non-resonance frequency, and measuring at an angle that places the microphone in a dip in the element's beam pattern. 20 cm is therefore chosen to be close enough so that the signal-to-noise ratio is still good, while far enough away from the element to be in the far field. There is still some

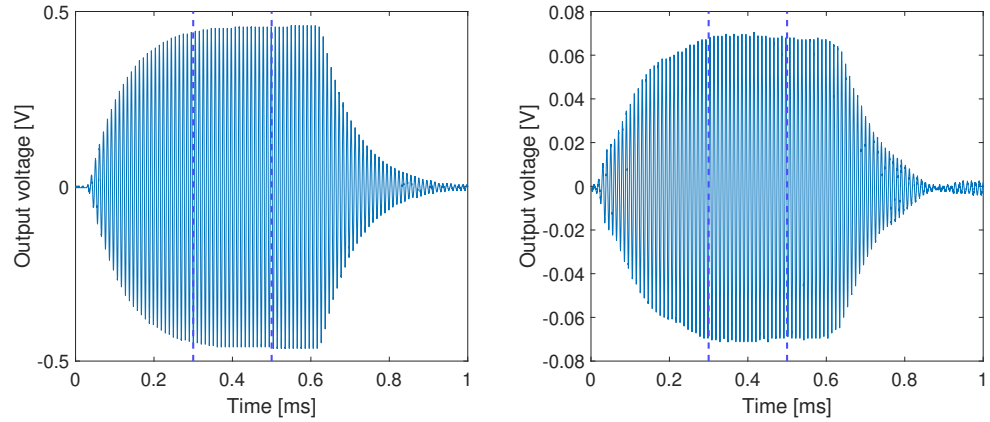


FIGURE 5.32: Recorded pulses at 0 degrees (left), and -40 degrees (right), at 98.14 kHz, pulse length of 0.6 ms. Blue lines mark FFT calculation interval.

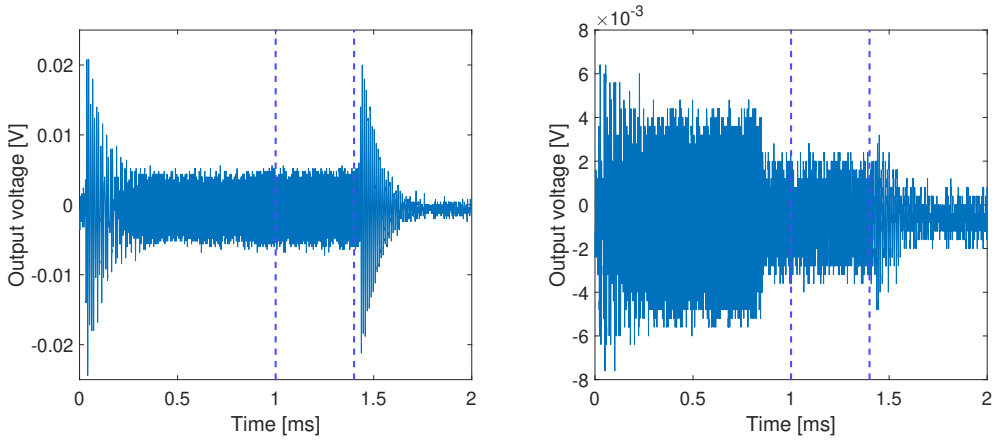


FIGURE 5.33: Recorded pulses at 0 degrees (left), and -40 degrees (right), at 140 kHz, pulse length of 1.4 ms. Blue lines mark FFT calculation interval.

possibility of near field effects, so to get a good comparison between the measurements and simulations, the directivity is also simulated at 20 cm.

The main lobes in the directivity beam patterns shows a good comparison between the measurements and simulations, but off the main lobes most of the comparison plots show discrepancies in the lobe amplitudes. In Fig. 5.40 - 5.42, it can be seen that the first side lobes in the measured data generally has a greater amplitude than the simulations. The measured directivities are also not symmetrical, as may be expected from a beam pattern produced by a symmetrical piezoelectric element.

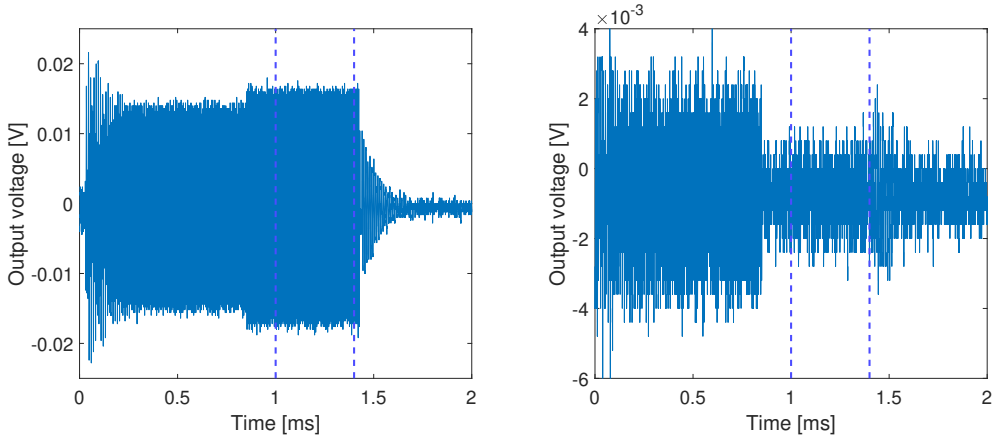


FIGURE 5.34: Recorded pulses at 0 degrees (left), and -40 degrees (right), at 180 kHz, pulse length of 1.4 ms. Blue lines mark FFT calculation interval.

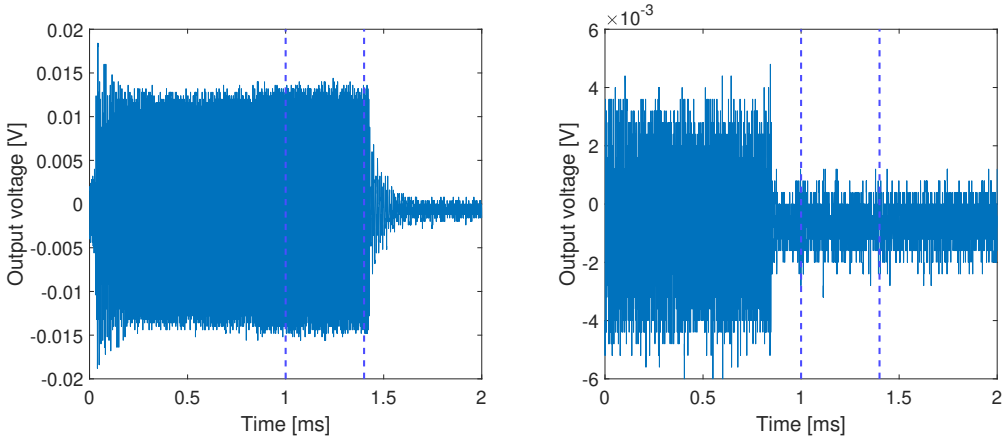


FIGURE 5.35: Recorded pulses at 0 degrees (left), and -40 degrees (right), at 220 kHz, pulse length of 1.4 ms. Blue lines mark FFT calculation interval.

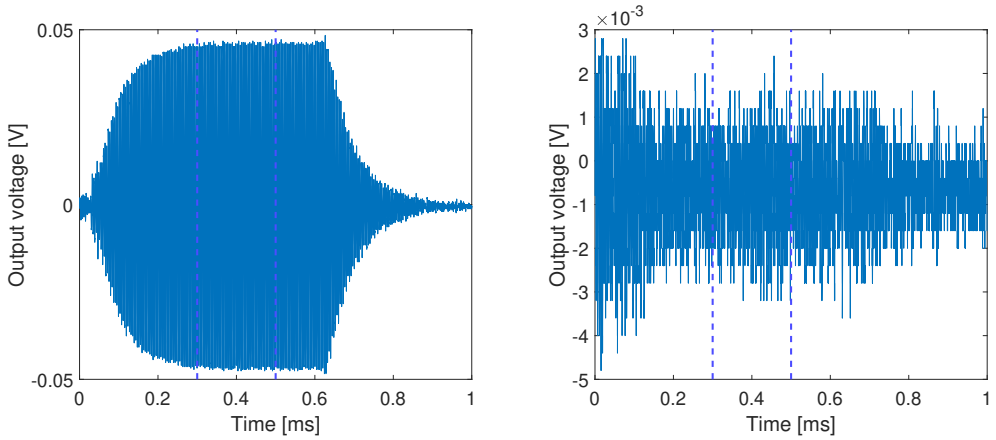


FIGURE 5.36: Recorded pulses at 0 degrees (left), and -40 degrees (right), at 249.52 kHz, pulse length of 0.6 ms. Blue lines mark FFT calculation interval.

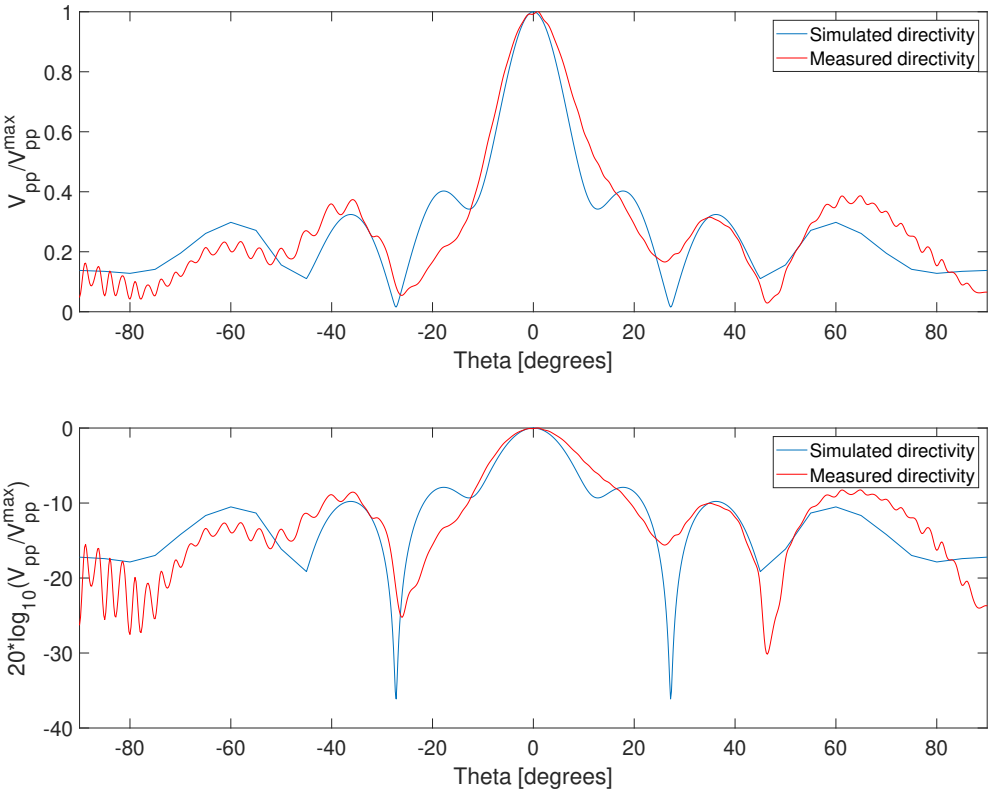


FIGURE 5.37: Linear and logarithmic normalized directivity measured at 60 kHz and 20 cm, and simulated at 60098 Hz.

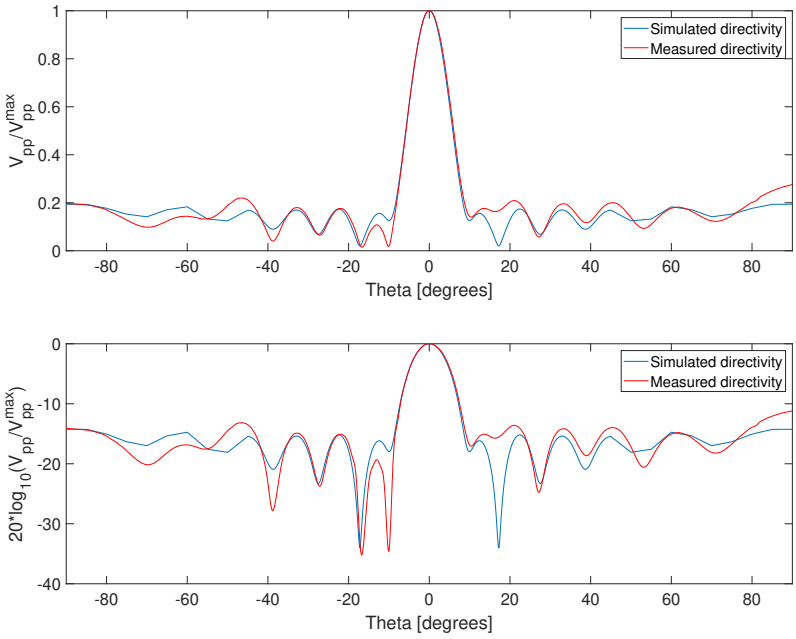


FIGURE 5.38: Linear and logarithmic normalized directivity at measured at 98.14 kHz and 20 cm, and simulated at 98160 Hz.

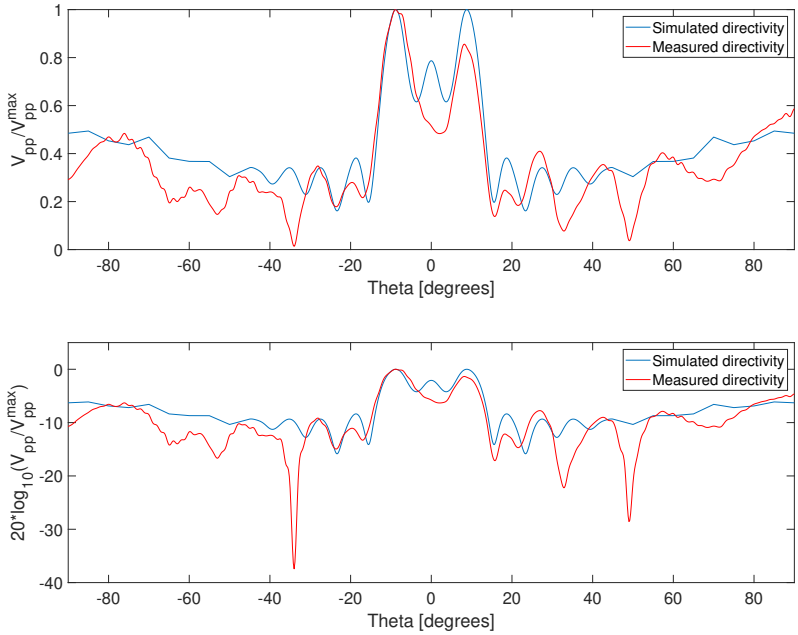


FIGURE 5.39: Linear and logarithmic normalized directivity measured at 140 kHz and 20 cm, and simulated at 139934 Hz.

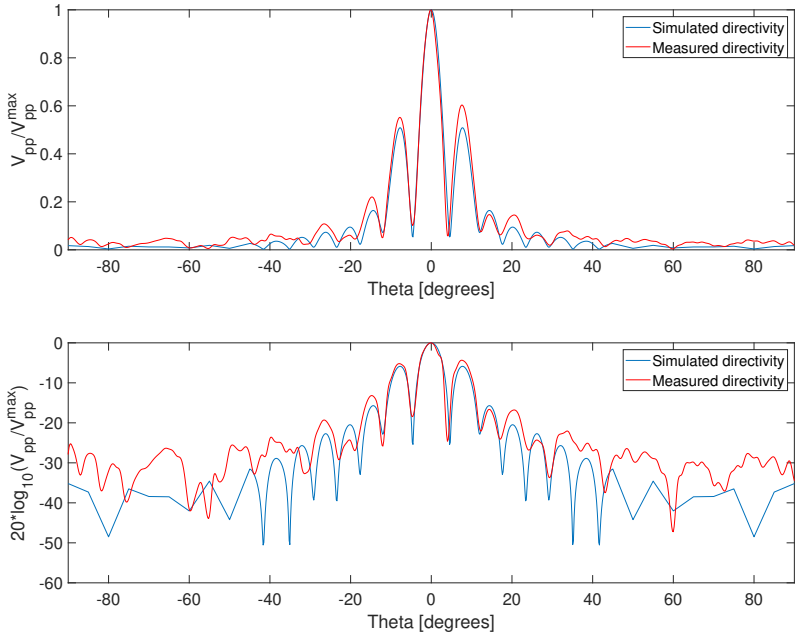


FIGURE 5.40: Linear and logarithmic normalized directivity measured at 180 kHz and 20 cm, and simulated at 179852 Hz.

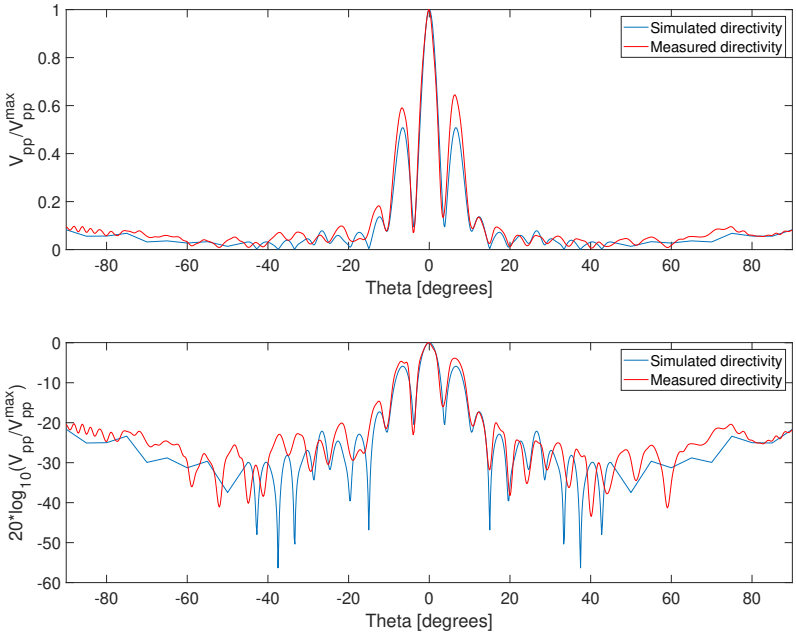


FIGURE 5.41: Linear and logarithmic normalized directivity measured at 220 kHz and 20 cm, and simulated at 219770 Hz.

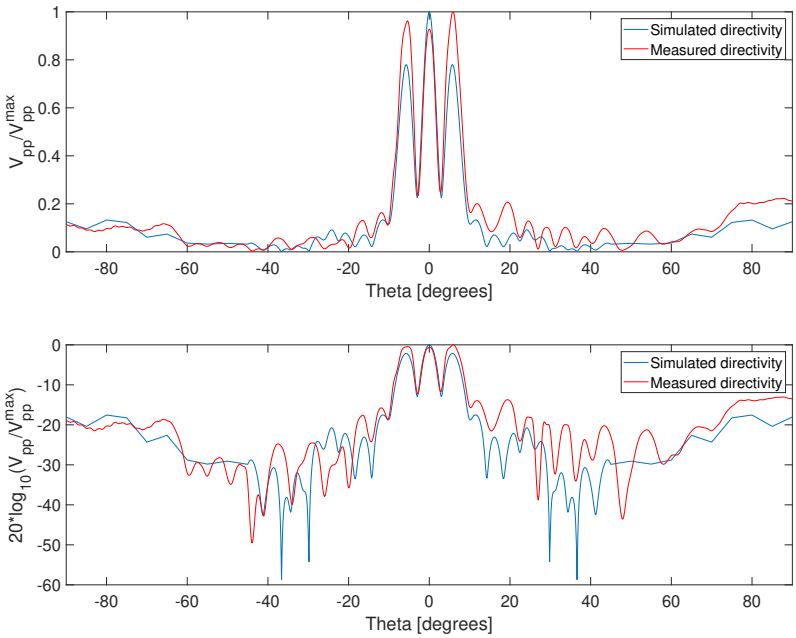


FIGURE 5.42: Linear and logarithmic normalized directivity measured at 249.52 kHz and 20 cm, and simulated at 249230 Hz

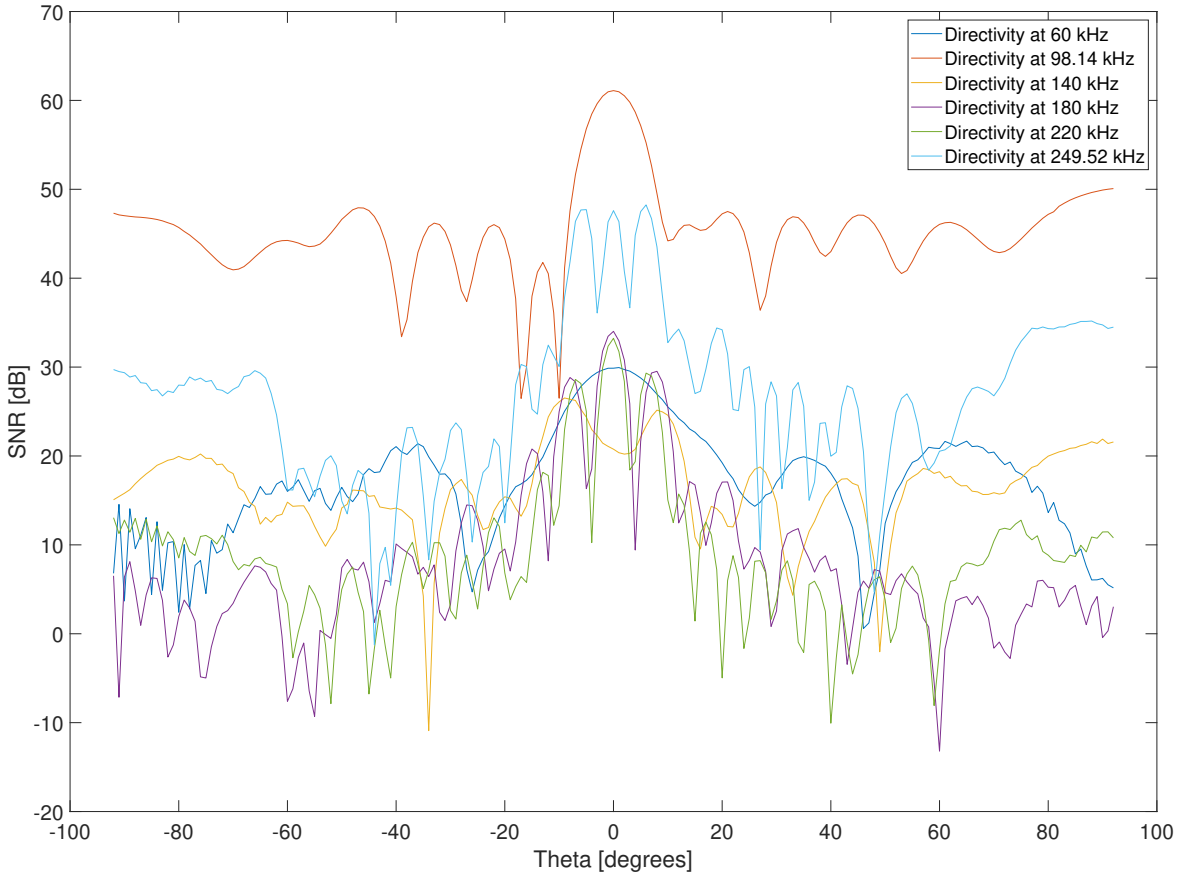


FIGURE 5.43: SNR of the measured directivity at all frequencies used.

The SNR of all the measured directivities are displayed in Fig. 5.43. Overall, the measurements span from a high SNR of 58 dB, to a low SNR of -11 dB. The SNR around the main lobes never drop below 20 dB, so uncertainty due to noise disturbances can be ignored there. For the side lobes, the uncertainty is larger. Only the directivity at the first radial mode, 98.14 kHz, stays consistently above SNR = 20 dB. The other measurements fall below that consistently, with the measurements at 180 and 220 kHz fluctuating around a SNR of 0 dB outside the main and first side lobes. This goes far in explaining the discrepancies between measured and simulated directivity that can be observed in Figs. 5.40 and 5.41.

5.6.3 Separation distance effect on beam pattern

To get the most accurate directivity beam pattern, the measurements needs to be performed in the distant far field. This is difficult to achieve in an experimental setup, where the signal strength is limited by the properties of the transmitter. In this work, the signal applied to the transmitting element is set to a value of $V_0^{pp} = 1$ V, to limit non-linear effects that can affect the transmitted signal. To achieve a good SNR, the directivity measurements are done at a separation of 20 cm. This separation technically puts the receiving microphone in the far field, but some differences in the beam pattern is still observed as the separation of the transmitter and receiver is increased. Fig. 5.44 show the recorded directivity at three different separations, 20, 50 and 80 cm. The general shape is similar across all separations, but the side lobes show greater convergence as the separation increases. The measurements takes at 50 and 80 cm separation is much more similar to each other than to the 20 cm measurement, supporting the notion that the further you move out into the far field, the more accurate the beam pattern measurements will be. Note that the disturbances seen in the 50 cm measurement, especially between $\pm$ 40 degrees and $\pm$ 90 degrees is due to a mistake in the recording setup, causing reflections from the mounting rod (see Section 3.5.2) to interfere with the measured pulses. Due to time constraints, the measurement could not be rerun.

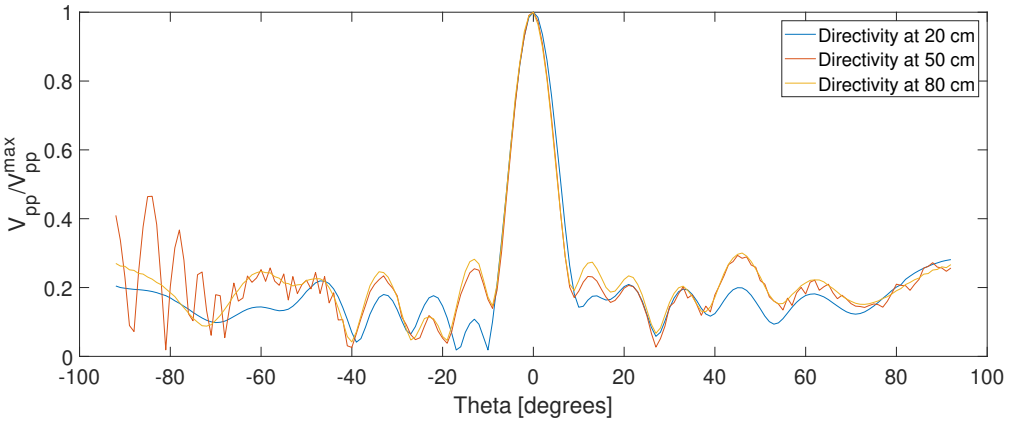


FIGURE 5.44: Measured directivity at 98.14 kHz, at 20, 50, and 80 cm separation of the transmitter and receiver.

The SNR of the measurements shown in Fig. 5.44 is presented in Fig. 5.45. As expected, the SNR is much higher with a lower separation. The SNR at 20 cm and 80 cm separation show a difference of 16 dB at the zero-degree point.

The difference in beam pattern due to separation distance can also be seen in the simulations. To get the most accurate comparisons, the same measurement distance is chosen for both measurements and simulations, although a much larger distance would normally be used when simulating the directivity beam pattern. An example of this discrepancy is shown in Fig. 5.46. The simulated directivity at 20 cm show a decrease in amplitude at the first side lobes, and a slight widening of the main lobe. The side lobe effect can also be seen in the measured directivity in Fig. 5.44.

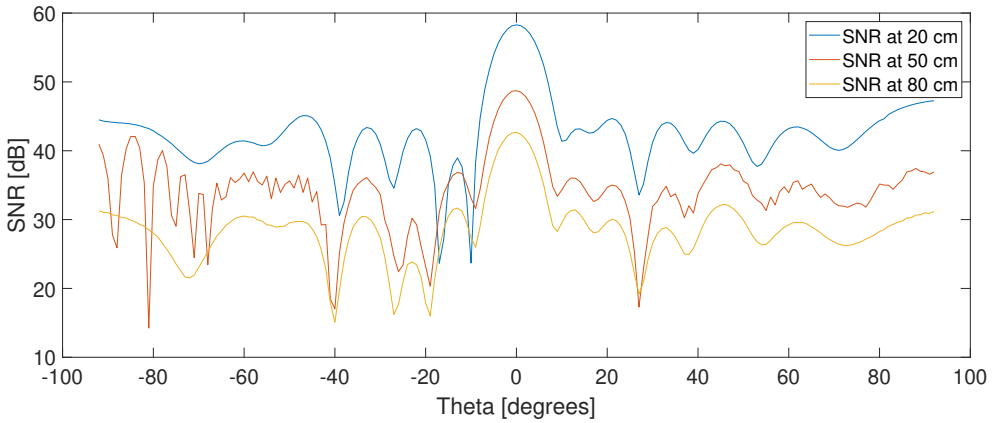


FIGURE 5.45: SNR of directivity measurements at 98.14 kHz, at 20, 50, and 80 cm separation of the transmitter and receiver.

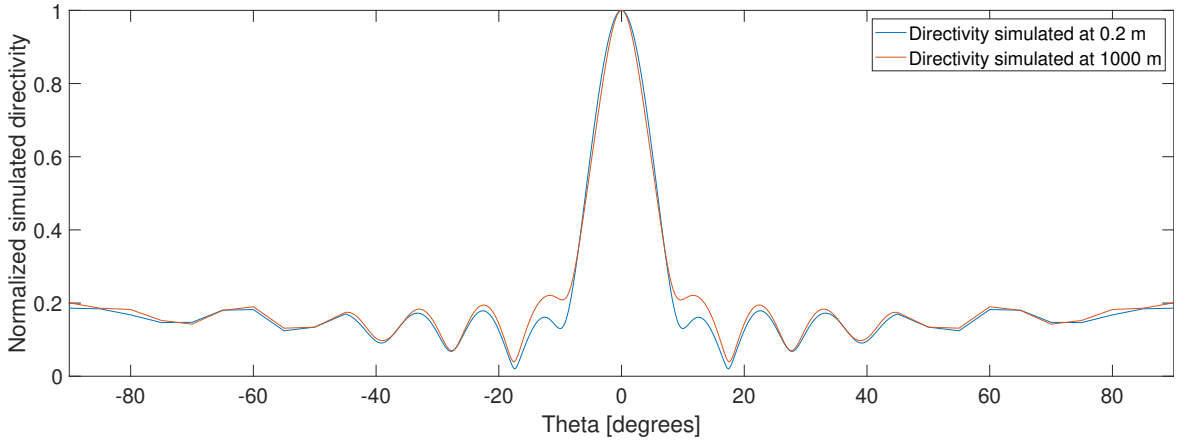
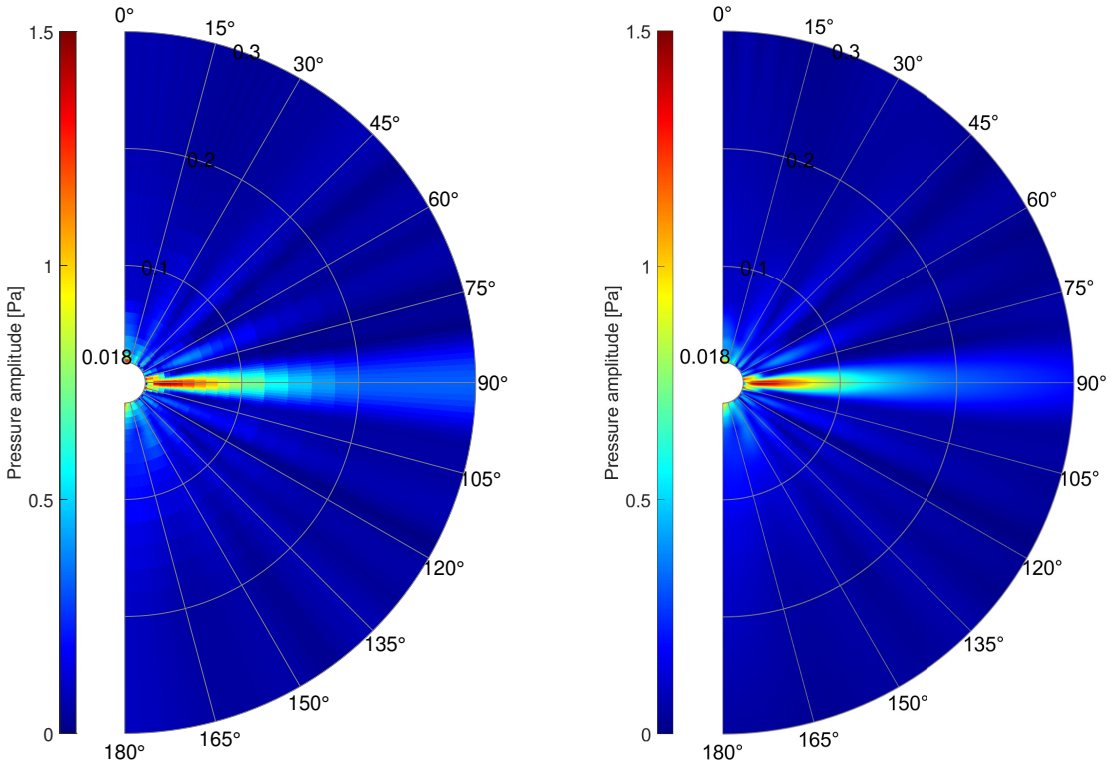


FIGURE 5.46: Simulated directivity at 98160 Hz, at 0.2 m and 1000 m.



(A) (B)

FIGURE 5.47: Measured 2D sound pressure field, from 0.0175 m out to 0.3 m, at 98.14 kHz (R1). Actual pressure values (A) and interpolated pressure values (B).

5.7 2D sound pressure field

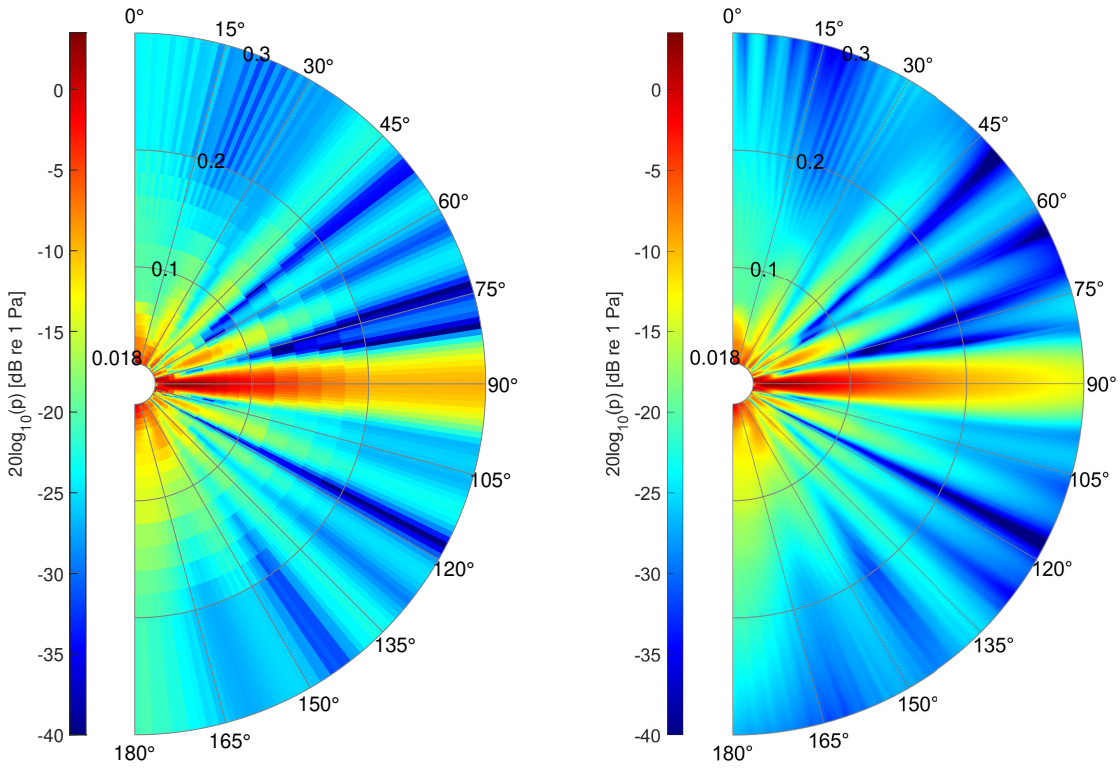
5.7.1 Measurement comparison

Here, the measured and simulated 2D near and far field pressure is presented and compared.

Note that FEMP simulates pressure values all the way in to the element surface. To do measurements in this close to the element is difficult, so the current measured pressure field stops a a minimum value of $r = 1.75cm$. The measured sound field is shown in Figs. 5.47 and 5.48, and the corresponding simulated sound field is shown in Fig. 5.49.

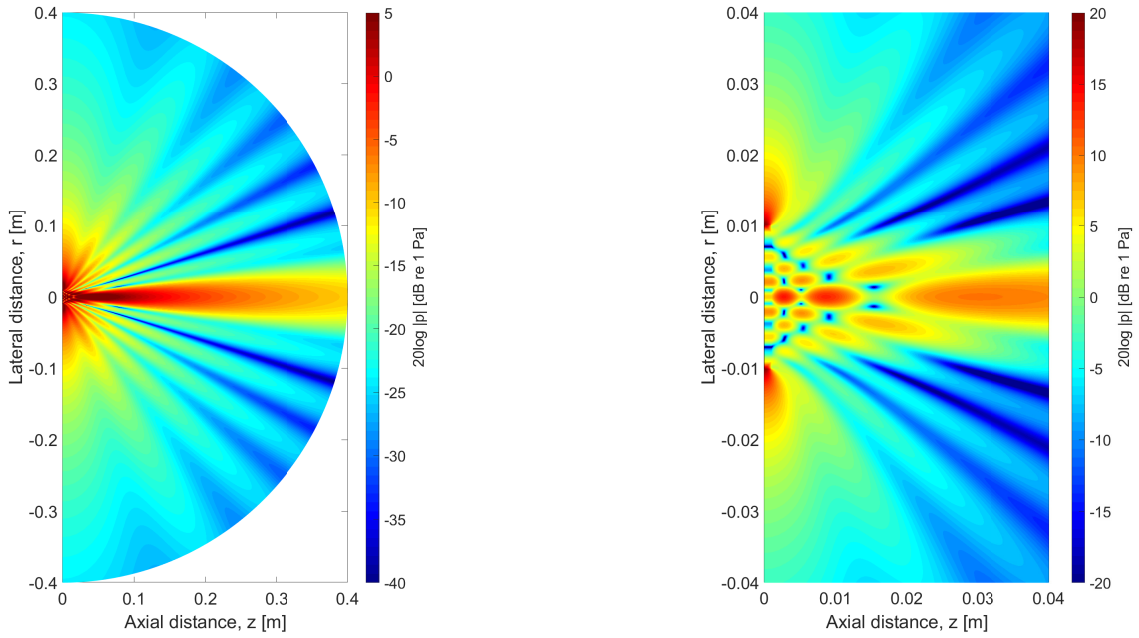
The measured pressure ranges from 0.002 Pa in the far field dips in the beam pattern, to 1.43 Pascal in the main lobe, close to the transmitter. The pressure is calculated the same way as earlier sections, so the same inaccuracy is expected, due to the lack of an accurate receiver sensitivity. While the actual pressure values can not be compared with the simulations, the shape of the beam pattern can be studied and compared.

The sound field produced by the element used in this work is not symmetrical, even though that may be expected from the models used for calculating beam patterns. This might be due to mistakes in the process of lining up the element and microphone (see Section. 3.4), so that the 2D plane where these measurements were taken does not cross the sound axis, but rather is located above or below. Some positional adjustments during this work can be trusted more than others. The horizontal angular alignment is performed using



(A) (B)

FIGURE 5.48: Measured logarithmic 2D sound pressure field, from 0.0175 m out to 0.3 m, at 98.14 kHz (R1). Actual pressure values (A) and interpolated pressure values (B). Each half-circle marks r = 0.1 m.



(A) (B)

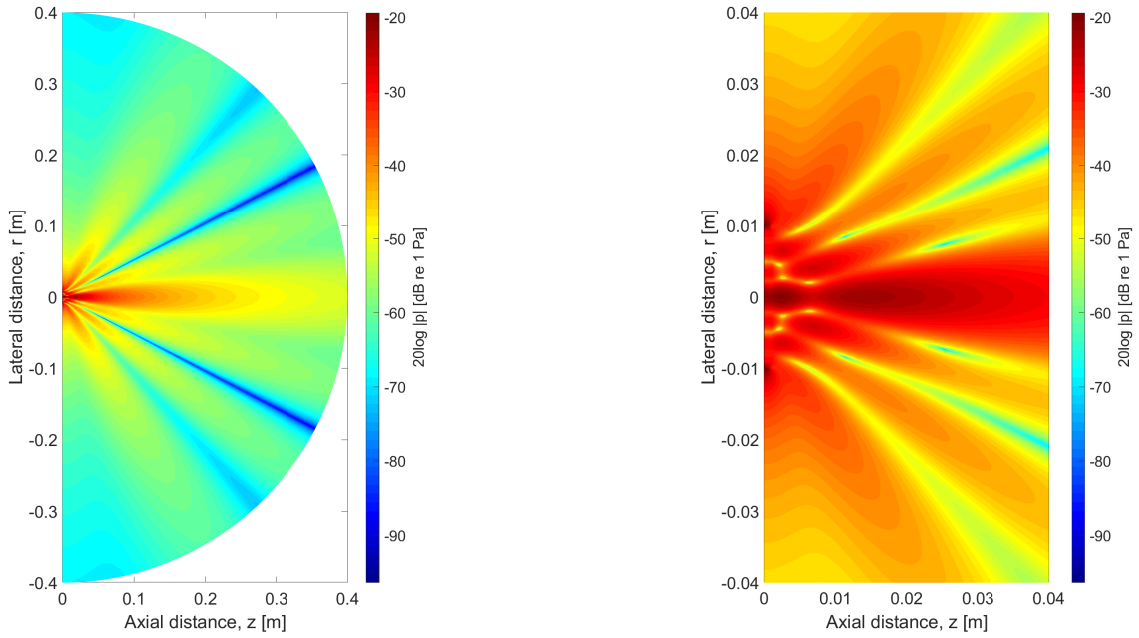
FIGURE 5.49: Simulated logarithmic 2D sound pressure field, from 0 m to 0.4 m, at 98.16 kHz (R1). Far field (A) and zoomed in near field (B).

high accuracy rotational and linear stages, while the vertical angle of the element had to be adjusted manually, with the element hanging from a pair of wires. Therefore, the uncertainty of the vertical angle alignment is expected to be larger.

5.7.2 Simulated sound pressure field in the frequency spectrum 50 kHz to 300 kHz

In this section, the simulated sound pressure field in a frequency range from 50 kHz to 300 kHz is presented, shown in Figs. 5.50 - 5.55. Some of the figures correspond with the directivity measurements and simulations presented in Section 5.6, with additional figures at 260 kHz and 300 kHz, to show the whole frequency spectrum used in this work. The near field of the 2D sound pressure field is also presented, so show how the behaviours in the near field affect the larger far field. No measurements of the near field was possible to perform with any adequate accuracy during this work, so no comparisons with measurements are shown. Due to the way FEMP displays data, a Cartesian coordinate system is used in the simulation figures, rather than the spherical coordinates in the measurement figures.

As the frequency of the transmitted signal increases, a shift from a single solid main lobe in the beam pattern, to a more split up main lobe is observed. At the first radial mode, see Fig. 5.49, the main lobe stands out clearly, with the lowest pressure amplitude being observed between the main lobe and the first side lobes. At 140 kHz, (see Fig. 5.51), the main lobe strength is decreased, and the two first side lobes become where most of the acoustical energy is dumped. In the near field at 140 kHz, it can be seen that the interaction of the sound pressure produced at different points on the surface create a solid beam, emitted orthogonal to the element surface. This effect is not seen at 60 kHz or 98.16 kHz, Figs. 5.50 and 5.49.

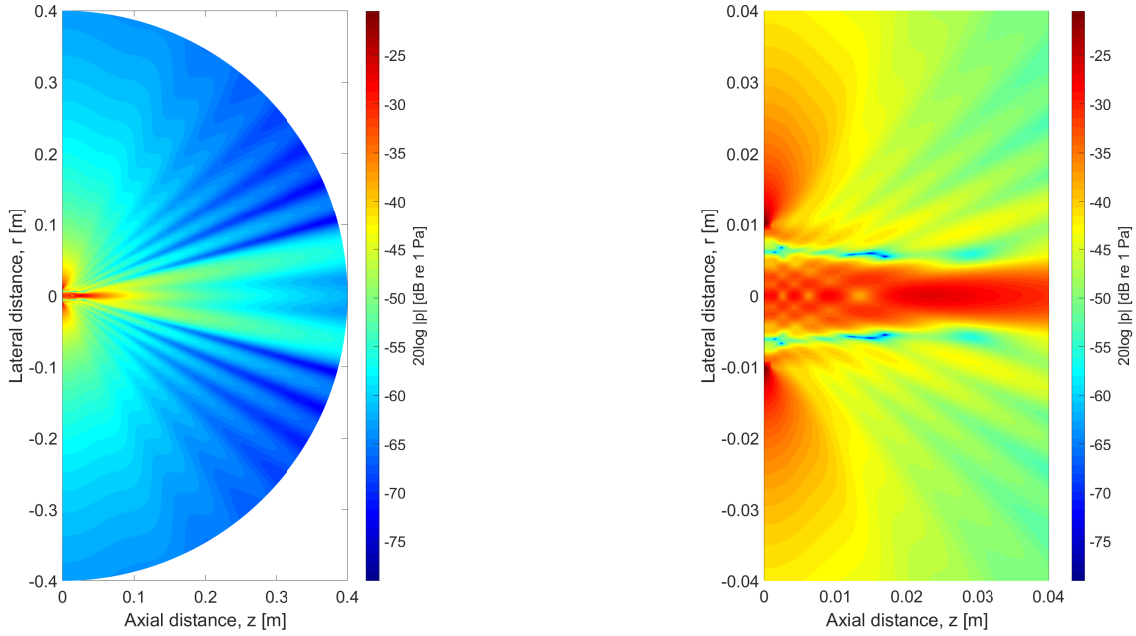


(A) (B)

FIGURE 5.50: Simulated 2D sound pressure field, from 0 m to 0.4 m, at 60 kHz. Far field (A) and zoomed in near field (B).

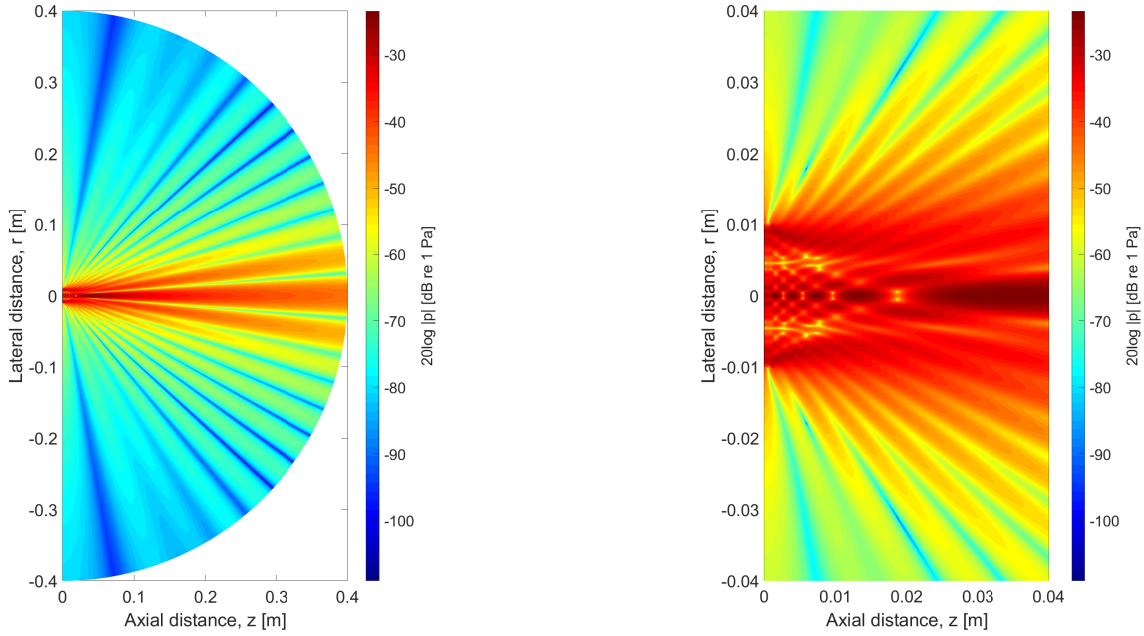
As the frequency is increased further, the pattern of a main lobe and two strong side lobes remain, (see Figs. 5.52 - 5.54). At 260 kHz, the same effect in the near field seen at 140 kHz (Fig. 5.51), can be seen, where most of the energy is concentrated in a beam that does not start spreading out until around 10 cm from the transmitting element surface. Another effect can be observed at 260 kHz, where the sound energy seems to be split across three areas, corresonding to around the two side lobes, and around the main lobe. The areas are divided from each other by the dip observed at around $\pm$ 45 degrees.

From what is observed, the directivity measurements correspond well to the simulations of the 2D sound pressure field.



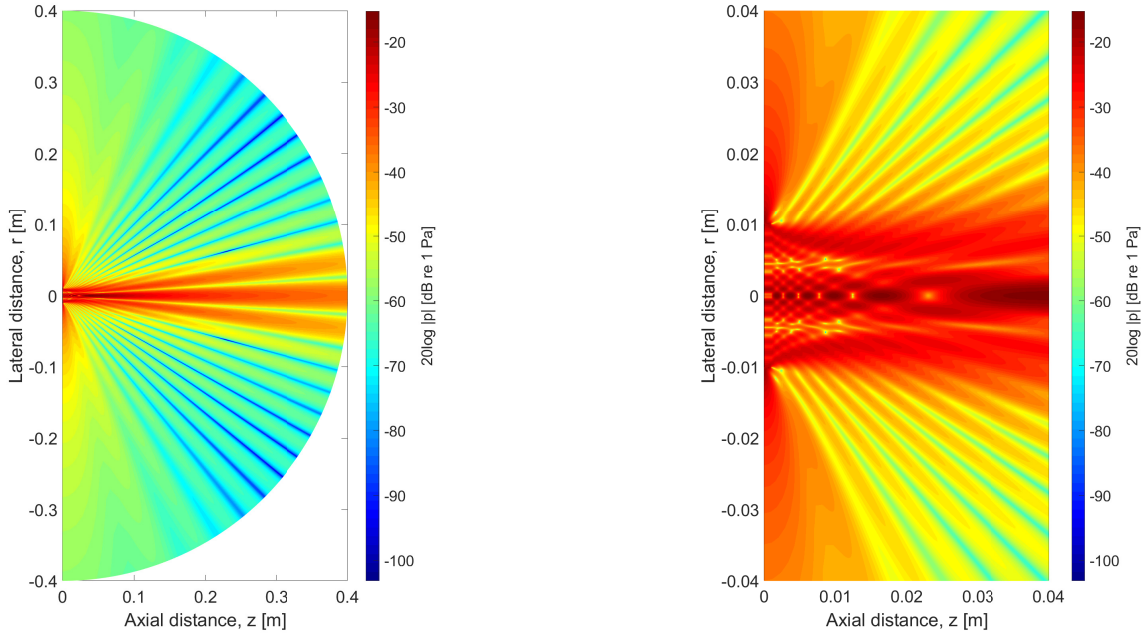
(A) (B)

FIGURE 5.51: Simulated 2D sound pressure field, from 0 m to 0.4 m, at 140 kHz. Far field (A) and zoomed in near field (B).



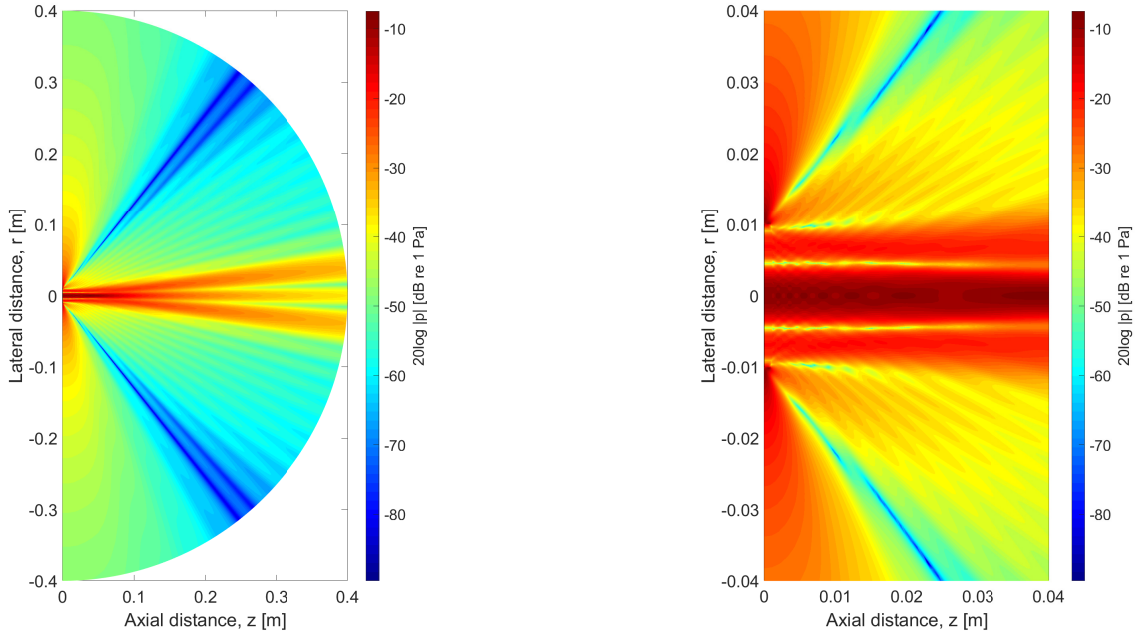
(A) (B)

FIGURE 5.52: Simulated 2D sound pressure field, from 0 m to 0.4 m, at 180 kHz. Far field (A) and zoomed in near field (B).



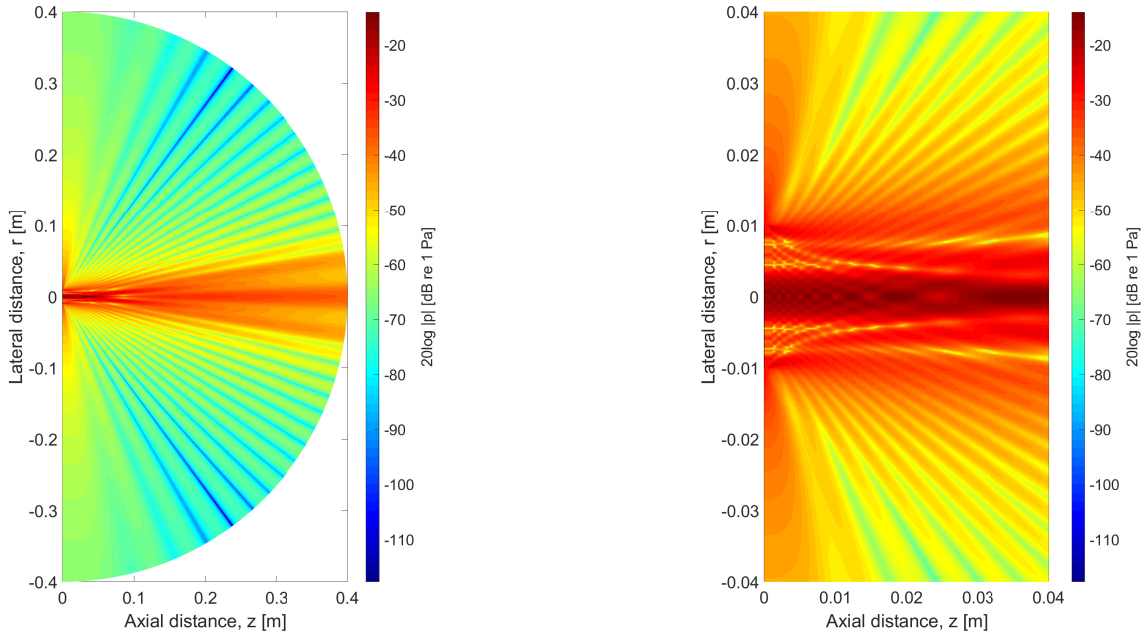
(A) (B)

FIGURE 5.53: Simulated 2D sound pressure field, from 0 m to 0.4 m, at 220 kHz. Far field (A) and zoomed in near field (B).



(A) (B)

FIGURE 5.54: Simulated 2D sound pressure field, from 0 m to 0.4 m, at 260 kHz. Far field (A) and zoomed in near field (B).



(A) (B)

FIGURE 5.55: Simulated 2D sound pressure field, from 0 m to 0.4 m, at 300 kHz. Far field (A) and zoomed in near field (B).

Chapter 6

Conclusions and Further work

6.1 Conclusions

In this work, the methods for measuring and simulating different parameters of a transmit-receive system consisting of a transmitting pieoelectric element and a receiving microphone system has been explored, and results has been compared. The measured and simulated admittance show similar values of resonance magnitude and frequency at R1 and R2, with the greatest difference seen is for element number 17, with a maximum magnitude difference of 1.3 dB at R1, and a maximum resonance frequency difference of 2450 Hz at R2, when compared to the simulations.

The on-axis pressure measurements were performed with good accuracy of positioning and signal setup. The problem encountered during this work was the accurate calculation of the receiver sensitivity $M_V(f)$ of the receiving electronics. Although a slight divergence of the measured and simulated on-axis pressure is expected, due to physical effects not yet modelled in the finite element model, the difference seen in the results to this work can probably not be explained by lack of accurate simulation software. More likely, the calculated sensitivity of the microphone system is not good enough. Due to the lack of numerical data for the frequency-dependant actuator response correction of the microphone, the correction had to be calculated using digitized scans of physical graphs on paper. This most likely added a large uncertainty to the values. Another possibility is that the correction data provided is wrong. The receiver sensitivity of a Bruel & Kjær 4138 microphone system has been calculated before, by Knappskog [51], and Mosland [2]. Both calculated a correction factor that peaks at around 100 kHz and 10 dB, similar to what is calculated in this work. Still, the simulated and measured on-axis pressure amplitude diverge with as much as 8 dB at some frequencies and distances. A more accurate method of characterizing the parameters studied in this work would maybe be by using a piezoelectric element or transducer transmit-receive pair. By using such a setup, three-transducer reciprocity calibration could be used to calculate a more accurate transmitter and receiver sensitivity across the whole desired frequency spectrum, as was demonstrated by Mosland [2]. Another problem encountered during this work was the fact that measurements of pressure in the near-field of an element includes many uncertainties, due to the lack of diffraction correction information and the fact that standing waves may be produced between the transmitter and receiver, further changing the pressure amplitude in the near field. Using a microphone with a small receiver surface, and highly accurate positioning stages, a good picture of the shape of the sound field in the near-field can be achieved, but the actual pressure amplitudes are much more difficult.

The measured transmitter sensitivity $S_V(f)$ calculated in this work suffer from the same problem as the on-axis pressure, as the sensitivity is dependant on the calculated pressure amplitude at a fixed distance, which requires a receiver with an accurate receiver sensitivity. AS it stands, the difference between the measured and simulated transmitter sensitivity lies at around 8 dB, similarly to the difference seen with the on-axis pressure. Knappskog

achieved a similar result, with a simulated transmitter sensitivity much higher than the measured [51].

The directivity beam pattern measurements was compared to simulations calculated at the same distance, r = 0.2 m. The main lobe of the measured beam pattern was very similiar to the simulated beam pattern, but a slight deviation was seen when moving to the first side lobes. It is likely that by increasing the measurement distance r, the measured values would be closer to the simulations, as could be seen in Fig. 5.44, where the beam pattern got closer to the simulated one as the distance increased. Although the measured directivity amplitudes may not line up, the position of the side lobes show little deviation from the simulations, at least for the first few side lobes. However, as the angle increases towards $\pm$ 90 degrees, the deviation in magnitude and lobe shape grows larger. The problem with comparisons at angles larger than $\pm$ 45 degrees, is that the angle resolution of the simulation drops to unusable levels. Between 80 and 90 degrees, the simulation only gives a few samples, meaning any side lobes in those angle ranges are not shown. It is possible that a change could be done to FEMP, to fix this problem.

The measurement of the 2D sound pressure field at 98.14 kHz achieved similar results as the directivity measurements, as the method used is just an extension of the measurement method for measuring directivity. An interesting effect that materialized when the sound field measurements were plottet, was the lack of symmetry, similarly to what is seen in the directivities. When compared to the simulations at the corresponding frequency, 98.16 kHz, the main lobe of the beam pattern seems to be of a similar width, but the pattern of the first side lobes seen in the measurements and simulations diverge noticeably. The first dip in the measured beam pattern seems to split in two outside of an distance of r = 0.2 m, while in the simulations, the dip stays as one. As for the other simulations, it is interesting to see how the behaviours in the near-field affect the pattern in the far-field.

6.2 Further work

Further work could be done around direct measurements of the pressure in the near field. In this work this is attempted by measuring the on axis pressure in the near field, but these values have high inaccuracies, because of the lack of values for the diffraction correction of the microphone system used, and the fact that reflections between the transducer and microphone becomes a significant effect.

It is of interest to run the measurements of on-axis pressure and transmitter sensitivity using a receiver with a better established and more accurate receiver sensitivity. From there, direct comparisons between pressure measurements and FEM simulations can be presented.

More work can also be done around the measurements of directivity beam patterns in the far field. Within the dips in the beam pattern, the signal produced by the transducer becomes so weak, it is difficult to record good signals for post-processing. Since the signal strength can't be increased without introducing non-linear effects to the element, a method for increasing the SNR, or increase the microphone sensitivity could be explored.

In this work, it is assumed that the element is symmetrical around the z-axis, but this is not necessarily true. The results from this work has shown that the beam pattern is not always mirrored around the main lobe, so the beam pattern in other flat planes could be different. Work could be done trying to measure the sound field in 3D, rotating the element around the z-axis as well as the y-axis. This would require a new setup.

The work done in this paper could also be replicated, using a piezoelectric element of a different dimension or type. The beam pattern of fully constructed transducers could also be interesting to measure.

Finally, 2D sound pressure field measurements can be run, using other frequencies than the first series resonance frequency used in this work. However, the measurement method used in this work is very time consuming, so methods of automation should be set up before further work is done on this front.

Appendix A

MATLAB Scripts

A.1 Electrical measurements

The piezoelectric element conductance and susceptance measurements were preformed using the MATLAB script impanal.m, provided by the supervisors.

A.1.1 impanal.m

```
clear all
clc
close all
instrreset
% vinfo = instrhwinfo('visa','agilent');
% % vinfo = instrhwinfo('visa','agilent','gpib');
% vinfo = instrhwinfo('gpib','ni');
vinfo = instrhwinfo('visa','agilent'); %Er ikke installert!
vinfo.ObjectConstructorName

% fff = instrfind;  %sterkere versjon av instrreset
% if ~isempty(fff);pause(0.1);fclose(fff);pause(0.1);delete(fff);
    end
% instrreset
% clear


% obj1 = visa('agilent','GPIB0::0::1::INSTR');
% obj1 = visa('agilent','GPIB0::17::INSTR');
obj1 = visa('agilent','GPIB0::17::INSTR');
% obj1 = gpib('NI', 0, 17);
% obj1 = gpib('agilent', 0, 1);
% obj1 = gpib('NI', 1, 17);


% Connect to instrument object, obj1.
fopen(obj1);
fprintf(obj1, 'V1');

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

     % Osc. level [V]
     amplitude = 1.0;
```

```
33
34
35 f = [1e3:0.1e3:96e3,96.01e3:0.01e3:105e3,105.1e3:0.1e3:245e3,245.01
       e3:0.01e3:255e3,255.1e3:0.1e3:300e3]/1e3;
36 % f = [50e3:5e3:300e3]./5e3;
37
38
39     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40     ol = sprintf('%3.3f',amplitude);
41     % Sett analysator i mode for admittans-maaling
42     fprintf(obj1, ['A2C3F1OL',ol,'EN']);
43
44     % Tids-streng paa format yyyymmddHHMMSS
45     time = datestr(now, 'yyyymmddHHMMSS');
46     % Tittelen som maalingen blir lagra under
47
48     i = 1;
49     ii = 1;
50     antal = length(f);
51     g = ones(1,antal);
52     b = ones(1,antal);
53     fr = ones(1,antal);
54     disp([num2str(antal),' frekvensar.'])
55     disp('Starter maaling...')
56     for freq = f
57         percent = i/antal*100;
58         if percent >= ii*10
59             disp([num2str(ii*10),' %'])
60             ii = ii + 1;
61         end
62
63         %disp(['Frekvens (',num2str(i),' av ',num2str(antal),'): ',
               num2str(freq), ' kHz'])
64         s = sprintf('%3.3f',freq);
65         fprintf(obj1, ['FR',s,'ENEX']);
66         pause(0.1)
67         data1 = fscanf(obj1);
68         d=sscanf(data1,'%4c%f,%4c%f,%2c%f');
69         g(i)=d(5);b(i)=d(10);fr(i)=d(13);
70         i = i + 1;
71     end
72     disp('Maaling ferdig.')
73     disp('Lagrer data...')
74     stoptime = datestr(now, 'yyyymmddHHMMSS');
75     save(stoptime,'g','b','fr')
76     disp('Ferdig!')
```

A.2 Acoustic measurements

The measurements over a frequency span, but with a fixed position were preformed using mainAlt.m, while the measurements where the transmitter moved or rotated between measurements were preformed using angularMeasMain.m and main.m. Both scrips use several subscripts to load measurement parameters and instrument addresses, and adjust instrument parameters.

A.2.1 main.m

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% main.m
% vinfo = instrhwinfo('visa','ni'); %Er ikke installert!
% vinfo.ObjectConstructorName
% Main software for acoustic measurements in air.
% Espen Storheim, 2011
% Based on work by Vidar Knappskog and Magne Aanes.
%
% Modified by Rune Hauge and Eivind Mosland, 2012/2013 (v1.1)
% Modified by Sverre Kongsro Finstad, 2021
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Version number.
airversion = '1.1';

%% Add the subfolders to MATLABs path, just in case.
% Change folder names to exclude spaces.
if (isunix || ismac)
    addpath([pwd '/User input'])
    addpath([pwd '/Kernel'])
    addpath([pwd '/Instrument control etc']);
else
    addpath([pwd '\User input'])
    addpath([pwd '\Kernel'])
    addpath([pwd '\Instrument control etc']);
end

% Load information about the measurement about to be performed.
measurement_parameters

% Initialization of the instruments prior to measurements.

init_instruments

if elMeasDone == 0

% Read the electrical signal.
```

```
38 ch = 1;
39
40 % Adjust scaling according to input voltage.
41 % voltage_scaling = 5;
42 voltage_scaling = [0.02 0.05 0.1 0.2 0.5 1 2 5 10];
43 for ii = 1:(length(voltage_scaling));
44     if (4*voltage_scaling(ii) >= meas.voltage_in)
45         fprintf(instrument.scope,['CH1:SCA ' num2str(
               voltage_scaling(ii))]);
46         break;
47     end
48 end
49
50 disp('Starting measurements of the electrical pulses.')
51 for ii = 1:length(meas.f)
52
53     % Adjust the burst length to ensure temporal resolution.
54     %t = 16*100e-6;
55     %t = 14*100e-6;
56     %t = 0.5e-03;
57     %t = 12*100e-6;
58     t = timeNow;
59
60     % Number if cycles is adjusted according to the given frequency
           so that
61     % the burst length equals 0.5 ms.
62     disp([num2str(meas.f(ii)/1000) ' kHz'])
63     fprintf(instrument.generator,['BM:NCYC ', num2str(floor(meas.f(
           ii)*t))]);
64     fprintf(instrument.generator,['FREQ ', num2str(meas.f(ii))]);
65
66     % Record environmental data.
67 %     [Temp RH] = VaisalaHMT313(instrument.humidity);
68     Temp = ASL_250();
69     C = regexp(Temp,'\d+\.?\d*','match');
70     results.temp_electric(ii) = str2double(C{1});
71 %     results.humidity_electric(ii) = RH;
72     results.electric_time(ii,:) = clock;
73 %     clear Temp RH
74     clear Temp
75
76     % Adjust time window.
77     adjustTime('electric',instrument,meas)
78
79     % Stop aquisition.
80     fprintf(instrument.scope,'ACQ:STATE STOP');
81     % Wait to ensure that the scope wipes its memory.
82     pause(1)
83     % Start aquisition.
84     fprintf(instrument.scope,'ACQ:STATE RUN');
85     % Wait for averaging.
```

```
86      pause(meas.wait_scaling)
87
88
89
90      % Read and save.
91      [dum1 dum2 dum3] = DPO_les(ch,instrument.scope);
92      results.electric_t(ii,:) = dum1;
93      results.electric(ii,:) = dum2;
94      results.electric_timescale(ii) = dum3;
95      results.electric_Vscale(ii) = str2num(query(instrument.scope,['
           CH',num2str(ch),':SCA?']));
96      results.electric_Termination(ii) = str2num(query(instrument.
           scope,['CH',num2str(ch),':TER?']));
97      clear dum1 dum2 dum3
98
99  end
100
101 disp(' ')
102 disp('Finished reading the electrical signal. Now readjusting the
       scope and continuing to acoustic...')
103
104
105 else
106     disp(' ')
107     disp('Electric measurement done. Moving on to acousic
           measurement.')
108 end
109
110 % Read the acoustic pulses.
111 ch = 2;
112
113 disp('Starting measurements of the acoustical pulses.')
114 for ii = 1:length(meas.f);
115
116     % Adjust the bandwidth of the KH-filter
117     % Set the cutoff frequency for channel 1.
118
119     if instAdjusted == 0
120     pause(0.1)
121     fprintf(instrument.filter,['CH1.1']);
122     pause(0.1)
123     fprintf(instrument.filter,['F' num2str((meas.f(ii)/1000)/2) 'K'
           ]);
124     pause(0.1)
125     % Set the cutoff frequency for channel 2.
126     fprintf(instrument.filter,['CH1.2']);
127     pause(0.1)
128     fprintf(instrument.filter,['F' num2str((meas.f(ii)/1000)*2) 'K'
           ]);
129     pause(0.1)
130     end
```

```
131
132     % Adjust the burst length to ensure temporal resolution.
133     %t = 16*100e-6;
134     %t = 14*100e-6;
135 %       t = 12*100e-6;
136     %t = 5*100e-6;
137      t = timeNow;
138
139     % Number if cycles is adjusted according to the given frequency
            so that
140     % the burst length equals 0.5 ms.
141      if instAdjusted == 0
142      disp([num2str(meas.f(ii)/1000) ' kHz'])
143      fprintf(instrument.generator,['BM:NCYC ', num2str(floor(meas.f(
            ii)*t))]);
144      fprintf(instrument.generator,['FREQ ', num2str(meas.f(ii))]);
145      end
146
147
148 %      % Record environmental data.
149 %      [Temp RH] = VaisalaHMT313(instrument.humidity);
150
151 %      results.humidity_acoustic(ii) = RH;
152 %      results.acoustic_time(ii,:) = clock;
153 %      clear Temp RH
154      Temp = ASL_250();
155      C = regexp(Temp,'\d+\.?\d*','match');
156      results.temp_acoustic(ii) = str2double(C{1});
157      results.acoustic_time(ii,:) = clock;
158      clear Temp
159
160     % Adjust time window.
161
162      adjustTime('acoustic',instrument,meas)
163
164     % Adjust amplitude scaling and read out signal.
165      [dum1 dum2 dum3] = adjustAmplitude(2,instrument,meas);
166      results.acoustic_t(ii,:) = dum1;
167      results.acoustic(ii,:) = dum2';
168      results.acoustic_timescale(ii) = dum3;
169      results.acoustic_Vscale(ii) = str2num(query(instrument.scope,['
            CH',num2str(ch),':SCA?']));
170      results.acoustic_Termination(ii) = str2num(query(instrument.
            scope,['CH',num2str(ch),':TER?']));
171      clear dum1 dum2 dum3
172
173 end
174
175 %% Storing data
176 results.electric_f = meas.f;
177 results.acoustic_f = meas.f;
```

```
178
179 xx = strcat(meas.name,'_',datestr(now,'yyyymmddHHMMSS'));
180
181 identity = whos('results');
182
183 if identity.bytes < 2e9
184     save(xx,'results', 'meas', 'instrument');
185 else
186     save(xx,'results', '-v7.3', 'meas', 'instrument');
187 end
188 %% Finishing touches.
189 % Close the instrument ports and clear device handles.
190 instrument_shutdown
```

A.2.2 angularMeasMain.m

```
1  %freq = [60000:20000:80000,98300,140000:40000:220000,249500];
2  freq = 98140;
3  dist = 0.20;
4  x = 1;
5  theta = -92:1:92;
6  %theta = [0 0 0 20 20 20 40 40 40 60 60 60 90 90 90 0 0 0 20 20 20
       40 40 40 60 60 60 90 90 90];
7  %theta = -1:0.1:1;
8  %theta = zeros(1,10) + 0;
9  elMeasDone = 0;
10 timeNow = 0.6e-3;
11 instAdjusted = 0;
12 %theta = 163.2:-0.5:143.2;
13 %theta = -100000:-10000:-170000;
14 while true
15     disp('Next measurement:')
16     disp(theta(x))
17 %     if theta(x) < -40
18 %         timeNow = 0.8e-3;
19 %     elseif theta(x) < -20
20 %         timeNow = 10e-3;
21 %     elseif theta(x) < 20
22 %         timeNow = 0.8e-3;
23 %     elseif theta(x) < 40
24 %         timeNow = 10e-3;
25 %     elseif theta(x) < 91
26 %         timeNow = 0.8e-3;
27 %     end
28     disp(timeNow)
29     loop = input('Continue? (1/0), 2 for redo: '); % 1 to continue
          0 to break loop
30     if isempty(loop)
31         loop = 1;
32     end
33     if loop == 1
```

```
        main
        resultsFull{x} = results;
        measFull{x} = meas;
        elData{x} = results.electric;
        acData{x} = results.acoustic;
        tempElData{x} = results.temp_electric;
        tempAcData{x} = results.temp_acoustic;
        x = x + 1;
        if x == length(theta)+1
           x = length(theta);
        end
        elMeasDone = 1;
        instAdjusted = 1;
    elseif loop == 2
        x = x - 1;
        main
        resultsFull{x} = results;
        measFull{x} = meas;
        elData{x} = results.electric;
        acData{x} = results.acoustic;
        tempElData{x} = results.temp_electric;
        tempAcData{x} = results.temp_acoustic;
        x = x + 1;
        if x == length(theta)+1
           x = length(theta);
        end
        elMeasDone = 1;
        instAdjusted = 1;
    else
        break
    end
end
```

A.2.3 mainAlt.m

```
%
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% main.m
% vinfo = instrhwinfo('visa','ni'); %Er ikke installert!
% vinfo.ObjectConstructorName
% Main software for acoustic measurements in air.
% Espen Storheim, 2011
% Based on work by Vidar Knappskog and Magne Aanes.
%
% Modified by Rune Hauge and Eivind Mosland, 2012/2013 (v1.1)
%
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clear;
```

```
13
14 %% Version number.
15 airversion = '1.1';
16
17 %% Add the subfolders to MATLABs path, just in case.
18 % Change folder names to exclude spaces.
19 if (isunix || ismac)
20     addpath([pwd '/User input'])
21     addpath([pwd '/Kernel'])
22     addpath([pwd '/Instrument control etc']);
23 else
24     addpath([pwd '\User input'])
25     addpath([pwd '\Kernel'])
26     addpath([pwd '\Instrument control etc']);
27 end
28
29 % Load information about the measurement about to be performed.
30 measurement_parameters
31
32 % Initialization of the instruments prior to measurements.
33 init_instruments
34
35 elMeasDone = 1;
36
37
38
39 if elMeasDone == 0
40
41 % Read the electrical signal.
42 ch = 1;
43
44 % Adjust scaling according to input voltage.
45 % voltage_scaling = 5;
46 voltage_scaling = [0.02 0.05 0.1 0.2 0.5 1 2 5 10];
47 for ii = 1:(length(voltage_scaling));
48     if (4*voltage_scaling(ii) >= meas.voltage_in)
49         fprintf(instrument.scope,['CH1:SCA ' num2str(
              voltage_scaling(ii))]);
50         break;
51     end
52 end
53
54 disp('Starting measurements of the electrical pulses.')
55 for ii = 1:length(meas.f)
56
57     % Adjust the burst length to ensure temporal resolution.
58     %t = 16*100e-6;
59     t = 14*100e-6;
60     %t = 0.6e-03;
61     %t = 12*100e-6;
62     %t = timeNow;
```

```
63
64      % Number if cycles is adjusted according to the given frequency
            so that
65      % the burst length equals 0.5 ms.
66      disp([num2str(meas.f(ii)/1000) ' kHz'])
67      fprintf(instrument.generator,['BM:NCYC ', num2str(floor(meas.f(
            ii)*t))]);
68      fprintf(instrument.generator,['FREQ ', num2str(meas.f(ii))]);
69
70      % Record environmental data.
71  %     [Temp RH] = VaisalaHMT313(instrument.humidity);
72      Temp = ASL_250();
73      C = regexp(Temp,'\d+\.?\d*','match');
74      results.temp_electric(ii) = str2double(C{1});
75  %     results.humidity_electric(ii) = RH;
76      results.electric_time(ii,:) = clock;
77  %     clear Temp RH
78      clear Temp
79
80      % Adjust time window.
81      adjustTime('electric',instrument,meas)
82
83      % Stop aquisition.
84      fprintf(instrument.scope,'ACQ:STATE STOP');
85      % Wait to ensure that the scope wipes its memory.
86      pause(1)
87      % Start aquisition.
88      fprintf(instrument.scope,'ACQ:STATE RUN');
89      % Wait for averaging.
90      pause(meas.wait_scaling)
91
92
93
94      % Read and save.
95      [dum1 dum2 dum3] = DPO_les(ch,instrument.scope);
96      results.electric_t(ii,:) = dum1;
97      results.electric(ii,:) = dum2;
98      results.electric_timescale(ii) = dum3;
99      results.electric_Vscale(ii) = str2num(query(instrument.scope,['
            CH',num2str(ch),':SCA?']));
100     results.electric_Termination(ii) = str2num(query(instrument.
            scope,['CH',num2str(ch),':TER?']));
101     clear dum1 dum2 dum3
102
103 end
104
105 % disp(' ')
106 % disp('Finished reading the electrical signal. Now readjusting the
        scope and continuing to acoustic...')
107 %
108 %
```

```
109 else
110     disp(' ')
111     disp('Electric measurement done. Moving on to acousic
            measurement.')
112 end
113
114 % Read the acoustic pulses.
115 ch = 2;
116
117 disp('Starting measurements of the acoustical pulses.')
118 for ii = 1:length(meas.f);
119
120     % Adjust the bandwidth of the KH-filter
121     % Set the cutoff frequency for channel 1.
122
123 %    if instAdjusted == 0
124     pause(0.1)
125     fprintf(instrument.filter,['CH1.1']);
126     pause(0.1)
127     fprintf(instrument.filter,['F' num2str((meas.f(ii)/1000)/2) 'K'
            ]);
128     pause(0.1)
129     % Set the cutoff frequency for channel 2.
130     fprintf(instrument.filter,['CH1.2']);
131     pause(0.1)
132     fprintf(instrument.filter,['F' num2str((meas.f(ii)/1000)*2) 'K'
            ]);
133     pause(0.1)
134  %   end
135
136     % Adjust the burst length to ensure temporal resolution.
137     %t = 16*100e-6;
138     t = 14*100e-6;
139 %     t = 12*100e-6;
140     %t = 6*100e-6;
141     %t = timeNow;
142
143     % Number if cycles is adjusted according to the given frequency
            so that
144     % the burst length equals 0.5 ms.
145     %if instAdjusted == 0
146     disp([num2str(meas.f(ii)/1000) ' kHz'])
147     fprintf(instrument.generator,['BM:NCYC ', num2str(floor(meas.f(
            ii)*t))]);
148     fprintf(instrument.generator,['FREQ ', num2str(meas.f(ii))]);
149     %end
150
151
152 %     % Record environmental data.
153 %     [Temp RH] = VaisalaHMT313(instrument.humidity);
154
```

```
%       results.humidity_acoustic(ii) = RH;
%       results.acoustic_time(ii,:) = clock;
%       clear Temp RH
    Temp = ASL_250();
    C = regexp(Temp,'\d+\.?\d*','match');
    results.temp_acoustic(ii) = str2double(C{1});
    results.acoustic_time(ii,:) = clock;
    clear Temp

    % Adjust time window.
    adjustTime('acoustic',instrument,meas)

    % Adjust amplitude scaling and read out signal.
    [dum1 dum2 dum3] = adjustAmplitude(2,instrument,meas);
    results.acoustic_t(ii,:) = dum1;
    results.acoustic(ii,:) = dum2';
    results.acoustic_timescale(ii) = dum3;
    results.acoustic_Vscale(ii) = str2num(query(instrument.scope,['
        CH',num2str(ch),':SCA?']));
    results.acoustic_Termination(ii) = str2num(query(instrument.
        scope,['CH',num2str(ch),':TER?']));
    clear dum1 dum2 dum3

end

%% Storing data
results.electric_f = meas.f;
results.acoustic_f = meas.f;

xx = strcat(meas.name,'_',datestr(now,'yyyymmddHHMMSS'));

identity = whos('results');

% if identity.bytes < 2e9
%     save(xx,'results', 'meas', 'instrument');
% else
%     save(xx,'results', '-v7.3', 'meas', 'instrument');
% end
%% Finishing touches.
% Close the instrument ports and clear device handles.
instrument_shutdown
```

A.2.4 measurement_parameters.m

```
%
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% measurement_parameters.m
% Information about the calibration of the measurement microphone.
% Part of the software for acoustic measurements in air.
% Espen Storheim, 2011
```

```
% Based on work by Vidar Knappskog and Magne Aanes.
%
% Modified by Rune Hauge and Eivind Mosland, 2012/2013
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This file is designed to be tampered with prior to each measurement.

%% Define initial bandpass filter low and high cutoff frequency
meas.cutoff_1 = 20; % kHz
meas.cutoff_2 = 600; % kHz
% At present, cutoff_1 and cutoff_2 are adjusted for each measurement frequency in
% main.m.

%% General measurement info.
% Version of this software which was used to make the measurements. Should
% be taken from elsewhere.
% meas.version = 'Updated 20/7-2011.';

% Name of the person performing the measurement and date.
meas.name = 'Piezo_mic_test';
TT = clock;
meas.date = [date ', ' num2str(TT(4)) ':' num2str(TT(5))];
clear TT

% Information about the transmitting transducer.
meas.source = 'Pz27 disk, D = 20.0 mm, T = 2.0 mm, Element No. 12 in batch 9/12.';

% Information about the receiving transducer.
meas.receiver = 'B&K Type 4138-A-015 mic';

% Additional notes regarding the specific simulation.
meas.notes = 'Test Sound pressure at z';

%% Distance from transmitter.
%meas.distance = dist;
meas.distance = 0.4;
%meas.z

%% Frequency information [Hz]

% Frekvensoppsett 50-200 kHz elm 16 til elm 10.
%meas.f = [50e3:100:98e3,98.01e3:10:99e3,99.1e3:100:180e3];

% meas.f = frequencies;
% meas.f = 150e3;
```

```
51
52      %meas.f = [50e3:500:88e3,...
53          %88.1e3:100:114.9e3,115e3:500:180e3];
54          %,...
55          %236.1e3:100:255.9e3,...
56          %256e3:500:300e3];
57  %meas.f = [50e3:5000:96e3,95.1e3:100:99.9e3,100e3:5000:180e3];
58  %meas.f = [70e3,98e3,120e3];
59  %meas.f = [50e3:5000:95e3,98100,100e3:5000:180e3];
60
61  %meas.f = 98.14e3;
62  %meas.f = freq;
63  %meas.f = 247e3:0.1e3:250e3;
64  %meas.f = [60000:20000:80000,98300,140000:40000:220000,249500];
65  meas.f = 50e3:1e3:300e3;
66
67  %% Input waveform data.
68  % Vpp voltage out from the signal generator [V]. This is the actual
        voltage
69  % level of the function generator
70  meas.voltage_in = 0.01;
71  % Number of periods in each burst [-]
72  % Only used in initialization of the generator. In main.m the
      number of
73  % cycles is adjusted to fit a certain burst length [ms] specified
      therein.
74  meas.cycles = 40;
75  meas.burst_cycles = meas.cycles;
76  % Burst repetition rate [Hz]
77  meas.burst_period = 40e-3;
78  meas.burst_rate = 1/meas.burst_period;
79
80  % Approximate time before the signal is steady after a voltage
      scaling change.
81   meas.wait_scaling = 7;
82
83  % A note on the input voltage: The signal generator claims that the
        voltage
84  % specified above is the peak to peak voltage. This is the case
      when the
85  % generator is connected to a 50 Ohm load. However, the
      transmitting
86  % transducer typically has an electrical impedance in the kilo Ohm
      range and is connected
87  % directly to the generator. This causes a voltage division which
      depends
88  % of the impedance of the transducer, and hence an impedance
      mismatch.
89
90  %% Oscilloscope parameters.
91  % Number of pulses which the signal is averaged.
```

```
meas.average = 128;
%meas.average = 32;
% Number of data points recorded by the scope.
meas.samples = 1e5;
% Channel used for measurements.
meas.channel = 1;

% Channel number where the signal generator is connected.
meas.channel_electrical = 1;
% Channel number where the oscilloscope is connected.
meas.channel_acoustical = 2;

%% Distance from transmitter to receiver [m].
%meas.distance = meas.z;

%% Total input gain in the B&K 2636 measurement amplifier [dB].
% Only recorded for later reference. Must be set manually.
meas.gain_in = 40;
meas.gain_out = 20;
meas.gain = meas.gain_in + meas.gain_out;
```

A.2.5 instruments.m

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Instruments.m
% Part of the software for acoustic measurements in air.
% Espen Storheim, 2011
% Based on work by Vidar Knappskog and Magne Aanes.
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initialization of the instrumens used in the measurement setup for air.
% Contains MATLAB-handles for the instruments used in the setup, both GPIB,
% serial, and the special functions used by the PI positioning equipment.
%
% This file contains information about many devices in the laboratory, many
% that are not in use on the setup for measurements in air. These are by
% default commented out in the code.
%
% Comment out instruments not in use!
%
% Notes about future updates.
% - Include a test to check for acoustic or impedance measurements.
```

```
20 % - Remove the _idn parameter since there is no common response.
21
22 instrument = {};
23
24 %% Signal generators.
25
26 % Signal Generator: Agilent 33220A. S/N:
27 %instrument.generator = gpib('find','DEV10');
28
29 % instrument.generator = gpib('ni',0,10);
30 % instrument.generator = visa('ni', 'USB0::0x0699::0x0410::C010246
       ::INSTR');
31 instrument.generator = visa('ni', 'GPIB0::10::INSTR');
32 % fopen(instrument.generator)
33
34 % 14.09.2012 Rune Hauge: Include 'instrfind' for locating a GPIB
       object
35 % instrument.generator = instrfind('Type', 'gpib', 'BoardIndex', 0,
        'PrimaryAddress', 10, 'Tag', '');
36
37 % if isempty(instrument.generator)
38 %     instrument.generator = gpib('NI', 0, 10);
39 % else
40 %     fclose(instrument.generator);
41 %     instrument.generator = instrument.generator(1);
42 % end
43
44 fopen(instrument.generator);
45
46 instrument.generator_name = 'Agilent 33220A. S/N: ';
47 instrument.generator_idn = query(instrument.generator,'*IDN?');
48
49 % Test the connection. Should be a command where the response can
       be
50 % verified.
51 if isempty(instrument.generator_idn)
52     disp('Warning: The signal generator is not connected or
          configured properly.')
53 else
54     disp('The signal generator is connected and appears to be
          working.')
55 end
56
57 %% Oscilloscopes.
58
59 % Digital oscilloscope: Tektronix DPO3012. S/N:
60 %instrument.scope =  gpib('ni',0,2);
61
62 % 14.09.2012 Rune Hauge: Include 'instrfind' for locating a GPIB
       object
```

```
63 % instrument.scope = instrfind('Type', 'visa-usb', 'RsrcName', '
       USB0::0x0699::0x0410::C010246::0::INSTR', 'Tag', '');
64 % Alternativt oscilloskop, 09.12.2020
65 instrument.scope = instrfind('Type', 'visa-usb', 'RsrcName', 'USB0
       ::0x0699::0x0410::C024017::INSTR', 'Tag', '');
66
67
68 if isempty(instrument.scope)
69     % Our oscilloscope
70 %     instrument.scope = visa('NI', 'USB0::0x0699::0x0410::C010246
       ::0::INSTR');
71     instrument.scope = visa('NI', 'USB0::0x0699::0x0410::C024017::
           INSTR');
72     % Magne Aanes's oscilloscope
73     % instrument.scope = visa('NI', 'USB0::0x0699::0x0410::C011044
           ::0::INSTR');
74 else
75     fclose(instrument.scope);
76     instrument.scope = instrument.scope(1);
77 end
78 % 20.09.2012 Rune Hauge: Set scope InputBufferSize to high enough
       value.
79 % Trying 1000000.
80 instrument.scope.InputBufferSize = 2000000;
81
82 fopen(instrument.scope)
83 instrument.scope_name = 'Tektronix DPO3012. S/N: ';
84 instrument.scope_idn = query(instrument.scope,'*IDN?');
85
86 % Old oscilloscope: LeCroy ###. S/N:
87
88 % Test the connection. Should be a command where the response can
       be
89 % verified.
90 if isempty(instrument.scope_idn)
91     disp('Warning: The oscilloscope is not connected or configured
           properly.')
92 else
93     disp('The oscilloscope is connected and appears to be working.'
           )
94 end
95
96 %% Environmental parameters.
97
98 % Temperature sensor: ASL F250. S/N:
99 %instrument.temperature = gpib('ni',0,3);
100 % 14.09.2012 Rune Hauge: Include 'instrfind' for locating a GPIB
       object
101 % instrument.temperature = instrfind('Type', 'gpib', 'BoardIndex',
       0, 'PrimaryAddress', 3, 'Tag', '');
102 %
```

```
103 % if isempty(instrument.temperature)
104 %     instrument.temperature = gpib('NI', 0, 3);
105 % else
106 %     fclose(instrument.temperature);
107 %     instrument.temperature = instrument.temperature(1);
108 % end
109 % fopen(instrument.temperature)
110 % set(instrument.temperature,'EOSmode','read&write');
111 % set(instrument.temperature,'EOSCharCode',10); % Set terminator to
      LF.
112 % instrument.temperature_name = 'ASL F250 mk II. S/N: ';
113 % fprintf(instrument.temperature,'A0');
114 % instrument.temperature_idn = fscanf(instrument.temperature);
115 %
116 %% Test the connection. Should be a command where the response can
      be
117 %% verified.
118 % if isempty(instrument.temperature_idn)
119 %     disp('Warning: The thermometer is not connected or configured
      properly.')
120 % else
121 %     disp('The thermometer is connected and appears to be working
      .')
122 % end
123
124 % Pressure sensor: Paroscientific DigiQuartz 740. S/N:
125 %instrument.pressure = serial('COM2','Baudrate',4800,'Terminator','
      cr','Databit',7,'Parity','even');;
126 % instrument.pressure_name = 'Paroscientific DigiQuartz 740. S/N:';
127
128 % Relative humidity and temperature sensor: Vaisala HMT313. S/N:
129 % instrument.humidity = serial('COM5','Baudrate',4800,'Terminator
      ','cr','Databit',7,'Parity','even');
130 % instrument.humidity_name = 'Vaisala HMT313. S/N:';
131 % fopen(instrument.humidity);
132
133 %% Signal processing.
134
135 % Bandpass filter: Krohn-Hite 3940A. S/N: AM2626.
136 %instrument.filter = gpib('ni',0,25);
137 % 14.09.2012 Rune Hauge: Include 'instrfind' for locating a GPIB
      object
138 instrument.filter = instrfind('Type', 'gpib', 'BoardIndex', 0, '
      PrimaryAddress', 25, 'Tag', '');
139
140 if isempty(instrument.filter)
141     instrument.filter = gpib('NI', 0, 25);
142 else
143     fclose(instrument.filter);
144     instrument.filter = instrument.filter(1);
145 end
```

```
fopen(instrument.filter)
instrument.filter_name = 'Krohn-Hite 3940A. S/N: AM2626';
instrument.filter_idn = query(instrument.filter,'*IDN?');


%% Impedance analyzer.

% Impedance analyzer: HP 4192 LF. S/N:
%instrument.impedance = gpib('find','DEV17');
%instrument.impedance_name = 'HP 4192 LF. S/N:';

% Impedance analyzer: Agilent 4294A. S/N:
%instrument.impedance2 = gpib('find','DEV17');
%instrument.impedance2_name = 'Agilent 4194A. S/N:';


%% Positioning equipment.
% Controlling the stages with MATLAB is not possible at present.

% Rotational stage: PI M-037

% Linear stage, horizontal: PI M-531

% Linear stage, vertical: PI M-535
```

A.2.6 init_instruments.m

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% init_instruments.m
% Initialize the instruments according to measurement_parameters
% Part of the software for acoustic measurements in air.
% Espen Storheim, 2011
% Based on work by Vidar Knappskog and Magne Aanes.
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This script is used to initialize the instruments to the proper
    settings.
% Most of the values are taken from the "meas" structure specified
    by the
% user in the m-file "measurement_parameters.m".

instrreset
instruments;

%% Initialize the oscilloscope.
% Code for the Tektronix DPO3012.
if strcmp(instrument.scope_name, 'Tektronix DPO3012. S/N: ')
```

```
19          % Set the acquisition mode to averaging.
20          fprintf(instrument.scope,'ACQ:MOD AVE');
21          % Set the number of cycles to average.
22          fprintf(instrument.scope,['ACQ:NUMAV ' num2str(meas.average
                )]);
23          % Number of points which shall be read from the scope.
24          fprintf(instrument.scope,['HOR:RECO ' num2str(meas.samples)
                ]);
25          % Start point for the recorded signal
26          fprintf(instrument.scope,'DAT:START 1');
27          % Stop point for the recorded signal
28          fprintf(instrument.scope,['DAT:STOP ' num2str(meas.samples)
                ]);
29      % Trigger specifications. Set to edge detection from external
            source.
30          fprintf(instrument.scope,'TRIG:A:EDGE:SOU EXT');   % HER!!!
31      % Set the trigger type to positive edge.
32          fprintf(instrument.scope,'TRIG:A:TYP EDG');
33      % 2012.11.19 EM: Added additional initialization.
34      % CH1
35      % Set Offset to zero.
36      fprintf(instrument.scope,'CH1:OFFS 0');
37      % Set position to zero.
38      fprintf(instrument.scope,'CH1:POS 0');
39      % Set coupling to AC.
40      fprintf(instrument.scope,'CH1:COUP AC');
41      % CH2
42      % Set Offset to zero.
43      fprintf(instrument.scope,'CH2:OFFS 0');
44      % Set position to zero.
45      fprintf(instrument.scope,'CH2:POS 0');
46      % Set coupling to AC.
47      fprintf(instrument.scope,'CH2:COUP AC');
48  end
49
50  %% Initialize the bandpass filter.
51  % Code for Krohn-Hite 3940A filter.
52  %if instAdjusted == 0
53  if strcmp(instrument.filter_name, 'Krohn-Hite 3940A. S/N: AM2626')
54      % There seems to be an overflow when the commands are combined,
            so they
55      % have been separated and a pause of 100 ms is set between each
56      % command.
57      %
58      % Set the input and output gain on both channels to 0 dB.
59      pause(0.1)
60      fprintf(instrument.filter,'AL;0IG;0OG;B');
61      pause(0.1)
62      % Set channel 1 to high pass mode.
63      fprintf(instrument.filter,'CH1.1;M2');
64      pause(0.1)
```

```
    % Set the cutoff frequency for channel 1.
    fprintf(instrument.filter,['F' num2str(meas.cutoff_1) 'K']);
    pause(0.1)
    % Set channel 2 to low pass mode.
    fprintf(instrument.filter,'CH1.2;M1');
    pause(0.1)
    % Set the cutoff frequency for channel 2.
    fprintf(instrument.filter,['F' num2str(meas.cutoff_2) 'K']);
end
% else
%     disp('Filter already adjusted...')
% end

%% Initialize the signal generator.
% This code is for the Agilent 33*** series signal generators
% if instAdjusted == 0
if strcmp(instrument.generator_name, 'Agilent 33220A. S/N: ')
    fprintf(instrument.generator,'OUTP OFF');
    fprintf(instrument.generator,['APPL:SIN ' num2str(meas.f(1)) ' HZ, ' num2str(meas.voltage_in) ' VPP']);
    % Set the trigger to internal and positive slope.
    fprintf(instrument.generator,'TRIG:SOUR IMM');
    fprintf(instrument.generator,'TRIG:SLOP POS');
    % Set the number of periods in one burst.
    fprintf(instrument.generator,['BURS:NCYC ' num2str(meas.burst_cycles)]);
    fprintf(instrument.generator,'BURS:STAT ON');
        % Set the burst rate, i.e. the frequency of the bursts.
        fprintf(instrument.generator,['BM:INT:RATE ' num2str(meas.burst_rate)]);
        % Set the peak voltage.
        fprintf(instrument.generator,['VOLT ' num2str(meas.voltage_in(1))]);
    % Activate the output.
    fprintf(instrument.generator,'TRIG:SLOP POS');
    fprintf(instrument.generator,'OUTP ON');
end
% else
%     disp('Signal generator already adjusted...')
% end
```

A.2.7 adjustAmplitude.m

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% adjustAmplitude.m
%
% [x wf timeDiv] = adjustAmplitude(ch,instrument,meas)
%
% Adjusts voltage scaling and records acoustic data.
```

```
%
% Rune Hauge & Eivind Mosland, 2012
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [x wf timeDiv] = adjustAmplitude(ch,instrument,meas)

% Stop aquisition.
fprintf(instrument.scope,'ACQ:STATE STOP');
% Wait to ensure that the scope wipes its memory.
pause(1)
% Start aquisition.
fprintf(instrument.scope,'ACQ:STATE RUN');
% Wait for averaging.
pause(meas.wait_scaling)

% Read waveform.
[x wf timeDiv] = DPO_les(ch,instrument.scope);
maxV = max(wf);

% Get current scaling.
Scaling = str2num(query(instrument.scope,['CH',num2str(ch),':SCA?']));

% A minimum scaling of 10 mV/div is used to ensure that the noise prior to
% averaging is within the voltage range.
verticalScalings = [10e-3, 20e-3, 50e-3, 100e-3, 200e-3, 500e-3 1 2 5 10];
ind = find(Scaling==verticalScalings);
if isempty(ind)
    disp('ind er tom!')
    ind = 1;
    fprintf(instrument.scope,['CH',num2str(ch),':SCA ',num2str(verticalScalings(ind))]);

    [x wf timeDiv] = DPO_les(ch,instrument.scope);
    maxV = max(wf);
end

% Half the number of vertical division. 8 visible divisons on the screen
% and one additional above and below.
scrnRows = 5;

% Adjust vertical scaling and measure until no clipping.
finished = 0;
while ~finished
    disp(['Current volt/div: ',num2str(verticalScalings(ind))])
    if maxV >= scrnRows*verticalScalings(ind)
```

```
            Scaling = verticalScalings(ind+1);
            fprintf(instrument.scope,['CH',num2str(ch),':SCA ',num2str(
                Scaling)]);
            ind = ind +1;

            % Wait for averaging to finish.
            pause(meas.wait_scaling)

            disp('Measuring')
            [x wf timeDiv] = DPO_les(ch,instrument.scope);
            maxV = max(wf);

        elseif ind ~= 1 && maxV < scrnRows*verticalScalings(ind-1)-(
            scrnRows*verticalScalings(ind-1)*0.1)
            disp('Decreasing scaling')
            Scaling = verticalScalings(ind-1);
            fprintf(instrument.scope,['CH',num2str(ch),':SCA ',num2str(
                Scaling)]);
            ind = ind -1;

            % Wait for averaging to finish.
            pause(meas.wait_scaling)

            disp('Measuring')
            [x wf timeDiv] = DPO_les(ch,instrument.scope);
            maxV = max(wf);
        else
            finished = 1;
        end
end

end
```

A.2.8 adjustTime.m

```
%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% adjustTime.m
%
% adjustTime(type,instrument,meas)
%
% Sets time scaling and adjusts window position.
%
% Rune Hauge & Eivind Mosland, 2012/2013
%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function adjustTime(type,instrument,meas)
    import instrument
```

```
13
14      % Set scaling.
15      if ~strcmp(type,'noise')
16
17          % Get frequeny and number of cycles.
18          freq = query(instrument.generator,'FREQ?');
19          cycles = query(instrument.generator,'BURS:NCYC?');
20          % Compute appropriate scaling.
21          SignalLength = 1/str2num(freq)*str2num(cycles);
22          minScaling = SignalLength/10;
23
24          if minScaling <= 40e-6
25              Scaling = 40e-6;
26          elseif minScaling <= 100e-6
27              Scaling = 100e-6;
28          elseif minScaling <= 200e-6
29              Scaling = 200e-6;
30          elseif minScaling <= 400e-6
31              Scaling = 400e-6;
32          elseif minScaling <= 1e-3
33              Scaling = 1e-3;
34          end
35
36          % disp(['Desired scaling found to be ',num2str(Scaling),...
37          % ' Adjusting scope...'])
38          fprintf(instrument.scope,['HOR:SCA ',num2str(Scaling)]);
39      else
40          noise_Scaling = 40e-6;
41          fprintf(instrument.scope,['HOR:SCA ',num2str(noise_Scaling)
                ]);
42      end
43
44      % Set window position.
45      if strcmp(type,'electric')
46          triggerDelay = Scaling*5;
47      elseif strcmp(type,'acoustic')
48          % Ensures that the onset of the acoustic signal is recorded
                .
49          triggerDelay = Scaling*5 + (meas.distance-0.01)/343;
50      elseif strcmp(type,'noise')
51          triggerDelay = (meas.distance+0.002)/343 - noise_Scaling*5;
52      end
53      fprintf(instrument.scope,['HOR:DEL:TIM ',num2str(triggerDelay)
            ]);
54  end
```

A.2.9 ASL_250.m

```
1   function Temp = ASL_250()
2   %     instrument.temperature = instrfind('Type', 'gpib', '
        BoardIndex', 0, 'PrimaryAddress', 3, 'Tag', '');
```

```
instrument.temperature = gpib('NI', 0, 3);
if isempty(instrument.temperature)
    instrument.temperature = gpib('NI', 0, 3);
else
    fclose(instrument.temperature);
    instrument.temperature = instrument.temperature(1);
end
fopen(instrument.temperature);
set(instrument.temperature,'EOSmode','read&write');
set(instrument.temperature,'EOSCharCode',10); % Set terminator to
    LF.
% instrument.temperature_name = 'ASL F250 mk II. S/N: ';
fprintf(instrument.temperature,'A0');
% instrument.temperature_idn = fscanf(instrument.temperature);

% Test the connection. Should be a command where the response can
    be
% verified.
% if isempty(instrument.temperature_idn)
%     disp('Warning: The thermometer is not connected or configured
    properly.')
% else
%     disp('The thermometer is connected and appears to be working
    .')
% end

Temp = fscanf(instrument.temperature);
fclose(instrument.temperature);



%%     a = fscanf(handle,'%f%f');
%     %a = fscanf(handle);
%     %a = strread(a,'%s',2);
%     T = fscanf(instrument.temperature);
%%     T = a(2);
%%     RH = a(1);
%     fclose(instrument.temperature);
% end
```

A.3 Post-processing scripts

Here, the scripts used for calculating V_{pp} of the signals, as well as scripts for converting data from voltage to pressure. Scripts for calculating and plotting different parameters from data is also included.

A.3.1 changeAcDataToCorrectShape.m

```
% Script to change the measurement data recorded using mainAlt.m,
    to a form that can be read by findPeakToPeak_FFT_k.m
```

```
% Takes in a vector of data cells, and outputs a single matrix with
    all the data
% Sverre Kongsro Finstad 2021
function data = changeAcDataToCorrectShape(acData)
for ii = 1:length(acData)
    data(ii,:) = acData{ii};
end
end
```

A.3.2 correctionAirAbsorbtion.m

```
function korr_trykk = correctionAirAbsorbtion(frekvens,temp)

%% Calculated the absorption coefficient  per meter for all
    frequencies
% Vidar Knappskog 2007
% Modified by Sverre Kongsro Finstad 2021

f = frekvens; %Frequency vector of measurements
T = temp; %Temperature at each measurement

p = 100.5; %Pressure measured using barometer, same value used for
    all measurements due to equipment failure
h_rel = 40; %Standard relative humidity, same value used for all
    measurements due to equipment failure

T_01 = 273.16; %Temperature at 0.01 C in kelvin
p_ref = 101.325; %Reference pressure
T_ref = 293.15; %Reference temperature

%First calculates the molar concentration of water in the air
V = 10.79586*(1-(T_01/T))-5.02808.*log10(T/T_01)+1.50474*...
    (10^-4)*(1-10^(-8.29692*(T/T_01-1)))+0.42873*(10^-3)*...
    (-1+10^(4.76955*(1-T_01/T)))-2.2195983;

h = h_rel*(10^V)*(p/p_ref)^-1;

%Relaxation frequency of oxygen
f_rO = (p/p_ref)*(24+((4.04e-4*h)*(0.02+h)/(0.391+h)));

%Relaxation frequency of nitrogen
f_rN = (p/p_ref)*(T/T_ref)^(-0.5)*(9+280*h*exp(-4.170*...
    ((T/T_ref)^(-1/3)-1)));

%absorption coefficient
alfa = 8.686*f^2*((1.84*(10^-11)*(p/p_ref )^-1*...
    (T/T_ref)^(0.5))+(T/T_ref)^(-5/2)*...
    (0.01275*(exp(-2239.1/T))*(f_rO /(f_rO^2+f^2))...
    + 0.1068*exp(-3352/T)*(f_rN/(f_rN^2+f^2))));

%Absorption in dB/m
```

```
korr_trykk = alfa;
```

A.3.3 correctionTransmittingElectronics.m

```
function [Hvv_1m1, Hvv_1m1_f] = correctionTransmittingElectronics(f
    , fr,g,b)

%% Based on work by Eivind Mosland 2013
% Modified by Sverre Kongsro Finstad 2021

%f = desired frequency vector
%fr = frequency vector of admittance measurement
%g = measured conductance of element
%b = measured susceptance of element

ind1 = find(fr == floor (min (f)/1e3 ));
ind2 = find(fr == ceil(max (f )/1e3 ));

fr = fr(ind1: ind2 )*1e3;
%Calculate impedance of transmitting element

Y = (g(ind1:ind2)+1i*b(ind1: ind2 ));

if length (f) == length (fr)
if (f ~= fr)
Y = interp1 (fr ,Y,f,'linear');
end
else
Y = interp1 (fr ,Y,f,'linear');
end

Z_T = 1./Y;

%Calculate the impedance of the oscilloscope
R = 1e6;
C = 11.5e-12;
Z_C = 1./(1i*2*pi*f*C);
Z_OSC = R*Z_C ./ (R+Z_C);

%% Coax cable

Cx = 100e-12; %cable capacitance per meter
Lx = 250e-9; %cable inductance per meter
cable_length = 3;

Z_0 = sqrt(Lx/Cx);
k_em = 2*pi*f*sqrt(Lx*Cx);

%% Oscilloscope branch to element
Z_a = 1i.*Z_0.*tan(k_em.*(cable_length/2));
Z_b = Z_0./(1i.*sin(k_em.*cable_length));
```

```

%% Calculate Hvv_11m

Hvv_1m1 = (Z_b.*Z_T)./(Z_T.*(Z_a+Z_b)+(Z_a+Z_b).^2 - Z_b.^2);
Hvv_1m1_f = f;
```

A.3.4 directivityCompPlot.m

```
function [normMeasData,logMeas] = directivityCompPlot(acData,
    results,result,simFrequency,theta,t)

%% Script for calculating and plotting measured and simulated
    directivity
% Sverre Kongsro Finstad 2021

%acData = Measurement data vector from mainAlt.m
%results = Measurement parameters from mainAlt.m
%result = Loaded simulated data
%simFrequency = Frequency used for the simulation
%theta = Measurement angle vector, in degrees
%t = Pulse length of measured data

%First calculates peak-to-peak voltage for measurements at all
    angles
signal = acData;
signal_f = results.acoustic_f + zeros(length(signal));
Fs = 100e3./(results.acoustic_timescale*10);
if t == 0.0014
    l_lim = 50000;
    u_lim = 70000;
else
    l_lim = 30000;
    u_lim = 50000;
end

signal = changeAcDataToCorrectShape(signal);

pp_FFT = findPeakToPeak_FFT_k(signal, signal_f, Fs, l_lim, u_lim, t
    );
acData = pp_FFT;

%Preformes curve fitting on measured peak to peak data, and
    normalizes
%measured and simulated directivity values
normMeasData = acData./max(acData);
yy = fit(theta',normMeasData','cubicinterp');
I = find(result.directivity_theta{1} == 0);
I2 = find(result.directivity_f{1} == simFrequency);
%Corrects for the different angle vectors used in measurements and
%simulations
```

```
38 SimData = [flip(result.directivity{1}(1:I,I2))' result.directivity
       {1}(1:I,I2)'];
39 %SimTheta = linspace(-90,90,length(SimData));
40 SimTheta = [-1*(flip(result.directivity_theta{1}(1,1:I)+1.5708))
       result.directivity_theta{1}(1,1:I)+1.5708]*180/pi;
41 NormSimData = abs(SimData)./max(abs(SimData));
42 logSim = 20*log10(NormSimData);
43 logMeas = 20*log10(normMeasData);
44
45 %Plots the measured and simulated directivity linearly and
       logarithmically
46 figure(1)
47 subplot(2,1,1)
48 plot(SimTheta,NormSimData)
49 hold on
50 plot(yy)
51 xlabel('Theta [degrees]')
52 ylabel('V_p_p/V_p_p^{max}')
53 legend('Simulated directivity','Measured directivity')
54 xlim([-90,90])
55 ylim([0,1])
56
57 yy = fit(theta',logMeas','cubicinterp');
58
59 subplot(2,1,2)
60 plot(SimTheta,logSim)
61 hold on
62 plot(yy)
63 xlabel('Theta [degrees]')
64 ylabel('20*log_1_0(V_p_p/V_p_p^{max})')
65 legend('Simulated directivity','Measured directivity')
66 xlim([-90,90])
67 %
68 % hold on
```

A.3.5 findPeakToPeak_FFT_k.m

```
1 function [pp_FFT, angle_FFT,stdDev] = findPeakToPeak_FFT_k(signal,
      signal_f, Fs, l_lim, u_lim, t)
2 % [pp_FFT, angle] = findPeakToPeak_FFT(signal, signal_f, Fs, l_lim,
       u_lim)
3 %
4 % findPeakToPeak_FFT calculates the fft of signal and returnes 2
      vectors:
5 %   pp_FFT: the peak to peak absolute values
6 %   angle: the corresponding angles in radians
7 %       Both vectors of length = length(signal_f)
8 %
9 % input parameters:
10 %       signal: input signal, matrix or vector
11 %       signal_f: frequency vector, vector
```

```
12 %       Fs: sample rate found from:
13 %           Fs = 100e3./(results.acoustic_timescale*10);
14 %               where results.acoustic_timescale is the timescale
   from the
15 %               oscilloscope
16 %       l_lim: lover limit of fft window
17 %       u_lim: uper limit of fft window
18
19 % author K. Andersen, february 2015
20 % based on work by Rune Hauge and Eivind Mosand
21 % modified by Sverre Kongsro Finstad 2021
22
23 %%    work variables, comment out when done coding
24 %     n_upr = 80000;
25 %     n_lvr = 60000;
26 %     Fs = 100e3./(results.acoustic_timescale(1)*10);
27 %     signal_f = results.acoustic_f(1,:);
28
29 %% create shorter signal, plot
30     disp_plot = 0;
31     short_signal = 0;
32     electric = 0;
33
34 if disp_plot == 1
35
36 %     indf_l = find(signal_f == 96e3);
37 %     indf_u = find(signal_f == 120e3);
38 %     signal = signal(indf_l:indf_u,:);
39 %     signal_f = signal_f(:,indf_l:indf_u);
40
41     co = [  0         0.4470    0.7410;
42             0.8500    0.3250    0.0980;
43             0.9290    0.6940    0.1250;
44             0.4940    0.1840    0.5560;
45             0.4660    0.6740    0.1880;
46             0.3010    0.7450    0.9330;
47             0.6350    0.0780    0.1840];
48
49   cnt = 1;
50   ind = 1:length(signal(1,:));
51
52 end
53
54 %% create shorter signal
55
56 if short_signal == 1
57
58     indf_l = find(signal_f == 100e3);
59     indf_u = find(signal_f == 120e3);
60     step = 5;
61     signal = signal(indf_l:step:indf_u,:);
```

```
62      signal_f = signal_f(:,indf_l:step:indf_u);
63  end
64
65  %%  % empty vectors to store fft values
66      pp_FFT = ones(1,length(signal(:,1)));
67      angle_FFT = ones(1,length(signal(:,1)));
68      stdDev = ones(1,length(signal(:,1)));
69
70  %%  % create frequency and timescale vectors
71      signal_f = signal_f + zeros(length(signal(:,1)));
72      Fs = Fs + zeros(length(signal(:,1)));
73
74      for nn = 1:size(signal,1)
75
76          % run filter on signal to smooth out the curve
77          signal(nn,:) = sgolayfilt(signal(nn,:),5,201);
78          % compensate for possible bias by removing mean from signal
79          sig_mean = mean(signal(nn,:));
80          signal(nn,:) = signal(nn,:) - sig_mean;
81          %% call script zeros to obtain list of zeros in signal
                vector
82          zro = crossing(signal(nn,:));
83          %% find local maximum within one period, use next zero
                crossing
84          T = 1/signal_f(nn); % period time
85          Ts = 1/Fs(nn); % time between samples
86          Ns = T/Ts; % number of samples per period
87          l_lim2 = ceil(l_lim + Ns + Ns/4); % making sure that we
                cover one full T
88          u_lim2 = ceil(u_lim + Ns + Ns/4); % making sure that we
                cover one full T
89          % find local max for l_lim and u_lim
90          [loc_l_lim_max_value, loc_l_lim_max] = max(signal(nn, l_lim
                :l_lim2));
91          l_lim_max = l_lim + loc_l_lim_max;
92          [loc_u_lim_max_value, loc_u_lim_max] = max(signal(nn, u_lim
                :u_lim2));
93          u_lim_max = u_lim + loc_u_lim_max;
94          % obtain all the zero crosses up until the limit
95          l_limx = find(zro <= l_lim_max);
96          u_limx = find(zro <= u_lim_max);
97
98          % select the last zero cross of the two vectors
99          try
100             l_limx = l_limx(end);
101             l_limt = zro(l_limx);
102         catch
103             l_limt = l_lim;
104         end
105
106         try
```

```
107             u_limx = u_limx(end);
108             u_limt = zro(u_limx);
109 
110         catch
111             u_limt = u_lim;
112         end
113 
114         % determine length of signal to be processed by FFT
115         signalx = signal(nn, l_limt:u_limt);
116         L = length(signalx);
117 
118         % find peaks to calculate uncertainty of fft transform
119 %         peaks = findpeaks(signalx,'MinPeakHeight',0,'
    MinPeakProminence',0.001);
120 %         maxPeak = max(peaks);
121 %         minPeak = min(peaks);
122 %         u = maxPeak - minPeak;
123 
124         % find standard deviation of the peaks
125         peaks = findpeaks(signalx);
126         stdDev(nn) = std(peaks);
127 
128         % determine the length of the fft signal, the number of fft
              points
129         % needs to be an integer number of the length of the signal
130         NFFT = pow2(nextpow2(20*L));
131         %NFFT = SigL*L;
132         % compute FFT (finally :D )
133         sig_specter = fft(signalx, NFFT)/L;
134 
135         % determine fft frequency vector
136         fFFT = Fs(nn)/2*linspace(0,1,NFFT/2+1);
137         % interpolate the fft values upon the measurement frequency
138         spect_meas = interp1(fFFT, sig_specter(1:NFFT/2+1),
              signal_f);
139         % store the modulus of peak2peak values and angles in
              vectors
140         pp_FFT(nn) = 2*2*abs(spect_meas(nn));
141         angle_FFT(nn) = angle(spect_meas(nn));
142 %         maxMinUncertainty(nn) = u;
143 
144         if mod(nn, 100) == 0
145             disp(['n = ', num2str(nn), ', NFFT = ', num2str(NFFT)])
146         end
147 
148         %% plot if 1
149         if disp_plot == 1
150 %             if mod(nn,10) == 0 % only plot each tenth frequency
151                 % time domain plot
152                 figure(999);
153                 plot(t(nn,:), signal(nn,:), 'b');hold on
```

```
154                 plot(t(nn,l_limt:u_limt), signal(nn,l_limt:u_limt),
                        'r');
155                 title(['frequency ', num2str(signal_f(nn))])
156                 hold off
157
158
159 %                   figure(99)
160 %                   subplot(1,2,1); hold on
161 %                   plot(fFFT, abs(sig_specter(1:NFFT/2+1)), 'color',
      co(cnt,:))
162 %                   title(['frequency ', num2str(signal_f(nn))])
163 %                   xlim([90e3, 130e3])
164 %                   figure(2); hold on
165 %
166 %                   %subplot(1,2,2); hold on
167 %                   xx = hilbert(signal(nn,:));
168 %                   plot(ind, abs(xx), 'color', co(cnt,:))
169 %                   plot(ind, -abs(xx), 'color', co(cnt,:))
170 %                   title(['frequency ', num2str(signal_f(nn))])
171 %
172 %                   if electric == 1
173 %                       plot([ind(l_lim), ind(l_lim)], [-10, 10], '--
      k', 'linewidth', 2)
174 %                       plot([ind(u_lim), ind(u_lim)], [-10, 10], '--
      k', 'linewidth', 2)
175 %                   else
176 %                       plot([ind(l_lim), ind(l_lim)], [-0.05, 0.05],
       '--k', 'linewidth', 2)
177 %                       plot([ind(u_lim), ind(u_lim)], [-0.05, 0.05],
       '--k', 'linewidth', 2)
178 %                   end
179 %
180 %                   cnt = cnt + 1;
181 %
182 %                   if cnt == 7
183 %                       cnt = 1;
184 %                   end
185
186                   k = waitforbuttonpress;
187  %              end
188           end
189
190       end
191
192 end
```

A.3.6 fullFindPtoP.m

```
1 % Script used to calculate full peak to peak voltage for
     directivity
```

```
% measurements at many different distances, combining them to map the peak
% to peak voltage in the whole r/theta-plane

%load the data from the directivity measurements at different distances, in
%the current folder, measured using mainAlt.m
loadHemispheresData
measNum = length(peakToPeakData);
ff = peakToPeakData{1}.freq;
theta = peakToPeakData{1}.theta;
dist = zeros(1,measNum);
distSortIndex = zeros(1,measNum);
peakToPeak2dDataAcoustic = zeros(measNum,length(theta));
peakToPeakStandardDeviationAcoustic = zeros(measNum,length(theta));
peakToPeak2dDataElectric = zeros(measNum,length(theta));
peakToPeakStandardDeviationElectric = zeros(measNum,length(theta));
% have to sort the results, so they get plotted properly

%calculates pp-voltage at each distance, and combines all data in a
%2D-matrix
for ii = 1 : measNum
    currentFile = peakToPeakData{ii};
    acousticData = changeAcDataToCorrectShape(currentFile.acData);
    dist(ii) = currentFile.dist;
end
[out,distIdx] = sort(dist,'ascend');
for ii = 1 : measNum
    currentFile = peakToPeakData{distIdx(ii)};
    acousticData = changeAcDataToCorrectShape(currentFile.acData);
    dist(ii) = currentFile.dist;
    signal = acousticData;
    signal_f = ff + zeros(size(acousticData,1),1);
    Fs = 100e3./(currentFile.results.acoustic_timescale*10) + zeros
        (size(acousticData,1),1);
    if dist(ii) > 5
        l_lim = 20000;
        u_lim = 40000;
    else
        l_lim = 50000;
        u_lim = 70000;
    end
    t = currentFile.t;

    [pp_FFT, angle_FFT,stdDev] = findPeakToPeak_FFT_k(signal,
        signal_f, Fs, l_lim, u_lim, t);
    peakToPeak2dDataAcoustic(ii,:) = pp_FFT;
    peakToPeakStandardDeviationAcoustic(ii,:) = stdDev;
end

%calculate PtoP for electric data
```

```
for ii = 1 : measNum
    currentFile = peakToPeakData{ii};
    electricData = changeAcDataToCorrectShape(currentFile.elData);
    dist(ii) = currentFile.dist;
end
[out,distIdx] = sort(dist,'ascend');
for ii = 1 : measNum
    currentFile = peakToPeakData{distIdx(ii)};
    electricData = changeAcDataToCorrectShape(currentFile.elData);
    dist(ii) = currentFile.dist;
    signal = electricData;
    signal_f = ff + zeros(size(electricData,1),1);
    Fs = 100e3./(currentFile.results.acoustic_timescale*10) + zeros
       (size(electricData,1),1);
    if dist(ii) > 0.05
        l_lim = 20000;
        u_lim = 40000;
    else
        l_lim = 50000;
        u_lim = 70000;
    end
    t = currentFile.t;

    [pp_FFT, angle_FFT,stdDev] = findPeakToPeak_FFT_k(signal,
       signal_f, Fs, l_lim, u_lim, t);
    peakToPeak2dDataElectric(ii,:) = pp_FFT;
    peakToPeakStandardDeviationElectric(ii,:) = stdDev;
end

%Linearly interpolates peak to peak data to smooth out plotting
distInterp = min(dist):0.0001:max(dist);
thetaInterp = -90:0.05:90;
interpData = griddata(theta',dist,peakToPeak2dDataAcoustic,
   thetaInterp',distInterp);
interpDataStdDev = griddata(theta',dist,
   peakToPeakStandardDeviationAcoustic,thetaInterp',distInterp);
dataCompUncert = interpData./interpDataStdDev;
dBpeakToPeak2Ddata = 20*log10(interpData);
dBpeakToPeak2DdataStdDev = 20*log10(interpDataStdDev);
dataCompUncertdB = 20*log10(dataCompUncert);

% figure(1)
% polarPcolor(distInterp,thetaInterp,interpData,'H_v_v')
% figure(2)
% polarPcolor(distInterp,thetaInterp,interpDataStdDev,'H_v_v
   standard deviation')
% figure(3)
% polarPcolor(distInterp,thetaInterp,dataCompUncert,'H_v_v stdDev /
    H_v_v')
% figure(4)
```

```
% polarPcolor(distInterp,thetaInterp,dBpeakToPeak2Ddata,'20*log10(H_v_v)')
% figure(5)
% polarPcolor(distInterp,thetaInterp,dBpeakToPeak2DdataStdDev,'20*log10(H_v_v stdDev)')
% figure(6)
% polarPcolor(distInterp,thetaInterp,dataCompUncertdB,'20*log10(H_v_v stdDev / H_v_v)')
```

A.3.7 loadHemispheresData.m

```
%% Script to load on-axis and 2D pressure sound field data
% Sverre Kongsro Finstad 2021

% The script looks for data files in the current folder open in MATLAB, and
% load all files in a single vector
fds = fileDatastore('*.mat', 'ReadFcn', @importdata);
fullFileNames = fds.Files; %the names of the files are identified and stored
numFiles = length(fullFileNames);
peakToPeakData = {};
%The files are loaded in a single data vector
for ii = 1 : numFiles
    dd = load(fullFileNames{ii});
    peakToPeakData{ii} = dd;
end
```

A.3.8 plothorizontalpressurefield_basic.m

```
% plothorizontalpressurefield_basic.m
% Fargeplott av trykkfeltet. Fungerar baade for PML og uendelege element.
% @author Espen Storheim, 2013(?)
%
% Skriptet er modifisert litt av Eivind Nag Mosland i desember 2020

% Simuleringsnummer, dersom parametrisk
sim = 1;

% Frekvens
f_ind = 1;
FFr = result.nearfieldpressure_f{sim};
disp(['Viser trykkfeltet for f = ' num2str(FFr(f_ind)/1e3) ' kHz'])

% modene = find(FFr == f);

% Sett til ein dersom det er ei pml_koyring, eller ein berre vil sjaa paa dei
% endelege elementa (ikkje det som er rekna ut i dei uendelege).
ispml = 0;

```

```
21 % Hentar ut koordinatane til nodane og det tilh?yrande komplekse
       trykkjet.
22 if ispml
23         P_r = [result.nearfieldpressure_r{sim}.'];
24         P_z = [result.nearfieldpressure_z{sim}.'];
25         P = [result.nearfieldpressure{sim}(:,f_ind)];
26 else
27         P_r = [result.nearfieldpressure_r{sim}.';result.
               farfieldpressure_r{sim}];
28         P_z = [result.nearfieldpressure_z{sim}.';result.
               farfieldpressure_z{sim}];
29         P = [result.nearfieldpressure{sim}(:,f_ind);result.
               farfieldpressure{sim}(:,f_ind)];
30 end
31
32 % Ser paa dB_magnituda til trykket. Kan ogsaa sjaa paa fasen.
33 P = 20*log10(abs(P));
34
35 NEWSIM_press = 0;
36
37 % Delaunay delar matrisa opp i nokre trekantar.
38 TRI = delaunay(P_z,P_r);
39
40 X = [];
41 Y = [];
42 C = [];
43
44 % Her er det endring fra FEMP originalt. Har bytta om P_z og P_r
       samanlikna
45 % med originalfila. Grunnen til dette er at det er enklare aa
       plotte det
46 % slik enn aa rotere labels og ticks, m.m.
47 for ii = 1:3
48         X(1:size(TRI,1),ii) = P_z(TRI(:,ii));
49         Y(1:size(TRI,1),ii) = P_r(TRI(:,ii));
50         C(1:size(TRI,1),ii) = P(TRI(:,ii));
51 end
52
53 %% Plotting av feltet.
54 % Plottar feltet for r => 0.
55 fig = figure;
56 h1 = patch(X.',Y.',C.');
57 %h1 = contourf(X.',Y.',C.');
58 set(h1,'edgecolor','none');
59
60 % Plottar feltet for r <= 0. No har ein dobbelt opp i r = 0.
61 hold on
62 h2 = patch(X.',-Y.',C.');
63 set(h2,'edgecolor','none');
64 hold off
65
```

```
%% Begrensar omraada.
xlim([-0 0.3])
ylim([-0 0.3])

%% Colorbar og aksetekstar.
defaxis = gca;
h = colorbar;
ylabel(h, '20log |p| [dB re 1 Pa]')
axes(defaxis)
xlabel('Axial distance, z [m]')
ylabel('Lateral distance, r [m]')
axis image

%% Teikn inn ein mottakar.
% Teiknar inn mottakaren (fritt-felt) i feltet.
%% hold on
%% fill([r_z+r_t/2 r_z+3*r_t/2 r_z+3*r_t/2 r_z+r_t/2],[-r_r -r_r
    r_r r_r],'w')
%% hold off
%%
%% hold on
%% rectangle('Position',[-0.04 -a-0.04 0.14+0.04 2*(a+0.04) ])
%% hold off


%% Teikn inn sendaren.
hold on
a = 10e-3;
t = 0.002;
z_rec = -0.002;
fill([z_rec z_rec+2e-3 z_rec+2e-3 z_rec],[-a -a a a],'w')
hold off


box on
axis on

%% Skalering og lagring av figur.
% set(gcf, 'PaperPositionMode', 'manual');
% set(gcf, 'PaperUnits', 'points');
% set(gcf, 'PaperPosition', [0 0 1200 600]);
%
% print(gcf,'-depsc2','-painters','figur_namn.eps')

xlim([-0.4 0.4])
ylim([-0.4 0.4])
```

A.3.9 polarPcolor.m

```
%% Script to plot measured 2D sound field data
%% Author
```

```
3  % Etienne Cheynet, University of Stavanger, Norway. 23/10/2019
4  % see also pcolor
5  % Modified by Sverre Kongsro Finstad 2021
6
7  function [varargout] = polarPcolor(R,theta,Z,colorbarlabel,varargin
       )
8  % [h,c] = polarPcolor1(R,theta,Z,varargin) is a pseudocolor plot of
       matrix
9  % Z for a vector radius R and a vector angle theta.
10 % The elements of Z specify the color in each cell of the
11 % plot. The goal is to apply pcolor function with a polar grid,
       which
12 % provides a better visualization than a cartesian grid.
13 %
14 %% Syntax
15 %
16 % [h,c] = polarPcolor(R,theta,Z)
17 % [h,c] = polarPcolor(R,theta,Z,'Ncircles',10)
18 % [h,c] = polarPcolor(R,theta,Z,'Nspokes',5)
19 % [h,c] = polarPcolor(R,theta,Z,'Nspokes',5,'colBar',0)
20 % [h,c] = polarPcolor(R,theta,Z,'Nspokes',5,'labelR','r (km)')
21 %
22 % INPUT
23 %       * R :
24 %         - type: float
25 %         - size: [1 x Nrr ] where Nrr = numel(R).
26 %         - dimension: radial distance.
27 %       * theta :
28 %         - type: float
29 %         - size: [1 x Ntheta ] where Ntheta = numel(theta).
30 %         - dimension: azimuth or elevation angle (deg).
31 %         - N.B.: The zero is defined with respect to the North.
32 %       * Z :
33 %         - type: float
34 %         - size: [Ntheta x Nrr]
35 %         - dimension: user's defined .
36 %       * varargin:
37 %         - Ncircles: number  of circles for the grid definition.
38 %         - autoOrigin: 'on' (the first circle of the plar grid has
       a radius
39 %           equal to the lowest value of R) or 'off'.
40 %         - Nspokes: number of spokes for the grid definition.
41 %         - colBar: display the colorbar or not.
42 %         - labelR: title for radial axis.
43 %         - RtickLabel: Tick label for the radial axis.
44 %         - colormap: Colormap for the pcolor function
45 %         - ncolor: Number of colors in the colorbar and pcolor
46 %         - circlesPos: position of the circles with respect to the
       origin
47 %       (it overwrites Ncircles if necessary)
48 %
```

```
%
% OUTPUT
% h: returns a handle to a SURFACE object.
% c: returns a handle to a COLORBAR object.
%
%% Examples
% R = linspace(3,10,100);
% theta = linspace(0,180,360);
% Z = linspace(0,10,360)'*linspace(0,10,100);
% figure
% polarPcolor(R,theta,Z,'Ncircles',3)
%
%% Author
% Etienne Cheynet, University of Stavanger, Norway. 23/10/2019
% see also pcolor
%

%%  InputParseer
p = inputParser();
p.CaseSensitive = false;
p.addOptional('Ncircles',5);
p.addOptional('autoOrigin','on');
p.addOptional('Nspokes',8);
p.addOptional('labelR','');
p.addOptional('RtickLabel',[]);
p.addOptional('colBar',1);
p.addOptional('Rscale','linear');
p.addOptional('colormap','parula');
p.addOptional('ncolor',[]);
p.addOptional('typeRose','meteo'); % 'meteo' or 'default'
p.addOptional('circlesPos',[]);
p.parse(varargin{:});

Ncircles = p.Results.Ncircles;
Nspokes = p.Results.Nspokes ;
labelR = p.Results.labelR ;
RtickLabel = p.Results.RtickLabel ;
colBar = p.Results.colBar ;
Rscale = p.Results.Rscale ;
autoOrigin = p.Results.autoOrigin ;
myColorMap = p.Results.colormap ;
ncolor = p.Results.ncolor ;
circPos = p.Results.circlesPos ;
typeRose = p.Results.typeRose ;


if ~isempty(circPos)
    Origin = max([min(circPos),min(R)]);
    circPos(circPos<min(R))=[];
    circPos(circPos>max(R))=[];
elseif strcmpi(autoOrigin,'on')
```

```
100     Origin = min(R);
101 elseif strcmpi(autoOrigin,'off')
102     Origin = 0;
103 else
104     error(' ''autoOrigin'' must be ''on'' or ''of'' ')
105 end
106
107 if Origin==0 && strcmpi(Rscale,'log')
108     warning(' The origin cannot be set to 0 if R is expressed on a
            logarithmic axis. The value ''Rmin'' is used instead')
109     Origin = min(R);
110 end
111
112 if isempty(circPos)
113     if ~isempty(RtickLabel)
114         if numel(RtickLabel)~=Ncircles
115             error(' The radial ticklabel must be equal to Ncircles'
                    );
116         end
117         if any(cellfun(@ischar,RtickLabel)==0)
118             error(' The radial ticklabel must be a cell array of
                    characters');
119         end
120     end
121 end
122
123 if ~isempty(circPos)
124     circPos = unique([min(R),circPos,max(R)]);
125 end
126 %% Preliminary checks
127 % case where dimension is reversed
128 Nrr = numel(R);
129 Noo = numel(theta);
130 if isequal(size(Z),[Noo,Nrr]) && Noo~=Nrr,
131     Z=Z';
132 end
133
134 % case where dimension of Z is not compatible with theta and R
135 if ~isequal(size(Z),[Nrr,Noo])
136     fprintf('\n')
137     fprintf([ 'Size of Z is : [',num2str(size(Z)),'] \n']);
138     fprintf([ 'Size of R is : [',num2str(size(R)),'] \n']);
139     fprintf([ 'Size of theta is : [',num2str(size(theta)),'] \n\n'
            ]);
140     error(' dimension of Z does not agree with dimension of R and
            Theta')
141 end
142 %% data plot
143 rMin = min(R);
144 rMax = max(R);
145 thetaMin=min(theta);
```

```
146 thetaMax =max(theta);
147
148 % Definition of the mesh
149 cax = newplot;
150 Rrange = rMax - rMin; % get the range for the radius
151 [rNorm] = getRnorm(Rscale,Origin,R,Rrange); % getRnorm is a nested
        function
152 YY = (rNorm)'*cosd(theta);
153 XX = (rNorm)'*sind(theta);
154 h = pcolor(XX,YY,Z,'parent',cax);
155
156 if ~isempty(ncolor)
157     cmap = feval(myColorMap,ncolor);
158     colormap(gca,cmap);
159 else
160     colormap(gca,myColorMap);
161 end
162
163 % disp([max(R/Rrange),max(rNorm)])
164
165 shading flat
166 set(cax,'dataaspectratio',[1 1 1]);axis off;
167 if ~ishold(cax);
168     % make a radial grid
169     hold(cax,'on')
170     % Draw circles and spokes
171     createSpokes(thetaMin,thetaMax,Ncircles,circPos,Nspokes);
172     createCircles(rMin,rMax,thetaMin,thetaMax,Ncircles,circPos,
            Nspokes)
173 end
174
175 %% PLot colorbar if specified
176 if colBar==1,
177     c =colorbar('location','WestOutside');
178     caxis([quantile(Z(:),0.01),quantile(Z(:),0.99)])
179     c.Label.String = colorbarlabel;
180 else
181     c = [];
182 end
183
184 %% Outputs
185 nargoutchk(0,2)
186 if nargout==1,
187     varargout{1}=h;
188 elseif nargout==2,
189     varargout{1}=h;
190     varargout{2}=c;
191 end
192
```

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Nested functions
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    function createSpokes(thetaMin,thetaMax,Ncircles,circlesPos,Nspokes)

        spokeMesh = linspace(thetaMin,thetaMax,Nspokes);
        if isempty(circlesPos)
            circleMesh = linspace(rMin,rMax,Ncircles);
        else
            circleMesh  =  circlesPos;
        end
        contourD = abs((circleMesh - circleMesh(1))/Rrange+R(1)/Rrange);

        if strcmpi(typeRose,'meteo')
            cost = cosd(90-spokeMesh); % the zero angle is aligned with North
            sint = sind(90-spokeMesh); % the zero angle is aligned with North
        elseif strcmpi(typeRose,'default')
            cost = cosd(spokeMesh); % the zero angle is aligned with east
            sint = sind(spokeMesh); % the zero angle is aligned with east
        else
            error('"type" must be "meteo" or "default" ');
        end

        for kk = 1:Nspokes

            X = cost(kk)*contourD;
            Y = sint(kk)*contourD;

            if  Origin==0
                X(1)=Origin;
                Y(1)=Origin;
            end
            plot(X,Y,'color',[0.5,0.5,0.5],'linewidth',0.75,...
                'handlevisibility','off');
            % plot graduations of angles
            % avoid superimposition of 0 and 360
            if and(thetaMin==0,thetaMax == 360),
                if spokeMesh(kk)<360,

                    text(1.05.*contourD(end).*cost(kk),...
                        1.05.*contourD(end).*sint(kk),...
```

```
234                         [num2str(spokeMesh(kk),3),char(176)],...
235                         'horiz', 'center', 'vert', 'middle');
236                 end
237             else
238                 text(1.05.*contourD(end).*cost(kk),...
239                     1.05.*contourD(end).*sint(kk),...
240                     [num2str(spokeMesh(kk),3),char(176)],...
241                     'horiz', 'center', 'vert', 'middle');
242             end
243
244         end
245     end
246     function createCircles(rMin,rMax,thetaMin,thetaMax,Ncircles,
        circlePos,Nspokes)
247
248         if isempty(circlePos)
249             if Origin ==0 % if the origin is set at rMin
250                 contourD = linspace(0,1+R(1)/Rrange,Ncircles);
251             else % if the origin is automatically centered at 0
252                 contourD = linspace(0,1,Ncircles)+R(1)/Rrange;
253             end
254         else
255
256             contourD = circlePos-circlePos(1);
257             contourD = contourD./max(contourD)*max(R/Rrange);
258             contourD =[contourD(1:end-1)./contourD(end),1]+R(1)/
                Rrange;
259         end
260
261         if isempty(circlePos)
262             if strcmpi(Rscale,'linear')||strcmpi(Rscale,'lin'),
263                 tickMesh = linspace(rMin,rMax,Ncircles);
264             elseif strcmpi(Rscale,'log')||strcmpi(Rscale,'
                logarithmic'),
265                 tickMesh  = logspace(log10(rMin),log10(rMax),
                    Ncircles);
266             else
267                 error('''Rscale'' must be ''log'' or ''linear'' ');
268             end
269         else
270             tickMesh  = circlePos;
271             Ncircles = numel(tickMesh);
272         end
273
274
275         % define the grid in polar coordinates
276
277
278         if strcmpi(typeRose,'meteo')
279             angleGrid = linspace(90-thetaMin,90-thetaMax,100);
280         elseif strcmpi(typeRose,'default')
```

```
281             angleGrid = linspace(thetaMin,thetaMax,100);
282         else
283             error('"type" must be "meteo" or "default" ');
284         end
285
286
287
288
289         xGrid = cosd(angleGrid);
290         yGrid = sind(angleGrid);
291
292         spokeMesh = linspace(thetaMin,thetaMax,Nspokes);
293
294         % plot circles
295         for kk=1:length(contourD)
296             X = xGrid*contourD(kk);
297             Y = yGrid*contourD(kk);
298             plot(X,Y,'color',[0.5,0.5,0.5],'linewidth',1);
299         end
300         % radius tick label
301         for kk=1:Ncircles
302
303             position = 0.51.*(spokeMesh(min(Nspokes,round(Ncircles
                    /2)))+...
304                  spokeMesh(min(Nspokes,1+round(Ncircles/2))));
305
306             if strcmpi(typeRose,'meteo'),position = 90-position;
                    end
307
308
309
310             if isempty(RtickLabel),
311                 rtick = num2str(tickMesh(kk),2);
312             else
313                 rtick = RtickLabel(kk);
314             end
315             if abs(round(position)) ==90,
316                 % radial graduations
317                 text((contourD(kk)).*cosd(position),...
318                     (0.1+contourD(kk)).*sind(position-4),...
319                     rtick,'verticalalignment','BaseLine',...
320                     'horizontalAlignment', 'center',...
321                     'handlevisibility','off','parent',cax);
322                 % annotate spokes
323                 text(contourD(end).*0.6.*cosd(position),...
324                     0.07+contourD(end).*0.6.*sind(position),...
325                     [labelR],'verticalalignment','bottom',...
326                     'horizontalAlignment', 'right',...
327                     'handlevisibility','off','parent',cax);
328             else
329                 % radial graduations
```

```
330                 text((contourD(kk)).*cosd(position),...
331                     (contourD(kk)).*sind(position),...
332                     rtick,'verticalalignment','BaseLine',...
333                     'horizontalAlignment', 'right',...
334                     'handlevisibility','off','parent',cax);
335
336                 % annotate spokes
337                 text(contourD(end).*0.6.*cosd(position),...
338                     contourD(end).*0.6.*sind(position),...
339                     [labelR],'verticalalignment','bottom',...
340                     'horizontalAlignment', 'right',...
341                     'handlevisibility','off','parent',cax);
342             end
343         end
344
345     end
346     function [rNorm] = getRnorm(Rscale,Origin,R,Rrange)
347         if strcmpi(Rscale,'linear')||strcmpi(Rscale,'lin')
348                 rNorm = R-R(1)+Origin;
349                 rNorm = (rNorm)/max(rNorm)*max(R/Rrange);
350         elseif strcmpi(Rscale,'log')||strcmpi(Rscale,'logarithmic')
351             if rMin<=0
352                 error(' The radial vector cannot be lower or equal
                        to 0 if the logarithmic scale is used');
353             end
354             rNorm = log10(R); %normalized radius [0,1]
355             rNorm =rNorm-rNorm(1);
356             rNorm = (rNorm)/max(rNorm)*max(R/Rrange);
357         else
358             error('''Rscale'' must be ''log'' or ''linear'' ');
359         end
360     end
361 end
```

A.3.10 PtoPOnAxis.m

```
1  %% Code to calculate and plot on axis pressure
2  % Sverre Kongsro Finstad 2021
3  function PtoPOnAxis(measFreq,simFreq)
4
5  %Load data and structure data storage vectors, current folder needs
       to be
6  %the folder with measurement data files
7  loadHemispheresData
8  measNum = length(peakToPeakData);
9  dist = zeros(1,measNum);
10 distSortIndex = zeros(1,measNum);
11
12 for ii = 1 : measNum
13     currentFile = peakToPeakData{ii};
14     dist(ii) = currentFile.meas.distance;
```

```
15 end
16
17 [out,distIdx] = sort(dist,'ascend');
18
19 %Calculate input signal peak to peak voltage
20 currentFile = peakToPeakData{distIdx(1)};
21 electricData = currentFile.results.electric;
22 dist(ii) = currentFile.meas.distance;
23 signal = electricData;
24 signal_f = currentFile.results.acoustic_f;
25
26 i = find(signal_f ~= measFreq);
27
28 signal(i,:) = [];
29 signal_f(i) = [];
30
31 Fs = 100e3./(currentFile.results.acoustic_timescale*10);
32 Fs(i) = [];
33
34 %FFT window is adjusted based on input data
35 l_lim = 50000;
36 u_lim = 70000;
37
38 t = currentFile.t;
39
40 [pp_FFT, angle_FFT,stdDev] = findPeakToPeak_FFT_k(signal, signal_f,
       Fs, l_lim, u_lim, t);
41
42 inSignalVoltage = pp_FFT;
43
44 %Calculate peak to peak voltage of all signal pulses
45 for ii = 1 : measNum
46     currentFile = peakToPeakData{distIdx(ii)};
47     acousticData = currentFile.results.acoustic;
48     dist(ii) = currentFile.meas.distance;
49     signal = acousticData;
50     signal_f = currentFile.results.acoustic_f;
51
52     %i = find(signal_f < 50000 | signal_f == 98100);
53     i = find(signal_f ~= measFreq);
54
55     signal(i,:) = [];
56     signal_f(i) = [];
57
58     Fs = 100e3./(currentFile.results.acoustic_timescale*10);
59     Fs(i) = [];
60
61     l_lim = 50000;
62     u_lim = 70000;
63
64     t = currentFile.t;
```

```
65
66      [pp_FFT, angle_FFT,stdDev] = findPeakToPeak_FFT_k(signal,
            signal_f, Fs, l_lim, u_lim, t);
67
68      % Add correction due to loss from absoption in air
69      temp = mean(currentFile.results.temp_acoustic);
70      korr_trykk = correctionAirAbsorbtion(signal_f,temp+273.15);
71      C_a = 10^(korr_trykk*(currentFile.meas.distance/20));
72
73      pp_FFT = pp_FFT*C_a;
74
75      peakToPeakResults(1:length(pp_FFT),ii) = pp_FFT;
76      ff(1:length(signal_f),ii) = signal_f';
77  end
78
79  %Load data for caculating pressure from voltage, adjust path based
        on file
80  %location
81
82  FileName    = 'finalReceiverSensitivitydB_0_180kHz.mat';
83  %FolderName = 'C:\Users\sfi011\OneDrive - University of Bergen\
        Mastergreier';
84  FolderName = 'D:\OneDrive - University of Bergen\Mastergreier';
85  File        = fullfile(FolderName, FileName);
86  load(File);    % not: load('File')
87
88  %Load simulation data. Change path based on desired data
89
90  FileName    = 'test3_result.mat';
91  %FolderName = 'C:\Users\sfi011\OneDrive - University of Bergen\
        Mastergreier\Simulations\On axis pressure\50k-180k_20cm_usable';
92  FolderName = 'D:\OneDrive - University of Bergen\Mastergreier\
        Simulations\On axis pressure\NyBrukDisse\OnAxis98160Hz';
93  File        = fullfile(FolderName, FileName);
94  load(File);    % not: load('File')
95
96  %Calculate pressure of measured data
97  [dBPressure,linearPressureAmplitude,linearPressurePtoP] = V_to_Pa(
        signal_f,peakToPeakResults,ff_MvFull,MvFullFrequencySpectrum);
98
99  %fitobject = fit(dist',peakToPeakResults','cubicinterp');
100
101 ii = find(result.onaxispressure_f{1} == simFreq);
102
103 figure(1)
104 subplot(2,1,1)
105 plot(dist,linearPressureAmplitude)
106 hold on
107 %plot(fitobject)
108 plot(result.onaxispressure_r{1},abs(result.onaxispressure{1}(:,ii))
        .*(inSignalVoltage/2))
```

```
109 xlim([0,0.4])
110
111 xlabel('z [m]')
112 ylabel('p_a_x [Pa]')
113
114 legend('Measured sound pressure amplitude','Simulated sound
    pressure amplitude')
115
116 subplot(2,1,2)
117 plot(dist,20*log10(linearPressureAmplitude./sqrt(2)))
118 hold on
119 %plot(fitobject)
120 plot(result.onaxispressure_r{1}-2.0407e-3/2,20*log10(abs(result.
    onaxispressure{1}(:,ii)).*(inSignalVoltage/2)./sqrt(2)))
121 xlim([0,0.4])
122
123 xlabel('z [m]')
124 ylabel('SPL [dB re 1 Pa]')
125
126 legend('Measured SPL','Simulated SPL')
```

A.3.11 V_to_Pa.m

```
1 %% Function to convert recorded voltage to pressure
2 % Sverre Kongsro Finstad
3
4 function [dBPressure,linearPressureAmplitude,linearPressurePtoP] =
    V_to_Pa(freq,peakToPeakAcData,Mv_f,Mv)
5 %freq = frequency vector of recorded signal
6 %peakToPeakAcData = Matrix of peak-to-peak voltages, calulated
    using
7 %findPeakToPeak_FFT_k.m
8
9 %Mv_f = frequency vector of receiver sensitivity
10 %Mv = receiver sensitivity vector, in dB re 1 V/Pa
11
12 %interpolates Mv for the signal frequencies used
13 MvdB = interp1(Mv_f,Mv, freq);
14 pp_FFTdB = 20*log10(peakToPeakAcData); %converts voltage to dB re 1
    V
15 pp_FFTdB = pp_FFTdB - 60; %subtracts amplifier gain
16 pp_FFTdB = pp_FFTdB - MvdB; %converts voltages to Pascal
17 dBPressure = pp_FFTdB;
18 linearPressurePtoP = 10.^(pp_FFTdB/20); %stores linear peak to peak
    pressure
19 linearPressureAmplitude = linearPressurePtoP./2; %stores peak
    pressure
20 %avgElPtoP = mean(peakToPeakElData,'all');
```

A.4 FEMP construction files

A.4.1 init_const_project.m

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialization of constants for FEMP
% Part of FEMP (Finite Element Modeling of Piezoelectric structures)
% Programmed by Jan Kocbach (jan@kocbach.net)
% (C) 2000 Jan Kocbach. This file is free software; you can redistribute
% it and/or modify it only under the the terms of the GNU GENERAL PUBLIC
% LICENSE which should be included along with this file.
% (C) 2000-2010 Christian Michelsen Research AS
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INIT_CONST_PROJECT: Initialize constants - project specifix
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Make a copy of this file in your project folder to
% make project specific definitions

%'piezodiskfront1',

commands = [commands,'piezodiskfront2','piezodiskfront5','smalltank','smalltank2','piezodiskfluidtest'];
```

A.4.2 read_inn_project.m

```
function [read]=read_inn_project(read,commands);
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Read .inn-file. Note that this function calls a project specific
% read_inn_project.m which should be in the working directory
%
% Part of FEMP (Finite Element Modeling of Piezoelectric structures)
% Programmed by Jan Kocbach (jan@kocbach.net)
% (C) 2000 Jan Kocbach. This file is free software; you can redistribute
```

```
% it and/or modify it only under the the terms of the GNU GENERAL PUBLIC
% LICENSE which should be included along with this file.
% (C) 2000-2010 Christian Michelsen Research AS
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Put a file read_inn_project.m in your project directory to define local
% FEMP input commands. Also include init_const_project.m in this directory
% and define the commands there.
global glob;
read=read;


%% smalltank
if ~isempty(read.smalltank)

 %rad_source = read.smalltank(1,1,:);
    %t_source = read.smalltank(1,2,:);
    %rad_mec = read.smalltank(1,3,:);
    %t_mec = read.smalltank(1,4,:);
    %rad_wa = read.smalltank(1,5,:);
    %t1_wa = read.smalltank(1,6,:);
    %t2_wa = read.smalltank(1,7,:);
    %elsrc_r=read.smalltank(1,8,:);
    %elsrc_t=read.smalltank(1,9,:);
    %elmec_t= read.smalltank(1,10,:);
    %el_fl = read.smalltank(1,11,:);
    %pmlthick = read.smalltank(1,12,:);
    %q_dampdist = read.smalltank(1,13,:);



rad_source = read.smalltank(1,1,:);
t_source = read.smalltank(1,2,:);
rad_buff = read.smalltank(1,3,:);
t1_buff = read.smalltank(1,4,:);
q_DampDist_z = read.smalltank(1,5,:);
q_DampDist_r = read.smalltank(1,6,:);
el_source = read.smalltank(1,7,:);
el_buff = read.smalltank(1,8,:);
back  = read.smalltank(1,9,:);


          matr_source = 77;
%pzt-5a = 40%pom = 991; wolf = 712; wolf_spiros = 704 wwater:10000,water_egen:1000; steel: 100
%steel_egen:102
```

```
% pz27_egen_dampened = 78   % pz27_egen_undampened = 79
                matr_buff = 10000;
                matr_back = 704;

    for s=1:size(rad_source,3)
        read.points(:,:,s)= ...
        [1 0 -t_source;
        2 rad_source -t_source;
        3 0 0;
        4 rad_buff(s) 0;
        5 0 t1_buff(s);
        6 rad_buff(s) t1_buff(s);
        7 rad_buff 0;
        8 rad_buff t1_buff;
        9 0 -t_source-back;
        10 rad_source -t_source-back];


        read.areas(:,:,s)=...
        [1 3 4 6 5 el_buff el_buff 0 0];
%        2 4 7 8 6 el_buff el_buff 0 0];
%        3 9 10 2 1 el_buff el_buff 0 0];
%

%         [1 glob.globvariables.piezo matr_source(s);     % 40 er
    pzt 5a
        read.materials(:,:,s)=...
        [1 glob.globvariables.fluid matr_buff];          %piezo
           koblingslag
%       3 glob.globvariables.fluid matr_buff;
%        3 glob.globvariables.mechanic matr_back];
%

 optimal_f = (read.directharmonicanalysis(1)+read.
    directharmonicanalysis(3))/2;    %bruker midterste frekvens for
     aa %%frekvensuavhengiggjore pml dempingen. 1 forste 2 step 3
    siste.
%
       glob.pmlform=glob.globvariables.manyregions;   %pml kan
          legges hvor jeg vil i koordinatsystemet, the general %way
   glob.pmlregions = {1,2};     %antall tall gir antall pml bokser
% %    glob.pmlregions = {1};     %antall tall gir antall pml
    bokser
     glob.pmlregions{1} = [1,rad_buff-q_DampDist_r,rad_buff,-1,1];%
        pml box
%      glob.pmlregions{3} = [2,rad_buff-q_DampDist_r,1,0,t1_buff];%
    pml box
glob.pmlregions{2} = [2,-1,1,t1_buff-q_DampDist_z,t1_buff];%pml box
     +2 = +y retning
   glob.pmlsigma_star_vec = 0;    %hvilken demp funksjon som brukes
      . 0 = optimal. et annet siffer gir annen uviss demp funksjon
```

```
94      glob.pmlomega_vec = 2*pi*optimal_f;             %frekvens som blir
            brukt
95  %     glob.pmlfluid = 1;      %deklarerer at pml blir brukt i
        simuleringen
96
97  % glob.lossvariation = 1;           %deklarerer at losss qm blir
        brukt i simuleringen
98  % glob.lossvarvecs{s} = [1,rad_buff, rad_buff+q_DampDistAddOn_rad,
        0,t1_buff,0.001,q_buff];
99  % glob.lossvarvecs{s} = [1,rad_buff, rad_buff+q_DampDistAddOn_rad,
        t1_buff,t1_buff+t2_buff,0.001,q_buff];
100 % glob.lossvarvecs{s} = [2,0,rad_buff, t1_buff+t2_buff-
        q_DampDistAddOn,t1_buff+t2_buff,0.001,q_buff];
101
102
103 % read.restraints(:,:,s)=[-1 1 -t_source-1e-9 -t_source+1e-9 glob.
        free.ep 1];
104 %       read.dof(:,:,s) = [-1 1 -1e-9 +1e-9 glob.free.ep;
105 ert = 50/(2*pi*500e3)^2 * 1500;
106       read.restraints(:,:,s)=[0 15e-3 -1e-9 +1e-9 glob.free.vp ert];
107       read.dof(:,:,s) = [15e-3 1 -1e-9 +1e-9 glob.free.dz;
108 %       read.restraints(:,:,s)=[-1 1 -t_source-1e-9 -t_source+1e-9
        glob.free.ep 1];
109 % %       read.restraints(:,:,s)=[0 rad_source -1e-9 +1e-9 glob.free
        .dz 1e-9];
110             -1 1 t1_buff-1e-9 t1_buff+1e-9 glob.free.vp;
111                     rad_buff-1e-9 rad_buff+1e-9 0 t1_buff glob.
                            free.vp];
112 % %                   -1 1 -1e-9 +1e-9 glob.free.ep];
113 %     read.dof(:,:,s) = [-1 1 -t_source-1e-9 -t_source+1e-9 glob.
        free.ep];
114 %     read.dof(:,:,s) = [rad_source 1  -1e-9 +1e-9 glob.free.dz];
115
116      end
117 end
118
119 %% piezodiskfluid_egen
120         if ~isempty(read.piezodiskfluidtest)
121             %KDL 30.06.04 has replaced all r{s},t{s},elr{s}...with
122             %r(s),t(s),elr(s)... for this geometry.
123             read.points=[]; read.areas=[]; read.materials=[]; read.
                    dof=[]; read.restraints=[];
124
125
126 %           r=read.piezodiskfluidtest(1,1,:);
127 %             t=read.piezodiskfluidtest(1,2,:);
128 %             elr=read.piezodiskfluidtest(1,3,:);
129 %             elt=read.piezodiskfluidtest(1,4,:);
130 %             matnum=read.piezodiskfluidtest(1,5,:);
131 %             rfluid=read.piezodiskfluidtest(1,6,:);
132 %             %rinffluid=rfluid*2;
```

```
133 %             %rinffluid=read.piezodiskfluid(1,7,:);
134 %              elfluid=read.piezodiskfluidtest(1,7,:);
135 %              matnumfluid=read.piezodiskfluidtest(1,8,:);
136 %              theta=read.piezodiskfluidtest(1,9,:);
137
138
139               r=10.120e-3;
140               t=2.0407e-3;
141               elr=2;
142               elt=2;
143               matnum=77;
144               rfluid=0.04;
145               rinffluid=rfluid*2;
146               %rinffluid=read.piezodiskfluid(1,7,:);
147               elfluid=2;
148               matnumfluid=10101;
149               theta=1.3;
150
151
152
153 %             theta=pi/4;
154               for s=1:size(r,3)
155 %               theta=atan(t(s)/2/r(s));
156 %                 rfluidtemp=0+rfluid(s);
157 %                 rinffluid=rfluidtemp*2;
158                   %rinffluid=rfluid*2;
159                   read.points(:,:,s)=[1 0 -t(s)/2;
160                   2 r(s) -t(s)/2;
161                   3 0 t(s)/2;
162                   4 r(s) t(s)/2;
163                   5 0 rfluid(s);
164                   6 rfluid(s)*sin(theta(s)) rfluid(s)*cos(theta(s));
165                   7 rfluid(s)*sin(theta(s)) -rfluid(s)*cos(theta(s));
166                   8 0 -rfluid(s);
167                   9 0 rinffluid;
168                   10 rinffluid*sin(theta(s)) rinffluid*cos(theta(s));
169                   11 rinffluid*sin(theta(s)) -rinffluid*cos(theta(s))
                          ;
170                   12 0 -rinffluid;
171                   13 0 0];
172
173
174 % 02.05.2011 Magne Aanes:
175                   % Problems with meshing when not elements per
                          wavelength!
176
177                   % This should be changed if direct method!!
178                   read.areas(:,:,s)=[1,1,2,4,3,elr(s),elt(s),0,0;
179                   2,3,4,6,5,elfluid(s),elfluid(s),0,13; % set to 2 -
                          let piezo
```

```
180                 2,4,2,7,6,elfluid(s),elfluid(s),0,13; % decide at
                        present!
181                 2,2,1,8,7,elfluid(s),elfluid(s),0,13;
182                 3,5,6,10,9,1,1,13,13;  % First should maybe be
                        elfluid(s)??
183                 3,6,7,11,10,1,1,13,13;
184                 3,7,8,12,11,1,1,13,13];
185
186
187
188
189
190                 read.materials(:,:,s)=[1 glob.globvariables.piezo
                        matnum(s);
191                 2 glob.globvariables.fluid matnumfluid(s);
192                 3 glob.globvariables.infinitefluid matnumfluid(s)];
193
194                 read.dof(:,:,s)=[-1 1 t(s)/2-1e-9 t(s)/2+1e-9 glob.
                        free.ep];
195                 %read.dof(:,:,s)=[-1 1 t(s)/2-1e-9 t(s)/2+1e-9 glob
                        .free.ep;
196                 %                   4.9999e-3,5.0001e-3,-1,1,glob.
                        free.vp];
197                 read.restraints(:,:,s)=[-1 1 -t(s)/2-1e-9 -t(s)/2+1
                        e-9 glob.free.ep 1];
198             end
199         end
200
201     %       rad_wa-1e-9 rad_wa+1e-9 t1_wa+t_mec t1_wa+t_mec+t2_wa
            glob.free.vp];
202
203     %-1 1 -t_source-bak_vann-1e-9 -t_source-bak_vann+1e-9 glob.free
            .vp;
204     %sette pml verdier bak TD
205     %read.constraints(:,:,s)=[-1 1 -1e-9 1e-9 glob.free.ep];     ta
            ut
206     %spenning paa mottkertransducer
207
208
209
210
211 % der pml slutter skal hast pot vaere null
```

A.4.3 test3.inn

```
1 set
2 rad_source,10.120e-3
3 t_source,2.0407e-3
4 rad_buff,30e-3
5 t1_buff,50e-3
6 q_DampDist_z,20e-3;
```

```
q_DampDist_r,10e-3;
el_source,5
el_buff,4
back,30e-3;
end

materialfile
1
end

meshingtype
elementsperwavelength,120e3
end

viewmesh
1
end

# The order of the finite elements is 2 - i.e. 8 node isoparametric
    elements are applied
order
2
end

infiniteorder
10
end

#smalltank
piezodiskfluidtest
rad_source,t_source,rad_buff,t1_buff,q_DampDist_z,q_DampDist_r,
    el_source,el_buff,back
end

#directharmonicanalysis
#0e3,2e3,300e3,no_loss
#end

directharmonicanalysis
300000,250,300000,complex_loss
end

admittance
0,0,0
end

sensitivity
0,0,0,1
end

nearfieldpressure
```

```
0,0,0,-1,1,-1,1
end

directivity
0,0,0,1000
end

# Calculate far-field pressure for the frequencies used in the time-harmonic
# analysis. Calculate out to 3 times the distance at which the infinite
# elements are applied (10*7.0 mm= 70.0 mm), with 20 divisions per 7.0 mm.
farfieldpressure
0,0,0,5,20
end

onaxispressure
0,0,0,10,20
end

save
admittance,admittance_f,nearfieldpressure,nearfieldpressure_z,nearfieldpressure_r,nearfieldpressure_f,farfieldpressure,farfieldpressure_f,farfieldpressure_r,farfieldpressure_z,onaxispressure,onaxispressure_f,onaxispressure_r,directivity,directivity_f,directivity_theta
end


#save
#admittance,admittance_f
#end

#save('test_res.mat','result','-v7.3');
```

Appendix B

Pulse Examples

Here, examples of received pulses at different angles and frequencies are presented. All pulses are from directivity measurements at r = 0.2 m.

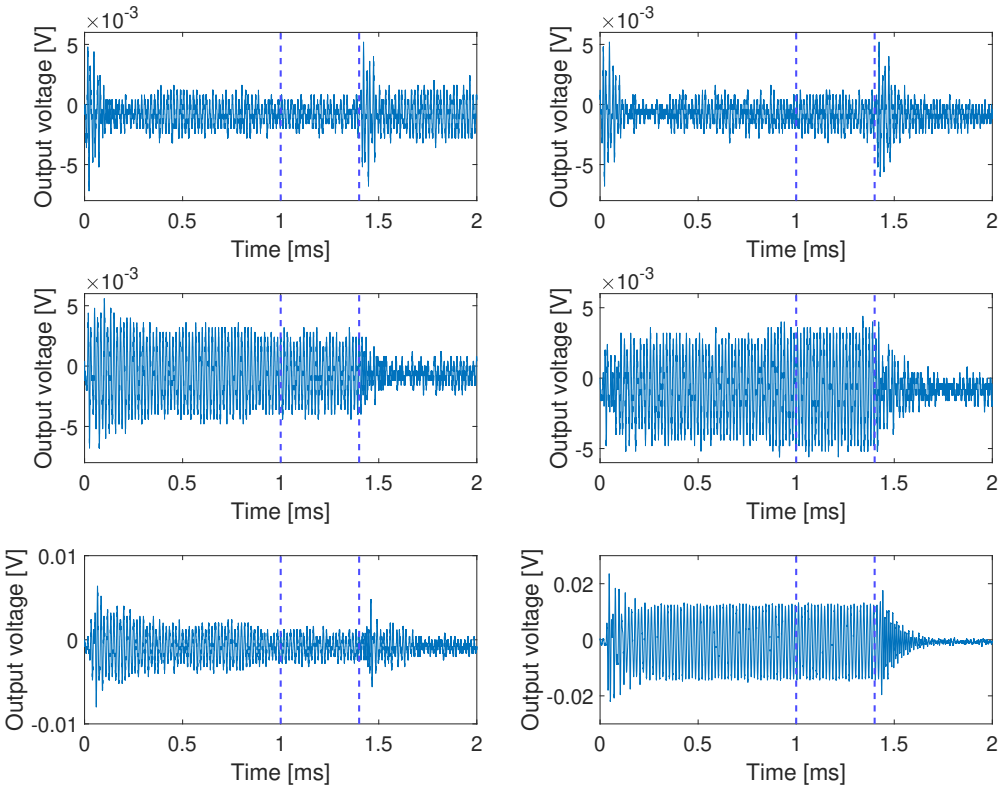


FIGURE B.1: Examples of pulses received at 60 kHz. From left to right, top to bottom: [90 degrees, 80 degrees, 60 degrees, 40 degrees, 20 degrees, 0 degrees.]

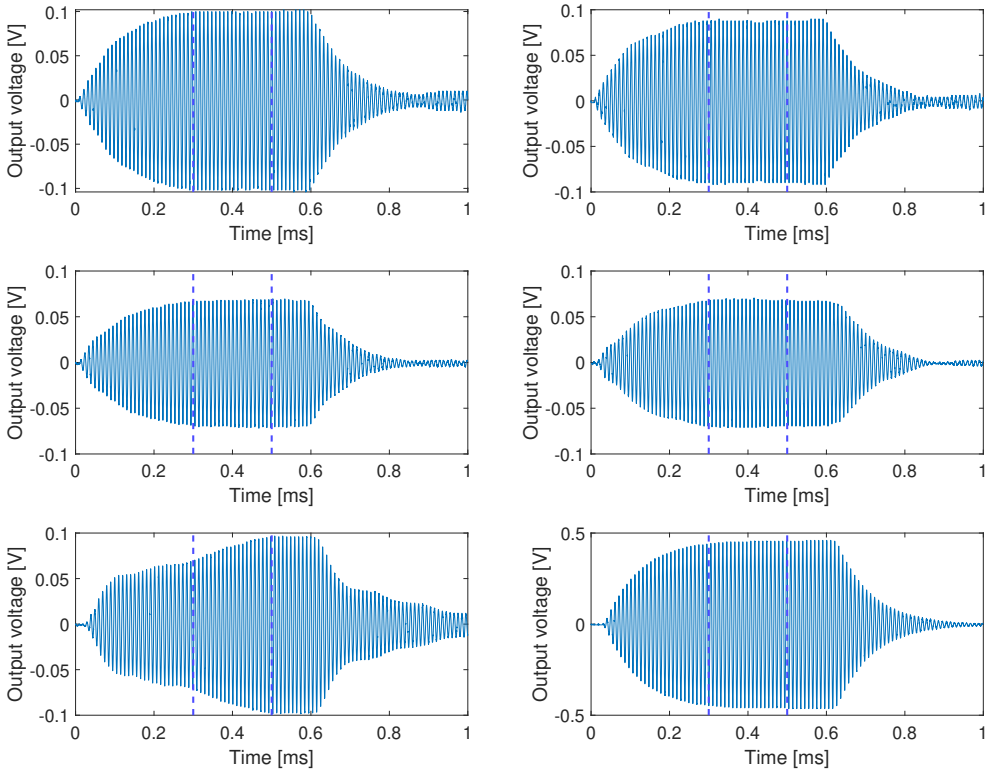


FIGURE B.2: Examples of pulses received at 98.14 kHz. From left to right, top to bottom: [90 degrees, 80 degrees, 60 degrees, 40 degrees, 20 degrees, 0 degrees.]

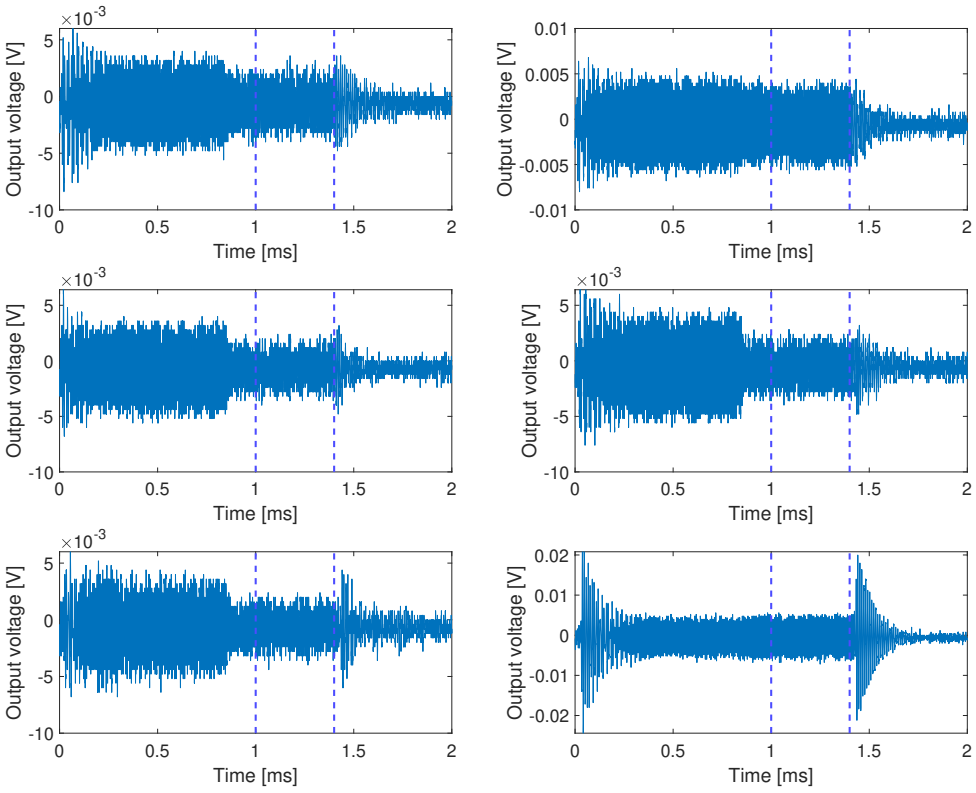


FIGURE B.3: Examples of pulses received at 140 kHz. From left to right, top to bottom: [90 degrees, 80 degrees, 60 degrees, 40 degrees, 20 degrees, 0 degrees.]

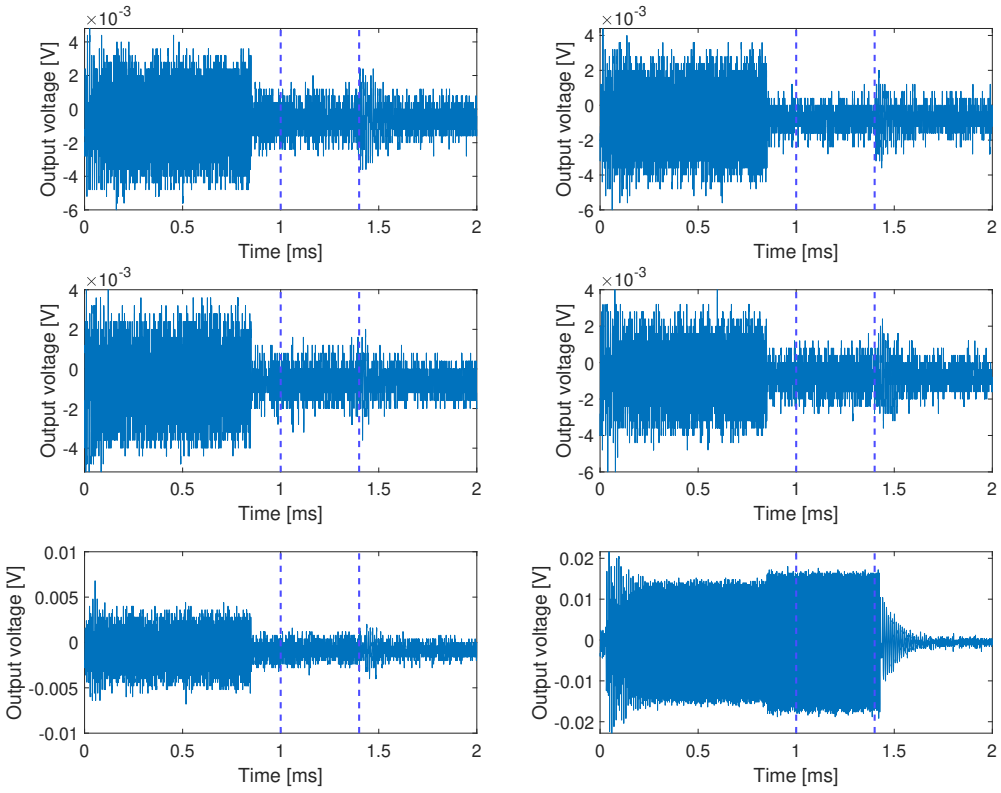


FIGURE B.4: Examples of pulses received at 180 kHz. From left to right, top to bottom: [90 degrees, 80 degrees, 60 degrees, 40 degrees, 20 degrees, 0 degrees.]

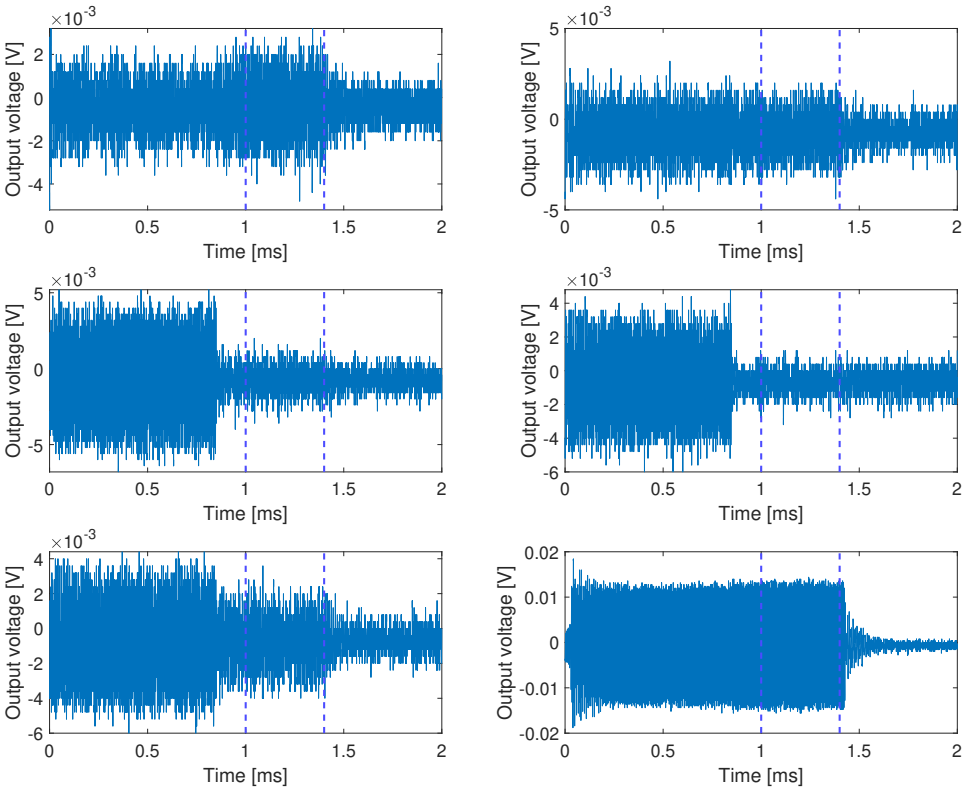


FIGURE B.5: Examples of pulses received at 220 kHz. From left to right, top to bottom: [90 degrees, 80 degrees, 60 degrees, 40 degrees, 20 degrees, 0 degrees.]

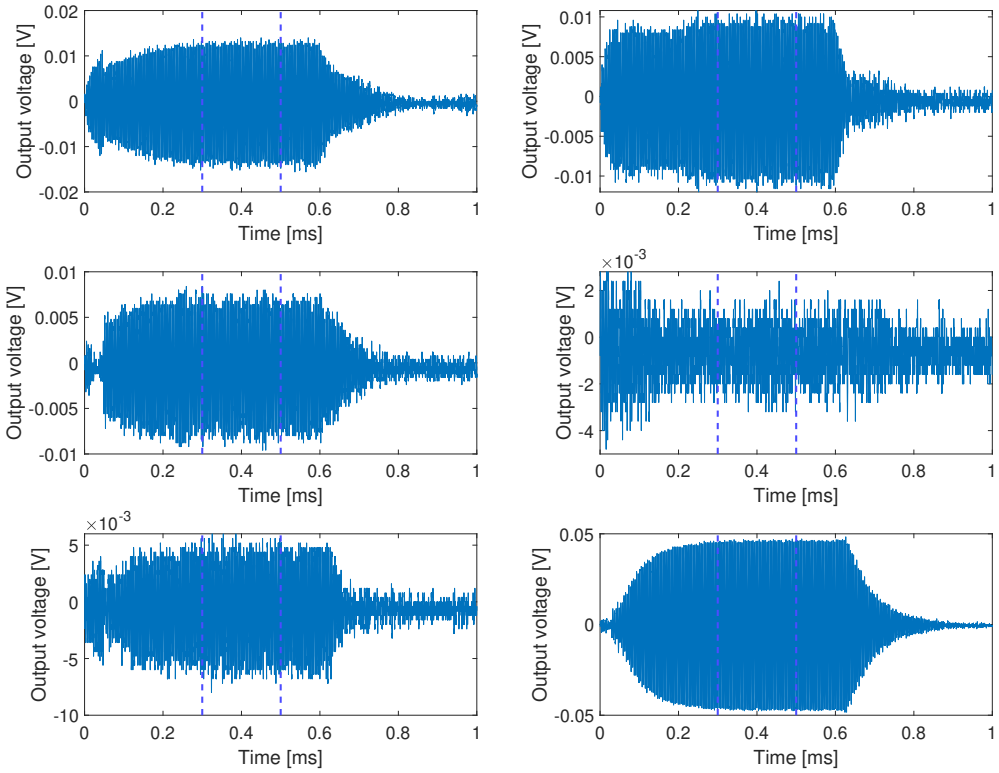


FIGURE B.6: Examples of pulses received at 249.52 kHz. From left to right, top to bottom: [90 degrees, 80 degrees, 60 degrees, 40 degrees, 20 degrees, 0 degrees.]

Bibliography

[1] E. Storheim, "Diffraction effects in the ultrasonic field of transmitting and receiving circular piezoceramic disks in radial mode vibration," *PhD, Department of Physics and Technology, University of Bergen, Norway*, 2015.

[2] E. Mosland, "Reciprocity calibration method for ultrasonic piezoelectric transducers in air - master thesis in acoustics," *Master's thesis, Department of Physics and Technology, University of Bergen, Norway*, 2013.

[3] R. Hauge, "Finite element modeling of ultrasound measurement systems for gas. comparison with experiments in air.," *Master's thesis, Department of Physics and Technology, University of Bergen, Norway*, 2013.

[4] R. Grindheim, "Ultrasonic measurement system for gas. experimental and theoretical characterization using piezoelectric elements at radial mode vibration in air," *Master's thesis, Department of Physics and Technology, University of Bergen, Norway*, 2019.

[5] "Regulations relating to measurement of petroleum for fiscal purposes and for calculation of co2-tax," Norwegian Petrolium Directorate, Stavanger, Norway, 2001.

[6] "Aga report no. 9. measurement of gas by multipath ultrasonic meters," American Gas Association, Transmission Measurement Committee, USA, 2017.

[7] "Iso 17089-1:2019 measurement of fluid flow in closed conduits — ultrasonic meters for gas — part 1: Meters for custody transfer and allocation measurement," International Organization for Standardization, Geneva, Switzerland, 2019.

[8] P. Lunde, K. E. Frøysa, R. A. Kippersrud, and M. Vestrheim, "Transient diffraction effects in ultrasonic meters for volumetric, mass and energy flow measurement of natural gas," Christian Michelsens Research, Norway and University of Bergen, Norway, 2002.

[9] P. Lunde and K. E. Frøysa, "Density and calorific value measurement in natural gas using ultrasonic flow meters," Christian Michelsens Research, Norway, 2005.

[10] L. Lynnworth and Y. Liu, "Ultrasonic flowmeters: half-century progress report, 1955-2005," Lynnworth Technical Services, USA, and GE Sensing, China, 2005.

[11] J. P. Bentley, *Principles of measurement systems*. Pearson education, 2005.

[12] M. Vestrheim, "Phys 272 - akustiske målesystem," *Lecture notes, Department of Physics and Technology, University of Bergen, Norway*, 2007.

[13] L. Schmerr and J.-S. Song, *Ultrasonic nondestructive evaluation systems*. Springer, 2007.

[14] M. Vestrheim, "Phys 272 - akustiske transdusere," *Lecture notes, Department of Physics and Technology, University of Bergen, Norway*, 2007.

[15] L. L. Thompson, "A review of finite-element methods for time-harmonic acoustics," *The Journal of the Acoustical Society of America*, vol. 119, no. 3, pp. 1315–1330, 2006.

[16] E. P. Papadakis, "Ultrasonic transducer evaluation in five" domains": time, space, frequency, surface motion, and theory," in *1977 Ultrasonics Symposium*, pp. 104–112, IEEE, 1977.

[17] A. Lygre, M. Vestrheim, P. Lunde, and V. Berge, "Numerical simulation of ultrasonic flowmeters," pp. 196–201, 07 1987.

[18] D. Collie and M. Player, "Extended computer method for predicting the transient response of ultrasonic ndt probes," *Ultrasonics*, vol. 27, no. 3, pp. 141–149, 1989.

[19] S. J. Sanabria, T. Marhenke, R. Furrer, and J. Neuenschwander, "Calculation of volumetric sound field of pulsed air-coupled ultrasound transducers based on single-plane measurements," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 65, no. 1, pp. 72–84, 2018.

[20] G. Benny, G. Hayward, and R. Chapman, "Beam profile measurements and simulations for ultrasonic transducers operating in air," *The Journal of the Acoustical Society of America*, vol. 107, no. 4, pp. 2089–2100, 2000.

[21] A. G. Bashford, D. Schindel, Hutchins, and Wright, "Field characterization of an air-coupled micromachined ultrasonic capacitance transducer," *The Journal of the Acoustical Society of America*, vol. 101, 1997.

[22] V. K. Chillara, C. Pantea, and D. N. Sinha, "Low-frequency ultrasonic bessel-like collimated beam generation from radial modes of piezoelectric transducers," *Applied Physics Letters*, vol. 110, no. 6, p. 064101, 2017.

[23] V. K. Chillara, E. S. Davis, C. Pantea, and D. N. Sinha, "Ultrasonic bessel beam generation from radial modes of piezoelectric discs," *Ultrasonics*, vol. 96, pp. 140–148, 2019.

[24] J. Kocbach, "Finite element modeling of ultrasonic piezoelectric transducers," *Department of Physics and Technology, University of Bergen*, 2000.

[25] A. A. Søvik, "Ultrasonic measurements systems for gas. finite element modeling compared with measurements in air," *Master's thesis, Department of Physics and Technology, University of Bergen, Norway*, 2015.

[26] H. Kunkel, S. Locke, and B. Pikeroen, "Finite-element analysis of vibrational modes in piezoelectric ceramic disks," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 37, no. 4, pp. 316–328, 1990.

[27] L. Kinsler, A. Frey, A. Coppens, and J. Sanders, *Fundamentals of Acoustics, Fourth edition*. Hoboken, USA: John Wiley and Sons, 1999.

[28] I. Getman and S. Lopatin, "Matching of series and parallel resonance frequencies for ultrasonic piezoelectric transducers," vol. 2, pp. 713–715 vol. 2, 2000.

[29] R. Øyerhamn, E. N. Mosland, E. Storheim, P. Lunde, and M. Vestrheim, "Finite element modeling of ultrasound measurement systems for gas. comparison with experiments in air," *The Journal of the Acoustical Society of America*, vol. 144, no. 4, pp. 2613–2625, 2018.

[30] "Methods for calculation of the absorption of sound by the atmosphere.," Acoustical Society of America, 1994.

[31] *Operation and service manual. Model 4192A LF Impedance analyzer*, Hewett Packard, 1982, Tokyo, Japan.

[32] *Data Sheet - Mixed Signal Oscilloscopes - MSO3000 Series, DPO3000 Series*, Tetronix, 2012.

[33] *Data Sheet - Agilent 33220A 20 MHz Function/Arbitrary Waveform Generator*, Agilent.

[34] *Product data - Wide Range Measuring Amplifiers — Types 2610 and 2636*, Brüel & Kjær, 1985, Nærum, Denmark.

[35] *Model 3940/3944 Programmable (IEEE-488) Filter - Operating and Maintenance Manual*, Krohn-Hite Corporation, 2014.

[36] *Microphone unit Type 4138-A-015 Calibration chart*, Bruel & Kjær, 2012, Nærum, Denmark.

[37] *Product Data - Pistonphone Type 4228*. https://www.bksv.com/en/transducers/acoustic/calibrators/pistonphone (19/7-21), Bruel & Kjær, 1996, Nærum, Denmark.

[38] *F250 MkII - Precision thermometer*. https://www.instrumart.com/products/20291/asl-f250-mkii-precision-thermometer (19/7-21), ASA, Milton Keyes, England.

[39] *High-Precision Linear Stage - M-511, M-521, M-531*, Physik Instrumente, Kahlsruhe, Germany, 2018.

[40] *M-5x1 Series Linear Positioning Stages, MP 33E User Manual*, Physik Instrumente, Kahlsruhe, Germany, 2004.

[41] *High-speed, High-accuracy CCD Laser Displacement Sensor, LK-G Series User Manual*, Keyence, Osaka, Japan, 2016.

[42] *LK-GD500 Separate type Display panel Data sheet*, Keyence, Osaka, Japan, 2016.

[43] *LK-G32 Sensor head: High accuracy, Small spot, Data sheet*, Keyence, Osaka, Japan, 2016.

[44] "Type pz27 data sheet," Ferroperm/Meggitt, 2018, Kvistgaard, Denmark.

[45] A. Rohatgi, "Webplotdigitizer: Version 4.4," 2020. https://automeris.io/WebPlotDigitizer, 15/4-21.

[46] R. W. Schafer, "What is a savitzky-golay filter? [lecture notes]," *IEEE Signal processing magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[47] *Digital Motor Controllers and Drivers*, Physik Instrumente, Kahlsruhe, Germany, 2021.

[48] M. Aanes, *Interaction of piezoelectric transducer excited ultrasonic pulsed beams with a fluid-embedded viscoelastic plate*. PhD thesis, Ph. D. thesis, Department of Physics and Technology, University of Bergen . . . , 2014.

[49] "High quality components and materials for the electronic industry," Ferroperm, 2012.

[50] K. D. Lohne, "Undersøkelse og utnyttelse av svingemoder i ultralyd transduserkonstruksjoner," *Department of Physics and Technology, University of Bergen*, 2005.

[51] V. Knappskog, "Radiellmode svingninger i piezoelektriske ultralydstransdusere for luft. målinger og endelig element analyse," *Department of Physics and Technology, University of Bergen*, 2007.

[52] J. Taylor, *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. ASMSU/Spartans.4.Spartans Textbook, University Science Books, 1997.