Norwegian University of Science and Technology

NTNU

Department of Electronic Systems
Faculty of Information Technology and Electrical Engineering

Master Thesis

Developing a method of absorbtion measurement for cylinder geometries with applications to tree bark

Liv Astrid Nygaard

Supervisor Guillaume Dutilleux

spring 2020

Liv Astrid Nygaard
Developing a method of absorbtion measurement for cylinder geometries with applications to tree bark
Master Thesis, spring 2020
Supervisor: Guillaume Dutilleux

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

Abstract

This report is a result of a master project performed at NTNU, Norwegian University of Science and Technology. The aim of this thesis is to build a method of measurement for absorbers that only exist in cylindrical formats, and for which the common standards for measurements might not apply. With the imagined application being tree bark, there is also a point in making these measurements *in situ*, as working larger trees into a lab environment is imagined a tiresome and inconvenient task.

The choice of method is by two-microphone measurements and then the usage of a simulation of cylindrical waves onto a cylinder from [Mec08], to match transfer functions from the measurement with a map of such from the simulation. This is done to determine the impedance, and from there on, find reflection and absorption factor. The microphone configuration chosen after a study of the conformal maps, and with general regards taken to trees not being entirely cylindrical and microphones interfering with each other, ended up becoming 3 and 8 cm from the cylinder, all on axis between source and cylinder. The measurements are done on five different cases, a smooth concrete column, the same column with an external absorber wrapped around, a tree with bark, the same tree with the same absorber, and finally a free field case.

Comparing the transfer functions of the simulations and the measurements regretfully gives non-compliant results, as the transfer functions does not even remotely fall on the same space in simulation and measurements. This renders the resulting impedance, and thus the derived reflection and absorption coefficient, intelligible.

Acknowledgement

There are many considerations that has to be taken during a longer thesis project, and even more when a worldwide pandemic is affecting day-to-day life. I would therefore like to thank firstly my supervisor at NTNU, Guillaume Dutilleux, for his continuous support and patience through repeated questions and bad internet connections, and not to mention keeping me on track throughout the half-year process it ha been to write this thesis.

Thanks also goes to my family, for re-emitting me into the household after the outbreak of Covid-19 and giving me both a physical space and the necessary head space to write, read, calculate, make calls, borrow vehicles back to Trondheim for measurements and keep me socially engaged during lock down of these resources that are usually provided by my school, my co-students and friends.

Contents

List of Figures

Introduction

1

1.1 Motivation

An important factor to take into account in acoustics in general is the absorption of a material represented by the material's impedance, complex reflection coefficient and, derived directly from this, the absorption coefficient, α. In most cases the subject of measurement exists in flat samples that are measured using standardized method such as ISO 354 [03], which has a minimum of 10m^2 sample laid on a surface of a reverberation room and, or ISO 10534 [98], where absorption is calculated from the transfer function obtained in a standing wave tube. Both these methods of measurements are done in lab environment, and not *in situ*. With the exception of ISO 354, the general requirement for absorption measurements (*in situ* or not) is that the sample has to be flat, which generally does not cause a problem. Most absorbers come in flat samples. However, for the samples that are curved, using methods that are made for flat samples may not yield correct results.

This problem becomes relevant when measuring bark on trees. Although bark is fairly rigid and might not have great absorption properties, the possibility of using trees and shrubberies as sound insulators for road traffic noise is worth exploring. In the cases where extensive measurements of tree bark has been done, the use of impedance tube measurements is usually preferred. Both the study by Li &Van Renterghem [LTB19] and the one by Reethof & McDaniel, [RMH77], does not pay any regard to the curvature of the bark as they do these measurements in standing wave tubes. Using ISO 354 is not recommended either, as spreading 10m^2 of bark flat on a floor or wall of a reverberation room in order to fulfill the requirements in ISO 354 is an unnecessary lengthy process, and would require careful dissection of at least an entire tree.

There exists other measurements done on trees and shrubberies, such as the study done by Price, Attenborough and Heap [PAH88] on the attenuation of sound through different types of woodland, where general shrubbery is explored, but not the tree bark individually. Another study, done by Van Renterghem [Ren15], explores the affect tree belts of different configurations has on road traffic noise, however with more emphasis on the arranging of the poles than the individual tree trunk's absorption properties. Preferably, a method on a singular induvidual which does not require cutting down and moving trees into a lab environment should exist, making the practical circumstances around the measurements simpler.

1.1.1 Objective and limitation

This project has the objective of carrying out accurate measurements of the absorption factor on cylindrical geometries. The point of this is to make it possible for measurements to be carried out *in situ* on cylindrically shaped absorbers, for example trees, such that the curvature of the trunk is taken into account and does not affect the calculation of absorption.

1.2 Report structure

This report will continue in the next chapter by explaining the general theoretical concepts behind the scattering of a cylinder. It also describes the use of two-microphone measurements and transfer function to find the absorption properties of a material. More theory shorty explained in this chapter is conformal mapping, convergence studies and time windowing.

The third chapter is separated into three sections, where the first explains the creation of the simulated model in Matlab, using the modelling in [Mec08]. After this, its test and optimization by using a convergence study is presented. The second section of this chapter goes into how, where and why the actual measurements are performed, on two different cylinders, a beech tree and a concrete column. Both these are situated at NTNU Gløshaugen, however at different locations on the campus. The section after in this chapter explains the post-processing, where theoretical model is joined with physical measurement to find an expression for the impedance Z_a, the reflection coefficient, r, and the absorption factor α.

After model and measurements are explained, the report goes on to present the results of these measurements and findings of the model. It explains that while the model and measurements by themselves looks feasible each by their own, the calculations break down when joining the two, for reasons speculated on in the discussion section.

And finally, in chapter 5, the report is concluded with the unsatisfactory results and the possible further work is presented.

Theory

2

2.1 Cylinder Scattering

The models for scattering from a cylinder is discussed in several pieces of literature. Section 8.8 in the book "Acoustics – An Introduction to Its Physical Principles and Applications" by Alan Pierce, [Pie89], discusses at length about the reflection of sound waves from a rigid cylinder. The scattering pattern of a rigid cylinder is also discussed by Li and Ueda in their article "Sound scattering of a plane wave obliquely incident on a cylinder", [LU89]. However, ideally for a model for measuring the absorption one would need soft boundaries, not rigid. Another problem these two models have is that the incoming wave is not a spherical one, but plane. Usually one has access to the former when doing measurements.

The article by Swearingen and Swanson [SS12], seems to have covered both the cylinder scattering and the impedance surface in its model for transmission and reflection, and it also works with a point source. However, the model uses increasingly complicated integrals, which takes computing power and a long time to solve. In addition, this method includes ground reflections, which adds remarkably to the computing power needed and is not particularly necessary for future measurements. It is easier to cancel out this reflection by using absorbers than trying to figure out the impedance of the ground beforehand for each *in situ* case.

Fridolin P. Mechel's book "Formulas of Acoustics" [Mec08, p. 185-201] has another approach. It contains a model for the sound field at a certain point, P, scattered from a cylinder with a given impedance when an incoming cylindrical wave from a source point Q scatters on the object. The mathematical expression for this wave at the point P is described in equation (2.1).

$$p(r,\theta) = p_q(r') + p_s(r,\theta) \tag{2.1}$$

Here, r is the cylindrical coordinate to the receiver point, and r' is the distance between the source and the receiver. θ is the corresponding angle off the x-axis in the coordinate system. p_q is the incoming wave from the source in point Q and p_s is the scattered wave from the cylinder. The cylinder radius is denoted a. See figure 2.1 for visualization.

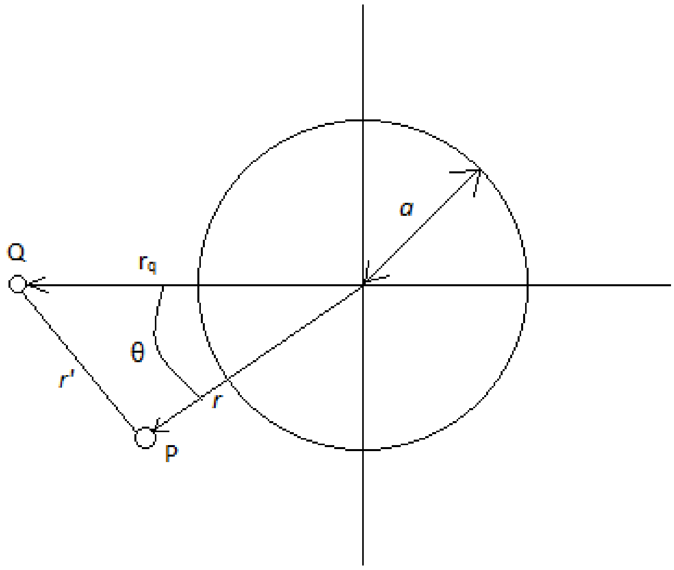


Fig. 2.1.: The cylinder with source and receiver position. Reconstructed from [Mec08, p. 199]

The incoming pressure field is expressed in two ways, based on whether r_q is smaller or bigger than r. For the first case, $r_q > r$, which is denoted (a), we get a formulation for the incoming wave as shown in equation (2.2). The second case, denoted (b), when $r_q < r$ gives the equation in (2.3).

$$p_{Q(a)}(r') = P_0 \cdot \sum_{m \geq 0} \delta_m \cdot J_m(k_0 r) \cdot H_m^{(2)}(k_0 r_q) \cdot \cos(m\theta) \tag{2.2}$$

$$p_{Q(b)}(r') = P_0 \cdot \sum_{m \geq 0} \delta_m \cdot J_m(k_0 r_q) \cdot H_m^{(2)}(k_0 r) \cdot \cos(m\theta) \tag{2.3}$$

Here, P_0 is the amplitude of the wave, J_m and H_m is the Bessel and Hankel functions of the m'th kind. k_0 is the wavenumber, and δ_m is defined as seen below in (2.4).

$$\delta_m = \begin{cases} 1; & m = 0, \\ 2; & m > 1. \end{cases} \tag{2.4}$$

The scattered field is then formulated as:

$$p_s(r, \theta) = -P_0 \cdot \sum_{m \geq 0} \delta_m \cdot c_m \cdot H_m^{(2)}(k_0 r_q) \cdot H_m^{(2)}(k_0 r) \cos(m\theta) \tag{2.5}$$

Where

$$c_m = \frac{\left(G + \frac{m}{k_0 a}\right) \cdot J_m(k_0 a) - j \cdot J_{m+1}(k_0 a)}{\left(G + \frac{m}{k_0 a}\right) \cdot H_m^{(2)}(k_0 a) - j \cdot H_{m+1}^{(2)}(k_0 a)} \tag{2.6}$$

G in this equation is tied to the surface impedance of the cylinder, Z_a, as $G = 1/Z_a$.

2.2 Transfer function method

A common way for determining the absorbing qualities of a material is by using the transfer function method with explores the ratio of pressure between two measurement points P1 and P2, its most famous usage is in the standard of standing wave tube measurement from ISO 10354 [98]. This ratio removes the need for measurements and calibrations with regard to sound power and makes post-processing calculations simpler. According to this standard, the two pressures measured on points P1 and P2, p_1 and p_2 respectively, are a combination of the incident and reflected waves as seen in equations (2.7)-(2.10).

$$p_1 = p_I(x_1) + p_R(x_1) \tag{2.7}$$
$$p_1 = \hat{p}_I e^{jk_0 x_1} + \hat{p}_R e^{-jk_0 x_1} \tag{2.8}$$
$$p_2 = p_I(x_2) + p_R(x_2) \tag{2.9}$$
$$p_2 = \hat{p}_I e^{jk_0 x_2} + \hat{p}_R e^{-jk_0 x_2} \tag{2.10}$$

Here, x_1 and x_2 denotes the positions of the measurement point for p_1 and p_2. p_I is the pressure from the incident wave, and p_R is the pressure from the reflected wave, their magnitudes denoted in $\hat{p}_I$ and $\hat{p}_R$.

The ratio of the pressure at these two points, is called the transfer function and is shown in equation (2.11).

$$H_{12} = \frac{p_2}{p_1} \tag{2.11}$$

The reflection factor, $r = \frac{p_R}{p_I}$, is directly tied to the absorbtion factor, α by the relation:

$$\alpha = 1 - |r|^2 \tag{2.12}$$

The reflection factor can be found if one knows the impedance Z_a of the material and Z_0, the specific acoustic impedance of air. These two are connected through the equation in equation (2.13).

$$\frac{Z_a}{Z_0} = \frac{1-r}{1+r} \tag{2.13}$$

2.2.1 Conformal mapping

Figure 2.2 illustrates a case of mapping from one plane to another, which allows the user to observe the scope a subset of values gives when applied a function. Explained by Dutilleux et al, [DVK01], utilizing mapping may be useful when attempting to find a good configuration of microphones while using the transfer function method to measure. When mapping a field on the complex plane of Z_a, to the complex transfer function plane one gets different sizes from different configurations. The largest area of the mapped transfer function, will give a configuration least susceptible to error when doing the measurement.

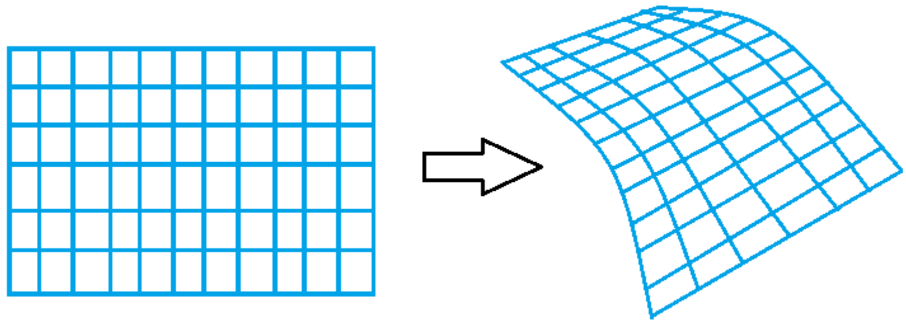


Fig. 2.2.: Illustration of a conformal map.

2.3 Convergence study

When dealing with the infinite sums as seen in section 2.1 in programming, one would have to restrain the maximum summation number, m_{hi}, to something smaller, preferably as little as possible to make the calculations go quickly. In order to make this happen a convergence study should be performed, determining on which iteration where another iteration is redundant, in other words where the sum converges.

2.4 Time windowing

Working *in situ* will in most cases give a resulting impulse response from not only the surface one wishes to explore, but also off the ground and surrounding surfaces. If one is to measure the absorption of the first reflection, and not the following, choosing a window of time which inly takes in the right information is crucial. Too long and the post-processing calculations might yield incorrect results. Time windowing is choosing such a window of time included in further calculations, disregarding the data points after. This choice should be after the reflections one wishes to explore, and before the following, such as the ground, walls or other surrounding surfaces. With less data points to consider, computing time in post-processing also goes down. Another option to rid oneself of these unwanted reflections is to dress the surfaces in absorbers, which makes effect the reflection has on the complete measurement negligible.

Method

3

In order to find the best physical setup, one has to explore the properties of cylinder scattering in a simulated model and then use the information to find a configuration that is most resilient of noise. Comparing simulated transfer functions with the one measured can also give an accurate measurement of the impedance, Z_a, and thus the reflection and absorbtion factor.

3.1 The model

All calculations are done in Matlab R2018b. Using the calculations from section 2.1, the model for a cylindrical wave interacting with a cylinder is implemented for a source at 5m distance, making the approximation from cylindrical wave to plane wave reasonable. This is because, as explained in section 2.1, the source in the actual measurement setup will be closer to a point source, but the literature is surprisingly devoid of mathematical models with incoming spherical waves interacting with cylinders with non-rigid surfaces and without a ground reflection. The function `cylWaveOnCyl` in appendix A.1 is set up to give a resulting pressure field at a specific position r, angle θ, frequency f, source position, r_q, cylinder radius a and impedance Z_a.

This model was tested for validity by comparing an example illustration of the pressure given from simulation on p. 200 in [Mec08], with an illustration made using the model made in Matlab, using the variables provided in the caption of the figure in the book, $G = 0.5 - j2$, $k_0 a = 2$, $k_0 r_q = 6.1$ and upper summation limit, $m_{hi} = 8$. The result of this validation research is presented in section 4.1.

3.1.1 Convergence study

A convergence study was carried out for the measured pressures at a fixed distance from the cylinder, as suggested in section 2.3. When running the simulations for higher ms and kas (k being the wavenumber, also previously denoted k_0), however, the Hankel functions will fail, as their overall value gets approximated towards zero. Matlab will then yield `NaN` as in equation (2.6), these Hankel functions are under the division line, and calculations will break down.

Figure 3.1 shows this problem. These are polar plots of the scattered field around a cylinder, measured at a fixed distance from the center, but at different angles. The five colors each represents a maxima of m for the sum, the smallest size in yellow, then increasing in size with the colors magenta, cyan, red and green, the last corresponding to the number in the title. In the lower middle polar plot, one can see the green line ($m_{hi} = n$=170) is missing entirely, and as one increases n even more, as seen in the last plot, the measurements disappear entirely. Here, the calculations return NaN for the pressure. This means that for the convergence study to work, it is convenient to find a function dependant on ka that gives an appropriate maximum m to sum over, that is large enough to converge but small enough to not break down by the use of Neumann functions. The making of these plots are done using the code in A.2.1.

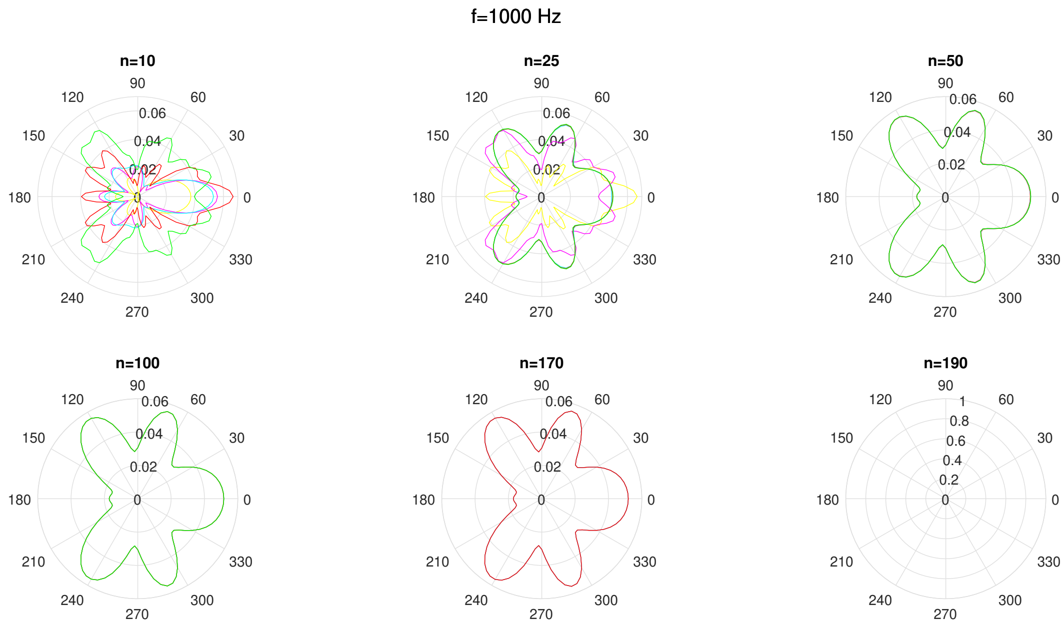


Fig. 3.1.: The polar plots of scattered fields around a cylinder for different maximas of the summation, $m_{hi} = n$. The five colors each represent different n, the smallest size in yellow, then increasing in size with the colors magenta, cyan, red and green, the last corresponding to the number in the title. The path towards convergence is seen on the top row and the entrance of NaN results is seen on the bottom row.

Sovling this means running simulations for different kas, and finding lower and upper limits to sum over, then trying to fit a curve in between these two, such that the m_{hi} for that specific ka is large enough for convergence, but still not so large that the calculations breaks down. The results of this convergence study is presented in section 4.2.

3.2 The measurement setup

The measurement uses two microphones and the transfer function method disclosed and reasoned for in section 2.2 to calculate the absorbtion.

Looking at the polar plots for the different frequencies, the shape is smooth and forward directed for smaller frequencies and then gets more side lobes as ka increases. The main lobe, however, stays for the most part the strongest and most consistent lobe, as seen in figure 3.2.

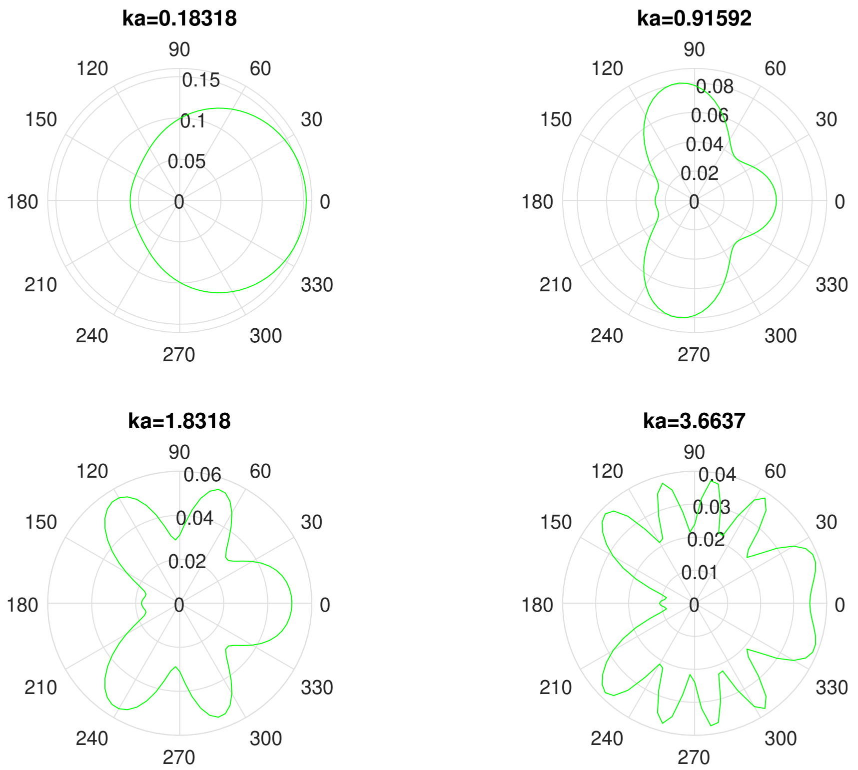


Fig. 3.2.: Polar plots of the scattering around a cylinder for different kas

As seen in the figure, the side lobes varies in placement with increasing ka. This means that for the best measurement for most frequencies, one needs to set the microphones at the angle $\theta = 0°$, which in the physical setup is along the axis from cylinder to loudspeaker.

In order to find the ideal position of the two microphones, conformal mapping as it was explained in section 2.2.1 is applied. Z_a is a complex number, ranging logarithmically from 0 to the step below ± 4000 rayl in the imaginary part and 4000 rayl in the real part. This range of values is portayed in the left side of figure 3.3. The values are chosen to cover a space of impedances that is likely to be included in an actual measurements, however not outrageously big, as calculations of pressure for two separate points, P1 and P2, for the full range of frequencies in third-octave band is already a quite time-consuming computation.

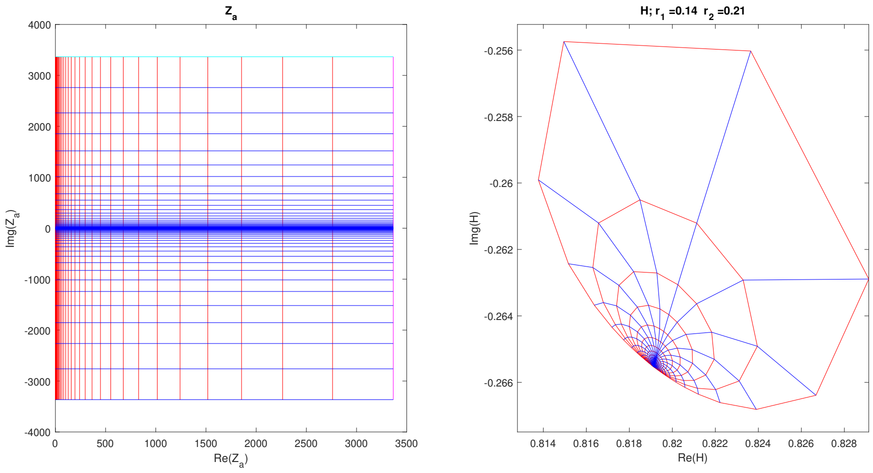


Fig. 3.3.: An example of a mapping, with the range of Z_as on the left side, and the mapped transfer function on the left for $f = 1000$Hz. The maximum of the real and imaginary part of Z_a are in cyan and magenta, and are, when zoomed in on the figure, represented in the vanishingly small lines in the transfer functions, here in the lower left corner.

The frequency that the pressures p_1 and p_2 are measured for in figure 3.3 is 1000Hz, measured at the microphone positions r_1 and r_2. The transfer function H is calculated from equation (2.11) in section 2.2, for the microphone positions $r_1 = 0.14$m and $r_2 = 0.21$m from the center of the cylinder, which has radius $a = 0.1$m. The code for this particular calculation is in appendix A.3.

To find the most ideal setup of the measurements, it was decided to look at the area of the mapped transfer functions. The bigger this area is, the less susceptible this measurement will be to background noise. Furthermore, as distances from one iteration of the transfer function H to another is, the easier the model has to appropriate to the correct Z_a in post-processing. By mapping the transfer function for different configurations of r_1 and r_2 and the range of frequencies 20 Hz - 20000 Hz (in third-octave bands) one can compare their areas and find the best configuration.

This comparison is also done in Matlab, by making and saving the area of the different mapped transfer functions. The area is calculated by drawing rectangles around the shapes the mapped H makes, touching the outer corners of the area of the mapping. These rectangles are what is saved in the `area` variable in the code in Appendix A.3. This bounding box approach is chosen for it simplicity, and is expected to be a well-working approximation. However, this might be a source of error further on. An illustration of this area restraint can be seen in figure 3.4.

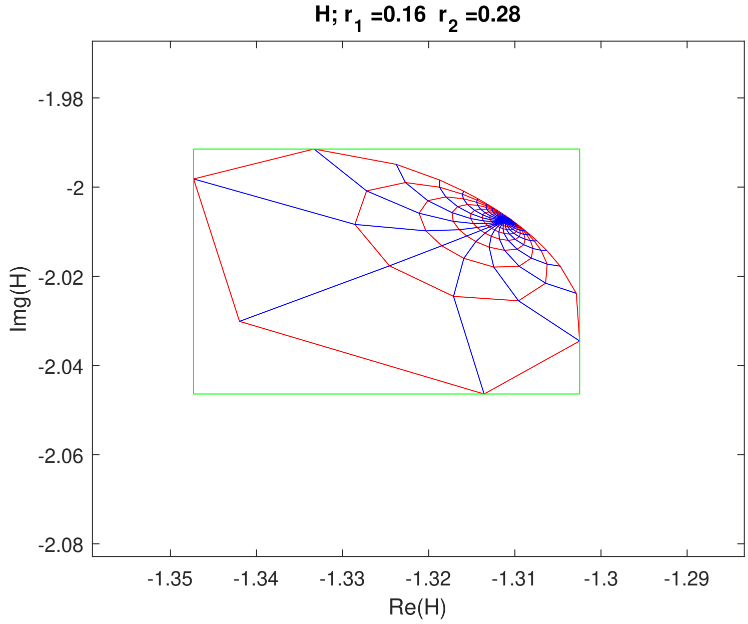


Fig. 3.4.: Illustration of the box (in green) around the mapped transfer function for $f = 1000$Hz.

Saving these `area` variables for different `r1`, `r2` and frequency, and then loading these in the code in Appendix A.4, one can find the combination of r_1 and r_2 that gives the largest area for most frequencies in the entire audible third-octave band. However, the model has not taken into account the imperfections of the microphone sensors, which might also be a source of error later on.

By looking at all frequencies within the audible spectra in third-octave bands, one finds that the best configuration of microphone positions is the one that is as close to the cylinder as possible. An example of this is for the frequency $f = 200$ Hz drawn up in figure 3.5, where one easily can see that the smallest configuration, $r_1 = 0.13$ m and $r_2 = 0.18$ m has the largest area out of the different cases. Note here that the radius of the model's cylinder is 10 cm, and that r_1 and r_2 are measured from the center of the cylinder, giving the respectful distances from the cylinder to be 3 and 8 cm.

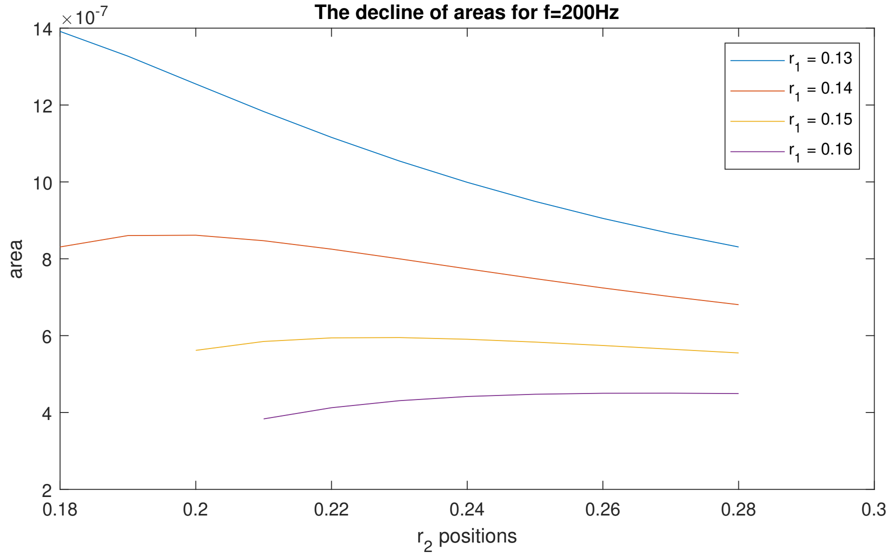


Fig. 3.5.: Comparisons of the areas of the mapped transfer function for one specific frequency. The configuration of the microphone positions from the center of the cylinder (with $a = 0.1$m) is in the x-axis for r_2 and the labelled lines for r_1

The reason for the lower restraint being 3cm is that the shape of the bark does not exactly give a cylindrical shape to the tree. Three centimeters from the stem should make the approximation done in the model less crude. The spacing between the two microphones should be at least 5 times the diameter of the microphones, in order to avoid the sensors affecting each others measurements. As the diameter can be as much as 1 cm, the spacing is chosen at 5cm. Thus, the configuration of one microphone placed 3 cm and the second 8 cm from the tree, is the best.

The loudspeaker is placed 5 m from the cylinder in the simulations. The available loudspeaker for measurement gives off a spherical wave, but in the simulations resembles a cylindrical wave. In order to unite this, the source has to be placed at least 4 meters away from the cylinder, such that both simulation and measurement can unite in the approximation to a plane wave.

The look of the setup becomes as seen in figure 3.6. Measurements are done on two surfaces, one tree in between Hovedbygget and El-bygget and one column at Realfagsbygget. Both measurements are performed at NTNU Gløshaugen. There is also done a free-field measurement at the second measurement site. Both tree and column are at least five meters from other surfaces, such that the direct sound from the loudspeaker is unobstructed. The ground is covered by four absorbers, such that the first reflection from the ground is dampened as much as possible. The height of loudspeaker and microphones are set above 1.5m to delay the reflection as much as possible, while having the setup also cleared of branches and eventual shrubbery higher up on the stem of the tree.

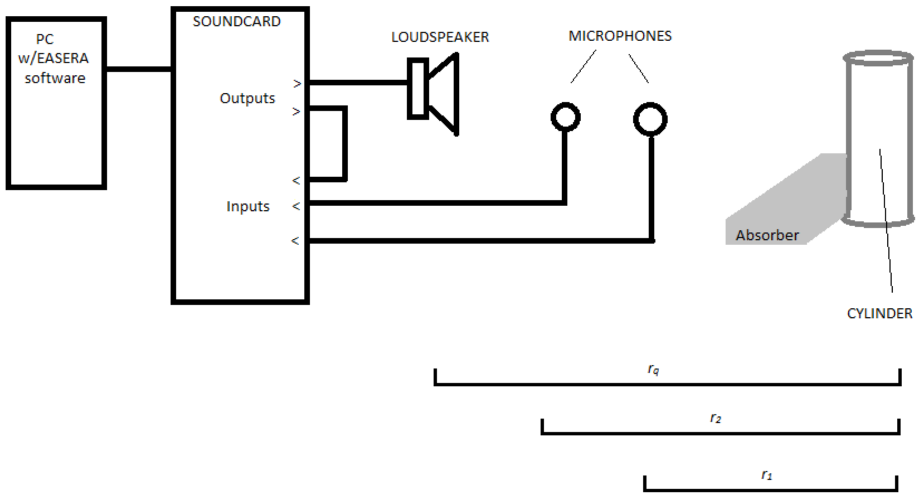


Fig. 3.6.: Measurement setup

The point of making measurements on all these different surfaces is to be able to compare different cases of absorbtion to see if and where the method is insufficient. The column is a plain cylinder of concrete, meaning that checking its impedance against external measurements is available. The smoothness of the column also makes it possible to observe the error source of bark making the surface of the tree jagged and obstructing the measurements. By wrapping absorbers around the cylinder and tree, one is also able to compare the measurement with a more well-developed method, done in a standing wave tube. The resulting absorbtion factor and impedance from this standing wave tube measurement are presented in Appendix B.

3.2.1 Tree absorbtion

The physical setup for the measurement of the tree is as seen in figure 3.7. The loudspeaker and microphones are placed at a height 1.5 m from the ground and 4.1 m from the stem of the tree. The microphones are placed 3 and 8 cm from the cylinder, measured from the middle of each microphone, and can be seen in figure 3.8. Wind speed is recorded continuously throughout the measurements.

Fig. 3.7.: The physical setup of the bark measurement

Fig. 3.8.: The microphone configuration in the bark measurement

This particular setup is done on the grounds of NTNU Gløshaugen, in between Hovedbygget and El-bygget. The tree is the last in a row, and over 5 meters from the closest surface, the next tree, which is to the left of the frame in figures 3.7 and 3.8. The tree is European beech, and with a radius of 0.234m, found by measuring the circumference of the tree and calculating the radius from it.

Firstly, measurements are done to the bare bark, and then, secondly, to the tree with two wood fibre absorbers, like the one on the ground in figure 3.7, wrapped around the tree above each other, and fastened with 4cm thick strap around and knotted together at the opposite side where the measurements are done. The absorber does not reach all around the tree, so there is a space on the opposite of the measurement without absorber, which is illustrated in figure 3.9.

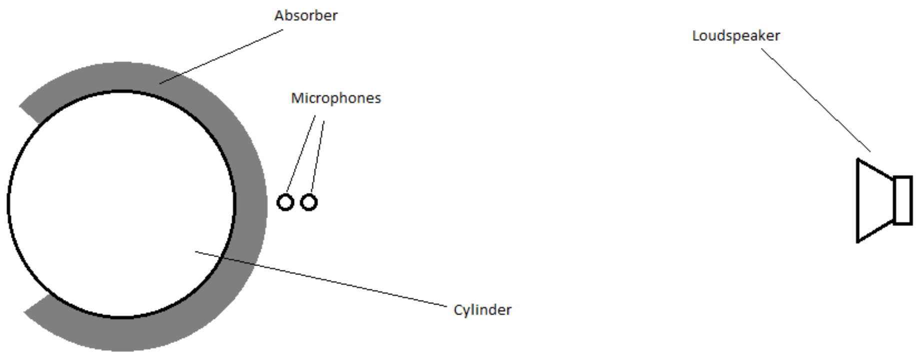


Fig. 3.9.: Illustration of the setup with absorber

3.2.2 Column absorbtion

The measurement is done on a concrete column with radius 0.23 m. A similar setup to the one used for the tree is used on the column, as seen in figure 3.10. However, there are a few notable differences. Because of the rigid surface of the concrete ground at this site, the absorbers are placed two on top of each other, longways, instead of sideways and side-by-side as with the tree trunk. The distance from loudspeaker to column is 4.6 m and the microphones, with the same spacing, are fitted with a wind shield as well, seeing as that day was windy during measurements.

Fig. 3.10.: The physical setup of the column measurement

In this setup, there was also made a free-field measurement, by moving the microphones, absorbers, and loudspeaker a few meters towards the camera in figure 3.10, where the same measurement is done, without the column, and with the distance from the furthest microphone to the loudspeaker being 4.29 m.

3.2.3 Equipment used

For the physical measurements, the following equipment (with serial number in parenthesis) was used:

- Soundcard; Roland OCTA-CAPTURE (A7E6783)
- Loudspeaker; GENELEC 1029A (029A041774)
- Computer with the EASERA software (version 1.2.13). Marked HP1 in acoutics lab at NTNU
- 6 Absorbers of wood fibres
- Measuring tape and laser measurement device
- Wind measurement device; WindMate WM-100 (13013)
- Sound calibrator; Norsonic NOR 1256 (125626366) with half-inch adapter

- 2 Microphones; BSWA 216 (4501090 and 4501095)

3.3 Post-processing

The post-processing is done in Matlab R2018b and is as seen in appendix A.5. The goal of the post-processing is to find the impedance, reflection and then the absorbtion factor, using the transfer function method in section 2.2. By comparing the measured transfer function with the mapping of transfer functions from section 3.1, using the same microphone configurations as the physical setup, one should be able to find the strongest correspondence between transfer functions in measurement and model.

From this, finding the corresponding impedance Z_a in the model and then the reflection and absorbtion from equation (2.12) in section 2.2 is only a matter of linking the transfer function of the model to the corresponding Z_a from before the mapping. Then the reflection factor can be found using equation (2.13).

There is also a time window of 1500 sample points chosen, which with sample frequency at 44100, gives 34 ms of the measurement actually used in calculations.

4 Results

4.1 Validity test of model

The test of validity as presented in section 3.1 yielded the figure as seen in figure 4.1. This model is identical to the figure at page 200 in [Mec08], which strongly indicates that the model is correctly set up, at least in accordance with Mechels calculations.

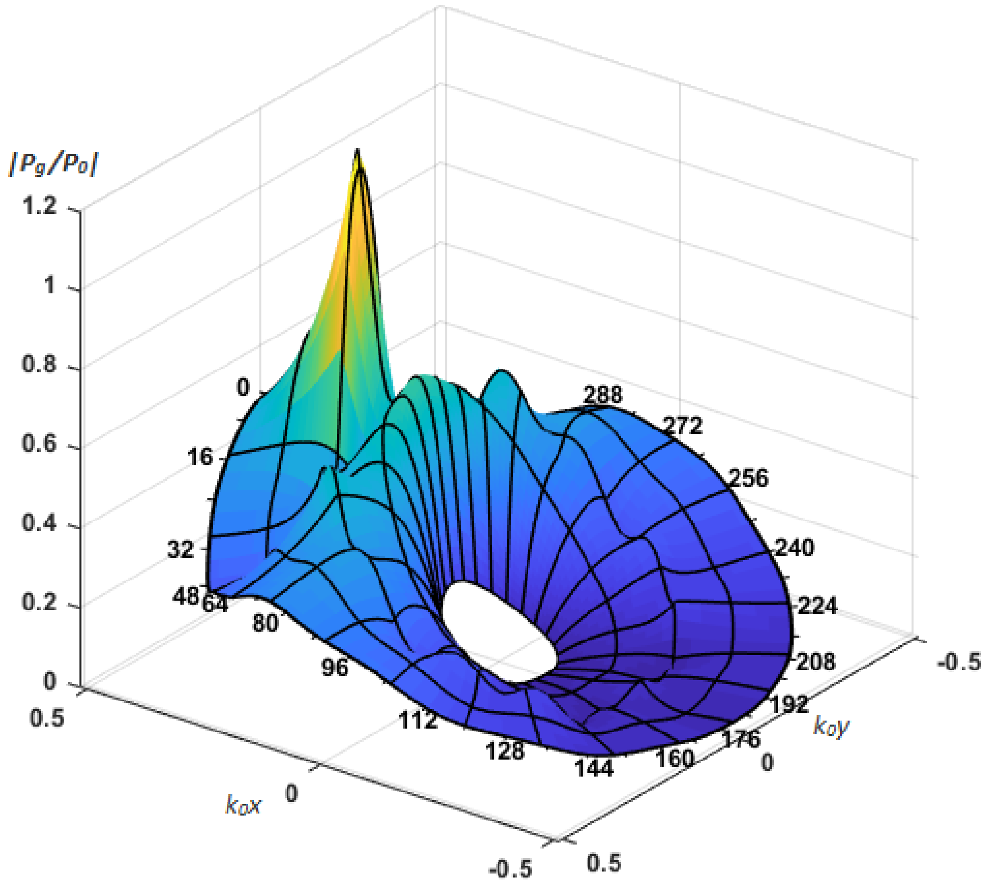


Fig. 4.1.: A test of the model by recreating the figure on page 200 in [Mec08]. $G = 0.5 - 2j$, $k_0a = 2$, $k_0r_q = 6.1$ and upper summation limit, $m_{hi} = 8$.

4.2 Convergence study

Performing the convergence study from section 3.1.1 on the range of ka being between 0.0366 and 73.27, ($0.1 \leq a \leq 0.2$ and $0.2 \leq f \leq 20000$) the upper and lower limits of n can be plotted and a curve is able to be fitted in between the two. Figure 4.2 shows this curve in red between the lower and upper limit of $m_{hi} = n$, with the function $n(ka) = 5 \cdot ka + 75$, which is also implemented for the number of maximum ms in further calculations.

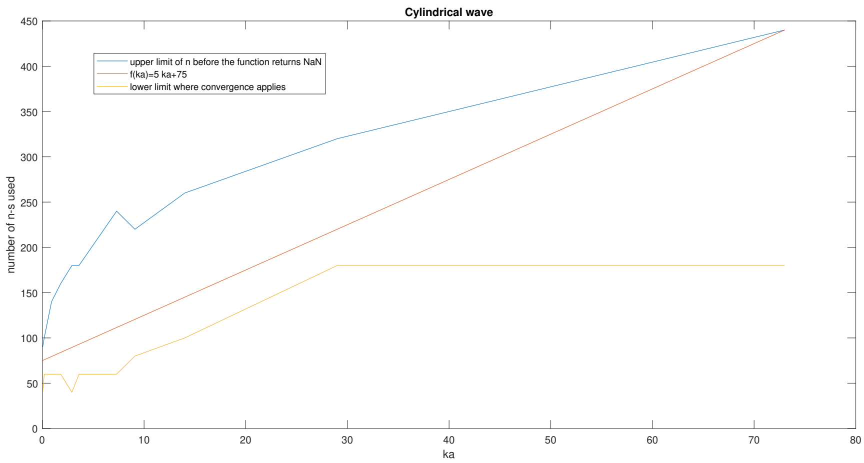
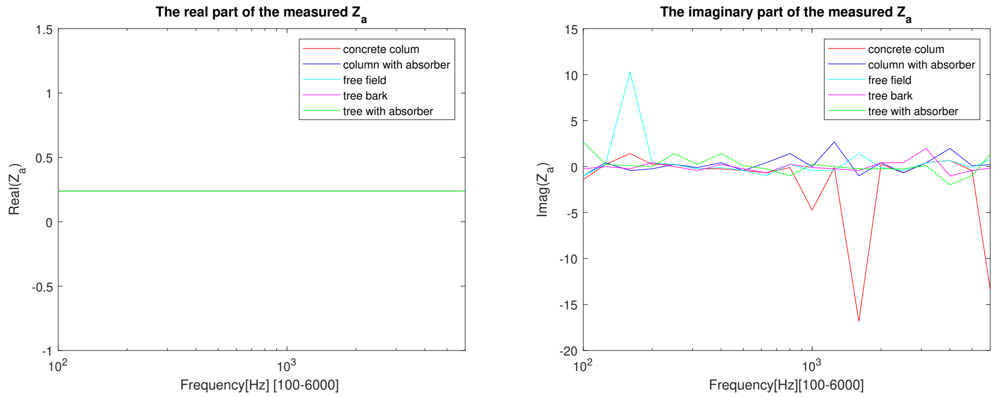


Fig. 4.2.: The result of the convergence study, the function (red) in between the upper and lower limits in blue and yellow.

4.3 Measurements

Measurements were carried out as explained in section 3.2. The wind speeds for the measurements at the site where tree measurements were carried out ranged between < 1m/s and 2.3m/s. For the column and free field measurements, the wind speed ranged between 3.2m/s and 0.9m/s, however this time the setup was equipped with a wind cap to account for the higher wind speeds. Other possible contributors to the background noise were several birds chirping in the proximity of the measurements, especially around the tree.

The measurements were compared and appropriated to the model using the code in Appendix A.5. The resulting impedances for the different measurements are shown in figure 4.3



(a) The real part of the impedance Z_a for all the different measurements
(b) The imaginary part of the impedance Z_a for all different measurements

Fig. 4.3.: Z_a measured and approximated through mapped H.

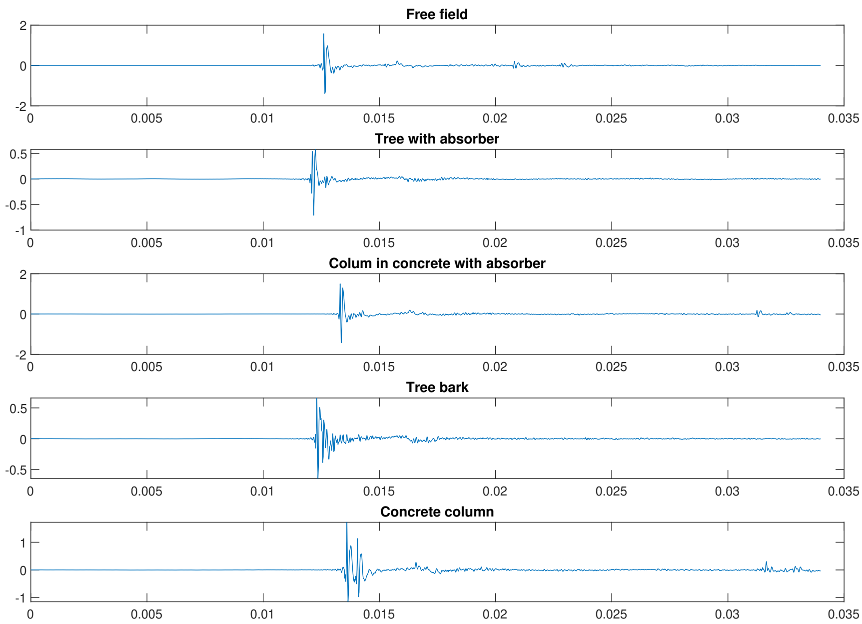


Fig. 4.4.: Impulse responses

The figures in 4.3 clearly shows that the measurement and following post-processing clearly does not yield the expected results. The five cases should give a range of impulse responses all the way from Z_a being the specific acoustic impedance of air in the free field case to it being huge, edging close to infinity for the case of the rigid concrete column. Instead, the impulse responses are generally the same, laying on top of each other for the real part and varying seemingly at random for the imaginary part.

Looking at the impulse responses in figure 4.4 one can see that they are for the most part consistent with expectations of measurements. The start of the first excitation vary with the small varying distance from the start of the measurement, which is expected, as the distance from loudspeaker to microphones differed with as much as 0.502 meters (which corresponds to a time shift at ≈ 0.0014s in the concrete column case). The measurement on the cylinder without absorber clearly shows the first excitation from the direct wave and then the second from the reflection off the concrete. The absorbed cases, both with tree and column, shows that the reflected wave is almost completely obviated, which is also expected with the previously measured absorbtion in a standing wave tube of the same sample, only laid flat (see appendix B). The first reflection coming from the ground can with simple mirror source calculations be identified as the small disturbances at around 0.016 s in all cases. This reflection is so slight that there is little chance it affects the measurements and further calculations gravely.

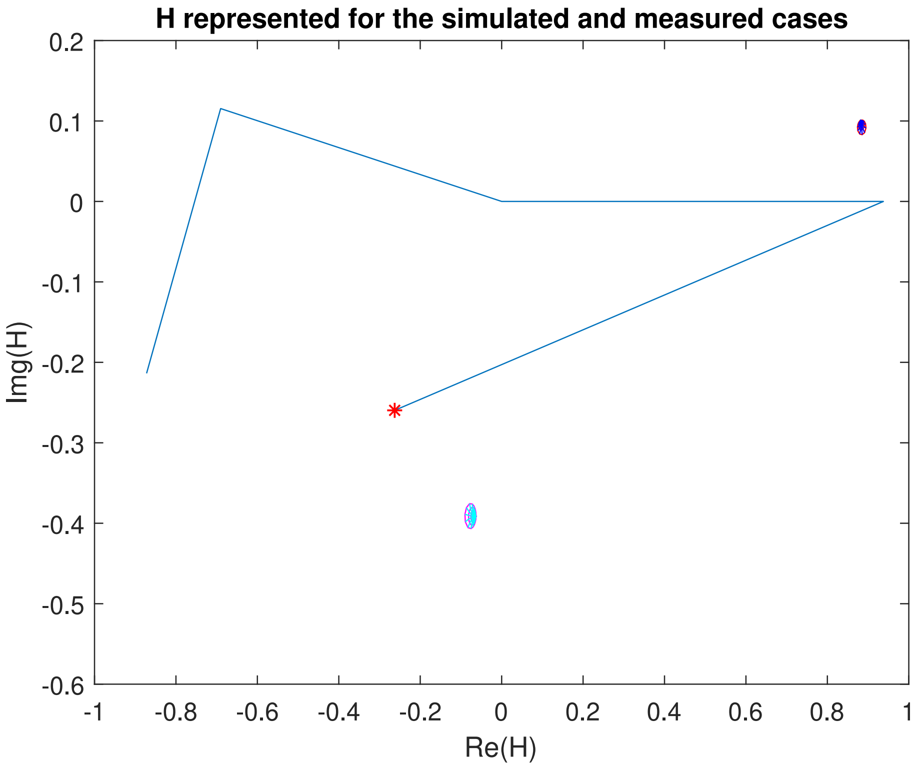


Fig. 4.5.: The transfer function data from measurements (blue line) compared to the two simulated mapped cases ($f = 3150$Hz). The shape in red and blue is the mapped H for the column case, while the one in cyan and magenta is for the tree configuration. The red star signals the concrete column measurement, and then the line followsthe expected lower Z_as, to tree bark, to column with absorber, to tree with absorber, to free field case.

The exception from these generally consistent impulse responses is in the tree bark case. Here, there are no clear second reflection, only fading peaks that does not look like a singular reflection, but instead a lot of noise.

The range of both measured and simulated transfer functions are presented in figure 4.5. Looking at them, one clearly sees something amiss. The point of these transfer function maps are to cover the entire space a transfer function could possibly cover, and then have the points of the transfer functions for the measurements be within this space. As seen in figure 4.5, this is not the case for either of the transfer functions from the measurements.

4.4 Discussion

The results presented in the above sections does witness that while both measurements and simulations are in line with the excepted results, there seems to be problems in joining theory and practice, which in turn leads to inefficiencies in the method.

There might be several reasons for this. Firstly, the shape of the tree. The approximation that the tree is entirely cylindrical might not be a good one and would result in some unexpected impulse responses in the tree bark measurements, especially when measurements are taken so close to the tree as the method would predict. The impulse response belonging to the tree bark measurement in figure 4.4 supports this theory. This is however not much of an approximation for the column case, nor for the cases with the absorber wrapped around the tree, as these impulse responses are fairly smooth. Moving the microphones further back might rectify this problem, but then at the expense of the transfer function.

In order to have the absorbers stay wrapped around the cylinders, one would need to tie them onto the tree or column with a strap. The absorbers did not reach all around the cylinders, however, as seen in figure 3.9, which might cause unwanted reflections from the back of the objects that has not been taken into account in simulations. The strap itself might also add to unwanted reflections.

The effect the background noise had on the measurements are rated as fairly small, as the noise the wind gives off is almost no-existent when listening to the impulse responses, and the bird chirps are low and not heard before long after the time windowing takes effect and blocks them out.

There might be a chance that the box approximation in section 3.2 is too crude and leads to errors. This is unlikely, though, as in most cases, orientation for the shape changes with frequency, not different r_1, r_2 and r_q. An example of this can be seen in figure 4.5, where size changes with configuration, but not which way the mapped shape is directed. There is also little chance that the model includes too few Z_as, as the outer boundaries of the transfer function simulation is not made out of the largest value of Z_a, which is in a different color, seen and explained in figure 3.3 and its caption.

Looking at figure 4.5, one sees the spread of data points from the measurement, which are more or less the same for all measured frequencies, but also the spread of the simulated cases. The only difference between these two simulations is a 0.004 m larger radius for the tree case than the column case (which in turn shifts the measurement points r_1 and r_2 the same amount). This strongly indicates that the measured distances using for the most part a measuring tape are not accurate enough, especially for such short lengths. Also, the movement and following correction of the microphone placements when adding the absorbers has not been taken into account in the simulations, which might add heavily to this problem, as the absorbers were $3 - 4$ cm thick, and would potentially make entirely different transfer functions.

5 Conclusion

This thesis has explored the possibilities of accurately measuring the absorbtion characteristics of cylindrical shapes, with the purpose to apply it to tree bark measurements. This has proven a challenge, from finding appropriate pre-existing models, to setting up these models in a satisfactiory fashion in Matlab, to measurement and finally to combining theory and practice.

The thesis has gone into detail on creating a model for cylindrical waves scattering on a cylinder. It has performed a convergence study to find the optimal maximum summation limit for this model, which ended up being a function of the ka. This was done in order to sum over enough ns to the point where it converges, while also not going into so small numbers in the Hankel functions such that the computer returns `NaN` instead of numbers. This function's expression ended in being $n(ka) = 5 \cdot ka + 75$, which fits between the limits for ka between 0.0366 and 73.27.

Using the model, conformal mapping was performed in order to find the best measurement positions for the transfer function measurement. Using the assumption that the tree and column the measurements were performed on were smooth cylinders, the mapping resulted in a configuration where distance to cylinder and spacing between the microphones should be as close as possible. Based on this, the choice was made to set the microphones at a spacing of 5 cm apart on the axis created by the source and the cylinder, the closest microphone 3cm from the cylinder. The former was chosen to avoid the microphones affecting each other's measurement, and the latter in an attempt to avoid physical limitations not included in the model such as the jagged shape of the bark on the tree.

From the impulse responses of the different measurements, it seems as though the measurement worked fine for the planar surfaces, but fail in the tree bark measurement. Furthermore, comparing the scope of mapped transfer functions for the different cases with the actual measured and calculated transfer function, these two does not correspond in the slightest. Thus further calculations of impedance on the cylinders, Z_a, are not at all in correspondence with theory, and it is fairly clear that something has gone awry in either calculations, modelling, comparison, measurement or all of the above.

5.1 Further work

There are some steps that can be done in order to find where the work towards calculation of absorbtion fails. It would be interesting to see how the transfer functions act when microphones are placed at a distance further away from the cylinder, to see if the shape of the tree stem has the suspected effect, or if there are other disturbing effects from having the microphones at the distances the mapping comparison study would have it. One could also try to find a more accurate way of setting and measuring the microphone-cylinder distances, as these metric distance measurements seems to have a bigger impact on the simulations than previously assumed, and should probably be measured using something more accurate than a millimeter-spaced measuring tape.

More alterations to the measurement setup could be to switch out the loudspeaker with a line source to see the effect an actual cylindrical wave would have to the measurement. One could also try to find an absorber to wrap all around the tree, to make sure reflections from the back does not affect the measurements.

Another suggestion would be to redo calculations to look for any obvious errors in coding the model from Mechel [Mec08]. There might also be a reasoning in trying for the calculation suggested by Swearingen et al. [SS12] although the calculations are more complicated and requires more computing power. The possibility of using other methods of measurements, such as one with intensity probe or possibly one more akin to ISO 354 should also be further explored in order to work out one well-performing method.

Bibliography

[03] *Acoustics - Measurement of sound absorbtion in a reverberation room*. Standard EN ISO 354:2003 E. Apr. 2003 (cit. on p. 3).

[98] *ISO 10534-2:1998(E) Acoustics - Determination of sound absorbtion coefficient and impedance in impedance tubes - Part 2: Transfer function approach*. Standard. Geneve, Switzerland: International Organization for Standardization, 1998 (cit. on pp. 3, 7).

[DVK01] Guillaume Dutilleux, Tor Erik Vigran, and Ulf R. Kristiansen. „An in situ transfer function technique for the assessment of the acoustic absorption of materials in buildings“. In: *Applied Acoustics* 62 (2001) (cit. on p. 8).

[LTB19] Mengmeng Li, Jian Kang Timothy Van Renterghem, and Dick Bottledooren. „Sound absorbtion by tree bark“. In: *Proceedings of the 23rd International Congress on Acoustic*. Aachen, Germany, 2019 (cit. on p. 3).

[LU89] Tai-bao Li and Mitchuhiro Ueda. „Sound scattering of a plane wave obliquely incident on a cylinder“. In: *Journal of Acoustical Society of America* 86 (1989), pp. 2363–2368 (cit. on p. 5).

[Mec08] Fridolin P. Mechel. *Formulas of Acoustics, Second Edition*. Springer-Verlag, 2008 (cit. on pp. iii, 4–6, 11, 23, 30).

[PAH88] Margaret A. Price, Keith Attenborough, and Nicolas W. Heap. „sound attenuation through trees: Measurements and models“. In: *The Journal of the Acoustical Society of America* 84 (1988), pp. 1836–1844 (cit. on p. 3).

[Pie89] Allan D. Pierce. *Acoustics - An Introduction to Its Physical Priciples and Applications, Third Edition*. McGraw-Hill Inc., 1989 (cit. on p. 5).

[Ren15] T. Van Renterghem. „Exploiting Supporting Pole to Increase Road Traffic Noise Shielding of Tree Belts“. In: *Acta Acustica united with Acustica* 101 (2015) (cit. on p. 3).

[RMH77] G. Reethof, H. McDaniel, and G.M. Heisler. „Sound Absorbtion characteristics of Tree Bark and Forest Floor“. In: *Proceedings of the conference on metropolitan physical environment*. Upper Darby, Pennsylvania, 1977 (cit. on p. 3).

[SS12] Michelle E. Swearingen and David C. Swanson. „A Numerical Model for Point Source Scattering from an Impedance Cylinder Placed Normal to an Impedance Plane“. In: *Acta Acustica united with Acustica* 98 (2012), pp. 523–533 (cit. on pp. 5, 30).

Coding in Matlab

A

A.1 cylWaveOnCyl

```
function p_tot=cylWaveOnCyl(f,a,r_q,r,theta,m,Z_a,c)
%Based on formulas from Formulas of Acoustics 2nd edition
    (2008)
%by F.P. Mechel p.199-200


%Input Variables:
%f - frequency
%a - radius of trunk
%r_q - source position the shortest direction from the
    cylinder
%r - distance to reciever
%theta - angle in radians of the reciever poisition
%m -summations should be done using an infinite series. n is
     a vector
%    starting at n(1)=0 and ends at the last number to sum
    over. Thus it dictates the
%   maximum number to sum over in this series
%Z_a - impedance of cylinder
%c - speed of sound in medium

%Output variables:
%Output variables are included and excluded as the user
    seems fit
%p_tot - the total pressure field, combined of the incoming
    and reflected
%wave from the cylinder
%absp_tot - the absolute value of the total pressure field,
    combination of
%incoming and reflected wave from the cylinder.
%ka - the wavenumber (k) multiplied by the radius of the
    cylinder (a)
```

```
k_0=2*pi*f/c; %Wavenumber
%ka=k_0*a; %easier access to k_0*a
G=1./Z_a;
p_q(length(theta))=0;%Incoming field from line source
p_s(length(theta))=0; %the scattered field

%rdash=sqrt((r.*sin(theta)).^2+(r_q-r.*cos(theta)).^2); %The shortest distance between source and reciever

epsilon_m(1)=1;
epsilon_m(2:length(m))=2;
P_0=1; %amplitude of incoming wave
if r<=r_q %in case (a)
    for j=1:length(theta)
        for i=1:length(m)
        p_q(j)=p_q(j)+P_0*epsilon_m(i)*besselj(m(i),...
            k_0*r)*besselh(m(i),2,k_0*r_q)*cos(m(i)*theta(j));
        end
    end
else %in case(b)
     for j=1:length(theta)
        for i=1:length(m)
        p_q(j)=p_q(j)+P_0*epsilon_m(i)*besselj(m(i),...
            k_0*r_q)*besselh(m(i),2,k_0*r)*cos(m(i)*theta(j));
        end
     end
end

%scattered field

%value for the scattered field
c_m(length(m))=0;
for i=1:length(m)
    c_m(i)=((G+(m/(k_0*a)))*besselj(m(i),k_0*a)-1i*besselj(m(i)+1,k_0*a))/...
      ((G+(m/(k_0*a)))*besselh(m(i),2,k_0*a)-1i*besselh(m(i)+1,2,k_0*a));
end
```

```
for j=1:length(theta)
      for i=1:length(m)
            p_s(j)=p_s(j)-P_0*epsilon_m(i)*c_m(i)*...
                  besselh(m(i),2,k_0*r_q)*besselh(m(i),2,k_0*r)*
                      cos(m(i)*theta(j));
      end
end

p_tot=p_q+p_s;%total field
%absp_tot=abs(p_tot);

end
```

A.2 Code for making the polar plots and the complete convergence study

```
f=[100 500 1000 2000];
r_q=20;%0.8*2.4; %distance from center of cylinder to source
    (note: line source)

r1=0.5;%a:(0.5-a)/96:0.5;
r2=0.6;
% Z_a(20,20)=0;%0.5+0*1i; %surface impedance of the cylinder
% for i=1:20
%       Z_a(i,:)=Z_a(i,:)+0.1*i;
% end
% for j=1:20
%       Z_a(j,:)=Z_a(j,:)+1i*0.1*j;
% end
c=343;
k_0=2*pi*f/c; %Wavenumber
Z_a=0.5;
theta=0:pi/48:2*pi; %Angle in radians between r and r_q
% upperlimit=k_0*a*5+75;
% m=0:upperlimit;
m=0:200;

ka=k_0*a;
p_tot1(length(f),length(m),length(theta))=0;
%p_tot2(length(m),length(theta))=0;
% absp_tot(length(f),length(m),length(theta))=0;
ka(length(theta))=0;
for h=1:length(f)
for i=1:length(m)
      for j=1:length(theta)
 p_tot1(h,i,j)=cylWaveOnCyl(f(h),a,r_q,r1,theta(j),m(1:i),
    Z_a,c);
 %[p_tot2(i,:),ka(i,:)]=cylWaveOnCyl(f,a,r_q,r2,theta(:),m
    (1:i),Z_a,c);
      end
end
end
%H=p_tot1./p_tot2;
absp_tot=abs(p_tot1);
```

```
fi=10;
sec=25;
thi=50;
fo=100;
fif=170;
six=190;
%placing=30;
figure(1)
sgtitle('f=1000 Hz');%(['ka=',num2str(ka(1,1))]);
subplot(2,3,1)
polarplot(theta,exVecFromMat(absp_tot(3,fi-6,:)),'y'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fi-5,:)),'m'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fi-4,:)),'c'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fi-2,:)),'r'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fi,:)),'g'), hold off;
title(['n=',num2str(fi)]);
subplot(2,3,2)
polarplot(theta,exVecFromMat(absp_tot(3,sec-17,:)),'y'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,sec-15,:)),'m'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,sec-11,:)),'c'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,sec-8,:)),'r'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,sec-5,:)),'g'), hold off;
title(['n=',num2str(sec)]);
subplot(2,3,3)
polarplot(theta,exVecFromMat(absp_tot(3,thi-17,:)),'y'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,thi-15,:)),'m'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,thi-11,:)),'c'), hold on;
```

```
polarplot(theta,exVecFromMat(absp_tot(3,thi-8,:)),'r'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,thi-5,:)),'g'), hold off;
title(['n=',num2str(thi)]);
subplot(2,3,4)
polarplot(theta,exVecFromMat(absp_tot(3,fo-17,:)),'y'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fo-11,:)),'m'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fo-8,:)),'c'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fo-5,:)),'r'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fo,:)),'g'), hold off;
title(['n=',num2str(fo)]);

subplot(2,3,5)
polarplot(theta,exVecFromMat(absp_tot(3,fif-17,:)),'y'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fif-13,:)),'m'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fif-9,:)),'c'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fif-5,:)),'r'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,fif,:)),'g'), hold off;
title(['n=',num2str(fif)]);

subplot(2,3,6)
polarplot(theta,exVecFromMat(absp_tot(3,six-18,:)),'y'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,six-12,:)),'m'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,six-9,:)),'c'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,six-5,:)),'r'), hold on;
polarplot(theta,exVecFromMat(absp_tot(3,six,:)),'g'), hold off;
```

```
title(['n=',num2str(six)]);
%%
%The gathered convergence information and the function in between setteld
%between and holding both requirements


ka2=[0.03, 0.18, 0.9, 1.8, 2.9, 3.6, 7.3, 9.1, 14, 29, 73];
pmin=[40,60,60,60,40,60,60,80,100,180,180];
pmax=[90,100,140,160,180,180,240,220,260,320,440];
g=5.*ka2 + 75;
figure(2)
plot(ka2,pmax), hold on;
plot(ka2,g), hold on;
plot(ka2,pmin),hold off;
xlabel('ka');
ylabel('number of n-s used');
legend('upper limit of n before the function returns NaN','f(ka)=5 ka+75','lower limit where convergence applies');
title('Cylindrical wave');
```

A.2.1 Code for polar plots with different ka's

This code is used to make figure 3.2.

```
figure(1)
subplot(2,2,1)
polarplot(theta,exVecFromMat(absp_tot(1,fi,:)),'g');
title(['ka=',num2str(k_0(1)*a)]);

subplot(2,2,2)
polarplot(theta,exVecFromMat(absp_tot(2,sec-5,:)),'g');
title(['ka=',num2str(k_0(2)*a)]);

subplot(2,2,3)
polarplot(theta,exVecFromMat(absp_tot(3,thi-5,:)),'g');
title(['ka=',num2str(k_0(3)*a)]);

subplot(2,2,4)
polarplot(theta,exVecFromMat(absp_tot(4,fo,:)),'g');
title(['ka=',num2str(k_0(4)*a)]);
```

A.2.2 exVecFromMat

```
function vec=exVecFromMat(m)
%When p_tot is a three-dimentional matrix (m), to be able to
    run its values
%through the polarplot function, we need to extract the
    values needed from
%p_tot and save them in a vector:

for i=1:length(m(1,1,:))
    vec(i)=m(1,1,i);
end

end
```

A.3 Mapping and making area

```
%Script for calculating the areas of the transfer function for specific
%microphone configurations, r1 and r2
clear all
a=0.1; %Radius of cylinder
f=[20 25 31.5 40 50 63 80 100 125 160 200 250 315 400 500 630 800 1000 ...
    1250 1600 2000 2500 3150 4000 5000 6300 8000 10000 12500 16000 20000]; %Frequencies



r_q=5; %distance from center of cylinder to source (note: line source)

r1=0.16; %Distance to the first microphone
r2=0.28; %Distance to the second microphone
theta=0; %Angle of the microphone in relation to the loudspeaker position
c=343; %speed of sound
k_0=2*pi*f/c; %wavenumber

upperlimit=k_0*a*5+75; %The limit of the m to sum over, dependant on frequency
for n=1:length(f)
m(n,:)=0:upperlimit;
end

Z_a(40,81)=0; %surface impedance of the cylinder

y=logspace(0,3,41);
for i=1:40
    Z_a(i,:)=Z_a(i,:)+0.1*i*y(i);
end
for j=-40:40
    if j<0
    Z_a(:,j+41)=Z_a(:,j+41)+1i*0.1*j*y(j*(-1));%10^(abs(j))*j/abs(j);
    elseif j==0
```

```
        Z_a(:,j+41)=Z_a(:,j+41)+1i*0.1*j;
        else
        Z_a(:,j+41)=Z_a(:,j+41)+1i*0.1*j*y(j);
        end
end
p_tot2(length(f),length(Z_a(:,1)),length(Z_a(1,:)))=0;
p_tot1(length(f),length(Z_a(:,1)),length(Z_a(1,:)))=0;

for q=1:length(f)
for i=1:length(Z_a(:,1))
    for j=1:length(Z_a(1,:))
        p_tot2(q,i,j)=cylWaveOnCyl(f(q),a,r_q,r2,theta,m(q
            ,:),Z_a(i,j),c);
        p_tot1(q,i,j)=cylWaveOnCyl(f(q),a,r_q,r1,theta,m(q
            ,:),Z_a(i,j),c);
    end
end
end

H=p_tot1./p_tot2;


%FINDING THE RESTRAINTS TO CALCULATE THE AREA
imagRes1=imag(H(:,1,1));
imagRes2=imag(H(:,1,1));
realRes1=real(H(:,1,1));
realRes2=real(H(:,1,1));
for i=1:length(H(:,1,1))
    for j=1:length(H(1,:,1))
        for l=1:length(H(1,1,:))
            if imag(H(i,j,l))<imagRes1(i)
            imagRes1(i)=imag(H(i,j,l));
            end
            if imag(H(i,j,l))>imagRes2(i)
            imagRes2(i)=imag(H(i,j,l));
            end
            if real(H(i,j,l))<realRes1(i)
            realRes1(i)=real(H(i,j,l));
            end
            if real(H(i,j,l))>realRes2(i)
            realRes2(i)=real(H(i,j,l));
            end
```

```
        end
    end
end

diffR(length(f))=0;
diffI(length(f))=0;
area(length(f))=0;
for i=1:length(f)
    diffR(i)=realRes2(i)-realRes1(i);
    diffI(i)=imagRes2(i)-imagRes1(i);
    area(i)=diffR(i)*diffI(i);
end

%FIGURES
figure(1)
subplot(1,2,1)
for i=1:length(Z_a(:,1))-1
plot(Z_a(i,:),'r')
hold on
end
plot(Z_a(length(Z_a(:,1)),:),'m')
hold on

for j=1:length(Z_a(1,:))-1
plot(real(Z_a(:,j)),imag(Z_a(:,j)),'b')
hold on
end
plot(Z_a(:,length(Z_a(1,:))),'c')
hold off
title('Z_a')
xlabel('Re(Z_a)');
ylabel('Img(Z_a)');



%figure for the mapping
subplot(1,2,2)
for i=1:length(H(18,:,1))-1
plot(exVecFromMat(H(18,i,:)),'-r')
hold on
end
plot(exVecFromMat(H(18,length(H(1,:,1)),:)),'-m')
```

```
hold on

for j=1:length(H(18,1,:))-1
plot(H(18,:,j),'-b')
hold on
end
plot(H(1,:,length(H(18,1,:))),'-c')
hold on

title(['H; r_1 =',num2str(r1),'  r_2 =',num2str(r2)])
xlabel('Re(H)');
ylabel('Img(H)');

figure(2)
for i=1:length(H(18,:,1))-1
plot(exVecFromMat(H(18,i,:)),'-r')
hold on
end
plot(exVecFromMat(H(18,length(H(1,:,1)),:)),'-m')
hold on

for j=1:length(H(18,1,:))-1
plot(H(18,:,j),'-b')
hold on
end
plot(H(1,:,length(H(18,1,:))),'-c')
hold on

plot([realRes1(18); realRes2(18)],[imagRes1(18); imagRes1(18)],'g'), hold on
plot([realRes1(18); realRes2(18)],[imagRes2(18); imagRes2(18)],'g'), hold on
plot([realRes1(18); realRes1(18)],[imagRes1(18); imagRes2(18)],'g'), hold on
plot([realRes2(18); realRes2(18)],[imagRes1(18); imagRes2(18)],'g'), hold off

title(['H; r_1 =',num2str(r1),'  r_2 =',num2str(r2)])
xlabel('Re(H)');
ylabel('Img(H)');
```

A.4 Area comparison

```
%Compare areas from mapping and plot
clear all


load('newarea013018.mat');
area1318=area;
clear area
load('newarea013019.mat');
area1319=area;
clear area
load('newarea013020.mat');
area1320=area;
clear area
load('newarea013021.mat');
area1321=area;
clear area
load('newarea013022.mat');
area1322=area;
clear area
load('newarea013023.mat');
area1323=area;
clear area
load('newarea013024.mat');
area1324=area;
clear area
load('newarea013025.mat');
area1325=area;
clear area
load('newarea013026.mat');
area1326=area;
clear area
load('newarea013027.mat');
area1327=area;
clear area
load('newarea013028.mat');
area1328=area;
clear area

load('newarea014018.mat');
```

```
area1418=area;
clear area
load('newarea014019.mat');
area1419=area;
clear area
load('newarea014020.mat');
area1420=area;
clear area
load('newarea014021.mat');
area1421=area;
clear area
load('newarea014022.mat');
area1422=area;
clear area
load('newarea014023.mat');
area1423=area;
clear area
load('newarea014024.mat');
area1424=area;
clear area
load('newarea014025.mat');
area1425=area;
clear area
load('newarea014026.mat');
area1426=area;
clear area
load('newarea014027.mat');
area1427=area;
clear area
load('newarea014028.mat');
area1428=area;
clear area

load('newarea015020.mat');
area1520=area;
clear area
load('newarea015021.mat');
area1521=area;
clear area
load('newarea015022.mat');
area1522=area;
clear area
```

```
load('newarea015023.mat');
area1523=area;
clear area
load('newarea015024.mat');
area1524=area;
clear area
load('newarea015025.mat');
area1525=area;
clear area
load('newarea015026.mat');
area1526=area;
clear area
load('newarea015027.mat');
area1527=area;
clear area
load('newarea015028.mat');
area1528=area;
clear area

load('newarea016021.mat');
area1621=area;
clear area
load('newarea016022.mat');
area1622=area;
clear area
load('newarea016023.mat');
area1623=area;
clear area
load('newarea016024.mat');
area1624=area;
clear area
load('newarea016025.mat');
area1625=area;
clear area
load('newarea016026.mat');
area1626=area;
clear area
load('newarea016027.mat');
area1627=area;
clear area
load('newarea016028.mat');
area1628=area;
```

```
clear area

f=[20 25 31.5 40 50 63 80 100 125 160 200 250 315 400 500 630 800 1000 ...
    1250 1600 2000 2500 3150 4000 5000 6300 8000 10000 12500 16000 20000];


area13=[area1318; area1319; area1320; area1321; area1322; area1323; ...
    area1324; area1325; area1326; area1327; area1328];
area14=[area1418; area1419; area1420; area1421; area1422; area1423;...
    area1424; area1425; area1426; area1427; area1428];
area15=[area1520; area1521; area1522; area1523; area1524; area1525;...
    area1526; area1527; area1528];
area16=[area1621; area1622; area1623; area1624; area1625; area1626;...
    area1627; area1628];

placing=11;

spot13=[0.18, 0.19, 0.20, 0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28];
spot14=[0.18, 0.19, 0.20, 0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28];
spot15=[0.20, 0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28];
spot16=[0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28];

%Comparing the areas, one position of r2 at a time
%The comparisons are for each frequency, finding the largest area for each
%frequency and then the configuration with the largest is awarded with a
%point in sumarea. The spot with the largest sumarea is found in pos--a,
%where the -- is the r2 position.
sumarea13(length(area13(:,1)))=0;
for i=1:length(area13(1,:))
    [max13,pos13]=max(area13(:,i));
```

```
        sumarea13(pos13)=sumarea13(pos13)+1;
end
[maxarea13a,pos13a]=max(sumarea13(:));

sumarea14(length(area14(:,1)))=0;
for i=1:length(area14(1,:))
        [max14,pos14]=max(area14(:,i));
        sumarea14(pos14)=sumarea14(pos14)+1;
end
[maxarea14a,pos14a]=max(sumarea14(:));

sumarea15(length(area15(:,1)))=0;
for i=1:length(area15(1,:))
        [max15,pos15]=max(area15(:,i));
        sumarea15(pos15)=sumarea15(pos15)+1;
end
[maxarea15a,pos15a]=max(sumarea15(:));

sumarea16(length(area16(:,1)))=0;
for i=1:length(area16(1,:))
        [max16,pos16]=max(area16(:,i));
        sumarea16(pos16)=sumarea16(pos16)+1;
end
[maxarea16a,pos16a]=max(sumarea16(:));

%finding the biggest area by comparing the biggest of each
    of the fixed r2
maxtogether=[area13(pos13a,:); area14(pos14a,:);area15(
    pos15a,:);area16(pos16a,:)];
sumareafull(length(maxtogether(:,1)))=0;
for i=1:length(maxtogether(1,:))
        [maxall,posall]=max(maxtogether(:,i));
        sumareafull(posall)=sumareafull(posall)+1;
end
%finding the right position of both r1 and r2
[maxareafulltoghether,posfull]=max(sumareafull(:));
switch posfull
        case 1
                r2pos=0.13;
                r1pos=spot13(pos13a);
        case 2
                r2pos=0.14;
```

```
            r1pos=spot14(pos14a);
        case 3
            r2pos=0.15;
            r1pos=spot15(pos15a);
        case 4
            r2pos=0.16;
            r1pos=spot16(pos16a);
end
%figure of the areas for different microphone configurations
    at one
%specific frequency
figure(1)
plot(spot13, area13(:,placing)), hold on;
plot(spot14, area14(:,placing)), hold on;
plot(spot15, area15(:,placing)), hold on;
plot(spot16, area16(:,placing)), hold on;
legend('r_1 = 0.13','r_1 = 0.14', 'r_1 = 0.15', 'r_1 = 0.16'
    )
xlabel('r_2 positions')
ylabel('area')
title(['The decline of areas for f=',num2str(f(placing)), '
    Hz']);
```

A.5 Post-processing

```
clear all
close all
%**************************************************
%**Code for post−processing the measurement results**
%**************************************************
%By: Liv Astrid Nygaard
%Last date of edit: June 19th, 2020


f=[20 25 31.5 40 50 63 80 100 125 160 200 250 315 400 500 630 800 1000 ...
    1250 1600 2000 2500 3150 4000 5000 6300 8000 10000 12500 16000 20000];
acol=0.23;
atree=0.234;
r_qcol=4.629+acol;%0.8*2.4; %distance from center of cylinder to source (note: line source)
r_qtree=4.127+atree;
r1col=acol+3;%0.28:0.01:0.3;%a:(0.5−a)/96:0.5;
r2col=acol+8;
r1tree=atree+3;
r2tree=atree+8;


theta=0;
c=343;
k_0=2*pi*f/c;

upperlimit=k_0*acol*5+75;
for n=1:length(f)
m(n,:)=0:upperlimit;
end


Z_a(40,81)=0;%0.5+0*1i; %surface impedance of the cylinder

y=logspace(0,3,41);
for i=1:40
    Z_a(i,:)=Z_a(i,:)+0.1*i*y(i);
```

```
end
for j=-40:40
    if j<0
    Z_a(:,j+41)=Z_a(:,j+41)+1i*0.1*j*y(j*(-1));%10^(abs(j))*j/abs(j);
    elseif j==0
    Z_a(:,j+41)=Z_a(:,j+41)+1i*0.1*j;
    else
    Z_a(:,j+41)=Z_a(:,j+41)+1i*0.1*j*y(j);
    end
end

%Simulation of the same case
p_tot1col(length(f),length(Z_a(:,1)),length(Z_a(1,:)))=0;
p_tot2col(length(f),length(Z_a(:,1)),length(Z_a(1,:)))=0;
p_tot1tree(length(f),length(Z_a(:,1)),length(Z_a(1,:)))=0;
p_tot2tree(length(f),length(Z_a(:,1)),length(Z_a(1,:)))=0;
%p_s1(length(f),length(Z_a(:,1)),length(Z_a(1,:)))=0;
%p_s2(length(f),length(Z_a(:,1)),length(Z_a(1,:)))=0;

for q=1:length(f)
for i=1:length(Z_a(:,1))
    for j=1:length(Z_a(1,:))
        p_tot1col(q,i,j)=cylWaveOnCyl(f(q),acol,r_qcol,r1col,theta,m(q,:),Z_a(i,j),c);
        p_tot2col(q,i,j)=cylWaveOnCyl(f(q),acol,r_qcol,r2col,theta,m(q,:),Z_a(i,j),c);
        p_tot1tree(q,i,j)=cylWaveOnCyl(f(q),atree,r_qtree,r1tree,theta,m(q,:),Z_a(i,j),c);
        p_tot2tree(q,i,j)=cylWaveOnCyl(f(q),atree,r_qtree,r2tree,theta,m(q,:),Z_a(i,j),c);
    end
end
end
Hcolsim=p_tot2col./p_tot1col;
Htree=p_tot2tree./p_tot1tree;

%%

%Loading the data for the five different cases:
%Col: Concrete cylinder without absorber
%Colabs: Concrete cylinder with absorber
```

```
%Nocyl: free field measurement without any reflecting
    surface
%Treeabs: beech measurement with absorber
%Treebark: beech measurement without absorber

%For all five cases the same operation is done, so comments
    on the "col"
%case is applicable to all

col=importdata('col1.txt','\t', 22);
colabs=importdata('meas1colabs.txt','\t',22);
treeabs=importdata('measurement1treeabs.txt','\t',22);
treebark=importdata('treebark1.txt', '\t', 22);
nocyl=importdata('nocyl2.txt','\t',22);

fs=44100; %Sampling frequency from the measurement data file
timelength=1500; %Windowing time to restrict the measurement
    to only direct sound and reflected sound from the tree
coltime=col.data(1:timelength,1); %the time in seconds
colp1=col.data(1:timelength,2); %measurement from the first
    microphone
colp2=col.data(1:timelength,3); %Measurement from the second
    microphone
% coltime=col.data(:,1); %Thehse are options that take in
    the whole length
% colp1=col.data(:,2);
% colp2=col.data(:,3);
COLp1=fft(colp1); %FFT of the first measurement
COLp2=fft(colp2); %FFT of the second measurement
Lcol=length(coltime);
fcol = fs*(0:(Lcol-1))/Lcol; %the frequency spectra
Hcol=COLp2./COLp1; %Transfer function dependant on frequency

colabstime=colabs.data(1:timelength,1);
colabsp1=colabs.data(1:timelength,2);
colabsp2=colabs.data(1:timelength,3);
% colabstime=colabs.data(:,1);
% colabsp1=colabs.data(:,2);
% colabsp2=colabs.data(:,3);
COLABSp1=fft(colabsp1);
COLABSp2=fft(colabsp2);
Lcolabs=length(colabstime);
```

```
fcolabs=fs*(0:(Lcolabs-1))/Lcolabs;
Hcolabs=COLABSp2./COLABSp1;

treeabstime=treeabs.data(1:timelength,1);
treeabsp1=treeabs.data(1:timelength,2);
treeabsp2=treeabs.data(1:timelength,3);
% treeabstime=treeabs.data(:,1);
% treeabsp1=treeabs.data(:,2);
% treeabsp2=treeabs.data(:,3);
TREEABSp1=fft(treeabsp1);
TREEABSp2=fft(treeabsp2);
Ltreeabs=length(treeabstime);
ftreeabs=fs*(0:(Ltreeabs-1))/Ltreeabs;
Htreeabs=TREEABSp2./TREEABSp1;

treebarktime=treebark.data(1:timelength,1);
treebarkp1=treebark.data(1:timelength,2);
treebarkp2=treebark.data(1:timelength,3);
% treebarktime=treebark.data(:,1);
% treebarkp1=treebark.data(:,2);
% treebarkp2=treebark.data(:,3);
TREEBARKp1=treebarkp1;
TREEBARKp2=treebarkp2;
Ltreebark=length(treebarktime);
ftreebark=fs*(0:(Ltreebark-1))/Ltreebark;
Htreebark=TREEBARKp2./TREEBARKp1;

nocyltime=nocyl.data(1:timelength,1);
nocylp1=nocyl.data(1:timelength,2);
nocylp2=nocyl.data(1:timelength,3);
% nocyltime=nocyl.data(:,1);
% nocylp1=nocyl.data(:,2);
% nocylp2=nocyl.data(:,3);
NOCYLp1=fft(nocylp1);
NOCYLp2=fft(nocylp2);
Lnocyl=length(nocyltime);
fnocol=fs*(0:(Lnocyl-1))/Lnocyl;
Hnocyl=NOCYLp2./NOCYLp1;

%%
%Averaging all the measurement over the third-ocatve bands
    that is used in the
```

```
%simulation to make accurate comparisons
%The new transfer functions are saved in Hcol2 and so on.

Hcol2(length(f))=0;
Hnocyl2(length(f))=0;
Htreebark2(length(f))=0;
Htreeabs2(length(f))=0;
Hcolabs2(length(f))=0;
Hcylabs2(length(f))=0;
lowerlimcol=1;
lowerlimnocyl=1;
lowerlimtreebark=1;
lowerlimtreeabs=1;
lowerlimcylabs=1;
for i=1:length(f)-1
    uplim=f(i)+(f(i+1)-f(i))/(2);
    [~,indexcol]=min(abs(fcol-uplim));
    [~,indexnocyl]=min(abs(fnocol-uplim));
    [~,indextreebark]=min(abs(ftreebark-uplim));
    [~,indextreeabs]=min(abs(ftreeabs-uplim));
    [~,indexcylabs]=min(abs(fcolabs-uplim));
    Hcol2(i)=mean(Hcol(lowerlimcol:indexcol));
    Hnocyl2(i)=mean(Hnocyl(lowerlimcol:indexnocyl));
    Htreebark2(i)=mean(Htreebark(lowerlimtreebark:indextreebark));
    Htreeabs2(i)=mean(Htreeabs(lowerlimtreeabs:indextreeabs));
    Hcylabs2(i)=mean(Hcolabs(lowerlimcylabs:indexcylabs));
    lowerlimcol=indexcol+1;
    lowerlimnocyl=indexnocyl+1;
    lowerlimtreebark=indextreebark+1;
    lowerlimtreeabs=indextreeabs+1;
    lowerlimcylabs=indexcylabs+1;

end

uplim=f(length(f));
[~,indexcol]=min(abs(fcol-uplim));
Hcol2(length(f))=mean(Hcol(lowerlimcol:indexnocyl));
[~,indexnocyl]=min(abs(fnocol-uplim));
Hnocyl2(length(f))=mean(Hnocyl(lowerlimcol:indexnocyl));
[~,indextreebark]=min(abs(ftreebark-uplim));
```

```
Htreebark2(length(f))=mean(Htreebark(lowerlimtreebark:indextreebark));
[~,indextreeabs]=min(abs(ftreeabs-uplim));
Htreeabs2(length(f))=mean(Htreeabs(lowerlimtreeabs:indextreeabs));
[~,indexcylabs]=min(abs(fcolabs-uplim));
Hcylabs2(length(f))=mean(Hcolabs(lowerlimcylabs:indexcylabs));

%%
%FInd Z_a by comparing the H from the model with H from the measurements
Z_col(length(f))=0;
realpos=1;
imagpos=1;
for i=1:length(f)
for re=2:length(Z_a(:,1))
for im=2:length(Z_a(1,:))
    if abs(Hcolsim(i,re,im)-Hcol2(i))<abs(Hcolsim(i,realpos,imagpos)-Hcol2(i))
        realpos=re;
        imagpos=im;
    end
end
end
Z_col(i)=Z_a(realpos,imagpos);
realpos=1;
imagpos=1;
end

Z_nocyl(length(f))=0;
realpos=1;
imagpos=1;
for i=1:length(f)
for re=2:length(Z_a(:,1))
for im=2:length(Z_a(1,:))
    if abs(Hcolsim(i,re,im)-Hnocyl2(i))<abs(Hcolsim(i,realpos,imagpos)-Hnocyl2(i))
        realpos=re;
        imagpos=im;
    end
end
```

```
end

Z_nocyl(i)=Z_a(realpos,imagpos);
realpos=1;
imagpos=1;
end

Z_treebark(length(f))=0;
realpos=1;
imagpos=1;
for i=1:length(f)
for re=2:length(Z_a(:,1))
for im=2:length(Z_a(1,:))
    if abs(Htree(i,re,im)-Htreebark2(i))<abs(Htree(i,realpos
        ,imagpos)-Htreebark2(i))
        realpos=re;
        imagpos=im;
    end
end
end

Z_treebark(i)=Z_a(realpos,imagpos);
realpos=1;
imagpos=1;
end

Z_treeabs(length(f))=0;
realpos=1;
imagpos=1;
for i=1:length(f)
for re=2:length(Z_a(:,1))
for im=2:length(Z_a(1,:))
    if abs(Htree(i,re,im)-Htreeabs2(i))<abs(Htree(i,realpos,
        imagpos)-Htreeabs2(i))
        realpos=re;
        imagpos=im;
    end
end
end

Z_treeabs(i)=Z_a(realpos,imagpos);
realpos=1;
```

```
imagpos=1;
end

Z_colabs(length(f))=0;
realpos=1;
imagpos=1;
for i=1:length(f)
for re=2:length(Z_a(:,1))
for im=2:length(Z_a(1,:))
     if abs(Hcolsim(i,re,im)-Hcolabs2(i))<abs(Hcolsim(i,
         realpos,imagpos)-Hcolabs2(i))
          realpos=re;
          imagpos=im;
     end
end
end

Z_colabs(i)=Z_a(realpos,imagpos);
realpos=1;
imagpos=1;
end


%%
%Figures


%
% figure(1)
% semilogx(f, real(Hcol2),'r'), hold on;
% semilogx(f, imag(Hcol2),'r:'), hold on;
% semilogx(f, real(Hnocyl2),'b'), hold on;
% semilogx(f, imag(Hnocyl2),'b:'), hold off;
% legend('Hcolreal','Hnocylimag', 'Hnocylreal','Hnocylimag')
    ;
% grid on;
% xlim([100 6000])
%[p_tot, partial]=cylWaveOnCyl(f,a,r_q,r1,theta,m,Z_a,c);'
%H=p_tot2./p_tot1;
figure(1)
semilogx(f, real(Z_col),'r'), hold on;
semilogx(f, real(Z_colabs),'b'), hold on;
```

```
semilogx(f, real(Z_nocyl),'c'), hold on;
semilogx(f, real(Z_treebark),'m'), hold on;
semilogx(f, real(Z_treeabs), 'g'), hold off;
legend('concrete colum','column with absorber','free field',
    'tree bark','tree with absorber')
xlabel('Frequency[Hz] [100-6000] ')
ylabel('Real(Z_a)')
title('The real part of the measured Z_a')
xlim([100 6000])

figure(2)
semilogx(f, imag(Z_col),'r'), hold on;
semilogx(f, imag(Z_colabs),'b'), hold on;
semilogx(f, imag(Z_nocyl), 'c'), hold on;
semilogx(f, imag(Z_treebark),'m'), hold on;
semilogx(f, imag(Z_treeabs), 'g'), hold off;
legend('concrete colum','column with absorber','free field',
    'tree bark','tree with absorber')
xlabel('Frequency[Hz][100-6000]')
ylabel('Imag(Z_a)')
title('The imaginary part of the measured Z_a')
xlim([100 6000])


plotfreq=find(f==3150);

figure(3)
for i=1:length(Hcolsim(plotfreq,:,1))-1
plot(exVecFromMat(Hcolsim(plotfreq,i,:)),'-r')
hold on
end
plot(exVecFromMat(Hcolsim(plotfreq,length(Hcolsim(1,:,1)),:)
    ),'-r')
hold on

for j=1:length(Hcolsim(plotfreq,1,:))-1
plot(Hcolsim(plotfreq,:,j),'-b')
hold on
end
plot(Hcolsim(1,:,length(Hcolsim(plotfreq,1,:))),'-b')
hold on
%Tree in cyan and magenta
```

```
for i=1:length(Htree(plotfreq,:,1))-1
plot(exVecFromMat(Htree(plotfreq,i,:)),'-m')
hold on
end
plot(exVecFromMat(Htree(plotfreq,length(Htree(1,:,1)),:)),'-m')
hold on

for j=1:length(Htree(plotfreq,1,:))-1
plot(Htree(plotfreq,:,j),'-c')
hold on
end
plot(Htree(1,:,length(Htree(plotfreq,1,:))),'-c')
hold on

title('H represented for the simulated and measured cases')
xlabel('Re(H)');
ylabel('Img(H)');

Hcomb=[Hcol2; Htreebark2; Hcolabs2; Htreeabs2; Hnocyl2];
plot(Hcomb(:,plotfreq)), hold on;
plot(Hcol2(plotfreq),'r*'), hold off;
% plot(Htreebark2(plotfreq),'m*'), hold on;
% plot(Hcolabs2(plotfreq),'b*'), hold on;
% plot(Hnocyl2(plotfreq),'c*'), hold on;
% plot(Htreeabs2(plotfreq),'g*'), hold off;



%plot of impulse responses
figure(4)
subplot(5,1,1)
plot(nocyltime, nocylp1);
title('Free field')
subplot(5,1,2)
plot(treeabstime, treeabsp1)
title('Tree with absorber')
subplot(5,1,3)
plot(colabstime,colabsp1)
title('Colum in concrete with absorber')
subplot(5,1,4)
plot(treebarktime, treebarkp1)
```

```
title('Tree bark')
subplot(5,1,5)
plot(coltime, colp1)
title('Concrete column')
```

B

Standing wave tube measurements

B.1 Impedance

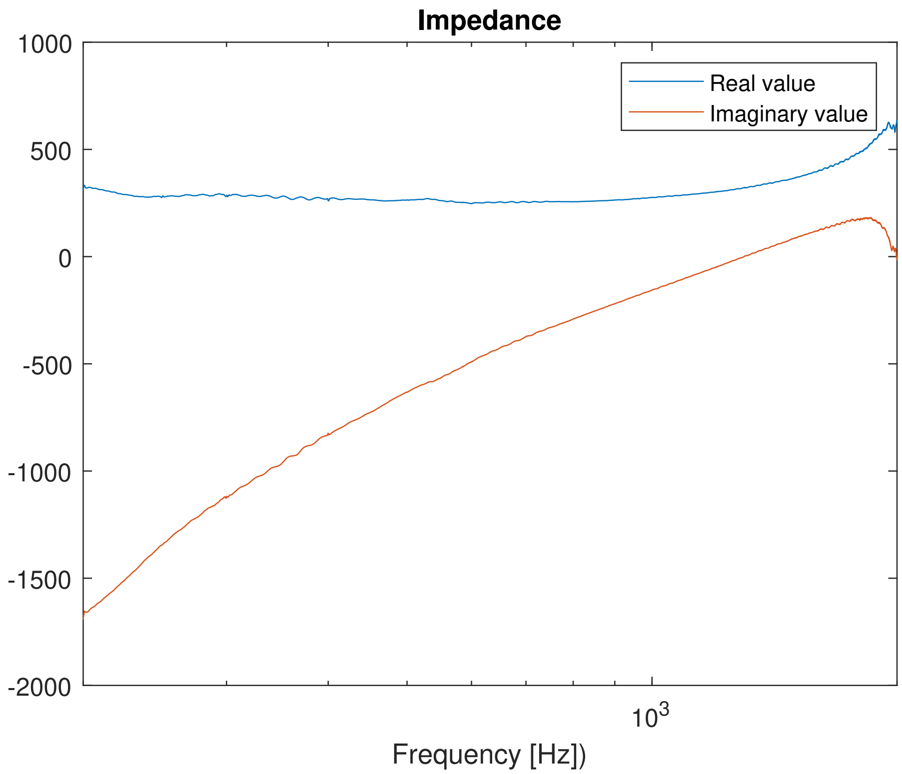


Fig. B.1.: The impedance measured and calculated for the samples of wood fibres laid flat and using a standing wave tube

B.2 Absorbtion

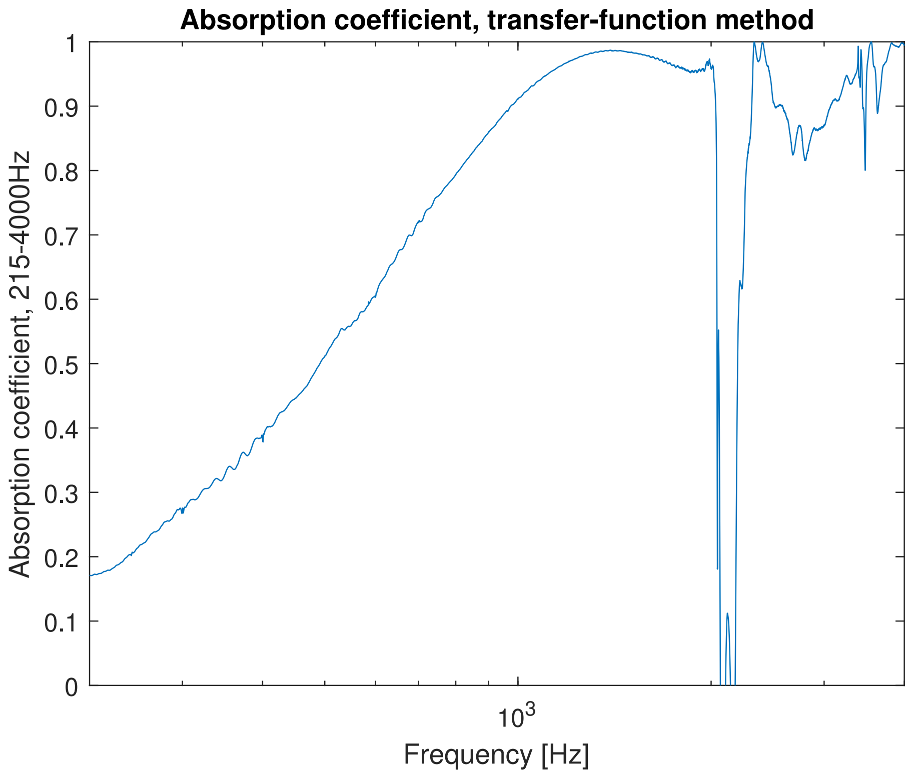


Fig. B.2.: The absorbtion measured for the samples of wood fibres laid flat and using a standing wave tube