TTT4250 - Acoustical Measurement Techniques

Laboratory Exercise 4

Sound Insulation Measurement

performed by

Erlend Kristiansen Berg
Hugo Suleyman Delarue
Stephanie Evers
Silje Toftevåg Urke

Report by

Silje Toftevåg Urke

Department of Electronic Systems
Norwegian University of Science and Technology

Summary

This report examines the measurement of the sound insulation of a material and how much it dampens a signal from a source room to a receving room. This is done by having two different loud speaker positions in the source room and 5 different microphone positions, in both the receiving room and the source room, for each loud speaker position. From this the apparent sound reduction index, R', is calculated. The reverberation time, $T30$, was measured. Furthermore the level difference, D, the normalized level difference, D_n, and the standardized level difference, D_{nT}, was calculated. Two theoretical estimations of the sound reduction index, R_{field} and R_{random}, is calulated and R_{random} is found to be best, but is only valid for frequencies above 1 kHz. It is not very accurate for the sound pressure levels as it is generally 2-3 dB to high, but it gives a good indication of the slope of R'. The weighted sound reduction index R'_W was found to be 33 dB.

These measurements are important in order to ensure good insulation between rooms in buildings giving good sound conditions for the people in these buildings. This is a very common measurement and the authorities have strict regulations to ensure that the sound level that travel between rooms are satisfactory for different purposes.

The report will discuss the necessary theory, the measurement method and the results and conclude with the findings of the report.

Contents

1 Introduction

This report examines the sound insulation of a material. This is done by measuring the apparent sound reduction index R', which is calculated by using the sound pressure in a source room and a receiver room. The source room has two differnt loud speaker positions, and for each loud speaker position 5 measurements were taken in the source room and 5 measurements in the receiving room. This gives a measure of how much the material dampens the signal from the loudspeaker. This is compared with the theoretically calulcated sound reduction indexes R_{random} and R_{field} to determine whether they give good approximations, and for which frequencies they give good estimations. These measurements are very useful when constructing buildings, as authorities set strict requirements regarding the sound insulation and level difference inbetween different rooms. This is to ensure good sound conditions in houses, offices, schools, etc., to ensure good living quality. These measurements are very common for consultants to conduct, and is needed to check that new buildings are within the requirements.

The report will address the necessary theory, the measurement method, the results and discussion of these before the report is concluded with whether the theoretical approximations are good estimators for the measured sound reduction index.

2 Theory

This chapter will present the necessary theory to reproduce the lab as well as general theory about level difference, sound reduction and the pressure method that is used. For this lab we want to examine the sound redcution index for a structure. This is be defined as

$$R = 10 \cdot \log_{10} \left(\frac{W_1}{W_2} \right), \tag{2.1}$$

where W_1 is the incident sound power and W_2 is the sound power transmitted through the structure. Assuming that the wall is infinite, the mass law is given as

$$R(\theta) \approx 10 \cdot \log_{10} \left(1 + \left(\frac{\omega \rho_s}{2\rho c} \cos(\theta) \right)^2 \right), \tag{2.2}$$

where ω is frequency-dependent and $\omega = 2\pi f$ and f is the frequency, θ is the angle of which the wave hits the structure, ρ_s is the surface density of the material. ρ is the density of air and c is the speed of sound in air. If we have that $\theta = 0$ this simplifies and we get the normal incidence mass law, defined as

$$R_0 = 10 \cdot \log_{10} \left(1 + \left(\frac{\omega \rho_s}{2\rho c} \right)^2 \right) \tag{2.3}$$

When the measurements are done on a source and receiving room it is relevant to look at the random incidence mass law, defined as

$$R_{random} \approx R_0 - 10 \cdot \log_{10} R_0, \tag{2.4}$$

where R_0 comes from the normal incidence mass law. Another aspect that is relevant is the field-incidence mass law, given as

$$R_{field} \approx R_0 - 5 \tag{2.5}$$

R_{random} is for incidence angles between 0 and 90 and R_{field} is for angles between 0 and 78, where 78 is found empirically from experiments. These are valid give that $R_0 > 15$ dB. These are theoretical estimations of the sound reduction index that is measured.

The transmitted power in the receiving room W_2 can be found using different methods. The pressure method is used for this report, but it is also possible to use a intensity method [1]. From [1] we have that when looking at the energy density inside the reverberation room and considering that waves from all directions we find that $W_1 = \frac{p_2^2}{\rho c} \cdot \frac{A}{4}$. Assuming the test sample behaves as the only sound source in the reciving room, which means there is no flanking transmission the relation between the transmitted sound power and the sound pressure in the receiving room is $W_2 = \frac{p_2^2}{\rho c} \cdot \frac{A}{4}$. Then Equation 2.1 simplifies, for details see [1] or [2], and becomes

$$R = L_1 - L_2 + 10 \cdot \log_{10} \left(\frac{S}{A} \right), \tag{2.6}$$

where L_1 is the sound pressure level in the source room, L_2 is the sound pressure level in the receiving room, S is the area of the separating surface between the rooms and A is the equivalent absorption area. The equivalent absorption area is determined from the reverberation time in the receiving room and calculated using Sabine's formula, which is

$$A = \frac{0.16V}{T} \tag{2.7}$$

In order to use Equation 2.6 the averaged sound pressure levels for the different microphone positions must be calculated. The recorded sound pressure level for all the microphone positions in each room are averaged according to

$$\bar{L} = 10 \cdot \log_{10} \left(\frac{1}{n} \sum_{j=1}^{n} 10^{\frac{Lj}{10}} \right), \tag{2.8}$$

where n is the number of microphone positions and L_j is the sound pressure for each of the microphone positions. From this it is possible to calculated the level difference, D, for the source room and for the receiving room. The level difference is given as

$$D = \bar{L_1} - \bar{L_2}, \tag{2.9}$$

where $\bar{L_1}$ is the averaged sound pressure level in the source room and $\bar{L_2}$ is the averaged sound pressure level in the receiving room. Furthermore it is possible to find the normalized level difference, D_n, which is given as

$$D_n = D - 10 \cdot \log_{10} \frac{A}{A_0}, \tag{2.10}$$

where A is the equivalent absorption area of the receiver room and A_0 is a reference area of 10 m^2.

An alternative to the normalized level difference is the standardized level difference, D_{nT}, which is given as

$$D_{nT} = D + 10 \cdot \log_{10} \frac{T}{T_0}, \tag{2.11}$$

where $T$is the reverberation time and T_0 is the reference reverberation time and is 0.5 s.

The reveberation time is defined as the time it takes the sound pressure level to reduce from a steady level to a 60 dB reduction of the steady level, and it is denoted T60. This is shown in Figure 2.1.

However, the signal-to-noise ratio, SNR is often not good enough to be able to accurately measure T60, and it is more common to measure either T30 or T20, depending on the sound pressure level of the noise floor. For T30 it is the time it takes for the sound level to reduce by 30 dB that is measured and multiplied by 2. This is assuming that the decrease is linear, and this gives an estimate of the time it takes for the sound pressure level to decrease by 60 dB. This time is called T30. This notation must not be confused with thinking that T30 is the time it takes for the sound pressure level to drop by 30 dB, as it is in fact an extrapolation from this measurement and an estimation of the time it would take the level to drop by 60 dB [2]. The same is true for T20, except it is used for lower SNR, and uses the time it takes for the sound pressure level to drop by 20 dB and estimates the time it would take the level to drop by 60 dB from this, by multiplying by 3.

To ensure good SNR the sound pressure level, which is the signal level combined with the background noise level, in the receiving room should be at least 10 dB higher than the background noise in any frequency band [1]. If this is not the case, the background noise needs to be corrected.

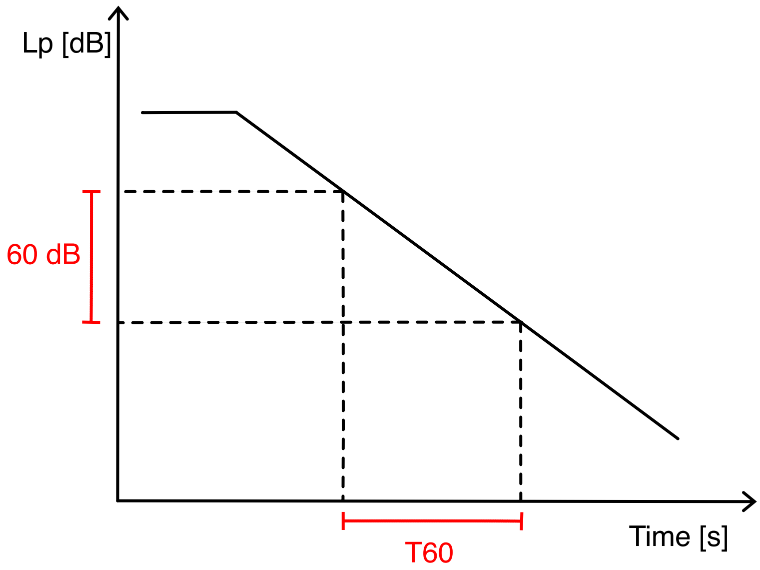


Figure 2.1: Illustration of how the reverberation time T60 is found.

The background noise should be at least 6 dB below the level of signal and background noise combined, and if it is between 6 dB and 10 dB the correction is calculated using

$$L = 10\log_{10}\left(10^{L_{sb}} - 10^{L_b/10}\right), \tag{2.12}$$

where L is the corrected signal level, L_{sb} is the combined signal and background noise and L_b is the background noise level. If, for any frequency band, the difference is smaller than 6 dB, the correction is 1.3 dB to a difference of 6 dB for this band.

The last index that is of interest is the apparent sound reduction index, R', which is defined as

$$R' = D + 10\log_{10}\left(\frac{S}{A}\right), \tag{2.13}$$

where, again, A is the equivalent absorption area of the receiving room and S is the area of the test material separating the rooms.

When the apparent sound reduction index R' is calulcated it is possible to detemine the weighted sound reduction index R'_W. For this a reference curve is used, as shown in Table 2.1.

The goal is to get the unfavorable deviation between R' and R'_W to be as large as possible, but below 32 dB, so that the next shift would bring the unfavourable deviation above 32 dB. The reference curve is shifted up or down depending on the measured R'. The unfavoruable deviation is the sum of difference between the measured R' and the shifted reference curve for each 1/3-octaveband. Finally when the reference curve is shifted so that the unfavoruable deviation is as wanted, the result is read of at 500 Hz, and this gives R'_W.

Table 2.1: Unshifted reference curve for 1/3-octave bands, [3].

Frequency band [Hz]	Unshifted refernece curve [dB]
100	33
125	36
160	39
200	42
250	45
315	48
400	51
500	52
630	53
800	54
1k	55
1.25k	56
1.6k	56
2k	56
2.5k	56
3.15k	56
4k	56
5k	56

3 Method and Equipment

This chapter describes the measuremnt methods and how it was performed. The measurements follow ISO 16283-1:2014 [5]. The measurements were conducted in two reverberation rooms, that are connected by an opening. In this opening there was placed a plate, which makes it possible to measure the sound reduction of this material, i.e. how much it dampens different frequencies. The test material is a two-layered plate consisting of equal parts plywood and rubber, and is the only thing connecting these two rooms.

The sound generated in the source room was steady and had a continous spectrum in the frequency range that is considered. The sound used was white noise. The largest of the two reverberation rooms is used as the source room and the smallest as the receiving room. The measurements were done in 1/3-octave bands. The rooms, the microphone postitions and the loudspeaker positions are shown in Figure 3.1.

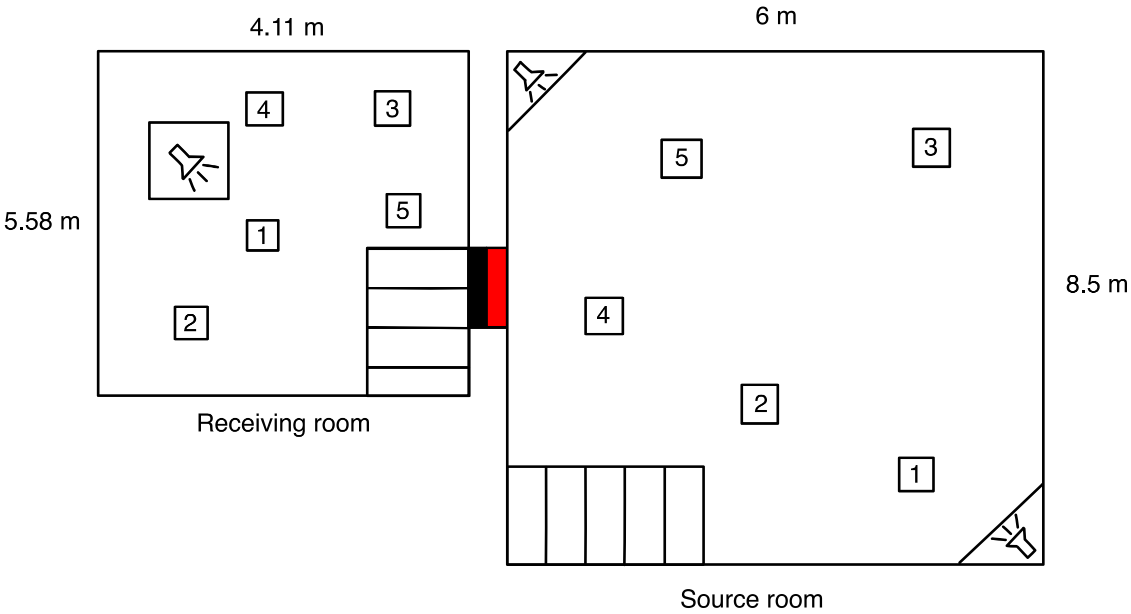


Figure 3.1: Sketch of the reciving room, source room, the loudspeaker positions and microphone positions in each room.

Before any measurements took place the sound and vibration analyser, NOR150, was calibrated using the built-in calibration and a calibrator. The door was almost closed for all the measurements. It could not be completely closed due to a microphone cable going into the room.

3.1 Source room

The source room is D0016 at NTNU Gløshaugen Elektro D+B2. This is a reverberation room. The dimensions of the room are presented in Table 3.1.

Table 3.1: Measurements of the source room.

	Length [m]	Width [m]	Heigt [m]
Measurement room	8.5	6.0	5.2

There are two loud speakers that are mounted in opposite corners of the room, one on the floor and the other one up against the ceiling. The locations of these loudspeakers as well as microhone positions is shown in Figure 3.1. Five different microphone positions were used, with two measurements per position, one for the lower loud speaker and one for the upper loudspeaker, which gives a total of 10 measurements for the source room. The minimum requirements for the placement of the microphones are shown in Table 3.2. This is in order to ensure good measurements and to be sure the measurements are done in the diffuse field of the sound sources.

Table 3.2: Minimum requirements for the distances to different parameters when deciding the microphone placements.

Parameter	Minimum requirement
Distance between microphones	0.7 m
Distance from microphone to any room boundary	0.5 m
Distance betwee microphones and sound source	1.0 m

The microphone positions were chosen and marked on the floor with tape. For each position it was verified that the reqquirements in Table 3.2 was fulfilled.

No persons were inside the room during the measurements.

3.2 Receving room

The source room is D0017 at NTNU Gløshaugen Elektro D+B2. This is also a reverberation room. The dimensions of the room are presented in Table 3.3.

Table 3.3: Measurements of the receiving room.

	Length [m]	Width [m]	Height [m]
Measurement room	5.58	4.11	4.58

For the measurements conducted in the receiving room the requirements for the microphone positions are the same as for the source room, shown in Table 3.2. The measurements are the same as for the source room. 5 different microphone positions were chosen and two measurements conducted per microphone position, one with the lower sound source connected and one with the upper sound source connected. This gives a total of 10 measurements for the receiving room. Since the test material is the only thing connecting the two rooms it can be consideredd the only sound source in the receiving room during these measurements.

For some of the measurements one person was inside the source room to make switching loud speakers more efficient, as it had to be done manually.

There were also taken two measurements of the background noise in the receiving room, for microphone position 3 and 5.

3.3 Reverberation time

The reverberation time was measured in the receiving room, for two different microphone positions. This was done by exciting the speaker in the receiving room with the same noise as used previously for the loud speakers in the source room. When the sound has reached a steady level in the receiving room, the sound was turned off. The sound and vibration analyser was set to a trigger, so that it started the measurements once the sound was turned off. The T30 was measured, and the SNR was found to be satisfactory to use T30, over T20. This was done two times for the same microphone position, which was microphone position 5. The two measuremnts were averaged.

3.4 Equipment

The equipment used this report is shown in Table 3.4.

Table 3.4: Equipment list.

Equipment	Model number/type
Sound and Vibration Analyser	NORSONIC Nor 150
Calibrator	Nor1256
Foam windsreen for microphone	
Microphone stand	K&M
Cables	NORSONIC Nor 1408A 5 metre
Microphone/Measurement probe	NORSONIC Type 1201/30490
Loudspeakers source room	Made in-house
Loudspeaker receiving room	Made in-house
NAD Electronics LTD	Model 208 Stereo Power Amplifier
Noise Generator	Type 1405 Elektronikklaboratoriet ved NTH elab AN 2005 B037
Ruler	1 metre
Two-layered isolation sample containing equal parts of plywood and rubber	

4 Results

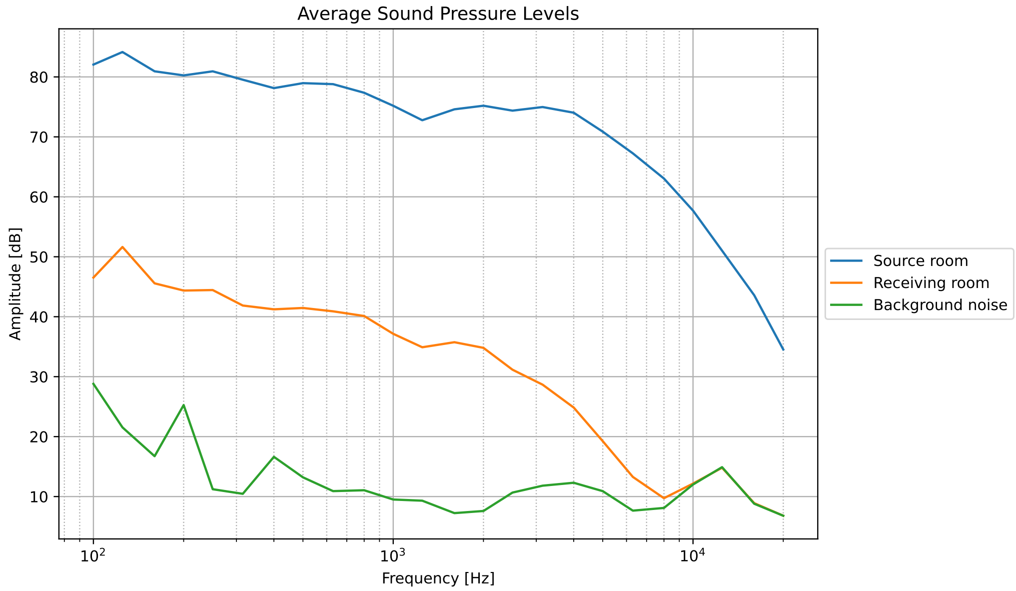
The background noise was recorded in the receiving room. From Chapter 2 we have that the background noise should be corrected if the difference between the background noise and the combined signal and background noise is below 10 dB. The difference between the background noise and the combined signal and background noise is shown in Table 4.1

Table 4.1: Uncorrected and corrected difference between the background noise in the receiving room and the sound pressure levels in the receiver room.

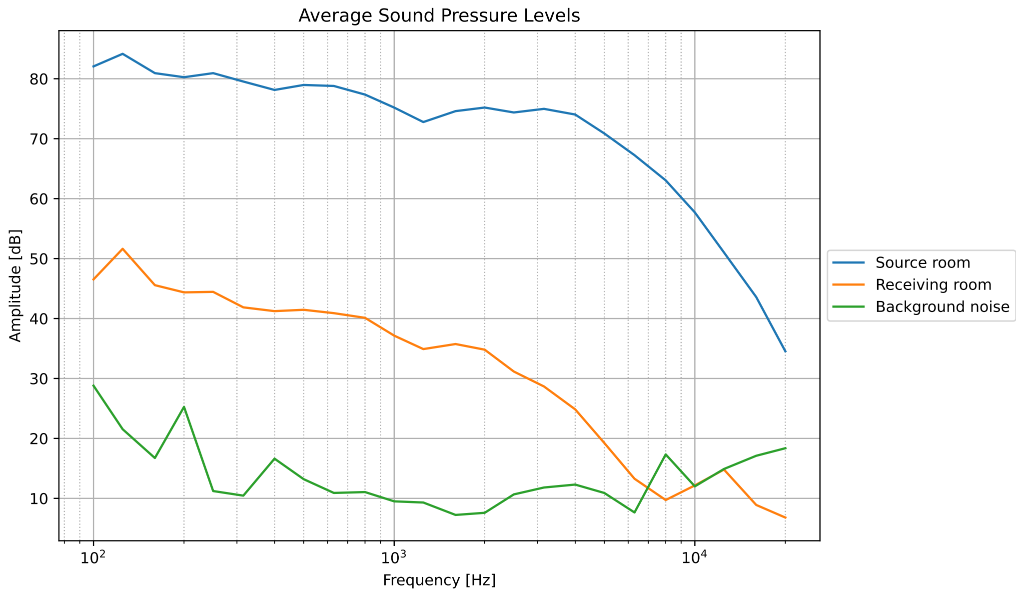
1/3-octave bands [Hz]	Uncorrected difference [dB]	Corrected difference [dB]
100	46.5	46.5
125	51.6	51.6
160	45.6	45.6
220	44.4	44.4
250	44.4	44.4
315	41.9	41.9
400	41.2	41.2
500	41.5	41.5
630	40.9	40.9
800	40.1	40.1
1k	37.1	37.1
1.25k	34.9	34.9
1.6k	35.7	35.7
2k	34.8	34.8
2.5k	31.2	31.2
3.15k	28.8	28.8
4k	24.8	24.8
5k	19.2	19.2
6.3k	13.3	13.3
8k	9.7	15.8
10k	12.2	12.2
12.5k	14.8	14.8
16k	8.9	15.6
20k	6.8	9.3

For the frequency band of 20 kHz, the difference after the first correction was 9.3 dB, so the same correction was applied one more time. For the bands 16 kHz and 8 kHz the correction was applied once.

The averaged sound pressure levels for the source room, the receiving room and the background noise of the receiving room is shown in Figure 4.1. Here we first see the uncorrected background noise and then the corrected background noise, as shown in Table 4.1. The sound pressure, for the source and the receiving room, was averaged over the five microphone positions and for the two loudspeaker positions, giving one sound pressure level for the different frequency bands for each room.



(a) Sound pressure levels for the source room, receiving room and uncorrected background noise for the receiving room.



(b) Sound pressure levels for the source room, receiving room and corrected background noise for the receiving room.

Figure 4.1: Averaged sound pressure levels for the source room, receiving room and the background noise for the receiving room, both uncorrected (a) and corrected (b).

The apparent sound reduction index R' was calculated from Equation 2.6 and is shown in Figure 4.2. Figure 4.2 also shows R_{field}, from Equation 2.5, and R_{random}, from Equation 2.4. The level difference, D from Equation 2.9, the nomalized level difference, D_n from Equation 2.10, and

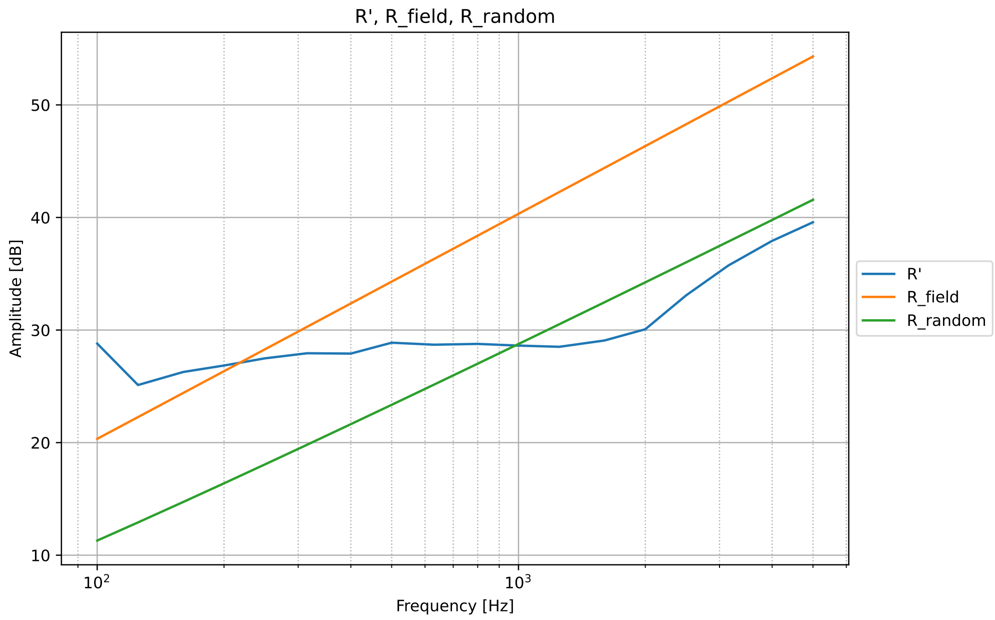


Figure 4.2: The apparent sound reduction index, R', and theoretical sound reduction indexes, R_{field} and R_{random}, for the measurements.

the standardized level difference, D_{nT} from Equation 2.11, is shown in Figure 4.3

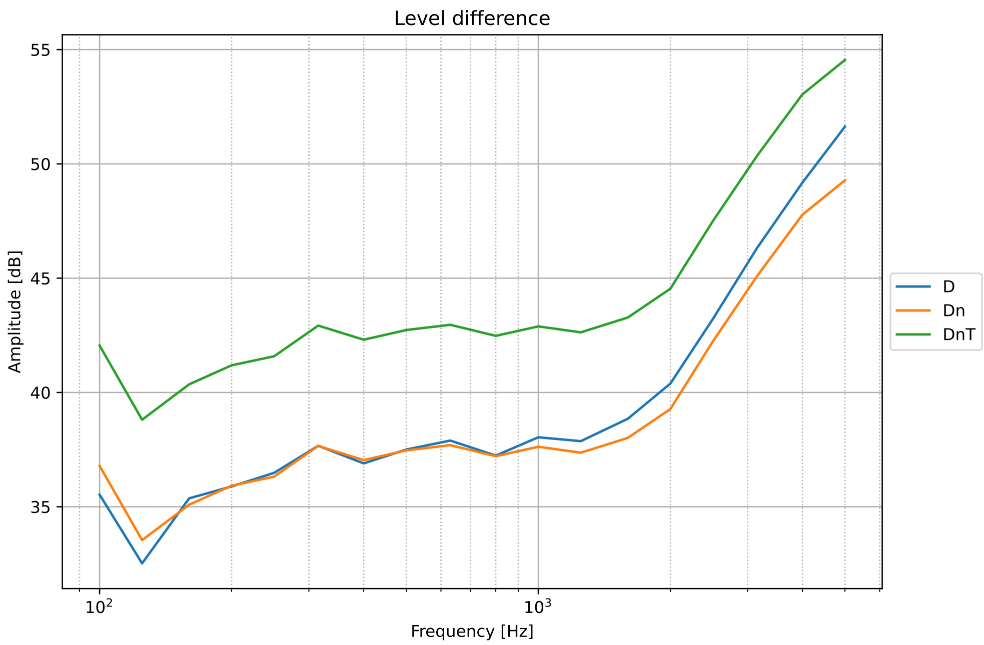


Figure 4.3: Level difference, D, normalized level difference, D_n, and standardized level difference, D_{nT} for the measurements.

In order to determine the weigthed sound reduction index, R'_W, the unfavourable deviation must be, as described in Chapter 2, as close to, but under 32 dB. Table 4.2 shows R', the unshifted reference curve, the shifted reference curve and the unfavourable deviation.

Table 4.2: Weighted sound reduction index

Frequency band [Hz]	Measured R' [dB]	Shifted reference curve [dB]	Unfavourable deviation [dB]	Unshifted reference curve [dB]
100	31.8	14	0	33
125	28.1	17	0	36
160	29.3	20	0	39
200	29.9	23	0	42
250	30.5	26	0	45
315	31	29	0	48
400	30.9	32	1.1	51
500	31.9	33	1.1	52
630	31.7	34	2.3	53
800	31.8	35	3.2	54
1k	31.5	36	3.2	55
1.25k	31.5	37	5.5	56
1.6k	32.1	37	4.9	56
2k	33.1	37	3.9	56
2.5k	36.1	37	0.9	56
3.15k	38.7	37	0	56
4k	40.9	37	0	56
5k	42.6	37	0	56

The unfavourable deviation is 27.3 dB and the shifting factor is 19 dB. This gives the weigthed reduction index $R'_W = 33$ dB, as read of from Table 4.2 at 500 Hz.

5 Discussion

The difference between the background noise and the combined signal and background noise in the receiving room was to small and needed correction for the 1/3-octave bands 8 kHz, 16 kHz and 20 kHz. The values were below 10 dB, but above 6 dB and was corrected according to 2.12. This should have very little effect on the results.

Two different theoretical estimations for the sound reduction index was presented, R_{field} and R_{random}. When comparing these two to the measured R', as shown in Figure 4.2, it is clear that the estimations deviate a bit from the measurements. R_{field} is slightly steaper than R_{random} and has generally higher values. For frequencies above around 1 kHz it seems that R_{random} is a relatively good estimate for the measured R', although the levels are generally around 2-3 dB too high. Since dB is a logrithmic scale a difference of +3 dB equals a doubling of the sound pressure level. However the slope of the R_{random} is a good estimate for the slope of R'. For the lower frequencies the curve for R' is much flatter than that for the estimates. From this it is clear that the test material that separates the rooms have a lower absorption for lower frequencies and a higher absorption for higher frequencies, as it is the higher frequencies that experience the largest reduction in sound pressure level. For the receiving room it is a visible spike for 125 Hz, indicating that this frequency band is less dampened, as also visible in Figure 4.2. R_{random} is the best estimation at higher frequencies, with generally a little too high levels, and neither of the estimations are accurat at lower frequencies.

It should also be noted that the level difference, D, the normalized level difference, D_n, and the standardized level difference, D_{nT}, follows roughly the same shape. D_{nT} uses the reverberation time and D_n uses the equivalent absorption area, from Equation 2.7 to calulate, and it seems that using the reverberation time, shown in Table A.1 in Appendix A, gives generally higher values than using the equivalent absroption area. As the rooms are reverberant it makes sense that the equivalent absorption area is relatively small and the reverberation time is larger as there is not much in the room that dissipates the sound energy.

Although this measurement procedure leaves little room for error, some aspects could still affect the results. It was not possible to close the doors to the reverberants rooms, as to not damage a microphone cable. The doors were closed as much as possible with regards to the cable. It was difficult to ensure that the doors were exactly the same amount of open/closed for each measurement, as the microphone positions and the loud speaker configurations had to be changed manually inside the rooms. For the measurements in the receiving room a person was inside the source room for most of the measurements, to make the changing of the loud speaker configuration quicker. This could have caused a slightly higher absorption area in the source room. Considering the source room is so large with many reflecting surfaces, the effect of this should be negliable, but in order to get very reliable results should either the same person be present inside the source room for every measurement, or for none of the measurement. It is difficult to say if this has affected the results, as the sound pressure levels were averaged for all microphone positions inn the same room, and the author did not note for which of the measurements this was the case. However, the effect should be negliable, and the results seem reasonable. The measurements of the reverberation time was corrupted and data from another measurement earlier the same day was used. It is hard to say what the possible sources of error could be, but the main thing would be the how closed the doors were. Assuming that the sound analyser was calibrated correctly the results should be usable, and it seems again that the results are reasonable and that the data set is valid. Of course, given that the background noise data from the same measurements as the reverberation time is not available, it is possible that the background noise was higher, which could give problems with the SNR, but it seems that this is not the case.

6 Conclusions

The sound reduction index of a two layered test material consisting of equal parts rubber and plywood was calculated. This was done by measuring the sound pressure levels in a source and a receiving room, which both were reverberant rooms. The reverberation time and background noise was also measured in order to calculate this. The test material is found to dampen high frequencies effectively, and dampens lower frequencies less effectively. The best theoretical estimation is R_{random}, but only for higher frequencies. Its levels are too high, but i follows the slope of R' relatively good. For the lower frequencies neither are good estimations. The weigthed sound reduction index was found to be 33 dB. The measurement method does not leave much room for error, but some sources of error was suggested, such as the doors being unable to close properly due to a microphone cable and people in the source room during some of the measurements. This is however not suspected to affect the results significantly.

Bibliography

[1] Rørstadbotnen, R. A., Sound Insulation Measurement, 2022-03-18

[2] Dutilleux, G., Sound Insulation Measurement, 2020-03-03

[3] NS-EN ISO 717-1:2013, Akustikk Vudering av lydisolasjon i bygninger og av bygningsdeler, Del 1: Luftlydisolasjon (ISO 717-1:2013), 2013-06-01

[4] TTT4250 Acoustical Measurement Techniques Laboratory Exercise Compendium, Chapter 5

[5] NS-EN ISO 16283-1:2014, Akustikk Feltmåling av lydisolasjon i bygninger og av bygningsdeler, Del 1: Luftlydisolasjon (ISO 16283-1:2014), ICS 91.120.20

A Appendix: Reverberation time T30

Table A.1: Reverberation time, $T30$, for the receiving room.

Frequency bands [Hz]	Reverberation time, $T30$, [s]
50	3.25
63	2.98
80	2.85
100	2.16
125	2.12
160	1.60
200	1.73
250	1.63
315	1.63
400	1.68
500	1.66
630	1.59
800	1.67
1k	1.55
1.25k	1.52
1.6k	1.42
2k	1.32
2.5k	1.35
3.15k	1.25
4k	1.19
5k	0.97

B Appendix: Python script

```python
import matplotlib.pyplot as plt
import numpy as np


A0=10
T0=0.5
c = 343
rho_s = (15.33+10.08)
rho = 1.2041
S = 1.21*1.18
octave_band = [100, 125, 160, 200, 250, 315, 400, 500, 630, 800, 1000, 1250, 1600, 2000, 2500, 3150, 4000, 5000, 6300, 8000, 10000, 12500, 16000, 20000]
octave_band = np.array(octave_band)
octave_band_short = [100, 125, 160, 200, 250, 315, 400, 500, 630, 800, 1000, 1250, 1600, 2000, 2500, 3150, 4000, 5000]
octave_band_short = np.array(octave_band_short)
omega = octave_band* int(2*np.pi)
omega = np.array(omega)
omega_short = octave_band_short* int(2*np.pi)
print("omega short", omega_short)
omega_short = np.array(omega_short)
A_weighting = np.array([-16.1, -8.6, -3.2, 0, 1.2, 1, -1.1, -4])
x_ticks_octaveband = ["100", "125", "160", "200", "250", "315", "400", "500", "630", "800", "1k", "1.25k", "1.6k", "2k", "2.5k", "3.15k", "4k", "5k", "6.3k", "8k", "10k", "12.5k", "16k", "20k"]
x_ticks_octaveband_short = ["100", "125", "160", "200", "250", "315", "400", "500", "630", "800", "1k", "1.25k", "1.6k", "2k", "2.5k", "3.15k", "4k", "5k"]
A_weighting_modded = A_weighting[:-1]
mic_pos = 5

MEASUREMENT = {
    "LFEQ": 0,
    "LFMAX": 1,
    "LFMIN": 2,
    "LFE": 3
}

unmodded_file_ref = "pressure.csv"
reverb_file = "reverberation.csv"


names_for_plot = ['Mic pos 1', 'Mic pos 2', 'Mic pos 3', 'Mic pos 4', 'Mic pos 5', 'Background noise']
```

```python


def read_csv(filename, skipheader, skipfooter):
    return np.genfromtxt(filename, skip_header=skipheader,
        skip_footer = skipfooter, delimiter=';')


# Array of type:  [ROWS, OCTAVEBANDS]
def pl tt(array, title,name_of_file, use_A_Weight=True, short=
    False) ->
    object:# limit,
    fig, ax = plt.subplots()
    for i in  range(array.shape[0]):
        if not short:
            ax.semilogx(octave_band, array[i] + (A_weighting * np.
                array(use_A_Weight).astype(
                int)),
                            label=f"Mic pos {i + 1}")
        else:
            ax.semilogx(octave_band[:-1], array[i] + (A_weighting *
                 np.array(use_A_Weight).astype(
                int))[:-1],
                            label=f"Mic pos  {i + 1}")
    ax.grid(which="major")
    ax.grid(which="minor", linestyle=":")
    ax.set_xlabel("Frequency [Hz]")
    ax.set_ylabel("Amplitude [dB]")
    ax.set_title(title)
    if not short:
        ax.set_xticks(octave_band)
        ax.set_xticklabels(x_ticks_octaveband)
    else:
        ax.set_xticks(octave_band[:-1])
        ax.set_xticklabels(x_ticks_octaveband[:-1])
    plt.savefig(name_of_file)
    plt.show()

def pl tt_multi(n_m_array, title, name_of_file, lengend_array):
    assert  len(n_m_array.shape) == 2  # hvis denne failer har du
        en funky array
    fig, ax = plt.subplots()

    for i in  range(n_m_array.shape[0]):
        ax.semilogx(octave_band, n_m_array[i],
                    label=f"{lengend_array[i]}")

    ax.grid(which="major")
    ax.grid(which="minor", linestyle=":")
    ax.set_xlabel("Frequency [Hz]")
    ax.set_ylabel("Amplitude [dB]")
    ax.set_title(title)
    plt.legend(bbox_to_anchor = (1,0.5), loc="center left")
    plt.savefig(name_of_file)
    plt.show()
```

```

def pl tt_leveldiff(n_m_array, title, name_of_file, lengend_array):
    assert len(n_m_array.shape) == 2  # hvis denne failer har du en funky array
    fig, ax = plt.subplots()

    for i in range(n_m_array.shape[0]):
        ax.semilogx(octave_band_short, n_m_array[i],
                    label=f"{lengend_array[i]}")

    ax.grid(which="major")
    ax.grid(which="minor", linestyle=":")
    ax.set_xlabel("Frequency [Hz]")
    ax.set_ylabel("Amplitude [dB]")
    ax.set_title(title)
    plt.legend(bbox_to_anchor = (1,0.5), loc="center left")
    plt.savefig(name_of_file)
    plt.show()


def pl tt_simple(array, title,name_of_file):
    fig, ax = plt.subplots()
    ax.semilogx(octave_band, array)
    ax.grid(which="major")
    ax.grid(which="minor", linestyle=":")
    ax.set_xlabel("Frequency [Hz]")
    ax.set_ylabel("Amplitude [dB]")
    ax.set_title(title)
    # plt.legend(loc="lower right")
    plt.savefig(name_of_file)
    plt.show()

def pl tt_soundreduction(array, title,name_of_file):
    fig, ax = plt.subplots()
    ax.semilogx(octave_band_short, array)
    ax.grid(which="major")
    ax.grid(which="minor", linestyle=":")
    ax.set_xlabel("Frequency [Hz]")
    ax.set_ylabel("Amplitude [dB]")
    ax.set_title(title)
    # plt.legend(loc="lower right")
    plt.savefig(name_of_file)
    plt.show()

def pl tt_dn(n_m_array, title, name_of_file, lengend_array):
    assert len(n_m_array.shape) == 2  # hvis denne failer har du en funky array
    fig, ax = plt.subplots()

    for i in range(n_m_array.shape[0]):
        ax.semilogx(octave_band_short, n_m_array[i],
                    label=f"{lengend_array[i]}")
```

```
    ax.grid(which="major")
    ax.grid(which="minor", linestyle=":")
    ax.set_xlabel("Frequency [Hz]")
    ax.set_ylabel("Amplitude [dB]")
    ax.set_title(title)
    plt.legend(bbox_to_anchor = (1,0.5), loc="center left")
    plt.savefig(name_of_file)
    plt.show()


def pl tt_theory(n_m_array, title, name_of_file, lengend_array):
    # assert len(n_m_array.shape) == 2  # hvis denne failer har du en funky array
    fig, ax = plt.subplots()

    for i in  range(n_m_array.shape[0]):
        ax.semilogx(octave_band_short, n_m_array[i],
                    label=f"{lengend_array[i]}")

    ax.grid(which="major")
    ax.grid(which="minor", linestyle=":")
    ax.set_xlabel("Frequency [Hz]")
    ax.set_ylabel("Amplitude [dB]")
    ax.set_title(title)
    plt.legend(bbox_to_anchor = (1,0.5), loc="center left")
    plt.savefig(name_of_file)
    plt.show()





def db_to_pressure(measurements):
    return 10 ** (measurements /10)


def pressure_to_db(measurements):
    return 10 * np.log10(measurements)

def R(L1,L2,S,A):
    return L1-L2 + 10 *np.log10(S/A)

def R_random(R0):
    return R0-10*np.log10(R0)

def R_field(R0):
    return R0 - 5

def R_marked(D, S, A):
    return D+ 10*np.log10(S/A)

def R0():
```

```
    return 10*np.log10(1+((omega_short *rho_s)/(2*rho*c))**2)

def D(L1,L2):
    return L1-L2

def Dn(D,A):
    return D-10*np.log10(A/A0)

def equivalent_absorption_area(V, T):
    return (0.16*V) /T

def DnT(D, T):
    return D+10*np.log10(T/T0)

def average_spl(data):
    meas_pressure = db_to_pressure(data)
    pressure_average = np.average(meas_pressure, axis = 0)
    avg_db = pressure_to_db(pressure_average)
    return avg_db

def average_spl_two_arrays(data1, data2):
    data1_pressure = db_to_pressure(data1)
    data2_pressure = db_to_pressure(data2)
    data_total_pressure = np.vstack((data1_pressure,data2_pressure))
    average_pressure = np.average(data_total_pressure, axis = 0)
    average_db_data = pressure_to_db(average_pressure)
    return average_db_data

def background_noise_correction(background, data):
    return 10*np.log10(10**((background+data)/10)-10**((background)/10))

# Reading the files
data = read_csv(unmodded_file_ref, skipheader=1, skipfooter=0)[:,4:]  # All rows minus the last, header is fucked anyway, fuck the first 4 colums
print("data shape", data.shape)
print("data", data)

reverb_data = read_csv(reverb_file, skipheader=3, skipfooter=9)[3:,[4]] #Avgerage revberation time per 1/3-octave band
print("reverb", reverb_data)


bg_noise = data[[10, 13], :]
print("bg noise shape", bg_noise.shape)

source_lower = data[[0,3,4,7,8], :]
print("source lower shape", source_lower.shape)

source_upper = data[[1,2,5,6,9], :]
print("source upper shape", source_upper.shape)
```

```
receiver_lower = data[[12,14,17,18,21], :]
print("receiver lower shape", receiver_lower.shape)

receiver_upper = data[[11,15,16,19,20], :]
print("receiver upper shape", receiver_upper.shape)



# Trash the first 15 columns as they are not relevant for the
    octavebands
# Split into microphone positions, measurement and octavebands
source_upper_splitted = source_upper[:, 15:].reshape(-1, 4, 33)[:,
    :,9:]  # The 9 first octave bands are invalid due to measurement
     equipment limitation
print("source upper slitted", source_upper_splitted.shape)
source_lower_splitted = source_lower[:, 15:].reshape(-1, 4, 33)[:,
    :, 9:]
receiver_upper_splitted = receiver_upper[:, 15:].reshape(-1, 4, 33)
    [:, :, 9:]
receiver_lower_splitted = receiver_lower[:, 15:].reshape(-1, 4, 33)
    [:, :, 9:]
background_splitted = bg_noise[:, 15:].reshape(-1, 4, 33)[:, :, 9:]

source_upper_splitted_short = source_upper[:, 15:].reshape(-1, 4,
    33)[:, :,9:-6]  # The 9 first octave bands are invalid due to
    measurement equipment limitation
print("source upper slitted", source_upper_splitted_short.shape)
source_lower_splitted_short = source_lower[:, 15:].reshape(-1, 4,
    33)[:, :, 9:-6]
receiver_upper_splitted_short = receiver_upper[:, 15:].reshape(-1,
    4, 33)[:, :, 9:-6]
receiver_lower_splitted_short = receiver_lower[:, 15:].reshape(-1,
    4, 33)[:, :, 9:-6]
background_splitted_short = bg_noise[:, 15:].reshape(-1, 4, 33)[:,
    :, 9:-6]




to_plot_source_upper =  source_upper_splitted [:, MEASUREMENT["LFEQ
    "], :]
to_plot_source_lower = source_lower_splitted[:, MEASUREMENT["LFEQ"
    ], :]
to_plot_receiver_upper= receiver_upper_splitted[:, MEASUREMENT["
    LFEQ"], :]
to_plot_receiver_lower = receiver_lower_splitted[:, MEASUREMENT["
    LFEQ"], :]
to_plot_receiver_lower = receiver_lower_splitted[:, MEASUREMENT["
    LFEQ"], :]
to_plot_back = background_splitted[:, MEASUREMENT["LFEQ"], :]

```

```
to_plot_source_upper_short =  source_upper_splitted_short [:, MEASUREMENT["LFEQ"], :]
to_plot_source_lower_short = source_lower_splitted_short[:, MEASUREMENT["LFEQ"], :]
to_plot_receiver_upper_short= receiver_upper_splitted_short[:, MEASUREMENT["LFEQ"], :]
to_plot_receiver_lower_short = receiver_lower_splitted_short[:, MEASUREMENT["LFEQ"], :]
to_plot_receiver_lower_short = receiver_lower_splitted_short[:, MEASUREMENT["LFEQ"], :]
to_plot_back_short = background_splitted_short[:, MEASUREMENT["LFEQ"], :]


##################
# Method 1
##################

# Averaged SPL
average_upper_source = average_spl(to_plot_source_upper)
average_lower_source = average_spl(to_plot_source_lower)
average_upper_receiver = average_spl(to_plot_receiver_upper)
average_lower_receiver = average_spl(to_plot_receiver_lower)
average_background = average_spl(to_plot_back)

average_source= average_spl_two_arrays(average_upper_source, average_lower_source)
average_receiver = average_spl_two_arrays(average_lower_receiver, average_upper_receiver)
average_source_short= average_spl_two_arrays(average_upper_source, average_lower_source)
average_receiver_short = average_spl_two_arrays(average_lower_receiver,average_upper_receiver)

averaged_vector = np.vstack((average_upper_source, average_lower_source,average_upper_receiver, average_lower_receiver,average_background))
averaged_vector_correct = np.vstack((average_source, average_receiver,average_background))


pl tt_multi(averaged_vector_correct,title="Average Sound Pressure Levels", name_of_file="avg_spl_all.pdf", lengend_array=["Source room", "Receiving room", "Background noise"])


average_upper_source_short = average_spl(to_plot_source_upper_short)
average_lower_source_short = average_spl(to_plot_source_lower_short)
average_upper_receiver_short = average_spl(to_plot_receiver_upper_short)
```

```
average_lower_receiver_short = average_spl(to_plot_receiver_lower_short)
average_background_short = average_spl(to_plot_back)

########################
# Background noise
########################

bg_and_receving = db_to_pressure(average_receiver)+db_to_pressure(average_background)
diference_bgnoise =db_to_pressure(average_background)-bg_and_receving


average_background[23] = background_noise_correction(average_background[23],average_receiver[23])
average_background[23] = background_noise_correction(average_background[23],average_receiver[23])
average_background[22] = background_noise_correction(average_background[22],average_receiver[22])
average_background[19] = background_noise_correction(average_background[19],average_receiver[19])

bg_and_receving_after = db_to_pressure(average_receiver)+db_to_pressure(average_background)
diference_bgnoise_after =db_to_pressure(average_background)-bg_and_receving


averaged_vector_correct = np.vstack((average_source,average_receiver,average_background))

pl tt_multi(averaged_vector_correct,title="Average Sound Pressure Levels", name_of_file="avg_spl_all.pdf", lengend_array=["Source room", "Receiving room", "Background noise"])


#####################################################
# Equivalent sound absorption area receiving room
#####################################################

height = 5.58
width = 4.11
length = 4.58
volume_receiving_room = height*width*length
# stigning = 30/reverb_data
# T60 = 60/stigning
# T60 = np.array(T60).squeeze()
T60 = reverb_data.squeeze()



####################################
```

```
# Calculating T60 from T30
####################################

absorp_t60 = equivalent_absorption_area(volume_receiving_room,T60)
absorp_t60 = absorp_t60.squeeze()




################################################################
# Calculating level difference between the source room and the receiving room
################################################################
D_upper = D(average_upper_source_short,average_upper_receiver_short)
D_lower = D(average_lower_source_short,average_lower_receiver_short)

D_calc = D(average_source, average_receiver)
D_calc_short = D_calc[:-6]

#########################################################
# Calculating normalized level difference Dn (receiver)
#########################################################
Dn_upper = Dn(D_upper,absorp_t60)
Dn_lower = Dn(D_lower, absorp_t60)



Dn_calc = average_spl_two_arrays(Dn_lower,Dn_upper)

DnT_upper = DnT(D_upper, T60)
DnT_lower = DnT(D_lower, T60)

DnT_calc = average_spl_two_arrays(DnT_upper,DnT_lower)


leveldiff_vec = np.vstack((D_calc_short, Dn_calc, DnT_calc))
pl tt_leveldiff(leveldiff_vec,title="Level difference", name_of_file="leveldiff.pdf", lengend_array=["D", "Dn", "DnT"])

#####################################################
# Sound Reduction Index
#####################################################
sound_red_index = R_marked(D_upper,S, absorp_t60)
pl tt_soundreduction(sound_red_index,"Sound Reduction Index R", name_of_file="sound_red_index.pdf")




###################################################
# Calculating R, R_marked, R_field, R_random
```

```
#################################################

sound_red_index_test = R_marked(D_upper,S,absorp_t60.squeeze())
pl tt_soundreduction(sound_red_index_test,"Sound Reduction Index", "sound_red_t60.pdf")


sound_R0 = R0()
sound_red_field= R_field(sound_R0)
print("field",sound_red_field.shape)
sound_red_random = R_random(sound_R0)
print("random",sound_red_random.shape)

print("sound red index", sound_red_index.shape)
theoretical_R = np.vstack((sound_red_index,sound_red_field, sound_red_random))
pl tt_theory(theoretical_R,"R', R_field, R_random", "theory_reduction.pdf",["R'", "R_field", "R_random"])
```