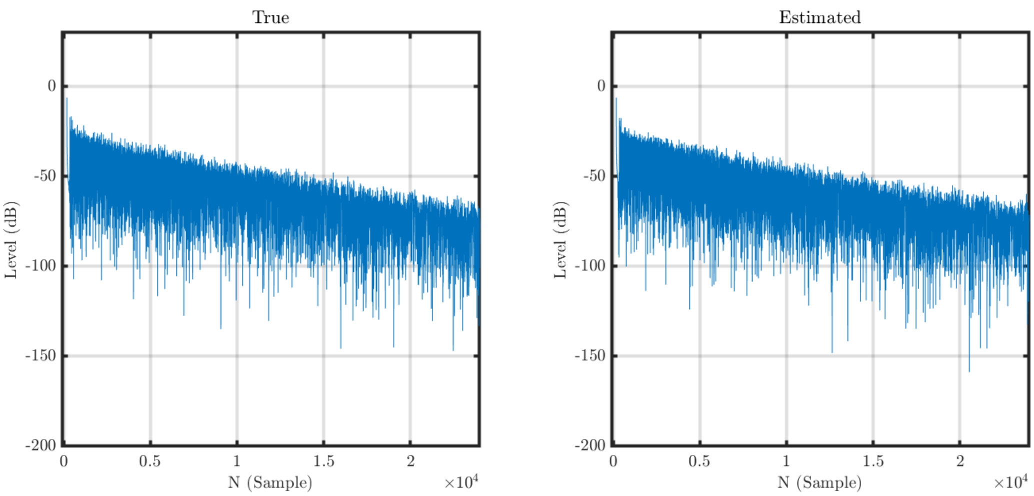
CHALMERS
UNIVERSITY OF TECHNOLOGY

Acoustic Impulse Response Prediction Using Regularized Regression Models

Master's thesis in Sound and Vibration Engineering

Karin Hulling



DEPARTMENT OF ARCHITECTURE AND CIVIL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

MASTER'S THESIS ACEX30 2021

Acoustic Impulse Response Prediction Using Regularized Regression Models

KARIN HULLING

CHALMERS
UNIVERSITY OF TECHNOLOGY

Division of Applied Acoustics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Acoustic Impulse Response Prediction Using Regularized Regression Models
KARIN HULLING

Supervisor: Piotr Biliński, University of Warsaw
Examiner: Jens Ahrens, Applied Acoustics at Chalmers University of Technology

Master's Thesis 2021
Division of Applied Acoustics
Chalmers University of Technology

Cover: Estimation of an impulse response using the LASSO model. To the left is the true impulse response and to the right is the estimation.

Typeset in LaTeX
Digital
Gothenburg, Sweden 2021

Acoustic Impulse Response Prediction Using Regularized Regression Models
KARIN HULLING
Division of Applied Acoustics
Chalmers University of Technology

Abstract

Different methods of calculating the impulse response of rooms have been developed over the years and is an important measure within e.g. room acoustics and building acoustics. In this thesis work, two different regularized regression models were evaluated to see if it is possible to estimate the rooms' impulse response simply by knowing some features of the room. This way, the room impulse response would be obtained in a much more simple and straightforward way compared to the already existing methods.

The two models that were evaluated in this thesis work are called LASSO (Least Absolute Shrinkage and Selection Operator) and ridge regression, these models both generate a weight vector, given a training set and a target. Considering a large data set of different room properties as the training set, the target describes a new set of room properties. The aim was to be able to describe the target as the superposition of the training set multiplied with the weight vector, generated by the models. This weight vector was then tested to see if the relationship also could be applied to the impulse responses, by the superposition of the impulse responses in the training set multiplied with the same weight vector.

One difference between the two models is that the LASSO model can shrink the coefficients belonging to the less important features to zero, while for ridge regression, the coefficients can only get close to zero. This is what encourages sparsity in the LASSO model, which turned out to be a winning concept.

Results showed that the LASSO model estimated room impulse responses around 20 dB better than the ridge regression model. The results also show that there is potential for these models with some adjustments within the model, but also by weighing the features in order of importance.

Keywords: Room impulse response, LASSO, Ridge regression, Acoustics, Regression models, MCRoomSim, Pyroomacoustics, Sparse representation

Acknowledgements

I want to thank my examiner and supervisor Jens Ahrens for his support and guidance with the structure and acoustic related parts throughout this thesis work. I would also like to thank Piotr Biliński for his support and knowledge within sparse representation and regularized regression models. Last but not least I would like to thank my two opponents Aditya Thombare and Yanan Liu.

Karin Hulling, Gothenburg, June 2021

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

Within room acoustics the main parameters affecting the impulse response of a room, are size and shape of the room as well as the absorption and scattering coefficients of the interior surfaces. A common method used to calculate the reverberation time of a room is to excite the room with either an impulse (deterministic impulse method) or an interrupted broadband noise (interrupted noise method) and measure the time it takes for the sound pressure level to drop 60 dB [7], [6]. In room simulators, concepts such as image source modeling and ray tracing are often used and have grown to be successful methods in order to estimate the room impulse response [4]. However, these methods can become very computationally heavy.

Imagine instead being able to find the impulse response simply by knowing the main parameters describing the room.

In this thesis the way of trying to achieve this by using two kinds of regularized regression models, will be explored. Assuming that the variables that, in a good way, can describe a room are known, the amount of input variables are quite many. In fact, if considering seven center frequencies, the amount of input variables is 93. Three variables describing the dimensions of the room, three variables describing the position of the sound source and three variables describing the position of the receiver (microphone). The rest of the input variables represent the absorption and scattering coefficients which are defined for six surfaces and seven center frequencies, resulting in $6 \cdot 7 \cdot 2 = 84$ variables. Together with the nine variables describing the room dimensions, source and receiver position this results in $9 + 84 = 93$ input variables.

1.2 Purpose

The purpose of this thesis is to be able to describe the impulse response of a room by only knowing a set of room properties such as room dimensions, source position, receiver position, absorption coefficients and scattering coefficients.

1.3 Aim

There are a few ways of determining an impulse response of a room, in this thesis there will be a focus on regularized regression models. By finding a relationship between the training set and the target containing the room properties, the same relationship will be applied to the training set containing the corresponding room impulse responses and in that way obtain the estimated room impulse response.

1.4 Limitations

During the room simulation process of this work, the room properties are limited. The width of the rooms will range from 6 m to 7 m with a step size of 0.5 m, while the depth of the room will consistently be 5 m and the height 3 m. This also implies that the rooms are of shoe-box character. The absorption coefficients will be limited in such a way that the reverberation time stays within the range of 0.3 s to 1 s. Because of this, the results of this thesis work will not be applicable for all sorts of rooms.

1.5 Previous and related work

The main inspiration to this thesis, is a work in which the authors used the LASSO model in order to estimate the HRTF (Head Related Transfer Function), using anthropometric features as input data [11]. The idea was that the same sparse combination of the input data, could be used to describe the magnitudes of an HRTF. By learning a sparse vector to represent a set of anthropometric features as the linear superposition of the anthropometric features of a training set, the same sparse vector could be used on the HRTF data. Among other models that where evaluated, the outcome of this study shows that the best results were obtained when using the LASSO-approach. Therefore, in this thesis work, the same concept will be evaluated, since the structure of the data is somewhat similar.

2
Theory

The following chapter will cover a recapitulation of the theory of some acoustics, the theory of data and model preparations and regularized regression model techniques that are required to understand in order to follow this thesis throughout.

2.1 Acoustics

2.1.1 Absorption and Scattering

Absorption coefficients, α, holds a value between 0 and 1, where 0 corresponds to total reflection and 1, total absorption [5, Chapter 6].

Scattering coefficients is a measure of the amount of energy that is reflected in a diffuse manner on a surface due to roughness or unevenness. The scattering coefficients holds a number between 0 and 1, where 0 indicates total specular reflection (90° angle between incoming and reflected sound wave) and 1 totally diffuse reflection as can be seen in Figure 2.1.

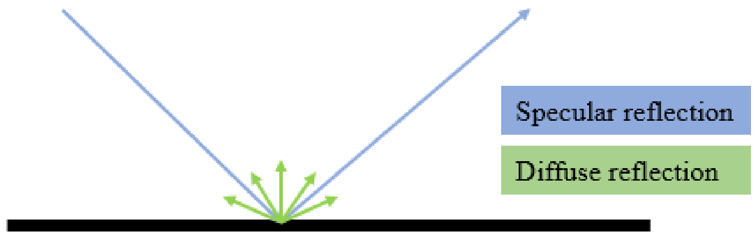


Figure 2.1: Considering the black line as a surface, such as a wall, the blue left hand side line represents the incoming sound wave and the blue line to the right hand side the reflected specular sound wave. It can be seen that the angle between the incident and reflected wave is 90°. The green arrows represents the diffuse reflection of the incident wave.

2.1.2 Image Source Model and Ray Tracing

The image source model (ISM) is a technique that is used to obtain a rooms' impulse response in a simple and straight forward way and lies under the category of geometrical acoustics. Instead of tracing each ray from source to receiver through reflections of the interior surfaces of the room, so called mirror sources can be found and the rays from the image source can be traced to the receiver as shown in Figure 2.2. The reflection order describes the number of times a reflection has been mirrored. In the figure the reflection order is displayed with a number in each of the mirrored rooms.

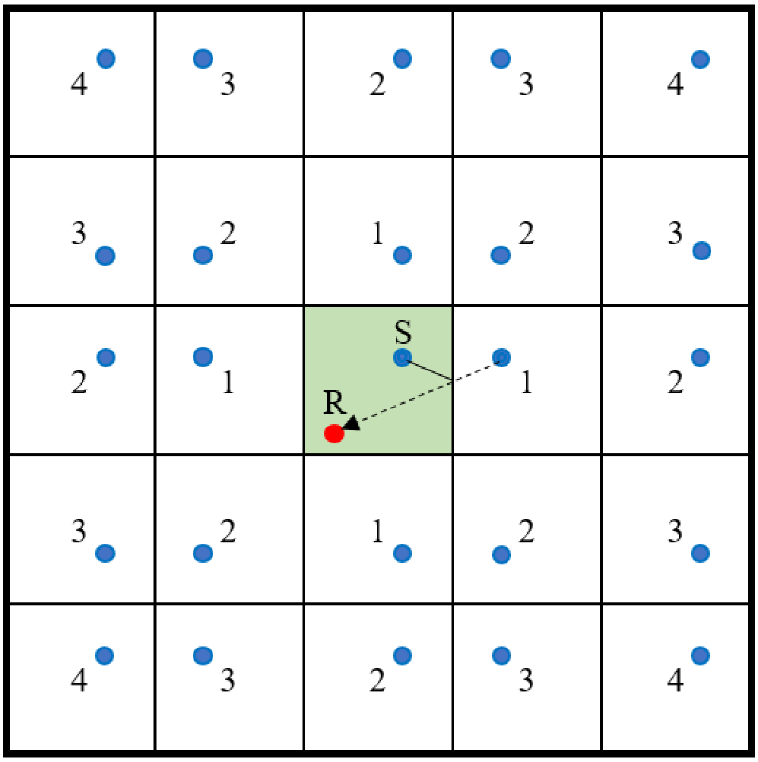


Figure 2.2: 2D visualization of the image source model concept. "R" (red circle) stands for receiver and "S" (blue circle) for the source. The square colored in green represents the room and the surrounding white squares represents the mirrored rooms. The outdrawn lines shows a reflection of order 1.

In a truly specular reflection, the squared pressure of the reflected wave is the squared pressure of the incoming wave multiplied with $1 - \alpha$, meaning the amount that is not absorbed by the material of the surface

$$p^2_{\text{refl}} = p^2_{\text{in}}(1 - \alpha) \tag{2.1}$$

The total squared pressure of e.g. the direct sound and the first reflected sound is obtained by uncorrelated addition of the squared rms-values

$$p^2_{tot} = W\rho_0 c \left(\frac{1}{4\pi R_0^2} + \frac{1}{4\pi R_1^2} \right) \tag{2.2}$$

where W is the sound power of the source, ρ_0 the density of the medium which the sound is traveling through, c the speed of sound, R_0 the distance between the source and receiver (direct sound) and R_1 the distance between the image source and receiver (first reflection)[8, Section 4.2].

2.1.3 Schröder's Backward Integration

The reverberation time of a room is defined as the time it takes for the level of the total squared pressure to drop 60 dB. The difficulty with this approach is that the pressure level fluctuates due to interference and resonating frequencies that appear and makes it hard to determine the exact time for the decay of 60 dB. A way to circumvent this was discovered by M.R. Schröder, and the method is now called the Schröder's Backward Integration [10]. The Schröder's backward integration is a method of determining the envelope of the decaying squared pressures by backward integration. Thus, the pressure, p, is squared and integrated backwards as

$$S = 10\log\left[\frac{1}{C_{\text{ref}}}\int_t^{\infty} p^2(t)dt\right] \tag{2.3}$$

where C_{ref} is a chosen constant [1]. This way it is possible to find the envelope, S, and determine the decay time. Figure 2.3 shows an example of a Schröder curve and its' belonging impulse response where the envelope is calculated for the reverberant field of the impulse response, i.e. not including the direct sound. It is then normalized to its' first value, hence the curve starts at 0.

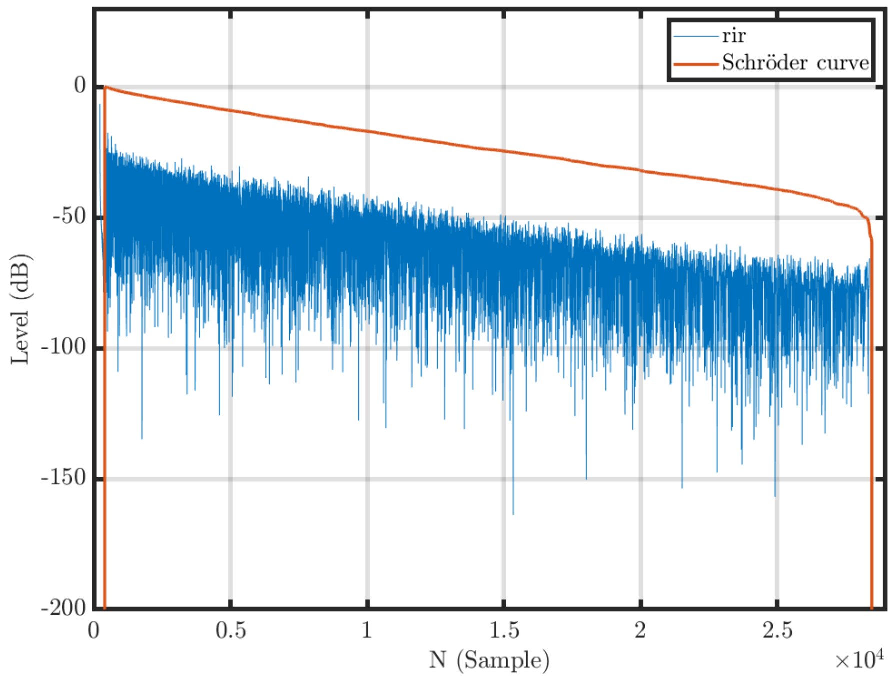


Figure 2.3: The figure shows a plot of an impulse response in dB (blue) and its' belonging Schröder curve (red). The Schröder curve is normalized with respect to it's first value.

2.2 Data and Model Preparations

2.2.1 Training and Test set

For a model to learn to generate a certain output given an input, it requires data to train with. This part of the data is called the training set, which in comparison to the test set should be large. The model is fit to the training set and the test set is used to evaluate the error of the output created by the model. In a real life scenario, the test set is unknown and before the model is used in such case, it should be evaluated and tuned in a way such that the prediction is as close to the true output as possible [17].

There is no general rule on how to split the data, the training set however, must be large enough to cover as many variations within the whole data set as possible. If the test set is very small, the results only show how good or bad the model predicts a certain type of data.

A data set can be represented by a matrix of size $N \times F$, where N is the amount of data examples and F the number of features. If the test set is considered as 10% of the data, the size will be $N \cdot 0.1 \times F$ and the training set $N \cdot 0.9 \times F$.

2.2.2 Normalization

To better represent a set of features, the data can be scaled with a min-max normalization as

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2.4}$$

which will result in features with a value between 0 and 1. The matrix containing the features for all data sets can then be presented as $\boldsymbol{X} \in [0, 1]^{N \times F}$.

2.2.3 Overfitted Model

Within the data set, each example is unique in its' own way. When a model becomes overfitted, it can effectively model the random uniqueness of the training set but fails to generalize for a broader data set meaning that it can easily fail to predict the test set. Another problem occurs when the model, rather than describing the relationship within the data, begins to describe random errors. These two in combination, are not very helpful when it comes to predicting data. Figure 2.4 shows an example of data points in two dimensions, where the red curve represents an overfitted model and the blue line represents a model which is fitted to the data points in a more generalized way. The overfitted model can effectively represent the training examples but fails with the test examples, in other terms this means that the model has low bias but high variance. The generalized model can better represent both the training and test examples, it has more bias but lower variance [9].

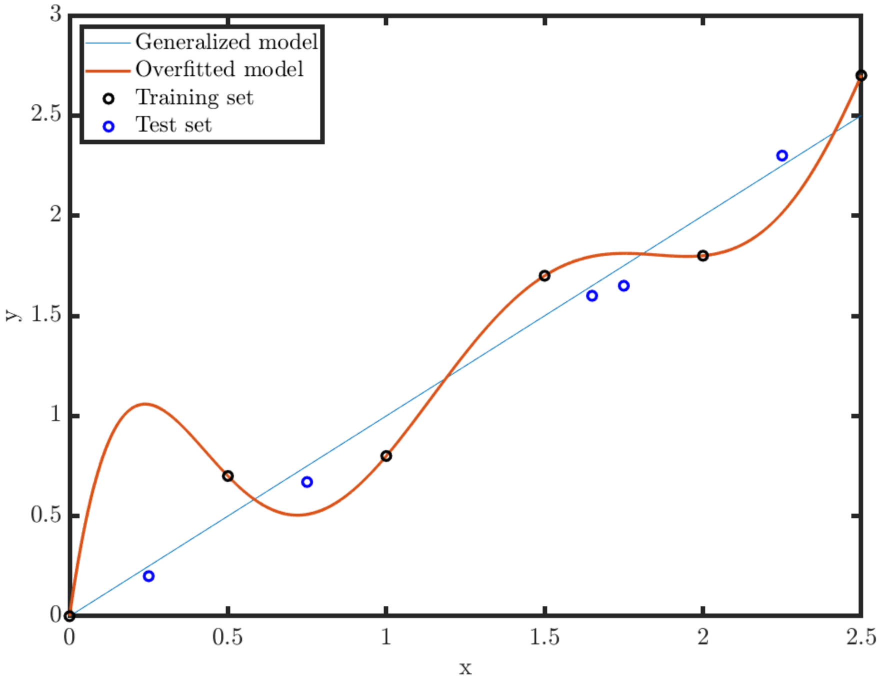


Figure 2.4: Visualization of an overfitted model (red curve) and a more generalized model (blue line). The overfitted model is fitting the training examples very well, but fails to predict the test examples.

2.2.4 Cross-validation

Presented in Figure 2.5 is a cross-validation scheme with five folds. One fold contains one test set (red box in the figure) and the rest the training set. When using cross-validation the model is trained several times, each time with a different training and test data setup. The idea of cross-validation is to train and test the model with different parts of the data. The model is then described as the average of all the iterations. This could result in a more balanced model, which in turn prevents overfitting [15, Section 3.1].

The number of folds used during cross-validation can vary by the choice of the user. The largest number of folds possible, is the amount of examples present in the data, this is called leave-one-out cross-validation.

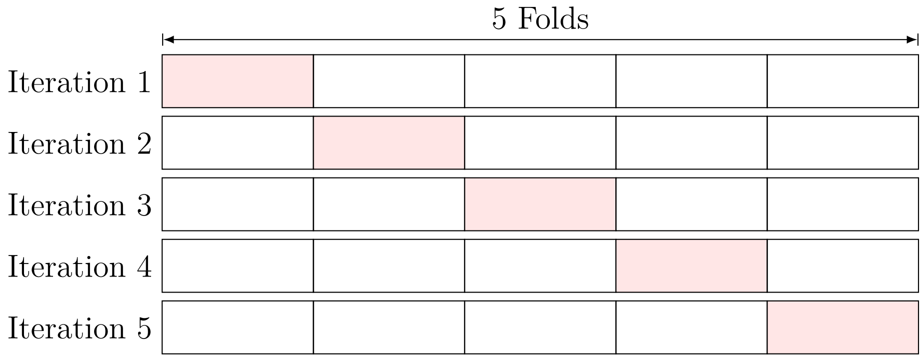


Figure 2.5: Illustration of cross-validation. The red segment represents the test set and the white segments the training sets. Each fold is individually used to create a model. The resulting model is the average of all iterations.

2.3 Regularized Regression Models

2.3.1 Sparse Representation and LASSO

LASSO is an abbreviation for the least absolute shrinkage and selection operator and is a regression model that uses ℓ_1 regularization technique. The model finds a vector, $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_N]$, which holds weight values such that a new set of features, $\boldsymbol{y}$, can be estimated as $\boldsymbol{y} = \boldsymbol{\beta}^T \boldsymbol{X}$, where $\boldsymbol{X}$ contains the N number of training sets. This means that $\boldsymbol{y}$ is generated as a linear superposition of the N training sets. This weight vector can be found through the following minimization problem

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \left(\sum_{f=1}^{F} \left(y_f - \sum_{n=1}^{N} \beta_n X_{n,f} \right)^2 + \lambda \sum_{n=1}^{N} |\beta_n| \right) \tag{2.5}$$

where the first part of the equation represents the minimization of the difference between $\boldsymbol{y}$ and the new version of $\boldsymbol{y}$. The right-hand side of the equation represents the ℓ_1 regularization and contains the variable λ which is a penalty term that regulates the strength of the ℓ_1 penalty and the sparsity of the model.

When the first part of the equation is large, meaning that $\beta_n X_n$ is not a good representation of y, λ will be large which in turn will shrink β_n to zero, introducing sparsity. When the opposite is obtained, the penalty will be low and β_n will be different from zero. When $\lambda = 0$, no parameters will be set to zero and the estimate will be the same as for a linear regression model.

As λ increases, the bias increases and the variance decreases. With low bias, the model can very well describe the examples in the training set. The variance is the sum of the squared difference between a predicted value and the actual value. If the model has low variance it can in a good way predict a test example given the training set [17]. This would mean that the ideal model, has both low bias and low variance. However, if the model has low bias it can easily fail to predict new data, meaning that the model is overfitting the training set i.e. the model has high variance.

This means that these two parameters in a way contradict each other, it is often not possible to have very low bias and at the same time a very low variance. The optimal model is therefore defined as the perfect balance between these two. To be able to find this combination, λ can be determined through cross-validation explained in Section 2.2.4, where the selected λ-value is the λ with the best cross-validation-score measured in the proportion of the variance (R^2). When implementing cross-validation a regression algorithm called LARS (Least-angle regression) can be used, which finds the feature that is most correlated to the target in each step and identifies the most relevant λ value [14, Section 1.1.3.1.1].

2.3.2 Ridge Regression

Unlike LASSO, ridge regression uses the ℓ_2 regularization technique, which in the minimization problem appear in the second term as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \left(\sum_{f=1}^{F} \left(y_f - \sum_{n=1}^{N} \beta_n X_{n,f} \right)^2 + \lambda \sum_{n=1}^{N} |\beta_n|^2 \right) \tag{2.6}$$

adding the magnitude squared of the coefficients as penalty. This means that as λ increases the coefficients get smaller, but will never be zero, since it gives higher penalty to the larger weights, resulting in no sparsity. The weight vector can be used in the same way as described in Section 2.3.1 and λ can be found with cross-validation and singular value decomposition (SVD) [12, Section Linear Models].

3

Methods

This chapter goes through the steps of choosing and using a room simulation software, which then is used to create the room impulse response data. Also, this chapter explains how the two regularized regression models were setup and tested.

3.1 Data Creation

3.1.1 Room Simulation Software

There are different room simulation software to choose from, the ones that were evaluated in this thesis work are MCRoomSim [2], pyroomacoustics [13], and gpuRIR [3]. The reason to not only work with one simulation tool, is to decrease the risk of working with a biased data set. With more variation, the more robust the end product will become.

MCRoomSim is a room simulation program that can be used with MATLAB. It allows for multiple sources and receivers and can simulate both diffuse and specular reflections. The software uses ISM (explained in Section 2.1.2) to simulate the specular reflections and a diffuse rain algorithm, which is a rapid stochastic ray tracing method, to simulate the diffuse part. A room created with MCRoomSim can be seen in Figure 3.1 with dimensions $(x, y, z) = [10, 5, 3]$ m, with a source placed in position $[5, 2, 1]$ m (●) and receiver placed in $[7, 3, 2]$ m (●).

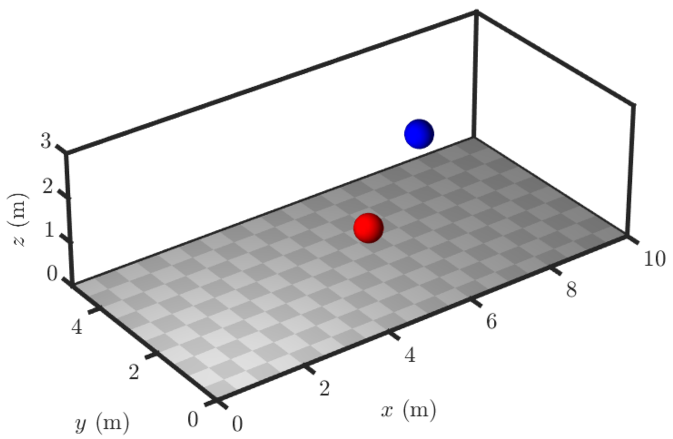


Figure 3.1: Visualization of a room created in the room simulation tool MCRoomSim, with dimensions $(x, y, z) = [10, 5, 3]$ m. A source is placed in position $[5, 2, 1]$ m (red sphere) and receiver in $[7, 3, 2]$ m (blue sphere).

The simulation tool pyroomacoustics has a Python object-oriented interface and allows for both specular and diffuse reflection simulations. The reflections are simulated with the ISM method, however there is no specified method for the diffuse part of the sound field. An impulse response created in both MCRoomSim and Pyroomacoustics, with the same room setup can be seen in Figure 3.2. The room impulse response obtained from pyroomacoustics drops 60 dB much quicker compared to MCRoomSim and seems to lack the same diffuse properties that MCRoomSim can simulate. This raised the idea of creating a room impulse response with only specular reflections in pyroomacoustics and adding that to an impulse response created in MCRoomSim with only diffuse reflection. This however, did not seem to work due to an uncertain delay in the impulse response.

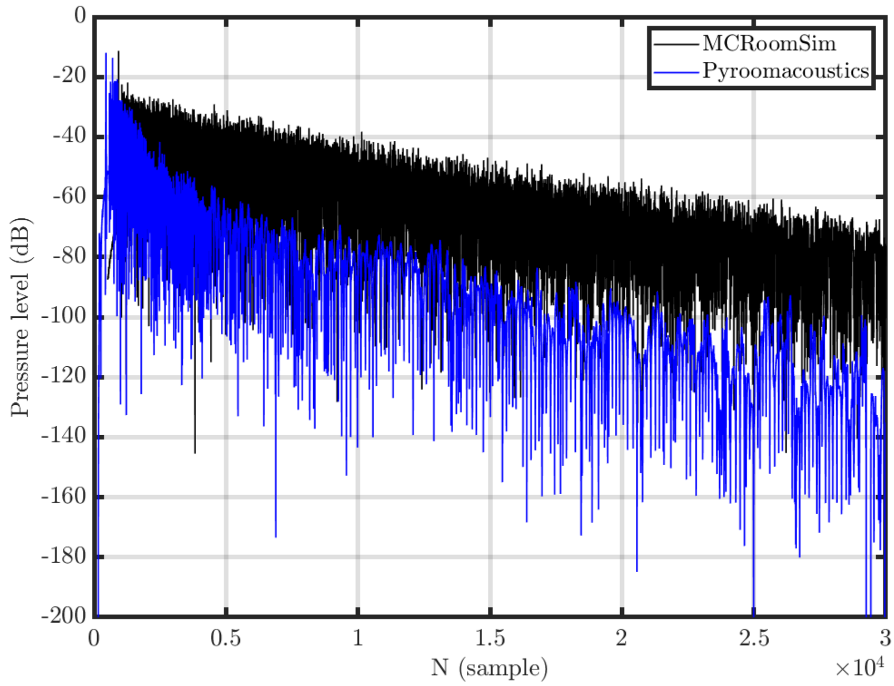


Figure 3.2: An impulse response created in Pyroomacoustics and Mcroomsim with the same settings.

If looking closer at the first peak (direct sound) in each of the impulse responses, one finds that the first peak created by pyroomacoustics is found at sample 451 while for MCRoomSim it is found at 925. By knowing the distance between source and receiver and the sampling frequency ($F_s = 44100$ Hz), it is possible to calculate where one could expect to find the first peak. Firstly the distance, d, between the source and receiver was calculated with the following equation

$$d = \sqrt{(x_{\text{source}} - x_{\text{receiver}})^2 + (y_{\text{source}} - y_{\text{receiver}})^2 + (z_{\text{source}} - z_{\text{receiver}})^2} \tag{3.1}$$

the delay in time was then calculated and when multiplying the time, T, with the sampling frequency, F_{s}, the delay in samples can be obtained

$$\begin{aligned} T &= \frac{d}{c_{\text{air}}} \\ N_{\text{peak}} &= T \cdot F_{\text{s}} \end{aligned} \tag{3.2}$$

which was approximately 411 for this particular example, but it can vary due to rounding errors. When using pyroomacoustics, it introduces a delay by default, called the global delay, of 40 samples meaning that the delay is actually $451 - 40 = 411$, which matches the delay that was calculated. For MCRoomsim the delay is $925 - 411 = 514$, which was consistent for all simulations.

Even though the method of using MCRoomSim and pyroomacoustics together now seemed to be possible, it was uncertain how to obtain entirely specular reflections from pyroomacoustics. When setting all scattering coefficients to zero, the result ended up not as expected. Because of uncertainties around default scattering values and running out of time, it was decided to not use pyroomacoustics.

The software gpuRIR, as the other two simulation software, uses ISM to calculate the impulse response of a room. It is different from the others since it uses GPUs (Graphics Processing Units), which increases the speed of the calculations [3]. Even though this software seemed to be a good candidate to the data creation step, there was no access to computers with GPUs and because of that, gpuRIR was ruled out. After evaluating the three software, the only candidate left was MCRoomSim.

3.1.2 MCRoomSim Input Variables

The simulation tool was used to create room impulse responses of several rooms, by stepping through a set of room dimensions created from a given minimum and maximum room dimension with a certain step size. The minimum and maximum room dimensions were set to $x_{\min} = 6$ m and $x_{\max} = 7$ m and the step size to 0.5 m. The depth and height of the room stayed the same for all simulations, $y = 5$ m and $z = 3$ m. The unique number of room dimensions is 3. This number of room dimensions was chosen because of the time limit, since the simulation program takes around 2 to 5 minutes per simulation there was not enough time for a larger room dimension set. Instead the focus was on just a few room dimensions and more variations of source positions, receiver positions, absorption and scattering coefficients.

In order to obtain features that does not follow a certain pattern, to later prevent overfitting (explained in Section 2.2.3), the rest of the input data was selected in a somewhat random manner.

When the room dimensions were set up, absorption and scattering coefficients (explained in Section 2.1.1) were generated. The scattering coefficients were obtained with a normal distributed random function in Matlab, following an underlying pattern, where the scattering coefficient overall increases with frequency. The amount of scattering coefficient sets was set to 5. Absorption coefficients were selected with a constraint, a maximum and minimum allowed reverberation time. The lowest acceptable reverberation time was set to 0.3 s and the maximum to 1 s. These coefficients were then generated in the same way as the scattering coefficients, resulting in about 3 to 10 sets of absorption coefficient per room dimension set.

The number of source positions was determined by the size of the room. For each simulation a source position was determined in a random manner, but within the constraint of not being closer than 1 m to each surface. For each source position a set of receivers were placed in the room. By randomizing 100 positions within the room, the ones that satisfied the requirement of being at least 1 m from all surfaces and the source, were saved and used in the simulation. This setup resulted in a total of around 50000 simulations.

3.1.3 Other settings

MCRoomSim also allows to implement other settings, such as air absorption, speed of sound, order of specular reflections, number of rays to trace from the source and the directivity of source and receiver.

The air absorption setting was set to "True", meaning that the simulation takes air absorption into consideration when calculating the room impulse response. The speed of sound is calculated in the simulation software as

$$c_{\text{air}} = 20.05 \cdot \sqrt{T + 273} \tag{3.3}$$

where the temperature, T, was set to 20°C, resulting in $c_{\text{air}} \approx 343.2$ m/s.

Given the size of the room, absorption and scattering coefficients, MCRoomSim can estimate the appropriate order of specular reflections. The number of rays should also preferably vary depending on the size of the room, this number was set for each simulation, where the number of rays increases with room size. The directivity of both the source and receivers were set to be omnidirectional.

3.1.4 Room Impulse Response

After each simulation, the calculated room impulse response was truncated to the length where the level of the squared impulse response was decreased by 60 dB compared to the direct sound (first peak). The room impulse response data is represented in pressure with its' original sign.

3.2 Estimating the Room Impulse Response

3.2.1 Structure data

The room features were ordered in such a way that each row of the input data corresponds to a room with 93 features, where each room has a corresponding room impulse response. Then the data was, in a random manner, split into training and test sets (explained in Section 2.2.1), where 10 % of the data was considered as the test set.

The room feature data was scaled according to Equation 2.4. Different minimum and maximum references were chosen, to be able to evaluate different scaling approaches. The first approach was to scale each feature with the minimum and maximum reference of itself (scaling method 1), i.e. each feature was scaled separately. The second approach was to scale the room dimensions, source position and receiver position with the maximum and minimum value within that data set (scaling method 2).

Likewise, all the absorption coefficients were scaled on their own and all the scattering coefficients on their own.

For a better prediction, any component that can be calculated manually should be eliminated from the data set. In this particular case, the index of the first peak (direct sound) of the impulse response can be found by knowing the source and receiver position with Equations 3.1 and 3.2. Therefore, the impulse response was shifted in a way such that the first peak is found at index 0. The first peak was kept in order to later be able to shift the estimated impulse response to the right index, however, even the value of the first peak can be calculated according to Equation 2.2 (using only the direct sound part). Because of this, the first peak along with the following 2 ms, was replaced with the true values.

3.2.2 LASSO model setup

In order to tune the parameter λ in Equation 2.5, cross-validation explained in Section 2.2.4 was implemented. The cross-validation was applied to the training set of each test setup, this way the λ with the best cross-validation score (R^2) was chosen and used in Equation 2.5 to generate the sparse vector, which contains one weight value per training example. By assuming that the linear relationship found between the room features of the training set and the test set, as

$$X_{\text{test}} = \sum_{n=1}^{N} \beta_n \cdot X_{\text{train},n} \tag{3.4}$$

also can be found between the impulse responses, the estimated impulse response was calculated as the superposition of the room impulse responses belonging to the training set multiplied with the sparse vector as

$$\hat{h} = \sum_{n=1}^{N} \beta_n \cdot h_n \tag{3.5}$$

where h_n is a vector containing the impulse response of one training example.

The algorithm used for these calculations was a `sklearn` class called `linear_-model.LassoLarsCV` [14, Section 1.1.3.1.1], which has a built in cross-validation algorithm, and LARS stands for least angle regression (explained in Section 2.2.4).

3.2.3 Ridge Regression model setup

The implementation of the ridge regression model was similar to the process of using LASSO. Now however, λ, was tuned to Equation 2.6. The weight vector $\boldsymbol{\beta}$ was generated, again, such that Equation 3.4 was fulfilled before using it to estimate the impulse response. The class `linear_model.RidigeCV` in `sklearn` [14, Section 1.1.2.4], which has a built in cross-validaiton algorithm, was used to find λ and to obtain the weight vector.

3.2.4 Model Validation

3.2.4.1 Schröder Curves

To evaluate the models, the Schröder's backward integration, explained in Section 2.1.3, was applied to the estimated and true impulse response and the root mean squared error (RMSE) was calculated between the estimated and true envelope in dB. In order to apply the backward integration, the trailing zeros of the impulse response were eliminated as well as the first peak along with the following 2 ms. The integral in Equation 2.3 was then calculated and flipped resulting in an envelope corresponding to the impulse response. The envelopes were then normalized to the maximum value of the true envelope. This process was performed for all the test examples and the models were evaluated by the average RMSE. To be able to compare the RMSE values, the worst and the best estimators were held as a reference. The best model estimation, is the estimation with the lowest RMSE value obtained over all estimations, and the worst is the highest RMSE over all estimations. As another reference, the average RMSE values between the Schröder curves in the test set created by the simulation software were calculated with the first test example as a reference. This will give a reference of how good the models are compared to an average guess based on the test examples. With Equations 3.1 and 3.2 the index of the first peak was calculated and the impulse response along with the Schröder curve was shifted to the correct index.

The RMSE value was calculated by subtracting the evelope of the true and estimated Schröder curve. However, if the estimated curve is either shorter or longer compared to the true, the obtained RMSE value is very high since one of the curves will have very high negative values while the other much lower negative values. Where the sound pressure is 0, the sound pressure level will, in reality, be minus infinity. To avoid this a small number, 10^{-300}, is added to the sound pressure, meaning that whenever the pressure is 0, the pressure level is -6000 dB. The RMSE value between the two whole Schröder curve will thus give an indication of how good the length of the impulse response is estimated. To better represent the length error however, the average length difference between the estimated and true impulse responses were calculated, so instead of comparing the whole Schröder curve RMSE, the average length difference was evaluated. To be able to compare how good the estimation is to the point where the true and estimated impulse response are different from 0 in sound pressure, the Schröder curves were set to the exact same length before the RMSE value was calculated, this is later referred to as the truncated RMSE value.

3.2.4.2 Tests

The models were tested with different room setups. The first setup that the models were tested on was a room with consistent room dimensions, absorption and scattering coefficients, but varying the source and receiver positions (Test 1). This test allowed for the evaluation of how good the models are at estimating the impulse response within the same room. The second room setup was the same as the first setup but now also varying the absorption and scattering coefficients (Test 2). Finally, the models were tested with varying all the room properties (Test 3).

3.2.4.3 Test Size Validation

With each model, a percentage of the data set was used as a test set as been mentioned before. To ensure that the test set is sufficiently large and that the results do not vary too much, the two models were used in a way that all examples in the data set (from Test 1) was once used as a test example. When one test example is chosen, the rest acts as the training set. The reason why this setup was not applied to all the tests was that with larger data sets the longer time it takes for the model to produce an estimation and because of time limitation, this was not an option. This means that the results from this test are based on the data setup from Test 1, and the outcome is applied to Test 1, 2 and 3.

4

Results and Discussion

In this chapter, the results of the various tests performed on the models are presented. The RMSE values and length differences are evaluated and presented in a table after each section to easier compare the different models. For each model, a random estimation example will be shown as a visualization of the estimations, this example can be compared with the true example shown in the subsection "True Impulse Response" under each Test section.

4.1 Test 1

During the first test, the rooms' dimensions were $(x, y, z) = (5, 6, 3)$ m and the simulations were carried out for 21 source positions and around 60 receiver positions per source position resulting in a total of 1274 data sets. In the following subsections, the reference or true impulse response and Schröder curve along with the results from the ridge regression and LASSO model will be shown for both scaling methods.

4.1.1 True Impulse Response

The room impulse response of test example 10 created by the simulation software is shown in Figure 4.1. In the following room impulse response estimations, this room impulse response is referred to as the true impulse response and is what the models are requested to recreate.

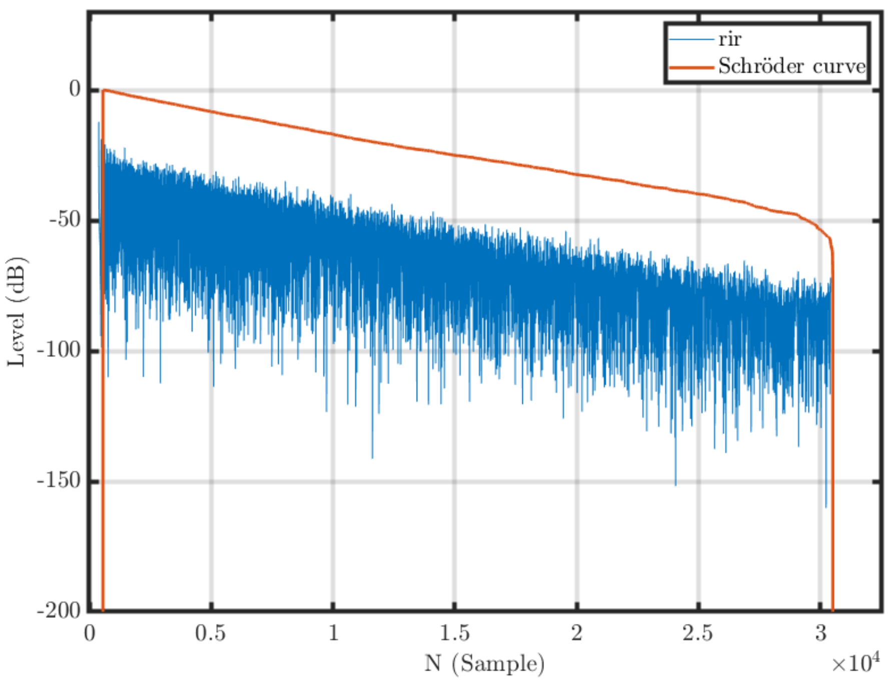


Figure 4.1: Room impulse response together with its' Schröder curve of test example 10, computed by the simulation software MCRoomSim. In comparison to the following figures, this is referred to as the true impulse response.

4.1.2 Ridge Regression, Scaling Method 1

Figure 4.2, shows the estimated impulse response of test example 10 using ridge regression and scaling method 1. The average RMSE of the truncated Schröder curve is 4.1279 dB, the average length difference is 2238 samples or 51 ms and around 17% of the estimations are shorter than the true. As can be seen in the figure, the fact that the estimated impulse response is a superposition of all training data is quite visible. Since the squared pressure level of the impulse responses fluctuates, the estimated looks like it represents the average level. This result could be an effect of having a too small training set and will be discussed later on.

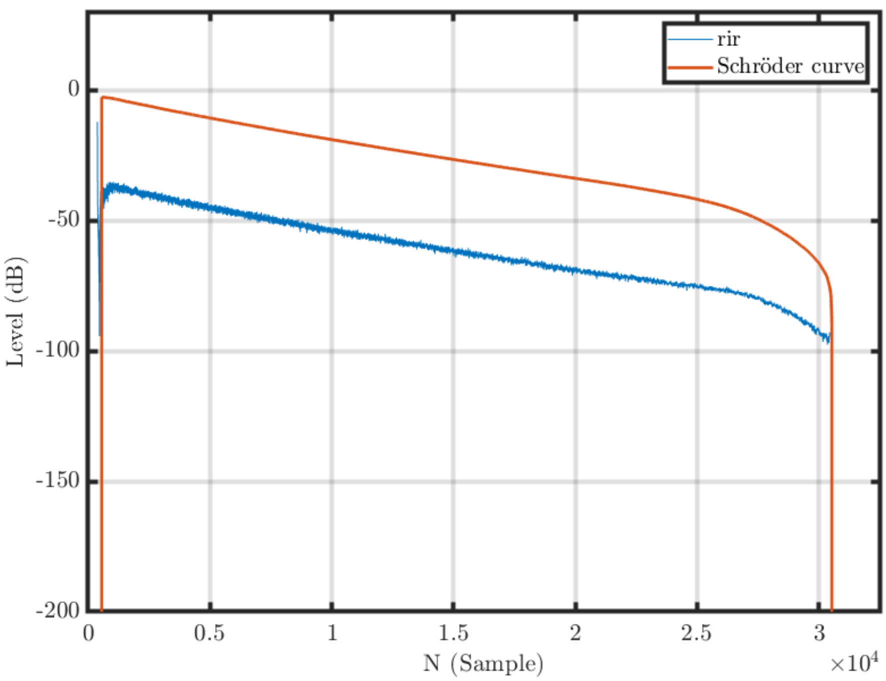


Figure 4.2: Estimated room impulse response of test example 10 using ridge regression and scaling method 1. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 3.8749 dB.

4.1.3 Ridge Regression, Scaling Method 2

Using ridge regression with scaling method 2 resulted in an average RMSE of 3.4531 dB. The estimated impulse response of test example 10 is shown in Figure 4.3 and results are similar to the first scaling method but slightly better. The average sample length error is at around 2010 samples or 46 ms, which is less than for scaling method 1, and around 15% of the estimations result in a too short sample length.

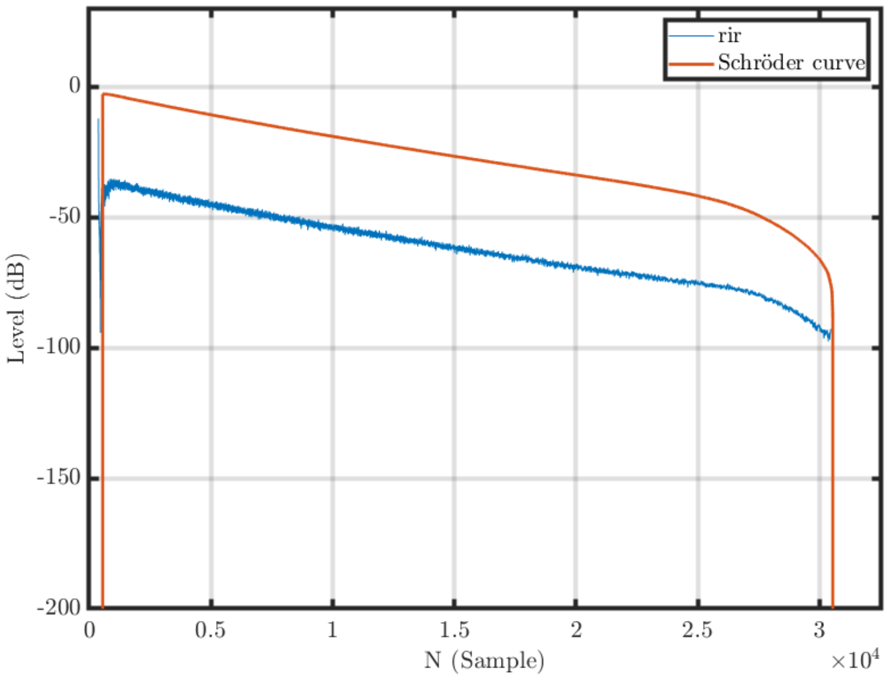


Figure 4.3: Estimated room impulse response of test example 10 using ridge regression and scaling method 2. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 3.8899 dB.

4.1.4 LASSO, Scaling Method 1

Moving on to the LASSO model, the first scaling method gave a result of 4.1607 dB in average RMSE. Figure 4.4, shows a plot of the estimated room impulse of test example 10, which shows that the estimated impulse response has larger fluctuations compared to the ridge regression models. The average sample length is around 2277 samples or 52 ms off, and 19% of the estimations are shorter than the true impulse response. The amount of sparsity in this model is around 96%, meaning that only around 50 training examples are used in the estimation.

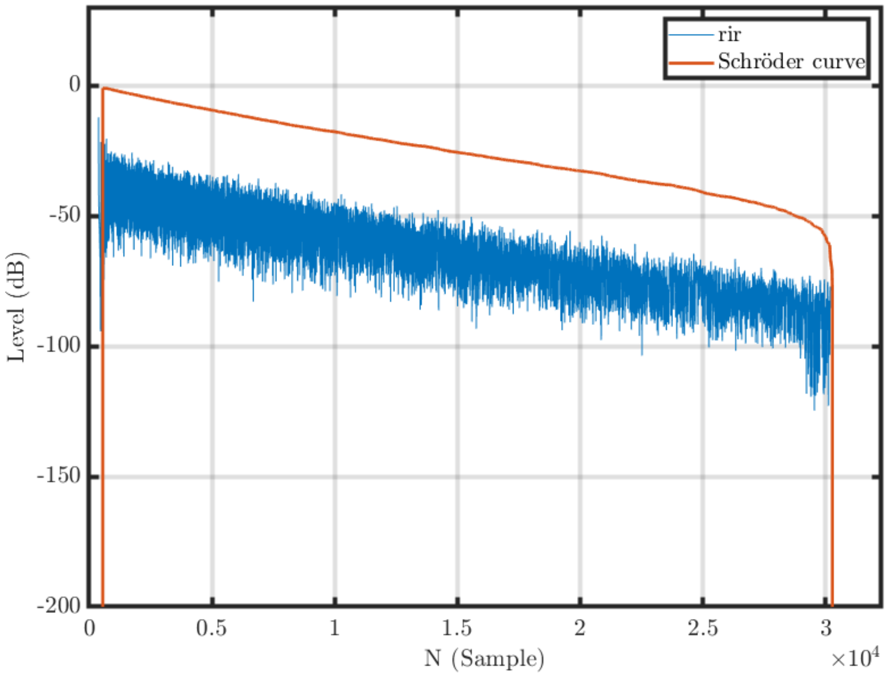


Figure 4.4: Estimated room impulse response of test example 10 using LASSO and scaling method 1. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 1.0010 dB.

4.1.5 LASSO, Scaling Method 2

Lastly, the second scaling method was applied to the LASSO model. This method resulted in an average RMSE of 3.4175 dB for the truncated version, which is just a bit lower compared to the first scaling method with the LASSO model. The impulse response of test example 10 is shown in Figure 4.5 and shows that even more of the fluctuations are incorporated in the estimation compared to the first scaling method. The difference in length is on average 1949 samples or 44 ms, and 37% of the estimations are shorter than the reference. For this model the sparsity is around 98%, and only around around 20 training examples are used in the estimation.

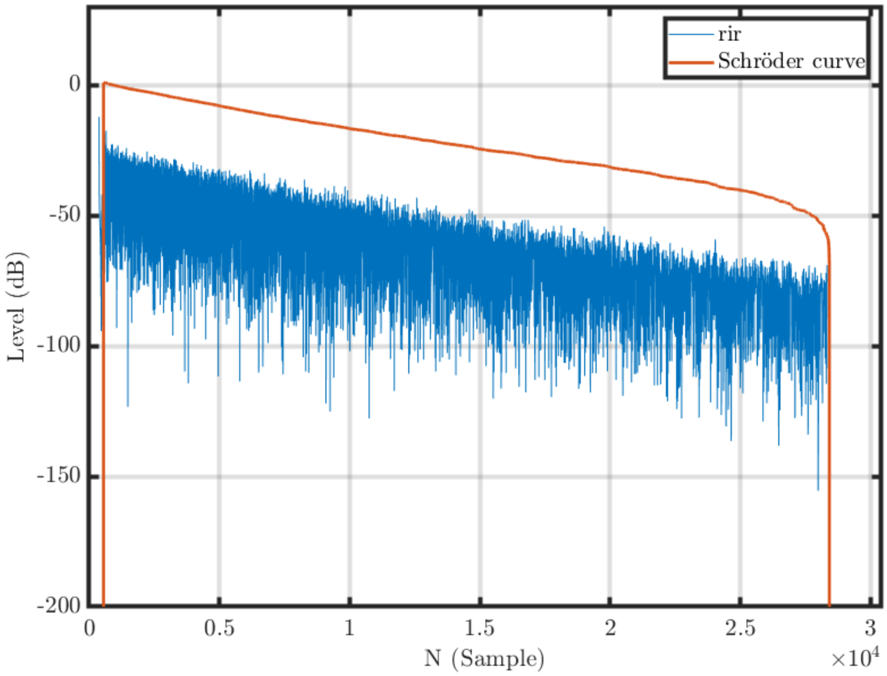


Figure 4.5: Estimated room impulse response of test example 10 using LASSO and scaling method 2. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 0.9526 dB.

4.1.6 Comprehension of the RMSE Values

In Table 4.1, the mean RMSE values for each model are listed together with the lowest and highest RMSE value as explained in Section 3.2.4. Overall, the second scaling method seems to work better for both models. The best estimation is obtained with the LASSO model together with scaling method 2, but with very small marginals. However, the major difference between LASSO and ridge regression seems to be that the LASSO model can better estimate the fluctuations in the impulse response. It seems that the ridge regression model needs more training data to base the estimate on. When looking at the average length difference, it can be seen that also the LASSO model together with scaling method 2 estimates the length of the impulse response most accurately.

As explained in Section 3.2.4.1, the average RMSE of test example 1 and the rest of the test examples created by the simulation software were calculated in order to see how different each test example is to one another. This resulted in an RMSE of 1.3631 dB for the truncated which in this context shows that the results obtained in Test 1 are quite poor.

Table 4.1: Comparing the average RMSE values within Test 1 for each of the models with different scaling methods, scaling method 1 is marked with s1 and scaling method 2 with s2. The best model value is the lowest obtained RMSE value/lowest length difference over all estimations, and the worst is the highest obtained RMSE value/lowest length difference over all estimations.

Model	Lowest	RR s1	RR s2	LASSO s1	LASSO s2	Highest
Truncated Schröder curve RMSE (dB)	0.3557	4.1279	3.4531	4.1607	3.4175	39.0522
Average Length Difference (samples)	2	2238	2010	2277	1949	8403

4.2 Test 2

During the second test, the rooms' dimensions was still $(x, y, z) = (5, 6, 3)$ m, but now also the absorption and scattering coefficients are varying. The simulations were carried out for 15401 data examples with 10% test size. In the following subsections, the reference or true impulse response and Schröder curve along with the results from the ridge regression and LASSO model will be shown for both scaling methods.

4.2.1 True Impulse Response

The room impulse response of test example 120 is shown in Figure 4.6. In the following visualizations of the estimations, this room impulse response is the true impulse response and is what the models are requested to recreate.

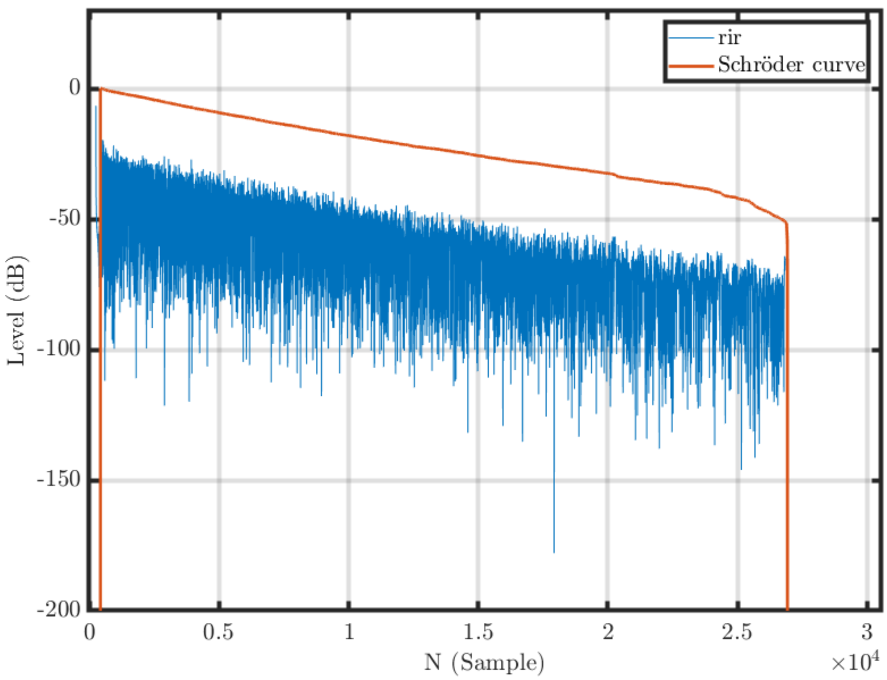


Figure 4.6: Room impulse response of test example 120 together with its' Schröder curve of a room, computed by the simulation software MCRoomSim. In comparison to the following figures, this is referred to as the true impulse response.

4.2.2 Ridge Regression, Scaling Method 1

In Figure 4.7 is the result of an estimation (test example 120) done with ridge regression and scaling method 1. The results in this test are very different from the first test since now the fluctuations are incorporated in the estimation, but the Schröder curve is around 25 dB lower than the true. The average RMSE over all test examples is 27.0605 dB, the average length error is 6120 samples or 139 ms and 23% of the estimations result in a too short sample length.

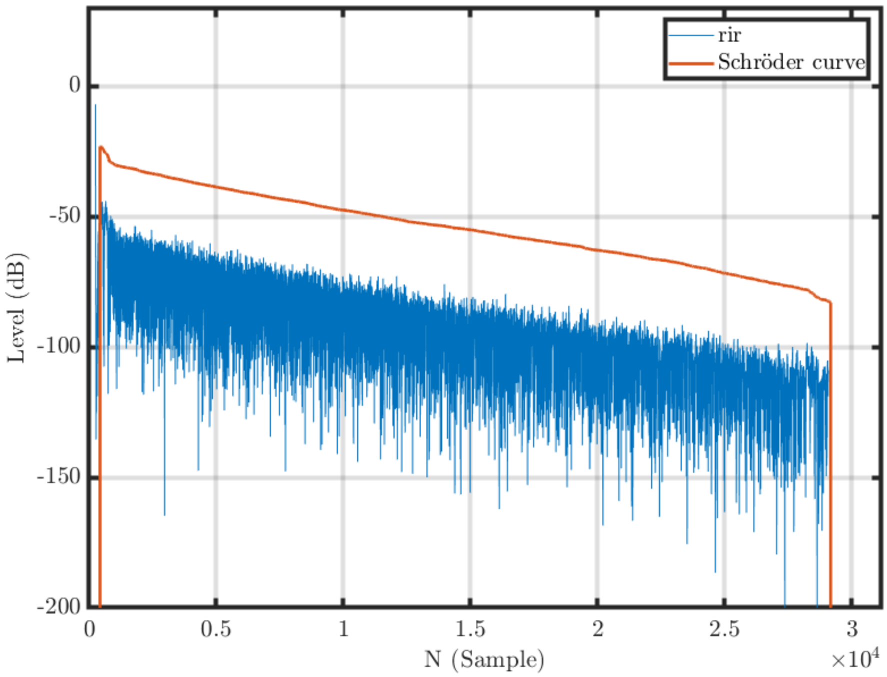


Figure 4.7: Estimated room impulse response of test example 120 using ridge regression and scaling method 1. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 30.0055 dB.

4.2.3 Ridge Regression, Scaling Method 2

Similar results to scaling method 1 were obtained when using ridge regression together with the second scaling method, and the result for test example 120 can be seen in Figure 4.8. The average RMSE 27.2972 dB, the average error in sample length is 6031 samples or 137 ms and 23% of the estimations are shorter than the true.

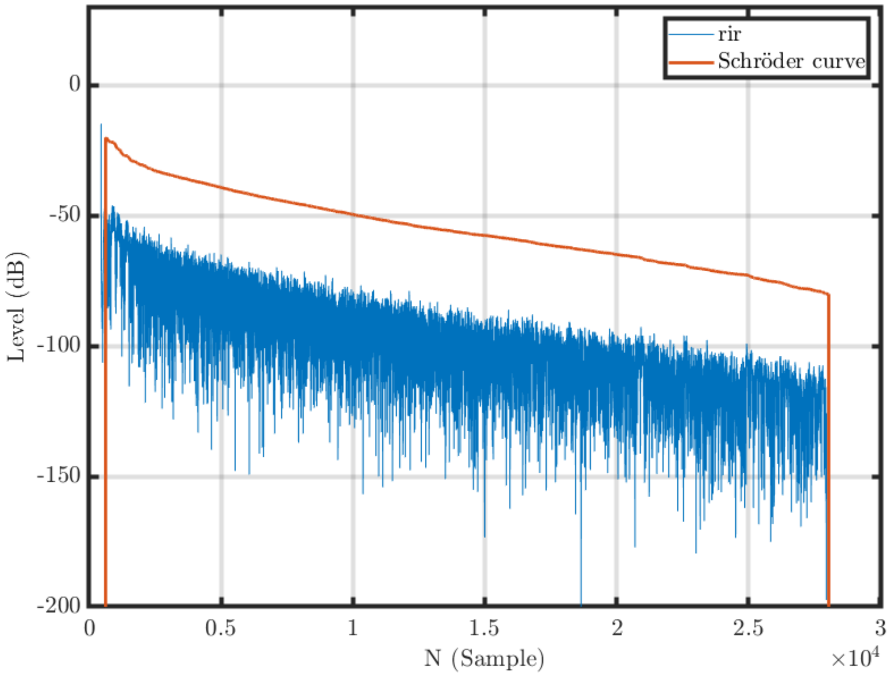


Figure 4.8: Estimated room impulse response of test example 120 using ridge regression and scaling method 2. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 29.9595 dB.

4.2.4 LASSO, Scaling Method 1

When using the LASSO model together with the first scaling method, the average RMSE turned out to be 5.6072 dB for the truncated version, which is around 2 dB higher compared to Test 1. The sample length error is 1612 samples or 37 ms and 26% of the estimation are too short. This model does a much better estimation compared to the ridge regression models. The sparsity for this model is around 99%, meaning that the estimation is based on around 148 training examples.

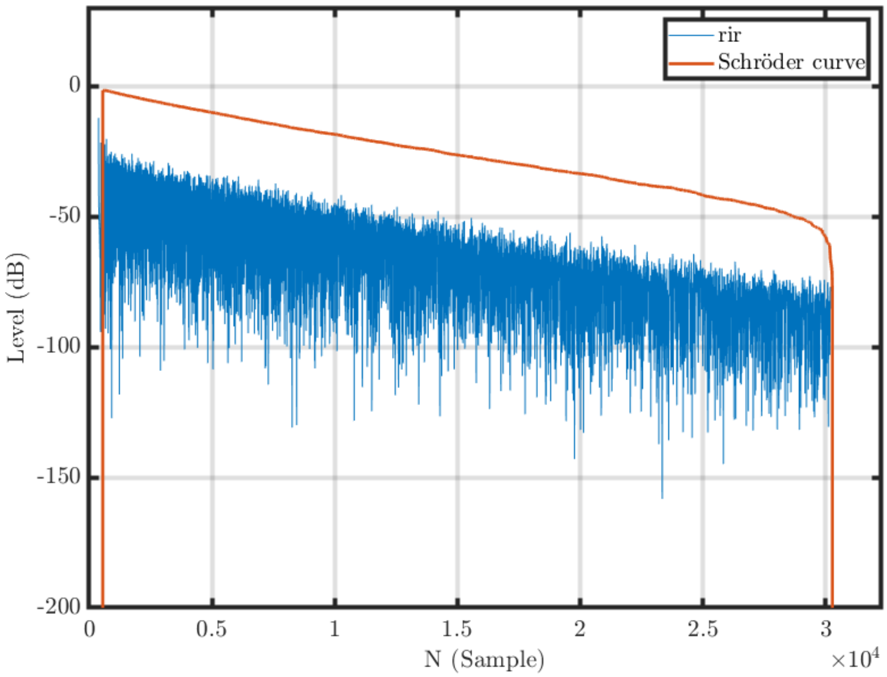


Figure 4.9: Estimated room impulse response of test example 120 using LASSO and scaling method 1. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 1.9166 dB.

4.2.5 LASSO, Scaling Method 2

For the second scaling method, using the LASSO approach, the obtained average RMSE is 7.4138 dB for the truncated Schröder curve which is surprisingly higher than the first scaling method. The estimation of test example 120 is shown in Figure 4.10 and is about 2 dB lower compared to the true value. Since for Test 1, the second scaling method showed a better result compared to the first for the LASSO model, it is surprising that Test 2 shows the opposite. The sample length error for this test is 5843 samples or 132 ms and the length is 37% of the time too short. The sparsity for this model is around 99%, meaning that the estimation is based on around 148 training examples.

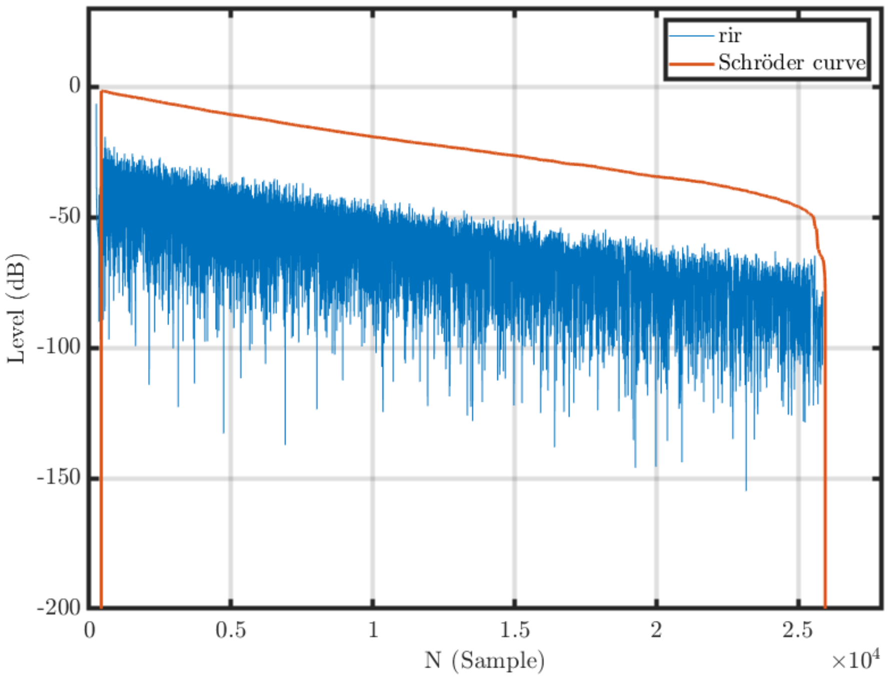


Figure 4.10: Estimated room impulse response of test example 120 using LASSO and scaling method 2. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 1.7755 dB.

4.2.6 Comprehension of the RMSE Values

All the Schröder curve RMSE results from Test 2 are gathered in Table 4.2 and shows that the LASSO model together with scaling method 1 gives the best result. For this test, the first scaling method gave better results for most of the RMSE values, which as mentioned before is different from the first test where the second scaling method gave better results. Overall, the LASSO model outperforms the ridge regression model but the RMSE values are a bit higher compared to Test 1. The reason for this could be that there are too many variation in the training set and since the estimation is a superposition of the training set, this could mean that impulse response to predict is not similar enough to the training set leading to a poor estimation. When looking at the average length difference, it can be seen that the LASSO model together with scaling method 1 is also estimating the length most accurately.

The average RMSE between true Schröder curves of test example 1 and the rest of the test examples is 7.5064 dB. This again shows that the estimations are quite poorly estimated for the ridge regression models. Both the LASSO models however, have lower RMSE values than than this reference, which shows that the relationship between the room features is somewhat correlated to the relationship between the impulse responses.

Table 4.2: Comparing the average RMSE values within Test 2 for each of the models with different scaling methods, scaling method 1 is marked with s1 and scaling method 2 with s2. The best model value is the lowest obtained RMSE value/lowest length difference over all estimations, and the worst is the highest obtained RMSE value/lowest length difference over all estimations.

Model	**Lowest RMSE**	**RR s1**	**RR s2**	**LASSO s1**	**LASSO s2**	**Highest RMSE**
Truncated Schröder curve RMSE (dB)	0.2604	27.0605	27.2972	5.6072	7.4138	31.9945
Average Length Difference (samples)	2	6120	6031	1612	5843	17215

4.3 Test 3

During the third and final model test, all room features are varying and the number of simulations were 49594 with a test size of 10%. In the following subsections, the reference or true impulse response and Schröder curve along with the results from the ridge regression and LASSO model will be shown for both scaling methods.

4.3.1 True Impulse Response

For visualization, the impulse response and Schröder curve of test example 1000 is shown in Figure 4.11 and is referred to as the true.

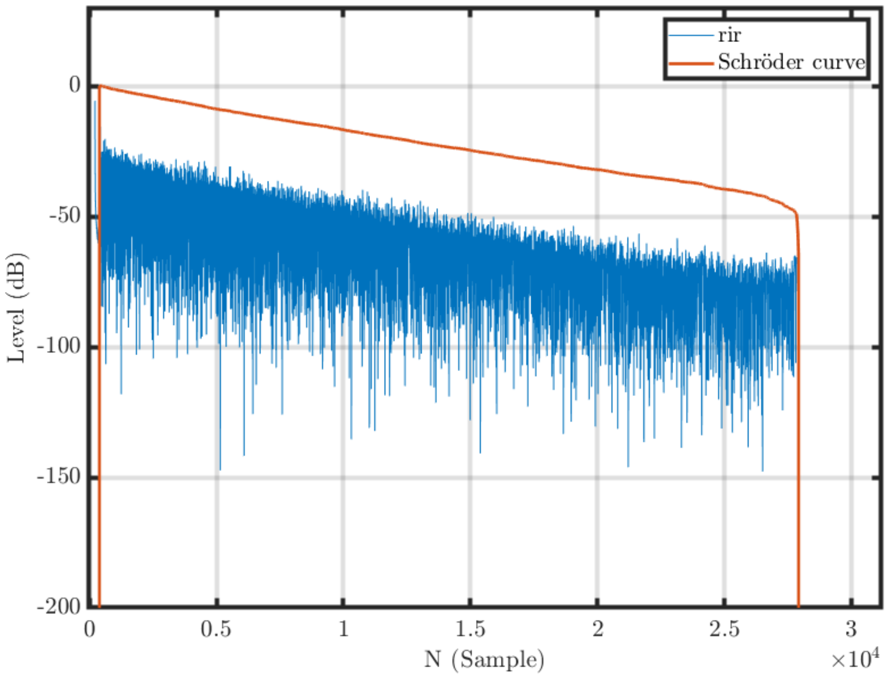


Figure 4.11: Room impulse response of test example 1000 together with its' Schröder curve of a room, computed by the simulation software MCRoomSim. In comparison to the following figures, this is referred to as the true impulse response.

4.3.2 Ridge Regression, Scaling Method 1

When applying the ridge regression model with scaling method 1 to the data set, the average RMSE resulted in 28.6582 dB for the truncated version. The estimation of test example 1000 can be seen in Figure 4.12 and as seen before, the Schröder curve is around 25 dB lower than the true. The length on average is off by about 6135 samples or 139 ms where about 20% of the estimations are too short.

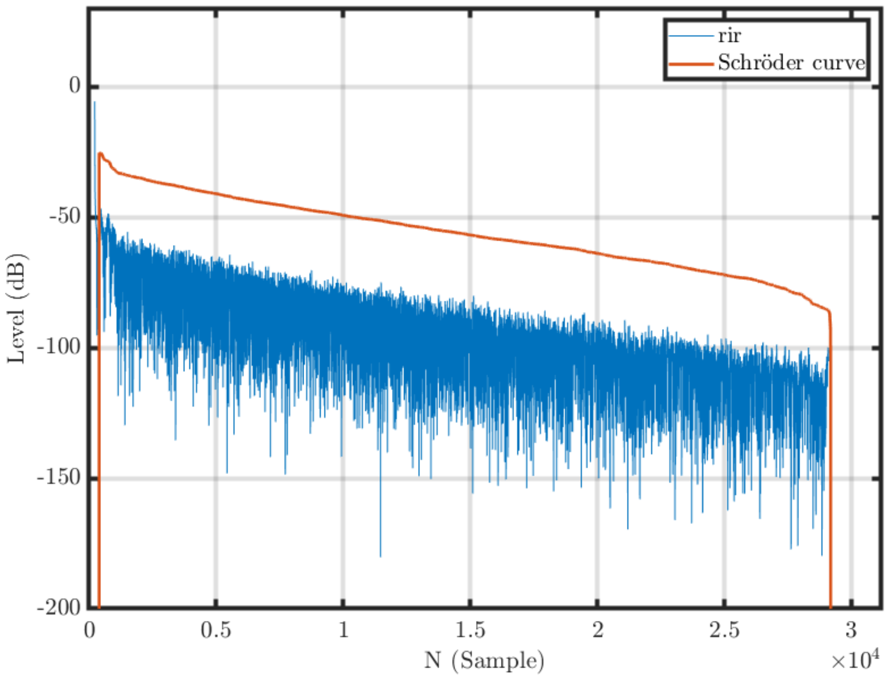


Figure 4.12: Estimated room impulse response of test example 1000 using ridge regression and scaling method 1. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 27.4692 dB.

4.3.3 Ridge Regression, Scaling Method 2

For the second scaling method when using the ridge regression model, the average RMSE is 28.6320 dB for the truncated version. The average length difference is 6102 samples or 138 ms and 21% of the estimations are estimated too short compared to the true. The impulse response of test example 1000 is shown in Figure 4.13 and it appears to be very similar to the one created with the first scaling method. Comparing the RMSE values with the first scaling method, the second scaling method is slightly better.

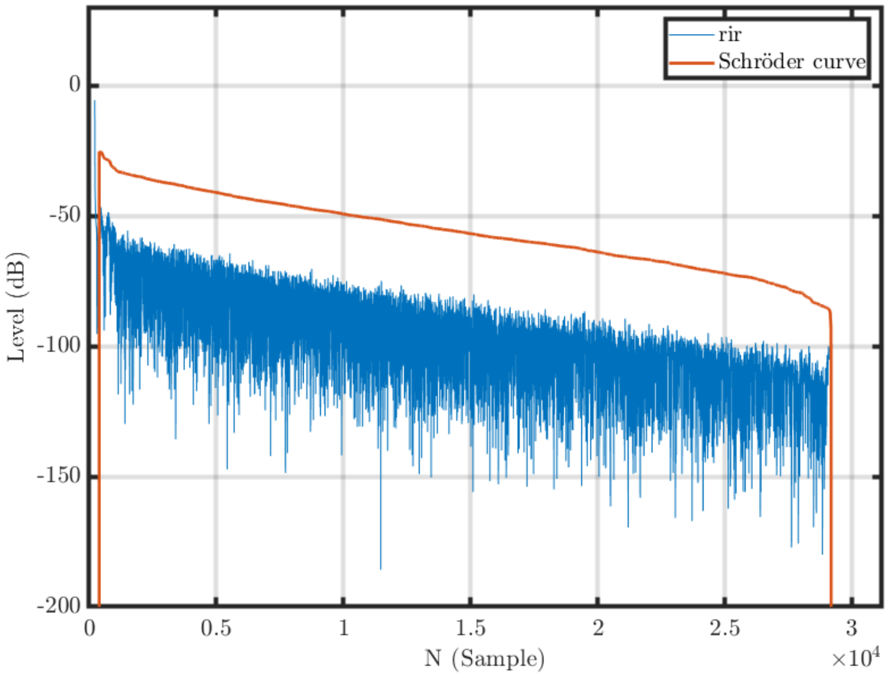


Figure 4.13: Estimated room impulse response of test example 1000 using ridge regression and scaling method 2. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 27.4208 dB.

4.3.4 LASSO, Scaling Method 1

In Figure 4.14 the estimation of test example 1000 done with the LASSO model and scaling method 1 is presented. The average RMSE for all estimations is 5.94783 dB for the truncated version and is the highest RMSE value so far with this setup. The average length difference is 1574 samples or 36 ms and 25% of the estimations are shorter than the true. Even for this test, the LASSO model outperforms the ridge regression model, it is consistently around 20 dB better. The weight vectors for this model has a sparsity of around 99%, which means that around 500 training examples are used in the estimations.

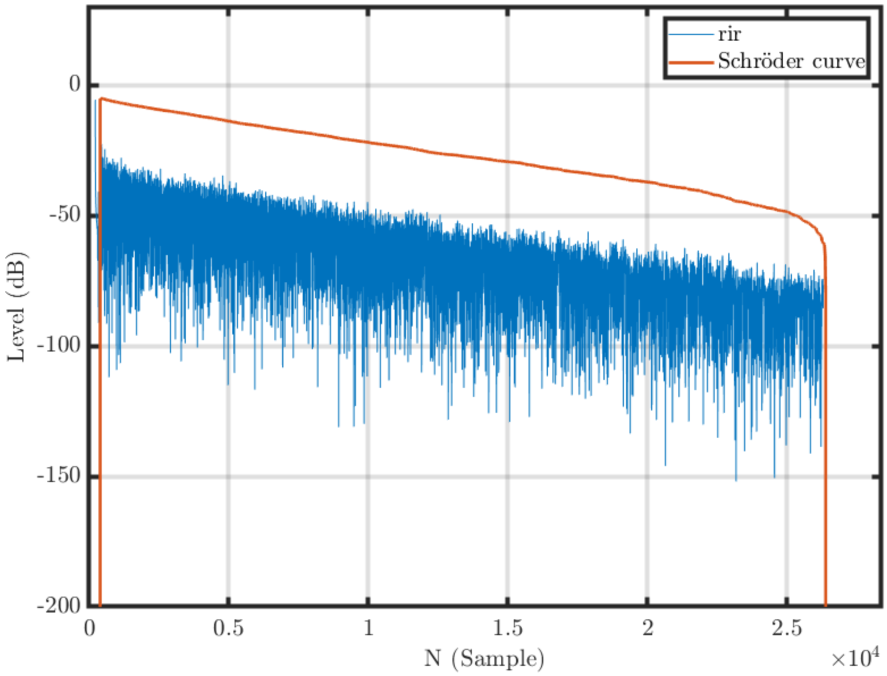


Figure 4.14: Estimated room impulse response of test example 1000 using the LASSO model and scaling method 1. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 5.1936 dB.

4.3.5 LASSO, Scaling Method 2

For the second scaling method the LASSO model estimations resulted in an RMSE of 6.0211 dB for the truncated version. The majority of the tests now show slightly better results when using the first scaling method. The average length difference is 1553 samples or 35 ms and 26% of the estimations are shorter than the true. In Figure 4.15 is the estimation of test example 1000, which has similar appearance as the true. The weight vectors for this model has a sparsity of around 99%, which means that around 500 training examples are used in the estimations.

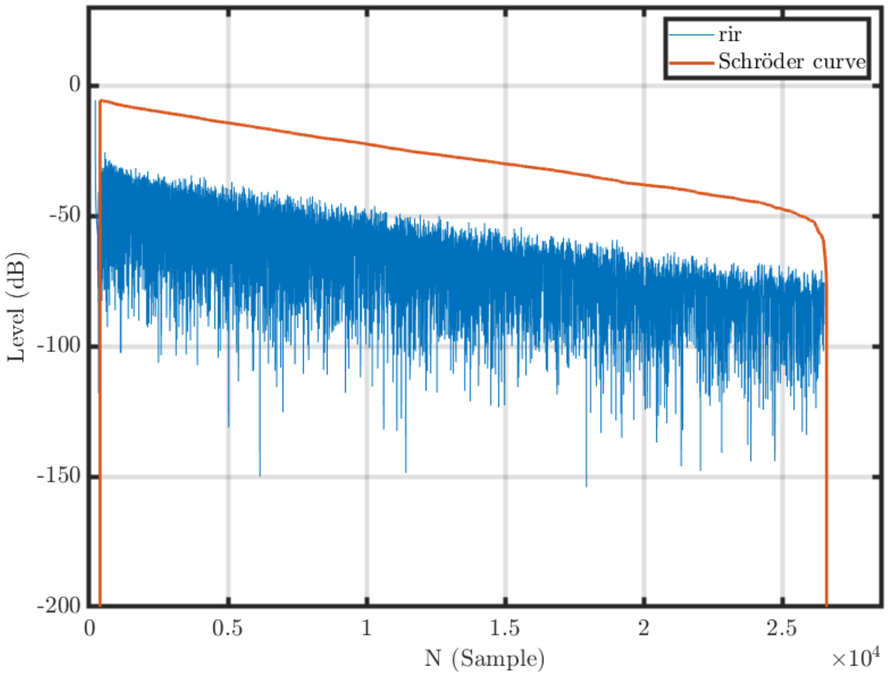


Figure 4.15: Estimated room impulse response of test example 1000 using the LASSO model and scaling method 2. The plot shows the estimated curve to the point where the pressure level has dropped 60 dB. This particular estimation has an RMSE of 5.4809 dB.

4.3.6 Comprehension of the RMSE Values

In Table 4.3 all the RMSE values and length difference from Test 3 are listed together with the lowest and the highest obtained RMSE and length difference values. Again, it is visible that the LASSO model outperforms the ridge regression model and as also seen in Test 2, the first scaling method give better results. Since the models in this test predicts room impulse responses of room also with different sizes, it is not surprising that the average RMSE values are a bit higher.

The average RMSE value calculated for all test examples with test example 1 as reference is 7.3736 dB. This shows that the LASSO models does a relatively good estimation.

Table 4.3: Comparing the average RMSE values within Test 3 for each of the models with different scaling methods, scaling method 1 is marked with s1 and scaling method 2 with s2. The best model value is the lowest obtained RMSE value/lowest length difference over all estimations, and the worst is the highest obtained RMSE value/lowest length difference over all estimations.

Model	Lowest RMSE	RR s1	RR s2	LASSO s1	LASSO s2	Highest RMSE
Truncated Schröder curve RMSE (dB)	0.4268	28.6582	28.6320	5.9478	6.0211	71.8733
Average Length Difference (samples)	0	6135	6102	1574	1553	20183

4.4 Test Size Validation

As described in Section 3.2.4, the ridge regression model was tested with the data set from Test 1, where each data example was used once as a test set, while the other acted as a training set. The results in average RMSE is presented in Table 4.4.

This test shows that Test 1 might have a too small training set, even though it resulted in lower RMSE values. However, the results from Test 1 seems to lack the property of a fluctuating impulse response which in this test it does not, but instead the Schröder curve start at around 20 dB lower compared to the true curve. This would mean that a test size of 10% is not enough.

Since the results ended up being relatively similar to Test 2 and 3 however, the test size of 10% was assumed to be a somewhat appropriate size and was used for the remaining tests. It seems that it does not depend on the percentage of the test size, but rather how large the training set is.

With more training data the RMSE values were expected to be lower, but that is not the case for this setup. The only reasonable explanation to this, is that in Test 1 the training set is simply to small for the model to be able produce a good estimate, especially for the ridge regression model.

Table 4.4: Comparing the average RMSE values obtained in Test 1 with a test size of 10% with a test setup where all data sets are once considered as a test set while the rest are considered training set. This test is called Test 4 in this table. TSC stands for truncated Schröder curve ans ALD average length difference.

Model	RR s1	RR s2	LASSO s1	LASSO s2
Test 1, ALD (sample)	2238	2010	2277	1949
Test 4, ALD (sample)	2267	2197	2719	2616
Test 1, TSC RMSE (dB)	4.1279	3.4531	4.1607	3.4175
Test 4, TSC RMSE (dB)	24.6400	26.1884	5.2014	4.4127

5
Conclusion

To summarize the results from this thesis work, it shows that the LASSO model outperforms the ridge regression model by about 20 dB when looking at the truncated Schröder curve. The estimations with the lowest RMSE values were obtained during Test 1, which is not too surprising since all room features in the data set were consistent except for the source and receiver positions. This means that the training and test sets are more similar to each other in comparison to the training and test sets for the other tests. However, the average RMSE between all true Schröder curves in the test sets has a lower value than the average RMSE of the estimates, which means that the estimates are relatively poor. This would mean that the relationship between the room features is not a good representation of the relationship between the impulse responses.

Even though the data set in Test 3 had about the same amount of source and receiver positions for each set of room dimensions as in Test 1, the room sizes were 0.5 m apart. Since the estimate is a superposition of all room impulse responses this means that even the smallest room, depending on the weight value, contributes to the estimation of the biggest room, even though they might not have very similar impulse responses. If the step size between each room instead was e.g. 0.1 m, a better estimation might have been obtained. Even though the LASSO model seems to estimate the impulse responses with low RMSE, its' RMSE value is still not very different from the average RMSE of the true curves from the test set. The relationship between the room features for both Test 2 and Test 3 is a better representation of the relationship between the impulse responses compared to Test 1, but the average RMSE values are still not majorly different from the average RMSE between the true curves in the test set.

Since the only difference between the LASSO model and the ridge regression model is that LASSO introduces sparsity, this would mean that sparsity has a major impact on the estimation. This in turn means that some training examples are more important to the estimation and perhaps even some features are more important than others. A possible way to achieve better estimations, would maybe be to weigh the features in order of importance. When scaling the features, the more important features can be multiplied with a factor representing its importance. For example, a change in one absorption coefficient affects the impulse response much less than changing one of the room dimension variables. By weighing the features, this potentially could help the models make a better prediction.

Even though I would think that this way of estimating an impulse response could work with some adjustments, it would require a lot more data if adjusting the model to handle all sorts of room sizes, including non shoe-box shaped rooms. As a last conclusion, this way of estimating an impulse response is a much more straightforward and simple way compared to the already existing methods but as it seems now, the relationship between the room features is not a very good representation of the relationship between the impulse responses. To improve these models, weighing the features in order of importance could perhaps make a huge difference in the estimations.

Bibliography

[1] Allan D. Pierce (2019) Third Edition "Acoustics An Introduction to Its Physical Principles" and Applications

[2] A. Wabnitz, N. Epain, C. Jin and A. van Schaik, in Proceedings of the International Symposium on Room Acoustics (ISRA) (2010) "Room acoustics simulation for multichannel microphone arrays"

[3] Diaz-Guerra, D., Miguel, A. and Beltran, J.R. (2020) "gpuRIR: A python library for room impulse response simulation with GPU acceleration"

[4] Emanuël A. P. Habets (2006) "Room Impulse Response Generator"

[5] Heinrich Kuttruff (2007) "Acoustics An introduction"

[6] ISO. (2003), "Acoustics - Measurement of sound absorption in a reverberation room"

[7] K. Jambrosic, M. Horvat and H. Domitrovic (2008), "Reverberation time measuring methods"

[8] Mendel Kleiner (2012) "Acoustics and Audio Technology, Third edition"

[9] Michael A. Babyak (2004) "What You See May Not Be What You Get: A Brief, Nontechnical Introduction to Overfitting in Regression-Type Models"

[10] M.R Schröder (1964) "New Method of Measuring Reverberation Time"

[11] Piotr Bilinski, Jens Ahrens, Mark R. P. Thomas, Ivan J. Tashev, John C. Platt (2014) "HRTF MAGNITUDE SYNTHESIS VIA SPARSE REPRESENTATION OF ANTHROPOMETRIC FEATURES"

[12] Rafael A. Irizarry and Michael I. Love (2021) Data Analysis for the Life Sciences

[13] Robin Scheibler, Eric Bezzam, Ivan Dokmanic (2017), "PYROOMACOUSTICS: A PYTHON PACKAGE FOR AUDIO ROOM SIMULATION AND ARRAY PROCESSING ALGORITHMS"

[14] scikit-learn developers (2007 - 2020) "Linear Models" `https://scikit-learn.org/stable/modules/linear_model`

[15] scikit-learn developers (2007 - 2020) "Cross-validation: evaluating estimator performance" `https://scikit-learn.org/stable/modules/cross_validation`

[16] S. Kukreja, J. Lofberg, and M. J. Brenner (2006) "A least absolute shrinkage and selection operator (LASSO) for nonlinear system identification"

[17] T. Hastie, R. Tibshirani, and J. Friedman, Springer, second edition (2009), "The Elements of Statistical Learning: Data Mining, Inference, and Prediction".

DIVISION OF APPLIED ACOUSTICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY