Automatic DOA estimator for spatial music recordings

Audio Engineering Project

Samuel Maurer, Simon Beck

Supervisors: Dipl.-Ing. Daniel Rudrich, Dipl.-Ing. Dr. Matthias Frank

Graz, February 7, 2021

Contents

1 Zusammenfassung

Die räumlichen Positionen der Klangobjekte und die Parameter eines Raumes spielen bei ambisonischen Musikaufnahmen eine wichtige Rolle. Im Zuge dieser Arbeit entsteht ein Audio Plug-in welches es ermöglichen soll die räumlichen Positionen der Klangobjekte verschiedener Instrumente zu schätzen, also auf welcher Position sie sich bezogen auf ein ambisonisches Mikrofonarray befinden. Für die Richtungsschätzung werden zwei verschiedene Herangehensweisen untersucht, wobei beide auf dem Grundsatz der Intensitätsvektorbildung beruhen. Bei dem ersten Ansatz wird ein Intensitätsvektor über das Array-Signal berechnet, welches mit einer Frequenzmaske gefiltert wird, um die einzelnen Instrumentensignale herauszufiltern. Der zweite Ansatz ist eine Intensitätsvektorberechnung einer ambisonischen Raumimpulsantwort. Die Raumimpulsantwort wird mithilfe eines LMS-Algorithmus aus den Musiksignalen geschätzt. Aus dieser Raumimpulsantwort werden in weiterer Folge die Raumparameter ermittelt. Da es eine schwierige Aufgabe ist eine stabile Echtzeitimplementierung dafür zu entwerfen, werden verschiedene Methoden nur untersucht, finden aber im Plug-in noch keine Anwendung.

2 Abstract

The position of sound objects in a room and its parameters are important knowledge when it comes to Ambisonic musical performances. In this project, an audio plug-in is developed which enables performers to estimate the position of their instruments in a room related to an Ambisonic microphone array. Two different approaches of this task will be examined. Both are based on the same idea of building an intensity vector for determining the direction of arrival of the sound sources. One approach is to calculate the intensity vector of the whole array signal in the frequency domain which is filtered with a mask to separate the different instrument signals. The second approach is to calculate the intensity vector of an Ambisonic room impulse response. The room impulse response is extracted from the musical signals with the help of an LMS-algorithm. With this room impulse response, it is also possible to find room parameters, such as reverberation time and the dimensions of the room. Since it is a difficult task to make a stable and realtime compatible algorithm for this, it will not find place in the plug-in but different methods to determine those parameters will be evaluated.

3 Introduction

In live musical performances or studio recordings, one job of the sound engineer is usually to do the panning of the instruments. This can be done for stereo signals, multichannel signals and of course Ambisonic signals. With the help of the plug-in which will be developed in the course of this project, it will be possible to estimate the directions of the instruments relative to an array microphone in real time and automatically place the spot microphone signals of those instruments in the right direction with the use of other plugins that already exist. The directions are updated in a given interval, so if a musician moves on the stage, the spot signals move with the musician.
To have enough data for testing and evaluation, a first recording is made in the CUBE (Chapter 6). During this recording, multiple first-order Ambisonic impulse responses as well as spot impulse responses are measured, so that afterwards, it is possible to use any kind of musical signals. Additionally some piano, guitar, and speech sounds were recorded live to get more realistic signals. Except for the piano, recordings of moving musicians and speakers were made as well. All positions of the microphones, loudspeakers, instruments and speakers including movements are tracked with the tracking system which is present in the CUBE to compare the estimated angles with the real ones. A second recording took place in the end of the project at Reiterkaserne (LS82EG12), which is primarily made for demonstration purposes, but is also used as a second set of measurements to test the algorithms.
Two algorithms were implemented in the plug-in and can be used to estimate the directions with the help of building an intensity vector. In the first approach, the array signal is transferred into the frequency domain where it then is filtered with as many masks as there are spot microphones present. With these filtered array signals, the direction of each spot microphone is estimated by calculating an intensity vector. Chapter 4 explains the basics of a least mean squares filter, which is used as a second approach, and furthermore a modification of the filter and its implementation. Chapter 7 presents the plug-in and its user interface.
The project includes a second part with the goal to estimate different room parameters, such as room dimensions and reverberation time from the impulse responses that are extracted from the music signals with the help of the LMS filter. This part of the project is discussed in Chapter 8.

4 Least mean squares algorithm (LMS)

For this project, the least mean square algorithm was chosen for the estimation of the impulse responses and directions of arrival, which were needed for further calculations of room parameters. There are several techniques to measure impulse responses of a room, e.g. recording the shot of an alarm gun/clapperboard, playback/record a sweep and post process the signals with a program such as Matlab. For these most common techniques it is necessary to bring some equipment to the recording scene (pistol, clapperboard, loudspeakers, etc.) which is not always easy to get or requires too much time to set up, when you just want a fast result. Since there are already instruments and musicians at the scene where the recording takes place, the idea originated to use them and estimate the impulse responses with the use of an adaptive filter. Due to the simplicity of the LMS algorithm, which was first presented in 1960 by B. Widrow and M. E. Hoff, it is still the most commonly applied algorithm for the adaptation of a FIR-based adaptive filter [BW60].

4.1 System identification

There are several applications where an LMS filter can be used. System identification, inverse modeling, linear prediction and elimination of interference are the most typical cases. This project will make use of the application for system identification.

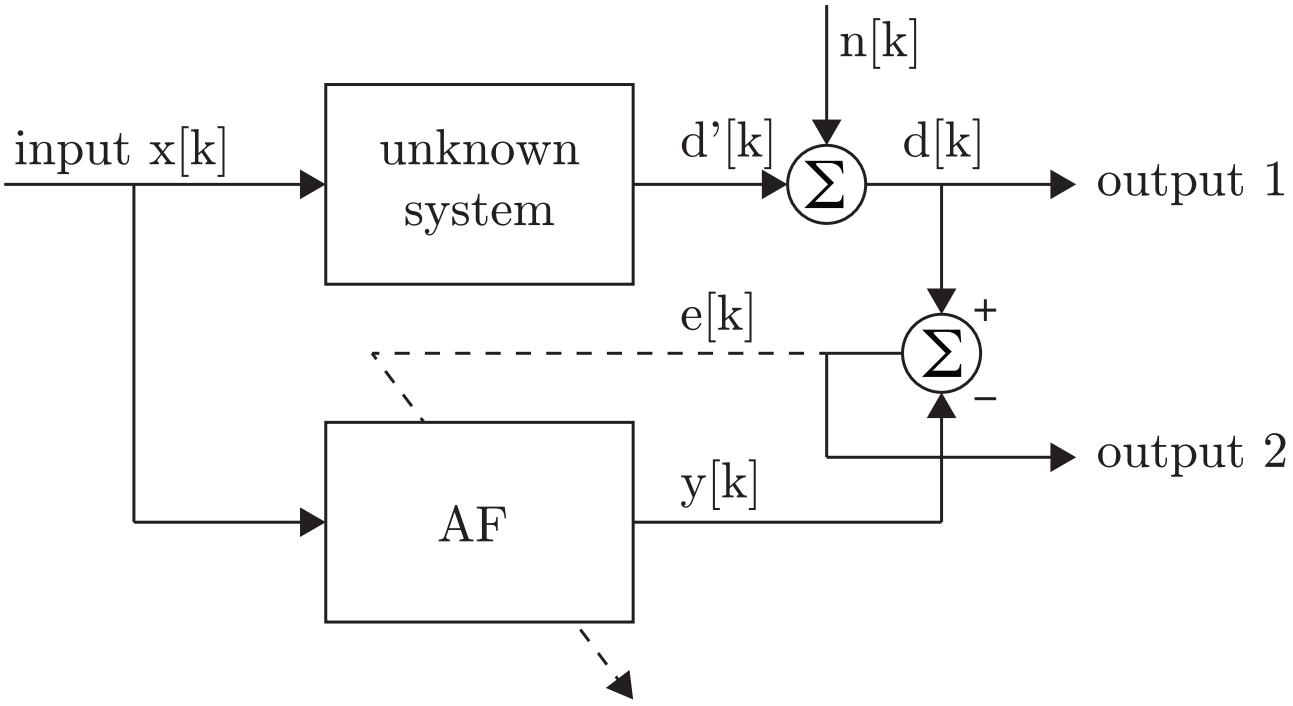


Figure 1: System identification block diagram [Hof00].

Figure 1 shows a block diagram for this kind of application, where the **unknown system** would be the room in which the recording takes place. The input signal $x[k]$ for the unknown system represents a spot microphone close to a musician, e.g. a guitar player. The desired signal $d[k]$ is one of the B-Format channels, recorded with a first-order Ambisonic microphone, which is placed in the middle of all present musicians. $n[k]$ is an additional measurement noise. In a perfect situation, for which the input signal would have to be white noise, the adaptive filter

will represent the unknown system and the coefficients will build the impulse response between the spot microphone and the array microphone. That includes not only room information, but also microphone information from both microphones.
To get the coefficients, the error value has to be calculated. According to the block diagram we can write:

$$e[k] = d[k] - y[k]. \tag{1}$$

By replacing $y[k]$ with $\underline{\omega}^t[k]\underline{x}[k]$ the equation (1) rewrites as:

$$e[k] = d[k] - \underline{\omega}^t[k]\underline{x}[k]. \tag{2}$$

Which then can be placed in the equation to calculate the coefficients vector as such:

$$\underline{\omega}[k+1] = \underline{\omega}[k] + \mu(d[k] - \underline{\omega}^t[k]\underline{x}[k])\underline{x}[k]. \tag{3}$$

The calculation of the error and coefficients at this point is done in the time domain. Due to the large filter length needed for system identification applications, it is also possible to implement the LMS filter in the frequency domain. The problem with the calculation in the time domain is that the convolution $y[k] = \omega[k] * x[k]$ is computationally intensive and is more efficient when done in the frequency domain. A disadvantage of this approach is, that the calculation is done in blocks and not per sample, so there is a latency between each coefficient update. To minimize this latency, it is possible to partition the impulse response into smaller blocks.
There is a system object present in the DSP toolbox for Matlab [TM30], which is based on the applications presented in [Far98]. Through experiments with different input signals, the **Partitioned Fast Block Least Mean Squares (PFBLMS)** algorithm was chosen for this project and will be explained in the following chapter 4.2.

4.2 Partitioned Fast Block LMS algorithm

The implementation of the named algorithm is shown in the block diagram in Figure 2 and will be explained in this chapter.

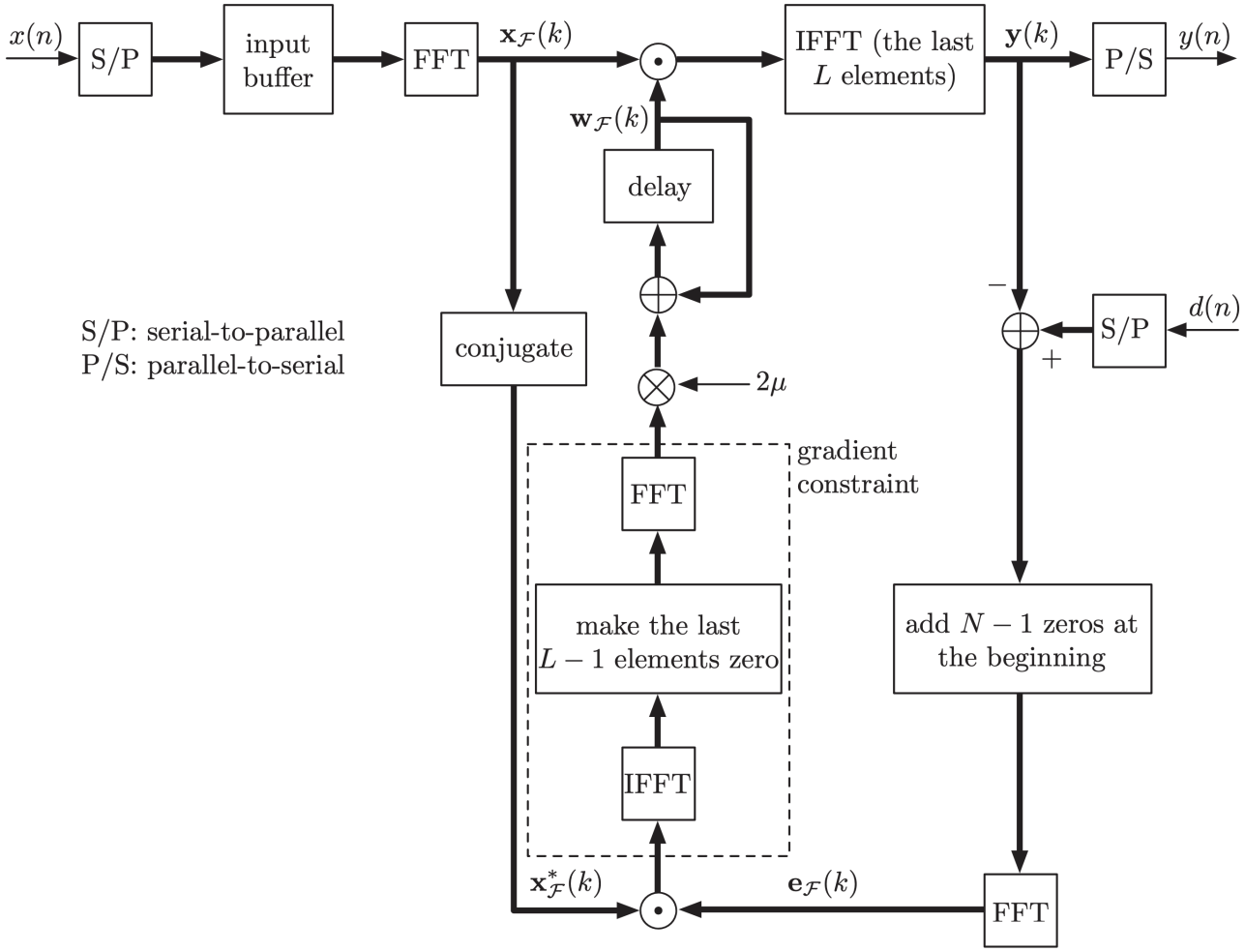


Figure 2: PFBLMS block diagram [Far98].

When we take a close look at the block diagram in Figure 2, we still see the very basics of the LMS algorithm. Because the computation in the frequency domain is done in blocks and not per sample, it is necessary to reorganize the data we send into the algorithm. This can be done with either the overlap-add or overlap-save method. The input buffer as well as the error buffer are then transferred to the frequency domain using the DFT transformation. When we take equation (3) and adapt it to the new algorithm we get:

$$\mathbf{w}_F(k+1) = \mathbf{w}_F(k) + 2\mu \mathbf{P}_{N,0} X_F^*(k) \mathbf{e}_F(k). \tag{4}$$

The term $\mathbf{P}_{N,0}$ represents the part of the block diagram with the dashed line and is used for the gradient constraint. It is necessary to ensure that the last $L-1$ elements of the time domain equivalent of the tap-weight vector $\mathbf{w}_F(k)$ are constrained to zero [Far98]. The different parameters that can be modified for this algorithm are presented in the next chapter.

4.3 Implementation and testing

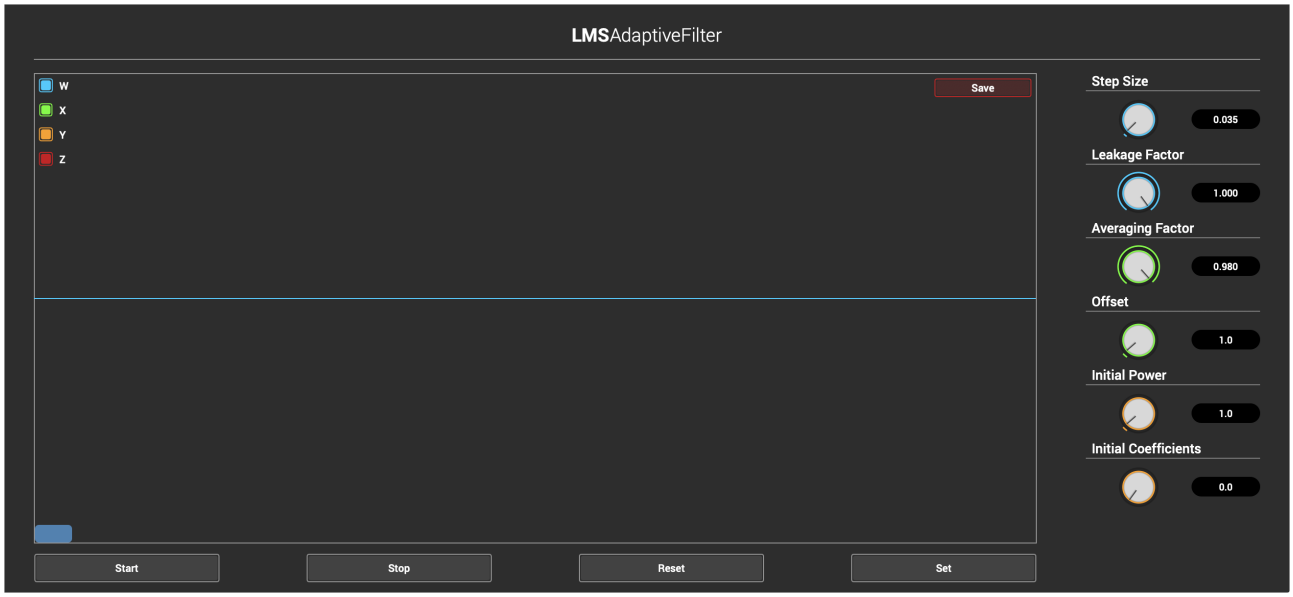


Figure 3: LMS Plugin for testing purposes

Figure 3 shows the **LMSAdaptiveFilter** plugin which was created to test and modify the algorithm discussed in chapter 4.2. With this plugin, it is possible to tweak the following parameters of the algorithm in real time using Reaper as the DAW. Once the values were found to be the best for the input signals, they were set as default (see table 1) in the main plugin for this project, which is presented in chapter 7.

Step Size Changing this parameter will modify the step size of the algorithm. For lower values, the algorithm converges slower, for higher values faster. Too high values will lead to no convergence.

Leakage Factor The leakage factor is used to tell the algorithm how much of the information should be forgotten after the calculation of each block. Setting this to 1.0 will result in no leakage.

Averaging Factor The averaging factor which will be set here is used to compute the FFT input signal powers for the coefficients updates.

Offset To avoid division by zero or by very small numbers, this parameter can be set to any non-negative real scalar. This is important when the FFT input signal power becomes very small.

Initial Power The values in the signal power vector will be initialized with this value when the filter is first created or resetted.

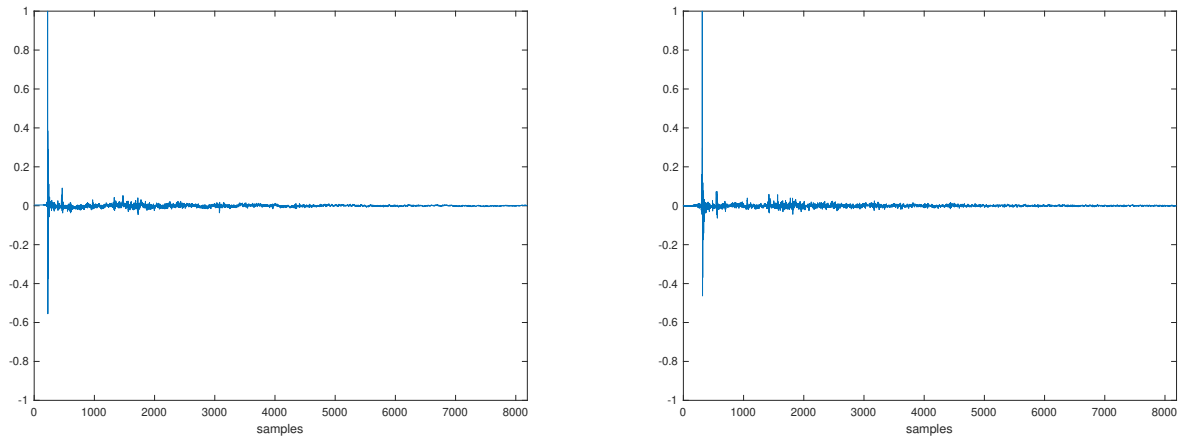
Initial Coefficients The values in the coefficient vector will be initialized with this value when the filter is first created or resetted.

Parameter	Range	Default
Step Size	(0,1]	0.035
Leakage Factor	[0,1]	1.0
Averaging Factor	(0,1]	0.980
Offset	nonnegative real scalar	1.0
Initial Power	positive numeric scalar	1.0
Initial Coefficients	scalar	0.0

Table 1: Possible range of the parameters and the default value, which was hard coded into the final plug-in.

4.4 Discussion of the LMS algorithm

What we see in Figure 4 is the measured impulse response on the left and the estimated impulse response on the right. the estimated version is based on a solo trumpet signal which was convoluted with the measured impulse response. There is no or just a very small difference noticeable when we look at the Figure. This is possible, because the trumpet signal is played through the loudspeaker and not from a real instrument. When using a real instrument, there is moving involved, which makes it hard for the algorithm to converge and which doesn't happen when played back over the loudspeaker. Still, the first peak can mostly be localized and therefore be used to build a pseudo intensity vector to calculate the azimuth and elevation angle based on the location of the array microphone. To find more than the first peak it is necessary that the input signal contains transients and has a decent signal to noise ratio. With those properties, a few reflections can be estimated.



(a) (b)

Figure 4: (a) Measured IR, (b) Estimated IR with LMS.

5 Direction of arrival (DOA)

Both approaches to estimate the DOA are based on the calculation of a pseudo-intensity vector. It is computed with the signals of the first-order Ambisonic microphone array:

$$\mathbf{I}_n = \frac{1}{\rho_0 c} \Re \left\{ w^*[n] \begin{pmatrix} x[n] \\ y[n] \\ z[n] \end{pmatrix} \right\} \tag{5}$$

with ρ_0 as the density of air and c the speed of sound. The DOA at the sample n is then expressed by the unit vector θ_n

$$\theta_n = \frac{\mathbf{I}_n}{||\mathbf{I}_n||}. \tag{6}$$

For the DOA estimation the frequency spectrum of the signals is limited from 200Hz to 4000Hz because in this range the main directional information is to be found. Below 200Hz the frequencies are too low for a good directional resolution and above 4000Hz it comes to aliasing of the microphone array because of the construction of the microphone.

5.1 Intensity Vector for a Room Impulse Response

To obtain the DOA of the direct sound from the LMS-estimated room impulse response (RIR), or any other RIR, the maximum has to be found. At the corresponding sample n the intensity vector gives the desired direction.
In the case of the estimated RIR, the DOA is not only gathered for the sample where the RIR reaches the maximum, but also for some neighboring samples in order to even out some possible errors in the LMS-Algorithm. The median value of those samples is then taken as the final DOA.

5.2 Close talker probability (CTP)

The second implemented algorithm is the close talker probability (CTP). The CTP is a filter that extracts the signal of an instrument's spot microphone from the array microphone with the help of all other spot signals. In order to be able to make use of the CTP, multiple spots for multiple instruments are needed. Otherwise the CTP simply calculates the intensity vector without filtering, which should also lead to a good DOA detection anyhow since there is no complication with only one instrument.
The CTP helps to estimate the DOA by filtering the array signal in the frequency domain and since there is no need to transform the signal back to the time domain after the DOA estimation, all transformation parameters can be optimized regarding only the frequency domain and good results for the algorithm. One assumption that has to be made is that the signal energy of instrument i is the highest at the corresponding spot microphone i. Also we assume that on average one time-frequency slot of each microphone signal is mostly dominated by a single instrument (spectral disjointedness). With these assumptions the CTP can be derived as a filter that extracts the signal of instrument i with the short-term spectral power ratios between

the signals of microphone i and all remaining spot microphones. The filter coefficient can be computed as

$$P_i(n,k) = \frac{\gamma(|X_i(n,k)^2| - max_{i \neq j}(|X_j(n,k)^2|))}{|X_i(n,k)^2|} \tag{7}$$

with

$$\gamma(z) = \begin{cases} z & if \quad z > 0 \\ 0 & else \end{cases} \tag{8}$$

to extract the signal of instrument i in time frame n and frequency bin k such that

$$Y_{ij}(n,k) = P_i(n,k) X_j(n,k) \tag{9}$$

approximates the signal of instrument i as picked up by microphone j [SFZ+16].
In our particular case j represents the Ambisonic channels W, X, Y and Z. P_i is the spectral mask for each spot microphone signal. To obtain the DOA we simply compute the intensity vector as in (5) for each filtered instrument signal i at each time frame n and frequency bin k:

$$\mathbf{I}_{i,k,n} = \Re \left\{ Y_{iw}^*[k,n] \begin{pmatrix} Y_{ix}[k,n] \\ Y_{iy}[k,n] \\ Y_{iz}[k,n] \end{pmatrix} \right\}. \tag{10}$$

Two more calculations lead to the final result: with averaging over the time frames every frequency bin has one corresponding direction of arrival and in order to get an overall result for the DOA of the instrument signal the median of all frequency-bin-directions is computed.

Since a single time frame does not deliver a good estimation and averaging over some frames is necessary, it is not possible to implement this algorithm without any delay. Therefore the directions in the plug-in are only updated once in a second which should still be sufficient for a musical performance.

6 Recording

To gather data, and in order to test the introduced algorithms, two recordings of impulse responses and musical signals in two different rooms were made. The first recording took place in the CUBE of the institute of electronic music and acoustics (IEM) where also the integrated tracking system was used to document the exact positions of microphones, loudspeakers and instruments. The recording setup is shown in Figure 5. As a first-order Ambisonic microphone the Soundfield ST450 was used and all the spot microphones had a cardioid polar pattern. Different distances of both loudspeakers and microphones where measured to cover many possible realistic setups. Anyhow for all following observations and results only the outer spot microphones are presented because both setups yielded similar results. The exact angles and distances of the microphone positions as well as the results of both algorithms are listed in table 2 and 3.

The recording in the second room at Reiterkaserne (LS82EG12) mainly served to produce a demo video for the plug-in and to produce a second set of room impulse responses to test the room parameter estimation (see chapter 8) in a different room.

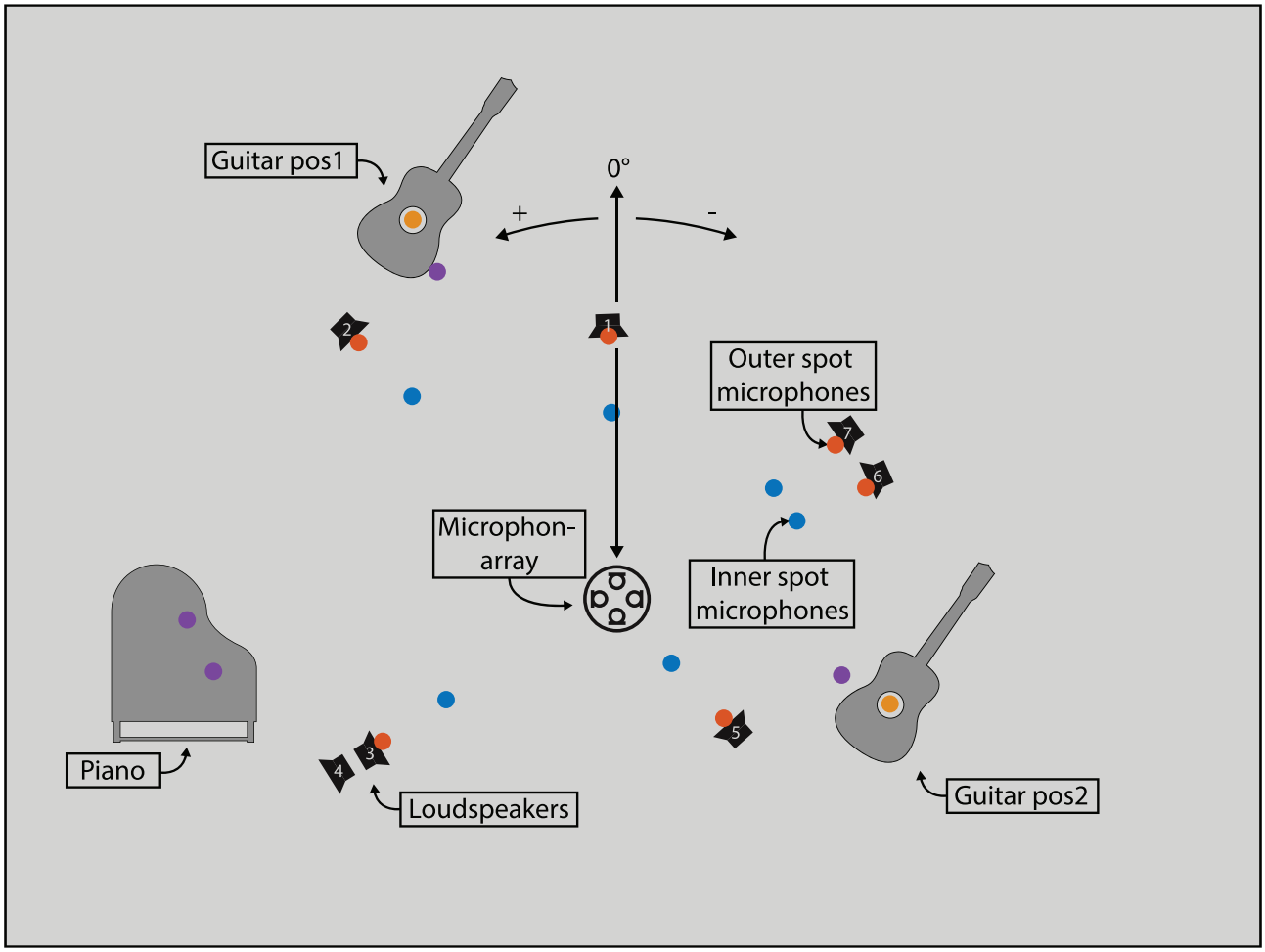


Figure 5: Loudspeaker and microphone setup.

Table 2 shows the real positions of the loudspeakers and the different estimated positions, based on the used algorithms.

Estimated LMS (solo) The positions were estimated with the trumpet signal which was already introduced in the LMS chapter. It was played as solo signal with each loudspeaker. All positions are relative to the Soundfield ST450 array microphone for the signals recorded in the CUBE.

Estimated LMS (ensemble) All six instruments were played back over the loudspeakers simultaneously. The setup contained the following instruments in the given order: trumpet, bass, piano, drums, percussion, guitar.

The same signals were used for **Estimated CTP (solo)** and **Estimated CTP (ensemble)**. Due to the orientation of the loudspeaker in position 4 (facing away from the middle) the estimations for an instrument in the ensemble at this position didn't show any relevant results. Therefore this position was ignored for the ensemble and considered as a non-realistic performance setup anyhow.

LS Position	1	2	3	4	5	6	7
Measured							
Azimuth in °	2	45	121	122	-138	-66	-55
Elevation in °	0	0	-1	-1	0	0	0
Radius in cm	235	321	244	281	146	243	238
Estimated LMS (solo)							
Azimuth in °	-6.28	43.59	127.11	126.85	-139.45	-65.00	-55.93
Elevation in °	-4.43	-4.33	-0.35	-10.38	2.19	-1.11	-0.92
Radius in cm	230	312	235	232	136	237	233
Estimated LMS (ensemble)							
Azimuth in °	-16.82	33.80	129.14	-	-124.70	-61.26	-56.17
Elevation in °	-3.97	-33.58	-1.93	-	-12.69	-1.75	-5.72
Radius in cm	225	359	240	-	136	232	233
Estimated CTP (solo)							
Azimuth in °	-1.96	41.21	122.30	117.78	-137.99	-60.07	-51.66
Elevation in °	17.71	15.96	8.87	12.71	8.41	10.34	10.13
Estimated CTP (ensemble)							
Azimuth in °	5.60	20.98	112.31	-	-86.12	-57.49	-48.03
Elevation in °	11.93	12.54	18.15	-	10.19	14.71	13.57

Table 2: Loudspeaker angles and distances.

Table 3 shows the real positions of the instruments and the different estimated positions, based on the used algorithms. All positions are relative to the Soundfield ST450 array microphone for the signals recorded in the CUBE.

Position	Piano Lo	Piano Hi	Guitar 1	Guitar 2
Measured				
Azimuth in °	93	101	29	-109
Elevation in °	-15	-7	-8	-24
Radius in cm	329	372	356	225
Estimated LMS				
Azimuth in °	124.33	63.70	20.71	-110.26
Elevation in °	-5.5	-48.00	-7.78	-19.73
Radius in cm	19	59	337	228
Estimated CTP				
Azimuth in °	102.22	97.87	22.4	-103
Elevation in °	23.67	19.24	18.52	17.63

Table 3: Instrument microphones angles and distances.

When we take a look at the values for the estimated piano position with the LMS algorithm it is obvious that the found peak is not right. This happens if the musical signal doesn't contain enough transients for the adaptive filter to converge. In this case the piano was playing long accords with soft attack.

7 Plug-in user interface

Figures 6 and 7 show the final plug-in with six spot microphones. In Figure 6 the plug-in is set to calculate the DOAs with the Close Talker Probability. The setting of which algorithm should be used is marked with A. In section B the directions are plotted on a sphere and in the upper left corner the activity threshold can be set which defines a threshold from which a source is detected active. Only for an active source the DOA is updated. C shows the azimuth (blue) and elevation (green) angles in degrees and also the distance of the source in meters though which can only be calculated with the LMS algorithm. Each source can be selected to show the according estimated impulse response and also locked if the direction should not change.

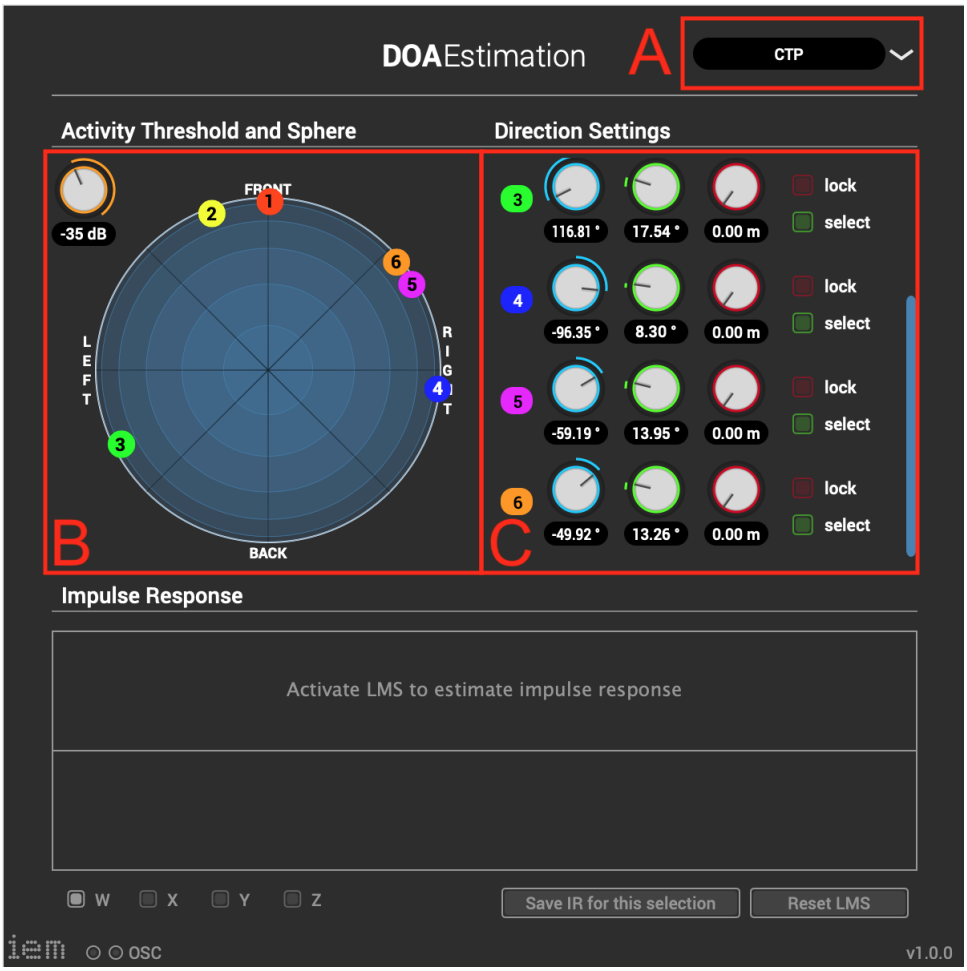


Figure 6: User interface of the plug-in, set on CTP DOA Estimation.

In Figure 7 the second source is locked (D), since the algorithm was not able to estimate the DOA correctly. In the same Figure source four is selected to show its estimated impulse response (E). Marked with F in Figure 7 is the display of the impulse response where it is also possible to view the single first-order Ambisonic channels (w, x, y, and z). Furthermore it is possible to reset the algorithm, e.g. when the input signal changes, and also to export the result in a .wav file.

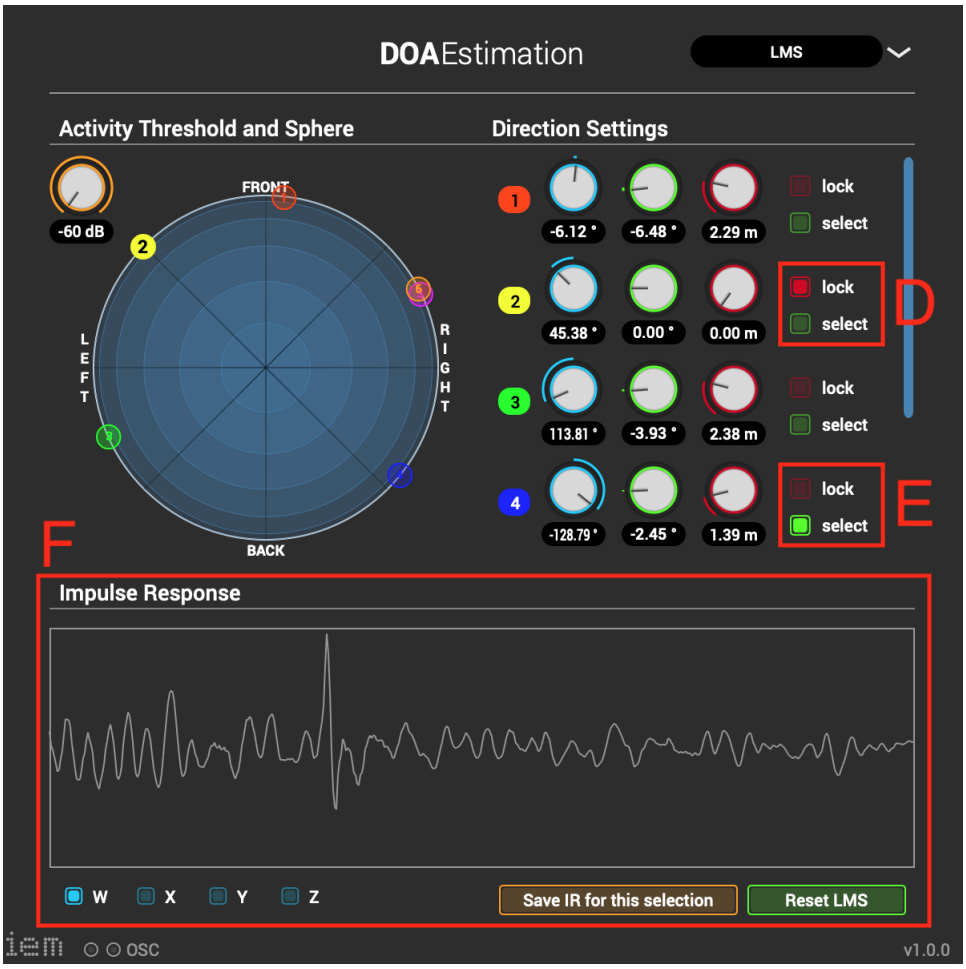


Figure 7: User interface of the plug-in, set on LMS DOA Estimation.

8 Room parameters

The initial idea for this plug-in was to also add an estimation for the room dimensions in which the musical performance takes place. In theory with the estimated room impulse responses from the LMS algorithm it should be an easy task to extract the directions of arrival of the direct sound (as it is used for the DOA estimation) and some early reflections. With this and the delays at which the direct sound and the reflections are located in the impulse responses, the positions, where the sources and the reflections are located, can be estimated. With multiple sources and therefore multiple impulse responses estimated at different source locations there should be sufficient information to recreate the dimensions of the room if enough reflections are located. With the measured impulse responses, it is indeed possible to approximate the room dimensions, but it proved to be a difficult task to extract enough information from the estimated impulse responses from musical signals. Since no stable real-time solution for this problem was found, we decided to include only the two different approaches to estimate the DOA in the plug-in and discuss the room dimension estimation in a separate chapter. The following method was tested in two different rooms: first the CUBE at the institute of electronic music and acoustics and second the LS82EG12 at Reiterkaserne, both located in Graz.

8.1 Peak Detection

The goal was to create an algorithm that is able to automatically estimate the dimensions of the room and therefore it is avoided to pick the peaks of the early reflections by hand. The chosen approach is a kurtosis with a threshold and static settings as suggested in [Ush10]. Considering a signal $\mathbf{x}$ where always one block of m samples length centered about the time sample n is observed, the kurtosis sample k_n at time n is computed by comparing the sample x_n with the mean μ_n and the standard deviation σ_n of the whole m-length block

$$k_n = \frac{(x_n - \mu_n)^4}{\sigma_n^4}. \tag{11}$$

The kurtosis is high if x_n is large compared to the samples $\mathbf{x}_n$ of one block. Otherwise if the block with samples $\mathbf{x}_n$ contains a flat or bi-modal distribution, the kurtosis will be low. In [Ush10] also a modified kurtosis is suggested which introduces a second window of length l that is also centered about n but is smaller in size. Then the kurtosis can be calculated with

$$k_{n,m,l} = \frac{(\mu_{n,l} - \mu_{n,m})^4}{\sigma_{n,m}^4} \tag{12}$$

which offers more parameters to adjust the peak detection. To avoid multiple peaks that actually only belong to one reflection, the kurtosis is smoothed with a Hann-window of 80 samples length.

For one detected peak in the kurtosis (Figure 8) the DOA of 20 samples around this peak will be observed in the impulse response and with a histogram approach it is determined if the peak is a possible reflection or if the peak that was found does not deliver a stable DOA. In order to fully recreate the room virtually it is assumed that the room is a regular room with 6

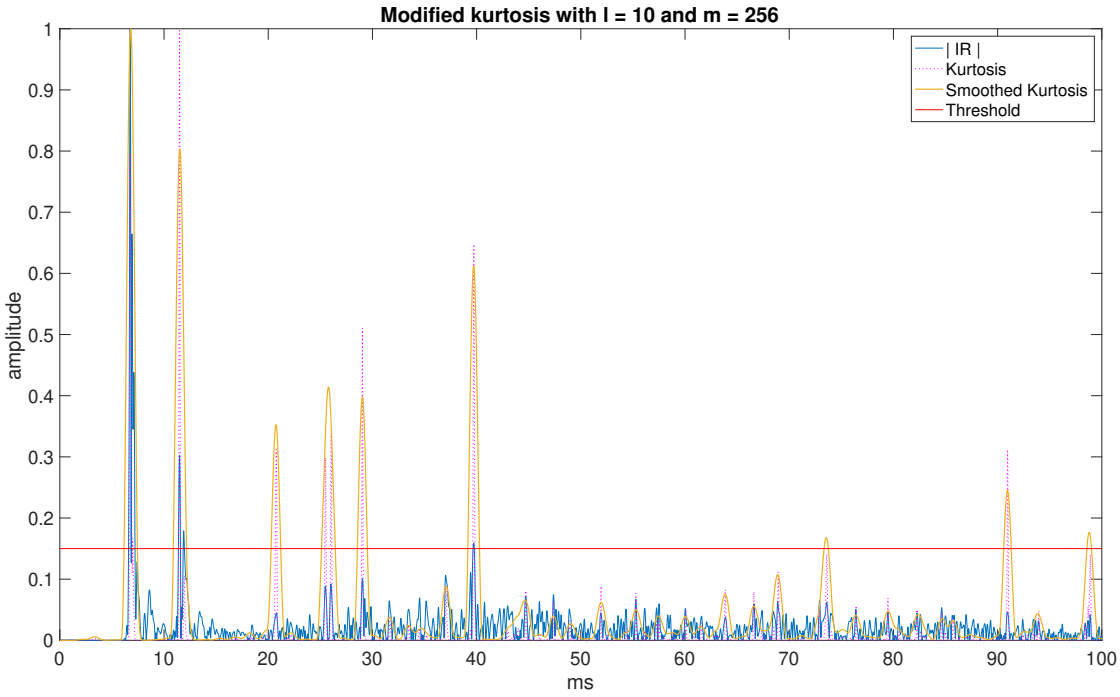


Figure 8: Kurtosis of one of the measured impulse responses (IR) in the CUBE.

surfaces and therefore the first 7 detected peaks of the kurtosis are considered as direct sound and the six first-order reflections from each surface.

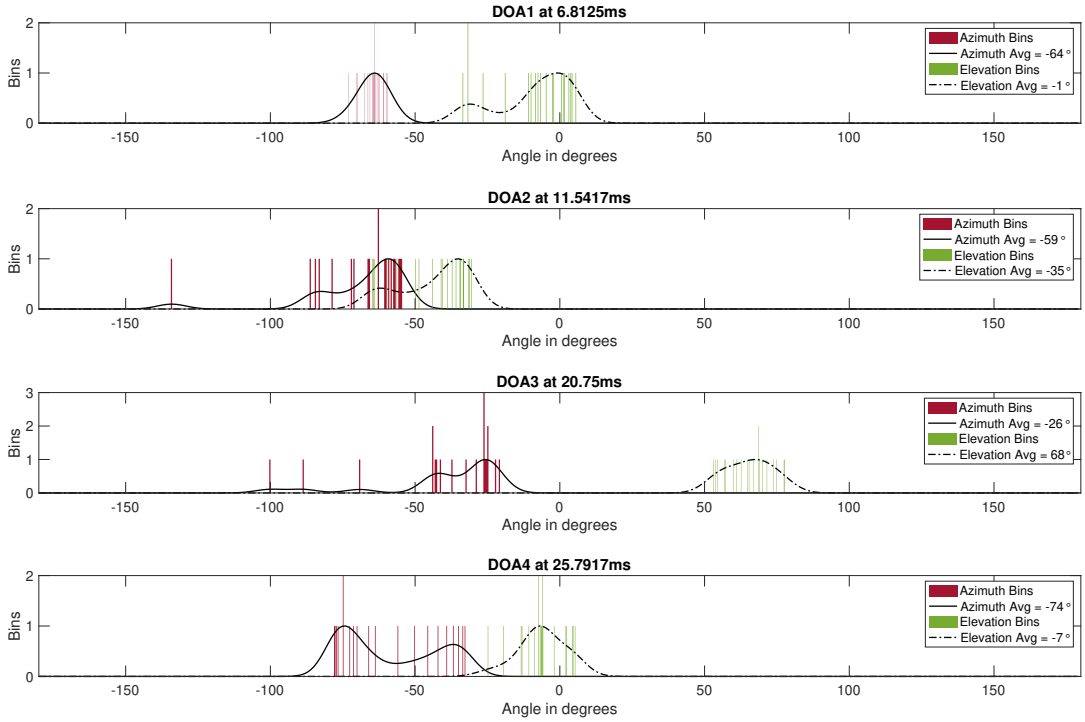


Figure 9: Histogram of the DOA of 20 samples around the first four detected peaks from Figure 8.

To finally extract the DOA of a peak from the histogram (Figure 9) the Matlab function `fitdist` with a kernel distribution (black lines in Figure 9) is used and the location of the maximum of the distribution is considered as the DOA. Furthermore it is tested if the detected peak is reliable or if it was detected by fault. 60% of all DOA samples from one peak in the histogram have to lay within 15° of the maximum of the distribution function. If this is not the case, the DOA of this peak will not be used for the room estimation.

8.2 Positions of the reflections in the room

For the estimation of the early reflections the cosine law is used to calculate the distance at which the reflection was located. Since the DOAs can be estimated as shown before, the azimuth and elevation angles of the direct sound and at least some reflections are known as well as the delay of the direct sound a and the total delay that the reflection needed to arrive at the microphone array $b + c$. It is assumed that an early reflection on the floor or ceiling has approximately the same azimuth angle as the direct sound and an early reflection on a wall has approximately the same elevation angle as the direct sound. To determine whether the captured reflection comes from the floor/ceiling or wall, simply the differences of the azimuth/elevation angle between direct sound and observed reflection were compared and if the elevation difference is higher the cosine law is computed with the elevation angle and vice versa if the azimuth angle is higher.

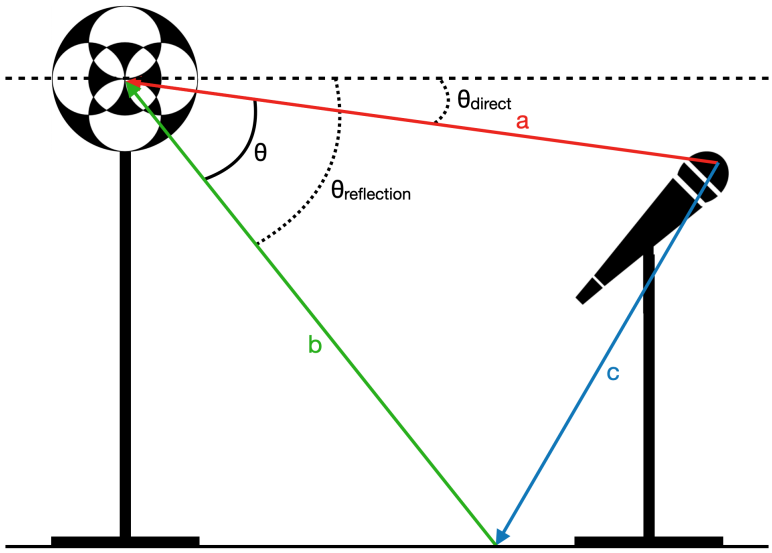


Figure 10: Estimation of the position of a first reflection on the floor.

$$\theta = \theta_{reflection} - \theta_{direct}. \tag{13}$$

$$cos(\theta) = \frac{a^2 + b^2 - c^2}{2ab}. \tag{14}$$

$$b = \frac{a^2 - (b + c)^2}{2(acos(\theta) - (b + c))}. \tag{15}$$

Another reliability test for wall reflections is executed with the calculated positions. For this test we assume that the walls are regular straight walls so that all first-order wall reflections are expected approximately on the same height as the source. A detected wall reflection will only be considered as such if the z-component lays within 0.5m of the z-component of the source position. Otherwise this reflection is not used for the room estimation.

8.3 Room Dimensions

With the estimated positions of the sources and the first six reflections, the borders of a virtual room can be generated by extracting the maximum distance in each axis. Therefore each position was taken with its x, y, and z coordinates and all values were compared with one another. Assuming that it is possible to capture at least one reflection on the floor and one from the ceiling, the highest and the lowest z-value provide the height of the room. Discovering that it was more difficult to get a reflection from each wall from our measurements, the width (y) and the length (x) of the room were taken from the maximum absolute value in y- and x directions if there was no reflection detected on both sides. With this method, it must also be assumed that the center of the recording (the microphone array) is also the center of the room according to length and width. The results of the estimated room dimensions are shown in table 4.

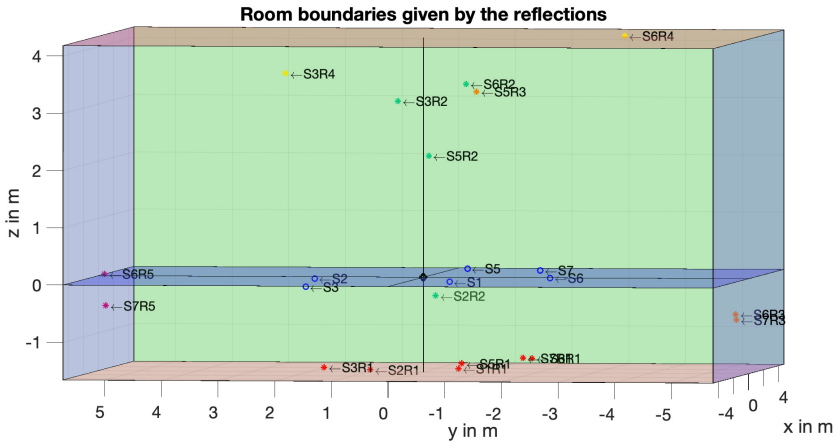


Figure 11: Estimated CUBE dimensions from the back; S1 = source one, S1R1 = first reflection caused by source one.

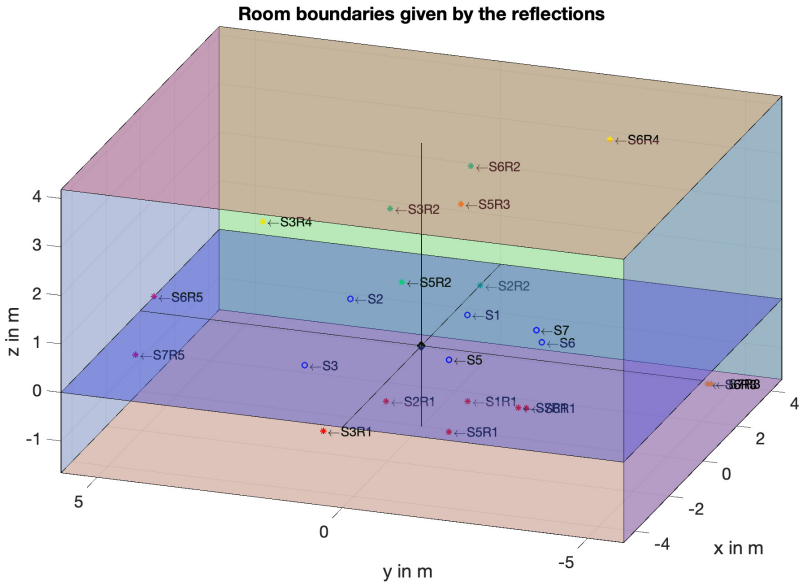


Figure 12: Estimated CUBE dimensions from the top.

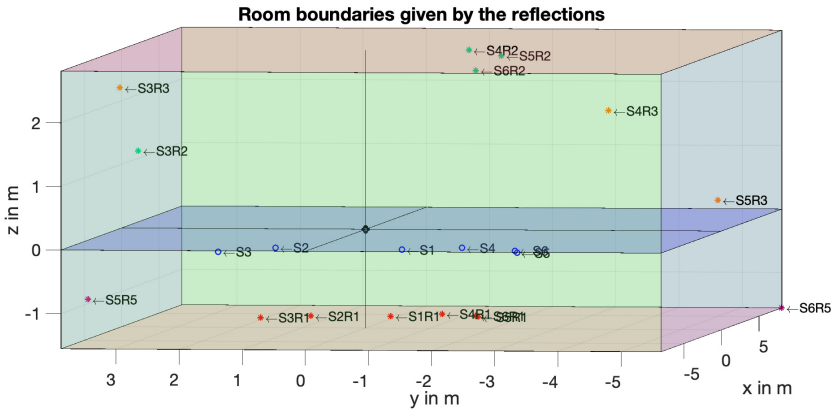


Figure 13: Estimated LS82EG12 dimensions from the back.

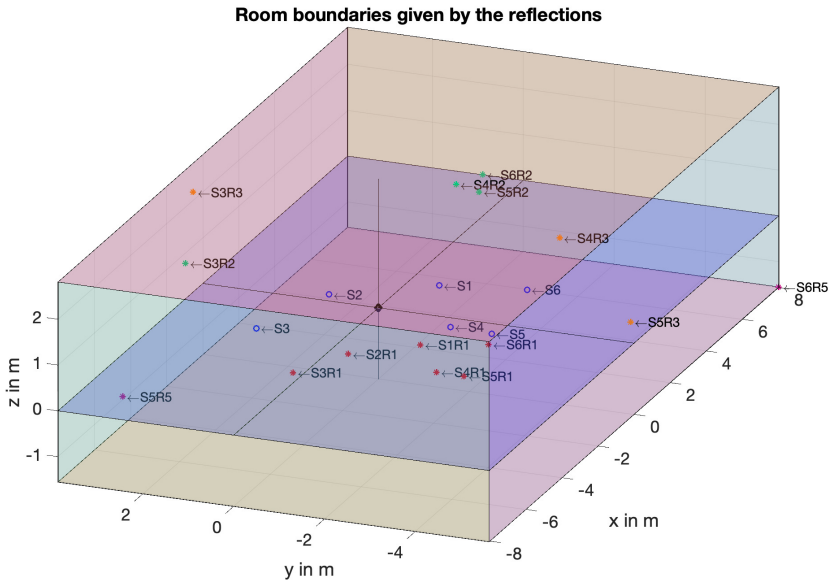


Figure 14: Estimated LS82EG12 dimensions from the top.

Dimensions	Height in m	Length in m	Width in m	Volume in m^3
CUBE	5	10	12	600
Estimated CUBE	5,8	11,5	9,3	620
LS82EG12	4,2	8,6	17	614
Estimated LS82EG12	4,3	9,5	16,1	658

Table 4: Comparison of actual- and estimated room dimensions.

Two problems with this approach can be observed. First, when looking at the estimated height of the CUBE in Figure 11, it seems that the reflection S6R4 which defines the height, was either detected by fault or it is not a first-order reflection and therefore the delay was higher than a first-order reflection would have had. Because of this reflection the room was estimated too high even though the majority of ceiling reflections is located at the correct height. If this is corrected this approach delivers a good result for this room.
The second observation is that in the other room in Figure 14, only one reflection (S6R5) could be detected in the front or back and this reflection could also be of a higher-order since the position is below the first reflections coming from the floor. This error could be explained with the shape of the room which is rather long compared to the other dimensions and therefore the majority of first reflections arrives from the side walls, floor and ceiling before the ones from the front wall. Expanding the amount of peaks to look for in the kurtosis could find more reflections from the front but could also cause detection of higher-order reflections from the other room borders.
We conclude that this method can lead to good results but also depends on the room dimensions itself.

8.4 Second approach

Although the first method that was introduced yields satisfying results, it requires many source positions in order to receive enough reflections to reconstruct the dimensions of a room. The second idea was to create a method that estimates the room dimensions with only one source and one detected reflection. With one source position and therefore at least one impulse response it is possible to calculate the reverberation time T which is the base of the following approach. The first step is similar to the first approach and an initial room with the dimensions limited by the source and a first reflection is assumed. Then as input parameters the absorption coefficients of the room borders are needed to calculate the room volume with the Sabine reverberation equation as in formulas (16) to (18).

$$\text{Sabine: } T = 0,163\frac{V}{A} \tag{16}$$

with

$$A = \sum_i \alpha_i S_i \tag{17}$$

where α_i are the absorption coefficients of the room borders and S_i are the according surfaces. With the given α_i as inputs and the calculated reverberation time, the volume can be computed.

$$V = \frac{1}{0,163}TA. \tag{18}$$

With the knowledge of the volume the firstly estimated minimal room can be expanded until it matches the volume calculated in (18), regarding some restrictions that normal rooms usually have. E.g., the height is the smallest dimension and all lengths differ with at least 10%. This method was only tested with the measurements of the CUBE and only with rather guessed absorption coefficients, but it also led to an acceptable approximation. However this idea was abandoned as well since it needs too many input parameters and the approximation is probably less accurate than a simple visual estimate.

9 Conclusion

In this project, a plug-in was created that estimates the positions of one or more instruments without any other information but the musical signal itself picked up by a first-order Ambisonic microphone array and spot microphones. Two different approaches to achieve this were introduced and implemented, both with a less accurate result for instruments which emit only low frequency sounds. Anyhow it was possible to estimate the positions even of those low frequencies but only when the instrument was playing separately. With the first algorithm introduced which estimates the positions from an impulse response extracted with an LMS algorithm it was possible to also collect the information of how far from the microphone array the instrument is positioned. The second approach, the CTP, proved to be less accurate mainly regarding the elevation angles.
Additionally, an estimation of room dimensions from impulse responses was shown, which initially was supposed to find a place in the plug-in. Due to the fact that the room estimation yields better results with more source positions and the LMS impulse response estimation gets worse with more source positions, it was not possible to implement a stable algorithm that estimates the room dimensions correctly. Furthermore the LMS could not compute an impulse response with enough detectable first-order reflections, which is crucial for the chosen approach of estimating the room. Therefore, we took the decision to look at this topic as a second part of this project. Nevertheless, with the sweep measurements made in addition to the instrument recordings the data to evaluate the room-estimation-algorithm was collected and in the end it showed that it is possible to get a satisfying approximation of the room dimensions, if the amount of source positions is sufficient. The most important and difficult component of the algorithm was a way to detect the peaks of the reflections automatically. The chosen peak picking method, the modified kurtosis, proved to detect a satisfactory amount of peaks which also led to a quite good estimation. Nonetheless, there are many parameters that influence the result of the detection and each impulse response would need an individual set of parameters to achieve a perfect peak detection. Since the goal was an automatic algorithm without hands-on parameter adjustment a compromise was found which made it possible to find enough peaks in every impulse response without changing the settings.

List of Figures

List of Tables

References

[BW60] B. Widrow, E.Hoff: Adaptive switching circuits. In: IRE Wescon Conv. Rec. Part 4 (1960), S. 96–104

[Far98] FarhangBoroujeny, B.: Adaptive Filters. 1. Chichester England : Wiley, 1998

[Hof00] Hofbauer, G. Moschytz· M.: Adaptive Filter. 1. Berlin : Springer-Verlag Berlin Heidelberg GmbH, 2000

[SFZ+16] Schörkhuber, Christian ; Frank, Matthias ; Zotter, Frank ; Höldrich, Robert ; Grosche, Peter: Automatic Mixing for Immersive Teleconferencing Systems. In: DAGA 2016 Aachen 54 (2016), Nr. 3, S. 359–362

[TM30] The MathWorks, Inc.: dsp.FrequencyDomainAdaptiveFilter. `https://www.mathworks.com/help/dsp/ref/dsp.frequencydomainadaptivefilter-system-object.html`. Version: 2020-11-30

[Ush10] Usher, John: An improved method to determine the onset timings of reflections in an acoustic impulse response. In: The Journal of the Acoustical Society of America 127 (2010)