Multichannel Quad 4.0 Reverberation with Feedback Delay Networks (FDN)

MUMT 307: Music and Computing II
Ma. Carolina Rodríguez

April 29th, 2020

1 Abstract

From the multiple approaches that have been studied for the production of reverb, the use of recursive filters is one of the most efficient approaches, as when treated properly through different gain factors, they can result in the expansion of the impulse response of a signal in time with a fixed decay rate. Feedback Delay Networks (FDN), are the applications of this filters with the implementation of a mixing matrix (that can take on different characteristics), that is required to be lossless, providing the additional combination of signals to feedback fixed delays and the optimum amount of diffussion. For this project, a FDN system is implemented to create a multichannel reverberation were all the channels have certain amount of correlation and decorrelation between them after being obtained with different combinations of a mixing matrix output. Additional filtering is added to improve the overall frequency response and provide a more "natural" decay for all frequency bands.

2 Introduction

There are several approaches to achieve reverberation through artificial means. Physical modelling, due to the complexity and density of the nature of reverberation, can easily become a very resource-demanding and unnecessary

process, up to a certain point, as we can not discriminate very fine characteristics in such short time frames. The density or reverberation grows fast, thus what we can mostly perceive is its decaying characteristics and frequency response. Other approaches to artificial reverberation have a more statistical and macro strategy. That is the case of Schoreder reverberators, for example, where a combination of comb and all pass filters depict the early reflections and the exponential decay of reverberation.

Feedback Delay Networks describe a system in which a set of delay lines are interconnected together through a mixing matrix in order to provide combination, and then feedback the delay lines. They can be thought of as being an IIR filter in which the first delay lines output exactly the delay lengths provided, and are then are fed back with a now "unknown" delay and amplitude. Therefore, it is a constant change of the feedback coefficients (depending on the choice of the matrix).

Several considerations have to be taken into account, such as: frequency response changes in time, loss of energy inside the combination loop, first delay lengths (pre delay), number of input and output channels, relation between reproduction channels, amount of combination in the mixing matrix, among others. These will be treated in more detail in the following sections. [1] [2] [3] [4] [5]

3 Materials and Methods

3.1 Inputs

The number of inputs of a modeled reverberation depends on how many sources we have incoming to our ears. These all could be summarized in:

1. a single source and its incoming sound from the three dimensions of space.
2. a series of delay lines that model the reflections of the sound and its behaviour in time.
3. a subject listening through two channels (each ear).

[3] [4]

3.2 Direct to reverberation ratio

The Direct-to-reverberant ratio is the relation of sound arriving in a straight path from the source to the listener, that have reflected once or twice off the surfaces around the source. If the output of the reverberation is going to be used at the end of the chain as a representation of the whole source-space system an additional path direct patch is necessary. It runs directly from the source towards the output. An extra gain factor can be added to control the amount of direct signal wanted in the resulting sound (wet/dry). [3]

3.3 Pre Delay - Early Reflections

Three distinct sections in time can be determined when analyzing reverberation. The first one is the direct sound coming from the source(s) directly. This sound has no significant energy loss (just that of the propagation medium). The second part, is the early reflection section, also known as pre delay; sound coming from the source that has already impacted anywhere from two to a few surfaces. The end of this part is marked when the reverberation becomes dense enough to be thought of as complex and dense decaying signal. Usually, this section is compared to decaying noise (with less presence of higher frequencies).[6] [1]

In this model, these first reflections can be obtained with the first pass of the signal through the entire delay line before the feedback loops. Another possibility is to add a separate set of delay lines that can come from a tapped delay line. Careful attention has to be given to the lengths of this delays, so that they are in accordance with the desired space that is being represented and to ensure that they do not have common factors, to prevent the reverberation of producing build ups at certain frequencies.

3.4 Reverberation time (T60)

As determined by Sabine, reverberation time is defined as how long it takes for a signal to decay -60 dB when no input signal is present anymore [7]. When setting an impulse as the input to a system, it should take $n = 1 + (t60 * fs)$ samples to reach the $-60dB$ point. This is modeled by setting the exponential decay of the system to match the desired time. In this digital

model, the reverberation time is dependent on the gain factors of each of the feedback delay outputs. As these are IIR filters, their impulse response will therefore be infinite. Even if the reverberation time can not be set to a determined value because of the nature of the filters, it has to be taken into account that the rest of the response will either be end being truncated by the resolution of the system (its calculation precision), or by a given parameter the developer assigns (in either time or magnitude terms affecting the filter). [7] [4] [8]

3.5 Mixing matrix

Diffusion is a noticeable characteristic of reverberation. If the sound is not diffuse enough it will tend to "ring" more in certain frequencies than others. This can be corrected by applying subsequent filters, but it can also be addressed through the use of a mixing matrix. The purpose of it is to provide enough combinations between the existing delay lines to provide a less even dispersion of reflections in time, thus generating a more "random" or "noise" like behaviour of the reverberation tail.

It is important to note that the used mixing matrix has to comply with certain conditions, including the conservation of most of the energy in its input. If the combinations are overly lossy the desired reverberation time will be harder to achieve, and the resulting impulse response of the system will be too short. Everything done inside of the feedback loop will be propagated very fast through the system, as it is the section that provides continuous input back to the delays, even if no more incoming signal is present.

The mixing matrix choices are limited to unitary and triangular matrices, including the Householder Matrix, which provides diffusion with a lot of efficiency, requiring only additions to be computed; and the Hadamard matrix, with a larger amount of diffusion but requiring more computational resources. It is important in most cases to make sure the dimensions of the matrix, and therefore the amount of delay lines in the system, is a power of two: 2^N. [9] [4] [2]

3.6 Feedback gain factors - Feedback Filters

After the mixing matrix, the loss of some energy is necessary to produce a decaying impulse response. Simple gain factors can be applied if desired. This way, the reverberation's decay will resemble a system were all frequencies decay roughly at the same rate. However, reverberation in natural spaces has different behaviours in frequency. For lower frequencies, for instance, reverberation time is usually longer. Providing frequency dependent gain at this stage (filtering), can enable the developer with tuning possibilities. As mentioned in the previous section, special care has to be taken when calculating the coefficients for this filters, which are usually of a significantly smaller order than those used for the delay lines.

The dependence of the reverberation time on this filters is crucial, as the energy loss rate in the loop will determine the impulse response decay time, and with the added filters, its tonal characteristics as well. [9] [4]

3.7 Multichannel Outputs

The final stage in a multichannel reverberator is making a compromise between correlation and decorrelation in the outputs. Several changes can be made to several characteristics to achieve this: frequency response, time arrival and output source. Each of these correspond in the physical world to: HRTFs (Head Related Transfer Functions), Interaural Time Differences, Interaural Level Differences, and position in space. If the difference is too significant, both channels will start to become completely decorrelated from each other resulting in an almost "multi-mono" perception, where two separate sources are reproduced in two separate ears.

For a true multichannel reverberation, it is necessary to keep some characteristics equal and then altering them to a certain extent to succesfully mimic the differences between two ears. For a two-channel stereo reverberation, for example, a simple time (delay) decorrelation could be applied to each channel. In another system with rear channels, certain filters could be applied to the back channels to account for the difference in frequency response produced by the pinna when a source is behind the listener.

In this particular reverberator, different combinations of the delays outputting from the mixing matrix-filtering stage were combined to provide

some decorrelation between channels. No set combination has been studied to provide the best results for this, but with some experimentation different results can be obtained. A gain reduction was applied as well. [10] [1]

4 Implementation

Below, the block diagram of this FDN application shows the structure of this particular FDN Reverberation application.

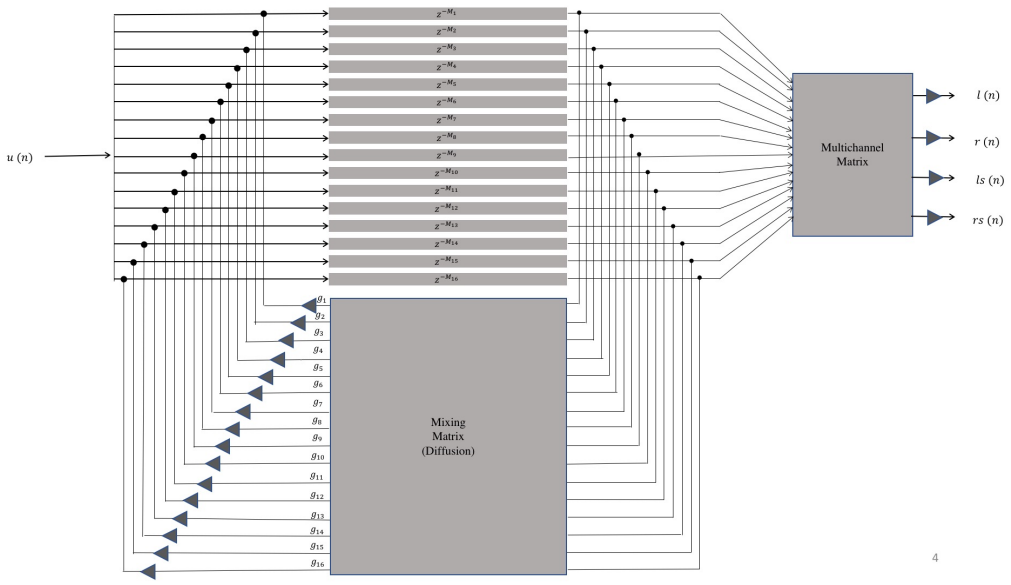


Figure 1: Block Diagram

All of these sections were discussed in the previous numeral. This diagram was the substrate to build the reverberator. MATLAB was used to write it, therefore a way of computing everything in an iterative way, sample by sample, was necessary. Saving the sample states of each of the delays, each of the filter coefficients and each of the gain factors was fundamental to achieve this. The implementation of the iterative sample processing can be consulted in Figure 2. [4]

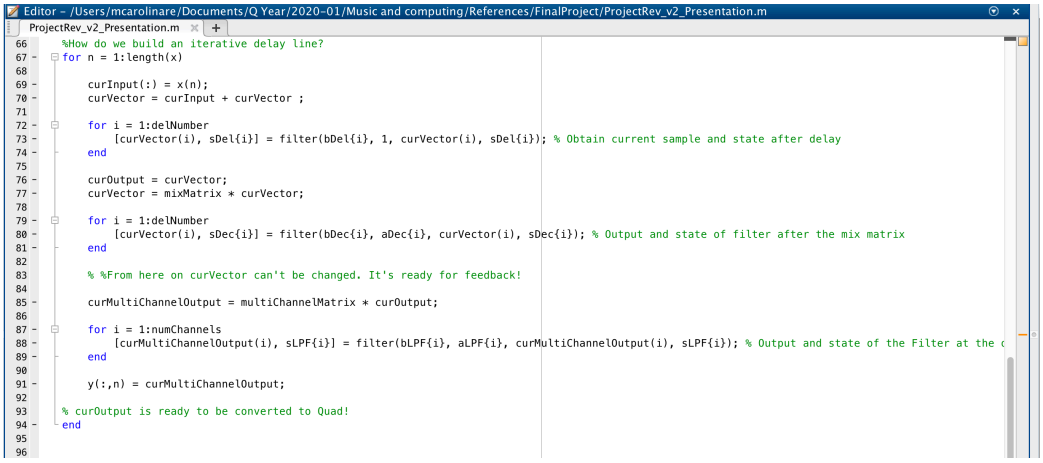


Figure 2: Iterative sample processing

4.1 Definitions

- INPUT: Mono input. It was necessary to build a copy of the input signal to enter each of the delay lines.

- DELAY LINES: Sixteen delay lines were used, and none of them had common factors to prevent the reverberation from being weighted in frequency.Two different delay length combinations can be heard in the examples (Figure 3). in the first version the delay lengths are more spaced in time, whereas in the second version the lengths are set to be closer prime numbers. To calculate the starting delay the Sabine-Norris-Eyring equation for the Mean Free Path and the Schroeder equation for the minimum Mode Density were used.

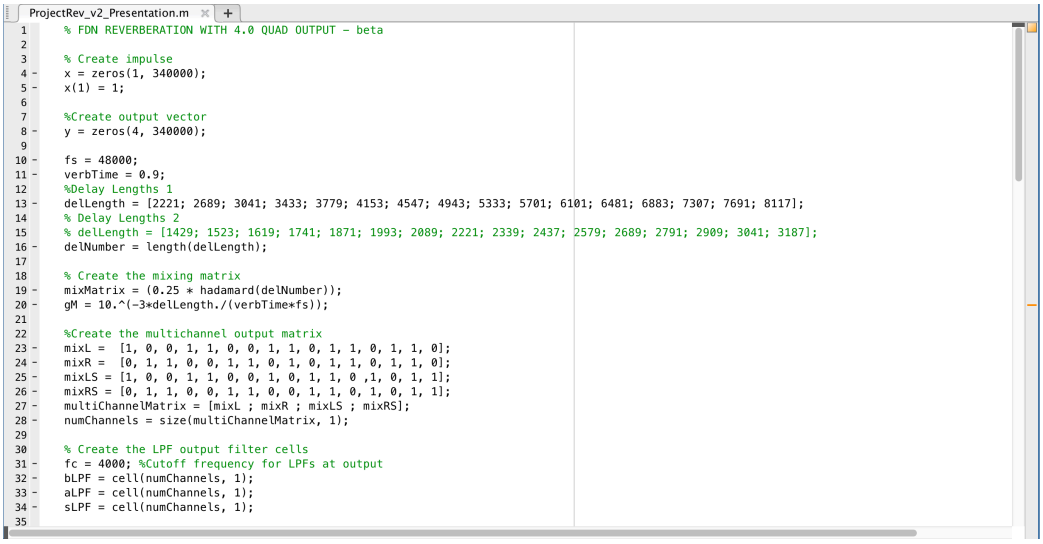
  The computation of the delay lines for this version are shown in Figure 7, using the 'filter' function. Two example scripts with possible optimizations of this calculations can be seen in Figures 4 and 5.[5] [2]

$$MeanFreePath = 4 \cdot \frac{Volume}{Surface} \tag{1}$$

$$ModeDensity \geq 0.15 \cdot ReverbTime \cdot SamplingFreq \tag{2}$$

- MIXING MATRIX: The Hadamard matrix was chosen for this application. Delay lengths had to therefore be a power of two (2^4).

  -



```
ProjectRev_v2_Presentation.m
% FDN REVERBERATION WITH 4.0 QUAD OUTPUT - beta

% Create impulse
x = zeros(1, 340000);
x(1) = 1;

%Create output vector
y = zeros(4, 340000);

fs = 48000;
verbTime = 0.9;
%Delay Lengths 1
delLength = [2221; 2689; 3041; 3433; 3779; 4153; 4547; 4943; 5333; 5701; 6101; 6481; 6883; 7307; 7691; 8117];
% Delay Lengths 2
% delLength = [1429; 1523; 1619; 1741; 1871; 1993; 2089; 2221; 2339; 2437; 2579; 2689; 2791; 2909; 3041; 3187];
delNumber = length(delLength);

% Create the mixing matrix
mixMatrix = (0.25 * hadamard(delNumber));
gM = 10.^(-3*delLength./(verbTime*fs));

%Create the multichannel output matrix
mixL =  [1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0];
mixR =  [0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0];
mixLS = [1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0 ,1, 0, 1, 1];
mixRS = [0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1];
multiChannelMatrix = [mixL ; mixR ; mixLS ; mixRS];
numChannels = size(multiChannelMatrix, 1);

% Create the LPF output filter cells
fc = 4000; %Cutoff frequency for LPFs at output
bLPF = cell(numChannels, 1);
aLPF = cell(numChannels, 1);
sLPF = cell(numChannels, 1);
```

Figure 3: Creation of necessary variables for the implementation

```
%Delay lines: Method 2

x = zeros(1, 100000);
x(1) = 1;
y = zeros(size(x));
maxDelay = 20;
delBuffer = zeros(maxDelay, 1);
delay = 10;

readPos = 1;
writePos = 11;

for i = 1:length(x)
    curSample = x(i);

    delBuffer(writePos) = curSample;
    curSample = delBuffer(readPos);

    writePos = mod(writePos + 1, maxDelay);
    readPos = mod(readPos + 1, maxDelay);

    y(i) = curSample;
end
```

Figure 4: Method 2: Optimization of delay line calculations

-

-

```
1    %Delay lines: Method 3
2
3    x = zeros(1, 100000);
4    x(1) = 1;
5    y = zeros(size(x));
6    maxDelay = 20;
7    delBuffer = zeros(maxDelay, 1);
8    delay = 10;
9
10
11   for i = 1:length(x)
12       curSample = x(i);
13
14       delBuffer(delay+1) = curSample;
15       curSample = delBuffer(1);
16       delBuffer = circshift(delBuffer, 1);
17
18       y(i) = curSample;
19   end
20
21
22
```

Figure 5: Method 3: Optimization of delay line calculations

To be lossless this matriz requires to be multiplied by a factor that is also associated with its dimensions. In this case:

$$MMFactor = \frac{1}{2 \cdot \sqrt{2}} \tag{3}$$

MATLAB has a function that calculates the Hadamard matrix when given the dimensions. To build this matrix, this function was used and then scaled by the given factor. This can be seen in Line 19 of the script in Figure 3. [2] [9] [6]

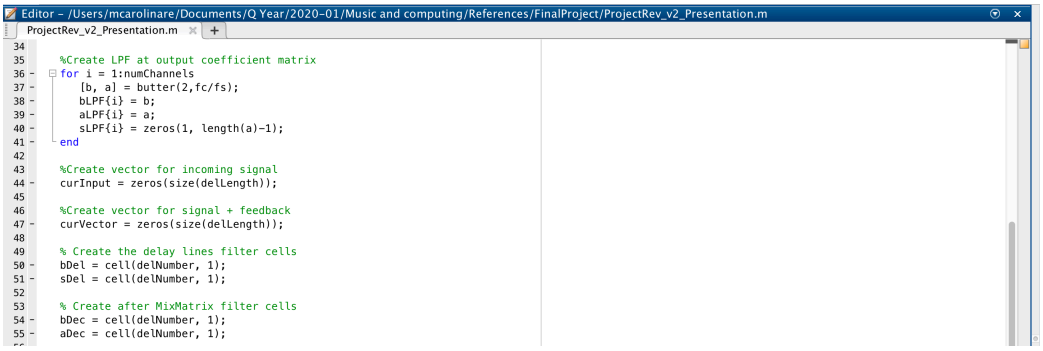


Figure 6: Creation of state keeping vectors

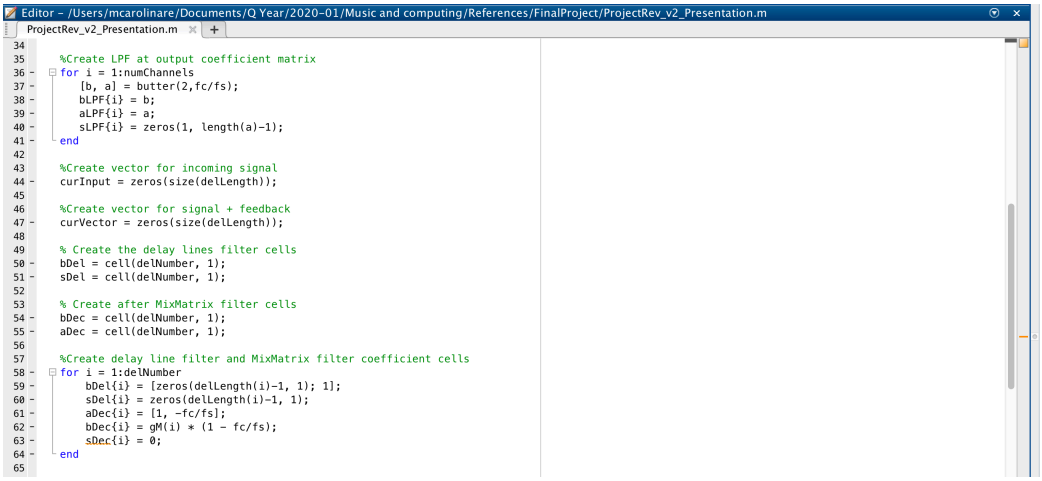


Figure 7: Filling the delay lines and preparing filter coefficient cells

- INTERNAL FILTER (Post mixing matrix): This is possibly the most delicate part of the system, as it will alter the reverberation time and the frequency response of this decay. Before this step the system is completely lossless and has an infinite impulse response with no decay, it can thus also be unstable. In this particular case, a series of low pass filters was applied to each of the outputs of the matrix. The computation of these decay filters can de seen in Figure 7. The filters need to have a smaller order, small amount of energy loss and a relation to the original delay lengths. The gain factor for the filter used here follows this expression: [4]

$$g_i = 10^{\frac{-3 \cdot M_i \cdot T_s}{verbtime \cdot fs}} \tag{4}$$

- MULTICHANNEL MATRIX: Through this matrix the final output vectors are obtained. To design how this matrix could work, instead of writing a single vector and transform it to have different characteristics on each channel, similar channels (Left and Right and Back and Front) shared half of their sample. Left and Right shared has the exact same sum of the last 8 delay lines, as did Left Surround and Right Surround. Left and Left Surround shared the exact same sum of the first 8 delay lines, so did Right and Right Surround. To consult the exact mixing arrangements Lines 22 through 28 of the script can be consulted in Figure 6.

- OUTPUT FILTERS: In order to have further control of the overall frequency response of the reverb, an additional set of low pass filters

were added, in this case of the Butterworth type, where the minimum phase shift is located in the band pass region. In this particular case all of the filters had the same cutoff frequency that can be set at the begining of the script. Also, a $-3dB$ gain reduction on the rear channels was performed. The creation of the coefficients for this filter can be seen in Figure 6, and their calculation in Figure 7.

5 Evaluation

5.1 Technical

Some optimizations can certainly be made. I started calculating the delay lines with the filter function, in future version they will been optimized to be a pointer and a vector writing in certain positions. This future optimization is shown in Figures 4 and 5 as a test code. MATLAB is a very convenient environment to work on this projects, especially if one does not have enough background to write a first script in a lower-level language. One of the possible optimizations could actually be trying to implement this same algorithm using C++.

Using cells in MATLAB instead of regular matrices opened a new set of building possibilites, as different vector or matrix sizes can be "grouped" together to obtain different lengths or charactertistics of every parameter in each iteration. The hardest part of this implementation was being able to create all of the necessary vector, matrices and cells with the proper sizes before running the code, as this made it more difficult to debug, and more prone to undetectable errors.

Some improvements could be made with relatively small changes in the code, but to make it work perfect and build several presets, more research is necessary. Even if this is not exactly physical modeling, having a precise acoustic characterization of a particular room or space could lead to a more informed design.

5.2 Sonic

I asked a sound engineer graduate to describe sonically what the impulse responses in the examples were the most closed to according with presets on digital reverberators that he uses on a daily basis. All their answers can be found in quotes below.

5.2.1 Delay lengths v.1

- 0.9 Seconds: "Plate like. Metallic."
- 1.5 seconds: "Plate like. Metallic."
- 3 seconds: "Plate like, much larger. Metallic. Same damping as the previous one."
- 5 seconds: "Plate like. Metallic. Symmetric reflections."

5.2.2 Delay lengths v.2

- 0.9 Seconds: "Very small room, for example a bedroom. More absorption.:
- 1.5 seconds: "Larger volume room. It is possible to distinguish were the reverb ends. More balanced in frequency to what we are used to."
- 3 seconds: "A hall. Balanced in frequency. Certain reflections can be heard individually."
- 5 seconds: "A controlled hangar with absorption treatment. More 'lineal development' of the reverb in time."

"In general, the Delay lengths v.1 where less clear than in v.2. The reverb tail is more noticeable in v.1 than the first reflections, so it sound less natural and more metallic; v.2 represents a space where it is easier to tell its dimensions and acoustic qualities. Pre delay is more clear in this version as well"

* Note of the author: The 'clearness' characteristics of v.2 could be explained as the delay lengths are closer together numerically than the others.

None of the lengths that were used in this application use subsequent prime numbers, therefore the first reflections are much spaced in time, signaling a larger space as well.

6 Conclusions

This was a great project for me, as personally I am very interested on the recreation of space, the interaction of acoustic sources and the way our brains "decode" them. I would like to continue feeding this project by exploring initialization settings that can give the charactertistics of different physical spaces, but also including binaural processing. The interesting thing about the digital world is that we are not bounded by physical phenomena and we can create things that are not available for us in the physical world; thus, exploring the sonic possibilities of the perceptual limits of reverberation is a tangible possibility. I have been permanently using all kinds of reverberators in the past, analog and digital, and I knew the theory behind them to certain extent, but delving into how they are actually build and how a sample-by-sample process occurs was something I never had the chance to explore or try.

This project was also very challenging, as I did not have any previous experience with text-based programming languages and I had been interested in starting for a while. Even if the script now looks short and is quite understandable with the comments, I had to have everything very well designed and understood before even running the code as an experiment, figuring out the exact lengths and sizes of vectors and matrices and making sure I was overwritting variables in a proper way. I am looking forward to using this first reverberation presets in my own project, even if it takes very long to run. Eventually I could get it running in real time.

7 Improvement possibilities

- Properly scale gain through the system as right now there is some clipping (even after normalizing).
- A change in the mixing matrix could signify a change in the quality of the diffusion. This could be interesting to try, especially when trying

to create presets that simulate a space with different surface materials.

- Using a different set of internal filters. Instead of using low pass filters, for example, use bell or shelving filters, as their gain loss can be assigned quite precisely, and therefore, with 16 different diffuse delay outputs, more inherence could be possible over several frequency band decays.
- Add direct path to sum at the output with a gain factor, so that the wet/dry or direct-to-reveberation ratio can be altered.
- Add a Pre delay path to provide a defined set of early reflections (from 0 to 100 ms approximately.
- Provide more combinations of delay lengths to build different early reflections characteristics and a different mode density during the reverberation decay, where the gains of each particular delay can also be assigned.
- Implementing more precise perceptual filtering to each of the outputs of the reverberation to obtain a more realistic sense of space and spatialization. Here, Head Related Transfer Functions or filters that simulate just the pinna could be used. Further level handling could also provide an additional layer of control.
- Experiment with several combinations of the multichannel matrix, were each of the samples coming from the delay could also be weighted to provide more decorrelation.
- Explore what happens if instead of having a mono source (input signal) a multichannel input was used. Question the possibilities of keeping the given decorrelation of the channels but making them still appear in the same virtual space.
- Include other signal processors in the system to design "non-natural" reverb models, such as a gated or inverse reverb, exploring how this could also affect the mixing matrix diffusion and characteristics.

References

[1] U. Schlemmer, "Reverb design," in *Pure Data Convention, Weimar-Berlin, Germany*, 2011.

[2] J. O. Smith, *Physical audio signal processing: For virtual musical instruments and audio effects*. W3K publishing, 2010.

[3] J. O. Smith, "Mus420/ee367a lecture 3 artificial reverberation and spatialization, year = 2018, url = https://ccrma.stanford.edu/ jos/Reverb/Reverb.pdf, urldate = 2020-04-01."

[4] G. P. Scavone, "Mumt 618:feedback delay networks (fdn)," 2020.

[5] G. P. Scavone, "Mumt 307:feedback delay networks (fdn)," 2020.

[6] D. Rocchesso and J. O. Smith, "Circulant and elliptic feedback delay networks for artificial reverberation," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 1, pp. 51–63, 1997.

[7] S. Schlecht and E. Habets, "Accurate reverberation time control in feedback delay networks," 09 2017.

[8] S. J. Schlecht and E. A. Habets, "Modal decomposition of feedback delay networks," *IEEE Transactions on Signal Processing*, vol. 67, no. 20, pp. 5340–5351, 2019.

[9] S. J. Schlecht and E. A. Habets, "On lossless feedback delay networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 6, pp. 1554–1564, 2017.

[10] F. Menzer, "Binaural reverberation using two parallel feedback delay networks," in *Audio Engineering Society Conference: 40th International Conference: Spatial Audio: Sense the Sound of Space*, Audio Engineering Society, 2010.