Virtual Acoustics for Immersive Audio Workshop
Week 1 Assignment

Orchisama Das and Gloria Dal Santo

This week's course topics will cover fundamentals of room acoustics and artificial reverberation - acoustic parameters, the feedback delay network (FDN) structure, how to optimize its parameters via gradient descent, and the analysis and synthesis of coupled spaces.

Before moving on:

- Go to the Google Drive at `Material/Assignments/Data/Week 1`
- Download the content of the folder and place it in the `/data` folder of the workshop repository

Day 1: Introduction to Room Acoustics

In this first assignment, you will learn how to compute the main parameters used in room acoustics to describe the properties of the impulse response of a space. Namely, the Energy Decay Curve (EDC), the Reverberation Time (RT or T_{60}), and the Normalized Echo Density (NED).

You will be using Room Impulse Responses (RIRs) measured in the *Promenadikeskus* concert hall in Pori, Finland. The responses were measured using an omnidirectional sound source on the stage and an omnidirectional microphone [1]. Figure 1 shows the floor plan of the concert hall. We'll be using the RIR recorded from position R4 while source S3 was playing.

Energy Decay Curve

The energy decay curve is computed from the Schroeder backward integration, as follows

$$\mathrm{EDC}_{\mathrm{dB}}(t) = 10 \log_{10} \sum_{\tau=t}^{L} h^2(\tau) \ . \tag{1}$$

Where $h(t)$ is the RIR and L is its length. Absorption, whether by the air or by the objects and materials in the room, is frequency-dependent, causing a frequency-dependent energy decay of the RIR. For this reason, it is more practical to compute the EDC at different frequency bands. If finer frequency resolution is desired, one can use the Energy Decay Relief computed from the Short-Time Fourier Transform (STFT) of the RIR. To compute

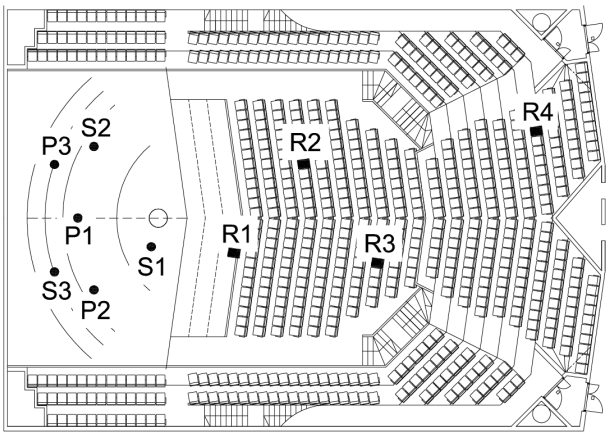


Figure 1: Floor plan of the Pori *Promenadikeskus* concert hall with indicated Source (S) and Receiver Positions (R/P). Adapted from [1].

the EDC at different frequency bands, $h(t)$ is first processed by a filterbank, and then the output of each band h_{f_c} with center frequency f_c will be backward integrated

$$\mathrm{EDC}_\mathrm{dB}(t; f_\mathrm{c}) = 10 \log_{10} \sum_{\tau=t}^{L} h_{f_\mathrm{c}}^2(\tau) \; . \tag{2}$$

Note: When using filterbank, it is important to use filters with unit norm, to avoid adding a bias to the overall energy.
Similarly, the EDR can be computed as follows

$$\mathrm{EDR}_\mathrm{dB}(t, f) = 10 \log_{10} \sum_{\tau=t}^{L} \mathbf{H}^2(\tau, f) \; . \tag{3}$$

where $\mathbf{H}$ is the STFT of h

Reverberation Time
From the EDC, you can now estimate the RT. To do so, you can use simple linear least squares regression on an interval of the EDC. You might want to start somewhere below 3dB from the initial energy. The first milliseconds of the EDC refer to the early and direct reflections, which, due to their sparsity, will lead to a staircase-like curve, which might introduce errors in the RT estimation. When the signal-to-noise ratio (SNR) is too low due to loud background noise, we can either estimate the RT from an interval shorter than 60dB or subtract the noise floor.

Normalized Echo Density
The NED examines the percentage of impulse response taps lying outside the local standard deviation, and it ranges from zero, indicating few echoes, to around one, indicating a fully

dense reverberation having Gaussian statistics. It is defined as follows:

$$\eta(n) = \frac{1}{\text{erfc}(1/\sqrt{2})} \sum_{\tau=n-\delta}^{n+\delta} w(\tau) \left|\{|h(\tau)| > \sigma\}\right| \tag{4}$$

where $|\{\cdot\}|$ indicates the cardinality of a set, $2\delta + 1$ is the length of the window w in samples, σ is the standard deviation of the windowed response, $\text{erfc}(1/\sqrt{2}) = 0.3173$ is the expected fraction of samples lying outside a standard deviation from the mean of a Gaussian distribution.

Assignment
The assignment is divided into the following points. Specific instructions are given in the Assignment's Jupyter notebook.

- Load, plot, and reproduce the impulse responses.
- Compute the EDC and estimate the RT from it using linear regression.
- Compute the frequency-dependent EDC and compute the RT at each band.
- Compensate for the influence of the background noise by noise subtraction.
- Compute and plot the EDR and the NED.

For this assignment, you will have to complete functions in `src.room_acoustics.analysis.py` to analyze the RIRs that you will find in `data/..`

Day 2: Artificial Reverberation

The most common method used to measure the impulse response of a room is to play a sine sweep in the room and record it. After that, we need to isolate the room impulse response from the sine sweep response. To do so, we have to perform a deconvolution. In this assignment, you will work on a response recorded in a racketball court, depicted in Figures 2. The court has size 6.1m × 12.2m × 6.1m.

Sine Sweep Recordings
A sine sweep signal with exponentially varying frequency is described as follows:

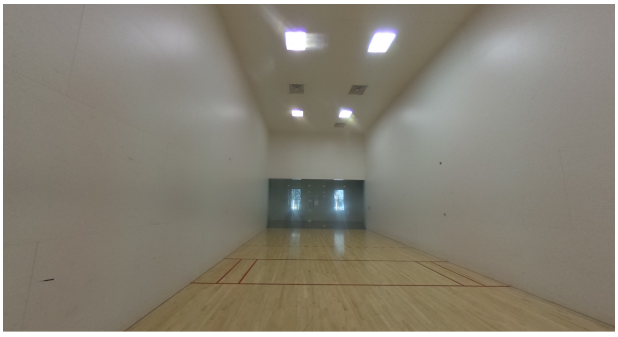
$$s(t) = \sin\left(\frac{\omega_1 T}{\ln(\frac{\omega_2}{\omega_1})}\left(e^{\frac{t}{T}\ln(\frac{\omega_2}{\omega_1})} - 1\right)\right) \tag{5}$$

where ω_1 is the starting frequency in rad/s, ω_2 is the ending frequency in rad/s, and T is the total duration in seconds. To deconvolve the room impulse response $h(t)$ from the recorder sweep response $y(t) = h(t) * s(t)$, you just need to flip the sweep signal and convolve it with the recorded signal, i.e.

$$h(t) = y(t) * s(-t) \tag{6}$$

where $*$ indicates the convolution operation.

(a)



(b)

Figure 2: Pictures of the racquetball court in which impulse responses were measured. (a) Image of the racquetball court taken from the door, facing the back wall. (b) Image of the racquetball court taken from the back wall, facing the door.

Image-Source Model (ISM)
For shoebox room geometries, like the one in Figure 2, the ISM is expected to produce a good approximation of the original RIR. For the ISM synthesis, we'll be using `pyroomacoustics` available at this link, which has a fast implementation of both ISM and ray tracing for general polyhedral rooms. We'll be using the `pyroomacoustics.ShoeBox` class to simulate the room. The absorptive and scattering properties of the room are given in the Jupiter notebook.

Assignment
In this assignment, you will

- Load, plot, and reproduce the recorded response and the original sine sweep used for the recording.
- Using the function implemented for Assignment 1, compute and plot EDC, RT, and NED.
- Use `pyroomacoustics` to synthesize the RIR using the ISM.
- Compare the EDC, RT, and NED between recorded and simulated RIRs.
- Convolve the RIR with the given anechoic signals.

Note that because we are not reproducing exactly the absorptive and scattering properties of the surfaces in the room, we should expect deviation from the ground truth.

Day 3: Artificial Reverberation with Delay Networks

Delay networks provide a fast and highly parameterized method for synthesizing room impulse responses. Among the different structures we discussed in this morning's class, Feedback Delay Networks (FDNs) are the most popular for simulating late reverberation and

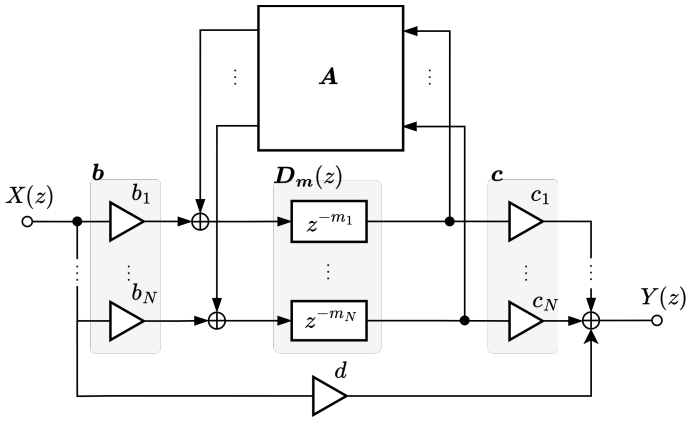


Figure 3: Block diagram of an FDN with N delay lines.

favored for their simple structure. In this assignment, you will implement an FDN class for time-domain processing and explore various parameter choices. The structure of a generic FDN is depicted in Figure 3, whose time-domain recursion is as follows:

$$y(n) = \mathbf{c}^\top \mathbf{s}(n) + dx(n) \tag{7}$$

$$\mathbf{s}(n + \mathbf{m}) = \mathbf{A}\mathbf{s}(n) + \mathbf{b}x(n) \tag{8}$$

where $\mathbf{A}$ is the feedback matrix, $\mathbf{b}$ and $\mathbf{c}$ are the input an output gains respectively, $x(n)$ is the input signal at time sample n, $y(n)$ is the output signal, and the elements of $\mathbf{s}(n)$ are the states of the delay lines $\mathbf{m}$

Among all the parameter values that we have to choose, delay line lengths require particular care. There are a few rules of thumb that you should follow to design a good-sounding FDN:

- The sum of the delay lines should be at least 6000 samples (this coincides with the number of modes),
- Use co-prime numbers to avoid degenerated delay patterns (i.e., clustering, common prime factors, and low-order dependency),
- Delay line lengths should be logarithmically distributed.

Moreover, the feedback matrix should be such that strong circulating short delays are avoided (this is a bit trickier to design).

You can implement a frequency-independent homogeneous decay. This refers to the case where all modes experience the same decay rate, i.e., $|\lambda_i| = \gamma$ for $i = 1, \ldots, N$. Homogeneous decay is achieved with a feedback matrix $\mathbf{A}$ being the product of a orthogonal matrix $\mathbf{U}$ and a diagonal matrix $\mathbf{\Gamma}$, whose entries are delay-proportional absorption coefficients, $\mathbf{\Gamma} = \text{diag}(\gamma^{\mathbf{m}})$, where the vector $\gamma^{\mathbf{m}}$ represents the γ value raised to the power of each corresponding delay in $\mathbf{m}$. The feedback matrix can be expressed as

$$\mathbf{A} = \mathbf{U}\mathbf{\Gamma} \, . \tag{9}$$

Table 1: *Values of the delay-line lengths for three different sizes N. In the delay set #1, all the delay lengths are logarithmically distributed prime numbers. For the delay set #2, half of the delay lengths are prime numbers with similar low values, and half are logarithmically distributed.*

N	**Delay Set #1: "Good"**
4	[1499, 1889, 2381, 2999]
6	[997, 1153, 1327, 1559, 1801, 2099]
8	[809, 877, 937, 1049, 1151, 1249, 1373, 1499]
	Delay Set #2: "Bad"
4	[797, 839, 2381, 2999]
6	[887, 911, 941, 1699, 1951, 2053]
8	[241, 263, 281, 293, 1193, 1319, 1453, 1597]

As $\mathbf{U}$ is orthogonal, the modal decay is controlled entirely by the gain-per-sample parameter γ, where $0 \leq \ \gamma \ \leq 1$. The gain-per-sample in dB is

$$\gamma_{\text{dB}} = \frac{-60}{f_{\text{s}} T_{60}} \,, \tag{10}$$

where f_{s} is the sampling rate in Hz, and T_{60} is the reverberation time in seconds.

Assignment
In this assignment, you will

- Generate different types of feedback matrices, namely: identity, random orthogonal, Hadamard, Householder, and circulant. Their definitions can be found in [2]. These feedback matrices are all orthogonal, ensuring system stability.
- Implement time-domain processing for a mono input and output, with an arbitrary number of delay lines.
- Process the anechoic signals used in the second assignment with the FDN. You may also want to process a simple impulse to listen to the impulse response of the FDN.
- Vary the FDN parameters and its size. Observe and listen to the effects that different parameter choices have on the output.

In Table 1, you can find some good and bad sounding delay lines that you can use.

Day 4: Differentiable Artificial Reverberation

Today, you will implement a differentiable FDN using the frequency sampling method. You will not start from scratch; instead, you'll use `flamo`, an open-source library for frequency-domain differentiable audio processing [3]. Let's recall the transfer function of an FDN:

$$H(z) = \mathbf{c}^\top \left[\mathbf{D}_{\mathbf{m}}(z)^{-1} - \mathbf{A} \right]^{-1} \mathbf{b} + d \tag{11}$$

In `flamo` in order to differentiate through the FDN, eq. (11) is sampled on a vector of linearly-spaced frequencies from 0 to π rad/s:

$$\mathbf{z}_M = [e^{\jmath\pi\frac{0}{M-1}}, e^{\jmath\pi\frac{1}{M-1}}, \dots, e^{\jmath\pi\frac{M-2}{M-1}}, e^{\jmath\pi}], \tag{12}$$

where M is number of frequency bins (equivalent to `nfft` in the code), and $\jmath = \sqrt{-1}$.

In `flamo`'s `processor.dsp` module, you have access to many differentiable DSP classes that you can use to construct your differentiable FDN. An example is also available at this link. Matrix $\mathbf{A}$ is implemented using a learnable `Matrix`-class instance with orthogonal mapping. Input and output gains are learnable `Gain`-class instances. Delays of given lengths are an instance of the `parallelDelay` class. The attenuation filters are then implemented using the class `parallelGEQ`. We will go through the example (available on the `flamo` repository) together to better understand `flamo`'s API.

The goal is to learn all the FDN parameters (all but the delays) to match a target RIR. In class, we observed that if the FDN is not properly parameterized, it can produce a metallic-sounding reverberation that lacks naturalness. We can mitigate this by optimizing the feedback matrix using the following sparsity loss function:

$$\mathcal{L}_{\text{sparsity}}(\mathbf{U}) = \frac{\sum_{i,j} |U_{ij}| - N\sqrt{N}}{N(1-\sqrt{N})}, \tag{13}$$

where $\mathbf{U}$ is the orthogonal feedback matrix, without the attenuation component.

The sparsity loss $\mathcal{L}_{\text{sparsity}}(\mathbf{U})$ needs to be combined with a loss that can capture the energy decay of the RIRs. To this scope, we can use the `flamo.optimize.loss.edr_loss` class, which is a loss based on the distance between the EDR of the target and FDN IRs.

Assignment
In this assignment, you will

- Build a differentiable FDN using the `flamo` library, configure delay lines, feedback matrices, and attenuation filters to simulate room acoustics.
- Load a target RIR, preprocess it by removing the onset time and normalizing its energy, then create a dataset for training.
- Complete the sparsity loss class to promote temporal sparsity in the feedback matrix, and configure the trainer with both sparsity loss and EDR loss to match acoustic characteristics.
- Compare the learned impulse response against the target through time-domain plots, spectrograms, and convolution with anechoic signals.

Day 5: Modeling Acoustics in Coupled Spaces

According to the common slope model for late reverberation, the EDC of the late reverb of a RIR, measured at any position within a coupled space for the b^{th} frequency band, can be

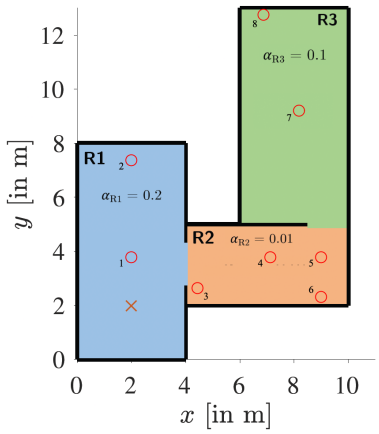


Figure 4: Layout of the coupled rooms set up where $\times$ indicates the source position and $\circ$ the 8 different receiver positions.

expressed as a sum of K exponentials,

$$d_b(\mathbf{x}, t) = \sum_{k=1}^{K} A_{k,b}(\mathbf{x}) e^{-\frac{13.8t}{T_{60,k,b} f_s}} + N_{0,b}(\mathbf{x})(L - t) \tag{14}$$

where the common decay times $T_{60,k,b}$ are position- and direction-independent, the amplitudes and noise terms $A_{k,b}(\mathbf{x})$, $N_{0,b}(\mathbf{x})$ are position- and direction-dependent. In this assignment, you will analyze the common slope parameters of a subset of spatial RIR. The complete dataset can be found on zenodo. Note that you don't need to download the dataset, as we will use a subset of 8 SRIRs already provided in the GDrive folder. The SRIRs were measured at different positions across the three rooms, as depicted in Figure 4.

Assignment
In this assignment, you will

- Load a set of SRIR from the `srirs_sampled.pkl` pickle file and plot their EDC,
- Complete the partially coded `decay_kernel` function in `room_acoustics.synthesis` that generates exponential envelopes with specified reverberation times,
- Use the `decay_kernel` function with common decay times and amplitude parameters to synthesize EDCs across 8 frequency bands (63Hz to 8kHz), plotting both individual slope components and their combined result,
- Filter the original SRIRs using a filterbank to get frequency-specific EDCs, then compare these ground truth measurements against the synthesized EDCs by plotting them together.

References

[1] Juha Merimaa, Timo Peltonen, and Tapio Lokki, "Concert hall impulse responses pori, finland: Reference," *Tech. Rep*, 2005.

[2] Sebastian Schlecht, "Fdntb: The feedback delay network toolbox," in *International Conference on Digital Audio Effects*. DAFx, 2020, pp. 211–218.

[3] Gloria Dal Santo, Gian Marco De Bortoli, Karolina Prawda, Sebastian J Schlecht, and Vesa Välimäki, "Flamo: An open-source library for frequency-domain differentiable audio processing," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.