Audio Engineering Society

Late Breaking Demo Paper

Presented at the AES International Conference on
Artificial Intelligence and Machine Learning for Audio
2025 September 8–10, London, UK

FLARE: An Open-Source Library for RIR Synthesis and Analysis in PyTorch

Gloria Dal Santo[1], Karolina Prawda[2], Sebastian J. Schlecht[3], and Vesa Välimäki[1]

[1]*Acoustics Lab, Dpt. of Information and Communications Engineering, Aalto University, Espoo, Finland*
[2]*AudioLab, School of Physics, Engineering, and Technology, University of York, York, United Kingdom*
[3]*Multimedia Communications and Signal Processing, Friedrich-Alexander-Universität (FAU), Erlangen, Germany*

Correspondence should be addressed to Gloria Dal Santo (`gloria.dalsanto@aalto.fi`)

ABSTRACT

We present FLARE, a PyTorch-based library for synthesizing and analyzing room impulse responses (RIRs) using feedback delay networks (FDNs). It provides a range of configurable FDN designs with parameters that can be inferred from reference data or sampled from a domain-aware distribution. FLARE is designed to support deep learning workflows by facilitating the efficient generation of diverse, metadata-rich synthetic RIR datasets.

1 Introduction

The FDN is an efficient recursive structure for synthesizing RIR [1]. In its simplest form, the FDN consists of N delay lines $\boldsymbol{m}$, a set of input and output gains $\boldsymbol{b}$ and $\boldsymbol{c}$, and a feedback matrix through which the delay outputs are coupled to the delay inputs. The transfer function of the FDN is

$$H(z) = \boldsymbol{c}^\top \left[\boldsymbol{D_m}(z)^{-1} - \boldsymbol{A}\right]^{-1} \boldsymbol{b} + d \tag{1}$$

where $\boldsymbol{A}$ is the $N \times N$ feedback matrix, $\boldsymbol{D_m}(z)$ is the diagonal delay matrix, and d is the direct gain. The operator $(\cdot)^\top$ denotes the transpose. A common design choice is to start with a lossless FDN prototype with a smooth response. This can be achieved by using an orthogonal matrix $\boldsymbol{A}$ [2] optimized for spectral flatness and temporal density [3, 4]. Attenuation filters can be combined with the feedback matrix to control frequency-dependent T_{60} [5], designed to give the same decay characteristics to all delays, and the echo density can be increased by introducing delays in the feedback matrix [6]. We denote the resulting filter matrix with $\boldsymbol{A}(z) = \boldsymbol{U}(z)\boldsymbol{\Gamma}(z)$ where $\boldsymbol{U}(z)$ is a paraunitary FIR filter matrix, and $\boldsymbol{\Gamma}(z)$ is the diagonal attenuation matrix.

FLAMO is a Python library for trainable differentiable digital signal processing (DDSP) modules based on frequency sampling and the PyTorch framework for automatic differentiation [7]. The library provides all the DDSP components, signal paths, and operations needed to implement differentiable FDNs.

Contribution — This paper introduces FLARE[1], an open-source PyTorch-based package for the synthesis and analysis of RIRs. The library implements various classes of FDNs derived from FLAMO's modules, with flexible parameterization options based on the user's input or sampled randomly from a domain-aware distribution. FLARE supports deep learning workflows by enabling efficient data generation for applications such as dereverberation, style transfer, and blind estimation. These tasks often require large volumes of RIR data, both recorded and synthetic. This work contributes to the field by expanding available resources through the generation of synthetic RIRs with detailed acoustic metadata.

2 Feedback Delay Network Designs

The reverb module in FLARE offers several common FDN designs listed below. The scalar coefficients of the FDNs can be optimized to reduce coloration, which is prominent for low N, and improve temporal density using the method in [4]. The attenuation $\Gamma(z)$ is designed from provided or randomly generated $T_{60}(\omega)$, and early reflections can be inserted by using an FIR filter in the direct path, in place of d. Each RIR is saved alongside the FDN parameters used to generate it and the derived acoustic parameters.

Scalar FM — The simplest form of the FDN follows (1), where $\boldsymbol{U}$ is a scalar orthogonal matrix. Common scalar orthogonal matrices include: random orthogonal, Hadamard, Householder, and rotation.

FIR FM — To increase echo density, $\boldsymbol{U}$ can be generalized as a FIR filter matrix $\boldsymbol{U}(z) = U_0 + z^{-1}U_1 + z^{-2}U_2 + \cdots + z^{-L}U_L$ with adjustable density [6]. The filters can be designed either as velvet noise sequences or as random sequences with arbitrary density.

Grouped FDN — To model coupled room spaces, $\boldsymbol{U}$ can be designed as a block matrix in which the diagonal sub-matrices control the mixing in the individual rooms, whereas the off-diagonal sub-matrices control the coupling between the rooms via a coupling coefficient [8]. The decay characteristics are controlled by groups of delay lines to model multi-stage decay [8].

3 Acoustic Parameters

The library includes a module for estimating acoustic parameters of RIRs, such as the energy decay curve, energy decay relief, normalized echo density profile, reverberation time (estimated via linear regression), clarity index, and definition. These are among the most commonly used acoustic parameters and can be computed directly from the generated RIRs, providing easy access when constructing a dataset with corresponding metadata.

4 Distribution

The source code is available online[1] and on the Python package index[2]. We plan to continuously expand the FLARE library to support more FDN designs.

References

[1] Jot, J.-M. and Chaigne, A., "Digital delay networks for designing artificial reverberators," in *AES Convention 90*, 1991.

[2] Stautner, J. and Puckette, M., "Designing multichannel reverberators," *Computer Music Journal*, 6(1), pp. 52–65, 1982.

[3] Dal Santo, G., Prawda, K., Schlecht, S. J., and Välimäki, V., "Differentiable Feedback Delay Network For Colorless Reverberation," in *Proc. of DAFx*, pp. 244–251, Copenhagen, Denmark, 2023.

[4] Dal Santo, G., Prawda, K., Schlecht, S. J., and Välimäki, V., "Optimizing tiny colorless feedback delay networks," *EURASIP Journal on ASMP*, 2025(1), p. 13, 2025.

[5] Välimäki, V., Prawda, K., and Schlecht, S., "Two-Stage Attenuation Filter for Artificial Reverberation," *IEEE Signal Process. Lett.*, 31, pp. 391–395, 2024.

[6] Schlecht, S. J. and Habets, E. A., "Scattering in feedback delay networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 28, pp. 1915–1924, 2020.

[7] Dal Santo, G., De Bortoli, G. M., Prawda, K., Schlecht, S. J., and Välimäki, V., "FLAMO: An Open-Source Library for Frequency-Domain Differentiable Audio Processing," in *Proc. of ICASSP*, pp. 1–5, IEEE, 2025.

[8] Das, O. and Abel, J. S., "Grouped feedback delay networks for modeling of coupled spaces," *Journal of the AES*, 69(7/8), pp. 486–496, 2021.

[1] `https://github.com/gdalsanto/flare`
[2] `https://pypi.org/project/flareverb`