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, Aalto University, Espoo, Finland

[2] AudioLab, University of York, York, United Kingdom

[3] Multimedia Communications and Signal Processing, FAU, Erlangen, Germany

AES Int. Conf. on AI and ML for Audio London, UK, September 8-10, 2025

github.com/gdalsanto/flare

```
pip install flareverb
```


Abstract

FLARE is an open-source PyTorch-based library for synthesizing and analyzing RIRs using Feedback Delay Networks (FDNs). It provides a range of configurable FDN designs derived from FLAMO's modules [1] 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 efficient generation of diverse, metadata-rich synthetic RIR datasets.

Applications: data augmentation for ML-based artificial reverberation, dereverberation, blind estimation, and characterization of acoustic environments.

Main Features

- Fast generation of metadata-rich synthetic RIRs
- Frequency-dependent attenuation based on a database of material properties [2]
- Gradient-based optimization to minimize coloration and enhance naturalness
- Modular architecture with flexible configuration

Feedback Delay Network

Support for standard SISO and MIMO FDN designs.

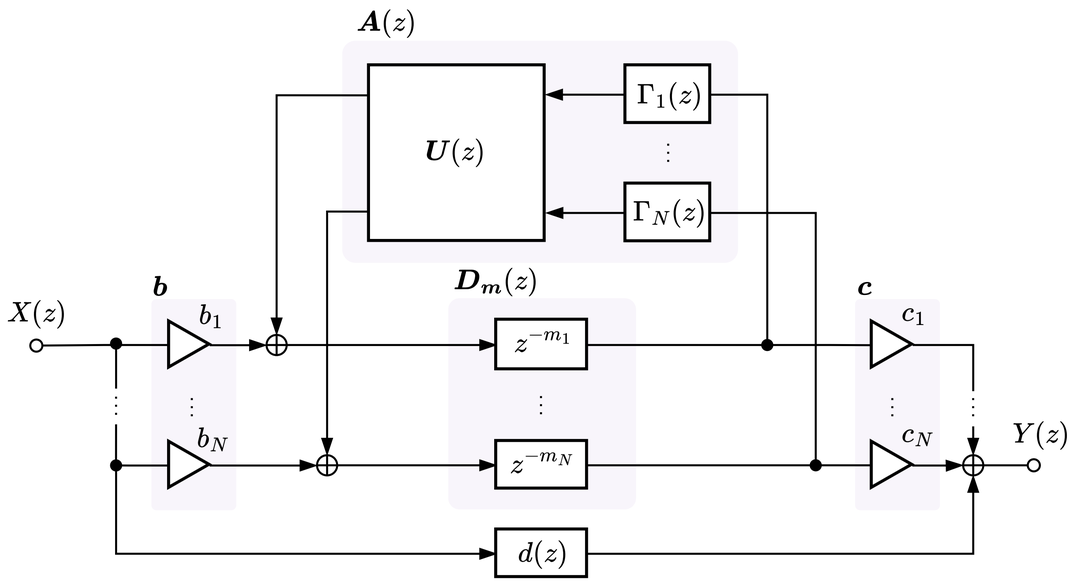


Figure 1. *Generalized structure of a SISO Feedback Delay Network*

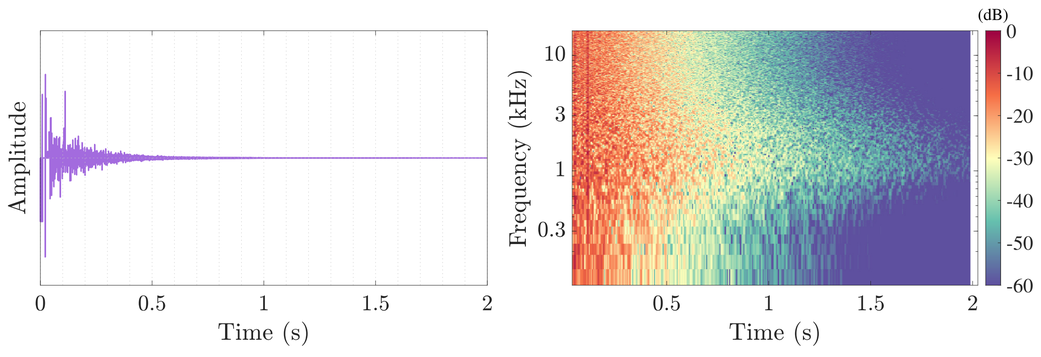
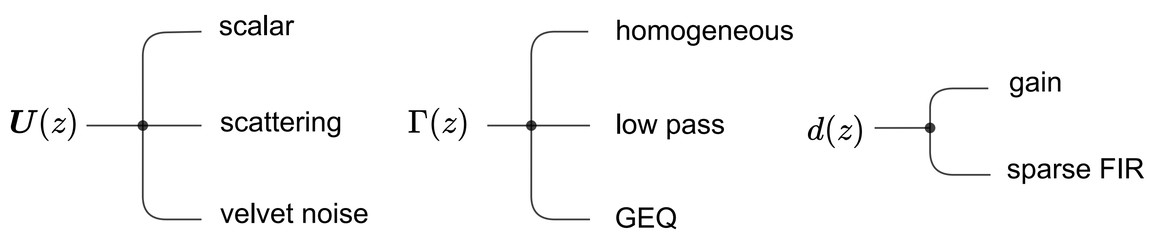


Figure 2. *Example of a generated RIR waveform (left) and corresponding spectrogram (right)*

Module configurations

The main FDN parameters can be sampled from specific designs depending on the required frequency accuracy, time density, and overall realism.



Grouped FDN for Coupled Spaces

Support for Grouped FDN to model the RIRs of coupled spaces. The interaction between the spaces is controlled by mixing and coupling angles [3].

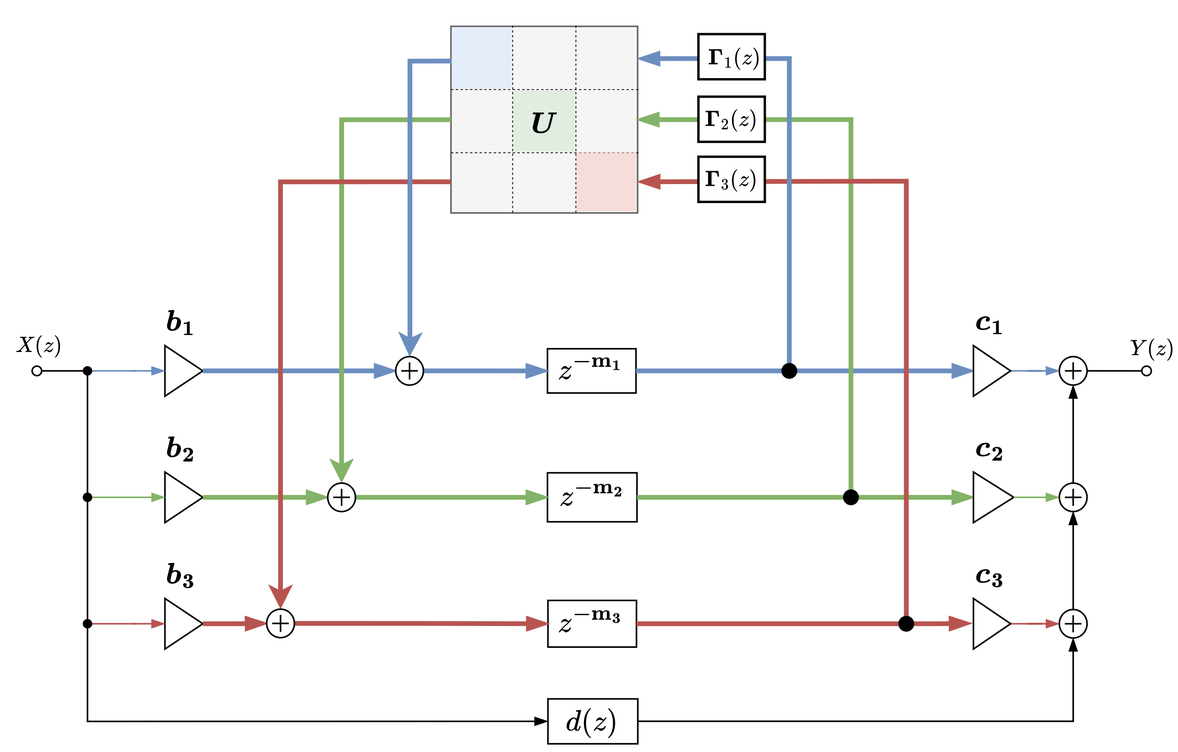


Figure 3. *Structure of a grouped FDN architecture. The bold lines represent multichannel signals, and the different colors represent different groups [3].*

Acoustic Parameters

The `AcousticAnalyzer` computes the most common acoustic parameters:

- Energy decay curves and energy decay relief
- Reverberation time
- Normalized echo density profile
- Clarity and definition

References

[1] G. Dal Santo, G. M. De Bortoli, K. Prawda, S. J. Schlecht, and V. Välimäki, "Flamo: An open-source library for frequency-domain differentiable audio processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[2] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[3] O. Das and J. S. Abel, "Grouped feedback delay networks for modeling of coupled spaces," *Journal of Audio Engineering Society*, vol. 69, no. 7/8, pp. 486–496, 2021.