EURASIP Journal on Audio, Speech, and Music Processing

EMPIRICAL RESEARCH Open Access

Data-driven room acoustic modeling via differentiable feedback delay networks with learnable delay lines

Alessandro Ilic Mezza[1*], Riccardo Giampiccolo[1], Enzo De Sena[2] and Alberto Bernardini[1]

Abstract

Over the past few decades, extensive research has been devoted to the design of artificial reverberation algorithms aimed at emulating the room acoustics of physical environments. Despite significant advancements, automatic parameter tuning of delay-network models remains an open challenge. We introduce a novel method for finding the parameters of a feedback delay network (FDN) such that its output renders target attributes of a measured room impulse response. The proposed approach involves the implementation of a differentiable FDN with trainable delay lines, which, for the first time, allows us to simultaneously learn each and every delay-network parameter via backpropagation. The iterative optimization process seeks to minimize a perceptually motivated time-domain loss function incorporating differentiable terms accounting for energy decay and echo density. Through experimental validation, we show that the proposed method yields time-invariant frequency-independent FDNs capable of closely matching the desired acoustical characteristics and outperforms existing methods based on genetic algorithms and analytical FDN design.

Keywords Automatic differentiation, Feedback delay networks, Room acoustics

1 Introduction

Room acoustic synthesis involves simulating the acoustic response of an environment, a task that finds application in a variety of fields, e.g., in music production, to artistically enhance sound recordings; in architectural acoustics, to improve the acoustics of performance spaces; or in VR/AR/computer games, to enhance listeners' sense of realism [1], immersion [2], and externalization [3].

Room acoustic models can be broadly classified in physical models, convolution models, and delay-network models [4]. Physical ones can be further divided in wave-based models, which provide high physical accuracy but at the cost of significant computational complexity, and geometrical-based ones, which make the simplifying approximation that sound travels like rays. Convolution models involve a set of stored room impulse responses (RIRs) and are therefore capable of replicating the true response of a real room [4]. Convolution is, however, an operation that despite recent advances [5] still carries a computational load that makes it ill-suited in certain real-time applications.

Delay-network models consist of recursively connected networks of delay lines and have a significantly lower computational cost than convolution. Rather than modeling the physical response of a specific room, delay-network models only aim to replicate certain perceptual aspects of room acoustics. These models have a long history, which can be traced back to the Schroeder reverberator [6]. Since then, a number of designs have been proposed, including feedback delay networks

*Correspondence:
Alessandro Ilic Mezza
alessandroilic.mezza@polimi.it
[1] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milan 20133, Italy
[2] Institute of Sound Recording, University of Surrey, Stag Hill, University Campus, Guildford GU27XH, UK

(FDNs) [7–9], scattering delay networks (SDNs) [10], and waveguide networks (WGNs) [11].

The parameters of delay-network models are typically designed to obtain certain desired acoustical characteristics, e.g., a target reverberation time. An alternative design paradigm is to fit the parameters such that the output is as close as possible to that of a measured RIR, hence combining the accuracy of convolution models with the low computational complexity of delay-network models. Several methods following this alternative design paradigm have been recently proposed for the case of FDNs, for instance using gradient-free methods [12–16] and gradient-based machine learning techniques [17, 18]. Existing approaches, however, involve a certain degree of human intervention and require heuristic-driven ad hoc choices for several model parameters.

This paper proposes a new method for automatic FDN parameter tuning. The present work is rooted in a recent framework for the parameter estimation of lumped-element models [19] based on automatic differentiation [20], and its novelty is twofold. First, the cost function combines two objective measures of perceptual features, i.e., the Energy Decay Curve (EDC) and a differentiable version of the normalized Echo Density Profile (EDP) [21]. Second, the delay line lengths are optimized via backpropagation along with every other FDN parameter, thus allowing exploiting the flexibility of delay-network models to the fullest. We thus introduce a simple, robust, and fully automatic method for matching acoustic measurements. The learned parameters can be then seamlessly plugged into off-the-shelf FDN software without further processing or mapping.

The paper is organized as follows. Section 2 introduces the background information on FDNs. Section 3 discusses the prior art on automatic FDN parameter tuning. Section 4 describes the proposed method, and Section 5 presents its evaluation. Finally, Section 6 concludes the manuscript.

2 Feedback delay networks

The block diagram of a single-input-single-output (SISO) FDN is shown in Fig. 1. This system is characterized by [22]

$$\begin{aligned} y[n] &= \mathbf{c}^T\mathbf{s}[n] + du[n] \\ \mathbf{s}[n+\boldsymbol{m}] &= \mathbf{A}\ \mathbf{s}[n] + \mathbf{b}u[n], \end{aligned} \tag{1}$$

where $u[n]$ is the input signal, $y[n]$ is the output signal, $\mathbf{b} \in \mathbb{R}^N$ is a vector of input gains, $\mathbf{c} \in \mathbb{R}^N$ is a vector of output gains, $(\cdot)^T$ denotes the transpose operation, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the feedback matrix, $d \in \mathbb{R}$ is the scalar gain associated to the direct path, and $\boldsymbol{m} = [m_1, ..., m_N]$ is a vector containing the length of the N delay lines expressed in samples. The vector $\mathbf{s}[n] \in \mathbb{R}^N$ denotes the output of the delay lines at time index n, and we use the following notation $\mathbf{s}[n+\boldsymbol{m}] = [s_1[n+m_1], ..., s_N[n+m_N]]^T$ to indicate N parallel delay operations of $m_1, ..., m_N$ samples, respectively, applied to $\mathbf{s}[n]$.

If $\boldsymbol{m} = [1, ..., 1]$, then (1) corresponds to the measurement and state equations of a state-space model. In other words, an FDN corresponds to a generalized version of a state-space model with non-unit delays [22].

The standard approach to designing the FDN parameters involves choosing the feedback matrix, delays, and input/output weights so as to obtain certain desired acoustic characteristics—usually a sufficient echo density and a pre-set reverberation time. The most important parameters are the ones associated to the recursive loop, i.e., $\boldsymbol{m}$ and $\mathbf{A}$, since they determine the energy decay behavior of the model, as well as its stability. The delays, $\boldsymbol{m}$, are typically chosen as co-prime of each other, so as to reduce the number of overlapping echoes and increase the echo density [23]. The design of the feedback matrix,

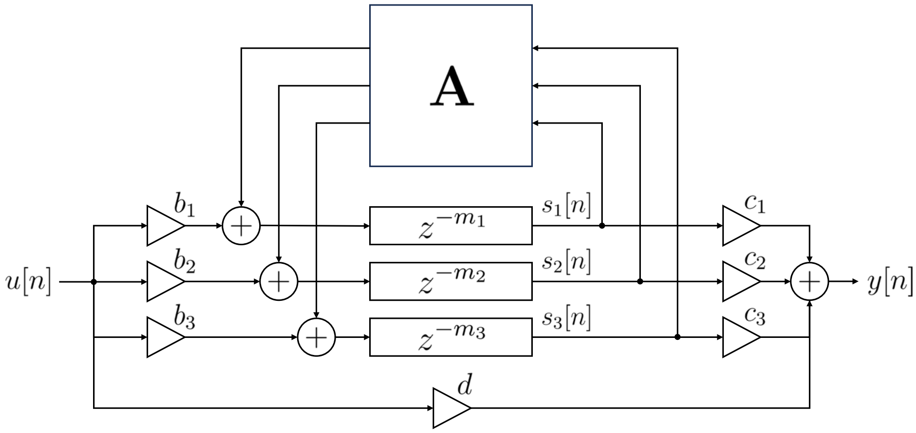


Fig. 1 Block diagram of a SISO FDN with $N = 3$

A, starts from a lossless prototype, usually an orthogonal matrix such as Hadamard or Householder matrix, which have been shown to ensure (critical) stability regardless of the delays, a property defined by Schlecht and Habets as *unilosslessness* [8]. Losses are then incorporated by multiplying the unilossless matrix by a diagonal matrix of scalars designed to achieve a pre-set reverberation time, T_{60}.

While it is possible to design feedback matrices as time varying and/or frequency dependent [7, 24], this paper focuses on the time-invariant and frequency-independent case. With this assumption, the stability of the system can be easily enforced throughout the training, thanks to the model reparameterization strategies discussed later in Section 4. Moreover, time-invariant frequency-independent FDNs benefit from having a low computational complexity. From visual inspection of Fig. 1, indeed, such an FDN only requires $2N + 1$ multiplications, $2N$ additions and one vector-matrix multiplication per sample. The vector-matrix multiplication requires N^2 scalar multiplications and $N(N - 1)$ additions for the case of a generic feedback matrix (while it becomes $O(N)$ for a Householder matrix). Assuming equal cost of additions and multiplications, the overall computational complexity of an FDN amounts to $f_s\,(2N^2 + 3N + 1)$ floating-point operations per second (FLOPS), where f_s is the sampling rate. For $N = 6$ and $f_s = 44.1$ kHz, that corresponds to a computational complexity of 4 MFLOPS. For comparison, modeling a 0.5 s long RIR using an FIR filter (i.e., naive convolution) at the same sampling rate would carry a computation complexity of 1945 MFLOPS. In real-time applications, one would normally use faster methods such as partitioned convolution [5] or overlap-add (FFT-based) convolution [25]. Under the same conditions and assuming a frame refresh rate of 50 Hz, overlap-add convolution carries a complexity of 207 MFLOPS [10], which is still nearly two orders of magnitude larger than an FDN.

3 Related work

As mentioned earlier, the automatic tuning of FDN parameters has been previously investigated by means of gradient-free methods, such as Bayesian optimization [12] and genetic algorithms [13–16], as well as gradient-based machine learning techniques [17, 18].

Some works are concerned with the automatic tuning of off-the-shelf reverberation plug-ins. In [26], Heise and colleagues investigate four gradient-free optimization strategies: simulated evolution [27], the Nelder-Mead simplex method [28], Nelder-Mead with brute-force parallelization, and particle swarm optimization [29]. More recently, [12] applies Bayesian optimization using a Gaussian process as a prior to iteratively acquire the control parameters of an external FDN plug-in that minimize the mean absolute error between the multiresolution mel-spectrogram of the target RIR convolved with a 3 s logarithmic sine sweep and that of the artificial reverberator output. The FDN control parameters include the delay line length, reverberation time, fade-in time, high/low cutoff, high/low Q, high/low gain, and dry-wet ratio.

Conversely, other studies assume to have white-box access to the delay-network structure and apply genetic algorithms (GA) to optimize a subset of the FDN parameters. In [13], a GA is used to find both the N^2 coefficients of the feedback matrix **A** and N cutoff frequencies of lowpass filters, one for each delay line. The authors of [14] aim at finding a mapping between room and FDN parameters for VR/AR applications. To this end, they synthesize the binaural RIRs of a set of virtual shoebox rooms, apply a GA to tune the FDN's delay lines and scalar feedback gain, and use the resulting training pairs to fit a support vector machine (SVM) regressor. In [15], Coggin and Pirkle apply a GA for the estimation of $\boldsymbol{m}$, **b**, and **c**. For every individual in a generation, attenuation and output filters are designed using the Yule-Walker method. The authors investigate several fitness functions before favoring the Chebyshev distance between the power envelopes of the target and predicted IR. The optimization is run for late reverberation only: the first 85 ms of the RIR are cut and convolved with the input signal, before being fed to the FDN to model late reverberation. Following [15], Ibnyahya and Reiss recently introduced a multi-stage method [16] combining more advanced analytical filter design methods and GAs to estimate the FDN parameters that would best approximate a target RIR in terms of an MFCC-based fitness function similar to the cost function used in [26].

Due to the well-known limitations of genetic algorithms, such as the high risk of finding sub-optimal solutions, overall slow convergence rate, and the challenges of striking a good exploration-exploitation balance [30], gradient-based techniques have been recently proposed.

Inspired by groundbreaking research on differentiable digital signal processing [31], Lee et al. [17] let the gradients of a multiresolution spectral loss flow through a differentiable artificial reverberator so that they may reach a trainable neural network tasked with yielding the reverberator parameters. This way, the authors train a convolutional-recurrent neural network tasked with inferring the input, output, and absorption filters of a FDN from a reference reverberation (RIR or speech). It is worth mentioning, however, that it is not the delay-network parameters those that are optimized via stochastic gradient descent, but rather it is the weights of the neural network serving as black-box parameter estimator. As such, the differentiable FDN is effectively used as a processing block in computing the loss of an end-to-end neural network instead of being the target of the optimization process.

In a different vein, several recent works aim at learning lumped parameters via gradient-based optimization directly within the digital structure of the model and forgo parameter-yielding neural networks altogether. In this respect, automatic differentiation has been recently proposed to find $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$ of an FDN (without parameterizing them as a neural network) so as to minimize spectral coloration and obtain a flat frequency response [18]. Similar yet distinct, other works adopt a white-box system identification approach and use backpropagation to find the parameters of predetermined mathematical models so as to match measured data as closely as possible [19, 32, 33].

In this work, we adopt the latter approach and use the method detailed in the next section to find the values of $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$, $\boldsymbol{m}$, and d such that the resulting FDN is capable of modeling perceptually meaningful characteristics of the acoustic response of real-life environments.

4 Proposed method

The proposed method involves an iterative gradient-based optimization algorithm. As a learning objective, we choose a perceptually informed loss function, $\mathcal{L}(h, \hat{h})$, between a target RIR, $h[n]$, and the time-domain FDN output, $\hat{h}[n]$, obtained by setting the FDN input to the Kronecker delta, i.e., $u[n] = \delta[n]$.

We initialize the FDN parameters with no prior knowledge of $h[n]$. Then, at the beginning of each iteration, we calculate $\hat{h}[n]$ by evaluating (1) while freezing the current parameter estimates. Thus, we evaluate $\mathcal{L}(h, \hat{h})$. Finally, each trainable FDN parameter θ undergoes an optimization step using the error-free gradient $\nabla_\theta \mathcal{L}$ computed via reverse-mode automatic differentiation [20].

A typical approach is to use a delay network to only model the late reverberation while handling early reflections separately [12, 14–16]. Instead, we optimize the FDN such that it accounts for both early and late reverberation at the same time, exploiting thus the advantages of synthesizing the entire RIR with an efficient recursive structure.

At training time, we strip out the initial silence due to direct-path propagation and disregard every sample beyond the T_{60} of the target RIR. In other words, we only consider the first $L_{T_{60}} := \lceil T_{60} \cdot f_s \rceil$ samples of $h[n]$ and $\hat{h}[n]$ in computing the loss. The reason behind restricting the temporal scope only to the segment of the RIR associated with the T_{60} is that, beyond this point, the values involved in the ensuing computations become so small that numerical errors might occur when using single-precision floating-point numbers, and the training process would unwantedly focus on the statistics of background/numerical noise. Notice that, at inference time, i.e., once the FDN parameters have been learned, the room acoustics simulation can be run indefinitely at a very low computational cost.

In this work, we optimize the input gains $\mathbf{b} \in \mathbb{R}^N_{\geq 0}$, the output gains $\mathbf{c} \in \mathbb{R}^N_{\geq 0}$, the direct gain $d \in \mathbb{R}_{\geq 0}$, the feedback matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, and the delays $\boldsymbol{m} \in \mathbb{R}^N_{\geq 0}$ expressed in fractional samples.

4.1 Model reparameterization

Let θ be a scalar parameter of the FDN such that $\theta \in \mathbb{X}$ where $\mathbb{X} \subseteq \mathbb{R}$. In general, instead of learning θ directly, we learn an unconstrained proxy $\tilde{\theta} \in \mathbb{R}$ that maps onto θ through a differentiable (and possibly nonlinear) function $f : \mathbb{R} \to \mathbb{X}$. Hence, we can use $f(\tilde{\theta})$ in place of θ in any computation involved in the forward pass of the FDN. In case of vector-valued parameters $\boldsymbol{\theta} \in \mathbb{X}^N$, we apply f in an element-wise fashion, i.e., $\boldsymbol{\theta} := [f(\tilde{\theta}_1), ..., f(\tilde{\theta}_N)]^T$.

The reason behind such an explicit reparameterization method is that, while we would like $\boldsymbol{\theta}$ to take values in $\mathbb{X}^N$ at every iteration, gradient-based optimization may yield parameters that do not respect such a constraint, even when using implicit regularization strategies, e.g., by means of auxiliary loss functions and regularizers.

In our FDN model, we treat every parameter in a different fashion. We discuss gain reparameterization in Section 4.2 and present the feedback matrix reparameterization in Section 4.3. Finally, we outline the implementation of the differentiable delay lines and their reparameterization in Section 4.4.

Figure 2 summarizes the proposed approach, listing all the unconstrained trainable parameters (top), the corresponding reparameterization (middle), and illustrating once again how, through (1), the FDN processes time-domain signals, including unit impulses $\delta[n]$ (bottom).

4.2 Trainable gains

We would like the input, output, and direct gains of our differentiable FDN to be nonnegative. This way, the gains only affect the amplitude of the signals and do not risk inverting their polarity. Instead, we let $\mathbf{A}$ model phase-reversing reflections. To enforce gain nonnegativity, we employ a differentiable nonlinear function $f_{\geq 0} : \mathbb{R} \to \mathbb{R}_{\geq 0}$, such as the Softplus or exponential function. We then learn, e.g., $\tilde{\mathbf{b}} = [\tilde{b}_1, ..., \tilde{b}_N]^T$ while using $\mathbf{b} = [f_{\geq 0}(\tilde{b}_1), ..., f_{\geq 0}(\tilde{b}_N)]^T$ in every computation concerning the FDN. Among other options, we select $f_{\geq 0}(x) = |x|$ [19], where the requirement of $f_{\geq 0}$ being differentiable everywhere was relaxed as it is common for many widely adopted activation functions, such as ReLU.

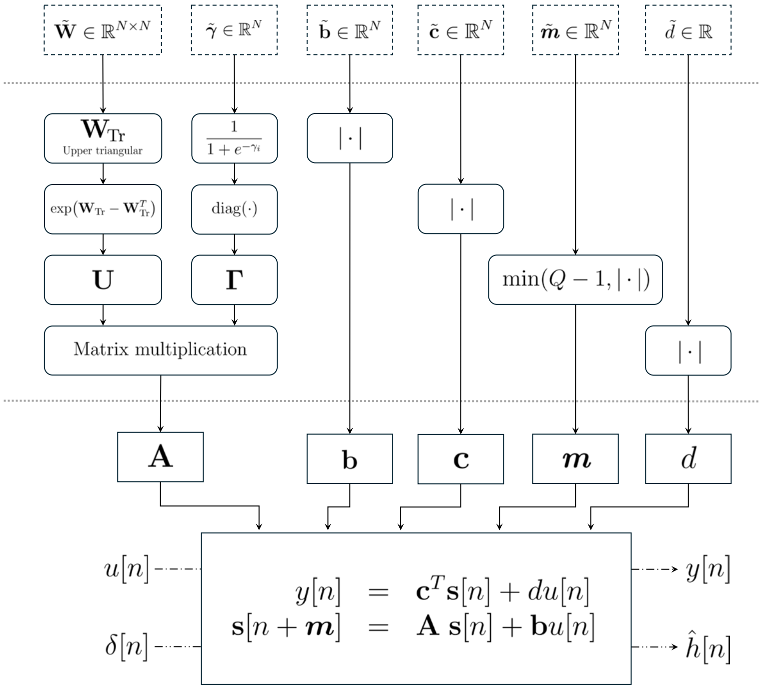

$\tilde{\mathbf{W}} \in \mathbb{R}^{N\times N}$
$\tilde{\gamma} \in \mathbb{R}^{N}$
$\tilde{\mathbf{b}} \in \mathbb{R}^{N}$
$\tilde{\mathbf{c}} \in \mathbb{R}^{N}$
$\tilde{\boldsymbol{m}} \in \mathbb{R}^{N}$
$\tilde{d} \in \mathbb{R}$
$\mathbf{W}_{\mathrm{Tr}}$
Upper triangular
$\frac{1}{1+e^{-\gamma_i}}$
$|\cdot|$
$\exp(\mathbf{W}_{\mathrm{Tr}} - \mathbf{W}_{\mathrm{Tr}}^{T})$
$\mathrm{diag}(\cdot)$
$|\cdot|$
$\mathbf{U}$
$\mathbf{\Gamma}$
$\min(Q-1, |\cdot|)$
$|\cdot|$
Matrix multiplication
$\mathbf{A}$
$\mathbf{b}$
$\mathbf{c}$
$\boldsymbol{m}$
d
$u[n]$
$\delta[n]$
$y[n] = \mathbf{c}^T\mathbf{s}[n] + du[n]$
$\mathbf{s}[n+\boldsymbol{m}] = \mathbf{A}\,\mathbf{s}[n] + \mathbf{b}u[n]$
$y[n]$
$\hat{h}[n]$

Fig. 2 Summary of the proposed method

4.3 Trainable feedback matrix

We focus on lossy FDNs. In prior work [18], frequency-independent homogeneous decay has been modeled by parameterizing $\mathbf{A}$ as the product of a unilossless matrix $\mathbf{U}$ and a diagonal matrix $\mathbf{\Gamma}(\boldsymbol{m}) = \mathrm{diag}(\gamma_1, ..., \gamma_N) = \mathrm{diag}(\gamma^{m_1}, ..., \gamma^{m_N})$ containing a delay-dependent absorption coefficient for each delay line, where $\gamma \in (0, 1)$ is a constant gain-per-sample parameter. The feedback matrix is thus expressed as

$$\mathbf{A} = \mathbf{U}\mathbf{\Gamma}(\boldsymbol{m}). \tag{2}$$

We let $\mathbf{U}$ be an orthogonal matrix, satisfying the unitary condition for unilosslessness [8]. To ensure this property, $\mathbf{U}$ is further parameterized by means of $\tilde{\mathbf{W}} \in \mathbb{R}^{N\times N}$ that, at each iteration, yields [18]

$$\mathbf{U} = \exp\left(\mathbf{W}_{\mathrm{Tr}} - \mathbf{W}_{\mathrm{Tr}}^{T}\right), \tag{3}$$

where $\mathbf{W}_{\mathrm{Tr}}$ is the upper triangular part of $\tilde{\mathbf{W}}$, and $\exp(\cdot)$ is the matrix exponential. In other words, instead of trying to directly learn a unilossless matrix, we learn an unconstrained real-valued matrix $\tilde{\mathbf{W}}$ that maps onto an orthogonal matrix through the exponential mapping in (3).[1] In particular, $\mathbf{U}$ is ensured to be orthogonal because $\mathbf{W}_{\mathrm{Tr}} - \mathbf{W}_{\mathrm{Tr}}^{T}$ is skew-symmetric [34].

As for the matrix $\mathbf{\Gamma}(\boldsymbol{m})$, we noticed that tying the values of the absorption coefficients $\gamma_1, ..., \gamma_N$ to those of the fractional delays $m_1, ..., m_N$ as previously done in [18] led to instability during training since the values in $\boldsymbol{m}$ were concurrently acting on the temporal location of the IR taps as well as their amplitude.[2] Conversely, we decouple $\mathbf{\Gamma}$ from $\boldsymbol{m}$, thus learning a possibly inhomogeneous FDN characterized by $\mathbf{A} = \mathbf{U}\mathbf{\Gamma}$, as opposed to the homogeneous FDNs studied in [18].

In learning the unconstrained absorption matrix $\tilde{\mathbf{\Gamma}} = \mathrm{diag}(\tilde{\boldsymbol{\gamma}}) = \mathrm{diag}(\tilde{\gamma}_1, ..., \tilde{\gamma}_N)$, we define $f_{(0,1)} : \mathbb{R} \to (0, 1)$ and optimize $\tilde{\boldsymbol{\gamma}} \in \mathbb{R}^N$ so that $\mathbf{\Gamma} = \mathrm{diag}\big(f_{(0,1)}(\tilde{\gamma}_1), ..., f_{(0,1)}(\tilde{\gamma}_N)\big)$. In the following, we use the well-known Sigmoid function to force the absorption coefficients to take values in the range of 0 to 1, i.e.,

$$f_{(0,1)}(x) = \frac{1}{1+e^{-x}}. \tag{4}$$

4.4 Trainable delay lines

In the digital domain, an integer delay can be efficiently implemented as a reading operation from a buffer that

[1] It is worth noting that, although $\tilde{\mathbf{W}}$ is a $N \times N$ matrix, only the $N(N-1)/2$ upper triangular entries are actually learned and used in downstream computations.

[2] This problem is unique to our approach, as previous studies employed non-trainable delay lines with fixed lengths [18].

accumulates past samples. Unfortunately, this approach is not differentiable. Instead, since the Fourier transform is a linear and differentiable operator, we opt to work in the frequency domain to circumvent the problem.

In [35], Pei and Lai proposed a closed-form variable fractional delay filter, which turns out to be inherently differentiable. In our FDN implementation, each delay line is equipped with a Q-sample buffer, so that the ith buffer stores the signal $x_i[n]$. First, we zero-pad $x_i[n]$ to reduce artifacts due to the ensuing circular convolution. Then, we compute the K-point fast Fourier transform (FFT) of the resulting signal, with $K = 2Q$. Following [35], we apply a delay of m_i (fractional) samples by multiplying the discrete spectrum with the conjugate symmetric frequency response $D_i[k]$ defined in (6). Finally, we go back in the time domain by computing the inverse FFT. We can express this sequence of differentiable operations as

$$x_i[n - m_i] = \mathrm{IFFT}\big\{D_i[k] \cdot \mathrm{FFT}\{x_i[n]\}\big\}, \tag{5}$$

where

$$D_i[k] = \begin{cases} 1, & k = 0 \\ e^{-jm_i(2\pi/K)k}, & k = 1, ..., \frac{K}{2} - 1 \\ \cos(m_i\pi), & k = \frac{K}{2} \\ e^{-jm_i(2\pi/K)(K-k)}, & k = \frac{K}{2} + 1, ..., K - 1 \end{cases} \tag{6}$$

which, in the time domain, corresponds to a windowed-sinc finite impulse response [35].

Delays $m_1, ..., m_N$ must be nonnegative to realize a casual system. Hence, we use $f_{\geq 0}$ to reparameterize them. Moreover, we clip the resulting values so not to exceed the given buffer length. This yields[3]

$$m_i = \min\big(Q - 1, f_{\geq 0}(\tilde{m}_i)\big), \tag{7}$$

where $\tilde{m}_i \in \mathbb{R}$ is the ith trainable delay-line length proxy, $i = 1, ..., N$.

Finally, it is worth highlighting three main reasons for labeling our FDN model as *time-domain*, despite implementing the differentiable delay lines in the frequency domain. First, we stress that our FDN yields the output one sample at a time according to (1). Second, frequency-domain operations are confined within the delay filterbank. Since delay lines being differentiable is only required at training time, the inference model can thus feature a different fractional delay implementation, possibly in the time domain. Third, we emphasize the difference between our approach and existing methods implementing every FDN operation in the frequency domain [17, 18].

4.5 Loss function

Our goal is to learn an FDN capable of capturing perceptual qualities of a target room. Hence, we avoid pointwise regression objectives such as L^p-losses between IR taps. Instead, we set out to minimize an error function ($\mathcal{L}_{\mathrm{EDC}}$) between the true and predicted EDCs. Additionally, we use a novel regularization loss ($\mathcal{L}_{\mathrm{EDP}}$) aimed at matching the echo distribution of the target RIR by acting on the normalized EDP. Namely, the composite loss function can be written as

$$\mathcal{L} = \mathcal{L}_{\mathrm{EDC}} + \lambda \mathcal{L}_{\mathrm{EDP}}, \tag{8}$$

where $\lambda \in \mathbb{R}_{\geq 0}$. Similarly to [19], the loss is evaluated in the time domain, and, at each iteration, requires a forward pass through the discrete-time model defined by the current parameter estimates. In the following sections, we analyze each of the terms in (8).

4.5.1 *Energy Decay Curve loss*

For a discrete-time RIR of length L, the EDC can be computed through Schroeder's backward integration [36]

$$\varepsilon[n] = \sum_{\tau=n}^{L} h^2[\tau]. \tag{9}$$

Since (9) is differentiable, we can train the FDN to minimize a normalized mean squared error (NMSE) loss defined on the EDCs, i.e.,

$$\mathcal{L}_{\mathrm{EDC}} = \frac{\sum_n \big(\varepsilon[n] - \hat{\varepsilon}[n]\big)^2}{\sum_n \varepsilon[n]^2}, \tag{10}$$

where $\hat{\varepsilon}[n] = \sum_{\tau=n}^{L} \hat{h}^2[\tau]$.

It is worth noting that, whereas the EDC is typically expressed in dB, (10) is evaluated on a linear scale. The idea here is that a linear loss emphasizes errors in the early portion of $\varepsilon[n]$, i.e., where discrepancies are perceptually more relevant [37], compared to a logarithmic loss that would put more focus on the reverberation tail.

4.5.2 *Differentiable normalized Echo Density Profile*

In [21], Abel and Huang introduced the so-called normalized Echo Density Profile (EDP) as a means to quantify reverberation echo density by analyzing consecutive frames of the reverberation impulse response. The EDP indicates the proportion of IR taps that fall above the local standard deviation. The resulting profile is normalized to a scale ranging from nearly zero, indicating a minimal presence of echoes, to around one, denoting a fully dense reverberation with Gaussian statistics [38].

[3] It is worth pointing out that (7) is not the only way to account for the implicit periodization of the buffered signal when computing the FFT. For instance, an alternative parameterization is $m_i = (Q - 1) \cdot f_{(0,1)}(\tilde{m}_i)$, which ensures that $m_i \in (0, Q - 1)$ at all times.

The EDP is defined as [21]

$$\eta[n] = \frac{1}{\mathrm{erfc}(1/\sqrt{2})} \sum_{\tau=n-\nu}^{n+\nu} w[\tau] \mathbb{1}\big\{|h[\tau]| > \sigma_n\big\}, \quad (11)$$

where erfc(·) is the complementary error function,

$$\sigma_n = \sqrt{\sum_{\tau=n-\nu}^{n+\nu} w[\tau] h^2[\tau]}, \quad (12)$$

is the standard deviation of the nth frame, $w[n]$ is a window function of length $2\nu + 1$ samples (usually 20 ms) such that $\sum_\tau w[\tau] = 1$, and $\mathbb{1}\{\cdot\}$ is an indicator function

$$\mathbb{1}\big\{|h[\tau]| > \sigma\big\} = \begin{cases} 1 & |h[\tau]| > \sigma, \\ 0 & |h[\tau]| \leq \sigma. \end{cases} \quad (13)$$

Notably, $\mathbb{1}\{\cdot\}$ is non-differentiable. Thus, the EDP cannot be utilized within our automatic differentiation framework.

To overcome this problem, this section introduces a novel differentiable EDP approximation, which we call *Soft Echo Density Profile*.

First, we notice that the indicator function $\mathbb{1}\big\{|h[\tau]| > \sigma\big\}$ can be equivalently expressed as a Heaviside step function $\mathcal{H}(|h[\tau]| - \sigma)$. Then, we let $g(x)$ denote the Sigmoid function. We define the *scaled Sigmoid* function $g_\kappa(x) = g(\kappa x)$, where $\kappa \in \mathbb{R}_{>0}$. Since

$$\lim_{\kappa \to \infty} g_\kappa(x) = \mathcal{H}(x), \quad (14)$$

we can define the Soft EDP function as

$$\eta_\kappa[n] = \frac{1}{\mathrm{erfc}(1/\sqrt{2})} \sum_{\tau=n-\nu}^{n+\nu} w[\tau] g_\kappa(|h[\tau]| - \sigma_n), \quad (15)$$

which approximates (11) for $\kappa \gg 1$.

It is worth mentioning that, whilst the EDP approximation improves as κ becomes larger, this also has the side effect of increasing the risk of vanishing gradients. In fact, the derivative of the scaled Sigmoid function can be written as

$$g'_\kappa(x) = g(\kappa x)\big(1 - g(\kappa x)\big), \quad (16)$$

which approaches zero for large or small inputs. Hence, $g'_\kappa(x)$ takes on near-zero values outside of a neighborhood of $x = 0$ whose size is inversely proportional to κ, which, in turn, may impede the gradient flow for $\kappa \gg 1$.

In practice, we would like to choose a large value of κ but not larger than what is needed. Notably, the need for a large scaling factor is not constant throughout the temporal evolution of a RIR. Early taps are typically sparse, and $(|h[\tau]| - \sigma_n)$ tends to fall within the saturating region of $g_\kappa(\cdot)$, even for lower values of κ. Conversely, in later portions of the RIR, κ must take on very large values to contrast the fact that the amplitude of $(|h[\tau]| - \sigma_n)$ progressively decreases. For this reason, we introduce a time-varying scaling parameter, $\kappa_n = \xi n + \varrho$, where $\xi \in \mathbb{R}_{>0}$ and $\varrho \in \mathbb{R}_{\geq 0}$ are hyperparameters. Progressively increasing the scaling coefficient has the benefit of enhancing the gradient flow for the early reflections, while improving the EDP approximation for late reverberation. In general, a more principled definition for κ_n could be devised, e.g., by tying it to the local statistics of the target RIR or its energy decay. In this work, however, we favor a simple and reproducible approach as it proved to work well in practice.

4.5.3 Soft EDP loss

Despite the trade-off between vanishing gradients and goodness of fit discussed in the previous section, every operation involved in the computation of (15) is differentiable almost everywhere. This allows us to use the following EDP loss term as a regularizer during the FDN training

$$\mathcal{L}_{\mathrm{EDP}} = \frac{1}{L_{T_{60}}} \sum_n \big(\eta_\kappa[n] - \hat{\eta}_\kappa[n]\big)^2, \quad (17)$$

where $\hat{\eta}_\kappa[n]$ is the Soft EDP of the predicted RIR, and $L_{T_{60}} = \lceil T_{60} \cdot f_{\mathrm{s}} \rceil$.

5 Evaluation

We evaluate the proposed method using real-world measured RIRs from the 2016 MIT Acoustical Reverberation Scene Statistics Survey [39]. The MIT corpus contains single-channel environmental IRs of both open and closed spaces. Of the 271 IRs, we select three according to their reverberation time, which, across the dataset, ranges from a minimum of 0.06 s to a maximum of 1.99 s. We select three indoor environments:[4] (i) a small room ($T_{60} \approx 0.2$ s), (ii) a medium room ($T_{60} \approx 0.6$ s), and (iii) a larger room ($T_{60} \approx 1.2$ s). For reproducibility, the ID of the chosen RIRs is reported: (i) `h214_Pizzeria_1txts`, (ii) `h270_Hallway_House_1txts`, and (iii) `h052_Gym_WeightRoom_3txts`. The full IR Survey dataset is

[4] Although the proposed parameter tuning method shares some similarities with neural network training, particularly in their use of backpropagation, differentiable FDNs require a dedicated optimization routine for each target RIR. When it comes to evaluation, this study thus focuses on a limited number of illustrative examples; this approach is consistent with white-box system identification literature while contrasting with the way deep learning models are typically evaluated, which, instead, involves large-scale training and test sets.

available online.[5] For the evaluation, all IRs are resampled to 16 kHz and scaled to unit norm.

5.1 Baseline methods

From the overview presented in Section 3, it appears that no method in the literature is directly comparable with ours. In fact, existing automatic tuning approaches either focus on off-the-shelf reverb plug-ins [12, 26], limit the set of target parameters to just a few [13, 14], or augment the FDN topology with auxiliary frequency-dependent components [15, 16]. To the best of our knowledge, there is no state-of-the-art method addressing the simultaneous estimation of every parameter of a time-invariant frequency-independent FDN in a purely data-driven fashion.

That being said, with the aim of comparing our approach with existing techniques, we implement three baseline methods.

The first is based on a classic method for homogeneous reverberation time control (HRTC) and involves choosing all FDN parameters but the absorption coefficients heuristically. For simplicity, we refer to this method as "HRTC baseline."

The second method, which we call "Colorless baseline," also relies on HRTC to control the decay rate. However, contrary to the HRTC baseline, all remaining FDN parameters are optimized following the approach detailed in [18] as so to achieve a maximally flat frequency response.

The third and final baseline, inspired by [16], makes use of a genetic algorithm (GA) to optimize every FDN parameter except for the feedback matrix. We call this method "GA baseline."

In the following sections, we detail the three baseline methods one at a time.

5.1.1 HRTC baseline

Given the FDN model shown in Fig. 1, a classic method to introduce homogeneous loss in an otherwise lossless prototype is to replace each unit delay z^{-1} with a lossy delay element γz^{-1}, where γ is thought of as a gain-per-sample coefficient [7, 18, 40].

In practice, the loss of each delay line is lumped into a single attenuation term proportional to its length. We can thus define $\mathbf{\Gamma}(\boldsymbol{m}) = \mathrm{diag}(\gamma^{m_1}, \dots, \gamma^{m_N})$ as discussed in Section 4.3, where γ controls the decay rate according to the desired reverberation time. Namely, γ should satisfy [7]

$$20 \log_{10} \gamma = \frac{-60}{f_s T_{60}}, \tag{18}$$

where T_{60} is estimated from the target RIR.

Whereas the absorption coefficients are given by (18), every other parameter in the HRTC baseline are determined by means of heuristics. We parameterize the feedback matrix as in (2), where $\mathbf{U}$ is a random orthogonal matrix. To ensure that the HRTC baseline is most comparable with the proposed method (Section 4.3) and the Colorless baseline (Section 5.1.2), we obtain $\mathbf{U}$ through the exponential mapping in (3). The scalar gain d is set equal to the amplitude of the target RIR at the time index associated with the direct path. We use unity input gains, i.e., $\mathbf{b} = \mathbf{1}_N$, where $\mathbf{1}_N$ is a vector of $N = 6$ ones. The output gains are chosen so that $\mathbf{c} = \frac{1}{N}\mathbf{1}_N$. As such, the dot product $\mathbf{c}^T\mathbf{s}[n]$ in (1) is equivalent to the arithmetic average of the outputs of the N delay lines at time n. Finally, the delays $\boldsymbol{m} = [997, 1153, 1327, 1559, 1801, 2099]$ consist of logarithmically distributed prime numbers from Delay Set #1 in [18], and the corresponding non-differentiable integer delay lines are implemented via buffer readout.

5.1.2 Colorless baseline

In the previous section, we discussed a baseline method consisting of a homogeneous FDN where the reverberation time is controlled by choosing γ according to (18), with the other parameters being manually selected. Here, we present an alternative baseline method that foregoes some of these arbitrary choices in favor of an optimization approach.

In [18], the authors implement a differentiable homogeneous FDN in the frequency domain and find $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$ via gradient descent so as to minimize spectral coloration.

Colorless reverberation [41] is here defined as the acoustic quality of an artificial reverberation algorithm whose frequency response is flat, i.e., constant at all frequencies.

To achieve this, $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$ are iteratively updated via backpropagation using Adam [42] to minimize a reference-free[6] loss function comprising two terms [18]. The first term encourages the magnitude of the sampled transfer function of each delay network channel to be as close to one as possible. The second term penalizes IR sparsity in the time domain and avoids trivial solutions.

The delays $\boldsymbol{m}$ are kept constant, and $\mathbf{A}$ is parameterized through (2) and (3). Like in the HRTC baseline, we

[5] [Online] IR Survey dataset: https://mcdermottlab.mit.edu/Reverb/IR_Survey.html

[6] With *reference-free*, we emphasize that, unlike our method, the loss function in [18] is computed solely on the FDN response and does not consider a reference RIR as the target of the optimization process.

use the lengths in samples comprised in Delay Set #1 from [18], and γ is set according to (18).

With this baseline, our goal is to test whether an FDN optimized to obtain a flat magnitude response brings about any benefit when it comes to modeling the energy decay and echo density of a target RIR. It is worth noting, however, that the learning objective in [18] is not concerned with matching the behavior of a reference RIR. As such, the resulting **A**, **b**, and **c** might prove suboptimal for what concerns reproducing the reverberation of the target space.

5.1.3 GA baseline

In [16], Ibnyahya and Reiss proposed a multi-stage automatic tuning approach that combines genetic optimization [43] and analytical filter design [44]. Adhering to well-established design principles [7], the prototype FDN considered in [16] is equipped with attenuation filters $H_i(z)$ that modify the frequency and total energy of the normal modes of the system's response, and a tone-correction filter $T(z)$ [45] that modify the system's power spectral density by imposing a desired magnitude frequency response. While we depict the model architecture with $N = 3$ in Fig. 3, our implementation uses $N = 6$.

As in [16], we aim to optimize $\boldsymbol{m}$, **b**, **c**, and d, whereas **A** is fixed throughout the procedure. The GA is run for 50 generations (i.e., ten times more than in [16]), each with a population of 50 FDNs. Each FDN is therefore an *individual* characterized by $3N + 1$ mutable parameters, namely, $\boldsymbol{m}$, **b**, **c**, and d. During the optimization, scalar gains are constrained to take values in [−1, 1]. Similarly, delays are constrained to take values in the range of 200 µs to 64 ms.

The attenuation filters are analytically determined according to the individual's delay values in $\boldsymbol{m}$ and the desired octave-band reverberation times [44]. In turn, the output graphic EQ filter [46] is found based on the initial level of the desired octave-band EDCs. All individuals implement the same random orthogonal feedback matrix [47], which is not affected by genetic optimization. The fitness of each individual at every generation is assessed through the mean absolute error between the MFCCs of the target RIR and those of the FDN output [16].

In our implementation, we avail of the Feedback Delay Network Toolbox by S. J. Schlecht [40] for fitting the graphic EQ filters and implementing the FDN, and use the GA solver included in MATLAB's Global Optimization Toolbox for finding $\boldsymbol{m}$, **b**, **c**, and d.

It is worth emphasizing the differences between the prototype FDN used in [16] (Fig. 3) and the proposed delay network (Fig. 1). First, [16] does not optimize the feedback matrix **A**, whereas we do. Second, [16] relies on IIR filters to achieve the desired reverberation time, whereas our model does not. Introducing $H_i(z)$ and $T(z)$ makes the baseline arguably more powerful in modeling a target RIR. At the same time, though, prior knowledge must be injected into the model by means of filter design to successfully run the GA and obtain meaningful results in a reasonable number of generations.

5.2 Evaluation metrics

As evaluation metrics, we select the T_{20}, T_{30}, and T_{60}, i.e., the reverberation time extrapolated considering the normalized IR energy decaying from −5 to −25 dB, −35 dB, and −65 dB, respectively. Ideally, these three metrics are the same if the EDC exhibits a perfectly linear slope. In practice, this is often not the case, as it can be seen, e.g., in Fig. 4. Hence, we believe that it is more informative to report all three of them, as together they provide a richer insight into the global behavior of the EDC as it approaches the −60 dB threshold. It is also worth pointing out that it is unclear whether the T_{60} is entirely reliable in measuring the reverberation time of real-world RIRs due to the often non-negligible noise floor.

Furthermore, we report the following ISO 3382 measures [48]: *Clarity* (C_{80}), expressed in dB, *Definition* (D_{50}),

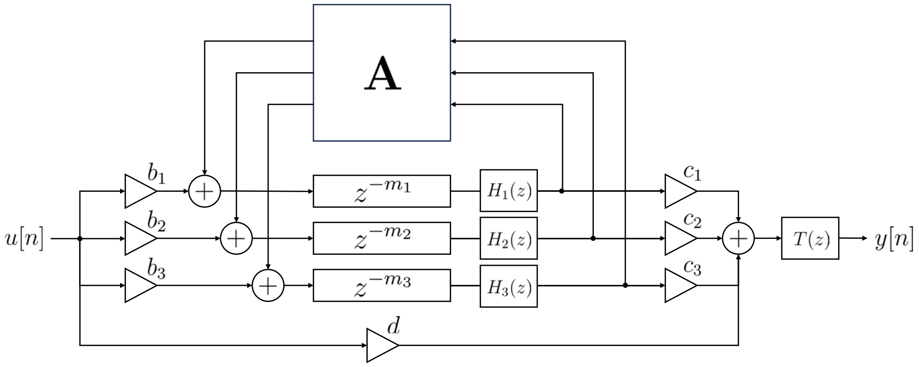


Fig. 3 Block diagram of the prototype FDN used in [16] with $N = 3$

Table 1 Reverberation time, iteration indices, loss values, and average time per training step in seconds (NVIDIA Tesla V100). Iterations denoted with (0) indicate pre-training random initialization

	T_{60}	iter	$\mathcal{L}$	$\mathcal{L}_{\text{EDC}}$	$\mathcal{L}_{\text{EDP}}$	time [s]
Gym	1.225	(0)	0.9959	0.9768	0.1909	41.86
		935	0.0067	0.0058	0.0095	
Hallway	0.607	(0)	1.0068	0.9812	0.2560	20.76
		796	0.0041	0.0034	0.0068	
Pizzeria	0.206	(0)	0.9254	0.9038	0.2161	5.71
		992	0.0526	0.0501	0.0255	

expressed as a percentage, and *Center time* (t_s), expressed in ms. Having defined $L_\tau := \lceil \tau \cdot f_s \cdot 10^{-3} \rceil$, these metrics are given by

$$C_{80} = 10 \log_{10} \frac{\sum_{n=0}^{L_{80}-1} h^2[n]}{\sum_{n=L_{80}}^{L-1} h^2[n]}, \tag{19}$$

$$D_{50} = 100 \cdot \frac{\sum_{n=0}^{L_{50}-1} h^2[n]}{\sum_{n=0}^{L-1} h^2[n]}, \tag{20}$$

$$t_s = 10^3 \cdot \frac{\sum_{n=0}^{L-1} n \cdot h^2[n]}{f_s \cdot \sum_{n=0}^{L-1} h^2[n]}. \tag{21}$$

For each metric, we report the error with respect to the target values. Absolute deviations are denoted by Δ in Tables 2, 3, and 4.

Finally, it is worth pointing out that the EDPs shown in the following sections are obtained using the (non-differentiable) formulation given in (11) unless explicitly stated otherwise.

5.3 Parameter initialization

We initialize the differentiable FDN parameters as follows. We let $\tilde{\mathbf{b}}^{(0)} \sim \mathcal{N}(\mathbf{0}, \frac{1}{N}\mathbf{I}_N)$, where $\mathbf{I}_N$ is the $N \times N$ identity matrix. We let $\tilde{\mathbf{c}}^{(0)} = \frac{1}{N}\mathbf{1}_N$, where $\mathbf{1}_N$ is a vector of N ones. We set $\tilde{d}^{(0)} = 1$. We initialize $\tilde{\mathbf{W}}^{(0)}$ and $\tilde{\mathbf{\Gamma}}^{(0)}$ so that $\tilde{\mathbf{W}}_{ij}^{(0)} \sim \mathcal{N}(0, \frac{1}{N})$ and $\tilde{\gamma}_i^{(0)} \sim \mathcal{N}(0, \frac{1}{N})$. We initialize $\tilde{\boldsymbol{m}}^{(0)}$ so that $\tilde{m}_i^{(0)} = \psi \tilde{m}_i^\star$ with $\tilde{m}_i^\star \sim \text{Beta}(\alpha, \beta)$, for $i = 1, ..., N$, where $\alpha \geq 1$ and $\beta > \alpha$. We empirically set $\psi = 1024$, $\alpha = 1.1$, and $\beta = 6$ to ensure a maximum possible delay of 64 ms (the same as in the GA baseline) and a mean value of about 10 ms. We let the Sigmoid scaling term κ_n increase linearly from 10^2 to 10^5 as $n = 0, ..., L_{T_{60}} - 1$.

5.4 Implementation details

We implement our differentiable model in Python using PyTorch. We define the FDN as a class inheriting from `nn.Module`. We thus define the unconstrained trainable parameters as instances of `nn.Parameter`. Our model operates at a sampling rate of 16 kHz. As a result, its memory footprint turns out to be contained, allowing us to train all FDNs considered in the present study on a single 16 GB NVIDIA Tesla V100 graphics card.[7] We optimize the models for a maximum of 1000 iterations using Adam [42] with a learning rate of 0.1, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and no weight decay. In all test cases, the EDP loss term is weighted by $\lambda = 0.1$.

The average training time per iteration is reported in Table 1. At each step, just below 24% of the time is taken by the forward pass of the FDN, approximately 12% is spent computing the loss function, and just above 64% is spent backpropagating the gradients and updating the parameters. Notably, we observe that the computation time increases linearly with the estimated T_{60}. For each test case, we present the model with the lowest composite loss.

5.5 Test case: Gym (h052)

We start by considering the Gym RIR (`h052`). As shown in Table 1, the best model is reached at iteration 935, after the loss has decreased by three orders of magnitude with respect to the initial value obtained with the random initialization described in Section 5.3. Here, we present a comparison between the target room acoustics (solid black line), the GA baseline [16] (dotted blue line), the HRTC baseline (dash-dotted green line), the Colorless baseline [18] (dash-dotted blue line), and, finally, the proposed differentiable FDN (dashed orange line).

Figures 4, 5, and 6 show that the proposed method is capable of closely matching the EDC, EDP, and envelope of the target RIR, respectively. Conversely, the baseline methods produce poorer results.

In Fig. 4, we may notice that the EDC of the GA baseline deviates from that of the target RIR after just 100 ms

[7] Preliminary experiments carried out with increased computational resources indicate that results comparable with those reported in the present study can be obtained at a sampling rate of 48 kHz.

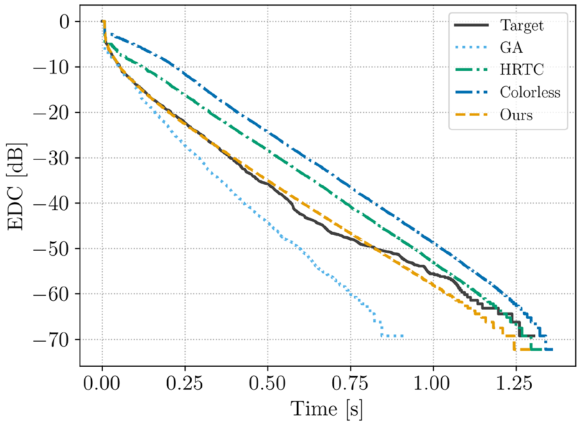


Fig. 4 Gym (h052) EDCs

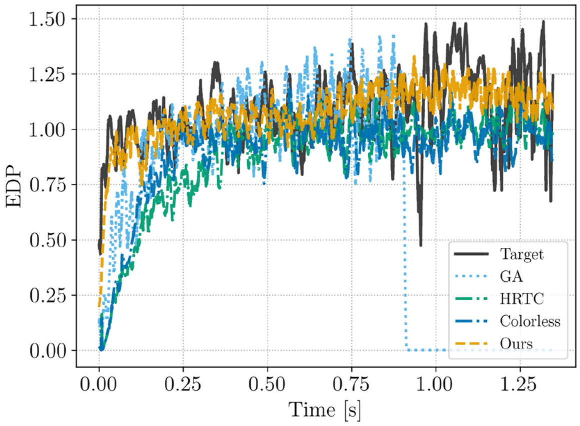


Fig. 5 Gym (h052) EDPs

and exhibits an overall steeper decay. The EDC of both the HRTC and Colorless baselines, instead, overshoot the target after the very first few ms before decaying with a linear slope. It is interesting to notice, though, that HRTC approaches and matches the target EDC at around the estimated T_{60}, i.e., 1.225 s, differently from the Colorless baseline, which had overshoot the target curve more.

In Fig. 5, the EDP of the baseline methods indicate a scarce echo density in the first 250 ms, especially for the HRTC and Colorless methods. The output of the FDN obtained with GA, instead, becomes identically zero just after approximately 0.9 s. The ensuing EDP pathologies are confirmed by the IR depicted in Fig. 6, where the baselines are shown to yield fewer, more prominent taps compared to the IR of the proposed FDN model.

Notably, Fig. 6 also suggests that all methods estimate a larger d than what would correctly render the direct sound. We attribute this phenomenon to an attempt at compensating the lack of a noise floor that, in real-life measurements, contributes to the total energy of the RIR. We argue that, in FDN models with tunable direct gain, offsetting this bias is naively achieved by increasing d.

Table 2 shows that the proposed method has an overall better performance in five out of the six reverberation metrics, with a ΔT_{20}, ΔT_{30}, ΔC_{80}, ΔD_{50}, and Δt_s of 16.5 ms, 55.2 ms, 0.02 dB, 0.09%, and 180 µs, respectively. In particular, T_{20}, T_{30}, and t_s are estimated with an error over one order of magnitude lower than those of the other methods. Likewise, the proposed FDN improves upon the baselines by two orders of magnitude as far as clarity C_{80} and definition D_{50} are concerned. On

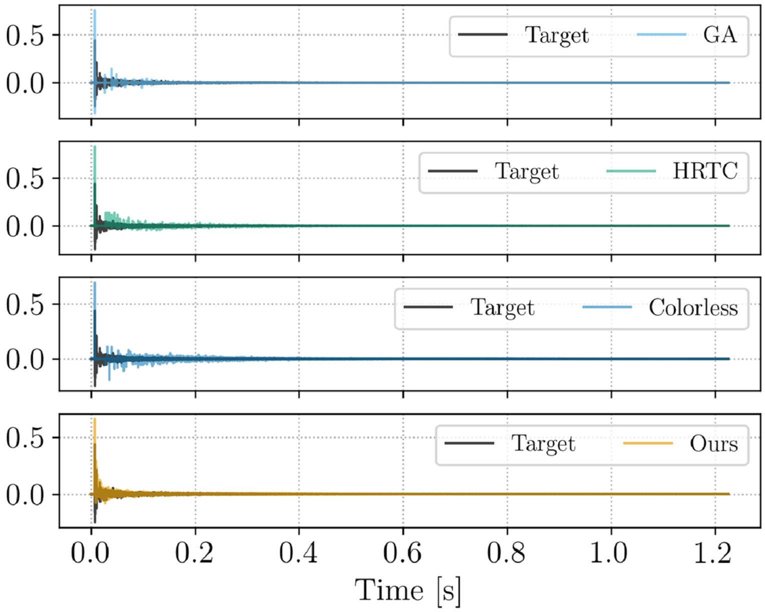


Fig. 6 Gym (h052) IRs. The time axis is limited to the T_{60} for visual clarity

Table 2 Metrics for the Gym RIR

	T_{20}	ΔT_{20}	T_{30}	ΔT_{30}	T_{60}	ΔT_{60}
Target	0.8616	–	0.9161	–	1.2257	–
GA	0.6334	0.2282	0.6969	0.2193	0.8117	0.4139
HRTC	1.2566	0.3951	1.2597	0.3436	1.2194	0.0063
Colorless	1.2589	0.3973	1.2504	0.3342	1.1900	0.0357
Ours	0.8451	0.0165	0.9714	0.0552	1.1355	0.0902
	C_{80}	ΔC_{80}	D_{50}	ΔD_{50}	t_s	Δt_s
Target	12.106	–	89.009	–	22.899	–
GA	11.985	0.1208	91.194	2.1848	19.343	3.5560
HRTC	7.8183	4.2877	79.402	9.6070	38.346	15.447
Colorless	2.7802	9.3258	56.329	32.679	77.035	54.135
Ours	12.126	0.0200	88.912	0.0974	23.079	0.1805

The best error values are underlined

the contrary, the proposed approach yields and error of 90.2 ms when it comes to the T_{60}, and it is surpassed by HRTC and Colorless, whose ΔT_{60} is 6.3 ms and 35.7 ms, respectively. This, however, was largely expected given that, in both baseline methods, the parameter γ is specifically designed to match the desired T_{60} according to (18).

5.6 Test case: Hallway (h270)

Let us now consider the Hallway RIR (`h270`), which is characterized by nearly half the T_{60} of the previous case. Table 1 shows that the best results are obtained at iteration 796, where the loss is again lower by three orders of magnitude with respect to the starting point.

Figure 7 reports the EDCs of target, baselines, and proposed methods following the color conventions reported in the previous subsection. Once again, we evince the good matching between ours and the target decay, especially in the early and late portion of the curve. GA correctly matches the target EDC only in the first 100 ms before rapidly decaying. The HRTC and Colorless methods, instead, present a sharp energy drop after the direct path, which results in IRs characterized by an almost total absence of reflections for the first few ms, as shown in Fig. 9. HRTC, in turn, presents a slight overshoot that lasts for the first 300 ms, after which it closely matches the target EDC. The proposed method, instead, deviates from the target in its central part, approximately from 180 to 500 ms. Arguably, however, these two kind of errors are not equivalent since the earlier portion of the RIR is known to be more relevant from a perceptual point of view [37].

Figure 8 indicates that our approach is able to closely match the target EDP, further validating the effectiveness of the proposed Soft EDP loss function. Indeed, the orange dashed curve follows the target curve until the T_{60}. We remind, in fact, that the training is performed only on such a span of time. Conversely, the EDP of the baseline methods show low values in the first 200 ms compared to the target one, and afterward, even if they take on comparable values, they do not follow the same trend. This is confirmed by the RIRs shown in Fig. 9. Indeed, even in this test case, the FDN optimized by means of the proposed approach has an IR that resembles more of a realistic RIR, even though the amplitude of individual taps is not entirely matched.

Table 3 reports the reverberation metrics. We can observe that the proposed optimization approach yields results comparable to the previous case, outperforming the baseline methods in five out of the six metrics. This time, due to the aforementioned mismatch in the central part of the EDCs in Fig. 7, HRTC and Colorless showcase a ΔT_{30} lower than that of the proposed method, with an error of 22.6 ms and 54.6 ms against the 85 ms obtained with our approach. Oppositely, both HRTC and the proposed FDN render the T_{60} equally well, with errors of 10.1 ms and 9.2 ms, respectively. Our ΔC_{80} is one order of magnitude less than what can be achieved with the other methods. Likewise, our ΔD_{50} is one order of magnitude less than what can be achieved with GA and HRTC, and two orders of magnitude less than Colorless'. In addition, the center time error Δt_s reaches 40.6 µs, while all baselines have errors one to three orders of magnitude larger, thus shifting the center of mass of the predicted IR energy more toward the reverberation tail.

5.7 Test case: Pizzeria (h214)

Finally, let us focus on the shortest RIR of the three considered in the present study, i.e., `h214`, having a T_{60} of just above 0.2 s.

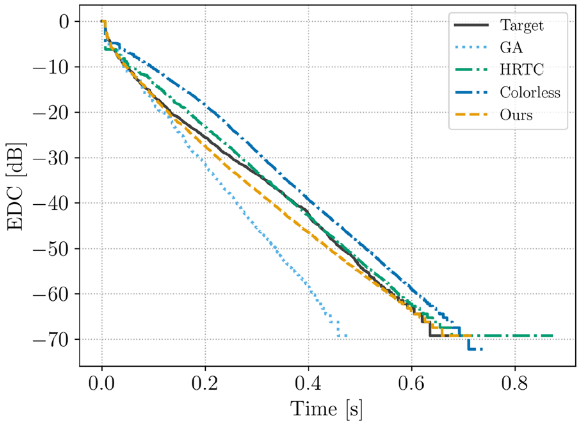


Fig. 7 Hallway (h270) EDCs

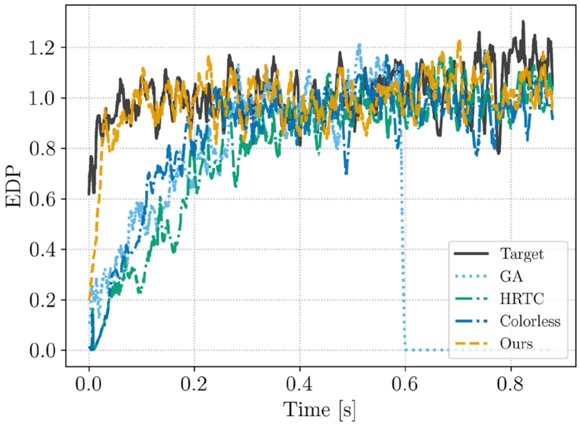


Fig. 8 Hallway (h270) EDPs

Here, all the baseline methods appear to fail at modeling the target room (Figs. 10 and 11). In particular, the IRs shown in Fig. 12 exhibit a few sparse taps, being thus far from resembling a real RIR. Since the backward-integrated energy abruptly decays with every peak, this results in a staircase-like behavior in the early portion of the EDC depicted in Fig. 10.

Facing the same difficulties, the proposed optimization method takes more gradient steps compared to the previous test case, converging at iteration 992 with a loss function two orders of magnitude lower than the starting value. In Fig. 10, the resulting EDC (dashed orange line) closely follows that of the target RIR (solid black line) until the two reach approximately −45 dB. Still, a direct comparison between EDCs in the range of −45 to −70 dB is not entirely reliable since the RIR has so little energy that background and sensor noise take on a much more relevant role when it comes to integrating the energy of the measured signal.

Overall, the proposed optimization method proves to perform well in fitting the RIR under scrutiny, with Table 4 reporting an error of 4.7 ms, 1.8 ms, and 12.6 ms when it comes to the three reverberation times. Furthermore, ΔC_{80} is 0.41 dB, ΔD_{50} is 0.13%, and Δt_s is 62.5 µs. As in Section 5.5, the only metric for which the proposed method shows a performance worse than one of the baselines is the T_{60}. As a matter of fact, the HRTC is the only baseline to be once again characterized by a lower ΔT_{60}, with a value of 5.2 ms.

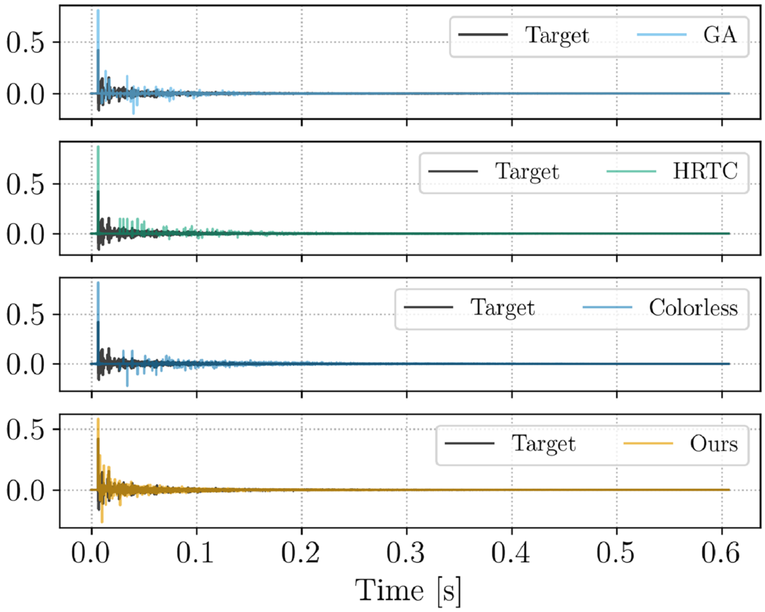


Fig. 9 Hallway (h270) IRs. The time axis is limited to the T_{60} for visual clarity

Table 3 Metrics for the Hallway RIR

	T_{20}	ΔT_{20}	T_{30}	ΔT_{30}	T_{60}	ΔT_{60}
Target	0.5289	–	0.6010	–	0.6067	–
GA	0.4200	0.1089	0.4233	0.1777	0.4329	0.1738
HRTC	0.6362	0.1072	0.6236	0.0226	0.6168	0.0101
Colorless	0.7007	0.1717	0.6556	0.0546	0.6335	0.0268
Ours	0.4749	0.0540	0.5160	0.0850	0.5975	0.0092
	C_{80}	ΔC_{80}	D_{50}	ΔD_{50}	t_s	Δt_s
Target	14.079	–	90.691	–	20.116	–
GA	15.502	1.4223	92.106	1.4147	18.365	1.7514
HRTC	11.412	2.6678	88.246	2.4453	20.793	0.6767
Colorless	8.327	5.7520	77.850	12.841	31.269	11.152
Ours	13.759	0.3204	90.856	0.1648	20.075	0.0406

The best error values are underlined

5.8 Excluding the EDP loss term

In developing our method, we noticed that using only the EDC loss function leads to ill-behaved IRs. Namely, we found that, while closely matching the desired EDC, the IR of an FDN trained with $\lambda = 0$, i.e., using only $\mathcal{L}_{\text{EDC}}$, tends to exhibit an unrealistic echo distribution compared to the RIRs of real-life environments. In this section, we compare the results of our differentiable FDN trained without Soft EDP regularization with those presented in Section 5.6 obtained using the composite loss function in (8). For conciseness, we limit our analysis to the Hallway RIR (`h270`); results obtained with other RIRs are comparable to what is shown below.

When excluding the EDP loss term from (8), the NMSE between true and predicted EDCs (10) is comparable with that of the proposed method, totaling 2.8×10^{-3} ($\lambda = 0$) and 3.4×10^{-3} ($\lambda = 0.1$), respectively. Yet, the MSE between true and predicted EDPs (17) is 0.342, i.e., two orders of magnitude higher than the 6.8×10^{-3} reported in Table 1 for the proposed method. This can be observed in Fig. 13b, showing that the echo density when $\lambda = 0$ (dash-dotted purple line) is far from the desired profile (solid black line). The target RIR and the proposed method (dashed orange line) produce an EDP with values consistently around one, indicating dense reverberation. On the contrary, the FDN trained without Soft EDP regularization yields an EDP with values below 0.5, signaling an uneven echo density due to the presence of a few prominent reflections [38]. This observation is confirmed by the IR shown in Fig. 13a that exhibits an exponentially decaying cluster of four taps periodically peaking above a denser reverberation tail with negative polarity.

Excluding the EDP loss term means solving a minimization problem with no constraints discouraging the model to use just a small number of delay lines to capture the overall EDC behavior. Since FDNs extend a parallel comb-filter structure, we believe that the behavior observed in Fig. 13a is related to the well-understood problem occurring when only some delay lines are strongly excited and recirculated, which, in turn, aggravates the comb-like behavior of the delay network, ultimately resulting in an unpleasing metallic sound quality [18, 41].

5.9 Soft EDP approximation

Finally, we discuss the approximation capabilities of the proposed Soft EDP function introduced in Section 4.5.2. Figure 14 shows the non-differentiable EDP (solid black line) defined in (11) against several Soft EDP approximations of the three RIRs considered in the present study. We test various scaling parameters κ, namely, 10^2, 10^3, and 10^4, along with the proposed time-varying κ_n linearly increasing from 10^2 ($n = 0$) to 10^5 ($n = L_{T_{60}} - 1$). We depict the profiles only for time indices below the T_{60}, as this range is the one considered when training the FDNs.

In Fig. 14, we may notice that $\kappa = 10^2$ yields a poor approximation of the reference EDP beyond the very first few ms. We also observe that $\kappa = 10^3$ and $\kappa = 10^4$ provide relative improvements. However, after some time, the approximation starts to degrade in a similar fashion as for $\kappa = 10^2$. Conversely, the proposed Soft EDP with time-varying scaling (dashed orange line) is able to closely match the non-differentiable reference profile all the way up to the T_{60} in Fig. 14a, while gracefully combating gradient vanishing (see Section 4.5.2). Notably, however, Fig. 14c shows that the approximation in the very last portion of the longest RIR considered, i.e., Gym (`h052`), significantly differs from the reference profile. To a lesser extent, this is also noticeable in Fig. 14b. Nevertheless, it is worth mentioning

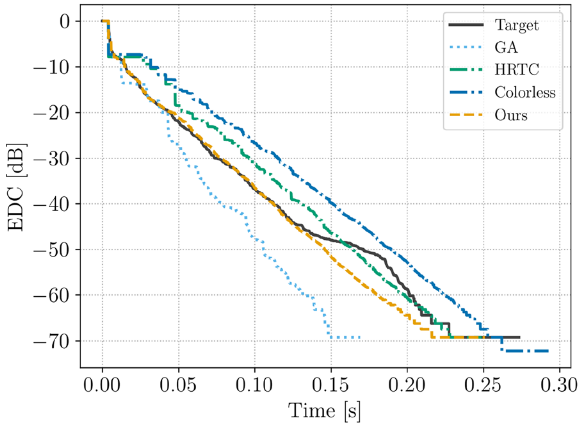


Fig. 10 Pizzeria (h214) EDCs

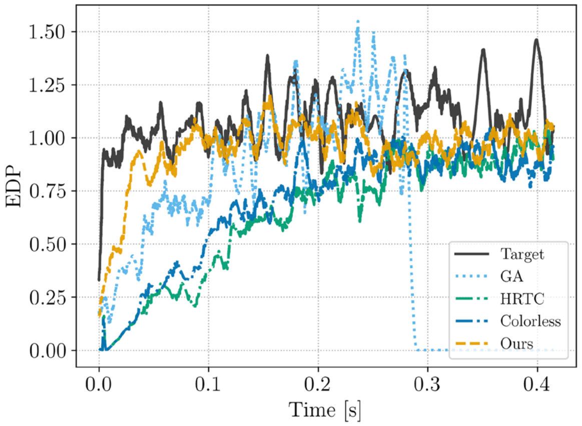


Fig. 11 Pizzeria (h214) EDPs

that, around the estimated T_{60}, the energy of the RIR is already almost entirely vanished, and the EDP itself suffers from statistical uncertainty due to sensor and 16-bit quantization noise.

5.10 Limitations

In this work, we focused on two important perceptual characteristics of room impulse responses: integrated energy decay and echo density. While rendering these time-domain features is key for any artificial reverberation algorithm that aims to be realistic, by themselves, Schroeder's EDC and Abel and Huang's EDP are not enough to comprehensively model the perceptual qualities of reverberation. In fact, it is well known that frequency- and time-frequency features play a crucial role in room acoustic simulation [4, 39].

However, Figs. 15, 16, and 17 show that none of the FDNs considered in the present study manages to capture the magnitude frequency response of the target RIRs. Likewise, Fig. 18 reveals a significant discrepancy between the target Energy Decay Relief (EDR) [49] and those of the FDN models.

Such a conspicuous mismatch entails that the output signals of the FDNs sound different not only from one another but also from the corresponding input signal convolved with the target RIR. In turn, this undermines the reliability of any perceptual test assigning similarity scores to each method with respect to the target, as subjective judgments would be significantly influenced by differing spectro-temporal coloration and decay. In this respect, pilot experiments proved inconclusive,

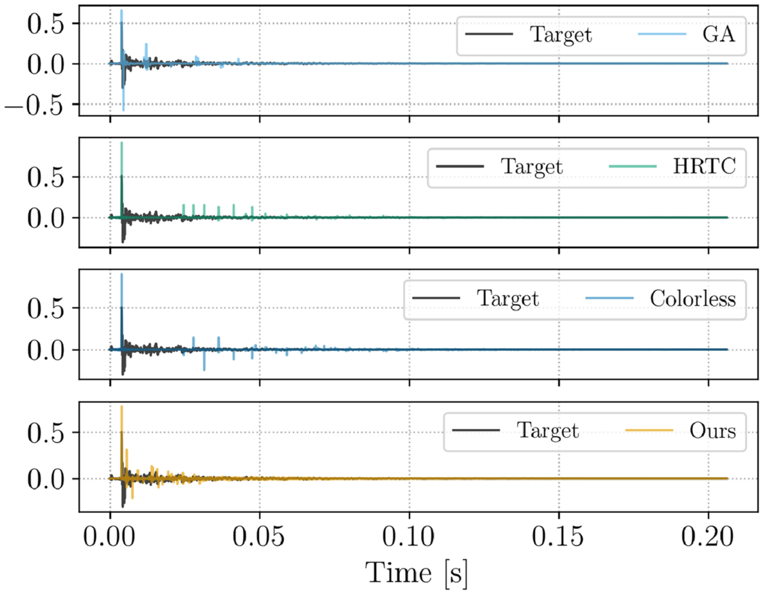


Fig. 12 Pizzeria (h214) IRs. The time axis is limited to the T_{60} for visual clarity

Table 4 Metrics for the Pizzeria RIR

	T_{20}	ΔT_{20}	T_{30}	ΔT_{30}	T_{60}	ΔT_{60}
Target	0.1643	–	0.1794	–	0.2062	–
GA	0.1172	0.0471	0.1235	0.0559	0.1417	0.0645
HRTC	0.2286	0.0643	0.2185	0.0391	0.2114	0.0052
Colorless	0.2668	0.1026	0.2557	0.0764	0.2367	0.0306
Ours	0.1689	0.0047	0.1811	0.0018	0.1936	0.0126
	C_{80}	ΔC_{80}	D_{50}	ΔD_{50}	t_s	Δt_s
Target	30.988	–	99.382	–	7.1698	–
GA	40.683	9.6950	99.799	0.4170	6.4524	0.7174
HRTC	24.860	6.1277	98.611	0.7708	9.4454	2.2756
Colorless	21.849	9.1389	96.895	2.4862	10.749	3.5798
Ours	30.576	0.4123	99.250	0.1322	7.2323	0.0625

The best error values are underlined

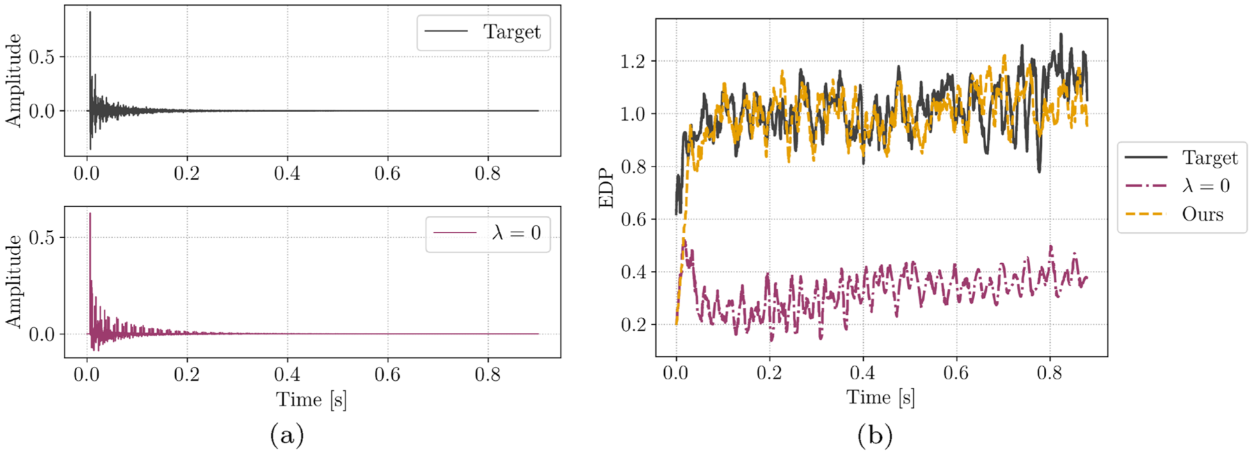


Fig. 13 **a** IR and **b** EDP obtained by training the FDN without EDP regularization term ($\lambda = 0$)

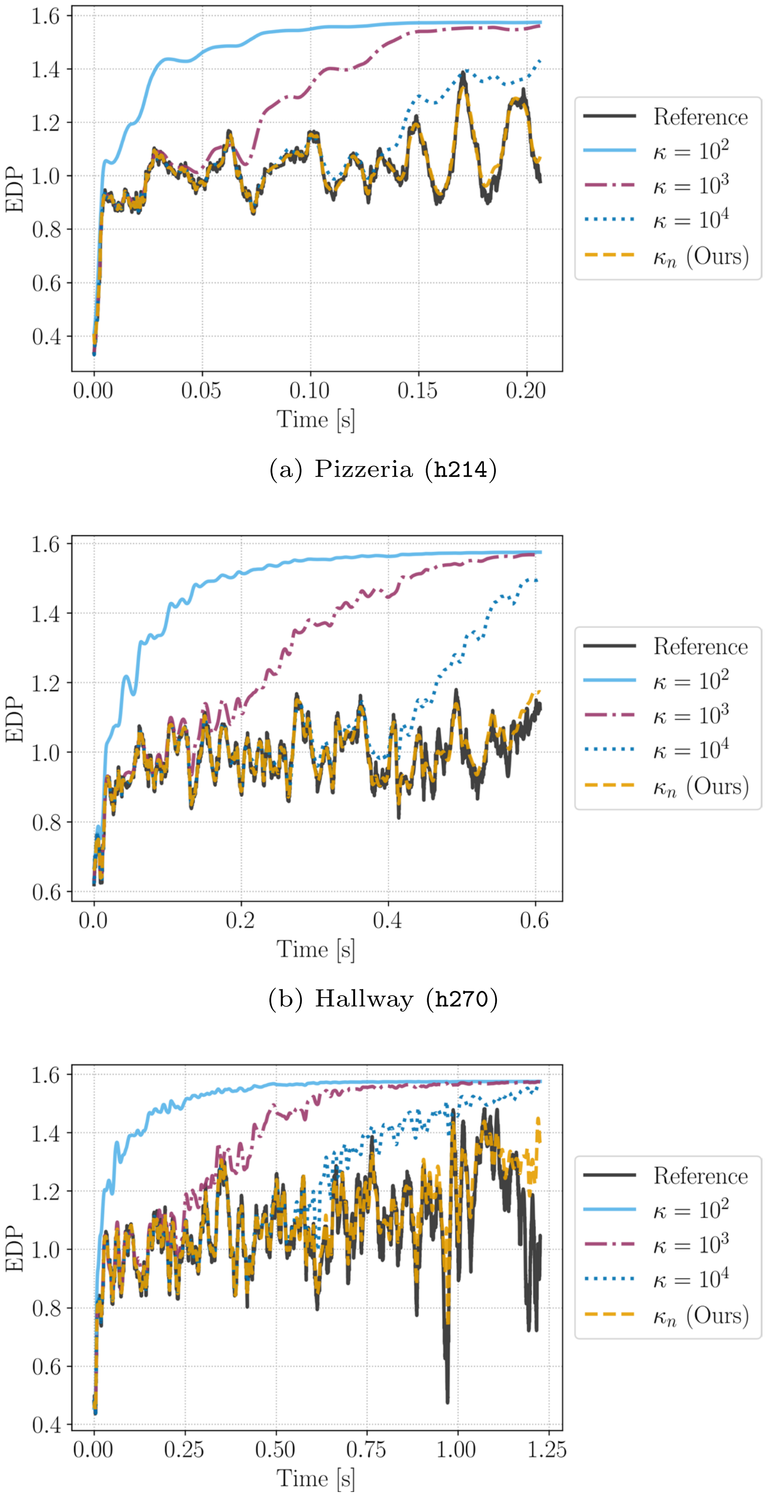
highlighting the need for further investigation into time-frequency modeling.

After all, matching the spectro-temporal characteristics of the target RIRs is not among the objectives of the parameter selection/tuning algorithms considered in the present study. Moreover, in this work, we mainly focused on time-invariant frequency-independent FDN prototypes. This holds true for proposed and baseline methods but GA, whose fitness function consists of a L^1-loss between MFCCs. Still, despite considering cepstral features, the result obtained with GA is far from resembling the reference spectro-temporal behavior. This evidences that it is not straightforward to accurately capture both time- and frequency-domain characteristics through an optimization process, even when including dedicated absorption and tone correction filters in the FDN prototype (cf. Fig. 3).

Although said filters can be implemented in a differentiable fashion, the poor performance of GA points out that extending the cost function (8) to the frequency-dependent case may not be sufficient, ultimately suggesting that a more thorough and comprehensive study is necessary to accomplish the goal. We leave such an investigation for future work.

6 Conclusions

In this work, we proposed a method for optimizing every parameter of a time-invariant frequency-independent feedback delay network (FDN) so as to match the reverberation of a given room through perceptually



(a) Pizzeria (h214)

(b) Hallway (h270)

(c) Gym (h052)

Fig. 14 Non-differentiable EDP (Reference) compared with the proposed Soft EDP function for different values of the Sigmoid scaling parameter

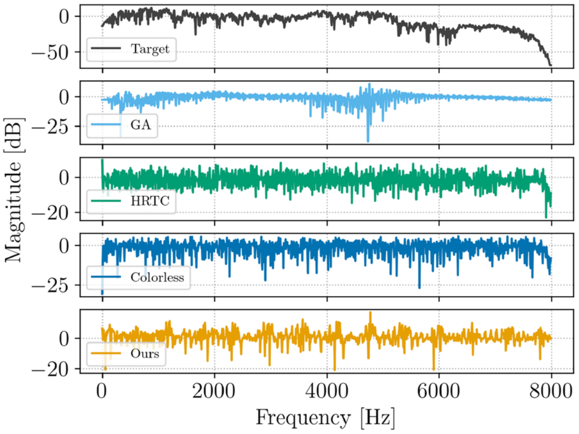


Fig. 15 Magnitude of the frequency response of the Gym RIR (`h052`) and corresponding FDN transfer functions

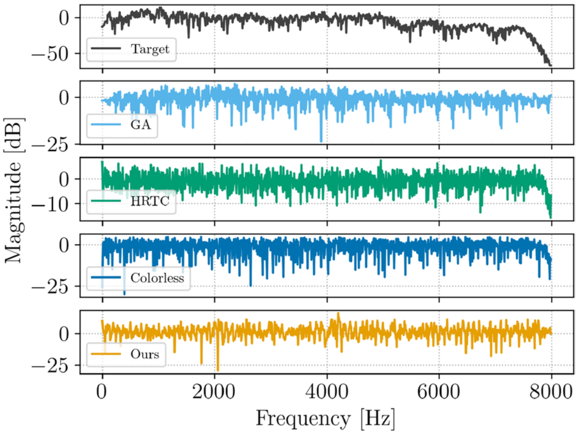


Fig. 16 Magnitude of the frequency response of the Hallway RIR (`h270`) and corresponding FDN transfer functions

meaningful metrics. The main contributions are the following:

i. We introduced a differentiable FDN with learnable delay lines.
ii. We developed a novel optimization framework for *all* FDN parameters based on automatic differentiation.

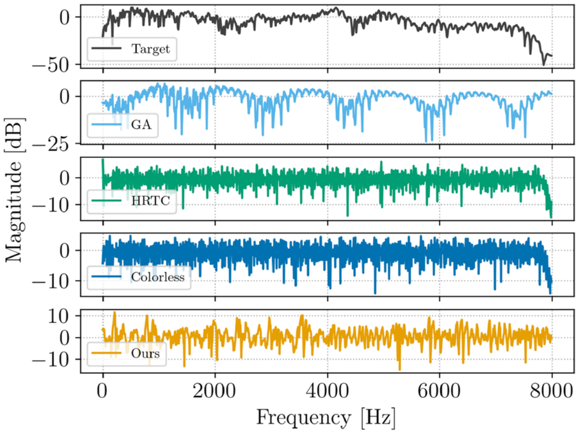
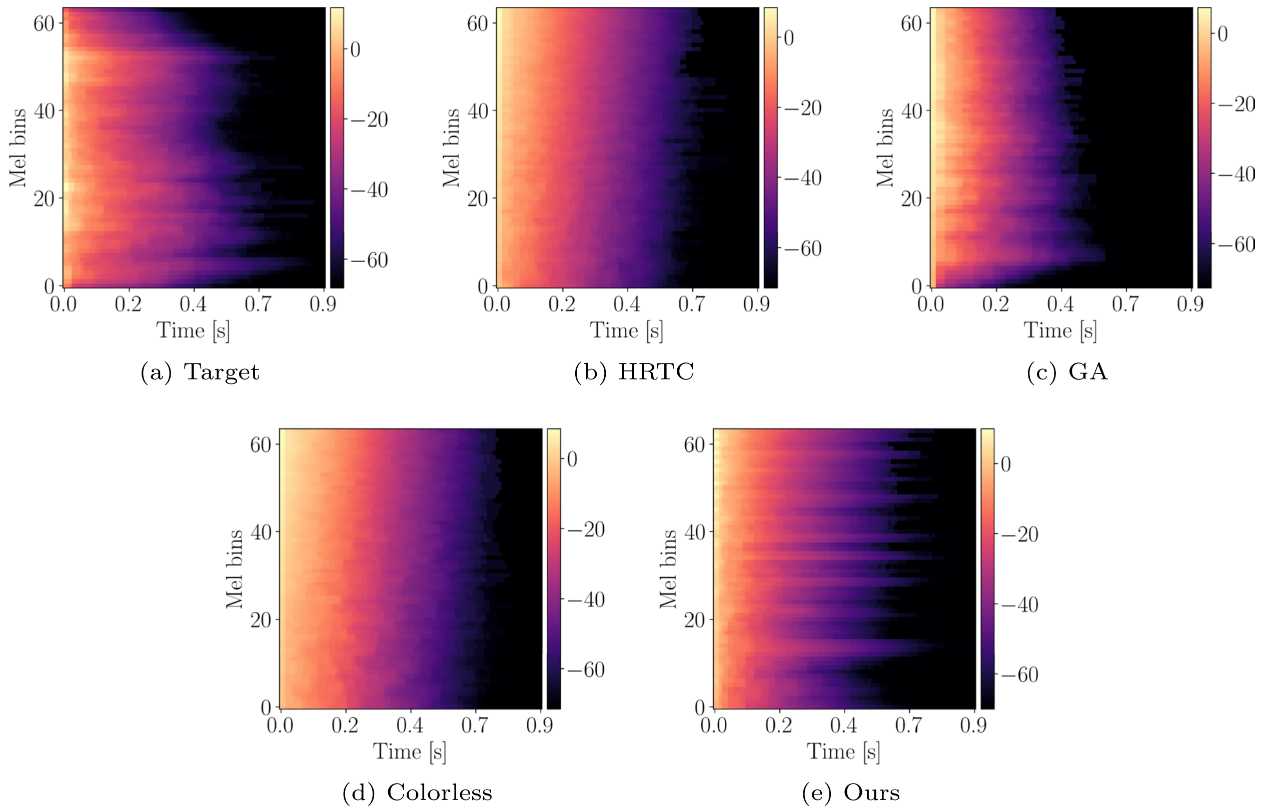


Fig. 17 Magnitude of the frequency response of the Pizzeria RIR (`h214`) and corresponding FDN transfer functions

iii. We applied gradient-based optimization with the objective of matching selected acoustic features of measured RIRs.
iv. We presented an innovative use of established perceptually motivated acoustic measures as loss terms.
v. We proposed a differentiable approximation of the well-known normalized Echo Density Profile named Soft EDP.

In particular, we presented a new differentiable FDN that is characterized by learnable delay lines realized exploiting operations in the frequency domain. Thus, we jointly trained all FDN parameters via backpropagation taking into account a composite loss consisting of two terms: the normalized mean square error between target and predicted backward-integrated EDCs and the mean square error between target and predicted Soft EDPs. We evaluated the proposed method on three real-world RIRs taken from a publicly available dataset, and we demonstrated that the Soft EDP term is essential for obtaining an IR that resembles a realistic RIR. Finally, we tested our approach against three baseline methods considering widespread metrics, including reverberation time, clarity, definition, and center time. Overall, the proposed approach was able to outperform



(a) Target (b) HRTC (c) GA (d) Colorless (e) Ours

Fig. 18 Mel-frequency Energy Decay Reliefs (EDRs) of the Hallway test case (h270). Proposed in [49], the EDR extends Schroeder's EDC to multiple frequency bands. Here, the 512-bin frequency axis is warped onto a 64-bin mel scale

the baseline methods by a large margin across different metrics.

Future work includes the application of the proposed framework to frequency-dependent FDNs, which are able to account for a frequency-specific decay in time, or to multiple-input multiple-output (MIMO) delay networks.

Acknowledgements
Alessandro Ilic Mezza and Riccardo Giampiccolo wish to thank Xenofon Karakonstantis for the helpful discussion.

Authors' contributions
A. I. Mezza conceptualized the study, designed and implemented the proposed method, run the experiments, and drafted the manuscript. R. Giampiccolo contributed to the design of the proposed method, run the experiments, and drafted the manuscript. E. De Sena drafted the manuscript and contributed to the methodology and codebase. A. Bernardini revised the manuscript and supervised the work.

Funding
This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 — program "RESTART"), and received funding support as part of the JRC STEAM STM–Politecnico di Milano agreement and from the Engineering and Physical Sciences Research Council (EPSRC) under the "SCalable Room Acoustics Modelling (SCReAM)" Grant EP/V002554/1.

Availability of data and materials
The datasets generated and/or analyzed during the current study are available on the MIT Acoustical Reverberation Scene Statistics Survey website, https://mcdermottlab.mit.edu/Reverb/IR_Survey.html.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 2 January 2024 Accepted: 11 September 2024
Published online: 08 October 2024

References
1. J.G. Apostolopoulos, P.A. Chou, B. Culbertson, T. Kalker, M.D. Trott, S. Wee, The road to immersive communication. Proc. IEEE **100**(4), 974–990 (2012)
2. T. Potter, Z. Cvetković, E. De Sena, On the relative importance of visual and spatial audio rendering on VR immersion. Front. Signal Process. **2** (2022). https://www.frontiersin.org/journals/signalprocessing/articles/10.3389/frsip.2022.904866/full
3. M. Geronazzo, J.Y. Tissieres, S. Serafin, in *Proc. 2020 IEEE Int. Conf. Acoust. Speech Signal Process.* A minimal personalization of dynamic binaural synthesis with mixed structural modeling and scattering delay networks (IEEE, New York, 2020), pp. 411–415
4. V. Välimäki, J.D. Parker, L. Savioja, J.O. Smith, J.S. Abel, Fifty years of artificial reverberation. IEEE Trans. Audio Speech Lang. Process. **20**(5), 1421–1448 (2012)

5. F. Wefers, *Partitioned Convolution Algorithms for Real-Time Auralization*, vol. 20 (Logos Verlag Berlin GmbH, Berlin, 2015)
6. M.R. Schroeder, Natural sounding artificial reverberation. J. Audio Eng. Soc. **10**(3), 219–223 (1961)
7. J.M. Jot, A. Chaigne, in *90th Audio Eng. Soc. Convention*. Digital delay networks for designing artificial reverberators (Audio Engineering Society, New York, 1991)
8. S.J. Schlecht, E.A.P. Habets, On lossless feedback delay networks. IEEE Trans. Sig. Process. **65**(6), 1554–1564 (2016)
9. H. Bai, G. Richard, L. Daudet, Late reverberation synthesis: from radiance transfer to feedback delay networks. IEEE Trans. Audio Speech Lang. Process. **23**(12), 2260–2271 (2015). https://doi.org/10.1109/TASLP.2015.2478116
10. E. De Sena, H. Hacıhabiboğlu, Z. Cvetković, J.O. Smith, Efficient synthesis of room acoustics via scattering delay networks. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(9), 1478–1492 (2015)
11. F. Stevens, D.T. Murphy, L. Savioja, V. Välimäki, Modeling sparsely reflecting outdoor acoustic scenes using the waveguide web. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(8), 1566–1578 (2017)
12. R. Bona, D. Fantini, G. Presti, M. Tiraboschi, J.I. Engel Alonso-Martinez, F. Avanzini, in *Proc. 17th Int. Audio Mostly Conf.* Automatic parameters tuning of late reverberation algorithms for audio augmented reality (Association for Computing Machinery, New York, 2022), pp. 36–43
13. M. Chemistruck, K. Marcolini, W. Pirkle, in *133rd Audio Eng. Soc. Convention*. Generating matrix coefficients for feedback delay networks using genetic algorithm (Audio Engineering Society, New York, 2012)
14. J. Shen, R. Duraiswami, in *Proc. 15th Int. Audio Mostly Conf.* Data-driven feedback delay network construction for real-time virtual room acoustics (Association for Computing Machinery, New York, 2020), pp. 46–52
15. J. Coggin, W. Pirkle, in *141st Audio Eng. Soc. Convention*. Automatic design of feedback delay network reverb parameters for impulse response matching (Audio Engineering Society, New York, 2016)
16. I. Ibnyahya, J.D. Reiss, in *153rd Audio Eng. Soc. Convention*. A method for matching room impulse responses with feedback delay networks (Audio Engineering Society, New York, 2022)
17. S. Lee, H.S. Choi, K. Lee, Differentiable artificial reverberation. IEEE/ACM Trans. Audio Speech Lang. Process. **30**, 2541–2556 (2022). https://doi.org/10.1109/TASLP.2022.3193298
18. G. Dal Santo, K. Prawda, S. Schlecht, V. Välimäki, in *Proc. 26th Int. Conf. Digital Audio Effects*. Differentiable feedback delay network for colorless reverberation (2023), pp. 244–251
19. A.I. Mezza, R. Giampiccolo, A. Bernardini, Data-driven parameter estimation of lumped-element models via automatic differentiation. IEEE Access **11**, 143601–143615 (2023). https://doi.org/10.1109/ACCESS.2023.3339890
20. A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey. J. Mach. Learn. Res. **18**, 1–43 (2018)
21. J.S. Abel, P. Huang, in *121st Audio Eng. Soc. Convention*. A simple, robust measure of reverberation echo density (Audio Engineering Society, New York, 2006)
22. D. Rocchesso, J. Smith, Circulant and elliptic feedback delay networks for artificial reverberation. IEEE Trans. Speech Audio Process. **5**(1), 51–63 (1997). https://doi.org/10.1109/89.554269
23. S.J. Schlecht, E.A.P. Habets, Feedback delay networks: echo density and mixing time. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(2), 374–383 (2016)
24. S.J. Schlecht, E.A.P. Habets, Time-varying feedback matrices in feedback delay networks and their application in artificial reverberation. J. Acoust. Soc. Am. **138**(3), 1389–1398 (2015)
25. A. Oppenheim, R. Schafer, J. Buck, *Discrete-Time Signal Processing*, 2nd edn. (Prentice Hall, Hoboken, 1999)
26. S. Heise, M. Hlatky, J. Loviscach, in *126th Audio Eng. Soc. Convention*. Automatic adjustment of off-the-shelf reverberation effects (Audio Engineering Society, New York, 2009)
27. L.J. Fogel, *Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming* (Wiley, Hoboken, 1999)
28. J.A. Nelder, R. Mead, A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)
29. J. Kennedy, R. Eberhart, in *Proc. Int. Conf. Neural Netw*. Particle swarm optimization, vol. 4 (IEEE, New York, 1995), pp. 1942–1948
30. M. Črepinšek, S.H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput. Surv. **45**(3), 1–33 (2013)
31. J. Engel, L.H. Hantrakul, C. Gu, A. Roberts, in *Int. Conf. Learning Representations*. DDSP: differentiable digital signal processing (2020)
32. F. Esqueda, B. Kuznetsov, J.D. Parker, in *Proc. 24th Int. Conf. Digital Audio Effects*. Differentiable white-box virtual analog modeling (2021), pp. 41–48
33. M. Shintani, A. Ueda, T. Sato, Accelerating parameter extraction of power mosfet models using automatic differentiation. IEEE Trans. Power Electron. **37**(3), 2970–2982 (2022). https://doi.org/10.1109/TPEL.2021.3118057
34. M. Lezcano-Casado, D. Martınez-Rubio, in *Int. Conf. Mach. Learning*. Cheap orthogonal constraints in neural networks: a simple parametrization of the orthogonal and unitary group (2019), pp. 3794–3803
35. S.C. Pei, Y.C. Lai, Closed form variable fractional time delay using FFT. IEEE Signal Process. Lett. **19**(5), 299–302 (2012). https://doi.org/10.1109/LSP.2012.2191280
36. M.R. Schroeder, New method of measuring reverberation time. J. Acoust. Soc. Am. **37**(6), 1187–1188 (1965)
37. D. Howard, J. Angus, *Acoustics and Psychoacoustics* (Routledge, London, 2013)
38. P. Huang, J.S. Abel, in *123rd Audio Eng. Soc. Convention*. Aspects of reverberation echo density (Audio Engineering Society, New York, 2007)
39. J. Traer, J.H. McDermott, Statistics of natural reverberation enable perceptual separation of sound and space. Proc. Natl. Acad. Sci. **113**(48), E7856–E7865 (2016). https://doi.org/10.1073/pnas.1612524113
40. S.J. Schlecht, in *Proc. 23rd Int. Conf. Digital Audio Effects*. FDNTB: the feedback delay network toolbox (2020), pp. 211–218
41. M. Schroeder, B. Logan, "Colorless" artificial reverberation. IRE Trans. Audio **AU-9**(6), 209–214 (Institute of Radio Engineers, New York, 1961). https://doi.org/10.1109/TAU.1961.1166351
42. D. Kingma, J. Ba, in *Int. Conf. Learning Representations*. Adam: a method for stochastic optimization (2015)
43. D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Boston, 1989)
44. S.J. Schlecht, E.A.P. Habets, in *Proc. 20th Int. Conf. Digital Audio Effects*. Accurate reverberation time control in feedback delay networks (2017), pp. 337–344
45. V. Välimäki, J. Liski, Accurate cascade graphic equalizer. IEEE Signal Process. Lett. **24**(2), 176–180 (2016)
46. V. Välimäki, J.D. Reiss, All about audio equalization: solutions and frontiers. Appl. Sci. **6**(5) (2016). https://doi.org/10.3390/app6050129
47. A. Edelman, N.R. Rao, Random matrix theory. Acta Numerica **14**, 233–297 (2005). https://doi.org/10.1017/S0962492904000236
48. Acoustics – Measurement of Room Acoustic Parameters. Part 1: Performance Spaces. ISO 3382-1:2009, International Organization for Standardization, Geneva, Switzerland, June 2009
49. J.M. Jot, in *Proc. 1992 IEEE Int. Conf. Acoust. Speech Signal Process*. An analysis/synthesis approach to real-time artificial reverberation, vol. 2 (IEEE, New York, 1992), pp. 221–224

Publisher's Note