Deep Optimization of Parametric IIR Filters for Audio Equalization

note finali coverpage

(Article begins on next page)

02 October 2024

Deep Optimization of Parametric IIR Filters for Audio Equalization

Giovanni Pepe, Leonardo Gabrielli, Stefano Squartini, Carlo Tripodi, and Nicolò Strozzi

Abstract—his paper describes a novel Deep Learning method for the design of IIR parametric filters for automatic multipoint audio equalization, that is the task of improving the sound quality of a listening environment at multiple listening points employing multiple loudspeakers. The filters are designed to approximate the inverse of the RIR and achieve almost flat magnitude response.his paper describes a novel Deep Learning method for the design of IIR parametric filters for automatic multipoint audio equalization, that is the task of improving the sound quality of a listening environment at multiple listening points employing multiple loudspeakers. The filters are designed to approximate the inverse of the RIR and achieve almost flat magnitude response.T A simple and effective neural architecture, named BiasNet, is proposed to determine the IIR equalizer parameters. This novel architecture is conceived for optimization and, as such, is able to produce optimal IIR equalizer parameters at its output, after training, with no input required. In absence of input, the presence of learnable non-zero bias terms ensures that the network works properly. An output scaling method is used to obtain accurate tuning of the IIR filters center frequency, quality factor and gain. All layers involved in the proposed method are shown to be differentiable, allowing backpropagation to optimize the network weights and achieve, after a number of training iterations, the optimal output according to a given RIR. The parameters are optimized with respect to a loss function based on a spectral distance between the measured and desired magnitude response, and a regularization term is used to keep the same microphone-loudspeaker energy balance after equalization. Two experimental scenarios are employed, a room and a car cabin, with several loudspeakers. The performance of the proposed method improves over the baseline techniques and achieves an almost flat band at a lower computational cost.

Index Terms—Deep Learning, Optimization, Audio Equalization, Machine Learning, IIR Filtering, Parametric Equalizer, Inverse Filtering

I. INTRODUCTION

PARAMETRIC equalizers are among the most used equalizers in audio and acoustics, thanks to the few parameters to control: center frequency, gain and quality factor [1], which boost or cut the spectrum. Such equalizers are usually composed of Infinite Impulse Response (IIR) filters, which are themselves formed by a cascade of Second Order Sections (SOS's), designed according to some constraints, such as a

G. Pepe is with the Department of Information Engineering, Università Politecnica delle Marche, Via Brecce Bianche, 60131, Ancona (AN), Italy and with ASK Industries S.p.A., Viale Ramazzini, 42124, Reggio Emilia (RE), Italy, e-mail: g.pepe@pm.univpm.it.

L. Gabrielli and S. Squaritni are with the Department of Information Engineering, Università Politecnica delle Marche, Via Brecce Bianche, 60131, Ancona (AN), Italy.

C. Tripodi and N. Strozzi are with ASK Industries S.p.A., Viale Ramazzini, 42124, Reggio Emilia (RE), Italy.

unitary gain at Direct Current (DC) and Nyquist frequency [2].

Audio equalization improves sound quality in listening environments, such as rooms, halls, car's cabins and so on [3], [4]. The in-car sound scene is changing in the last years [5], improving acoustic insulation and absorption and, thanks to hybrid and electric engines, leading to an interior sound quieter than the combustion vehicles [6]. Therefore, audio equalization for automotive applications has greater opportunity to improve the user experience inside the cars cabin. Audio enhancement in the automotive industry is generally done using Digital Signal Processing (DSP) systems [7], which have gotten cheaper and better performing in recent years.

Machine learning techniques, however, offer an alternative approach with respect to standard approaches and they are progressively adopted in audio equalization. Early works employing neural networks to invert a room response, can be found e.g. in [8], while recent end-to-end approaches have been proposed in [9]. Completely neural-based algorithms, however, require a dedicated hardware to run in real-time. Currently, embedded Tensor Processing Units (TPU) are getting commercially available as embedded devices, paving the way for the implementation of neural audio processing techniques in many application scenarios.

On the other hand, DSP chips are by far more convenient and energy-efficient for usage in audio equalization in scenarios such as the automotive one. For this reason, machine learning techniques may be used for the offline design of linear filters and then implemented in real time on a DSP chip. In the literature, several IIR filter design techniques are present [10], e.g., in [11], Prony's method [12] is used for warped IIR-type filter design to directly optimize filter coefficients. Other techniques are linear programming [13], non-linear optimization methods [14], [15] and evolutionary algorithms [16]. These are gradient-free algorithms [17], and among them, Particle Swarm Optimization (PSO) [18] and Gravitational Search Algorithm (GSA) [19] are the most widely used.

Recent works started employing neural networks for the design of IIR filters and parametric equalizers, mainly following the idea of Differentiable Digital Signal Processing (DDSP) layers [20]. These layers can backpropagate the error and are, thus, amenable for integration in a machine learning framework. In [21], [22], the filters are designed using state-of-the-art backpropagation algorithms have been presented, removing the convex optimization constraints imposed by different algorithms such as the least-squares approach which is widely used in classic adaptive filter theory. Another work which adopts the DDSP approach is introduced in [23], where

a neural network is trained to estimate the equalizing filter coefficients from a desired frequency response. Earlier works dealing with backpropagation for the design of IIR and Finite Impulse Response (FIR) filters are described in [24], [25]. Finally, the works in [26], [27] deal with the correction of octave and third-octave graphic equalizers gain, in order to reduce the error introduced by the interaction of neighboring filters, and achieve an accurate prediction of the equalizer gains with a feedforward neural network.

A. Scope of the work

Current filter design approaches based on the backpropagation of the error address the system identification problem [21], [22] or frequency response matching [23]. Our work presents two main differences with respect to the above techniques: the goal and the method.

Our goal is to design filters for room equalization, therefore the neural network does not have a reference frequency response to directly match. This task is challenging as the network must find suitable filter coefficients to approximate the room inverse response. Furthermore, we aim to obtain a room equalization not only in the Single-Input Single-Output (SISO) case, but also for any number of sound sources and listening points (Multiple-Input Multiple-Output or MIMO), which arguably makes the task more challenging. We assume that the test environment is static, enabling an offline filter design. Factors that may change the Room Impulse Responses (RIRs) over time (such as changes in the environment geometry, listener positions, temperature change, etc.) may impair the performance, and must be addressed using other techniques, such as adaptive filtering [28], [29], not covered in this paper. Our technique is not suitable for online adaptive equalization due to the computing time required for the filter design, but it is very useful in the pre-tuning phase, which always plays an important role in equalization of real acoustic environments.

As for the method, we employ neural networks with an unconventional approach, that we will refer to as Deep Optimization (DO), owing the name from the use of the neural network as an non-convex optimization algorithm. This approach stems from our previous works in equalizing filter design [30], [31]. However, here it is investigated in more detail and implemented in a simpler and more efficient way.

In our previous works, we first tackled room equalization filter design using evolutionary algorithms in [30], where FIR filters were designed in an automotive MIMO scenario. In [31], we largely improved the audio equalization performance by exploiting neural networks and outperforming the traditional Frequency Deconvolution (FD) method [32] by several orders of magnitude. This new filter design method opened several research questions related to network design that we are trying to address here. Furthermore, the design of FIR filters has some downsides: they have a high computational cost, and cannot be hand-tuned, which is an often desired feature, since listeners may require a personalized audio experience. To overcome this, IIR equalizers can be used. These provide a lower computational cost and the possibility of hand tuning after the automatic filter design is completed. Automatic IIR

parametric equalizers were first discussed in [33], where the Direct Search Method (DSM), a recursive technique, is used to optimize the parameters. The Rosenbrock method was presented in [34], where it was used to equalize an LCD TV speaker. We applied both methods to the room equalization problem in [35], and compared to GSA. The GSA algorithm gave superior performance, but still far from that obtained by the deep neural design of FIR filters proposed in [31]. For this reason, deep neural techniques are worth investigating for the design of an IIR equalizer for room equalization that can match the performance previously obtained with FIR filters.

The present work is organized as follows: in Section II we introduce the MIMO room equalization problem in formal terms and IIR parametric filters transfer functions and coefficients design. In Section III we propose a novel neural network architecture for the design of IIR equalizers for room equalization, along with a novel scaling method and loss function. In Section IV we introduce two alternative neural architectures and two baseline techniques for comparison purposes. In Sections V and VI, we introduce two experimental setups and their results, respectively. Finally, Section VIII concludes the paper.

II. PROBLEM STATEMENT

As previously stated, our main goal is to achieve MIMO room equalization with a parametric IIR equalizer, which is composed of a cascade of N second order digital IIR filters of the biquad type, i.e. SOS's. Parametric filters can be of the *peaking* or *shelving* type [36]. The former introduces a gain V_0 at a specific frequency f_c. The gain around f_c goes to 1 with a slope depending on the quality factor Q. The shelving filter, instead, introduces a flat gain starting from or up to a specific frequency. In this work, we restrict our attention to peaking filters to avoid unnecessary equalization at the extremes of the hearing range. Their coefficients are computed according to a closed form expression with its three free parameters. In our work, additionally, we include another free parameter, a *channel gain* V_s, one for each audio channel, i.e. sound source. Each equalizer can be expressed as a product of rational functions in the Z-domain (the individual SOS's transfer functions), i.e.:

$$G_s(z) = V_s \prod_{\kappa=1}^{N} \frac{b_{0,s,\kappa} + b_{1,s,\kappa}z^{-1} + b_{2,s,\kappa}z^{-2}}{a_{0,s,\kappa} + a_{1,s,\kappa}z^{-1} + a_{2,s,\kappa}z^{-2}} \tag{1}$$

where $G_s(z)$ is the transfer function of a parametric IIR filter of the s-th speaker, κ is the index of κ-th SOS, N is the number of SOS's and V_s is the channel gain in linear scale.

The coefficients are calculated according to the type of filter (boost or cut) as described in Table I. A parametric IIR filter composed of three SOSs is shown in Fig. 1 as an example.

One of the advantages of optimizing the peaking filter parameters, rather than directly design its b, a coefficients, is that the Q, center frequency and gain of each SOS can be easily constrained in the optimization problem. As we shall see later, e.g., the center frequency of each SOS is constrained in a given range, to avoid excessive overlap with neighboring filters. Similarly Q and gain are bounded to avoid excessive

resonance and ringing or too deep notches. Our previous approach [35] would not make it possible to express these constraints explicitly.

Overall, the MIMO room equalization system is composed of M listening points and $\mathcal{S}$ sound sources. Each sound source has an associated IIR equalizer, composed of N SOS's. Clearly, the SISO case can be regarded as a special case with $\mathcal{S} = 1, \mathcal{M} = 1$. The measured signal at the m-th listening position is

$$y_m(n) = \sum_{s=1}^{\mathcal{S}} x(n) * g_s(n) * h_{s,m}(n), \qquad (2)$$

where $g_s(n)$ is the impulse response of the s-th equalizer, $h_{s,m}(n)$ is the impulse response between the m-th point and the s-th source and $x(n)$ is an audio input signal.

The objective of an automatic room equalizer using parametric IIR filters is to determine the equalizer parameters f_c, Q, V_0 for each SOS and V_s for each speaker to achieve a flat (or any other desired) frequency response at the listening points. This can be seen as an optimization problem, as discussed in the following section.
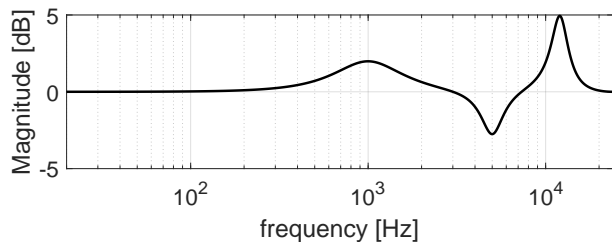

Fig. 1: Example of a parametric IIR filter composed of three SOS's. The center frequencies are: 1 kHz, 5 kHz and 12 kHz.

III. PROPOSED METHOD

As described in Section I-A, in this work we propose a novel and simpler neural network architecture for the design of room equalization IIR parametric filters. As demonstrated in [37], deep neural networks can solve non-convex problems through the minimization of a cost function. The design of equalizing filters can be formalized as such. This involves employing the neural network in a different way with respect to common classification and regression problems. The proposed approach is depicted in Fig. 2: the neural network initially computes filter coefficients based on random initial values of the weights. The coefficients are used, in conjunction with the room impulse responses to predict the frequency response obtained at one or more listening points. This is used to calculate the error with respect to the desired frequency response according to a given loss function. The error is then backpropagated to optimize the network weights in order to iteratively produce a more accurate frequency response with respect to the desired one.

In [31], we employed a convolutional neural network for FIR filter design for audio equalization. This, however, requires substantial computing power if the number of neurons in the convolutional layers is large. Their number is
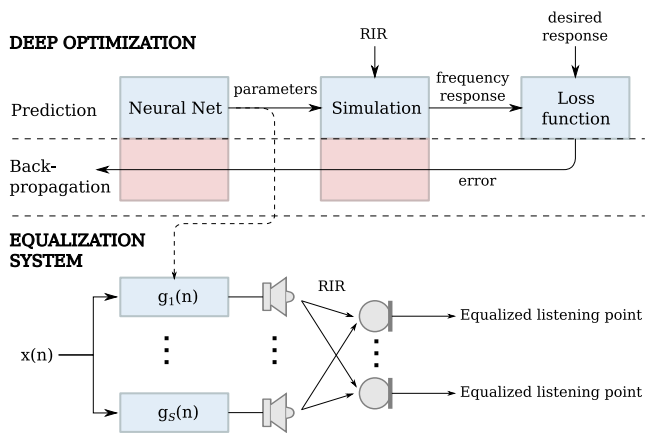

Fig. 2: Overview of Deep Optimization for room equalization. The algorithm consists of a forward pass (top) to predict equalizer parameters and the backpropagation of the error (bottom) to optimize the performance iteratively. The red boxes perform the backward flow of the partial derivatives of the operations performed by the blue boxes. Once the optimal filters parameters have been computed, these are stored into the audio equalizers $g_1(n)$ ... $g_{\mathcal{S}}(n)$ and used for real-time signal processing in the target room.

constrained by the size of the input. In absence of prior works, the neural network employed in [31] was fed with a three-dimensional input tensor containing the RIRs. This however, was shown to be not necessary for the network: any input of the same dimensions and size filled with non-zero values would achieve similar results. More specifically, a tensor of 1s would lead to slightly inferior results, while changing the input with random values at each iteration would result in a slow convergence of the network. Our experiments, discussed as part of this paper, allow concluding that the network does not require any informative content at its input. The only information that is required to perform an effective equalization is the knowledge of the room acoustic response in terms of RIR, used in the simulation stage. Building on these observations, in this work we propose BiasNet, an efficient neural network for optimization which avoids the need of convolutional input layers, as described in Section V and VI and in Table II.

As described previously, the architecture must be also designed to cope with IIR filters. In the FIR design task, the neural network directly computes the filters coefficients. This strategy is not suitable for IIR filters, since recursive filter poles are subject to stability and quantization issues [35]. We choose to estimate the parameters of a parametric IIR equalizer, then compute the IIR coefficients according to the closed form equations of Table I. The output layer of the network is, thus, composed of $(3 \times N) \times \mathcal{S} + \mathcal{S}$ neurons. As an additional novelty, the parameters are then denormalized by using linear interpolation in their defined range. However, to avoid overlapping filters and reduce the center frequency prediction error of each SOS, we propose a constrained scaling method of the center frequency parameter $p_{f_c,\kappa} \in [-1, 1]$, which is related to the center frequency of the κ-th SOS as

TABLE I
Coefficients calculated according to the boost or cut filter [36].

IIR Coefficients	Boost ($V_{0_{s,\kappa}} \geq 1$)	Cut ($0 < V_{0_{s,\kappa}} < 1$)
$b_{s,\kappa,0}$	$\dfrac{1+\frac{V_{0,s,\kappa}}{Q_{s,\kappa}}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}{1+1/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$	$\dfrac{1+\frac{1}{Q_{s,\kappa}}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}{1+V_{0,s,\kappa}/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$
$b_{s,\kappa,1}$	$\dfrac{2\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2-1}{1+1/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$	$\dfrac{2\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2-1}{1+V_{0,s,\kappa}/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$
$b_{s,\kappa,2}$	$\dfrac{1-\frac{V_{0,s,\kappa}}{Q_{s,\kappa}}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}{1+1/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$	$\dfrac{2\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2-1}{1+V_{0,s,\kappa}/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$
$a_{s,\kappa,0}$	1	1
$a_{s,\kappa,1}$	$\dfrac{2\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2-1}{1+1/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$	$\dfrac{2\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2-1}{1+V_{0,s,\kappa}/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$
$a_{s,\kappa,2}$	$\dfrac{1-\frac{1}{Q_{s,\kappa}}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}{1+1/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$	$\dfrac{1-\frac{V_{0,s,\kappa}}{Q_{s,\kappa}}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}{1+V_{0,s,\kappa}/Q_{s,\kappa}\cdot tan(\pi\cdot f_{c_{s,\kappa}}/f_s)+tan(\pi\cdot f_{c_{s,\kappa}}/f_s)^2}$

follows:

$$f_c = \frac{f_{c,max} - f_{c,min}}{2} \cdot p_{f_c,\kappa} + \frac{f_{c,max} + f_{c,min}}{2} \quad (3)$$

where $f_{c,max}$ and $f_{c,min}$ are the maximum and minimum allowed values for the center frequency. These values can be devised, e.g., according to third-octave bands, or any other subdivision of the audio range. This subdivision constrains the number of SOS's to the number of frequency bands, avoiding the overlap of the filters operative bandwidth which, in turn, may result in excessive gains for some bands. Furthermore, mapping the range $[-1, 1]$ to a narrow portion of the spectrum reduces the prediction error of the f_c. The other parameters are denormalized mapping $[-1, 1]$ to their full range which can be defined according to the application. Gains are designed by the network on a dB scale and are then converted into linear values when computing the IIR biquad equations of Table I. From now on, dB scale gains will be denoted as $V_{0,dB}, V_{s,dB}$ to avoid confusion with their linear counterparts V_0, V_s.

After computing the IIR filters, these are used to simulate the frequency response at the desired listening positions employing the related impulse responses. With respect to [22], instead of using convolution in time we compute the product in the frequency domain [38] due to its reduced computational cost. This optimization is important to reduce the overall computing time, as the optimization process involves a large number of iterations.

The frequency response of the numerator and the denominator of a SOS is given by:

$$B_{s,\kappa}(k) = \mathcal{F}[b_{s,\kappa}(z)] = b_{0_{s,\kappa}} + b_{1_{s,\kappa}}e^{-\frac{j2\pi k}{N}} + b_{2_{s,\kappa}}e^{-\frac{j2\pi 2k}{N}} \quad (4)$$

$$A_{s,\kappa}(k) = \mathcal{F}[a_{s,\kappa}(z)] = a_{0_{s,\kappa}} + a_{1_{s,\kappa}}e^{-\frac{j2\pi k}{N}} + a_{2_{s,\kappa}}e^{-\frac{j2\pi 2k}{N}} \quad (5)$$

The equalized frequency response $\tilde{H}_{s,m}(k)$ between the s-th speaker and the m-th microphone is given by:

$$\tilde{H}_{s,m}(k) = H_{s,m}(k) \cdot V_s \prod_{\kappa=1}^{K} \frac{B_{s,\kappa}(k)}{A_{s,\kappa}(k)} \quad (6)$$

where $H_{s,m}(k)$ is the frequency response achieved from the RIR by the discrete Fourier transform (DFT).

All the operations described above (scaling method, closed-form IIR coefficients computing, DFT filtering, etc.) must be differentiable in order to be employed in a neural network setting and to allow error backpropagation [20]. This will be discussed in Section III-C.

A. BiasNet

Within the DO scenario, our neural network can be seen as a unidirectional graph of nonlinear computations. If the input is fixed and only the network weights can change, then convergence is only determined by the update of the network weights, and is sufficiently fast as we shall see later. As shown in [31], if the activation functions satisfy the relation $f(0) = 0$ (i.e. no bias term is applied), an input tensor with 0s will not produce any output, as the result of the computations will be, similarly, 0s. This prevents the network to work properly. Therefore, some energy must be injected into the system. We do so by taking advantage of the bias terms of the input layer. These are learnable parameters, therefore, they will be updated during the optimization procedure. The vector of input neurons' bias terms $\mathbf{b_0}$ can be seen, i.e., as the optimal input vector for the given problem. The proposed architecture is called BiasNet, and consists of a feedforward network with no input but a learnable bias term, as shown in Fig. 3.

The advantage of using this architecture, with respect to the convolutional neural networks used previously, lies in: *(a)* the absence of inputs, which avoids useless tweaking of the input size and content; and *(b)* the low number of network parameters to be learned, which influences the convergence speed, as we shall see later.

B. Loss function

The cost function has a salient role in the optimization process since it computes the error and thus drives the network weights update procedure. The loss used in this work is a combination of a spectral loss L_1, and a multichannel energy regularization term L_2:

$$L = \gamma_1 \cdot L_1 + \gamma_2 \cdot L_2 \quad (7)$$

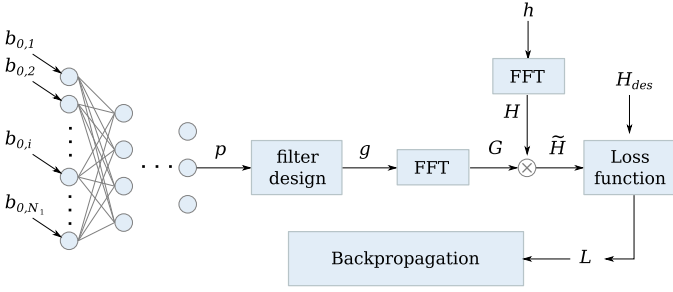with $\gamma_1 = 1$ and $\gamma_2 = log_2(\mathcal{S}) + log_2(\mathcal{M})$ being weights.

Fig. 3: BiasNet overview: the network provides parameters p that are denormalized and used of the filter design. The filters are transformed in the frequency domain, then frequency convolution is performed to compute the equalized response, that is compared with the desired response to compute the loss value. This is used in backpropagation to learn the optimal network weights and input bias terms.

The loss function L_1 is given by the Euclidean distance (ℓ_2-norm) between the simulated magnitude response and the desired magnitude response $|H_{des}(k)|$:

$$L_1 = \sum_{m=1}^{\mathcal{M}} \sqrt{\sum_k (|\tilde{H}_m(k)| - |H_{des}(k)|)^2} \tag{8}$$

where: $\tilde{H}_m(k)$ is the discrete Fourier transform of the equalized impulse response at the m-th microphone $\tilde{h}_m(n)$.

The regularization term L_2 is required when $\mathcal{S} > 1$ to keep the original energy balance between a reference speaker and each other speaker, avoiding unwanted change in spatial perception. The term is defined as:

$$L_2 = \sum_{m=1}^{\mathcal{M}} \sqrt{\sum_{s=1}^{\mathcal{S}} (\hat{r}_{s,m} - r_{s,m})^2} \tag{9}$$

where $r_{s,m}$ and $\tilde{r}_{s,m}$ are the ratio between the energy of a reference speaker and the s-th speaker before and after equalization, respectively.

C. Differentiating the DSP operations

The backpropagation of the error is done by computing the gradient of the loss function with respect to the network layers and operators, therefore these must be differentiable. As done in other works [22], [39], in this section we show that the analytical expressions of the gradients for the proposed method are computable and can be easily derived, making our solution viable. The results of the partial derivatives are reported for the sake of completeness, however, most Deep Learning software frameworks can automatically compute these derivatives if the implementation of the forward methods uses the operators provided by the framework, for which derivatives have been already developed, therefore for a practical application none of these must be implemented manually.

In our case, the backpropagation is performed using the partial derivative of the loss function with respect to the control parameters $\partial L/\partial f_c$, $\partial L/\partial Q$, $\partial L/\partial V_{0,dB}$ and $\partial L/\partial V_{s,dB}$. The partial derivatives are calculated as the product of cascaded local ones. While these are well known for all the

typical neural network layers used in this work, for the sake of completeness, we need to develop the mathematical expressions of the partial derivatives of all the DSP operations that allow to simulate the equalized RIRs, i.e.: those in Table I, Eq. 6 and so on.

In order to obtain the partial derivative of the loss function with respect to the control parameters, we first need to calculate the partial derivative of the loss function with respect to a generic filtered impulse response:

$$\frac{\partial L}{\partial \tilde{h}_{s,m}} = \frac{\partial L}{\partial L_1} \cdot \frac{\partial L_1}{\partial \tilde{h}_{s,m}} + \frac{\partial L}{\partial L_2} \cdot \frac{\partial L_2}{\partial \tilde{h}_{s,m}} \tag{10}$$

where, from Eq. (7), $\frac{\partial L}{\partial L_1} = \gamma_1$ and $\frac{\partial L}{\partial L_2} = \gamma_2$.

The partial derivative $\partial L_1/\partial \tilde{h}_{s,m}(n)$ is given by the product of:

$$\frac{\partial L_1}{\partial \tilde{h}_{s,m}(n)} = \frac{\partial L_1}{\partial \tilde{h}_m(n)} \cdot \frac{\partial \tilde{h}_m(n)}{\partial \tilde{h}_{s,m}(n)}. \tag{11}$$

Knowing that at one listening point $\tilde{h}_m(n) = \sum_{s=1}^{\mathcal{S}} \tilde{h}_{s,m}(n)$, we can state that $\frac{\partial \tilde{h}_m(n)}{\partial \tilde{h}_{s,m}(n)} = 1$, thus $\frac{\partial L_1}{\partial \tilde{h}_{s,m}(n)} = \frac{\partial L_1}{\partial \tilde{h}_m(n)}$. Since the filtered room response is computed in the frequency domain we expand Eq. 8 and use Wirtinger calculus [39] to get:

$$\frac{\partial L_1}{\partial \tilde{h}_m(n)} = \sum_{k=0}^{N-1} \left[\frac{|\tilde{H}_m(k)| - |H_{d,m}(k)|}{\sqrt{\sum_k (|\tilde{H}_m(k)| - |H_{d,m}(k)|)^2}} \cdot \frac{Re[\tilde{H}_m(k)]}{|\tilde{H}_m(k)|} \cdot \cos\left(\frac{2\pi}{N} kn\right) \right]$$
$$- \sum_{k=0}^{N-1} \left[\frac{|\tilde{H}_m(k)| - |H_{d,m}(k)|}{\sqrt{\sum_k (|\tilde{H}_m(k)| - |H_{d,m}(k)|)^2}} \cdot \frac{Im[\tilde{H}_m(k)]}{|\tilde{H}_m(k)|} \cdot \sin\left(\frac{2\pi}{N} kn\right) \right] \tag{12}$$

To calculate the local derivative of $\tilde{h}_{s,m}(n)$ with respect to a generic control parameter p, we first use the Wirtinger calculus [39] to determine the local derivative:

$$\frac{\partial \tilde{h}_{s,m}(n)}{\partial \tilde{H}_{s,m}(k)} = \sum_{n=0}^{N-1} \cos\left(2\pi \frac{kn}{N}\right) + \sin\left(2\pi \frac{kn}{N}\right) \tag{13}$$

The partial derivative of the magnitude response with respect to the channel gain $\frac{\partial \tilde{H}_{s,m}(k)}{\partial V_{s,dB}}$ is:

$$\frac{\partial \tilde{H}_{s,m}(k)}{\partial V_{s,dB}} = H_{s,m}(k) \cdot \frac{\log(10) \cdot 10^{V_{s,dB}/20}}{20} \prod_{j=1}^{K} \frac{B_{s,j}(k)}{A_{s,j}(k)} \tag{14}$$

For the other parameters, we first calculate the partial derivative of $\frac{\partial \tilde{H}_{s,m}(k)}{\partial B_{s,\kappa}(k)}$ and $\frac{\partial \tilde{H}_{s,m}(k)}{\partial A_{s,\kappa}(k)}$:

$$\frac{\partial \tilde{H}_{s,m}(k)}{\partial B_{s,\kappa}(k)} = H_{s,m}(k) \cdot V_s \cdot \frac{1}{A_{s,\kappa}(k)} \prod_{j=1,j\neq\kappa}^{K} \frac{B_{s,j}(k)}{A_{s,j}(k)} \tag{15}$$

$$\frac{\partial \tilde{H}_{s,m}(k)}{\partial A_{s,\kappa}(k)} = -H_{s,m}(k) \cdot V_s \cdot \frac{B_{s,\kappa}(k)}{A_{s,\kappa}^2(k)} \prod_{j=1,j\neq\kappa}^{K} \frac{B_{s,j}(k)}{A_{s,j}(k)} \tag{16}$$

The partial derivative of $\tilde{h}_{s,m}$ with respect to a generic parameter $p_{s,\kappa}$ is calculated exploiting the conversion from

time to frequency and from frequency to time using again the Wirtinger calculus:

$$
\begin{aligned}
\frac{\partial \tilde{h}_{s,m}}{\partial p_{s,\kappa}} =& \left(\sum_{n=0}^{N-1} cos\left(2\pi \frac{kn}{N}\right) + \sum_{n=0}^{N-1} sin\left(2\pi \frac{kn}{N}\right)\right) \cdot \left[\frac{\partial b_{0_{s,\kappa}}}{\partial p_{s,\kappa}} \cdot \sum_{k=0}^{N-1} \frac{\partial \tilde{H}_{s,m}(k)}{Re(B_{s,m}(k))} +\right. \\
& \frac{\partial b_{1_{s,\kappa}}}{\partial p_{s,\kappa}} \cdot \left(\sum_{k=0}^{N-1} \frac{\partial \tilde{H}_{s,m}(k)}{Re(B_{s,m}(k))} cos\left(2\pi \frac{k}{N}\right) - \sum_{k=0}^{N-1} \frac{\partial \tilde{H}_{s,m}(k)}{Im(B_{s,m}(k))} sin\left(2\pi \frac{k}{N}\right)\right) + \\
& \frac{\partial b_{2_{s,\kappa}}}{\partial p_{s,\kappa}} \cdot \left(\sum_{k=0}^{N-1} \frac{\partial \tilde{H}_{s,m}(k)}{Re(B_{s,m}(k))} cos\left(2\pi \frac{2k}{N}\right) - \sum_{k=0}^{N-1} \frac{\partial \tilde{H}_{s,m}(k)}{Im(B_{s,m}(k))} sin\left(2\pi \frac{2k}{N}\right)\right) + \\
& \frac{\partial a_{1_{s,\kappa}}}{\partial p_{s,\kappa}} \cdot \left(\sum_{k=0}^{N-1} \frac{\partial \tilde{H}_{s,m}(k)}{Re(A_{s,m}(k))} cos\left(2\pi \frac{k}{N}\right) - \sum_{k=0}^{N-1} \frac{\partial \tilde{H}_{s,m}(k)}{Im(A_{s,m}(k))} sin\left(2\pi \frac{k}{N}\right)\right) + \\
& \left. \frac{\partial a_{2_{s,\kappa}}}{\partial p_{s,\kappa}} \cdot \left(\sum_{k=0}^{N-1} \frac{\partial \tilde{H}_{s,m}(k)}{Re(A_{s,m}(k))} cos\left(2\pi \frac{2k}{N}\right) - \sum_{k=0}^{N-1} \frac{\partial \tilde{H}_{s,m}(k)}{Im(A_{s,m}(k))} sin\left(2\pi \frac{2k}{N}\right)\right)\right]
\end{aligned}
\tag{17}
$$

where, through Wirtinger calculus, we have that: $\frac{\partial B(k)}{\partial b_{0,\kappa}} = \frac{\partial A(k)}{\partial a_{0,\kappa}} = 1$ and $\frac{\partial B(k)}{\partial b_{1,\kappa}} = \frac{\partial B(k)}{\partial b_{2,\kappa}} = \frac{\partial A(k)}{\partial a_{1,\kappa}} = \frac{\partial A(k)}{\partial a_{2,\kappa}} = cos(2\pi \frac{k}{N}) - sin(2\pi \frac{k}{N})$.

The local derivative of the IIR filter coefficients with respect to the parameters f_c, $V_{0,dB}$ and Q can be calculated through the partial derivatives of the equations in Table I. For example, the local derivative of the coefficient $b_{s,\kappa,0}$ with respect to $f_{c_{s,\kappa}}$ for the Boost filter is:

$$
\frac{\partial b_{s,\kappa,0}}{\partial f_{c_{s,\kappa}}} = \frac{\pi \cdot \left(\frac{1}{Q_{s,\kappa}} - \frac{10^{V_{s,\kappa,dB}/20}}{Q_{s,\kappa}}\right) \cdot \left(tan^2(\pi \frac{f_{c_{s,\kappa}}}{f_s}) - 1\right) \cdot \frac{1}{cos^2(\pi \frac{f_{c_{s,\kappa}}}{f_s})}}{f_s \cdot \left(\frac{1}{Q_{s,\kappa}} tan(\pi \frac{f_{c_{s,\kappa}}}{f_s}) + tan^2(\pi \frac{f_{c_{s,\kappa}}}{f_s}) + 1\right)^2}
\tag{18}
$$

Finally, the derivative of the scaling method step is:

$$
\frac{\partial q}{\partial p} = \frac{q_{max} - q_{min}}{2}
\tag{19}
$$

where p is the normalized parameter, q is any of the equalizer parameters (denormalized), i.e. $f_c, Q, V_{0,dB}, V_{s,dB}$ and the terms q_{max}, q_{min} denote its range. We note that for f_c the range is different for each SOS (see Eq. (3)), while the other parameters have identical ranges for each SOS.

The local derivative with respect to the regularization term is given by the cascade of the following local derivatives:

$$
\frac{\partial L_2}{\partial \tilde{h}_{s,m}(n)} = \frac{\partial L_2}{\partial \hat{r}_{s,m}} \cdot \frac{\partial \hat{r}_{s,m}}{\partial \hat{\epsilon}_{s,m}} \cdot \frac{\partial \hat{\epsilon}_{s,m}}{\partial \tilde{h}_{s,m}(n)}
\tag{20}
$$

where:

$$
\frac{\partial L_2}{\partial \hat{r}_{s,m}} = \frac{\hat{r}_{s,m} - r_{s,m}}{\sqrt{\sum_{s=1}^{\mathcal{S}} (\hat{r}_{s,m} - r_{s,m})^2}},
\tag{21}
$$

$$
\frac{\partial \hat{r}_{s,m}}{\partial \hat{\epsilon}_{s,m}} = -\frac{\hat{\epsilon}_{1,m}}{\hat{\epsilon}_{s,m}^2},
\tag{22}
$$

$$
\frac{\partial \hat{\epsilon}_{s,m}}{\partial \tilde{h}_{s,m}(n)} = 2 \cdot \tilde{h}_{s,m}(n).
\tag{23}
$$

D. Direct Optimization

It is worth mentioning that the flexibility of the neural network allows to shrink the network down to one single layer. In this case, the algorithm reduces to the direct optimization of the filter parameters by means of a simple update rule. Indeed,

with one layer only, during backpropagation, the parameter p is updated as:

$$
p(n+1) = p(n) - \phi \frac{\partial L}{\partial p}
\tag{24}
$$

the parameter p is the output of the activation function with the bias term as input, thus:

$$
p(n+1) = f(b_0(n+1)) = f(b_0(n)) - \phi \frac{\partial L}{\partial p}
\tag{25}
$$

where ϕ is the learning rate. In the experimental section we will test a configuration of the BiasNet (#4, see Table II) with one layer only, and a version of the same configuration (#4L) with linear activation function. The latter shows how the direct optimization performs in absence of any of the features mutuated from neural networks.

IV. COMPARATIVE METHODS

To conduct a thorough analysis of the proposed method, we compare it to several additional techniques that can be employed for the room equalization task. We propose two DO techniques, both based on convolutional neural networks, an evolutionary IIR filter design method based on the Direct Search Method [33] and the well-known Frequency Deconvolution method for the design of FIR filters [32].

A. Alternative Deep Optimization Networks

In addition to BiasNet, other DO architectures are proposed based on similar principles. The first is a regular CNN, as in our previous work [31]. This network has been adapted to the design of IIR filters as described with BiasNet, i.e. by generating the IIR parameters $f_c, Q, V_{0,dB}, V_{s,dB}$ and using the scaling method rule described in Eq. (3). The network employs a fixed input, the RIR tensor, as done in our previous work.

The second network is a CNN with variable input. The idea stems from the same observations drawn in describing the BiasNet, i.e. that of a network adapting its input, in addition to its weights. At the first iteration we use the RIR as input tensor, however, at each next iteration the network is fed with the equalized RIR. This network, therefore, establishes an input-output feedback, since the equalized response generated at the j-th iteration is fed as input at iteration $(j + 1)$. In the following, we shall call this architecture Convolutional Feedback Network (CFN).

The rationale for testing the CNN is verifying how well it performs for the new task, after adapting its output layers for the IIR equalizer parameters design, and assessing the computational cost and performance gain of the proposed BiasNet architecture. The CFN, on the other hand, provides an alternative approach to input optimization, that is worth investigating in terms of performance. Similarly to BiasNet it is able to learn an optimal input, however it uses a different strategy and a larger number of parameters.

B. Direct Search Method

The Direct Search Method is a heuristic technique that is used for optimization problems [40]. This method is a derivative-free optimization algorithm and can work with no constraints. As will be described below, this technique is simple to implement and has been used to solve many optimization problems [41].

The authors in [33] implemented the Direct Search Method for an IIR parametric equalizer. This heuristic algorithm is described as in [34]: a random variation Γ in the range $-\gamma \leq \Gamma \geq \gamma$ is added to a generic parameter vector c:

$$\hat{c}_i = c_i \cdot (1 + \Gamma) \qquad (26)$$

If the new parameters achieve a better cost function, they are kept, otherwise they are rejected and another random variation Γ is performed. The process continues until the demands have been met. For our case, c is composed of the parameters of the IIR filters $f_c, Q, V_{0,dB}, V_{s,dB}$.

C. Frequency Deconvolution

This method is based on deconvolution in the frequency domain to estimate a room inverse that is optimal in the Least Squares sense [32], i.e. the cost function is convex. The optimal filters $G(k)$ are computed in the frequency domain by:

$$G(k) = [H^H(k)H(k) + \beta_{FD}I]^{-1}H^H(k)D(k) \qquad (27)$$

where I is the identity matrix, ^{H} is the Hermitian operator, $H(k)$ is the transfer function matrix of the impulse responses, $D(k)$ is the target frequency responses matrix and β_{FD} is a regularization term that avoids extreme peaks in the inverse filters. The inverse FFT and then a circular shift of K are computed, where K is the Fast Fourier Transform (FFT) size, to achieve the filters in the time-domain. Conducting deconvolution in the frequency domain speeds up the computation, by taking advantage of the FFT algorithm.

This method is generally taken as a reference baseline for FIR equalization of a room impulse response and its solution is regarded as optimal. It must be noted that the solution is such only in the Least Squares sense, therefore by posing the problem in non-convex terms, as is done with our proposed method, other solutions may exist that solve the room equalization problem with similar or superior spectral flatness.

V. Experiments

To validate the proposed method, we perform experiments with two scenarios: a regular room and a car cabin. The first scenario is a rectangular room of dimension $4.0 \times 5.5 \times 3.0\,\text{m}$ and composed of 8 speakers circularly placed around two seats, as shown in Fig. 4. The left seat is fitted with two omnidirectional microphones simulating the listener's ears. The loudspeakers are mid-woofers with a frequency range between 100 Hz and 15 kHz. Impulse responses were measured using the exponential sine sweep method [42] with a sampling
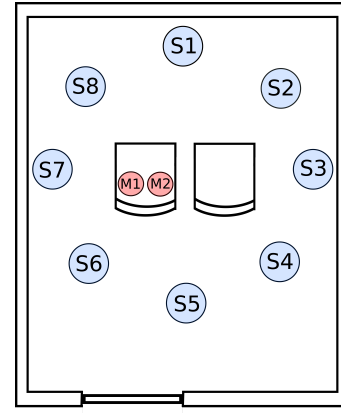

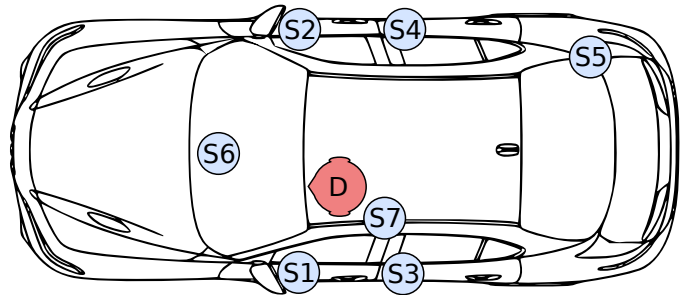Fig. 4: Top view of the room showing the placement of the speakers and microphones.

Fig. 5: Top view of the car's cabin showing the placement of the speakers and the microphones. D corresponds to the mannequin head with microphones mounted in the ears.

frequency of 48 kHz, using a RME Madiface audio interface, and a Dante-equipped amplification system.

The second scenario is a car cabin, an Alfa Romeo Giulia, the same used for our previous works. Impulse responses were measured using the sine sweep method [42] and measuring them using Aurora plug-in [43] and a Kemar 45BA mannequin placed on the driver's seat. The sampling frequency equals 28.8 kHz and then oversampled to 48 kHz.

The two scenarios exhibit different characteristics. The room employs only one type of loudspeakers arranged in a regular pattern. These cover slightly more than two decades of the audio range and the maximum excursion between minima and maxima in the unequalized frequency responses never exceed $\sim 15\,\text{dB}$. On the contrary, the car is fitted with loudspeakers of different size and bandwidth, arranged in a irregular fashion. Furthermore, the car material absorption coefficient varies largely. This results in a large excursion of the frequency response ($>20\,\text{dB}$) and a wider range to equalize, covering almost three decades. The car cabin is, therefore, a more challenging scenario.

Preliminary experiments were, thus, conducted in the room, to gather more insights on the described equalization techniques, to compare the DO methods discussed above and find suitable hyperparameters. More specifically, we conducted a hyperparameters search in the MIMO case and kept the best architecture for all subsequent experiments. The experiments

include SISO, MIMO and multiple-input single-output (MISO) cases for the room environment. After discussing these results we turn our attention to the car cabin scenario, using all its speakers and microphones (7×2 MIMO scenario). The desired frequency response $|H_{des}|$ for our work is a $0 \, \text{dB}$ flat band in the range $100 \, \text{Hz}$-$14 \, \text{kHz}$ for the room scenario, and $20 \, \text{Hz}$-$14 \, \text{kHz}$ for the car scenario.

Both optimization and evaluation are conducted using a third-octave band averaging for the frequency responses. This choice is motivated by the human ear resolution which is not very sensitive to narrow dips and notches. The number of SOS's (and, similarly, the bands used during scaling method of the f_c parameter) is equal to the number of third-octave bands within the speaker's operating frequency range. This choice avoids the overlapping of filters: if they cancel or concur to the same objective there is a waste of computational resources. This makes for 22 SOS's for each speaker in the room scenario, and from 21 to 29 in the cabin scenario. Moreover, each SOS's is composed of three parameters (f_c, Q and V_0) plus the overall gain (V_s), one per speaker. Thus, the number of parameters to optimize is 536 and 529 for the room and car cabin scenario, respectively. The ranges of the other parameters, identical for all SOS's, are: $Q_{min} = 0.05$, $Q_{max} = 5.0$, $V_{0,min,dB} = -10 \, \text{dB}$, $V_{0,max,dB} = 10 \, \text{dB}$, $V_{s,min,dB} = -20 \, \text{dB}$, $V_{s,max,dB} = 20 \, \text{dB}$.

Regarding the FD, we searched for the best regularization term β_{FD} following the strategy used in [31] and found that a constant $\beta_{FD} = 10^{-4}$ works well for all the experiments. Finally, for the DSM we set $\gamma = 0.01$.

The performance is evaluated using the averaged Mean Square Error ($\overline{MSE}$) and the standard deviation ($\overline{\sigma}$) [31] in the third-octave band and within the desired frequency range:

$$\overline{MSE} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \left(\frac{\sum_{\omega=\omega_l}^{\omega_h} \left(|\tilde{H}_{m,1/3}(\omega)| - |H_{des,1/3}(\omega)| \right)^2}{\omega_h - \omega_l} \right) \tag{28}$$

The average standard deviation $\overline{\sigma}$ is calculated as:

$$\overline{\sigma} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \sigma_m \tag{29}$$

where σ_m is the standard deviation of m-th microphone:

$$\sigma_m = \sqrt{\frac{1}{\omega_h - \omega_l + 1} \sum_{\omega=\omega_l}^{\omega_h} (10 \cdot log_{10}|\tilde{H}_{m,1/3}(\omega)| - D)^2} \tag{30}$$

$$D = \frac{1}{\omega_h - \omega_l + 1} \sum_{\omega=\omega_l}^{\omega_h} (10 \cdot log_{10}|\tilde{H}_{m,1/3}(\omega)|) \tag{31}$$

In addition, an online companion page with binaural audio renderings of the equalized and non-equalized audio is provided[1].

Before performing the optimization, pre-processing is performed. To achieve the peaks of the impulsive responses at the same time instant, the delay of each speaker is automatically

determined by finding the instant in which its direct sound reaches the reference microphone. Then we determine the offset gain to be added in the optimization, so as to normalize the output frequency responses to $0 \, \text{dB}$, by calculating the average one-third octave band difference between the frequency response and the desired frequency response on the reference microphone.

The number and dimension of the layers of a neural network impact significantly the performance. Therefore, a preliminary random hyperparameter search has been conducted to find a suitable network architectures that have been, thus, retained for all the experiments. In that regard, the BiasNet has two main configuration dimensions: the number of layers and their size, while the CFN and the CNN have more degrees of freedom: the convolutional layer number, horizontal and vertical size and pooling, the fully connected layer number and size. The activation function used in the network was found to be a less sensitive parameter. We adopted the sine activation function, which was recently proposed in [44] and shows to be well suited for optimization as it avoids local minima during network optimization. This activation function, behaves well for backpropagation as its derivatives do not vanish.

We tested 30 BiasNet configurations varying the number of layers from 1 to 10 and the number of neurons from 16 to 4096. In Table II the configurations that have obtained the best performance in preliminary tests are presented. For the CNN and CFN we tested 1 or 2 convolutional layers: the number of kernels equals 25 in the case of 1 convolutional layer, 48 and 24 or 100 and 10 in the case of 2 layers, respectively. The dimension of the kernel for the first and second convolutional layer equals $\mathcal{M} \times 1$ and $1 \times \mathcal{S}$, respectively. The number of hidden layers varied from 1 to 4, with the number of neurons ranging from 32 to 1024.

In all the experiments the Adam optimizer [45], was employed using a learning rate of 10^{-4}, and parameters $\beta_1 = 0.9, \beta_2 = 0.999$. The number of iterations was set to 10,000 and the weights and bias have been initialized with a uniform distribution.

Deep optimization methods were implemented in *Python* using *Tensorflow*[2] 2.0.0, while DSM and FD were implemented in *Matlab*. The experiments were performed using a machine with an Intel i7 processor, $32 \, \text{GB}$ of RAM and an Nvidia Titan GPU with $12 \, \text{GB}$ of dedicated RAM.

VI. RESULTS

A. Neural Architectures Comparison

We first consider the three neural networks (BiasNet, CFN and CNN) for equalization in the room scenario. We conduct experiments to find the best network hyperparameters, giving priority to the MIMO case. The same networks will be tested for SISO and MISO cases without a further hyperparameters search, for the sake of conciseness. Results are reported in Table II. As can be seen, many architectures in the test provide similar performance ($\overline{MSE} \sim 10^{-5}$), however the difference in the number of trainable parameters is extremely

---

[1] https://gitlab.com/a3labPapers/CompanionFiles/-/tree/master/TASLP-Multipoint_Audio_Equalization_Parametric_IIR_Filter.

[2] www.tensorflow.org

TABLE II
Preliminary test comparing several neural networks in the MIMO configuration for the room scenario. The number of neurons for each hidden layer is shown in round brackets in the Layers column.

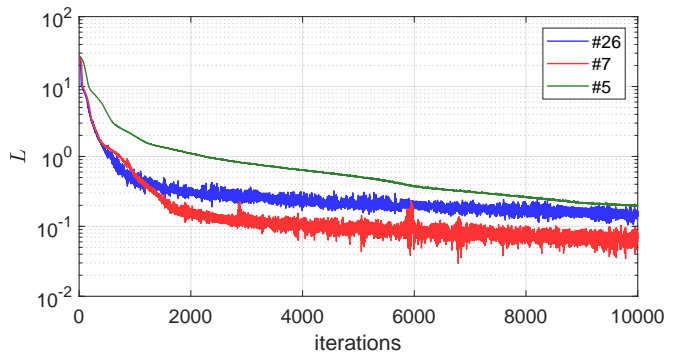| Architecture	$\overline{MSE}$	Layers	No. Learnable Parameters
BiasNet #26	$1.18 \cdot 10^{-5}$	(1024, 512, 256, 128)	758,784
BiasNet #5	$1.19 \cdot 10^{-5}$	(256)	137,728
BiasNet #9	$1.29 \cdot 10^{-5}$	(128)	68,864
CFN	$1.38 \cdot 10^{-5}$	best CFN[3]	4,673,390
BiasNet #7	$1.46 \cdot 10^{-5}$	(256, 256, 256)	268,800
CNN	$1.66 \cdot 10^{-5}$	same as [31]	2,369,390
BiasNet #25	$1.68 \cdot 10^{-4}$	(16,32,64,128,256,512, 256,128,64,32,16)	357,792
BiasNet #8	$4.24 \cdot 10^{-4}$	(64)	34,432
BiasNet #10	$6.36 \cdot 10^{-4}$	(32)	17,216
BiasNet #4L	$3.80 \cdot 10^{-1}$	out only (with bias)	536
BiasNet #4	$9.56 \cdot 10^{-1}$	out only (with bias)	536
No EQ	0.377	-	-

Fig. 6: Minimization of the loss function for configurations #26 (blue line), #7 (red line), #5 (green line) for MIMO equalization in the room scenario. The optimization speedup is directly related to the number of trainable parameters. It must be noted that a lower loss does not mean a lower result as the loss includes the regularization term and is computed as the root of squares, while the MSE is computed as the average of squared terms.

large. Taking the CNN as a reference, the best CFN achieves slightly better performance at the cost of an order of magnitude more parameters. On the other hand, the BiasNet excels in the test with a significantly lower number of parameters. Configuration #5, has only 17,664 trainable parameters, but achieves almost identical performance as configuration #26 which is 40 times larger to train. However, there is a critical number of trainable parameters under which the performance decreases: configuration #9 is similar to #26, besides having half the neurons in its only layer, but closely follows the performance of #26. On the contrary, progressively halving the neurons of the only layer (see configurations #8 and #10) makes the performance fall. The worst case is achieved with configuration #4 and #4L, i.e. with direct optimization, with an equalization performance which is orders of magnitude inferior to the best networks. In these preliminary tests we also considered the ℓ_∞-norm as a replacement for the ℓ_2-norm in the computation of the L_1 loss term, however, the results achieved with that loss were lower than those seen in Table II. The best result was $\overline{MSE} = 1.1 \cdot 10^{-2}$, $\overline{\sigma} = 0.38$.

Although several networks obtain almost identical performance in terms of performance, it is worth considering the optimization time achieved by networks with very different number of trainable parameters. The optimization time depends on two factors: the number of iterations to reach a target goal and the time required by each iteration. The latter, in our case, mostly depends on: *(a)* the time required for computing of the neural network graph and its backpropagation; *(b)* the time required for computing the filter coefficients, and simulating the room. When neural networks run on a GPU, the latter can be quite expensive, especially for the MIMO case, and adds a fixed cost that is not negligible, constraining the overall optimization time to be of the order of several minutes, with the current hardware and software implementation. For this reason, it is worth aiming at a reduction of the number of iterations to reach a desired goal (a target $\overline{MSE}$ or

[3]Note: the CFN was the best among all the tested CFN and is composed of 2 convolutional layer of 100 and 10 kernels, respectively, 3 dense layers of 64 neurons each.

a substantial stop in the loss descent). We trace the loss descent during optimization, plotted in Fig. 6. As we can see, with BiasNet, the number of parameters directly relates to a speedup of the optimization process.

For the reasons outlined above among the tested networks we select the first in Table II, that provides the best performance and fast optimization times.

B. Room Scenario

This section describes results for SISO, MISO and MIMO equalization in the room scenario, considering the BiasNet and the other comparative methods. The SISO results are presented in Table III, showing that the neural approach achieves better results than the baseline techniques by one or more orders of magnitude, without the need for refining the neural network size and hyperparameters. It is worth noting that the DSM is unable to improve much on the non-equalized case given the same number of iterations of the DO approach. The FD approach achieves excellent performance, however its (convex) optimum does not coincide with the (non-convex) optimal solution found by the BiasNet, which is superior. Furthermore, the computational cost of a 8192-th order FIR filter is very high. Other methods have been considered and are reported here, for the sake of completeness. Prony's method allows the optimization of IIR filters coefficients [11]. To the best of our knowledge, it has not been extended to the MIMO case, therefore we conducted tests in the SISO case only. Another method that was proposed for the SISO case is the neural end-to-end equalization from Ramìrez et al. [9]. Prony's method obtains $\overline{MSE} = 1.38 \cdot 10^{-1}$ and $\overline{\sigma} = 1.65$ with the same number of SOS used for our method. Ramìrez's method obtains $\overline{MSE} = 1.29 \cdot 10^{-1}$ and $\overline{\sigma} = 1.52$.

MISO results are presented in Table IV. In this table we report the results at the listening point used for optimization (right microphone), as well as the average of the $\overline{MSE}$

TABLE III
Results for SISO equalization, room scenario.

Method	MSE	σ
BiasNet	$\mathbf{1.32 \cdot 10^{-5}}$	$\mathbf{1.58 \cdot 10^{-2}}$
FD_{8192}	$2.08 \cdot 10^{-4}$	$5.54 \cdot 10^{-2}$
FD_{1024}	$9.74 \cdot 10^{-3}$	$4.36 \cdot 10^{-1}$
DSM	$3.43 \cdot 10^{-2}$	$9.92 \cdot 10^{-1}$
No EQ	$3.80 \cdot 10^{-1}$	1.69

TABLE IV
Results for MISO equalization, room scenario evaluated at the listening point used during optimization (Right Mic) and both microphones (L+R Mic).

Method	Right Mic		L+R Mic	
	MSE	σ	$\overline{MSE}$	$\overline{\sigma}$
BiasNet	$8.95 \cdot 10^{-6}$	$1.26 \cdot 10^{-2}$	$\mathbf{5.53 \cdot 10^{-2}}$	$\mathbf{8.04 \cdot 10^{-1}}$
DSM	$1.87 \cdot 10^{-2}$	$6.98 \cdot 10^{-1}$	$8.76 \cdot 10^{-2}$	1.42
FD_{1024}	$5.26 \cdot 10^{-2}$	$3.55 \cdot 10^{-1}$	$1.63 \cdot 10^{-1}$	$9.05 \cdot 10^{-1}$
FD_{8192}	$\mathbf{4.88 \cdot 10^{-6}}$	$\mathbf{9.58 \cdot 10^{-3}}$	$1.30 \cdot 10^{-1}$	$9.52 \cdot 10^{-1}$
No EQ	$3.92 \cdot 10^{-1}$	1.91	$3.77 \cdot 10^{-1}$	1.99

calculated at both listening positions. As can be seen, the performance obtained in the former case is extremely high, showing that an increase in the number of loudspeakers improves the equalization performance with respect to the SISO case. However, considering only one listening position for optimization provides a solution that does not work well for the other listening position, therefore the performance of the best methods in the table drops by at least 4 orders of magnitude when evaluating the performance at both microphones. It is worth noting that in this case the FD_{1024} method is inferior to the IIR filters provided by DSM, while the FD_{8192} slightly improves over the proposed method. However, the solution found by FD_{8192} suffers a more sever degradation of performance when the error is computed at both listening positions.

Finally, Table V shows that the proposed method scores optimally. All methods fail to achieve results as good as those computed for the MISO case (right microphone), since the computed solution must fit two listening positions. This is expected, since increasing the number of microphones $\mathcal{M}$ increases the complexity of the problem.

As an example of the frequency response that these method achieve, we show some of the magnitude responses in Fig. 8. The difference between the unequalized response (red line) and the equalized response (blue line) is evident. While the unequalized spectrum exhibits an excursion of more than 10 dB, the equalized response is flat in the frequency range covered by the loudspeakers (vertical dashed lines).

As discussed previously, the equalization process should provide the desired spectral profile without altering the energy provided by the individual speakers. Fig. 7 shows that the energy of each speaker is preserved before and after the equalization process.

TABLE V
Results for MIMO equalization, room scenario.

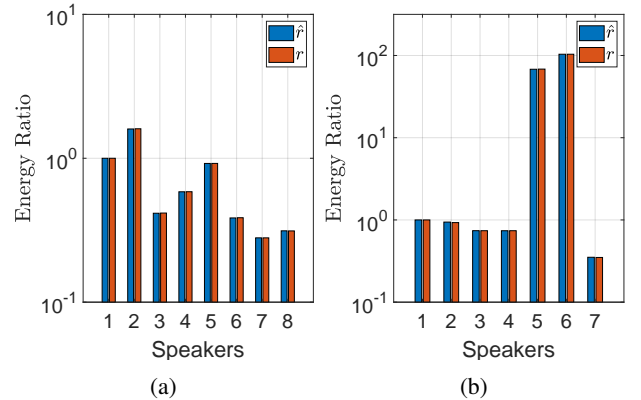| Method	$\overline{MSE}$	$\overline{\sigma}$
BiasNet	$\mathbf{1.18 \cdot 10^{-5}}$	$\mathbf{1.40 \cdot 10^{-2}}$
DSM	$2.54 \cdot 10^{-2}$	$7.18 \cdot 10^{-1}$
FD_{1024}	$4.57 \cdot 10^{-2}$	$4.36 \cdot 10^{-1}$
FD_{8192}	$1.38 \cdot 10^{-5}$	$1.57 \cdot 10^{-2}$
No EQ	$3.77 \cdot 10^{-1}$	1.99

Fig. 7: Bar graph of energy ratio after ($\hat{r}$) and before r the optimization for the room scenario (a) and the car scenario (b). Speakers 5 and 6 in (b) are woofer and subwoofer, therefore have larger energy.

C. Car Scenario

Results for the car cabin scenario are reported in Table VI. Since the car provides a more challenging scenario, the $\overline{MSE}$ are higher than that of the room and the DSM fails to provide a decent equalization performance. The FD method does not match the performance achieved by the proposed method, which as in the previous experiments obtains the best results.

In Fig. 9 one third octave band frequency responses are presented: at high frequencies the amplitude responses are flat, both for the overlapping of the loudspeakers frequency responses and because the network optimizes in this range only two loudspeakers out of the seven installed in the car. At low frequencies the network has not been able to optimize as well as at high frequencies, indeed it presents a maximum deviation of 5 dB around 40 Hz.

This scenario makes a more challenging testbed for our method: the loudspeakers are not arranged in a regular pattern, their frequency ranges are different and the internal cabin volume is irregular. Despite these difficulties, the equalization

TABLE VI
Results for MIMO equalization, car cabin scenario.

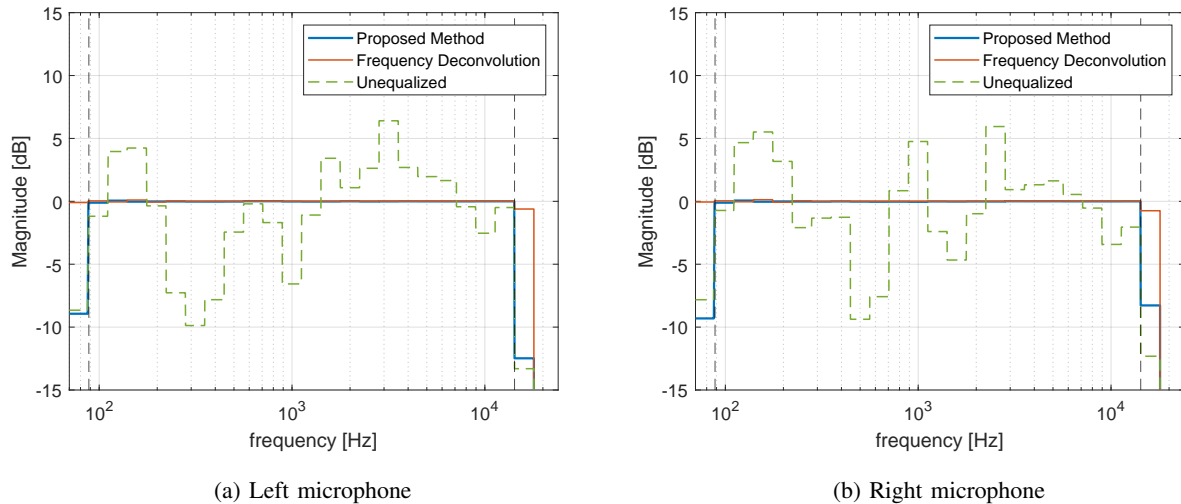| Method	$\overline{MSE}$	$\overline{\sigma}$
BiasNet	$\mathbf{5.74 \cdot 10^{-3}}$	$\mathbf{1.83 \cdot 10^{-1}}$
DSM	3.62	2.76
FD_{1024}	$4.22 \cdot 10^{-2}$	$8.15 \cdot 10^{-1}$
FD_{8192}	$1.84 \cdot 10^{-2}$	$5.02 \cdot 10^{-1}$
No Eq	13.47	3.16

(a) Left microphone

(b) Right microphone

Fig. 8: One-third octave band magnitude response of (a) left and (b) right microphone in the room scenario. Green line is the unequalized frequency response, the blue line is the equalized one when the proposed method is used to optimized Parametric IIR filters, red line is the equalized frequency response when FIR filters optimized with the FD are used and the black dotted lines refer to the minimum and maximum frequency to be equalized.

(a) Left microphone
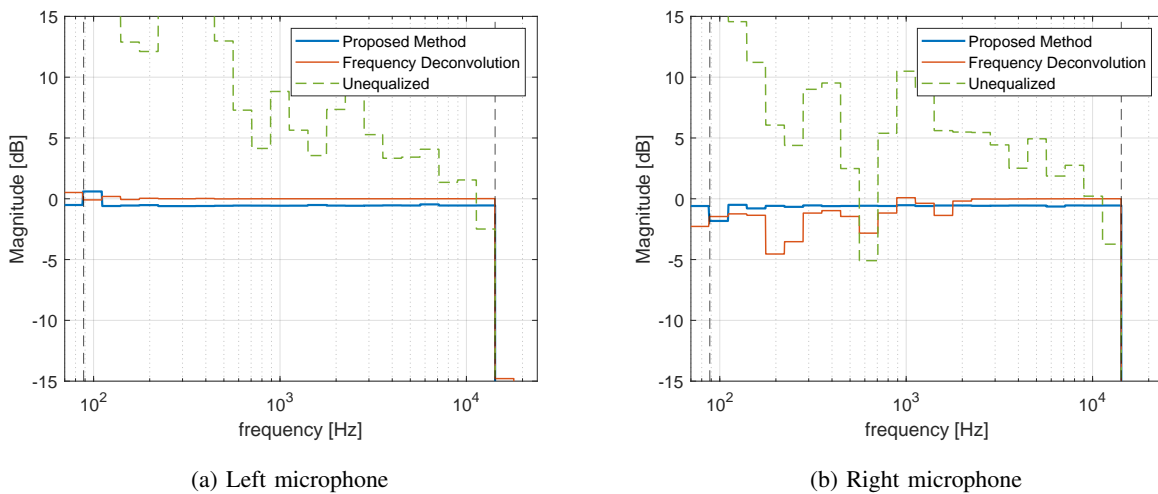
(b) Right microphone

Fig. 9: One-third octave band magnitude response of the measured signal at the reference microphones: (a) left and (b) right microphone in the car's cabin scenario. Green line is the unequalized frequency response, the blue line is the equalized one when the proposed method is used to optimized Parametric IIR filters, red line is the equalized frequency response when FIR filters optimized with the FD are used and the black dotted lines refer to the minimum and maximum frequency to be equalized.

is superior to the FD method, which is usually considered the optimal method for room equalization, and the energy of the loudspeakers signals is preserved.

VII. REMARKS

Our experiments show that the proposed method is comparable or slightly better than the FD_{8192} method in terms of performance, however, the computational cost and the phase properties of the two differ.

A. Computational Cost

The computational cost of each of the S equalizers is 16383 floating point operations per sample in the FD_{8192} case. The IIR equalizers are composed of 22 SOS's in the room scenario and 29 SOS's in the car scenario, totalling 198 and 261 operations per sample respectively, which means a reduction of the cost of nearly two orders of magnitude. The cost grows further with an increase of the loudspeakers, e.g. in the MIMO room scenario the proposed method requires 1584 operations against 131064 operations for the FD_{8192}.

Clearly, it must be remarked that the FD method allows to design shorter filters. With FIR filters of order 1024 the performance is significantly lower but still acceptable, yielding an improvement of approximately one order of magnitude with respect to the non equalized case. However, the number

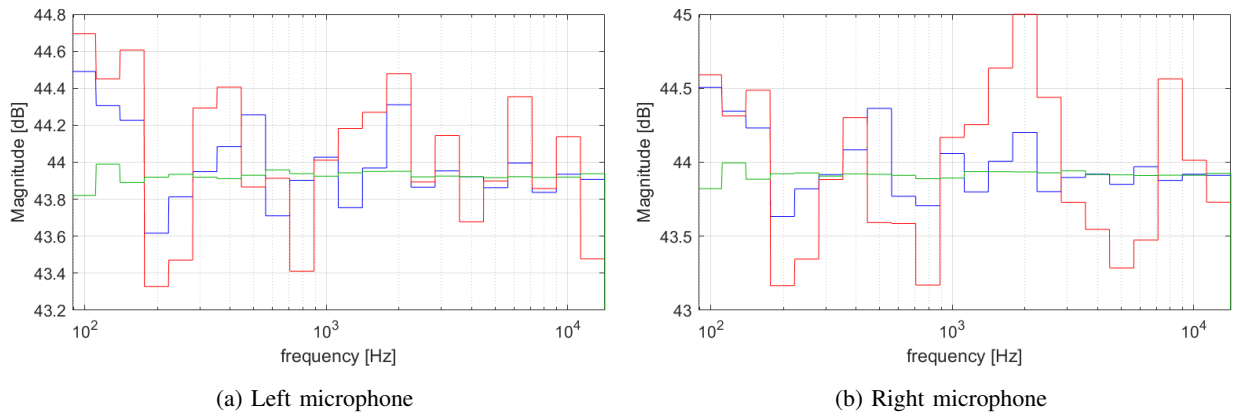(a) Left microphone   (b) Right microphone

Fig. 10: One-third octave band magnitude response of the measured signal at the reference microphones: (a) left and (b) right microphone in the room scenario. The green line is the equalized magnitude response depicted in Fig. 8 which is computed by filtering a discrete-time impulse and transforming to the time domain. The blue line is simulated using white noise as input, while the red line is measured in the real scenario using white noise. Please note: the magnitude range is only 2 dB to emphasize the small differences.

of operations per filter is still significantly larger than the proposed method (2047 operations per sample).

B. Real-World Validation

The evaluations provided above are referred to a simulated room, employing the RIRs. However, it is worth validating the experiments in the real scenario, to assess whether the simulations are accurate enough and the results are reproducible in a real environment.

The IIR filters designed for the room scenario in the MIMO case are taken to the real room and applied for equalization. More specifically, we load the designed IIR filters on a Simulink[4] patch to preprocess the signal and feed the loudspeakers. The hardware setup is the same as described in Section V. We measure the frequency response of the environment by reproducing white noise and compare these to the simulated magnitude responses obtained above. These are shown in Fig. 10, where the red line is the measured one and the green line is the ideal one (obtained by filtering a discrete-time impulse sequence). The observed deviation is at most 2 dB, but is inherent to the use of white noise as the input signal. Indeed, by computing the magnitude response of the room using white noise in the simulated computer environment, we obtain similarly random deviations from the flat band (blue line). Therefore we can conclude that the IIR filters designed in the simulated environment are suitable for the real scenario.

VIII. CONCLUSIONS

In this paper, we described an IIR parametric equalizer design for automatic room equalization at multiple listening points, exploiting a Deep Optimization method based on a spectral loss function and a regularization term which avoids energy unbalance between the speakers. The error is backpropagated to update the neural network weights. We also introduced BiasNet, a novel neural network architecture specifically

---

designed for Deep Optimization that requires no input. Its behavior is compared to other neural network architectures. We also compared BiasNet with other two baseline techniques, one based on FIR filters for frequency deconvolution and another based on a iterative approach for IIR filter design. Compared to the baseline techniques, our method achieves better performance at a remarkably lower computational cost. We also tested the designed IIR filters in a real scenario, showing almost no difference to the simulated one.

The proposed method works offline and assumes a linear environment, therefore other factors, such as the movement of the listening positions are not considered. Adaptive filtering techniques could be introduced to provide online equalization starting from the filters obtained with the proposed approach. In the future we plan to perform subjective tests to assess the validity of the approach and devise other metrics based on psychoacoustics, for the optimization.

Further studies will address the use of end-to-end deep learning techniques for audio equalization, with an analysis of the real-time factor on an embedded system with tensor processing units (TPU) and general purpose processors. Another aspect worth investigating is the subjective response to these equalization systems, which will be done by conducting listening tests.

REFERENCES

[1] V. Välimäki and J. D. Reiss, "All about audio equalization: Solutions and frontiers," *Applied Sciences*, vol. 6, no. 5, 2016.
[2] J. D. Reiss, "Design of audio parametric equalizer filters directly in the digital domain," *Trans. Audio, Speech and Lang. Proc.*, vol. 19, no. 6, p. 18431848, August 2011.
[3] M. Karjalainen, T. Paatero, J. N. Mourjopoulos, and P. D. Hatziantoniou, "About room response equalization and dereverberation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 183–186.
[4] S. Cecchi, A. Carini, and S. Spors, "Room response equalization - a review," *Applied Sciences*, vol. 8, no. 1, 2018.
[5] Y. Soeta and Y. Sakamoto, "An exploratory analysis of sound field characteristics using the impulse response in a car cabin," *Environments*, vol. 5, no. 4, 2018.

---

[4]https://www.mathworks.com/products/simulink.html

[6] H. B. Huang, J. H. Wu, X. R. Huang, M. L. Yang, and W. P. Ding, "The development of a deep neural network and its application to evaluating the interior sound quality of pure electric vehicles," *Mechanical Systems and Signal Processing*, vol. 120, pp. 98 – 116, 2019.

[7] S. Cecchi, L. Palestini, P. Peretti, F. Piazza, F. Bettarelli, and R. Toppi, "Automotive audio equalization," *Journal of the Audio Engineering Society*, June 2009.

[8] P. Chang, C. G. Lin, and B. Yeh, "Inverse filtering of a loudspeaker and room acoustics using time-delay neural networks," *The Journal of the Acoustical Society of America*, vol. 95, no. 6, pp. 3400–3408, 1994.

[9] M. A. Martinez Ramirez and J. D. Reiss, "End-to-end equalization with convolutional neural networks," in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18), Aveiro, Portugal*, 2018.

[10] N. Agrawal, A. Kumar, V. Bajaj, and G. Singh, "Design of digital IIR filter: A research survey," *Applied Acoustics*, vol. 172, p. 107609, 2021.

[11] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, "Frequency-warped signal processing for audio applications," *Journal of the Audio Engineering Society*, vol. 48, no. 11, pp. 1011–1031, November 2000.

[12] S. Lang and J. McClellan, "Frequency estimation with maximum entropy spectral estimators," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 6, pp. 716–724, 1980.

[13] D. J. Krusienski and W. K. Jenkins, "Adaptive filtering via particle swarm optimization," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 1, 2003, pp. 571–575.

[14] S. K. Saha, R. Kar, D. Mandal, and S. Ghoshal, "Gravitation search algorithm: Application to the optimal IIRfilter design," *Journal of King Saud University - Engineering Sciences*, vol. 26, no. 1, pp. 69 – 81, 2014.

[15] P. Dodds, "A flexible numerical optimization approach to the design of biquad filter cascades," *Journal of the Audio Engineering Society*, October 2020.

[16] A. Jiang and H. K. Kwan, "Minimax design of IIR digital filters using iterative SOCP," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 6, pp. 1326–1337, 2010.

[17] A. N. Sloss and S. Gustafson, *2019 Evolutionary Algorithms Review*. Cham: Springer International Publishing, 2020, pp. 307–344.

[18] F. Serbet, T. Kaya, and M. T. Ozdemir, "Design of digital IIR filter using particle swarm optimization," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017, pp. 202–204.

[19] R. C. Nongpiur, D. J. Shpak, and A. Antoniou, "Improved design method for nearly linear-phase IIR filters using constrained optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 895–906, 2013.

[20] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *International Conference on Learning Representations*, 2020.

[21] B. Kuznetsov, J. D. Parker, and F. Esqueda, "Differentiable IIR filters for machine learning applications," in *Digital Audio Effects (DAFx) 2020, Vienna, Austria*, September 2020.

[22] P. Bhattacharya, P. Nowak, and U. Zölzer, "Optimization of cascaded parametric peak and shelving filters with backpropagation algorithm," in *Digital Audio Effects (DAFx) 2020, Vienna, Austria*, September 2020.

[23] S. Nercessian, "Neural parametric equalizer matching using differentiable biquads," in *Digital Audio Effects (DAFx) 2020, Vienna, Austria*, September 2020.

[24] A. D. Back and A. C. Tsoi, "FIR and IIR synapses, a new neural network architecture for time series modeling," *Neural Computation*, vol. 3, no. 3, pp. 375–385, 1991.

[25] N. Allakhverdiyeva, "Application of neural network for digital recursive filter design," in *2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)*, October 2016, pp. 1–4.

[26] V. Välimäki and J. Rämö, "Neurally controlled graphic equalizer," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–1, 2019.

[27] J. Rämö and V. Välimäki, "Neural third-octave graphic equalizer," in *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19), Birmingham, UK*, September 2019, pp. 1–6.

[28] S. Cecchi, A. Primavera, F. Piazza, and A. Carini, "An adaptive multiple position room response equalizer," in *2011 19th European Signal Processing Conference*, 2011, pp. 1274–1278.

[29] H.-N. Kim and W.-J. Song, "An adaptive IIR equalizer for nonminimum-phase channels," in *ICSP '98. 1998 Fourth International Conference on Signal Processing (Cat. No.98TH8344)*, 1998, pp. 441–444 vol.1.

[30] G. Pepe, L. Gabrielli, S. Squartini, and L. Cattani, "Evolutionary tuning of filters coefficients for binaural audio equalization," *Applied Acoustics*, vol. 163, p. 107204, 2020.

[31] ——, "Designing audio equalization filters by deep neural networks," *Applied Sciences*, vol. 10, no. 7, p. 2483, 2020.

[32] O. Kirkeby, P. A. Nelson, H. Hamada, and F. Orduna-Bustamante, "Fast deconvolution of multichannel systems using regularization," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 189–194, March 1998.

[33] G. Ramos and J. J. Lpez, "Filter design method for loudspeaker equalization based on IIR parametric filters," *Journal of the Audio Engineering Society*, vol. 54, no. 12, pp. 1162–1178, December 2006.

[34] H. Behrends, A. Von dem Knesebeck, W. Bradinal, P. Neumann, and U. Zlzer, "Automatic equalization using parametric IIR filters," *Journal of the Audio Engineering Society*, vol. 59, no. 3, pp. 102–109, March 2011.

[35] G. Pepe, L. Gabrielli, S. Squartini, L. Cattani, and C. Tripodi, "Gravitational search algorithm for IIR filter-based audio equalization," in *2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam*, 2020.

[36] U. Zölzer, *DAFX: digital audio effects*. John Wiley & Sons, 2011.

[37] D. Lopez-Paz and L. Sagun, "Easing non-convex optimization with neural networks," in *International Conference on Learning Representations (ICLR 2018)*, 2018.

[38] A. V. Oppenheim and R. V. Schafer, *Discrete-time Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.

[39] H. Caracalla and A. Roebel, "Gradient conversion between time and frequency domains using wirtinger calculus," in *Digital Audio Effects (DAFx) 2017, Edinburgh, United Kingdom*, September 2017.

[40] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *J. ACM*, vol. 8, no. 2, p. 212229, April 1961.

[41] R. M. Lewis, V. Torczon, and M. W. Trosset, "Direct search methods: then and now," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1, pp. 191 – 207, 2000.

[42] A. Farina, "Advancements in impulse response measurements by sine sweeps," *Journal of the Audio Engineering Society*, May 2007.

[43] S. Campanini and A. Farina, "A new Audacity feature: room objective acoustical parameters calculation module," *Linux Audio Conference, Parma, Italy, paper B*, vol. 2, 2009.

[44] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," vol. 33, 2020.

[45] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *3rd International Conference for Learning Representations, San Diego*, 2015.

Giovanni Pepe Giovanni Pepe was born in Benevento (BN), Italy in 1992. He gained the Bachelor degree in Electronic Engineering for Automation and Telecommunication at Universit degli Studi del Sannio, Benevento, in 2015. In 2018 he gained the Masters degree (summa cum laude) in Electronic Engineering at Universit Politecnica delle Marche, Ancona. He is currently a PhD student in Information Engineering since November 2018. The main research is related to machine learning for audio processing.

Leonardo Gabrielli Leonardo Gabrielli got his M.Sc. and PhD degrees in Electronics Engineering from Universit Politecnica delle Marche, Italy, respectively in 2011 and 2015. His main research topics are related to audio signal processing and machine learning with application to sound synthesis, Computational Sound Design, Networked Music Performance and audio classification. He has been co-founder of DowSee srl, he holds several industrial patents and developed several industrial products. He is coauthor of more than 50 scientific papers and two books.

Stefano Squartini Stefano Squartini (Senior Member, IEEE) was born in Ancona, Italy, on March 1976. He received the Italian Laurea with honors in electronic engineering and the Ph.D. degree in electronics and telecommunications from the Polytechnic University of Marche (UnivPM), Ancona, Italy, in 2002 and 2005, respectively. He joined the Department of Information Engineering as an Assistant Professor in circuit theory in 2007. He became Full Professor at UnivPM on May 2020. He is author or coauthor of more than 220 international scientific papers. His current research interests include the area of computational intelligence and digital signal processing, with special focus on digital audio and energy management in smart grids. Prof. Squartini is Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He joined the Organizing and the Technical Program Committees of more than 100 International Conferences and Workshop.

Nicol Strozzi Nicol Strozzi was born in Reggio Emilia, Italy, in 1988. He studied Telecommunication Engineering at the University of Parma, Italy, where he obtained his MSc in 2015 and PhD in Information Technologies in 2019. Since February 2019 he has been working in ASK Industries JVC Kenwood Group as algorithm designer. His major research field includes localization technologies, audio signal processing, V2X and 5G communications.
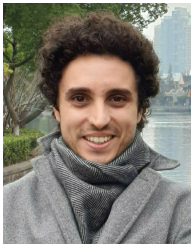
Carlo Tripodi Carlo Tripodi received the Ph.D. degree in Information Technology in 2016 from the University of Parma (Italy). He was associate researcher in the same institute, working on adaptive signal processing and information theory until March 2017, when he joined the R&D department of ASK Industries Spa. Here he was core developer of Active Noise Cancellation algorithms for the automotive industry. Since 2020 he is responsible for the management of innovative audio projects for automotive systems.