applied sciences

MDPI

Article

Designing Audio Equalization Filters by Deep Neural Networks

Giovanni Pepe [1], Leonardo Gabrielli [1,*], Stefano Squartini [1] and Luca Cattani [2]

[1] Department of Information Engineering, Università Politecnica delle Marche, 60131 Ancona, Italy; g.pepe@pm.univpm.it (G.P.); s.squartini@univpm.it (S.S.)

[2] ASK Industries SpA, 42124 Reggio Emilia, Italy; cattaniL@askgroup.it

* Correspondence: l.gabrielli@univpm.it

Received: 26 February 2020; Accepted: 31 March 2020; Published: 4 April 2020

Abstract: Audio equalization is an active research topic aiming at improving the audio quality of a loudspeaker system by correcting the overall frequency response using linear filters. The estimation of their coefficients is not an easy task, especially in binaural and multipoint scenarios, due to the contribution of multiple impulse responses to each listening point. This paper presents a deep learning approach for tuning filter coefficients employing three different neural networks architectures—the Multilayer Perceptron, the Convolutional Neural Network, and the Convolutional Autoencoder. Suitable loss functions are proposed for each architecture, and are formulated in terms of spectral Euclidean distance. The experiments were conducted in the automotive scenario, considering several loudspeakers and microphones. The obtained results show that deep learning techniques give superior performance compared to baseline methods, achieving almost flat magnitude frequency response.

Keywords: deep neural networks; FIR filter design; audio equalization; automotive audio

1. Introduction

Listening environments are characterized by reflections and reverberations that can adversely affect listening [1] and attention [2], adding unwanted artifacts to the sound produced by an acoustic source. For this reason, audio equalization is needed in order to improve sound quality reproduction. Of particular interest is the car scenario, where people daily listen to music, radio programs or take hands-free phone calls. The audio quality in such an environment is very important, but is adversely affected by several factors, including the loudspeakers quality and the reflective materials inside the cabin. The impulse response at the listening position is characterized by the sum of multiple signals: those coming from the loudspeakers and their reflections. Furthermore, the loudspeakers impose their frequency response on the signal. The frequency response is thus colored and usually results in deep notches and peaks, that reduce the audio quality.

These issues are generally addressed by the design of linear filters [3] that are applied to the signal before being transduced by the loudspeakers. The filters are designed to improve the audio quality at specific listening positions by inverting the car impulse response. However, the task is challenging, as the existence of the inverse may not be guaranteed, and the complexity increases with the number of sources and microphones. A plethora of design techniques have been proposed in the past for room equalization [1], and some have been proposed, more specifically, for the car scenario. These are, generally, based on linear optimization and inversion algorithms. Considering the complexity of this task, however, novel design techniques may be applied in this context, relying on nonlinear methods such as evolutionary algorithms, machine learning and neural networks.

Since digital communication systems are subject to the multipath problem, that is, the sum of multiple reflections in a linear channel with multiple sources and receivers, it is worth investigating the literature for equalization techniques applied to this application field. Indeed, several novel techniques have been proposed for the design of equalizing filters for digital communications, relying on nonlinear methods. In Reference [4], the authors use Particle Swarm Optimization (PSO) to equalize the impulse response of an optical fiber communication. This is shown to provide better results than Least Mean Square (LMS) and Recursive Least Square (RLS) techniques. Another interesting PSO approach is reported in Reference [5], where the PSO particles are used to obtain optimal poles and zeros of an IIR filter. In Reference [6] Genetic Algorithms (GA) are exploited for Adaptive Channel Equalization, in order to reduce the Inter Symbol Interference (ISI) present in the trasmission channel.

Although inspiring, these algorithms cannot be employed in the audio equalization scenario, as the two tasks differ in several aspects. While in communication systems equalizers are implemented at the receiving end, in the audio case they can only be implemented at the sound source. This can represent an issue, as the equalizing filters must provide satisfying results at several listening positions, while with telecommunication devices, each one can adapt its equalizing filter depending on the incoming signal. In communication systems the main goal is reducing symbol error rate, thus, allowing a robust classification of the symbols constellation, while in the audio field the goal is to achieve near-perfect audio quality taking psychoacoustic factors into consideration. Finally, the communication scenario may or may not consider time-varying environmental conditions (e.g., mobile receiving stations), while in the audio field time-invariance is often assumed, thus, room impulse responses are measured and treated statically.

A first attempt at the use of deep learning for audio equalization is found in Reference [7], where the authors use a Time Delay Neural Network (TDNN) to solve the problem of equalization, using the input sequence, delayed by a time unit, as input and the signal recorded by the microphone as output: the error between the input signal and the output of the network is used for the back-propagation algorithm. The forward approach is also employed using a delayed copy of the input signal as input and the difference between the output given from the loudspeaker and the network as error. In References [8,9], the authors describe a system that maps the gain of each frequency band with the user's preferred equalizer settings as training data. A similar approach is undertaken in Reference [10], where k-Nearest Neighbour (KNN) is used to implement a timbre equalizer based on user preference in terms of brightness, darkness and smoothness. Specifically, sound professionals and music students were asked to manually equalize 41 audio segments. Equalization for music production were performed in Reference [11], where a Dilated Residual Network (DRN) was used to automate the resonance equalization in music, predicting the optimal attenuation factor, while an end-to-end equalization was used in Reference [12], substituting filter banks with Convolutional Neural Network (CNN) and without prior knowledge of filter parameters, like gains, cut-off frequency and quality factor.

Recent works addressing the design of IIR filters using PSO can be found in References [13–15]. In particular, Foresi et al. [15] use PSO with fractional derivative constraints to design a quasi-linear phase IIR filter for Audio Crossover Systems. The algorithm gives the parameters of the desired filter (like cut-off frequency) with a flat magnitude response and a linear phase. In Reference [16], the authors use Gravitational Search Algorithm (GSA) to model an IIR filter and a nonlinear rational filter, then they compare the technique with PSO and GA. In this case, the algorithms provide filter coefficients as outputs. Another approach is used in Reference [17], where the authors achieved an IIR filter using the Artificial Immune Algorithm and compared the results with GA, the Touring Ant Colony Optimization (TACO) and Tabu Search (TS).

Neural networks have been also proposed for the filter design task. In Reference [18] a neural network is devised to design an IIR filter with the error calculated as the difference in magnitude response between the desired and the generated filter. Kumari et al. [19] provide a performance comparison of some neural network architectures to design a low pass Finite Impulse Response

(FIR)filter, including Radial Basis Function (RBF), General Regression Neural Networks (GRNN), Radial Basis Exact (RBE), Back-Propagation Neural Network (BPNN) and the Multilayer Perceptron (MLP). Wang et al. [20] proposes a two step optimization Frequency-Response Masking (FRM) technique based on the design of a FRM filter optimizing the subfilters, further optimized by decomposing it into several linear neural networks.

In previous work from the same authors [21], evolutionary algorithms were employed for binaural audio equalization in the car cabin. PSO and GSA were tested, leading to an improvement with respect to baseline techniques. In this work we introduce a different approach, based on deep neural networks, with the aim of improving previous results and broadening the scope to multipoint equalization. To the best of our knowledge, no deep learning technique has been proposed in the literature to obtain filter coefficients for multipoint audio equalization. In this work we conduct the offline design of the filters coefficients exploiting deep neural networks trained according to a set of frequency-domain constraints. Three architectures are proposed and several experiments are conducted in two car cabins characterized by multiple impulse responses, comparing the results of the proposed method to the state of the art methods. The car scenario introduces different issues with respect to room equalization as the impact of early reflections and standing waves, caused by the peculiar geometry and the small size of the environment, are prominent [22,23].

The work is organized as follows: in Section 2 the problem is introduced. In Section 3 the proposed solution is explained, while in Section 4 the baseline methods are briefly described. Section 5 reports experimental conditions and Section 6 provides the results. Finally, in Section 7 conclusions are reported.

2. Problem Statement

Multi-point audio equalization is a very complex task: considering an environment with several sound sources and microphones, as depicted in Figure 1, a large number of impulse responses must be equalized, and the complexity of this problem increases with the number of sources $\mathcal{S}$ and microphones $\mathcal{M}$. Several optimization algorithms can be employed to generate filter coefficients able to obtain the desired frequency response at the microphone positions in a specific frequency range. In this work we compare our approach with state of the art methods to design the FIR filters offline. We assume, as in those works, that the listening environment is linear and time-invariant.

The generated FIR filters g_s, one for each sound source s, are employed for pre-processing the input signal x. The signal recorded at the m-th microphone is [24,25]:

$$y_m = \sum_{s=1}^{\mathcal{S}} h_{m,s} * (g_s * x) \quad m = 1..., \mathcal{M}. \tag{1}$$

The frequency response at the microphone is given by:

$$Y_m(\omega) = |\mathscr{F}(y_m)| \quad m = 1..., \mathcal{M}, \tag{2}$$

where $\mathscr{F}$ is the Fourier transform operator.

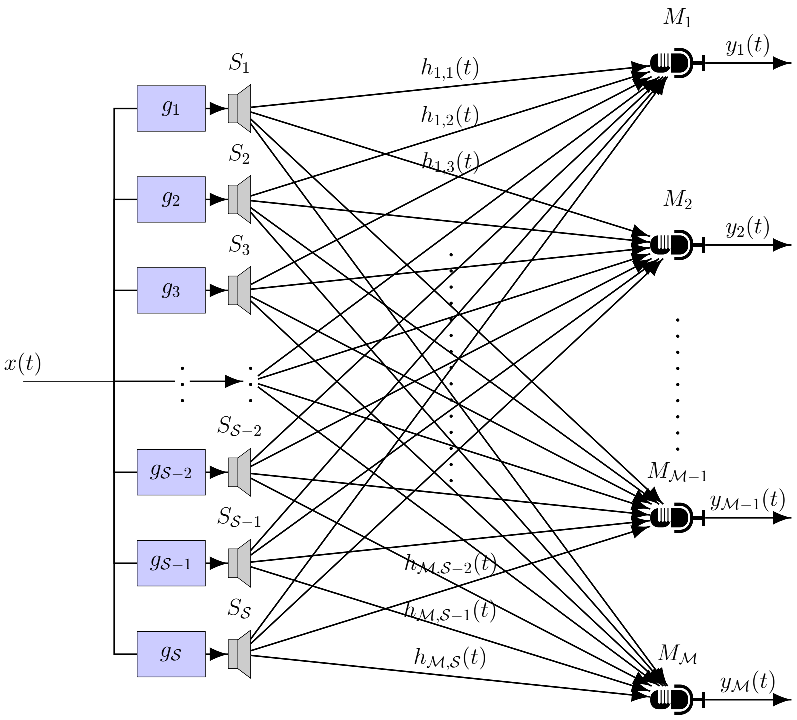


Figure 1. Multi-point equalization problem: $\mathcal{S}$ loudspeakers are displaced in an environment together with $\mathcal{M}$ microhones. The equalizing filters g_s are designed to invert the environment impulse responses $h_{m,s}$.

3. Proposed Method

The rationale behind the proposed approach stems from the following reflection—the training of a deep neural network is, in fact, an optimization problem, where a loss term is minimized by the back-propagation of the error through the neural network. This idea is not completely new and shallow neural network have been previously proposed for optimization (see, e.g., References [26,27]). Deep neural networks, however, were shown in Reference [28] to perform better in optimization tasks, possibly due to their parameter redundancy. In this work thus we propose to exploit deep neural networks for the optimization of equalizing filter coefficients.

Our approach consists of training a neural network by backpropagation in order to obtain, as output, optimal coefficients that minimize a frequency-domain loss. Each set of impulse responses requires a different training, meaning that the network is not expected to generalize, but rather perform optimization by fitting its weights, differently to common Deep Learning classification and regression tasks. We test a shallow network, that is, a Multilayer Perceptron (MLP), and two deep network architectures: a CNN and a convolutional Autoencoder (AE). In the absence of prior art, we feed the networks with the only available data, that is, the measured impulse responses. The neural networks, in turn, provide filter coefficients that are iteratively optimized to minimize a loss function. In the following we describe the architectures and the respective loss functions. In all cases the loss function contains at least one term based on the distance between the achieved frequency response and the desired curve. In our case, for simplicity, the desired curve is flat and the distance is computed in the frequency range $\omega_l : \omega_h$, to be defined according to the use case. In our work we use the Euclidean distance [29,30] to compute the distance, which was found to converge faster than the L_1-norm [31].

3.1. Multilayer Perceptron

The MLP is a shallow network composed of several fully-connected layers: one input, one or more hidden layers, and an output layer. The input is constrained to the number of samples in the impulse responses, that are concatenated in a long vector. Considering $\mathcal{S} \times \mathcal{M}$ impulse responses of length L, the input will have length $\mathcal{S} \times \mathcal{M} \times L$. The network produces a vector concatenating all the FIR coefficients, thus has size $\mathcal{S} \times \mathcal{T}$, where $\mathcal{T}$ is the number of taps for each filter. The architecture is shown in Figure 2.

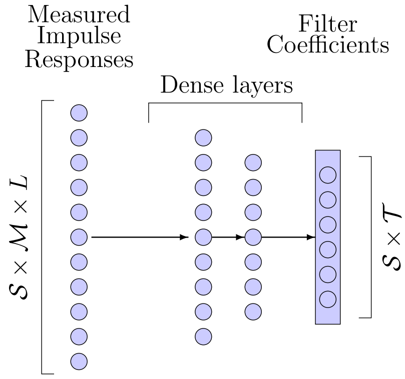

Measured Impulse Responses
Filter Coefficients
Dense layers
$\mathcal{S} \times \mathcal{M} \times L$
$\mathcal{S} \times \mathcal{T}$

Figure 2. Scheme of the proposed method using an Multilayer Perceptron (MLP). The impulse responses are all concatenated into a vector and fed to the first layer, which must have size $\mathcal{S} \times \mathcal{M} \times L$.

The loss function for the MLP is defined as the Euclidean distance between the given response at each iteration $|\tilde{Y}_m(\omega)|$, computed according to (2), and the desired frequency response:

$$J = \left(\sum_{m=1}^{\mathcal{M}} \left\| |\tilde{Y}_m(\omega)| - |Y_{des}(\omega)| \right\|_2 \right). \tag{3}$$

3.2. Convolutional Neural Networks

CNN are composed of a series of convolutional layers and a stack of fully connected layers [32]. The convolutional layers help reducing the dimensionality of the input and extract useful features for the fully connected layers. The input consists of a 3D matrix that stacks all the measured impulse responses, as shown in Figure 3. It is a tensor of size $\mathcal{S} \times \mathcal{M} \times L$. The last fully connected layer provides the filters coefficients and has, thus, length $\mathcal{S} \times \mathcal{T}$, as in the MLP. The loss function is the same as the one in (3).

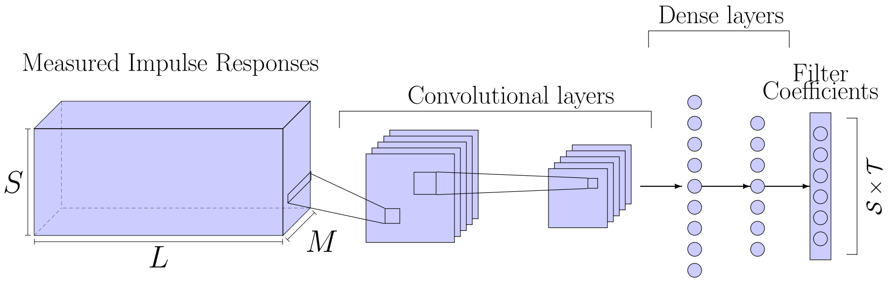


Figure 3. Scheme of the proposed method using a Convolutional Neural Network (CNN).

3.3. Autoencoder

An Autoencoder is a generative model [32] based on an encoder, a decoder and an internal representation that interconnects the two, often called latent space.

In our case, the encoder is composed of convolutional and fully connected layers, similar to the CNN of Section 3.2. The decoder performs the inverse mapping, thus, it is based on fully connected layers and de-convolutional layers. Filters coefficients are sampled from the internal representation, that has, thus, a size of $\mathcal{S} \times \mathcal{T}$. Impulse responses are used as input to the encoder, as shown in Figure 4.

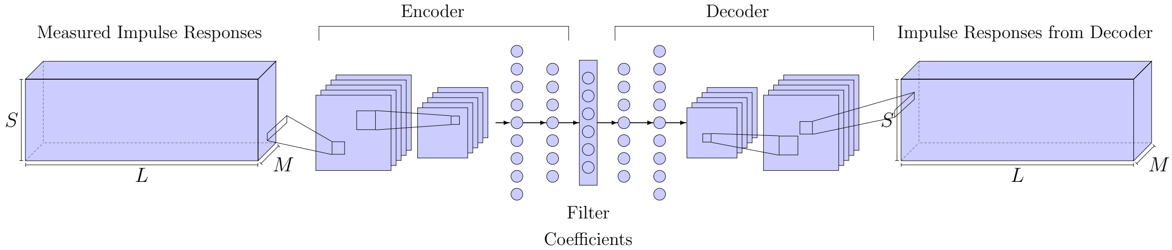


Figure 4. Scheme of the proposed method using Autoencoder.

The loss function for the autoencoder is defined as the sum of the Euclidean distance of Equation (3), and the reconstruction loss. The latter is expressed as the Euclidean distance between the input impulse response and the reconstructed one. Overall the loss for the autoencoder is:

$$J_{AE} \quad = \quad \left(\alpha \cdot \sum_{m=1}^{\mathcal{M}} \left\| |\tilde{Y}_m(\omega)| - |Y_{des}(\omega)| \right\|_2 + (1 - \alpha) \cdot \sum_{m=1}^{\mathcal{M}} \sum_{s=1}^{\mathcal{S}} \left\| \tilde{h}_{m,s}(n) - h_{m,s}(n) \right\|_2 \right). \quad (4)$$

The term α allows to weight the two losses, but for the rest of the paper it is kept equal to 0.5.

4. Baseline Methods

To compare the proposed approach, we have selected two methods from the literature, namely the Fast Deconvolution (FD) from Kirkeby et al. [33] and the Steepest Descent inverse filter design (SD) [24]. These approaches are described and motivated in the following.

4.1. Frequency Deconvolution Method

The fast deconvolution method is described in Reference [33], where deconvolution is performed in the frequency domain and made fast by taking advantage of the Fast Fourier Transform (FFT) algorithm. A matrix of optimal filters is computed in the frequency domain according to the following

$$\mathbf{G}(k) = [\mathbf{H}^H(k)\mathbf{H}(k) + \beta\mathbf{I}]^{-1}\mathbf{H}^H(k)\mathbf{A}(k), \quad (5)$$

where k is the FFT bin, $\mathbf{I}$ is the identity matrix, ^{H} is the Hermitian operator, $\mathbf{H}(k)$ contains the FFT of the impulse responses, $\mathbf{A}(k)$ contains the target frequency responses. The term β is an empirical regularization term that is necessary to avoid extreme peaks in the inverse filters, that would result in an excessive length of the filters in the frequency domain. Once $\mathbf{G}(k)$ is computed, its inverse FFT is computed and a circular shift of $K/2$ samples is performed, where K is the FFT size. This method is used to design $\mathcal{S}$ different filters, one per loudspeaker, for a fair comparison with our approach.

The FD method is expressed as a least-squares optimization problem in the frequency domain. The loss is:

$$J = \mathbf{e}^H\mathbf{e} + \beta\mathbf{v}^H\mathbf{v}, \quad (6)$$

where $\mathbf{e}$ is the error and $\beta\mathbf{v}^H\mathbf{v}$ is a regularization term meant as an effort penalty proportional to the total energy of the filtered input signals to the sources $\mathbf{v}$. The problem is, thus expressed as a convex optimization problem, where the squared error is minimized by a unique solution that is analytically found in Reference [33] by imposing the gradient of the loss function to zero.

There are important differences between the proposed method and FD. In our framework there are no assumptions on the convexity of the error surface. The gradient of the loss J in the proposed method is a nonlinear function. Specifically, our loss can be expressed as a function f of the magnitude frequency response of the impulse responses $\mathbf{h}$ and of the network output $\mathbf{g}$. The network output, that is, the filter coefficients $\mathbf{g}$, in turn, is a nonlinear function of the network weights θ and the network input $\mathbf{u}$ (i.e., the impulse responses, when not specified differently). In more rigorous terms:

$$J = f(|\mathscr{F}(\mathbf{g})|, |\mathscr{F}(\mathbf{h})|). \tag{7}$$

$$\mathbf{g} = \phi(\theta, \mathbf{u}). \tag{8}$$

4.2. Steepest Descent Method

FIR filters can be obtained by applying the Steepest Descent algorithm to audio equalization [24,25]. The first step of the algorithm consists in defining a target impulse response:

$$d = \underbrace{[0 \quad \ldots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]}_{L+\mathcal{T}-1}^{T}, \tag{9}$$

where L is the length of the impulse response, ^{T} denotes the transpose operator, and $\mathcal{T}$ is the number of taps of the FIR filters. The filters are adapted to match the target impulse response:

$$y_m = h_{m,1} * g_1 + h_{m,2} * g_2 + \cdots + h_{m,\mathcal{S}} * g_{\mathcal{S}} = \sum_{s=1}^{\mathcal{S}} h_{m,s} * g_s \approx d, \tag{10}$$

where g_s are the FIR filters, $h_{m,s}$ are the impulse responses and y_m is the output at the m-th microphone.

The optimization goal is achieved by minimizing the cost function:

$$J = ||d_{\mathcal{M}} - y||_2, \tag{11}$$

where y is the vector containing the output impulse response $y = [y_1, y_2, \ldots, y_{\mathcal{M}}]$ and $d_{\mathcal{M}}$ is the vector containing $\mathcal{M}$ times the target impulse response. The inverse system g can be obtained by:

$$g = H^+ d_{\mathcal{M}}, \tag{12}$$

where H^+ is the pseudo inverse of the system matrix $H = \begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,\mathcal{S}} \\ \vdots & \vdots & \vdots & \vdots \\ H_{\mathcal{M},1} & H_{\mathcal{M},2} & \cdots & H_{\mathcal{M},\mathcal{S}} \end{bmatrix}$ and its elements $H_{m,s}$ are $(L + \mathcal{T} - 1) \times \mathcal{T}$ circular matrices composed by the impulse responses $h_{m,s}$ [24]:

$$H_{m,s} = \begin{bmatrix} h_{m,s}(0) & 0 & \cdots & 0 \\ h_{m,s}(1) & h_{m,s}(0) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ h_{m,s}(L-1) & \cdots & \vdots & \vdots \\ 0 & h_{m,s}(L-1) & \ddots & \vdots \\ 0 & \cdots & 0 & h_{m,s}(L-1). \end{bmatrix} \tag{13}$$

If H is full rank, then $H^+ = H^{-1}$.

The FIR filters are calculated adaptively: the gradient of the cost function ∇J is given by:

$$\nabla J = -2H^T d_{\mathcal{M}} + 2H^T H g \tag{14}$$

and the inverse system can be obtained by:

$$G(k+1) = G(k) - \mu \nabla J, \tag{15}$$

where μ is the step-size.

5. Experiments

The performance of the proposed and the baseline methods have been assessed by computer experiments using impulse responses measured inside real car cabins. Two car models have been considered, an Alfa Romeo Giulia and a Jeep Renegade. The Giulia was first taken for binaural equalization experiments ($\mathcal{M} = 2$). The impulse responses were obtained using the sine sweep method [34] as implemented by the Aurora plugins (http://pcfarina.eng.unipr.it/Aurora_XP/index.htm). Sampling was done at 28.8 kHz with a Roland Octa-Capture audio interface, then the impulse responses were resampled to 48 kHz. A Kemar 45BA mannequin was placed on the driver's seat; the distance between its ears is $d = 18$ cm. The Giulia provides $\mathcal{S} = 7$ loudspeakers—four door woofers, one subwoofer in the trunk, one speaker in the center of the dashboard and one speaker in the driver's headrest, as shown in Figure 5a.

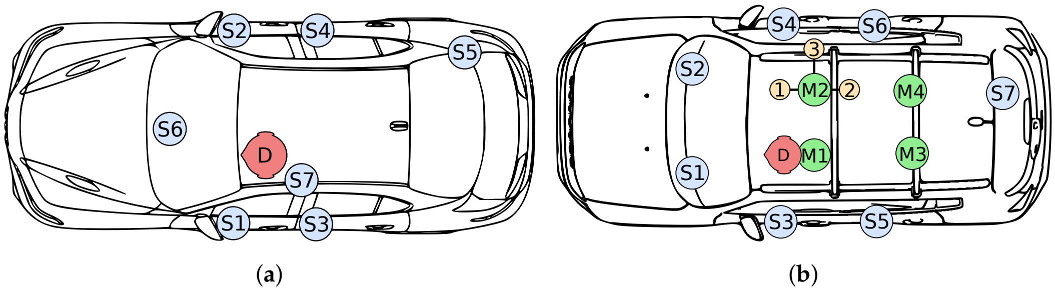


Figure 5. Top view of the Alfa Romeo Giulia (**a**) and the Jeep Renegade (**b**) showing the placement of the $\mathcal{S}$ loudspeakers and the $\mathcal{M}$ microphones. D indicates the dummy head. The three yellow labels around M2 are the proximity test microphone PM1, PM2, PM3.

To assess the equalization performance of the proposed approach in a different environment we have measured the impulse responses of another car, a Jeep Renegade and measured the impulse responses at multiple listening points. Its cabin response has been measured using $\mathcal{M} = 4$ omnidirectional microphone, one per seat. Three additional microphones have been mounted around microphone M2 for proximity tests, to assess the effect of head movements on the equalization performance. These microphones, labeled as PM1, PM2 and PM3 are placed at a distance of 6.5 cm (forward), 6.5 cm (backward) and 22.5 cm (lateral), respectively. For a one-ot-one comparison with the binaural tests done on the Giulia, a binaural mannequin was also mounted at the driver seat to capture binaural impulse responses. The sine sweep method has been used as well, in this case sampling at 48 kHz using an Audio Precision APX-586 analyzer and a Crown D-75A power amplifier to drive the loudspeakers. The Renegade loudspeakers are located in the car dashboard, on the four doors and a subwoofer is placed in the trunk.

The baseline methods have been implemented in *Matlab* (https://mathworks.com/products/matlab.html), while the proposed methods have been implemented in Python using *Keras* (https://keras.io/) with *Tensorflow* (https://www.tensorflow.org/) as backend. They have been executed on a machine with Intel Core i7-4930K 3.40 GHz clock processor, 32 GB of RAM and Nvidia GTX-Titan GPU with 12 GB of dedicated RAM.

The results are provided in terms of the mean square error (MSE) and average standard deviation $\overline{\sigma}$. The MSE of the magnitude response is calculated bin-by-bin for each microphone between the desired frequency response and the measured magnitude frequency response. The results are averaged between all microphones, that is,:

$$\overline{MSE} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \left(\frac{\sum_{\omega=\omega_l}^{\omega_h} \Big(|Y_m(\omega)| - |Y_{des}(\omega)| \Big)^2}{\omega_h - \omega_l} \right). \tag{16}$$

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

$$\overline{\sigma} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \sigma_m, \tag{17}$$

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

$$\sigma_m = \sqrt{\frac{1}{\omega_h - \omega_l + 1} \sum_{\omega=\omega_l}^{\omega_h} (10 \cdot log_{10}|F_m(\omega)| - D)^2} \tag{18}$$

$$D = \frac{1}{\omega_h - \omega_l + 1} \sum_{\omega=\omega_l}^{\omega_h} (10 \cdot log_{10}|F_m(\omega)|). \tag{19}$$

F_m is the sum of the frequency responses on the m-th microphone without equalization filters or with equalization filters, following Reference [35].

Since the Giulia impulse responses have been originally sampled at 28.8 kHz, we set the upper frequency bound ω_h to 14.4 kHz. The lower frequency bound ω_l is set to 20 Hz to avoid unnecessary equalization below the human hearing range.

We desire the FIR filters to have linear phase, that is, be symmetric. Following the frequency deconvolution approach, we impose an odd number of taps for all methods.

Preliminary experiments were conducted to determine the values for the training hyperparameters. During these experiments we observed that a sufficiently high number of iterations allows the networks to converge to very low errors. The learning rate was set to $1 \cdot 10^{-3}$ for all the proposed approaches. w was set to 100.0 and the batch size is set to 1. The Adam optimizer [36] was used, with *decay* equal to $3 \cdot 10^{-8}$ for all architectures. The number of iterations of the SD was set to 250,000, as in Reference [21]. A similar number of iterations, 200,000, was set for the proposed methods. This leaves enough time for convergence and allows direct comparison to the evolutionary algorithms in Reference [21], where the number of iterations times the agents gives approximately 200,000.

Four convolutional layers configurations were generated randomly. These were applied to the convolutional networks used in the CNN and AE architectures. The first convolutional layer has kernels of size $M \times 1$ while the second, if present, has kernels of size $1 \times S$. The fully connected layers following the convolutional ones have been varied in their number (1, 2) and size. Four MLP architectures were derived from the convolutional ones by retaining the size of the fully connected layers. Three additional configurations have been added to achieve a number of trainable parameters similar to those of the CNN, as reported in Table 1.

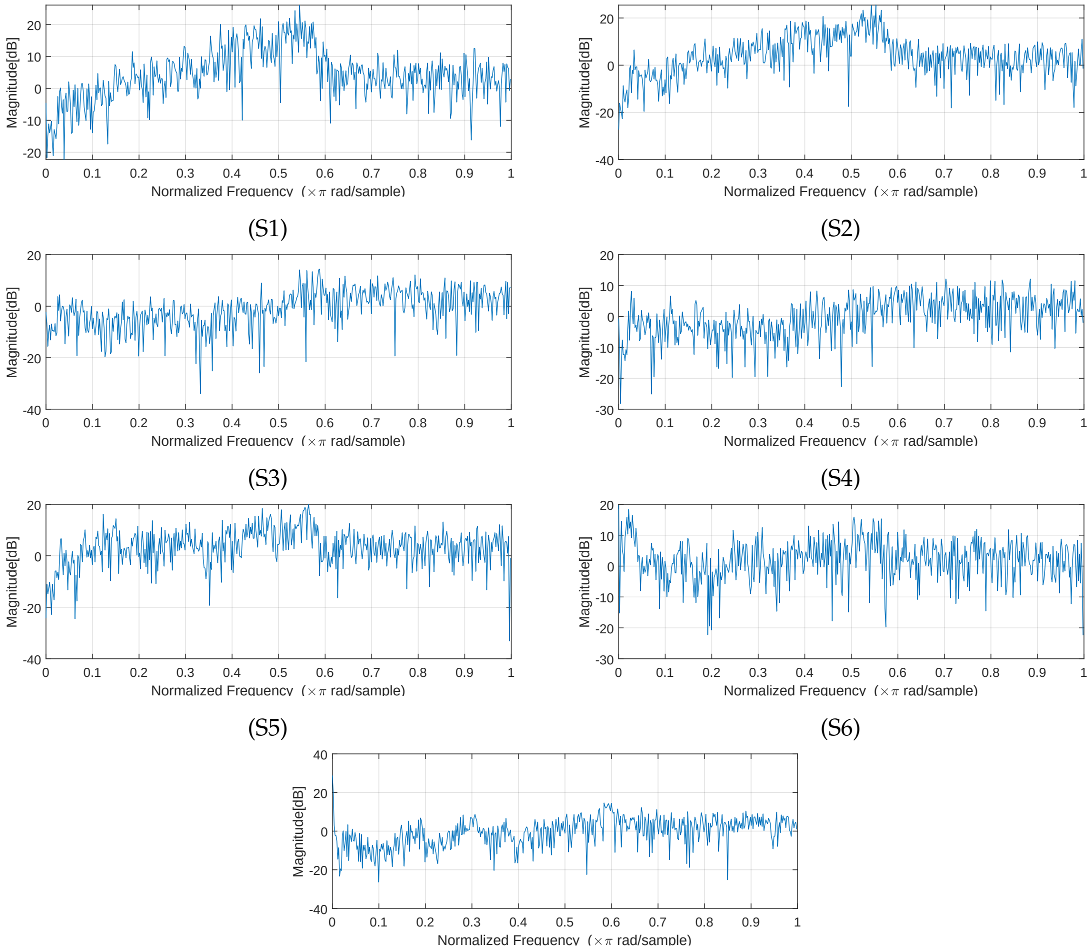
Table 1. The CNN and MLP configurations used in the experiments. The number of parameters are referred to filters of 1024-th order.

CNN				MLP		
Configuration	**Number of Kernels**	**Number of Units**	**Trainable Parameters**	**Configuration**	**Number of Units**	**Trainable Parameters**
Conv #1	[48, 24]	[10]	7,481,943	MLP #1	[10]	6,798,935
Conv #2	[10, 5]	[100, 10]	3,826,153	MLP #2	[100, 10]	67,280,035
Conv #3	[100, 25]	[100, 100]	12,483,433	MLP #3	[100, 100]	67,934,875
Conv #4	[10]	[1000]	3,825,863	MLP #4	[1000]	679,183,175
				MLP #5	[100]	67,924,775
				MLP #6	[100, 100, 100]	67,944,975
				MLP #7	[5]	36,003,713
				MLP #8	[10, 1000, 1000]	14,914,185

6. Results

6.1. Alfa Romeo Giulia

Binaural equalization results are shown in Table 2 for the Alfa Romeo Giulia. The two proposed methods based on deep neural networks outperform significantly any other method in the test, while the MLP does not reach the same performance as the FD and the SD. The CNN achieves slightly better results compared to the convolutional AE despite being simpler in terms of implementation and computational cost. Best overall results have been achieved using the CNN with FIR filters of order 1024. Their magnitude frequency response is shown in Figure 6. Shorter filters designed by the convolutional methods are subject to a slight performance degradation, however, their MSE remains very low.



(S1)

(S2)

(S3)

(S4)

(S5)

(S6)

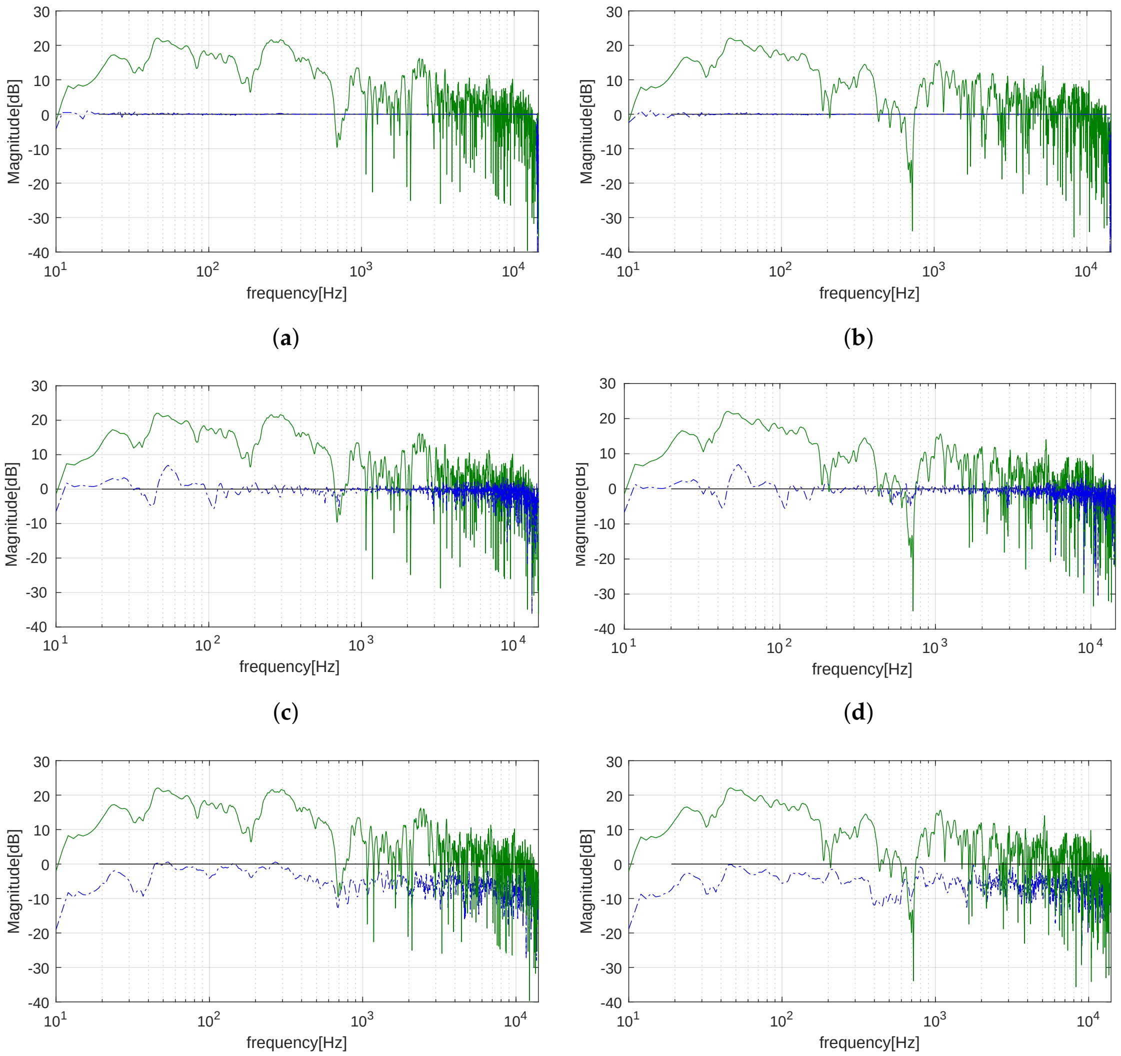
(S7)

Figure 6. Magnitude frequency response of the 1024-th order FIR filters designed by the CNN for each one of the Alfa Romeo Giulia loudspeakers S1-S7 shown in Figure 5a.

In Figure 7, we compare the non-equalized (green) and equalized (blue) magnitude frequency response at the dummy head left and right microphone obtained from the filters designed with the CNN and the baseline approaches. The CNN filters correct the frequency responses obtaining an exceptionally flat magnitude. No relevant peaks or notches are present in the equalized frequency response. The FD method achieves a rather flat spectrum, but peaks and notches are still visible. The SD presents the higher $\overline{MSE}$, while its $\overline{\sigma}$ is lower than the FD. Indeed, the frequency responses have less peaks, but the magnitude response is biased and sits below 0 dB. The same happens for other FIR filter orders.

Table 2. Audio equalization results for the Alfa Romeo Giulia with binaural microphones. Please note that the $\overline{MSE}$ in the absence of equalization is 2.19, with $\overline{\sigma}$ 3.52. Best results for each column are highlighted in bold.

Filter Order	**MLP**			**AE**			**CNN**			**FD ($\beta = 0.1$)**		**SD**	
	Conf.	$\overline{MSE}$	$\overline{\sigma}$	**Conf.**	$\overline{MSE}$	$\overline{\sigma}$	**Conf.**	$\overline{MSE}$	$\overline{\sigma}$	$\overline{MSE}$	$\overline{\sigma}$	$\overline{MSE}$	$\overline{\sigma}$
512	MLP #5	0.32	2.877	Conv #1	$9.72 \cdot 10^{-4}$	0.136	Conv #2	$7.90 \cdot 10^{-4}$	0.122	0.18	2.52	0.40	1.95
640	MLP #8	0.36	2.730	Conv #1	$3.80 \cdot 10^{-4}$	0.085	Conv #2	$3.74 \cdot 10^{-4}$	0.084	0.15	2.34	0.35	1.72
768	MLP #5	0.46	2.796	Conv #1	$1.66 \cdot 10^{-4}$	0.056	Conv #2	$1.79 \cdot 10^{-4}$	0.058	0.14	2.23	0.33	1.60
896	MLP #2	0.45	2.799	Conv #1	$1.07 \cdot 10^{-4}$	0.045	Conv #1	$1.02 \cdot 10^{-4}$	0.044	0.12	2.07	0.31	1.54
1024	MLP #7	**0.32**	**2.746**	Conv #1	$\mathbf{6.85 \cdot 10^{-5}}$	**0.036**	Conv #1	$\mathbf{6.31 \cdot 10^{-5}}$	**0.034**	**0.10**	**1.93**	**0.30**	**1.50**



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 7. Magnitude frequency responses at the left and right microphones of the dummy head in the Alfa Romeo Giulia after applying filters obtained from the CNN (**a**,**b**), Frequency Deconvolution (**c**,**d**) and Steepest Descent (**e**,**f**) methods. The original magnitude frequency response is shown in green while the equalized frequency response is shown in blue. The target magnitude response is shown in black.

The performance of the FD method is known to be dependent on the β parameter, which can be adjusted as a fixed constant or a frequency-dependent parameter, usually having dominance in the denominator for very low and high frequencies, to avoid excessive gain in the inverse filter in those ranges or to avoid equalization at all. We have tested different configurations of β to search for the best performance of the FD method for a given filter order. Table 3 reports the MSE and sigma for several values of β and for two frequency-dependent β with filter order 1024. Although, theoretically, with lower β the inversion should get closer to ideal, thus reaching a lower MSE, the filter order constraints the performance by truncating the very long ideal impulse response. A sweet spot is obtained for β in the range $10^{-2} < \beta < 10^{-1}$. With larger β the performance decreases, as expected. Some frequency-dependent configurations for β have been selected that obtain good results. The V-shaped one is able to reduce the MSE by a tiny amount, however, no significant improvement can be found by using a frequency-dependent β. Overall, the MSE values do not change much from those of Table 2, thus confirming that the choice of β in the experiments above is not adversely affecting the performance.

Table 3. Effect of the parameter β on the performance. The V-shaped configuration refers to a frequency-dependent β with a minimum of 10^{-4} at 1 kHz and maxima of 10^{-1} at DC and Nyquist, varying linearly on a dB scale. The U-shaped configuration takes a value of 10^{-4} in the range 100 Hz–10 kHz and one elsewhere. Best results for each column are highlighted in bold.

β	$\overline{MSE}$	$\overline{\sigma}$
10^{-4}	0.123	1.83
10^{-3}	0.118	1.82
10^{-2}	0.108	1.81
10^{-1}	0.108	1.93
1	0.281	2.71
10	0.686	4.2
100	0.937	5.09
V-shaped	**0.101**	**1.829**
U-shaped	0.124	1.86

As seen above, even though the elimination of the regularization term β should lead to an almost perfect inversion, the ideal inverse response is limited by the filter order, thus increasing the MSE for very low β. On the contrary, the proposed approach seems to achieve a very low error even with short filters.

6.2. Jeep Renegade

Taking the CNN as the best of the proposed methods and the FD as the best among the baseline methods, we continue our experiments in a different cabin, increasing the complexity of the problem by increasing the number of microphones, that is, listening points, to equalize and by increasing their distance. We also conduct a binaural experimental case, as a one-to-one comparison to the Giulia case.

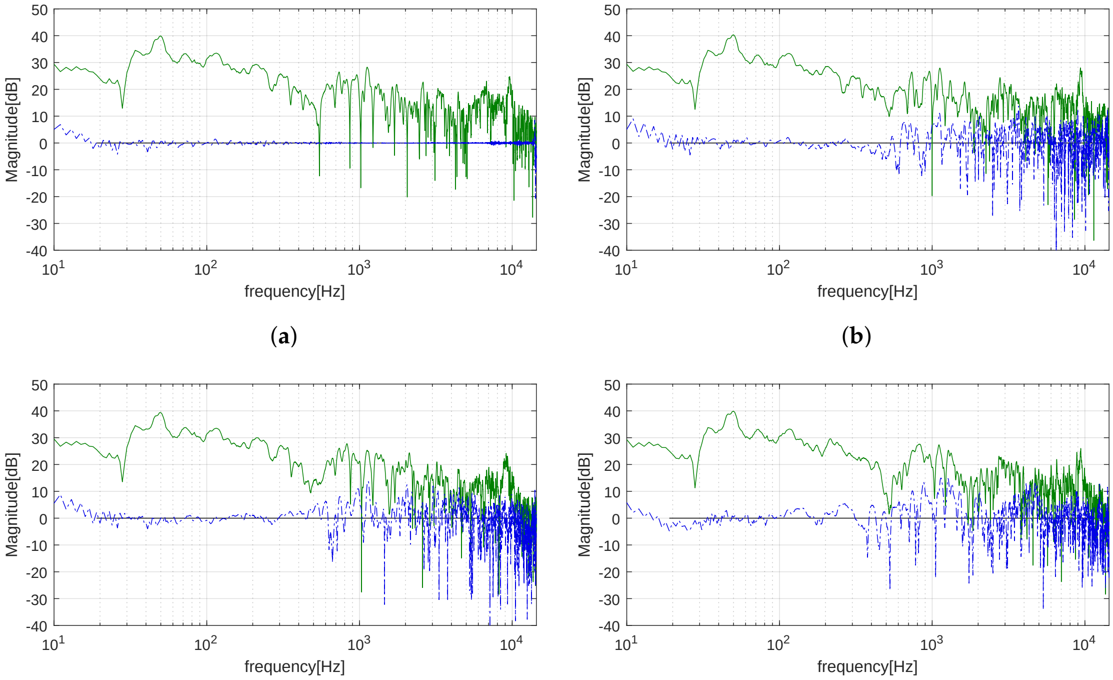
Table 4 reports the results for filters of order 1024. As can be seen, the CNN achieves approximately the same results as in the Giulia on the binaural equalization scenario ($6.19 \cdot 10^{-5}$ vs. $6.31 \cdot 10^{-5}$). As expected, there is a performance decrease with the 4-seats equalization, however, the MSE is still extremely low ($5.7 \cdot 10^{-4}$). With respect to the Giulia, the FD method achieves a reduction of the MSE in the binaural case. A slight degradation of the performance is found for the 4-seats equalization as well. In conclusion, despite the degradation of the performance, results are still far superior than the state of the art method even in the 4-point scenario.

Table 4. Audio equalization results for the Jeep Renegade with binaural microphones and four microphones (one per seat). The FIR order is 1024.

Setup	**CNN**			**FD $\beta = 0.1$**	
	Conf	$\overline{MSE}$	$\overline{\sigma}$	$\overline{MSE}$	$\overline{\sigma}$
Binaural	#1	$6.19 \cdot 10^{-5}$	0.035	0.05	1.21
4 seats	#1	$5.7 \cdot 10^{-4}$	0.106	0.15	1.95

6.3. Sensitivity to Head Movements

Small head movements may result in a degradation of the equalization performance. For this reason, we assessed the validity of the proposed approach in response to small and large head movements. We analyzed the frequency response at three additional points: PM1 and PM2 (small head movement) and PM3 (large head movement). Their frequency response is shown in Figure 8, while their $\overline{MSE}$ and $\overline{\sigma}$ are presented in Table 5, and compared to the one at the M2 microphone, for reference. In line with theory, the error tends to rise for high frequencies, for which the wavelength is shorter or of the same order of magnitude as the distance between microphone M2, however, in the low end the response is almost flat.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. Frequency response at microphone M2 (**a**); microphones PM1 and PM2 (**b**,**c**), corresponding to small forward and backward head movements; microphones PM3 (**d**), corresponding to a large lateral head movement.

Table 5. Audio equalization results for microphone M2 and microphones PM1, PM2 and PM3. The evaluation is achieved by the experiments performed using the Jeep Renegade with four microphones (see Table 4).

Mic.	**CNN**		**FD**	
	$\overline{MSE}$	$\overline{\sigma}$	$\overline{MSE}$	$\overline{\sigma}$
M2	$5.07 \cdot 10^{-4}$	0.10	0.14	1.82
PM1	0.61	2.88	1.2	2.9
PM2	0.50	3.31	0.57	3.07
PM3	0.80	3.09	0.84	3.12

This issue is common to many widely used offline equalization algorithms, including that in Reference [33]. These algorithms can be complemented with adaptive solutions to tune the filters in real-time. Several solutions have been previously proposed, based, for example, on Kalman filtering and Steepest Descent to adaptively track the frequency response [25] or on the virtual microphone technique [37]. The proposed method could also be expanded to equalize a broader area using multiple microphones concentrated around a volume of space surrounding the listener's head.

6.4. Sensitivity to the Input

Finding the best input features and dimensions is an issue in audio tasks that usually has no clear answer, and requires, thus, experimentation. In this work, furthermore, we exploit deep neural networks as optimizing algorithms, which is rather uncommon in the signal processing literature. Up to our knowledge, there is no prior experience in the application of neural networks in such a configuration for the goal of generating audio filters, thus the choice of the input is not trivial. To improve our understanding of this task, we have performed a new batch of experiments to observe the role of the input features in the optimization task. Specifically, we want to assess the role of the input in driving the optimization process.

For these experiments the input matrix is filled with either: *(a)* random values changing at each iteration, *(b)* random values constant for all the training, *(c)* all ones, *(d)* all zeros. We kept the same matrix size used in previous experiments, in order to leave the input layers and the number of trainable parameters unchanged. We conducted these experiments with all the four CNN configurations and all four kinds of inputs, and generated FIR filters of order 1024 for the Alfa Romeo Giulia case. Results are shown in Table 6. In case *(a)*, results are comparable to the FD method, but worse than the ones achieved by the proposed method in Table 2. The fixed random input and a unitary matrix get much closer to the results seen in Table 2, but are still not on par with the best result of the test. Finally, with the null matrix, all filters coefficients are zero, making this method unsuitable to the optimization process. Overall, it seems that our method can gain some advantage from the use of the measured impulse responses as input features, however, the network is able to design suitable filters even with non-informative input content, gaining information about the problem setup from the loss, where the impulse responses are employed to calculate the distance.

Table 6. Effect of the input type on the results of the CNN (filter order 1024). For each case, the best result and the related configuration is reported.

Input	$\overline{MSE}$	$\overline{\sigma}$	**Conf.**
Impulse Responses	$6.31 \cdot 10^{-5}$	0.034	Conv #1
Random Iteration	0.14	2.152	Conv #1
Random Fixed	$1.35 \cdot 10^{-4}$	0.052	Conv #1
All 1s	$1.17 \cdot 10^{-4}$	0.049	Conv #1
All 0s	ill-conditioned		

6.5. Over-Determined Case

In the selected use cases, the number of filters is larger than the number of microphones. To assess the validity of the method in single-channel configurations and in the over-determined case ($\mathcal{M} > \mathcal{S}$) we have conducted further experiments selecting a subset of the available impulse responses, thus simulating the presence of a lower number of speakers. The results are reported in Table 7. As can be seen, the CNN scores better than the FD, meaning that the optimal solution in the least-squares sense can be further improved by non-convex optimization techniques. The performance degradation from the 1×1 case to the 2×1 case is extremely low. This suggests that the two impulse responses are quite similar. On the other hand, the performance improvement achieved by the CNN with the 2×7 or the 4×7 cases (Tables 2 and 4) with respect to the 1×1 cases suggests that the proposed method is able to efficiently exploit a large number of filters to greatly reduce the error at all microphones.

Table 7. Audio equalization in the single-channel and over-determined cases. Setup is $\mathcal{M} \times \mathcal{S}$.

Car	**Setup**	**CNN**		**FD**	
		$\overline{MSE}$	$\overline{\sigma}$	$\overline{MSE}$	$\overline{\sigma}$
Giulia	1×1	0.52	8.57	0.62	9.84
	2×1	0.57	7.81	0.64	9.19
Renegade	1×1	0.03	1.34	0.12	2.01
	4×1	0.22	2.76	0.44	3.62

6.6. Remarks

One concern related to the proposed filter design technique is the computational cost, since the design procedure requires a complete training of the network. However, despite the very large number of iterations set for the experiments, the loss decays exponentially as it is typical of neural networks. As an example, in the Alfa Romeo Giulia 1024-th order CNN experiments, the MSE decays below $1 \cdot 10^{-4}$ after 4200 iterations. It is thus possible to set a desired error threshold and stop the training as soon as it is reached.

For what concerns the filters, we have concentrated our attention on the frequency response, without considering the phase. The Frequency Deconvolution method provides symmetrical, thus linear, phase frequency responses, while the Steepest Descent algorithm does not. We would expect an arbitrary phase response from the proposed approach, since we do not constrain the phase in any way. However, from all our experiments we observe an almost linear phase response, as seen in Figure 9, where this is compared with a linear phase response, showing a close match. As an example, the mean squared phase error compared against a perfectly linear phase response and averaged over all the filters generated in the 1024th-order CNN case from Table 2, is 0.8 rad.

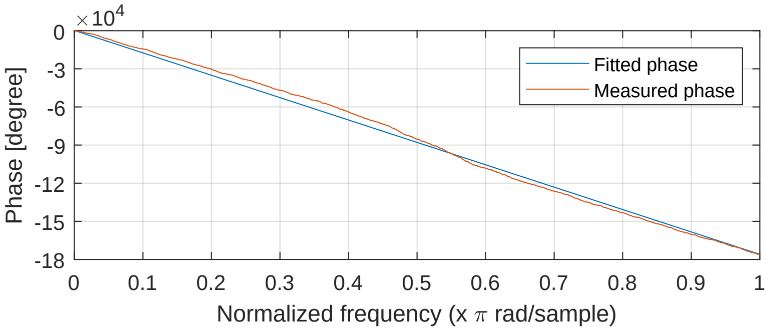


Figure 9. Phase response of one of the filters achieved with the CNN method (FIR order 1024) and a linear fitting. Frequency is normalized according to Nyquist.

Another important issue to consider is the group delay introduced by the filters. As shown by the results, the most performing ones in terms of frequency response equalization are 1024-th order. This filter length, however, may not be acceptable in some applications due to computational cost and the introduction of a group delay of 513 taps (approximately 1.1 ms at a 44,100 Hz sampling rate). Experimental tests have proven that FIR filters of 512-th order present very good equalization capabilities, inferior by 1 order of magnitude compared to the 1024-th order case, but still largely superior than baseline techniques.

6.7. Results Summary

To conclude this section, we report a brief summary of the experiments. We have performed binaural equalization experiments in two environments, the cabin of an Alfa Romeo Giulia and a Jeep Renegade. In Figure 10 we report the best results obtained for the best proposed architecture, a CNN and the best of the comparative methods, the FD method, a widely used approach for inversion of the impulse response in single and multipoint scenarios. As shown, the CNN architecture outperforms FD by several orders of magnitude (see Section 6.1), highlighted by the logarithmic-scaled plot, in both the mean squared error $\overline{MSE}$ and the standard deviation $\overline{\sigma}$. The best result achieved by the CNN in the binaural case has been obtained for the Jeep Renegade ($6.19 \cdot 10^{-5}\overline{MSE}$ in Section 6.2).

With the Jeep Renegade, we also conducted tests with four equalization points, leaving all other parameters identical. The results are slightly lower, but still remarkable: $5.7 \cdot 10^{-4}\overline{MSE}$, meaning that it is still feasible to obtain an almost flat equalization profile for four passengers at the same time. Furthermore, in Section 6.3 we have tested for performance degradation for head movements using three additional microphones around one of the reference microphones used for the 4-points equalization. The results show, in line with theory, a slight degradation of the performance at high frequency (see Figure 8), as with other multipoint equalization approaches.

Finally, we have analyzed the loss decay with the CNN and concluded that the number of training epochs can be reduced significantly, for example, from 200,000 to 4200 with a reasonable degradation of performance ($\overline{MSE} < 10^{-4}$).

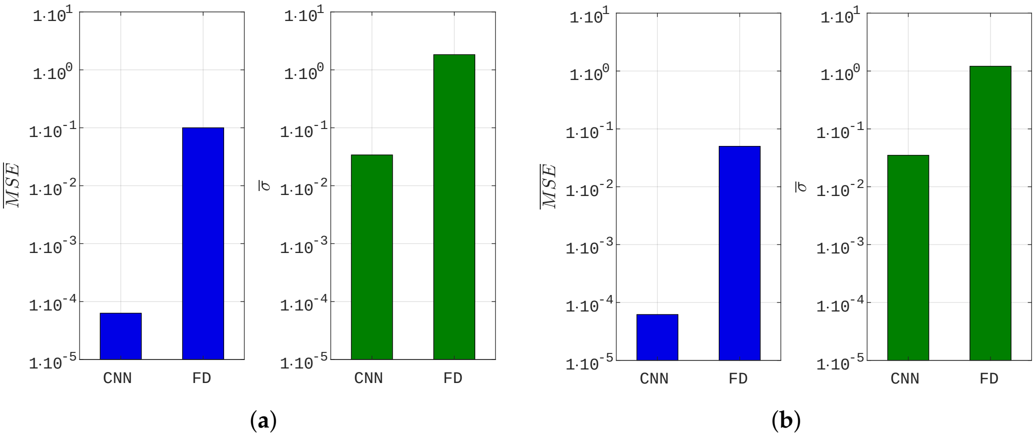


Figure 10. The $\overline{MSE}$ and $\overline{\sigma}$ of the best among the proposed approaches (CNN) and the best among the comparative methods (FD), from the Alfa Romeo Giulia experiments (**a**), and the Jeep Renegade experiments (**b**). Refer to Tables 2 and 4 for more details.

7. Conclusions

In this work, we have shown a binaural and a multipoint audio equalization system based on a deep neural network approach to tune FIR filter coefficients. We proposed the use of the back-propagation algorithm as an optimization method in order to train a neural network to produce FIR coefficients able to satisfy specific criteria provided as loss function.

Three neural network architectures—MLP, CNN, and AE—are compared with state-of-the-art methods. Results show that deep learning approaches outperform other techniques by several orders of magnitude, yielding extremely flat magnitude frequency responses with a quasi linear phase. Among the networks, the CNN provided best results. Additional experiments highlighted the ability of the CNN to converge to a solution that is slightly superior to the least-squares one even when the system to solve is over-determined, motivating further studies on non-convex optimization methods for audio equalization. Finally, the effect of head movements has been studied using additional microphones. The proposed technique cannot be used in a real-time context, thus other techniques can be envisioned to tune the filters adaptively by tracking the head movements, as suggested in Section 6.3. Another possibility is the extension of the current work to a broader area by using multiple microphones in the vicinity of the head.

Although the training stage can be heavy in computational cost, the convergence speed is quite fast, allowing a user to set a desired error threshold to stop the iterations as soon as the objective is reached.

Since the deep neural network approach has shown to be capable in the design of audio filters meeting the expected goals, this research topic may be expanded in the future to different applications and constraints.

Several topics have been left for future works and need to be addressed, such as a subjective evaluation and the design of IIR filters. Given their lower computational cost, compared to FIR filters, they may be suitable for real-time implementation. The use of psychoacoustically oriented metrics, such as 1/3 octave band smoothed frequency responses, may drive the optimization to a frequency response that better represents the human auditory perception. Finally, a thorough exploration of the hyperparameters, the input features and their size, may lead to smaller neural networks with the same performance, improving the filter design speed.

Author Contributions: Conceptualization, L.G. and S.S.; Software, G.P. and L.G.; Investigation, G.P. and L.G.; Writing—original draft, L.G. and G.P.; Writing—review and editing, L.C.; supervision, S.S. and L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This project is founded under the Italian Ministry of Economic Development (MiSE) Ministerial Decree 24/05/2017 (Agreements for Innovation).

Acknowledgments: The authors would like to thank ASK industries S.P.A. for financial support and technical assistance.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cecchi, S.; Carini, A.; Spors, S. Room Response Equalization—A Review. *Appl. Sci.* **2017**, *8*, 16. doi:10.3390/app8010016. [CrossRef]
2. D'Orazio, D.; Garai, M. The autocorrelation-based analysis as a tool of sound perception in a reverberant field. *Riv. Estet.* **2017**, 133–147. [CrossRef]
3. Karjalainen, M.; Paatero, T.; Mourjopoulos, J.N.; Hatziantoniou, P.D. About room response equalization and dereverberation. In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, USA, 16 October 2005; pp. 183–186. doi:10.1109/ASPAA.2005.1540200. [CrossRef]
4. Shaymah, Y.; Angela, A. Channel impulse response equalization scheme based on particle swarm optimization algorithm in mode division multiplexing. *EPJ Web Conf.* **2017**, *162*, 01023. doi:10.1051/epjconf/201716201023. [CrossRef]
5. Krusienski, D.J.; Jenkins, W.K. The application of particle swarm optimization to adaptive IIR phase equalization. In Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, QC, Canada, 17–21 May 2004; Volume 2, p. ii-693. doi:10.1109/ICASSP.2004.1326352. [CrossRef]
6. Mohammed, J.R. A Study on the Suitability of Genetic Algorithm for Adaptive Channel Equalization. *Int. J. Electr. Comput. Eng. (IJECE)* **2012**, *2*, 285–292. doi:10.11591/ijece.v2i3.312. [CrossRef]
7. Chang, P.; Lin, C.G.; Yeh, B. Inverse filtering of a loudspeaker and room acoustics using time-delay neural networks. *J. Acoust. Soc. Am.* **1994**, *95*, 3400–3408. doi:10.1121/1.409959. [CrossRef]
8. Sabin, A.T.; Pardo, B. A Method for Rapid Personalization of Audio Equalization Parameters. In Proceedings of the 17th ACM International Conference on Multimedia (MM '09), Vancouver, BC, Canada, 19–24 October 2009; ACM: New York, NY, USA, 2009; pp. 769–772. doi:10.1145/1631272.1631410. [CrossRef]
9. Pardo, B.; Little, D.; Gergle, D. Building a Personalized Audio Equalizer Interface with Transfer Learning and Active Learning. In Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM '12), Nara, Japan, 29 October–2 November 2012; ACM: New York, NY, USA, 2012; pp. 13–18. doi:10.1145/2390848.2390852. [CrossRef]
10. Reed, D. A Perceptual Assistant to Do Sound Equalization. In Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI 00), New Orleans, LA, USA, 9–12 January 2000; ACM: New York, NY, USA, 2000; pp. 212–218. doi:10.1145/325737.325848. [CrossRef]
11. Grachten, M.; Deruty, E.; Tanguy, A. Auto-adaptive Resonance Equalization using Dilated Residual Networks. *arXiv* **2018**, arXiv:1807.08636.
12. Martinez Ramirez, M.A.; Reiss, J.D. End-to-End Equalization with Convolutional Neural Networks. In Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18), Aveiro, Portugal, 4–8 September 2018. Available online: http://dafx2018.web.ua.pt/papers/DAFx2018_paper_27.pdf (accessed on 3 April 2020).
13. Agrawal, N.; Kumar, A.; Bajaj, V. A New Design Method for Stable IIR Filters With Nearly Linear-Phase Response Based on Fractional Derivative and Swarm Intelligence. *IEEE Trans. Emerg. Top. Comput. Intell.* **2017**, *1*, 464–477. doi:10.1109/TETCI.2017.2748151. [CrossRef]
14. Kamra, I.; Sidhu, D.S.; Sidhu, B.S. Design of Digital IIR Low Pass Filter Using Particle Swarm Optimization (PSO). *Int. J. Sci. Res. Eng. Technol. (IJSRET)* **2014**, *6*, 275–280.
15. Foresi, F.; Vecchiotti, P.; Zallocco, D.; Squartini, S. *Designing Quasi-Linear Phase IIR Filters for Audio Crossover Systems by Using Swarm Intelligence*; Audio Engineering Society Convention 144; Audio Engineering Society: Milan, Italy, 2018.

16. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. Filter modeling using gravitational search algorithm. *Eng. Appl. Artif. Intell.* **2011**, *24*, 117–122. doi:10.1016/j.engappai.2010.05.007. [CrossRef]
17. Kalinli, A.; Karaboga, N. Artificial immune algorithm for IIR filter design. *Eng. Appl. Artif. Intell.* **2005**, *18*, 919–929. doi:10.1016/j.engappai.2005.03.009. [CrossRef]
18. Allakhverdiyeva, N. Application of neural network for digital recursive filter design. In Proceedings of the 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 12–14 October 2016; pp. 1–4, doi:10.1109/ICAICT.2016.7991720. [CrossRef]
19. Kumari, M.; Kumar, M.; Saxena, R.; Wal, A. Performance analysis of FIR Low Pass FIR Filter using Artificial Neural Network. *Int. J. Eng. Trends Technol.* **2017**, *50*, 58–62. doi:10.14445/22315381/IJETT-V50P210. [CrossRef]
20. Wang, X.H.; He, Y.G.; Li, T.Z. Neural Network Algorithm for Designing FIR Filters Utilizing Frequency-Response Masking Technique. *J. Comput. Sci. Technol.* **2009**, *24*, 463–471. doi:10.1007/s11390-009-9237-0. [CrossRef]
21. Pepe, G.; Gabrielli, L.; Squartini, S.; Cattani, L. Evolutionary tuning of filters coefficients for binaural audio equalization. *Appl. Acoust.* **2020**, *163*, 107204. [CrossRef]
22. Azzali, A.; Bellini, A.; Farina, A.; Ugolotti, E. *Design and Implementation of Psychoacoustics Equalizer for Infotainment*; DSP Implementation Day, Politecnico di Milano: Milano, Italy, 2002; Volume 23.
23. Bellini, A.; Farina, A.; Cibelli, G.; Ugolotti, E.; Bruschi, F. *Experimental Validation of Equalizing Filters for Car Cockpits Designed with Warping Techniques*; Audio Engineering Society: New York, NY, USA, 2000.
24. Zhang, W.; Khong, A.W.H.; Naylor, P.A. Adaptive inverse filtering of room acoustics. In Proceedings of the 2008 42nd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 26–29 October 2008; pp. 788–792. doi:10.1109/ACSSC.2008.5074517. [CrossRef]
25. Dagar, A.; Nitish, S.S.; Hegde, R. Joint Adaptive Impulse Response Estimation and Inverse Filtering for Enhancing In-Car Audio. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 416–420, doi:10.1109/ICASSP.2018.8462329. [CrossRef]
26. Cichocki, A.; Unbehauen, R. *Neural Networks for Optimization and Signal Processing*; John Wiley & Sons, Inc.: New York, NY, USA, 1993.
27. Villarrubia, G.; De Paz, J.F.; Chamoso, P.; De la Prieta, F. Artificial neural networks used in optimization problems. *Neurocomputing* **2018**, *272*, 10–16. [CrossRef]
28. Lopez Paz, D.; Sagun, L. Easing non-convex optimization with neural networks. In Proceedings of the International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018.
29. Isola, P.; Zhu, J.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976. doi:10.1109/CVPR.2017.632. [CrossRef]
30. Pathak, D.; Krähenbühl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context Encoders: Feature Learning by Inpainting. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544. doi:10.1109/CVPR.2016.278. [CrossRef]
31. Pascual, S.; Bonafonte, A.; Serrà, J. SEGAN: Speech Enhancement Generative Adversarial Network. *arXiv* **2017**, 3642–3646. doi:10.21437/Interspeech.2017-1428. [CrossRef]
32. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.
33. Kirkeby, O.; Nelson, P.A.; Hamada, H.; Orduna-Bustamante, F. Fast deconvolution of multichannel systems using regularization. *IEEE Trans. Speech Audio Process.* **1998**, *6*, 189–194. doi:10.1109/89.661479. [CrossRef]
34. Farina, A. *Advancements in Impulse Response Measurements by Sine Sweeps*; Audio Engineering Society: Vienna, Austria, 2007.
35. Cecchi, S.; Palestini, L.; Peretti, P.; Piazza, F.; Carini, A. Multipoint equalization of digital car audio systems. In Proceedings of the 2009 6th International Symposium on Image and Signal Processing and Analysis, Salzburg, Austria, 16–18 September 2009; pp. 650–655, doi:10.1109/ISPA.2009.5297665. [CrossRef]

36. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
37. Moreau, D.; Cazzolato, B.; Zander, A.; Petersen, C. A review of virtual sensing algorithms for active noise control. *Algorithms* **2008**, *1*, 69–99. doi:10.3390/a1020069. [CrossRef]