Evolutionary Tuning of Filters Coefficients for Binaural Audio Equalization

Giovanni Pepe[a,b,*], Leonardo Gabrielli[a], Stefano Squartini[a], Luca Cattani[b]

[a] *Department of Information Engineering, Università Politecnica delle Marche, Via Brecce Bianche, 60131, Ancona (AN), Italy*
[b] *ASK Industries S.p.A., Viale Ramazzini, 42124, Reggio Emilia (RE), Italy*

Abstract

This paper presents a novel multichannel audio equalization technique based on evolutionary computation algorithms for tuning the filters coefficients. Specifically, two distinct evolutionary algorithms are used on purpose, i.e. the Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA). Two alternative solutions for the definition of evolutionary particles have been devised and tested with both techniques. Given the desired frequency response, the fitness function is formulated in terms of amplitude spectral distance. These techniques have been assessed by computer experiments, conducted on a in-car binaural equalization scenario considering 7 loudspeakers and a binaural microphone. The obtained results show that the proposed solutions achieve a remarkably superior performance compared to the baseline methods, with a 5 times reduction of the mean square error in the amplitude spectral domain.

Keywords: evolutionary computation, FIR filter design, audio equalization, automotive audio, particle swarm optimization, gravitational search algorithm

1. Introduction

The interior sound quality of a car is an important decision factor for potential customers [1], and the car cabin is among the most used audio listening

[*] Corresponding author
Email addresses: g.pepe@pm.univpm.it (Giovanni Pepe), l.gabrielli@univpm.it (Leonardo Gabrielli), s.squartini@univpm.it (Stefano Squartini), CattaniL@askgroup.it (Luca Cattani)

environments [2].

For this reason, enhancing the tone quality is a necessary step and audio equalization plays a fundamental role. The goal is to modify the cabin frequency response at some sweet spots to approximate the desired results in terms of audio quality [2]. A cabin impulse response will generally differ from a room impulse response (RIR) due to no separation between direct wave and reflections. Moreover cars are characterized by very small enclosure volume, small reverberation time and high Schroeder frequency [3].

The main target of the equalization procedure is to increase sound comfort and make the response closer to a target curve at the driver and passenger positions [4].

In this work, we consider a multiple-input, multiple-output problem (MIMO), where the environment (e.g. a car or a room) is fitted with several sound sources and listening positions (or microphones), as shown in Figure 1. One possible solution is the addition of equalizing filters g_s at each sound source, aiming at compensating for the effect of the system. Their design, however is far from trivial. Despite being linear a problem, a large number of impulse responses must be equalized by a lower number of filters, resulting in an underdetermined problem. The complexity of this problem increases with the number of sources $\mathcal{S}$ and microphones $\mathcal{M}$. The input signal x is convolved with FIR filters g_s, thus, the signal recorded y_m at the microphone m is: [5, 6]:

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

where $h_{s,m}$ is the impulse response related to microphone m and source s.

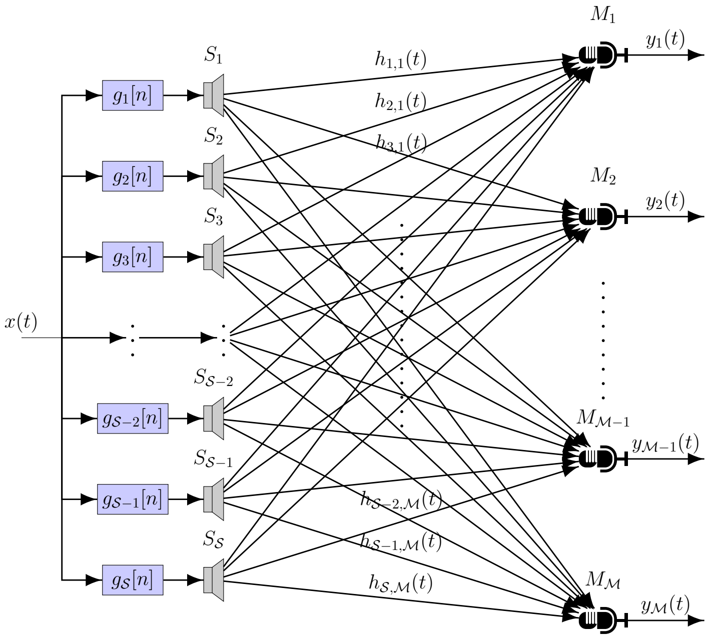


Figure 1: MIMO equalization problem: $\mathcal{S}$ loudspeakers are displaced in an environment together with $\mathcal{M}$ microphones.

1.1. State of the art analysis

To the best of our knowledge no machine learning technique has been proposed in the literature to obtain filter coefficients for audio equalization. A state of the art analysis follows, showing previous application of computational techniques for filter design and equalization in the audio and digital communication fields. Specifically, current audio equalization techniques are briefly described, outlining their shortcomings and focusing on the automotive use case. Then, machine learning techniques for equalization in audio and digital communication will be described. Finally, machine learning methods for FIR and IIR filter design will be outlined.

A large number of audio equalization techniques are proposed by the literature, however, most of these are meant for room equalization. Part of these can, nonetheless, be adapted to the car scenario.

In [7] a report on room equalizers is available, where the filters are classified as minimum-phase and mixed-phase filters. Minimum-phase filters can only provide Room Transfer Function (RTF) magnitude equalization as they only affect the minimum-phase part of the phase response. Mixed-phase filters affect the excess phase RTF component, removing some of the room reverberation.

An early work on a room digital equalization can be found in [8], where a minimum phase inverse filter is designed to remove the effect of room impulse response from a speech signal. Current trends in equalization techniques aim at the elimination of the impulse response effect in a real-time context, usually employing conventional digital signal processing techniques, although recent trends such as computational intelligence and Deep Learning are paving the way for new strategies [9, 10, 11].

A review of room response equalization algorithms [12] classifies design techniques into the following five classes: *Homomorphic filtering*, where the RIR equalizer is obtained by direct inversion of the minimum phase part [13]; *Linear predictive coding* (LPC) analysis, where the room response is modeled with a minimum-phase all-pole filter and the equalizer is a finite impulse response (FIR) filter [14]; *Least-squares optimization* and techniques alike, used for adaptive equalization [15] but sensitive to the peaks and notches of the room response; *Frequency domain deconvolution*, where the equalizer can be directly designed in the discrete Fourier transform domain by considering the reciprocal of the room response; *Multiple-input/multiple-output inverse theorem (MINT) solutions*, that construct the inverse RIR from multiple FIR filters, using multiple loudspeaker or microphones. The latter approach exhibits strong limitations: as shown in [16] the numerical performance is enhanced as the number of loudspeakers is increased. A last approach employs parametric equalizers, that allows to add peaks or notches at arbitrary frequency [17].

Major issues in the equalization approaches described above are: the length of the equalizer impulse response; the reduced size of sweet spots where equalization is effective; the slow time variations of the room response [12]. An ideal equalizer has a very long impulse response capable of compensating peaks and notches: the latter correspond to zeros close to the unit circle in the transfer function, thus, the inverse filter should have poles close to the unit circle, determining a long impulse response. Moreover the notches at high frequencies are extremely variable with position and time, so the equalizer is effective in small regions of the acoustic space. For this reason, a pre-processing technique is required to contrast these variations.

The most important pre-processing techniques are: *Short filters* [18], an LPC-based technique that reduces the length of the equalizer impulse response and can extend the sweet spot; *Non-Uniform Frequency Resolution* improves the accuracy and the effectiveness of equalization, taking in account the characteristics of the room response and the human ear [19], that is non uniform and non linear, and has a logarithmic dependence on frequency; *Room Impulse Response Reshaping*, reshapes the impulse response in such a way that the alteration of the room becomes inaudible. This is used to reduce the reverberation time [20] or to avoid echoes [21].

In [15] one of the first multi-point equalization algorithm is presented, based on the Filtered-X algorithm. In [22] the algorithm is improved for the car audio equalization. Another multi-point techinque is the *clustering method*, which analyze the similarities between different spatially distributed room responses by clustering them to a chosen distance measure [23]. Other techniques use the *common acoustical poles compensation*, like in [24], where the common acoustical poles are estimated as the common pole values of many low-frequency room transfer functions and determined using the LPC model of the room response. *Modal equalization* is used to control excessively long decays in listening room caused by low-frequency modes, minimizing the audibility of these resonances [25]. Another technique is the *Plane Wave Approach*, which generates a plane wave that propagates from one wall to the opposite one, where it is absorbed by the loudspeakers. This method equalizes the sound in the low-frequency region [26]. Another low-frequency approach is the *pressure-field chamber approach* [27]. The quasi-anechoic approach is suited to mid- and high-frequency equalization, and designs the equalizer in two steps also taking into account the loudspeaker response: a mixed-phase equalizer is derived from the quasi-anechoic room impulse response, then a minimum-phase equalizer is used to correct the remaining part of the room response [28].

Regarding the equalization inside the car passenger compartment, Cecchi et al. [2] proposed a multi-point equalization approach using fractional octave smoothing of impulse responses and inverting the model using the frequency deconvolution or LPC.

Farina et al. [3] developed an equalizer for automotive applications smoothing the magnitude of the impulse response and then inverting the filter. Finally the filter is multiplied by the target curve obtained from subjective tests. In [4] a technique is discussed that exploits objective metrics to design an inverse filter: the inverse filter shape is based on the dynamic frequency response rather than the steady-state frequency response.

Some contributions that use machine learning techniques for equalization have been proposed in the literature. In the last years, machine learning techniques have been used for radio channel equalization [29, 30, 31] and audio equalization [32, 33, 34]. The latter ones are aimed at learning pleasant audio equalization frequency responses.

In the communication systems literature a great deal of equalization techniques using machine learning have been proposed. In these works, the aim is the equalization of the transmission channel, having a linear impulse response, that can be improved by FIR or infinite impulse response (IIR) equalizers.

In [35], the authors use the Particle Swarm Optimization (PSO) algorithm to equalize the impulse response of an optical fiber communication. This is shown to provide better results than LMS and RLS technique. Another interesting approach using the PSO algorithm is reported in [36], where the PSO particles are used to obtain optimal poles and zeros of an IIR filter. In [29] a hybrid PSO (HPSO) is used to avoid the algorithm to stick to local minima. A mix of PSO and Support Vector Machines (SVM) is proposed in [30] to average the fitness function on 10 cross-validation folds. Yogi et al. [37] employs PSO to train a Functional Link Artificial Neural Networks (FLANNs): the PSO algorithm is used to update weights on training. In [31] several approaches for initial weight update in PSO are compared for an Adaptive Equalization. Every particle is a filter with various coefficients. The authors based their algorithm on [38].

In [39] Genetic Algorithms (GA) are exploited for Adaptive Channel Equalization, in order to reduce the Intersymbol Interference (ISI) present in the trasmission channel. The GA algorithms are compared with traditional LMS and RLS technique, showing that the GA algorithm converges faster than the LMS techinques and yields better results in terms of MSE.

A first approach for audio equalization is proposed in [40] using neural networks. 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 features and as output of the signal registered by the microphone: the error between the input signal and the output of the network is used for the backpropagation algorithm. The forward approach is also employed, using a delayed copy of the input signal as feature and the difference between the output of the dynamic system and the network as error.

In [41] an end-to-end architecture based on Convolutional Neural Networks (CNN) in the time-domain is implemented. The network is divided into three parts: adaptive front-end, synthesis back-end and a latent-space DNN. The adaptive front-end consists of a convolutional enconder formed by two CNN layers, one pooling layer and one residual connection for the back-end.

In [32, 33] the authors describe a system that maps the gain of each frequency band with the user's preference equalization as training data. A similar approach is undertaken by [34] where k-nearest neighbour (KNN) is used to implement a timbre equalizer based on user preference in terms of brightness, darkness and smoothness.

Recent works addressing the design of IIR filters using PSO can be found in [42, 43, 44]. Foresi et al. [44] 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 with a flat magnitude response and a linear phase. In [45] 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 [46] 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).

Some filter design techniques that use neural networks can be found in the literature, such as [47], where a neural network is devised to design an IIR filter. In this work the error is calculated by the difference in magnitude response between the desired and the calculated filter. Kumari et al. [48] provide a performance comparison of some neural network architectures to design a low pass 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. [49] 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.

1.2. Scope of this Work

In the present work we have implemented two evolutionary algorithms, the PSO and GSA, for the design of equalizing FIR filters. These will be tested on a MIMO equalization scenario within the car environment where a binaural listening point ($\mathcal{M} = 2$) is considered. These techniques can be extended to multi-point audio equalization ($\mathcal{M} > 2$) and to other use cases as well, not addressed in our experiments. Compared to linear adaptive techniques, evolutionary algorithms are nonlinear and may, thus, have more flexibility in solving the problem.

These two evolutionary techniques generate FIR filter coefficients able to achieve, after convolution with the input signal, the desired frequency response, in this case a flat frequency response in a given frequency range. We impose FIR filters to be odd and symmetrical to achieve linear phase. Moreover, two strategies to design particles and agents are defined, referred to as time-domain method and frequency-domain method, described in Section 2.3. Finally two fitness functions are analyzed, described in Section 2.4.

A real time implementation of the proposed methods can be problematic due the high computational cost, which, in turn, depends on the number of filters and their length. Lighter adaptive algorithms for audio equalization [12] can be exploited for real-time fine tuning of the filter coefficients obtained offline from the evolutionary algorithms.

Evolutionary algorithms have been compared with two baseline methods already tested in the car scenario [2, 6]: The Prototyping Design with Frequency Deconvolution technique and the Steepest Descent. The last is an adaptive filtering approach.

The outline of the paper follows. In Section 2 the proposed method is detailed, including a brief explanation of the theory of PSO (Section 2.1) and GSA (Section 2.2). In Section 3 a description of the two baseline methods is provided. In Section 4 the results are presented and finally in Section 5 conclusions are drawn.

2. Proposed method

2.1. Particle Swarm Optimization

The Particle Swarm Optimization is an optimization algorithm based on the social behaviour of bird flocking and fish schooling and it is related to evolutionary computation [50]. The PSO algorithm is based on a population, or swarm, of individuals called particles.

In order to search the global optimum, each particle crosses through the solution space. The algorithm iteratively evaluates the fitness function at different locations creating a map of the best fitness values. Each particle, then, modifies its position using the information of the distance between the current position, the local best p_{best} and the global best g_{best}.

The PSO algorithm steps are:

- Initialization: the swarm is generated with particles taking random positions in the solution space;
- Evaluation: the fitness function is evaluated for each particle;
- p_{best} evaluation: each particle's fitness is compared with the current particle's fitness function;
- g_{best} evaluation: each fitness function is compared with the swarm's overall previous best to obtain g_{best}.
- Update: the position x_i and the velocity v_i at instant k of all particles is updated according to:

  $$v_i(k+1) = W \cdot v_i(k) + c_1 \cdot \zeta(k) \cdot (p_{best} - x_i(k)) + c_2 \cdot \zeta(k) \cdot (g_{best} - x_i(k)) \quad (2)$$

  $$x_i(k+1) = x_i(k) + v_i(k+1) \quad (3)$$

  where W is the inertia weight, $\zeta(k)$ is a random value in the range [0, 1] and c_1 and c_2 are constants;
- Repeat the algorithm from the evaluation process to update until the stopping criterion is met;

2.2. Gravitational Search Algorithm

The GSA is an evolutionary algorithm inspired by the Newtonian laws: a set of agents called masses are introduced to find the optimum solution in a space, crossing it by following Newtonian laws of gravity and motion [51].

Let us define a system with A agents in which the position of the i-th agent is:

$$X_i = (x_i^1, ..., x_i^d, ..., x_i^n), i = 1, 2, ..., A \quad (4)$$

where x_i^d is the position of the i-th agent in the d-th dimension and n is the dimension of each search space. The mass of each agent M_i is calculated after computing the current population's fitness:

$$q_i(k) = \frac{fit_i(k) - worst(k)}{best(k) - worst(k)} \quad (5)$$

$$M_i(k) = \frac{q_i(k)}{\sum_{j=1}^{A} q_j(k)} \tag{6}$$

$fit_i(k)$ is the fitness value of the agent i at instant k, $worst(k)$ and $best(k)$ are:

$$best(k) = \min_{j \in 1,\ldots,A} fit_j(k) \tag{7}$$

$$worst(k) = \max_{j \in 1,\ldots,A} fit_j(k) \tag{8}$$

Then the total forces from a set of heavier masses for an agent F_i^d should be considered based on the law of gravity:

$$F_i^d = \sum_{j \in k_{best}, j \neq i} rand_j G(k) \frac{M_j(k) M_i(k)}{R_{ij} + \epsilon} (x_j^d(k) - x_i^d(k)) \tag{9}$$

After acceleration of an agent a_i^d is computed:

$$a_i^d(k) = \frac{F_i^d}{M_i(k)} = \sum_{j \in k_{best}, j \neq i} rand_j G(k) \frac{M_j(k)}{R_{ij}(k) + \epsilon} (x_j^d - x_i^d) \tag{10}$$

The velocity and the position of the agent, v_i and x_i are then computed, respectively, as:

$$v_i^d(k+1) = rand_i \times v_i^d(k) + a_i^d(k) \tag{11}$$

$$x_i^d(k+1) = x_i^d(k) + v_i^d(k+1) \tag{12}$$

where $rand_i$ and $rand_j$ are two uniformly distributed random numbers in the interval [0, 1], ϵ is a small value $R_{ij}(k)$ is the Euclidean distance between two agents i and j, defined as $R_{ij}(k) = ||X_i(k), X_j(k)||_2$, k_{best} is the set of first A agents with the best fitness value and biggest mass, which is a function of time, initialized to A_0 at the beginning and decreasing with time.

G is the gravitational constant, but it takes an initial value G_0 and then it will be reduced with time:

$$G(t) = G(G_0, k) \tag{13}$$

GSA and PSO are somewhat related [51, 45]. The main differences are described:

- In PSO the direction of an agent is calculated using only two best positions, p_{best} and g_{best}, while in GSA, the agent direction is calculated based on the overall force obtained by all other agents;
- In PSO, updating is performed without considering the quality of the solutions, and the fitness values are not important in the updating procedure, while in GSA the force is proportional to fitness value, thus, allowing the agents to see the influence of force in the sorrounding search space;
- PSO uses a kind of memory for updating the velocity (due to p_{best} and g_{best}). Instead, GSA is memory-less and only the current position of the agents plays a role in the updating procedure [45].
- In PSO, updating is performed without considering the distance between solutions while in GSA the force is inversely proportional to the distance between solutions.

2.3. Evolutionary algorithms for FIR filters design

2.3.1. Time-domain method

One approach to generate filter coefficients using PSO and GSA directly derives the FIR in the time domain. The PSO and GSA algorithms provide a vector $\mathbf{g}_{\mathcal{S}}$ of coefficients that correspond to half of the linear phase FIRs. The complete s-th FIR is obtained as follows,

$$\{g_s[L], g_s[L-1], ..., g_s[1], g_s[0], g_s[1], ..., g_s[L]\} \tag{14}$$

Then the FIR is convolved with a lowpass prototype with cutoff at 15 kHz. The magnitude frequency response at the microphone M_i is obtained by the DFT of the microphone signal, i.e. the sum of the convolutions of signal x with the cascade of the FIR filter and the impulse response from source S_j to M_i. The error is computed using one of the fitness functions later explained. A diagram of the prososed method is shown in Figure 2.

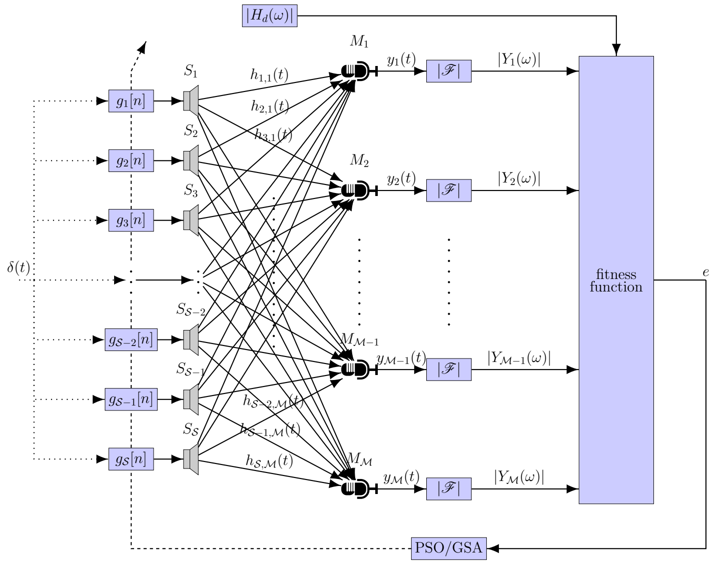


Figure 2: Proposed method for MIMO equalization employing PSO or GSA and direct derivation of the FIR.

2.3.2. Frequency-domain method

Another design method is based on particles or agents expressing the magnitude response of the FIR filters. In this process, the particles or agents are vectors representing the magnitude response of the filters. After the computation, the FIR taps are obtained using the windowing method [52]. This approach should reduce the computational cost because it does not require convolutions and DFT to be performed at each iteration. The fitness functions used are later detailed. The schematic is presented in Figure 3.

2.4. Fitness functions

We propose two fitness functions for the PSO and GSA algorithms: the *mean square error (MSE) approach* and the *MinMax approach*[53].

MSE approach: The mean square error of the magnitude response is calculated bin-by-bin for each microphone from the desired frequency response and the microphone signal magnitude frequency response. The results are averaged

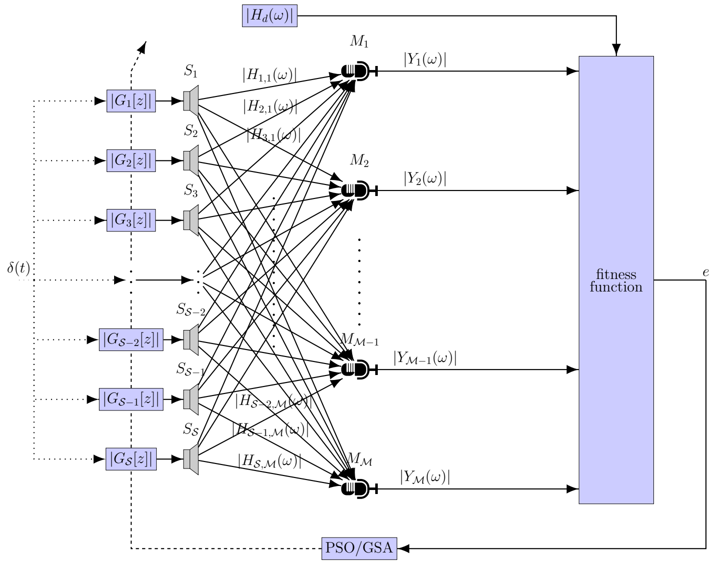


Figure 3: Proposed method for MIMO equalization employing PSO or GSA and generation of the FIR coefficients in the frequency domain.

between all microphones, i.e.:

$$\bar{E}_{MSE} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \frac{1}{\omega} \left(\sum_{\omega} \Big(|H_m(\omega)| - |H_d(\omega)| \Big)^2 \right) \tag{15}$$

where $\mathcal{M}$ is the total number of microphones used for the MIMO equalization. Finally, the minimum MSE is choosen as p_{best} (PSO) or $best(k)$ (GSA).

MinMax approach: The fitness function used in this case is the *MinMax* criterion, thus, the maximum error is first calculated as the maximum bin-by-bin difference between the desired response and the microphone signal frequency response among all the microphones:

$$E_{max} = \max_{m \in \mathcal{M}} \left(\max_{\omega} \Big(\Big| |H_m(\omega)| - |H_d(\omega)| \Big| \Big) \right) \tag{16}$$

Finally the minimum value is chosen as p_{best} (PSO) or $best(k)$ (GSA).

3. Baseline methods

3.1. Design of Inverse Filters using Prototyping design and Frequency Deconvolution technique

One of the employed baseline methods is the frequency deconvolution method [2]. With respect to [2] we generalize the technique from the stereo case to the binaural case.

The mean of the frequency responses, $\widetilde{H}_p(\omega)$, is used as prototype frequency response by averaging all frequency responses after smoothing them using the fractional octave smoothing (with a fraction of 1/32th of octave):

$$\widetilde{H}_{s,m}(\omega) = \sum_{i=0}^{K-1} W_{s,m}(h_w(\omega), i)|H((\omega - 1)modK)| \tag{17}$$

$$\widetilde{H}_p(\omega) = \frac{1}{\mathcal{MS}} \sum_{m=1}^{\mathcal{M}} \sum_{s=1}^{\mathcal{S}} \widetilde{H}_{s,m}(\omega) \tag{18}$$

where $W_{s,m}(h_w(\omega), \omega)$ is a zero-phase window function, $h_w(\omega)$ is the half-window length which is a monotonically increasing function of the frequency index, K is the FFT length and *mod* is the modulo operation.

Inversion is done by frequency deconvolution with regularization:

$$H_{inv}(\omega) = \frac{\widetilde{H}_p^*(\omega)}{|\widetilde{H}_p(\omega)|^2 + \beta(\omega)} \tag{19}$$

where $\beta(\omega)$ is the frequency dependent regularization factor. Finally the filters are obtained using the windowing method.

3.2. Design of Inverse Filters using Steepest Descent

Inverse filters can be obtained by steepest descent as discussed in [5, 6]. One of the most recent techniques is Multiple Input/Multiple Output Inverse Technique (MINT) [16].

The technique consists in defining a target impulse response:

$$d = [\underbrace{0 \quad \ldots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0}_{L+L_g-1}]^T \tag{20}$$

where L is the length of impulse response and L_g is the number of taps of the FIR filters. In this way, the impulse responses are equalized relying to the target ones:

$$y_m = h_{1,m} * g_1 + h_{2,m} * g_2 + \cdots + h_{\mathcal{S},m} * g_{\mathcal{S}} = \sum_{s=1}^{\mathcal{S}} h_{s,m} * g_s \approx d \tag{21}$$

where $g_s, \quad s = 1, \cdots, \mathcal{S}$ are the inverse responses and y_m is the received signal of the m-th microphone. The aim is to minimize the cost function:

$$J = ||d_{\mathcal{M}} - y||^2 \tag{22}$$

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

$$G = H^+ d_{\mathcal{M}} \tag{23}$$

where H^+ is the pseudo inverse of the system matrix H:

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,\mathcal{S}} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,\mathcal{S}} \\ \vdots & \vdots & \vdots & \vdots \\ H_{\mathcal{M},1} & H_{\mathcal{M},2} & \cdots & H_{\mathcal{M},\mathcal{S}} \end{bmatrix} \tag{24}$$

and $H_{m,s}$ is an $(L + L_g - 1) \times L_g$ convolution matrix of $h_{m,s}$ [5]:

$$H_{m,s} = \begin{bmatrix} h_{m,s}(0) & 0 & \cdots & 0 \\ h_{m,s}(1) & h_{m,s}(0) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ h_{m,s}(L-1) & \cdots & \vdots & \vdots \\ 0 & h_{m,s}(L-1) & \ddots & \vdots \\ 0 & \cdots & 0 & h_{m,s}(L-1) \end{bmatrix} \tag{25}$$

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

$$\nabla J = -2H^T d_{\mathcal{M}} + 2H^T H G \tag{26}$$

the inverse system can be obtained by:

$$G(k+1) = G(k) - \mu \nabla J \tag{27}$$

where μ is the step-size.

4. Results

The performance of the proposed and the baseline methods has been assessed by computer experiments. The baseline methods have been implemented in *Matlab*[1]. The implementation of the proposed methods, however, is well suited to parallelization, given the large number of independent particles or agents. The algorithms have been, thus, implemented as *Tensorflow*[2] libraries in order to carry out the processing with GPUs.

The experiments were conducted using impulse responses recorded inside the cockpit of a car, the Alfa Romeo Giulia: binaural impulse responses were measured using a Kemar mannequin type 45BA placed on the driver's seat. The mannequin's ears are 18 cm apart, the dummy is 73 cm high and 44 cm wide. Audition 3.0 with Aurora plug-in was used to measure impulse responses, using Roland Octa-Capture as audio interface and Sine Sweep [54] as input signal, with a sampling rate equal to 28.8 kHz and later oversampled to 48 kHz. The loudspeakers were $\mathcal{S} = 7$, localized on the front left and right door, rear left and right door, inside the trunk, in the central part of the car cockpit and on the front left headrest. The microphones were $\mathcal{M} = 2$, for a total of 14 impulse responses. 7 FIR filters need to be designed, one per source.

In our experiments we first compared the PSO/GSA approaches in both the time- and the frequency-domain with FIR length of 1024 samples. We, therefore, evaluated the performance degradation with reduced FIR length (512, 640, 768, 896 and 1024). Then perfomances with baseline techinques are compared, evaluating the results also with the weighing functions described in Section 4.4.

[1] https://it.mathworks.com/products/matlab.html

[2] https://www.tensorflow.org/

Finally we compare the computational complexity of the evolutionary algorithms and the baseline methods.

Results are provided in terms of the MSE and average standard deviation $\overline{\sigma}$:

$$\overline{\sigma} = \frac{\sum_{m=1}^{\mathcal{M}} \sigma_m}{\mathcal{M}} \tag{28}$$

The MSE is calculated as in (15), while the standard deviation is calculated as:

$$\sigma_m = \sqrt{\frac{1}{Q_h - Q_l + 1} \sum_{i=Q_l}^{Q_h} (10log_{10}|F_m(i)| - D)^2} \tag{29}$$

$$D = \frac{1}{Q_h - Q_l + 1} \sum_{i=Q_l}^{Q_h} (10log_{10}|F_m(i)|) \tag{30}$$

where Q_l e Q_h are the frequency region of interest to the equalization issue, in our case 20 Hz and 14.4 kHz respectively. F_m is the sum of the frequency responses on m-th microphone without equalization filters or with equalization filters following [2].

The input signal is a delta function of length equals to 48000 samples (1 s). For the FFT processing, K is equal to 65536.

Without equalization the MSE and σ are:

Mic 1		Mic 2		Average	
MSE	SD	MSE	SD	MSE	$\overline{\sigma}$
2.81	3.63	1.48	3.51	2.14	3.57

Table 1: Mean and standard deviaton between the unequalized magnitude frequency response and the reference

4.1. Experiments using PSO

The PSO algorithm exposes several hyperparameters affecting its performance that cannot be determine a-priori. A search of the hyperparameters W_{max}, W_{min}, c_1, c_2 has been performed in the ranges: $0.01 < W_{max} < 10$, $0.0001 < W_{min} < 0.1$, $2 \times 10^{-6} < c_1, c_2 < 2$.

The inertia weight W is calculated after every iteration as

$$W = W_{max} - (W_{max} - W_{min}) * \frac{k}{N}, \tag{31}$$

where N is the total number of iterations and k is the current iteration.

The algorithm stops if more than 500 iterations expire without an improvement of g_{best}. The total number of iterations was set to 2000. We use both the fitness function for the analysis. In the initialization step of the PSO algorithm random values are selected for all particles.

Time-domain method: In Table 2a the best results are reported when the time-domain particle design is used. Results are ordered based on lower average MSE. Figure 4a and Figure 4b show the magnitude responses of the two microphones used for the equalization: the blue dashed line is the equalized magnitude response, while the green line is the unequalized magnitude response; the black line is the flat band taken as reference. There are deep notches at 80Ḣz, 110 Hz, almost 200 Hz and over 2 kHz, but compared to unequalized magnitude response, the equalized one is very close to the reference band, this can also be seen from the results in Table 2a, with an average MSE value of 0.22 and a $\overline{\sigma}$ equals to 2.67.

Frequency-domain method: with the frequency-domain particle design the $MinMax$ fitness function achieves better performance. Table 2b shows the results. The average MSE is 0.44 and the $\overline{\sigma}$ is 3.54 In Figure 4c and Figure 4d the magnitude responses of the best configuration are reported, showing a reduction of the average MSE and $\overline{\sigma}$, but the magnitude responses are higher than the desired response at low frequencies (almost 10 dB).

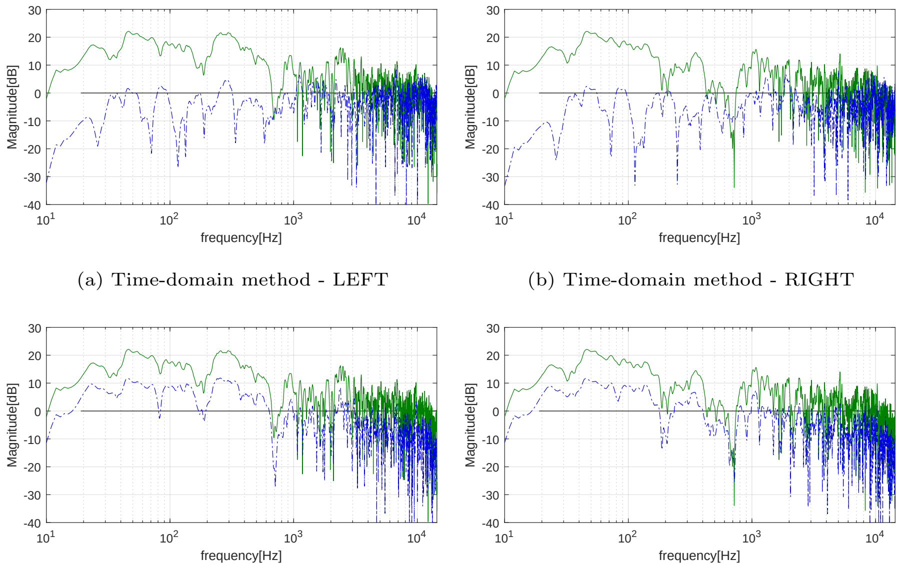
W_{max} - W_{min}	c_1	c_2	Fitness function	Mic 1		Mic 2		Average	
				MSE	σ	MSE	σ	MSE	$\overline{\sigma}$
1.0 - 0.1	2.0	2.0	MSE	0.21	2.67	0.23	2.67	**0.22**	**2.67**
0.1 - 0.01	0.02	0.02	MSE	0.19	2.66	0.25	2.90	0.22	2.78
0.1 - 0.01	0.02	0.02	MSE	0.21	2.72	0.23	2.87	0.22	2.80
0.01 - 0.001	0.2	0.2	MSE	0.24	2.87	0.21	2.74	0.22	2.80
0.01 - 0.001	0.2	0.2	MSE	0.21	2.79	0.24	2.79	0.23	2.84

(a)

W_{max} - W_{min}	c_1	c_2	Fitness function	Mic 1		Mic 2		Average	
				MSE	σ	MSE	σ	MSE	$\overline{\sigma}$
0.1 - 0.01	2.0	2.0	MinMax	0.46	3.50	0.43	3.58	**0.44**	**3.54**
0.01 - 0.001	2.0	2.0	MSE	0.47	3.70	0.42	3.46	0.45	3.58
0.01 - 0.001	2.0	2.0	MSE	0.45	3.76	0.45	3.53	0.45	3.65
0.1 - 0.01	2.0	2.0	MinMax	0.48	3.66	0.43	3.56	0.46	3.61
1.0 - 0.1	2.0	2.0	MSE	0.49	3.62	0.43	3.58	0.46	3.60

(b)

Table 2: Results obtained by PSO: with particles used as filter coefficients (Table 2a); with particles used as magnitude response for each frequency bin (Table 2a).



(a) Time-domain method - LEFT

(b) Time-domain method - RIGHT

(c) Frequency-domain method - LEFT

(d) Frequency-domain method - RIGHT

Figure 4: Audio equalization using PSO: comparison of the magnitude frequency response without equalization (green line), with equalization (dashed blue line). The desired magnitude response is in black.

4.2. Experiments using GSA

The hyperparameters for the GSA case are: $G_{0_{max}}$, $G_{0_{min}}$, A, k_{best}. Since we have used few agents, there is not decrease of them at each repetition, so $A_0 = A$ at every iterations.

The gravitational constant $G(k)$ decreases linearly starting from $G_{0_{max}}$ up to $G_{0_{min}}$:

$$G(k) = G_{0_{max}} - (G_{0_{max}} - G_{0_{min}})\frac{k}{N} \tag{32}$$

where N is the total number of iterations and k is the current iteration.

Time-domain method: Results of the time-domain GSA filter design are reported in Table 3a. In Figure 5a and Figure 5b the equalized magnitude frequency responses of the two microphones are shown. The response is almost flat at low frequency except for notches at 90 Hz and 200 Hz for the first and second output respectively. At high frequencies, significant notches appeared from 12 kHz. Comparing the results with the PSO, GSA has better performance, reaching an average MSE of 0.13.

Frequency-domain method: Table 3b reports the best results using the frequency-domain design method. In this case the performance is lower than PSO. Figure 5c and Figure 5d show the filter magnitude frequency responses. The peaks between 20 Hz and 200 Hz are reduced, however notches are present from 1 kHz.

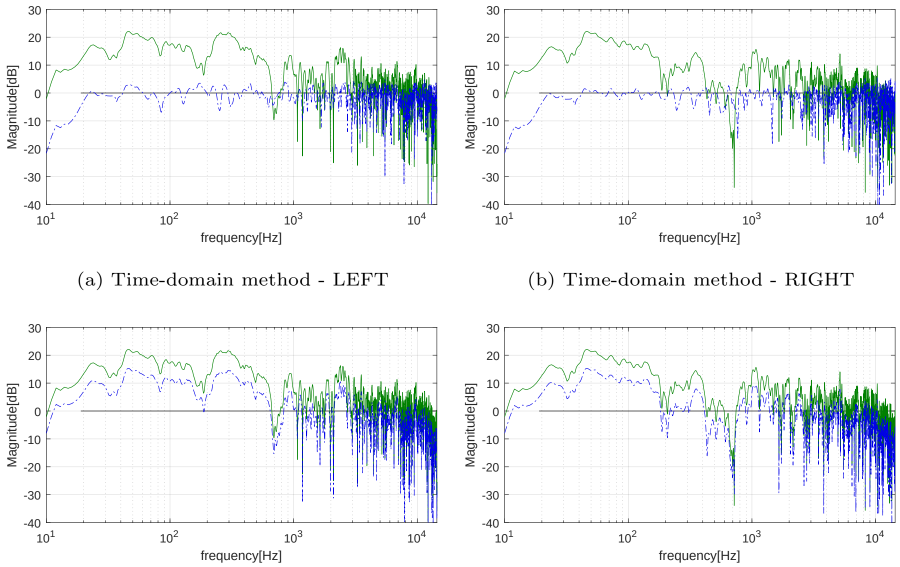
$G_{0_{max}}$ - $G_{0_{min}}$	A	k_{best}	Fitness	Mic 1		Mic 2		Average	
			function	MSE	σ	MSE	σ	MSE	$\overline{\sigma}$
10.0 - 0.01	10	2	MSE	0.12	2.01	0.14	2.34	**0.13**	**2.18**
10.0 - 0.01	5	1	MSE	0.17	2.50	0.20	2.58	0.19	2.54
1.0 - 0.001	10	2	MSE	0.19	2.60	0.20	2.67	0.19	2.63
10.0 - 0.01	10	2	MinMax	0.23	2.87	0.28	3.03	0.25	2.95
1.0 - 0.001	10	2	MinMax	0.26	2.96	0.27	3.11	0.27	3.03

(a)

$G_{0_{max}}$ - $G_{0_{min}}$	A	k_{best}	Fitness	Mic 1		Mic 2		Average	
			function	MSE	σ	MSE	σ	MSE	$\overline{\sigma}$
10.0 - 0.01	5	1	MSE	0.57	3.58	0.40	3.53	**0.49**	3.56
10.0 - 0.01	10	2	MSE	0.59	3.60	0.40	3.40	0.50	**3.50**
1.0 - 0.001	10	2	MSE	0.61	3.63	0.41	3.49	0.52	3.56
1.0 - 0.001	5	1	MSE	0.63	3.58	0.41	3.54	0.52	3.56
0.1 - 0.0001	10	2	MSE	0.62	3.63	0.42	3.50	0.52	3.56

(b)

Table 3: Results obtained by GSA with agents: Table 3a used as filter coefficients, Table 3b used as magnitude response.



(a) Time-domain method - LEFT

(b) Time-domain method - RIGHT

(c) Frequency-domain method - LEFT

(d) Frequency-domain method - RIGHT

Figure 5: Audio equalization using GSA: comparison of the magnitude frequency response without equalization (green line), with equalization (dashed blue line). The desired magnitude response is in black.

4.2.1. Results with varying filter order

The experiments have been performed defining the particles and the agents using the time-domain method: the experiments with particles and agents the frequency-domain technique were not considered because it was seen above that in Section 4.1 and in Section 4.2 the first tipology of particles and filters are better than the second ones.

Experiments using PSO: In Table 4a are presented the best results obtained with the different filter orders: the best results are obtained using the MSE fitness function, moreover it was seen that using filters of order 640, the MSE is the less than the other orders. In Figure 8e and Figure 8f are presented the magnitude responses.

Experiments using GSA: In Table 4b are presented the best results obtained with the different filter orders: the best was occured when it was used filters of 768-*th* order, with a MSE respect to flat band equals to 0.13 and $\overline{\sigma}$ equals to 2.07. The magnitude responses are presented in Figure 8g and Figure 8h.

W_{max} - W_{min}	c_1	c_2	Fitness	Filter	Mic 1		Mic 2		Average	
			function	order	MSE	σ	MSE	σ	MSE	$\overline{\sigma}$
0.01 - 0001	0.02	0.02	MSE	512	0213	2.65	0.22	2.75	0.21	2.70
0.01 - 0001	2.0	2.0	MSE	640	0.22	2.63	0.21	2.71	**0.21**	**2.67**
0.01 - 0001	0.2	0.2	MSE	768	0.19	2.59	0.24	2.89	0.22	2.74
10.0 - 0.1	0.2	0.2	MSE	896	0.20	2.54	0.24	2.76	0.22	2.65
1.0 - 0.1	2.0	2.0	MSE	1024	0.21	2.67	0.23	2.67	0.22	2.67

(a)

$G_{0_{max}}$ - $G_{0_{min}}$	A	k_{best}	Fitness	Filter	Mic 1		Mic 2		Average	
			function	order	MSE	σ	MSE	σ	MSE	$\overline{\sigma}$
10.0 - 0.01	10	2	MSE	512	0.13	2.14	0.15	2.31	0.14	2.23
10.0 - 0.01	10	2	MSE	640	0.12	2.08	0.15	2.39	0.14	2.24
10.0 - 0.01	10	2	MSE	768	0.12	2.02	0.13	2.12	**0.13**	**2.07**
10.0 - 0.01	10	2	MSE	896	0.12	2.04	0.15	2.23	0.14	2.14
10.0 - 0.01	10	2	MSE	1024	0.12	2.01	0.14	2.34	0.13	2.18

(b)

Table 4: Results obtained with: 4a PSO particles with time-domain method; 4b GSA agents with time-domain method.

4.3. Results with baseline methods

4.3.1. Results with Prototyping Design using the frequency deconvolution Method

The first baseline method used is the Prototyping Design using the Frequency Deconvolution explained in Section 3.1. After some attempts, $\beta(\omega)$ has been set to 1.

In Table 5 is presented the obtained results: the average MSE is less than the unequalized response, but the average standard deviation is double. Analyszing the Figure 8a and Figure 8b, it has been seen that the equalization decrease the amplitude at low frequencies, but at high frequencies there are problems caused by $\beta(\omega)$.

Filter	Mic 1		Mic 2		Average	
order	MSE	σ	MSE	σ	MSE	$\overline{\sigma}$
512	0.87	5.71	0.74	5.49	**0.80**	**5.60**
640	0.88	5.75	0.73	5.53	0.81	5.64
768	0.89	5.78	0.74	5.57	0.81	5.68
896	0.90	5.81	0.74	5.61	0.82	5.71
1024	0.91	5.81	0.74	5.61	0.82	5.71

Table 5: Mean and standard deviaton of the equalized magnitude frequency response using the prototyping design using the frequency deconvolution.

4.3.2. Results with the Steepest Descent Method

The other baseline method is the Steepest Descent, explained in Section 3.2. After some attempts, the number of iterations was setted to 250000 and the step size μ was choosen to be 0.00001; as the desired response was chosen the IFFT of the desired frequency response.

The results are presented in Table 6: even if there are many loudspeakers and a very high number of iterations, there is a very large value both of average MSE, $\overline{\sigma}$ and MSE$_w$(0.98, 7.13 and 0.25 respectively). Figure 8c and 8d show that the large error is mostly related to the higher frequency bands, where the method fails to equalize properly.

Filter	Mic 1		Mic 2		Average	
order	MSE	σ	MSE	σ	MSE	$\overline{\sigma}$
512	0.98	7.10	0.98	7.43	0.98	7.26
640	0.98	7.09	0.99	7.17	**0.98**	7.13
768	0.99	7.08	0.99	7.11	0.99	7.09
896	0.99	7.23	0.99	7.24	0.99	7.09
1024	1.02	6.30	1.03	6.47	1.03	**6.39**

Table 6: Mean and standard deviaton of the equalized magnitude frequency response using the Steepest Descent Method.

4.4. Comparison with the Baseline Methods

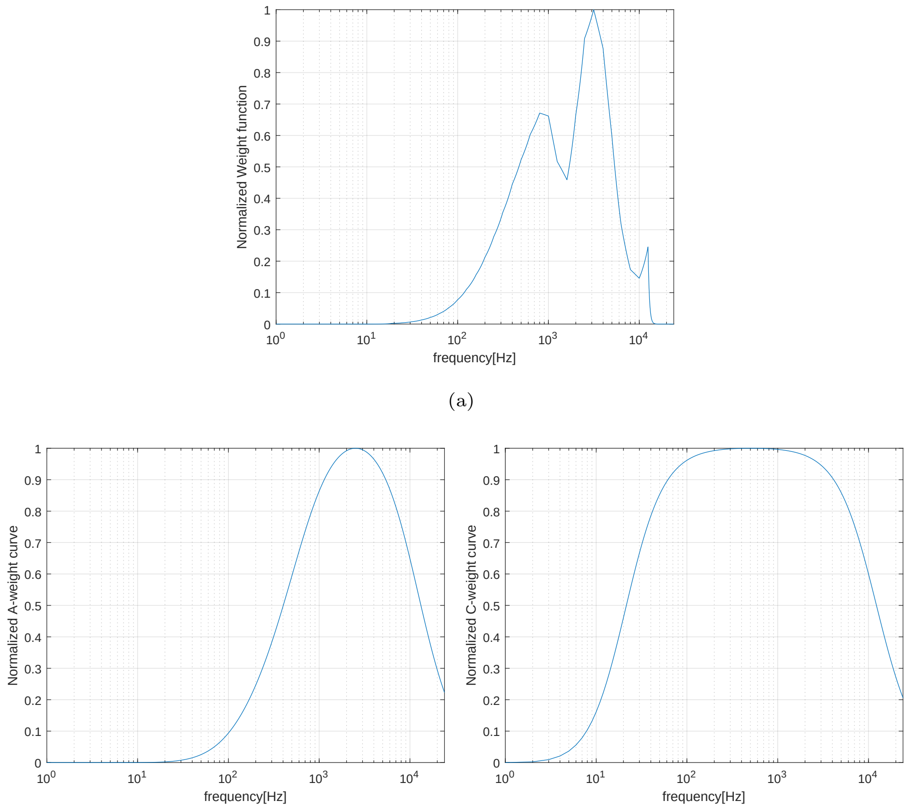
Performances with PSO and GSA have been compared with baseline techniques. In addition to using the MSE described in Eq. (15), the weighted MSE is also used with three different normalized weighing functions: the first normalized function $\Psi_w(\omega)$ derives from the equal-loudness contour, the second $\Psi_A(\omega)$ and the third $\Psi_C(\omega)$ are the A-weighting and C-weigthing curve, properly normalized [55].

The weighting function $\Psi_w(\omega)$ has been determined taking the 60 phons equal loudness contour from the ISO226:2003 standard [56], inverting the curve in the frequency range of interest and normalizing it. The weight function is presented in Figure 6a. The psycho-acoustically weighted MSE_w is used to give more importance to frequency ranges that are strongly perceived by the human ear. A-weighting and C-weigthing curves are also widely employed in acoustics, and are used to evaluate the effect of the error on frequency ranges that are perceptually relevant.

For each weighing function, a weighted MSE has been derived according to the following:

$$MSE_{w,A,C} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \frac{1}{\omega} \left(\sum_{\omega} \Psi_{w,A,C}(\omega) \Big(|H_m(\omega)| - |H_d(\omega)| \Big)^2 \right) \tag{33}$$

where, the three erros: MSE_w, MSE_A, MSE_C are computed from $\Psi_w(\omega)$, $\Psi_A(\omega)$, $\Psi_C(\omega)$, respectively.



(a)

(b)

(c)

Figure 6: Normalized function: (6a) Weight function used for the evaluation, (6b) normzalized A-Weight function used for the evaluation, (6c) normalized C-Weight function used for the evaluation

In Table 7 we compare the results obtained by PSO and GSA, with baseline methods and with no equalization applied. The PSO and GSA algorithms obtain better results in terms of MSE. The average variance, computed bin by bin, also reduces with these two methods, with the time-domain particles and agents providing best results. In addition, it can be seen from the frequency plots that the matching with the target response improves when using the GSA algorithms (see Figure 8 for comparison).

The best results are obtained using GSA, yielding a 0.13 MSE ($\overline{\sigma} = 2.07$).

Similarly, all three perceptually-motivated MSE have been achieved by GSA ($\text{MSE}_w = 0.04$, $\text{MSE}_A = 0.10$ and $\text{MSE}_C = 0.1$), suggesting that the technique performs better also from a perceptual standpoint. Another consideration is that GSA uses just 10 agents, while PSO uses 50 particles, resulting in a smaller number of elements to insert in the search space. As a downside, the GSA has a higher computational cost as later discussed. The obtained filters are symmetrical, thus, each filter has linear phase (see Figure 7).

Algorithm	Average		Weight	A-Weight	C-Weight
	MSE	$\overline{\sigma}$	MSE_w	MSE_A	MSE_C
No Equalization	2.14	3.57	0.90	1.23	1.98
Prototyping Design with Frequency Deconvolution	0.80	5.60	0.25	0.52	0.58
Steepest Descent	0.98	7.13	0.36	0.72	0.72
PSO	0.21	3.83	0.08	0.15	0.16
GSA	**0.13**	**2.07**	**0.04**	**0.10**	**0.10**

Table 7: Comparison of best results obtained with the different algorithms.

4.5. Computational cost analysis

In this section we analyze the computational complexity of all the algorithms under test analyzing the number of floating-point operations (flop) for single iteration. Table 8 shows the number of floating-point operations of the baseline techniques and the evolutionary algorithms using FIR filter of 512-*th* order, $\mathcal{S} = 7$ sources and $\mathcal{M} = 2$ microphones. The result of each method is referred to an individual iteration, except for the Frequency Deconvolution method, that is not iterative, thus, the cost reported in Table 8 is the overall cost. Among the iterative algorithms, the Steepest Descent is the least expensive. Please note, however, that the first iteration requires 360.06 Gflop due to the products $H^T H$ and $H^T d$. These need be calculated only at the first iteration, since the matrix H is not time-varying in this context, differently from on-line adaptive applications of the Steepest Descent algorithm. It also worth noting that this implementation of the Steepest Descent is block-based, while iterative frame-by-frame processing could result in a significant computational cost reduction. GSA has the highest computational cost per iteration overall, which stands as a challenge for a potential real-time implementation.

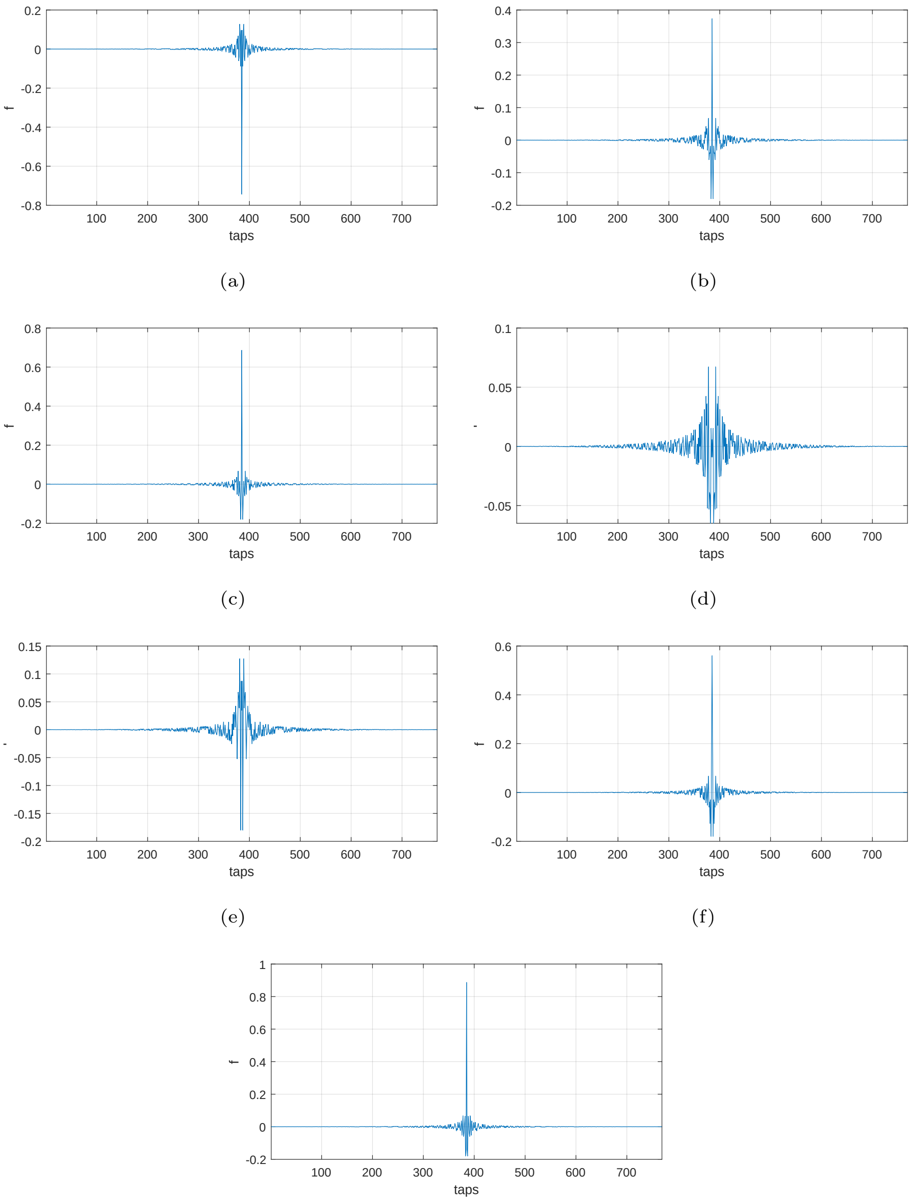
Going into deeper detail: the computational cost of the Frequency Deconvolution method is mainly due to the calculation of the Fourier transform and the smoothing of the frequency responses (see Eq. (17)). For the Steepest Descent algorithm, the largest number of operations is given by the update of the filter coefficients (see Eq. (27)). The high cost of PSO and GSA is related to the fitness function calculation (Eq. (15)) and the update of particles and agents. In particular, the highest cost of the GSA can be attributed to the calculation of the force (see Eq. (9)).

Algorithm	Floating-point operations per iteration
Prototyping Design with Frequency Deconvolution	16.13 G
Steepest Descent	1.89 G
PSO	7.25 G
GSA	14.51 G

Table 8: Number of floating point operations of Prototyping Design with Frequency Deconvolution, Steepest Descent, PSO and GSA. Flops of PSO and GSA are referred to one particle and one agent. Frequency Deconvolution operations is executed in one single iteration.

5. Conclusion

This paper tackles the binaural equalization problem using computational intelligence techniques, namely, the PSO and GSA algorithms. Two approaches are proposed to define particles and agents, either in the time-domain or in the frequency-domain, and fitness functions are proposed. Computer experiments have been performed to evaluate the proposed approach for a specific use-case, the equalization of a car cockpit with several loudspeakers and a binaural mi-



(a)

(b)

(c)

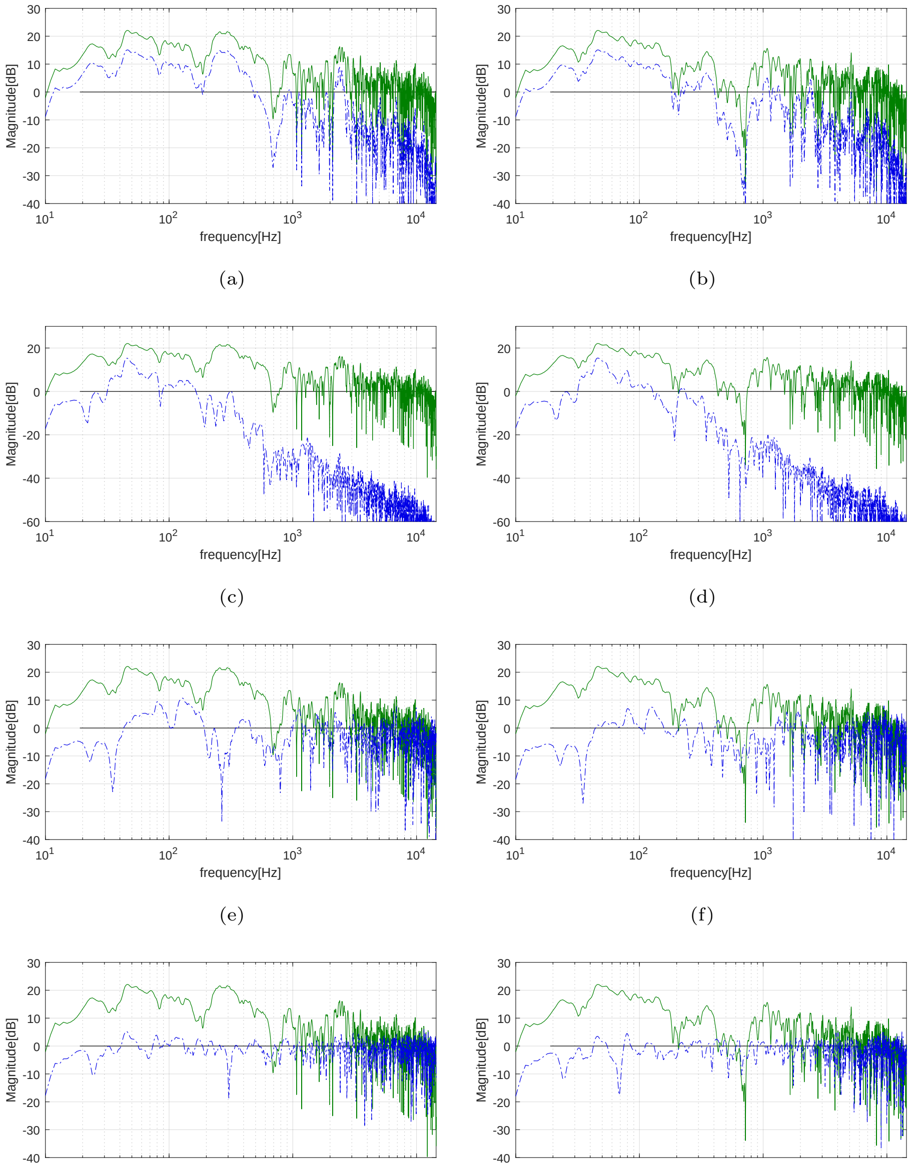
(d)

(e)

(f)

(g)

Figure 7: Filters of the best configuration



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 8: Magnitude frequency response of the two microphones used for the equalization. Green line is the magnitude frequency response without equalization, dashed blue line is the one with equalization, black line is the reference magnitude response. Figure 8a Figure 8b are the magnitude frequency response when it was used the Prototyping Design with Frequency Deconvolution method, Figure 8c Figure 8d are referred to the Steepest Descent Method, Figure 8e Figure 8f are the ones of PSO, Figure 8g Figure 8h are referred to the GSA algorithm.

crophone. The obtained results are compared with two baseline methods: the Steepest Descent method and the Frequency Deconvolution method. The tests seem to confirm the validity of the approach, with results improving over state of the art methods. The best results achieved in the experiments are 6.15 times better in terms of average MSE than the Frequency Deconvolution method, reducing peaks and notches introduced by the sum of the sound sources and their reflections in the cockpit, especially in the low-to-mid frequency range. Reducing the filter order even by an order of 2 does not degrade the performance significantly, suggesting that the method can scale well to embedded digital signal processors with low computational power.

Some issues still need to be addressed. In the high frequency range, although improved, the equalized frequency response still has large variations, resulting in peaks and notches.

Furthermore, the proposed techniques are not currently aimed at real-time equalization. Movements of the listener head can affect the impulse response and decrease the equalization performance. As discussed above, offline methods can be complemented with adaptive equalization techniques for real-time tuning of the filters [12, 57]. As an example, Kalman filters and Steepest Descent could be used for real-time estimation of the impulse responses and their inversion by equalizing filters [6]. Offline methods can be also employed to extend the volume of space to be equalized. For instance, virtual microphone methods [58, 59] can be used. Additionally, in future works we aim at employing machine learning algorithms aiming at further improving audio equalization performance and possibly run these online.

References

[1] G. Accardo, M. El-Kafafy, B. Peeters, F. Bianciardi, D. Brandolisio, K. Janssens, M. Martarelli, Experimental acoustic modal analysis of an automotive cabin, in: Experimental Techniques, Rotating Machinery, and

Acoustics, Volume 8, Springer International Publishing, Cham, 2015, pp. 33–58. doi:10.1007/978-3-319-15236-3_4.

[2] S. Cecchi, L. Palestini, P. Peretti, F. Piazza, A. Carini, Multipoint equalization of digital car audio systems, in: 2009 Proceedings of 6th International Symposium on Image and Signal Processing and Analysis, 2009, pp. 650 – 655. doi:10.1109/ISPA.2009.5297665.

[3] M. Binelli, A. Farina, Digital equalization of automotive sound systems employing spectral smoothed FIR filters, in: Audio Engineering Society Convention 125, 2008.
URL http://www.aes.org/e-lib/browse.cfm?elib=14727

[4] A. Azzali, A. Bellini, A. Farina, E. Ugolotti, Design and implementation of psychoacoustics equalizer for infotainment, in: DSP Implementation Day, Politecnico di Milano, Vol. 23, Citeseer, 2002, Retrieved from: citeseerx.ist.psu.edu.

[5] W. Zhang, A. W. H. Khong, P. A. Naylor, Adaptive inverse filtering of room acoustics, in: 2008 42nd Asilomar Conference on Signals, Systems and Computers, 2008, pp. 788–792. doi:10.1109/ACSSC.2008.5074517.

[6] A. Dagar, S. S. Nitish, R. Hegde, Joint adaptive impulse response estimation and inverse filtering for enhancing in-car audio, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 416–420. doi:10.1109/ICASSP.2018.8462329.

[7] M. Karjalainen, T. Paatero, J. N. Mourjopoulos, P. D. Hatziantoniou, About room response equalization and dereverberation, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005., 2005, pp. 183–186. doi:10.1109/ASPAA.2005.1540200.

[8] S. Neely, J. Allen, Invertibility of a room impulse response., Vol. 66, Acoustical Society of America, 1979, pp. 165–169. doi:10.1121/1.383069.

[9] X. Feng, Y. Zhang, J. Glass, Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition, in: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2014, pp. 1759–1763. doi:10.1109/ICASSP.2014.6853900.

[10] D. S. Williamson, D. Wang, Time-frequency masking in the complex domain for speech dereverberation and denoising, IEEE/ACM transactions on audio, speech, and language processing 25 (7) (2017) 1492–1501. doi:10.1109/TASLP.2017.2696307.

[11] K. Kinoshita, M. Delcroix, H. Kwon, T. Mori, T. Nakatani, Neural network-based spectrum estimation for online WPE dereverberation., in: Proc. Interspeech 2017, 2017, pp. 384–388. doi:10.21437/Interspeech.2017-733.

[12] S. Cecchi, A. Carini, S. Spors, Room response equalization - a review, Applied Sciences 8 (1) (2017) 16. doi:10.3390/app8010016.

[13] J. Mourjopoulos, P. Clarkson, J. Hammond, A comparative study of least-squares and homomorphic techniques for the inversion of mixed phase signals, in: ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 7, 1982, pp. 1858–1861. doi:10.1109/ICASSP.1982.1171447.

[14] J. Durbin, The fitting of time-series models, Revue de l'Institut International de Statistique / Review of the International Statistical Institute 28 (3) (1960) 233–244. doi:10.2307/1401322.

[15] S. J. Elliott, P. A. Nelson, Multiple-point equalization in a room using adaptive digital filters, J. Audio Eng. Soc 37 (11) (1989) 899–907.
URL http://www.aes.org/e-lib/browse.cfm?elib=6063

[16] W. Putnam, D. Rocchesso, J. Smith, A numerical investigation of the invertibility of room transfer functions, in: Proceedings of 1995 Workshop

on Applications of Signal Processing to Audio and Accoustics, 1995, pp. 249–252. doi:10.1109/ASPAA.1995.483001.

[17] V. Välimäki, J. D. Reiss, All about audio equalization: Solutions and frontiers, Applied Sciences 6 (5) (2016) 129. doi:10.3390/app6050129.

[18] J. Mourjopoulos, M. Paraskevas, Pole and zero modeling of room transfer functions, Journal of Sound and Vibration 146 (2) (1991) 281–302. doi:10.1016/0022-460X(91)90764-B.

[19] B. Bank, Audio equalization with fixed-pole parallel filters: An efficient alternative to complex smoothing, J. Audio Eng. Soc 61 (1/2) (2013) 39–49.
URL http://www.aes.org/e-lib/browse.cfm?elib=16666

[20] M. Kallinger, A. Mertins, Room impulse response shortening by channel shortening concepts, in: Conference Record of the Thirty-Ninth Asilomar Conference onSignals, Systems and Computers, 2005., 2005, pp. 898–902. doi:10.1109/ACSSC.2005.1599885.

[21] T. Mei, A. Mertins, M. Kallinger, Room impulse response shortening with infinity-norm optimization, in: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 3745–3748. doi:10.1109/ICASSP.2009.4960441.

[22] S. J. Elliott, L. P. Bhatia, F. S. Deghan, A. H. Fu, M. S. Stewart, D. W. Wilson, Practical implementation of low-frequency equalization using adaptive digital filters, J. Audio Eng. Soc 42 (12) (1994) 988–998.
URL http://www.aes.org/e-lib/browse.cfm?elib=6916

[23] J. N. Mourjopoulos, Digital equalization of room acoustics, J. Audio Eng. Soc 42 (11) (1994) 884–900.
URL http://www.aes.org/e-lib/browse.cfm?elib=6921

[24] Y. Haneda, S. Makino, Y. Kaneda, Common acoustical pole and zero modeling of room transfer functions, IEEE Transactions on Speech and Audio Processing 2 (2) (1994) 320–328. doi:10.1109/89.279281.

[25] A. Mäkivirta, P. Antsalo, M. Karjalainen, V. Välimäki, Modal equalization of loudspeaker - room responses at low frequencies, J. Audio Eng. Soc 51 (5) (2003) 324–343.
URL http://www.aes.org/e-lib/browse.cfm?elib=12226

[26] A. O. Santillàn, Spatially extended sound equalization in rectangular rooms, The Journal of the Acoustical Society of America 110 (4) (2001) 1989–1997. doi:10.1121/1.1401740.

[27] C. S. Pedersen, H. Møller, Sound field control for a low-frequency test facility, in: Audio Engineering Society Conference: 52nd International Conference: Sound Field Control - Engineering and Perception, 2013.
URL http://www.aes.org/e-lib/browse.cfm?elib=16910

[28] B. Bank, Combined quasi-anechoic and in-room equalization of loudspeaker responses, in: Audio Engineering Society Convention 134, 2013.
URL http://www.aes.org/e-lib/browse.cfm?elib=16727

[29] A. A. Al-Shaikhi, A. H. Khan, A. T. Al-Awami, A. Zerguine, A hybrid particle swarm optimization technique for adaptive equalization, Arabian Journal for Science and Engineeringdoi:10.1007/s13369-018-3387-8.

[30] Z. Li, C. Li, Y. Chien, A particle swarm optimization based support vector machine for digital communication equalizers, WSEAS Transactions on Signal Processing 10 (2014) 95–105, Retrieved from: wseas.org/.

[31] D. Diana, S. Rani Joy Vasantha, Enhancement in channel equalization using particle swarm optimization technique, Circuits and Systems 7 (2016) 4071–4084. doi:10.4236/cs.2016.712336.

[32] A. T. Sabin, B. Pardo, A method for rapid personalization of audio equalization parameters, in: Proceedings of the 17th ACM International Con-

ference on Multimedia, MM '09, ACM, New York, NY, USA, 2009, pp. 769–772. doi:10.1145/1631272.1631410.

[33] B. Pardo, D. Little, D. Gergle, 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, ACM, New York, NY, USA, 2012, pp. 13–18. doi:10.1145/2390848.2390852.

[34] D. Reed, A perceptual assistant to do sound equalization, in: Proceedings of the 5th International Conference on Intelligent User Interfaces, IUI 00, ACM, New York, NY, USA, 2000, pp. 212–218. doi:10.1145/325737.325848.

[35] Y. Shaymah, A. Angela, Channel impulse response equalization scheme based on particle swarm optimization algorithm in mode division multiplexing, EPJ Web Conf. 162 (2017) 01023. doi:10.1051/epjconf/201716201023.

[36] D. J. Krusienski, W. K. Jenkins, The application of particle swarm optimization to adaptive IIR phase equalization, in: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, 2004, pp. 693–696. doi:10.1109/ICASSP.2004.1326352.

[37] S. Yogi, K. R. Subhashini, J. K. Satapathy, A PSO based functional link artificial neural network training algorithm for equalization of digital communication channels, in: 2010 5th International Conference on Industrial and Information Systems, 2010, pp. 107–112. doi:10.1109/ICIINFS.2010.5578726.

[38] A. T. Al-Awami, A. Zerguine, L. Cheded, A. Zidouri, W. Saif, A new modified particle swarm optimization algorithm for adaptive equalization, Digital Signal Processing 21 (2) (2011) 195 – 207. doi:10.1016/j.dsp.2010.05.001.

[39] J. R. Mohammed, A study on the suitability of genetic algorithm for adaptive channel equalization, International Journal of Electrical and Computer Engineering (IJECE) 2 (3) (2012) 285–292. doi:10.11591/ijece.v2i3.312.

[40] P. Chang, C. G. Lin, B. Yeh, Inverse filtering of a loudspeaker and room acoustics using time-delay neural networks, The Journal of the Acoustical Society of America 95 (6) (1994) 3400–3408. doi:10.1121/1.409959.

[41] M. A. Martinez Ramirez, 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, Retrieved from: http://dafx2018.web.ua.pt.

[42] N. Agrawal, A. Kumar, V. Bajaj, A new design method for stable IIR filters with nearly linear-phase response based on fractional derivative and swarm intelligence, IEEE Transactions on Emerging Topics in Computational Intelligence 1 (6) (2017) 464–477. doi:10.1109/TETCI.2017.2748151.

[43] I. Kamra, D. S. Sidhu, B. S. Sidhu, Design of digital IIR low pass filter using particle swarm optimization (PSO), International Journal of Scientific Research Engineering and Technology (IJSRET) 3 (2) (2014) 275–280, Retrieved from: http://www.ijsret.org.

[44] F. Foresi, P. Vecchiotti, D. Zallocco, S. Squartini, Designing quasi-linear phase IIR filters for audio crossover systems by using swarm intelligence, in: Audio Engineering Society Convention 144, 2018.
URL http://www.aes.org/e-lib/browse.cfm?elib=19509

[45] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Filter modeling using gravitational search algorithm, Engineering Applications of Artificial Intelligence 24 (1) (2011) 117 – 122. doi:10.1016/j.engappai.2010.05.007.

[46] A. Kalinli, N. Karaboga, Artificial immune algorithm for IIR filter design,

Engineering Applications of Artificial Intelligence 18 (8) (2005) 919 – 929. doi:10.1016/j.engappai.2005.03.009.

[47] 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), 2016, pp. 1–4. doi:10.1109/ICAICT.2016.7991720.

[48] M. Kumari, M. Kumar, R. Saxena, A. Wal, Performance analysis of FIR low pass FIR filter using artificial neural network, International Journal of Engineering Trends and Technology 50 (2017) 58–62. doi:10.14445/22315381/IJETT-V50P210.

[49] X.-H. Wang, Y.-G. He, T.-Z. Li, Neural network algorithm for designing FIR filters utilizing frequency-response masking technique, Journal of Computer Science and Technology 24 (3) (2009) 463–471. doi:10.1007/s11390-009-9237-0.

[50] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948 vol.4. doi:10.1109/ICNN.1995.488968.

[51] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Gsa: A gravitational search algorithm, Information Sciences 179 (13) (2009) 2232 – 2248, special Section on High Order Fuzzy Sets. doi:10.1016/j.ins.2009.03.004.

[52] A. V. Oppenheim, R. W. Schafer, Discrete-Time Signal Processing, 3rd Edition, Prentice Hall Press, Upper Saddle River, NJ, USA, 2009.
URL https://lib.hpu.edu.vn/handle/123456789/29085

[53] A. Mutapcic, S. Kim, S. Boyd, Robust chebyshev fir equalization, in: IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference, 2007, pp. 3074–3079. doi:10.1109/GLOCOM.2007.582.

[54] A. Farina, Advancements in impulse response measurements by sine sweeps, in: Audio Engineering Society Convention 122, 2007.
URL http://www.aes.org/e-lib/browse.cfm?elib=14106

[55] D. M. Howard, J. Angus, Acoustics and psychoacoustics, Routledge, 2017. doi:10.4324/9781315716879.

[56] ISO 226 (2003), Acoustics – normal equal-loudness-level contours, Standard, International Organization for Standardization, Geneva, Switzerland (Aug 2003).

[57] S. Cecchi, L. Palestini, P. Peretti, F. Piazza, F. Bettarelli, R. Toppi, Automotive audio equalization, in: Audio Engineering Society Conference: 36th International Conference: Automotive Audio, 2009.
URL http://www.aes.org/e-lib/browse.cfm?elib=15199

[58] D. Moreau, B. Cazzolato, A. Zander, C. Petersen, A review of virtual sensing algorithms for active noise control, Algorithms 1 (2) (2008) 69–99. doi:10.3390/a1020069.

[59] S. Qi, Y. Wang, Q. Ge, S. Zheng, Active engine order noise equalization control combined with virtual microphones, in: 25th International Congress on Sound and Vibration 2018 (ICSV 25): Hiroshima Calling, Vol. 7, International Institute of Acoustics and Vibration (IIAV), 2018, pp. 4000–4007.
URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85058712689&partnerID=40&md5=edcb03533c000e776ea35a37edf1d4c6