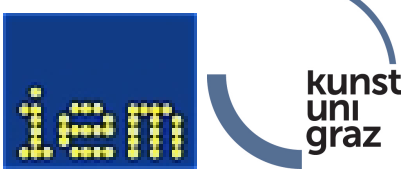
SE Algorithms in Acoustics and Computer Music 02
LV-Nr.17.0033

Automatic Loudspeaker Equalization with parallel filters

Leon Merkel, 11745386
Simon Windtner, 11727158

Lecturer:

Assoc.Prof. DI. Dr.rer.nat. Franz Zotter
Ass.Prof. DI. Matthias Frank, PhD



Institute for Electronic Music and Acoustics
University of Music and Performing Arts
June 6, 2024

Equalization is a crucial tool contributing to high audio quality for playback loudspeakers in their acoustic environment. In our work, we implement a method to automatically calculate filter coefficients to flatten the loudspeaker's frequency response over an averaged listening area. Different spectral resolutions and filter designs are evaluated by using the prototype implementation. The parallel filters impress with the low latency and consistent spectral discretization. The minimum-phase finite impulse response (FIR) filter benefits from an even better resolution at high frequencies; the higher latency does not enable real-time applications. Similar to the FIR filter, the main disadvantages of linear-phase FIR filters are high latency and lower linear frequency resolution. Due to its symmetry, an unwanted pre-ringing effect can also occur. An acoustical measurement of an example loudspeaker in a room verifies the functionality of the designed filters.

Contents

1 Introduction

„The word equalization suggests a process of making things equal, and it is fair to ask what, for whom, and why?“

Floyd E. Toole

1.1 Motivation

As this piece of literature provides an interesting introduction to young professionals, we quote F. Toole's question above from the book [Too09] in which it is discussed several aspects: the influence of loudspeakers [chp. 17 & 18], their surrounding acoustical environment [chp. 16], and in terms of the resulting human auditory perception [chp. 19]. In [chp. 22], equalization strategies for improved sound reproduction in space are summarized.

The question of what target frequency response should be aimed for a loudspeaker is presented in *ISO 2969:2015*, widely known as X-Curves. This standard contains the typical electroacoustic frequency response for motion-picture dubbing theatres (mixing rooms), screening rooms, and indoor theatres. The X-Curves are distinguished by constantly perceived loudness over the frequency [ISO15], often called flat frequency response. These targets might be helpful, especially for reference listening rooms, while different approaches could be more meaningful for other applications.

Parametric filters are commonly employed to flatten arbitrary frequency response to match a desired target frequency response. Parametric filters (peak or shelf) typically shapes certain frequency bands without touching other frequency bands. Equalizers often use a set of four to six of such serially connected first and second-order shelving and peak filters. Shelving filters enhance or attenuate low or high-frequency bands, whereas peak filters are used to enhance or attenuate mid-frequency bands [Zö11]. They offer simple implementations but are also limited in terms of number of bands and adjustability. By increasing the set of parametric filters, the control detail increased, but also nonlinear phase or unequal group delay effects can become audible.

Solutions like Sonaworks [1] - speakers and headphones calibration - as well as smart:EQ 3 from Sonible, [2] which establishes spectral balance, are commercial products, and their principle of operation are not known. As far as the software allows linear-phase FIR filters, a standard n-tap FIR is suspected. The main disadvantage is having a relatively long filter for acceptable frequency resolution, which results in a long delay during convolution.

The filters considered in the implementation of this work are designeed based on third-octave smoothed responses and implemented either as linear-phase filters, minimum-phase filters (both FIR), or the main approach considered here: parallel second-order IIR filters after Balázs Bank [Ban08]. Compared to serially connected IIR filters, parallel filters benefit from fewer multiplications and significantly less delay.

[1] `www.sonaworks.com`
[2] `www.sonible.com/smarteq3`

1.2 Structure of this Work

The first section gives an overview of the theory of the equalization scheme employed. Furthermore, the theoretical fundamentals are described in detail, such as sweep calculation, room impulse response (RIR) measurements, averaging of the RIR, coefficient calculation for pole filters with given bandwidth, calculation for center frequencies for the parallel filter bank. The implementation of parallel filter, linear-phase FIR filter, minimum-phase FIR filter, and mixed-phase FIR filter is presented in the second section. The third section shows several filter results of the implemented loudspeaker equalization tool and their performance in the field. The fourth section gives a conclusion and an outlook of this work.

2 Measurement and Post-Processing

2.1 System Identification

Any electrical or acoustical time-invariant system can be determined by its answer of excitation with a Dirac unit pulse $\delta[n]$. This answer is called the impulse response (IR). Therefore, measuring IRs is a common task in audio signal processing. As far as a perfect Dirac unit pulse can not be perfectly reproduced with a loudspeaker, in the current work, the exponential sweep (ES) is used as an excitation signal due to several advantages. The ES is gainful because of its low crest factor and the possibility of suppressing any harmonic distortion. [MBL07].

2.1.1 Impulse Response

In order to measure the IR of any linear time-invariant system, an excitation signal $x[n]$ is generated in such a way that the signal $x_{\text{inv}}[n]$ can be easily determined by inverting the signal. According to [HCZ09], convolution of the inverted signal and the signal itself leads to a potentially scaled by a factor C and time-shifted unit pulse $\delta[n_0 - n]$, such that

$$\sum_{l=0}^{N-1} x[l]\, x_{\text{inv}}[n-l] = C\, \delta[n_0 - n]\,. \tag{2.1}$$

Thus, the convolution of the measured signal $y[n]$ and the inverse excitation signal will give the scaled and time-shifted impulse response $h[n]$

$$C\, h[n_0 - n] = \sum_{l=0}^{N-1} y[l]\, x_{\text{inv}}[n-l]\,. \tag{2.2}$$

The IR can also be derived in the frequency domain where the convolution becomes multiplication by applying the FFT ($\mathcal{F}$) to the signals

$$H[k] = \mathcal{F}\{y[n]\}\mathcal{F}\{x_{\text{inv}}[n]\}\,. \tag{2.3}$$

Applying the inverse FFT ($\mathcal{F}^{-1}$) to the transfer function $H[k]$ leads to the IR $h[n]$ in the time domain

$$h[n] = \mathcal{F}^{-1}\left\{H[k]\right\}\,. \tag{2.4}$$

2.1.2 Exponential Sweep

In order to derive the exponential sweep (ES), a basic discrete sinus signal can be described as

$$x[n] = \sin(\phi[n]) \quad , \quad \phi[n] = \int_0^{N-1} \omega[n]\mathrm{d}n\,, \tag{2.5}$$

where $\phi[n]$ is the current phase value and N the number of samples. A sinus sweep with exponentially increasing frequency can be derived using the general exponential ansatz for $\omega[n]$

$$\omega[n] = ce^{bn}\,. \tag{2.6}$$

By evaluating the exponential ansatz, from eq. (2.6), with respect to the starting frequency ω_0 and the end frequency ω_1 yields to the phase $\phi[n]$ with its constants c and b

$$c = \omega_0 \quad , \quad b = \frac{1}{N-1} \ln\left(\frac{\omega_1}{\omega_0}\right) . \tag{2.7}$$

Therefore, $x[n]$ can be calculated by evaluating the sinusiod with the phase $\varphi[n]$ that starts with 0 as

$$x[n] = \sin\left[\frac{c}{b}\left(e^{b \cdot n} - 1\right)\right] . \tag{2.8}$$

In order to also end the exponential sweep with zero phase, an optimal starting frequency was calculated as presented in [ZG22]. By having an exemplary $f_s = 48\,\text{kHz}$ and a sweep length of $N = 96000$ samples, a starting frequency of $f_0 = 19.993\,\text{Hz}$ is calculated. The resulting normalized sweep in frequency/time representation and the convolution with its inverse

$$x_{\text{inv}}[n] = \frac{2}{\sum_{k=0}^{N-1} \omega[k]}\, x[-n]\, \omega[-n] \tag{2.9}$$

are depicted in Fig. 2.1 and Fig. 2.2.

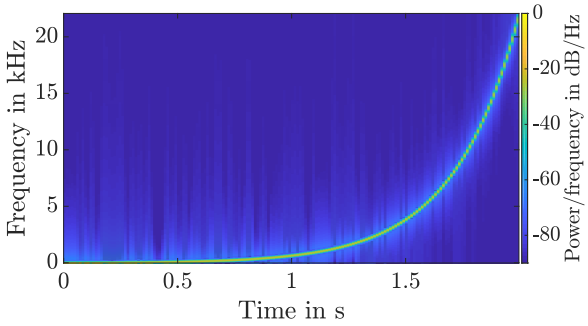


Figure 2.1: exponential sweep

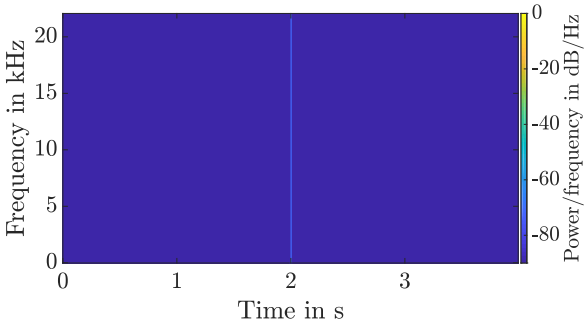


Figure 2.2: convolution of $x[n]$ and $x_{\text{inv}}[n]$

2.2 Post-Processing of the Measurements

2.2.1 Microphone Correction and Magnitude and Phase Averaging

In order to determine an equalization filter for an entire audience area, averaging the magnitude-squared frequency responses of several measurement locations is required. After measurement, the IRs $h_i[n]$ per microphone position i are calculated as described in section 2.1. The measurement microphone's magnitude can be compensated by adding a calibration file. The arbitrary amount of given microphone calibration values will be automatically interpolated to match the sample frequency f_s. The inverse magnitude of the microphone $|H_{\text{mic}}[k]|^{-1}$ is multiplied with each measured magnitude before averaging, respectively as

$$|H_i[k]| = |\mathcal{F}\{h_i[n]\}|\, |H_{\text{mic}}[k]|^{-1} \tag{2.10}$$

The normalized sum of the squared absolute values of the Fourier transformation of each $i \in [1..I]$ measured IR results in the spectral average $H_{\text{av}}[k]$

$$H_{\text{av}}[k] = \sqrt{\frac{1}{I} \sum_{i=1}^{I} |H_i[k]|^2} . \tag{2.11}$$

Subsequent calculations are performed with $H_{\text{av}}[k]$ from eq. (2.11).

2.2.2 Magnitude Smoothing

A common technique to adapt equalization to the spectral resolution of the human auditory system is to smoothen the frequency response by separation in relative bandwidths. Correspondingly, the averaged magnitude $H_{\mathrm{av}}[k]$ is filtered with several n-octave band-pass filters. Thus, the volatile behavior of the magnitude will be smoothed. For every nth-octave center frequency k_{c} we calculate $H_{\mathrm{av,s}}^{(k_{\mathrm{c}})}[k]$ with

$$H_{\mathrm{av,s}}^{(k_{\mathrm{c}})} = \sum_{k=0}^{\mathrm{NFFT}/2} W^{(k_{\mathrm{c}})}[k] H_{\mathrm{av}}[k]\,, \tag{2.12}$$

where $W^{(k_{\mathrm{c}})}[k]$ denotes a normalized $\cos^2$-filter-bank with the respective centre frequencies k_{c}

$$W^{(k_{\mathrm{c}})}[k] = \frac{\widetilde{W}^{(k_{\mathrm{c}})}[k]}{\sum_m \widetilde{W}^{(k_{\mathrm{c}})}[m]} \quad \text{with} \quad \widetilde{W}^{(k_{\mathrm{c}})}[k] = \cos^2\left[\frac{\pi}{2}\mathrm{Clip}\left\{\mathrm{ld}\left(\frac{k}{k_{\mathrm{c}}} n_{\mathrm{oct}}\right), -1, 1\right\}\right]. \tag{2.13}$$

The resulting filter bank for third octaves is shown in Fig. 2.3. In order to map the response back to the initial frequency resolution of $\mathrm{NFFT}/2+1$ bins, the smoothed values are multiplied with the whole filter bank consisting of N_{oct} frequency bands, such that

$$H_{\mathrm{av,s}}[k] = \sum_{k_c=k_{c,0}}^{k_{c,N_{\mathrm{oct}}}} W^{(k_{\mathrm{c}})}[k] H_{\mathrm{av,s}}^{(k_{\mathrm{c}})}\,. \tag{2.14}$$

A stronger overlap of the frequency bands can be considered for improved smoothing, using a higher density of the center frequencies than an nth-octave.

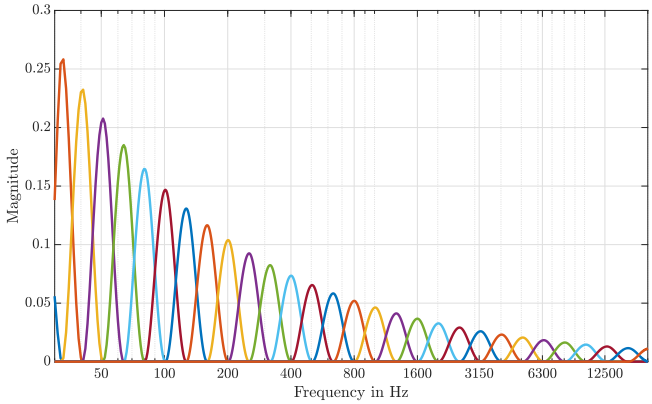


Figure 2.3: Normalized $\cos^2$ third-octave filterbank $W^{(k_{\mathrm{c}})}[k]$.

2.2.3 Minimum-Phase Version of IR

In order to obtain an IR from the averaged magnitude $H_{\mathrm{av,s}}[k]$, the minimum phase version of the magnitude is calculated using the complex cepstrum. The following steps lead to the averaged minimum phase version of the IR $h_{\mathrm{min}}[n]$, which is written as $h_{\mathrm{s}}[n]$ for simplicity in the next chapter:

$$\begin{aligned} h[n] &= \mathcal{F}^{-1}\{\ln(|H_{\mathrm{av,s}}[k]|)\} \\ h_c[n] &= \begin{cases} h[n] & \text{for} \quad n = 0 \\ 2\,h[n] & \text{for} \quad 1 \le n \le \frac{\mathrm{NFFT}}{2} - 1 \\ h[n] & \text{for} \quad n = \frac{\mathrm{NFFT}}{2} \\ 0 & \text{for} \quad \frac{\mathrm{NFFT}}{2} + 1 \le n \le \mathrm{NFFT} \end{cases} \\ h_{\mathrm{min}}[n] &= \mathcal{F}^{-1}\{e^{\mathcal{F}\{h_c[n]\}}\} \end{aligned} \tag{2.15}$$

3 Equalization Filter Design

The current section describes the equalization filter design for parallel filter, minimum-phase FIR filter, linear-phase FIR filter, and mixed-phase FIR filter in detail.

3.1 Parallel IIR Filter

3.1.1 General Filter Structure

The general form of the parallel filter can be denoted as the sum of second-order IIR-filters in the z-domain [OSB99, chp.6] as

$$H(z) = \sum_{k=1}^{K} \frac{d_{k,0} + d_{k,1} z^{-1}}{1 + a_{k,1} z^{-1} + a_{k,2} z^{-2}} + \sum_{m=0}^{M} b_m z^{-m}\,, \tag{3.1}$$

where K is the number of second-order IIR sections, and M is the order of the finite impulse response (FIR) sections. The filter coefficients d_k, a_k, and b_m have to be determined such that the filter exhibits bandpass character of pre-defined bandwidth around center frequencies. The structure is depicted in Fig. 3.1. Based on the concept of parallel filters by *B. Bank*, the idea is to design a parallel filter

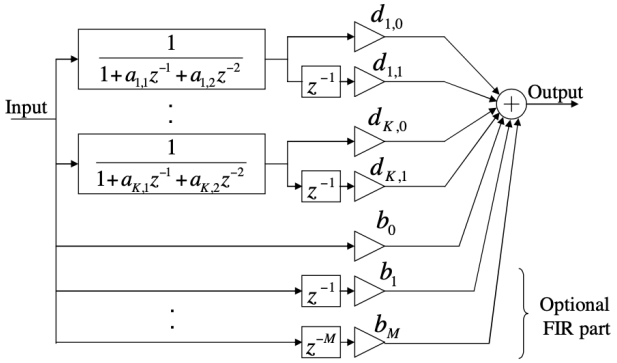


Figure 3.1: structure of parallel filter [Ban13b]

bank of these bandpasses and then find the weights that minimize the error between the equalized response $h_{\mathrm{eq}}[n]$ and the target frequency response $h_{\mathrm{t}}[n]$. It yields a system identification problem with output error minimization: the input of the parallel filter is the system response $h_{\mathrm{s}}[n]$, and the tool should estimate the filter parameters; thus, the output $h_{\mathrm{eq}}[n]$ matches best with the target response $h_{\mathrm{t}}[n]$. Therefore, the transfer functions from eq. (3.1) can rewritten as

$$H_{\mathrm{eq}}(z) = H(z) H_{\mathrm{s}}(z) = \sum_{k=1}^{K} \frac{d_{k,0} + d_{k,1} z^{-1}}{a_{k,0} + a_{k,1} z^{-1} + a_{k,2} z^{-2}} H_{\mathrm{s}}(z) + \sum_{m=0}^{M} b_m z^{-m} H_{\mathrm{s}}(z)\,. \tag{3.2}$$

3.1.2 Pole Positioning

In the first step of the filter design, the filter frequencies f_k are set to a logarithmic frequency scale, as the frequency resolution of the human ear also exhibits logarithmic behavior. Therefore, the lowest

and highest filter frequencies (f_{low}, f_{high}) are queried by the user. The number of poles per octave (n_{oct}) can also be chosen from 1 to 24. An example filter frequency vector with $f_{\text{low}} = 100\,\text{Hz}$, $f_{\text{high}} = 2000\,\text{Hz}$ and $n_{\text{oct}} = 3$ with frequencies in Hz reads as

$$f_k = [100, 125, 160, 200, 250, 315, 400, 500, 630, 800, 1000, 1250, 1600, 2000]^T\,. \tag{3.3}$$

In order to calculate the pole frequencies θ_k from the filter frequencies, the bandwidth $\Delta\theta_k$ needs to be determined. $\Delta\theta_k$ can be directly calculated by hand over the Q-factor of each filter frequency f_k respectively by

$$\Delta\theta_k = \frac{\theta_k}{Q_k}\,, \qquad \text{where} \qquad \theta_k = \frac{2\pi f_k}{f_s} \qquad \text{with sampling frequency } f_s\,. \tag{3.4}$$

For simplification, we calculate $\Delta\theta_k$ concerning the neighbouring frequencies with

$$\Delta\theta_k = \left(\frac{\theta_{k+1} - \theta_{k-1}}{2}\right)\,. \tag{3.5}$$

Thus, the corresponding poles p_k are determined by

$$p_k = e^{\frac{-\Delta\theta_k}{2}} e^{\pm j\theta_k}\,, \tag{3.6a}$$

with the pole frequencies θ_k in radians [ban13a]. From filter theory, we can calculate the coefficients $a_{k,0}, a_{k,1}, a_{k,2}$ for a second-order all-pole filter directly. If we filter any signal using one pole p_k and its complex conjugate p_k^*, the transfer function reads as

$$H_k(z) = \frac{1}{(1 - p_k\, z^{-1})(1 - p_k^*\, z^{-1})} = \frac{1}{1 + 2\mathfrak{Re}\left\{p_k\right\} z^{-1} + |p_k|^2\, z^{-2}}\,, \tag{3.7}$$

so that the filter coefficients result in

$$a_{k,0} = 1\,, \qquad a_{k,1} = 2\,\mathfrak{Re}\left\{p_k\right\}\,, \qquad a_{k,2} = |p_k|^2\,. \tag{3.8}$$

3.1.3 Problem Formulation in Matrix Notation

After determining the denominator coefficients from the poles as described in the previous subsection, the problem becomes linear in its free parameters $d_{k,0}, d_{k,1}$ and b_m. The general form of a second-order parallel filter, equation 3.1, in matrix notation can be written as

$$\mathbf{h}_{\text{eq}} = \mathbf{M}\mathbf{d}\,, \tag{3.9}$$

where $\mathbf{d}$ is a column vector that contains the free parameters,

$$\mathbf{d} = [d_{1,0}, d_{1,1}, \ldots, d_{K,0}, d_{K,1}, b_0, \ldots, b_M]^T\,, \tag{3.10}$$

and $\mathbf{M}$ is the modeling matrix consisting the measured IR $h_{\text{s}}[n]$ filtered with all K all-pole filters $h_k[n]$ respectively, resulting in $s_k[n]$ and $S_k[z]$ with

$$\begin{aligned} s_k[n] &= \sum_{i=0}^{N} h_k[i] h_{\text{s}}[n-i] = h_k[n] * h_{\text{s}}[n]\,, \\ S_k[z] &= H_k[z] H_{\text{s}}[z] = \sum_{k=1}^{K} \frac{1}{a_{k,0} + a_{k,1} z^{-1} + a_{k,2} z^{-2}} H_{\text{s}}(z)\,. \end{aligned} \tag{3.11}$$

In order to fulfill eq. (3.2) using a matrix multiplication that holds the equation, we need to delay each resulting $s_k[n]$ to match the one sample delay of every second free nominator parameters $d_{k,1}$. Inserting $S_k[z]$ we can rewrite eq. (3.2) to

$$H_{\mathrm{eq}}(z) = \sum_{k=1}^{K} \left(d_{k,0} + d_{k,1} z^{-1}\right) S_k[z] + \sum_{m=0}^{M} b_m z^{-m} H_{\mathrm{s}}(z)\,. \tag{3.12}$$

The last columns are the measured IR, again delayed respectively corresponding for the FIR-coefficients b_m, so that we obtain for $M = 1$

$$\mathbf{M} = \begin{bmatrix} s_0[0] & 0 & s_1[0] & 0 & \dots & s_K[0] & 0 & h_{\mathrm{s}}[0] & 0 \\ s_0[1] & s_0[0] & s_1[1] & s_1[0] & \dots & s_K[1] & s_K[0] & h_{\mathrm{s}}[1] & h_{\mathrm{s}}[0] \\ s_0[2] & s_0[1] & s_1[2] & s_1[1] & \dots & s_K[2] & s_K[1] & h_{\mathrm{s}}[2] & h_{\mathrm{s}}[1] \\ \vdots & s_0[2] & \vdots & s_1[2] & \dots & \vdots & s_K[2] & \vdots & h_{\mathrm{s}}[2] \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ s_0[N] & s_0[N-1] & s_1[N] & s_1[N-1] & \dots & s_K[N] & s_K[N-1] & h_{\mathrm{s}}[N] & h_{\mathrm{s}}[N-1] \end{bmatrix}. \tag{3.13}$$

3.1.4 Weight Estimation using Least-Squares Algorithm

In the next step, the task is to determine the optimum parameters $\mathbf{d}_{\mathrm{opt}}$ in a way that $\mathbf{h}_{\mathrm{t}} = \mathbf{M}\mathbf{d}_{\mathrm{opt}}$ is closest to the target impulse response $\mathbf{h}_{\mathrm{t}}$, with a flat frequency response

$$\mathbf{h}_{\mathrm{t}} = [h_{\mathrm{t}}[0], \dots, h_{\mathrm{t}}[N]]^T = [1, 0, \dots, 0]^T\,. \tag{3.14}$$

In order to prevent over-compensation of the low-frequency range, the target response $h_{\mathrm{t}}[n]$ is filtered with a four-order high-pass with a variable cut-off frequency. The transfer function of the filter reads as

$$H_{\mathrm{HP},4}[z] = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}\,, \qquad \text{and} \qquad H_{\mathrm{t,HP}}[z] = H_{\mathrm{t}}[z] H_{\mathrm{HP},4}[z]. \tag{3.15}$$

In the time-domain the mean square error e_{LS} of the filtered target response $h_{\mathrm{t,HP}}[n]$ is evaluated with

$$\begin{aligned} e_{\mathrm{LS}} = \sum_{n=0}^{N} |h_{\mathrm{eq}}[n] - h_{\mathrm{t,HP}}[n]|^2 &= (\mathbf{h}_{\mathrm{eq}} - \mathbf{h}_{\mathrm{t,HP}})^H (\mathbf{h}_{\mathrm{eq}} - \mathbf{h}_{\mathrm{t,HP}}) = (\mathbf{M}\mathbf{d} - \mathbf{h}_{\mathrm{t,HP}})^H (\mathbf{M}\mathbf{d} - \mathbf{h}_{\mathrm{t,HP}}) \\ &= \mathbf{d}^H \mathbf{M}^H \mathbf{M}\mathbf{d} - 2\mathbf{d}^H \mathbf{M}^H \mathbf{h}_{\mathrm{t,HP}} + \mathbf{h}_{\mathrm{t,HP}}^H \mathbf{h}_{\mathrm{t,HP}}\,, \end{aligned} \tag{3.16}$$

and the optimal coefficients $\mathbf{d}_{\mathrm{opt}}$ are determined by searching the minimum of the derivative of e_{LS} in respect to $\mathbf{d}$, such that

$$\begin{aligned} \nabla_{\mathbf{d}} (e_{\mathrm{LS}}) &\stackrel{!}{=} 0 \\ 2\mathbf{M}^H \mathbf{M}\mathbf{d}_{\mathrm{opt}} - 2\mathbf{M}^H \mathbf{h}_{\mathrm{t,HP}} &= 0 \\ \mathbf{M}^H \mathbf{M}\mathbf{d}_{\mathrm{opt}} &= \mathbf{M}^H \mathbf{h}_{\mathrm{t,HP}} \\ \mathbf{d}_{\mathrm{opt}} &= \left(\mathbf{M}^H \mathbf{M}\right)^{-1} \mathbf{M}^H \mathbf{h}_{\mathrm{t,HP}}\,. \end{aligned} \tag{3.17}$$

The resulting well-known least squares (LS) solution contains the Moore-Penrose pseudo-inverse $\mathbf{M}^\dagger = \left(\mathbf{M}^H \mathbf{M}\right)^{-1} \mathbf{M}^H$ so that we can simplify eq. (3.17) to

$$\mathbf{d}_{\mathrm{opt}} = \mathbf{M}^\dagger \mathbf{h}_{\mathrm{t,HP}}\,. \tag{3.18}$$

The resulting filter-coefficients $d_{\mathrm{k},0}, d_{\mathrm{k},1}, a_{\mathrm{k},0}, a_{\mathrm{k},1}, a_{\mathrm{k},2}, b_{\mathrm{m}}$ can then be used to filter directly in real-time using the parallel filters difference equation or compute an equalizing IR $h[n]$ by the filtering of a unit Dirac delta $h_{\mathrm{t}}[n]$ with the parallel filter as

$$a_{\mathrm{k},0} h[n] = \sum_{k=1}^{K} \left(d_{\mathrm{k},0} h_{\mathrm{t}}[n] + d_{\mathrm{k},1} h_{\mathrm{t}}[n-1] - a_{\mathrm{k},1} h[n-1] - a_{\mathrm{k},2} h[n-2]\right) + \sum_{m=0}^{M} b_{\mathrm{m}} h_{\mathrm{t}}[n-m]\,. \tag{3.19}$$

3.2 Minimum-Phase FIR Filter

A straightforward minimum-phase FIR implementation is performed using the minimum-phase version of the measurement's inverted, smoothed, and averaged magnitude response. The averaged and smoothed magnitude $H_{\mathrm{av,s}}[k]$ from eq. (2.14) is inverted and filtered with the target magnitude $H_{\mathrm{t}}[k]$, and its minimum phase version is calculated as described in section 2.2.3 as

$$
\begin{aligned}
h_{\mathrm{inv}}[n] &= \mathcal{F}^{-1}\left\{\ln\left(\frac{|H_{\mathrm{t}}[k]|}{|H_{\mathrm{av,s}}[k]|}\right)\right\} \\
h_{\mathrm{c,min,FIR}}[n] &= \begin{cases} h_{\mathrm{inv}}[n] & \text{for} \quad n = 0 \\ 2\,h_{\mathrm{inv}}[n] & \text{for} \quad 1 \leq n \leq \frac{\mathrm{NFFT}}{2} - 1 \\ h_{\mathrm{inv}}[n] & \text{for} \quad n = \frac{\mathrm{NFFT}}{2} \\ 0 & \text{for} \quad \frac{\mathrm{NFFT}}{2} + 1 \leq n \leq \mathrm{NFFT} \end{cases} \\
h_{\mathrm{min,FIR}}[n] &= \mathcal{F}^{-1}\left\{e^{\mathcal{F}\{h_{\mathrm{c,min,FIR}}[n]\}}\right\},
\end{aligned}
\tag{3.20}
$$

where $h_{\mathrm{min,FIR}}[n]$ contains the FIR coefficients of the equalizing minimum-phase IR. Further truncation of the IR to L coefficients bound the frequency resolution. Using a sampling frequency of $f_s = 44.1\,\mathrm{kHz}$ and a $L = 4096$ tap length filter results in a theoretical frequency resolution of

$$
\Delta f_{\mathrm{min,FIR}} = \frac{f_s}{L} = \frac{44.1\,\mathrm{kHz}}{4096} = 10.77\,\mathrm{Hz}\,, \tag{3.21}
$$

which almost corresponds to the third-octave resolution at low frequencies as used for the parallel filter ($\Delta f_{\mathrm{3th/Oct,LF}} = 40\,\mathrm{Hz} - 31.5\,\mathrm{Hz} = 8.5\,\mathrm{Hz}$). Having an initial coarse smoothing of $H_{\mathrm{av,s}}[k]$, this resolution may not be necessary, and the filter can be designed with a shorter length. This filter length results in a minimal simply convolution delay $\Delta t_{\mathrm{min,FIR}}$ of

$$
\Delta t_{\mathrm{c,FIR}} = \frac{L}{f_s} = \frac{4096}{44.1\,\mathrm{kHz}} = 92.88\,\mathrm{ms}\,, \tag{3.22}
$$

which can be neglected by implementing the filtering process in an FFT-based *Overlap-Save*-procedure presented in [PCR+11] because the IR is a minimum phase, and its prominent peak is at sample location zero. Fig. 3.2 compares the $L = 4096$ tap FIR filter with a $K = 28$ poles parallel IIR filters with $M = 1$, both filtered with an exemplary $h_{\mathrm{s}}[n]$. Besides a slight offset in magnitude, after filtering

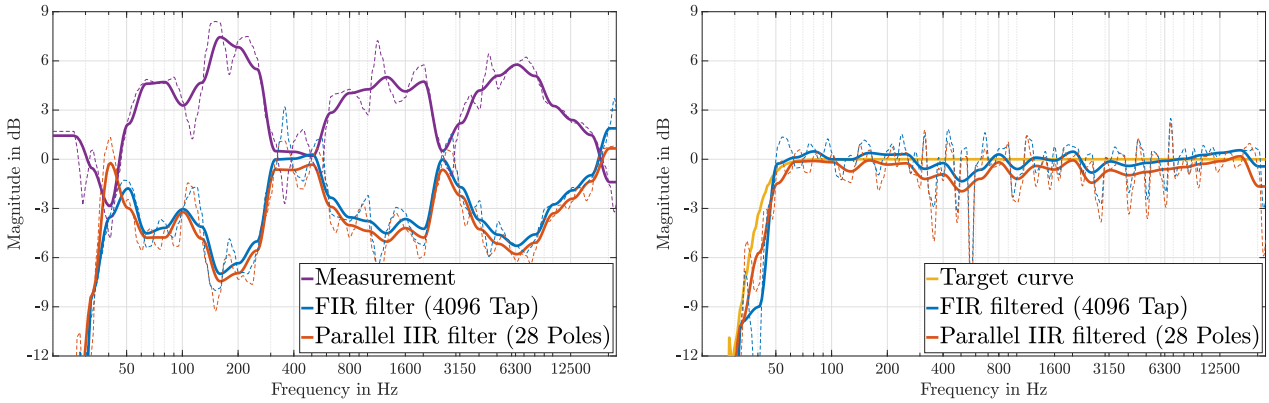


Figure 3.2: Comparison of 4096 Tap FIR filter and 28 Pole Parallel IIR filter.

with the initial measured IR, the resulting filter and magnitude show almost the same behavior.

3.3 Linear-Phase FIR Filter

Another common demand for equalizing loudspeakers' frequency response in a room, preserving the initial phase, is linear-phase FIR filtering. Similar to the calculation of minimum-phase filters, $H_{\mathrm{av,s}}[k]$ gets inverted and filtered with the target response $H_{\mathrm{t}}[k]$. As the filter should feature zero-phase behavior, any phase information is neglected by only considering the magnitudes, and an inverse FT leads to

$$\hat{h}_{\mathrm{lin,FIR}}[n] = \mathcal{F}^{-1}\left\{\left|\frac{H_{\mathrm{t}}[k]}{H_{\mathrm{av,s}}[k]}\right|\right\} . \tag{3.23}$$

As the FT shows a mirrored frequency spectrum, the inverse FT mirrors its discrete time values. To obtain a symmetric IR, which is mandatory to achieve linear-phase behavior [OSB99], the result $\hat{h}_{\mathrm{lin,FIR}}[n]$ has to be circularly shifted by $\frac{\mathrm{NFFT}}{2}$ such

$$h_{\mathrm{lin,FIR}}[n] = \begin{cases} \hat{h}_{\mathrm{lin,FIR}}[n] & \text{for} \quad \frac{\mathrm{NFFT}}{2} \leq n \leq \mathrm{NFFT} - 1 \\ \hat{h}_{\mathrm{lin,FIR}}[n] & \text{for} \quad 0 \leq n \leq \frac{\mathrm{NFFT}}{2} - 1 \end{cases} . \tag{3.24}$$

Further truncation is then applied symmetrically around the most prominent peak to maintain the linear-phase conditions to reduce latency with the drawback of decreased frequency resolution, which can be determined with

$$\Delta f_{\mathrm{lin,FIR}} = \frac{f_s}{L/2} , \tag{3.25}$$

and having the same filter length, the linear phase filter shows twice the frequency resolution compared to the mixed phase filter. Fig. 3.3 compares the impulse, magnitude, and phase responses of a linear phase FIR filter of 8193 samples with a minimum phase FIR filter of 4096 samples for comparable frequency resolution. As expected, the prominent peak of the linear-phase IR is located at exactly half of the overall samples, and the IR shows a symmetric behavior. The magnitude responses of both

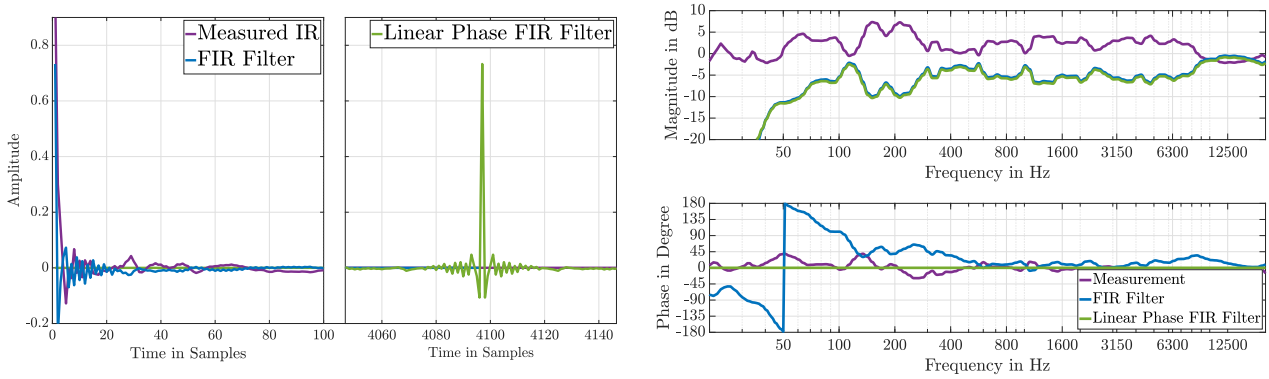


Figure 3.3: Comparison of linear-phase FIR with normal FIR filter

filters show the same trace, while the phase response of the linear-phase filter is exceptionally zero. The disadvantage of the linear-phase filter is the significantly higher latency. The phase shift caused by the FIR filter is relatively tiny, and the compensation of the measured phase response can be seen as an advantage. The linear-phase filter is a reasonable option only for purposes where linear-phase is mandatory.

3.4 Mixed-Phase FIR Filter

In order to benefit from both the linear-phase behavior (sec.3.3) and minimum latency (sec.3.2), a mixed version is also considered. Therefore, the higher frequencies are filtered linear-phase because short filters are feasible and do not exceed typical buffer sizes. The lower frequencies are then filtered

in the minimum phase to avoid introducing high latency of long enough symmetric filters that show sufficient frequency resolution. The easiest way of designing such a filter is first calculating a linear-phase filter $h_{\mathrm{lin}}[n]$ with proper truncation (e.g., $n_{\mathrm{lin}} = 256+1$ samples causing a latency of 129 samples and frequency resolution of $\Delta f_{\mathrm{lin,FIR}} = 345\,\mathrm{Hz}$) as shown in sec. 3.3. Subsequently, filter the measured and averaged IR $h_{\mathrm{s}}[n]$ with the linear-phase filter as

$$H_{\mathrm{lin,meas}}[k] = H_{\mathrm{lin}}[k]\, H_{\mathrm{s}}[k]\,. \tag{3.26}$$

Afterwards, calculate a minimum-phase filter $h_{\mathrm{min,mix}}[n]$ as described in sec. 3.2 using the result of the filtering process $h_{\mathrm{lin,meas}}[n]$ not the measured IR and calculate the final mixed-phase filter by filtering $h_{\mathrm{min,mix}}[n]$ with $h_{\mathrm{lin}}[n]$ such that we obtain

$$h_{\mathrm{mixed}}[n] = \mathcal{F}^{-1}\left\{H_{\mathrm{lin}}[k]\, H_{\mathrm{min,mix}}[k]\right\}\,, \tag{3.27}$$

with its IR $h_{\mathrm{mixed}}[n]$. The resulting filter shows a minimum delay of $\frac{n_{\mathrm{lin}}}{2}+1 = 129$ samples and having the advantage to be linear phase above 2 kHz.

3.5 Computational Cost

The main difference in evaluating the computational cost of implementing the filters on any processing device is the need for a convolution for the FIR filters. In contrast, IIR can be directly implemented using the difference equation (3.19). A convolution implemented conventional as in eq. (2.1) needs $\mathcal{O}(\,NL\,)$ multiplications per block, where N is the block size of samples and L is the length of the IR. For long IRs, the implementation in the frequency domain is computationally more efficient since it needs $\mathcal{O}(\,3\,(N+L-1)\log_2(N+L-1) + (N+L-1)\,)$ multiplications per block, resulting from two FFTs with $\mathcal{O}(\,(N+L-1)\log_2(N+L-1)\,)$ one for the signal and the other for the IR, a matrix multiplication with $\mathcal{O}(\,N+L-1)$ and an IFFT with $\mathcal{O}(\,(N+L-1)\log_2(N+L-1)\,)$. Compared to parallel IIR filters, the costs are $\mathcal{O}(\,N(5K+(M+1))\,)$ where K is the number of parallel IIR sections and M the number of serial FIR sections.

For example, assuming a design goal of a third octave and a block size of $N = 512$, the filter length for the minimum phase FIR filter needs to be at least $L = 4096$. The third octave parallel IIR filter requires a total of $K = 28$ poles and $M = 2$ FIR sections such that we summarize the computational costs with:

Minimum Phase FIR: 172800 multiplications per block, 338 multiplications per sample
Parallel IIR: 73216 multiplications per block, 143 multiplications per sample

3.6 Latency

In modern block-wise computations using *Overlap-Save*-algorithms for long IRs, the latency mainly depends on the utilized block size and, therefore, is not necessarily considerable for the minimum phase FIR and parallel IIR filters. By demanding linear phase even in the mixed or linear phase FIR filters, latency is not avoidable and arises by the number of samples before the prominent peak. Concerning linear phase filters, the latency is precisely half of the total length of the IR, and in mixed-phase filters, it is specified by half of the length of the linear phase part of the IR.

4 Software Implementation

4.1 Structure

The current chapter gives an overview of the implementation structure, shown in Figure 4.1. The implementation was done in two steps. First, a general implementation was realized in *MATLAB*. In

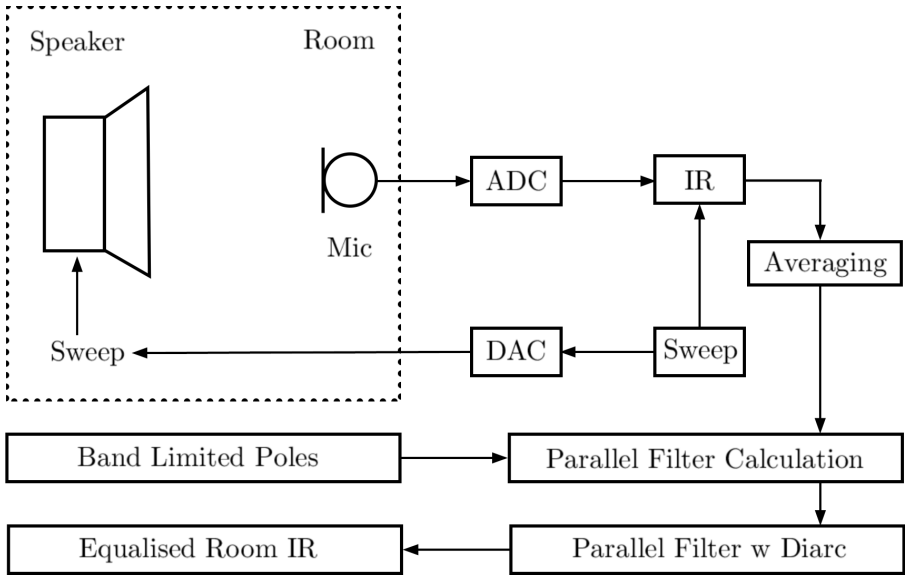


Figure 4.1: Implementation structure

order to have better and smarter handling, an app was created via the *MATLAB* App - Designer. In general, the operation procedure can be described as follows:

1. The DUT plays back several exponential sweeps, created as in sec. 2.1.2, and a microphone measures them at different locations.
2. According to sec. 2.1.1, the different IRs are calculated.
3. The post-processing procedure of averaging, smoothing, and calculating the minimum phase version is performed as presented in chp. 2.2.
4. The logarithmic band limited poles are determined as in sec. 3.1.2.
5. The modeling matrix $\mathbf{M}$ comprises the all-pole filtered average measured IR calculated corresponding to sec. 3.1.3.
6. The LMS algorithm calculates the optimal nominator weights as described in sec. 3.1.4.
7. Optionally, a Dirac impulse $\delta[n]$ can be filtered with the parallel filter, resulting in an IR with rectifying properties.

4.2 Operation Procedure

In order to get started with the app, the following initial settings must be set:

- Sampling Rate in Hz
- Available input device
- Available output device
- Estimated reverberation time (RT60) in seconds
- Sweep length in seconds
- Number of IR measurements

The sampling rate should match the sampling rate of the interface. The dropdown menu will automatically display the available input and output interfaces found by the system, which the user can choose from. In the next step, the user has to enter the estimated RT60 of the room. Furthermore, the length of the sweep can be chosen, and the amount of different IR measurements at several locations. By clicking on ≪**[read Settings]**≫ the defined initial settings are loaded into the software. Fig. 4.2 shows the app's graphical user interface (GUI).

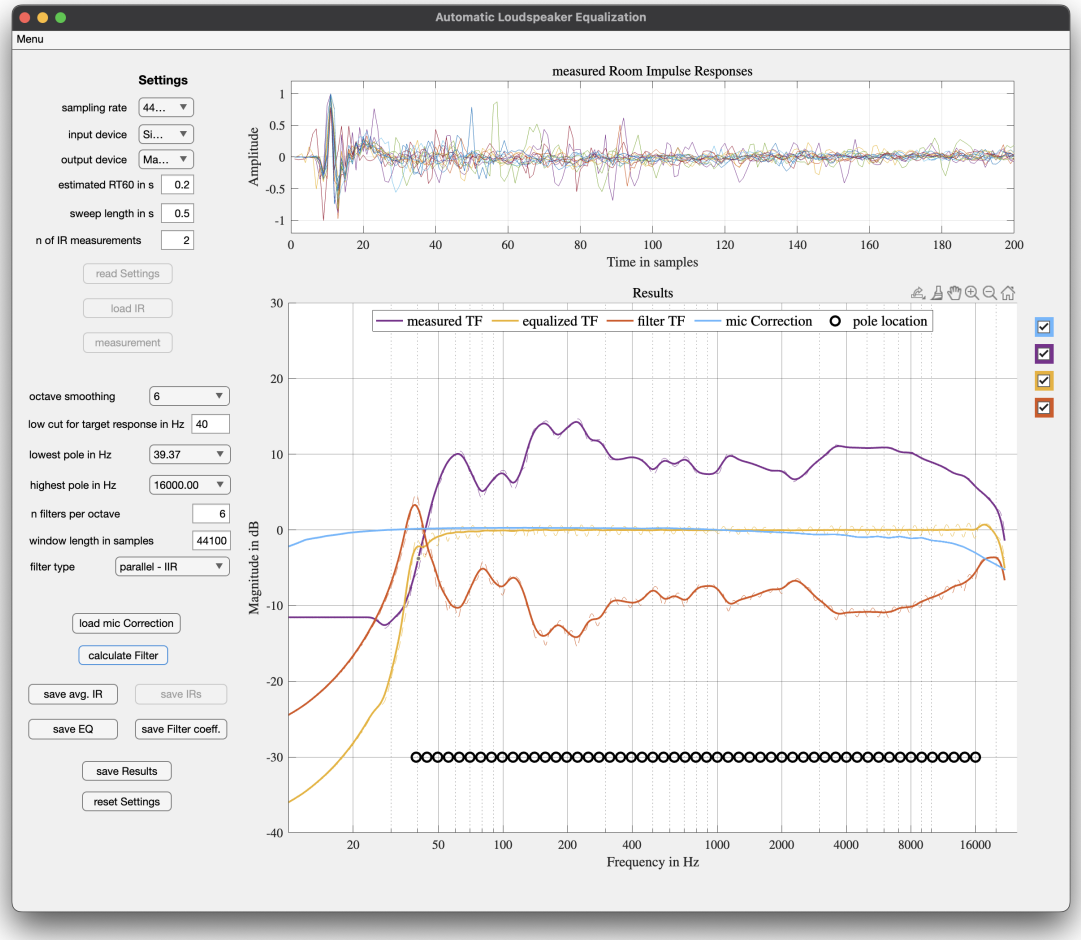


Figure 4.2: App overview

In the next step, the measurement run of the IRs can be started immediately by clicking on the button ≪[**Measurement**]≫. After each sweep, the user is asked to modify the microphone's location to get a complete set of IRs in the considered measurement area. After confirmation via the ≪[**OK**]≫ button, the subsequent measurement will trigger automatically. This procedure is performed for the chosen number of IR measurements. Weighing special regions can be realized by increasing the number of measurements within these regions. After each measurement, the calculated IR is aligned and appended to the top plot of the GUI. Therefore, the most prominent peak is found, and the IR is normalized so that all measured IRs are aligned and comparable.

The number of poles must be pre-determined to calculate the resulting filter. Therefore, the user picks the lowest and highest pole frequency (f_{low}, f_{high}) of a given set and the number of poles per octave n_{oct}. In order to obtain the most reasonable distribution of the poles, which determine the resolution of the filter around the corresponding frequency, a logarithmic frequency resolution with a constant Q on a logarithmic axis is implemented. The logarithmic frequency spacing corresponds best to the resolution of the human ear following the ERB scale [KP06]. Twenty-four pole frequencies per octave are pre-calculated between an interval $f_k \in [15.625, 32000]$Hz, diminished and adjusted as the user specifies. Fig. 4.3 shows an exemplary logarithmic pole distribution in the entire interval with $n_{\text{oct}} = 3$.

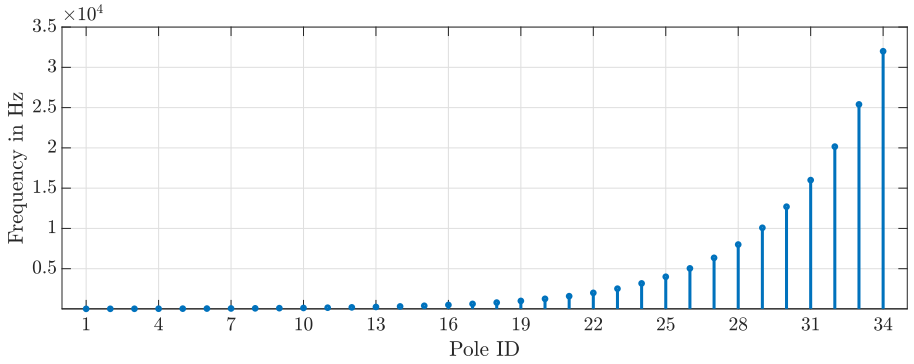


Figure 4.3: logarithmic pole distribution.

The dimensions of the enclosure and chassis of a loudspeaker bound their ability to produce very low frequencies. To prevent overcompensation in that region, the target response, which the LMS algorithm optimizes the filter coefficients, can be adapted to match the speaker's capabilities. Before step 6. in sec. 4.1, the target response is filtered with a fourth-order high-pass filter with variable cut-off frequency, defined by the user, such as described in eq. (3.15) in sec. 3.1.4. For most cases, a meaningful cut-off frequency can be obtained by the -3 dB value provided by the manufacturer or by just identifying the bound in the measured magnitude.

Creating an equalizing IR - *an IR that filters the loudspeaker such that the desired target frequency response arrives in the measured area* - requires truncation of the infinite IR. The window size can adjust the number of samples that pass the truncation. The smaller the window size, the smaller the frequency resolution and, therefore, the capability of equalizing and reproducing low frequencies. A good rule of thumb is that a tenth of the sampling frequency in samples results in a third-octave resolution in the low frequencies, see eq. (3.22).

The button ≪[**load mic correction**]≫ allows loading a mic correction curve for the used microphone. This correction is applied to the single measurements before averaging, as described in sec.2.2. After setting the parameters, the button ≪[**calculate filter**]≫ performs the LMS calculation of the filter coefficients.

The n_{Oct}-Octave smoothed magnitudes of

- the averaged measurement in violet,
- the resulting filter in amber,
- and the filtering result of both in yellow,

are displayed in the main plot window. Additionally, the pole frequencies are indicated by the black circles at the bottom of the plot. Visualizing these three magnitudes aims to understand how the calculated filter performs on the measurement. As the app allows the user to change the parameters for poles, high-pass filter, smoothing, and truncation amount, the visualization helps find the optimal filter. The filter can be calculated several times, and the user has immediate feedback on the impact on the resulting equalized loudspeaker room response.

If a resulting design matches the user's requirements, the equalizing IR (click ≪[**save EQ(.wav)**]≫) and the measured and averaged IR (click ≪[**save avg. IR(.wav)**]≫) can be saved as *.wav* file. Additionally, the app gives access to the filter coefficients and all measurements separately collected in a *.mat* file by clicking on ≪[**save results**]≫. The button ≪[**reset Settings**]≫ leads back to a new measurement run and resets all loaded measurements and preferences. If previous measurements should feed the algorithm, the button ≪[**load IR**]≫ allows the user to select saved measured IR as *.wav* file for the filter calculation.

5 Results

5.1 Window Length

First, the impact of the window length is evaluated. Fig. 5.3 shows the results with two different window lengths. Short window lengths lead to an inaccurate frequency resolution in the low frequencies. Hence, the equalization in this range is also impaired. Increasing the window length results in an appropriate equalization transfer function. Further increasing the window length can lead to overcompensation of the low frequencies.

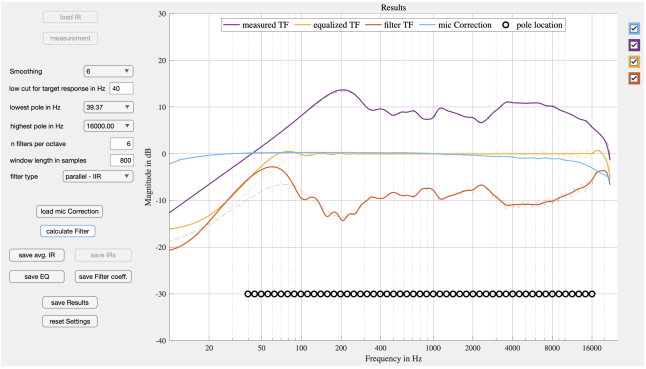


Figure 5.1: 800 samples truncation

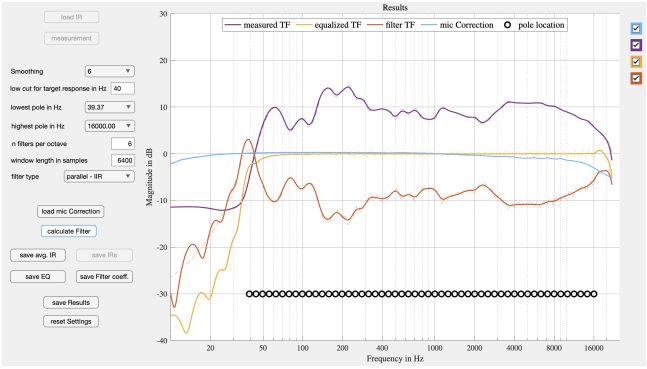


Figure 5.2: 6400 samples truncation

Figure 5.3: Calculation with pole distribution with n = 6, and truncation of measured IRs

5.2 Validation Measurement

In order to validate the results of the application, a test measurement was performed. Four IRs were calculated with different poles per octave. The effect of the calculated IR on a loudspeaker was measured using *Smaart v8*[1]. A two-channel fast Fourier transformation calculates the transfer function of the path between a measurement microphone and the loudspeaker. Figure 5.4 shows the speaker's frequency response and the responses after convolution with the different calculated IRs (for 3,4,12,24 poles per octave). For better consideration, *Smaart v8* smoothed all measurements to 1/12 octaves, the results with different smoothing adjustments are shown in the appendix.

By viewing Fig. 5.4, it is visible that the equalization succeeds, and the magnitude flattens out even by using three poles per octave. It stands out that some significant ripples occur in the frequency range from 250 Hz to 500 Hz in all four corrected magnitudes. Why even the most dense resolution cannot flatten out this range needs to be discussed.

5.3 Pole Resolution

Four different filtering IRs are calculated with pole resolutions from three to 24 poles per octave to investigate different pole frequency resolutions. By comparing the red curve in Fig. 5.4, which is the

[1] `http://www.rationalacoustics.de/smaart-v8.html`

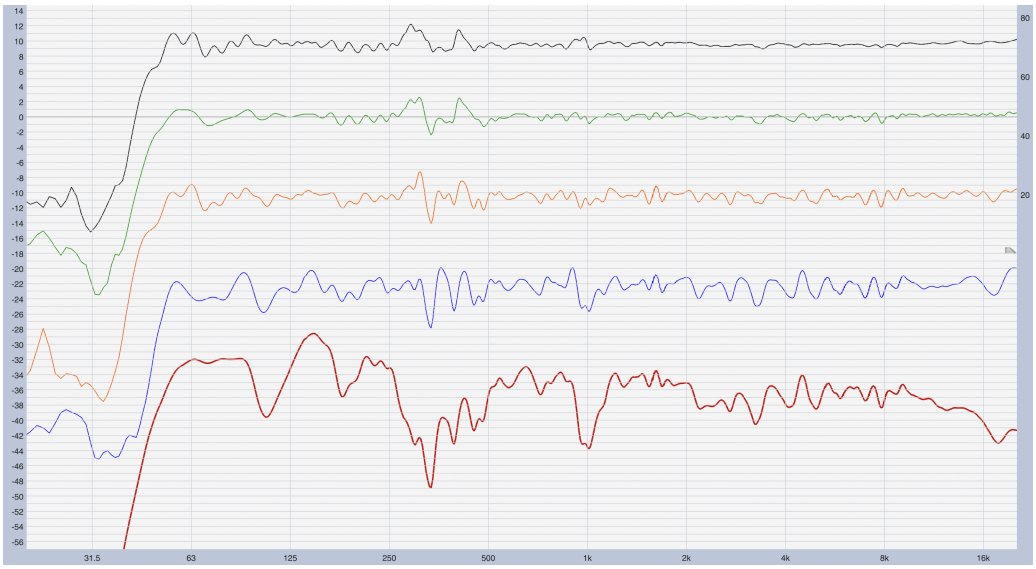


Figure 5.4: Validation measurement with Smaart v8. red: measurement, blue: corrected with three poles/octave, orange: corrected with six poles/octave, green: corrected with 12 poles/octave, black: corrected with 24 poles/octave. All smoothed to 1/12 octaves

unequalized frequency response, against the blue curve, it is visible that the frequency response of the speakers appears flatter. The blue graph results from the convolution with the three-poles-per-octave IR. By increasing the number of poles per octave during calculation, the resulting speaker curve gets more and more flat. Because of the low-frequency poles' density, more than 12 poles per octave add ripples in this range. Depending on the application, fewer poles could also be a good choice.

Calculating the IR with poles in the low-frequency range and a shallow high-pass filter for the target response can result in overcompensation and clipping effects. To prevent this, a well-considered selection of calculating parameters is mandatory. The graphical interface of the app helps to find suitable parameters.

5.4 Magnitude: Parallel IIR, Minimum-, Linear-Phase FIR Filter

In order to compare the different types of filter designs, the same averaged magnitude is corrected with a 30-pole parallel filter, a 4096-tap minimum-phase FIR filter, and a 4097-tap linear-phase FIR filter. In the appendix 7, Fig. 7.4, 7.5,7.6 depict the filter magnitudes, the measurements, and the resulting equalized responses. Overall, the magnitudes are extremely similar, especially in the higher frequency range, which indicates that all three designs work well. The only difference is visible in the low-frequency range below 100 Hz. The third-octave resolution of the parallel filter leads to an accurate equalization down to 20 Hz. The minimum-phase FIR magnitude with 4096 taps a resulting linear frequency resolution of 10.77 Hz is also capable of correcting the magnitude with some minor ripples down to 20 Hz. Due to the symmetry of a linear-phase filter with 4097 samples, the frequency resolution

$$\Delta f_{\mathrm{lp,FIR}} = \frac{f_\mathrm{s}}{L/2} = \frac{44.1\,\mathrm{kHz}}{4097/2} = 21.53\,\mathrm{Hz}\,, \tag{5.1}$$

limits the exact correcting in the very low-frequency range, which results in the ripples between 20 Hz and 80 Hz.

5.5 Phase: Parallel IIR, Minimum-, Linear-, and Mixed-Phase FIR Filter

The mixed-phase implementation was also taken into account when evaluating the phase responses. The results in magnitude and phase are shown in Fig. 5.5. All filter algorithms are fed again with the

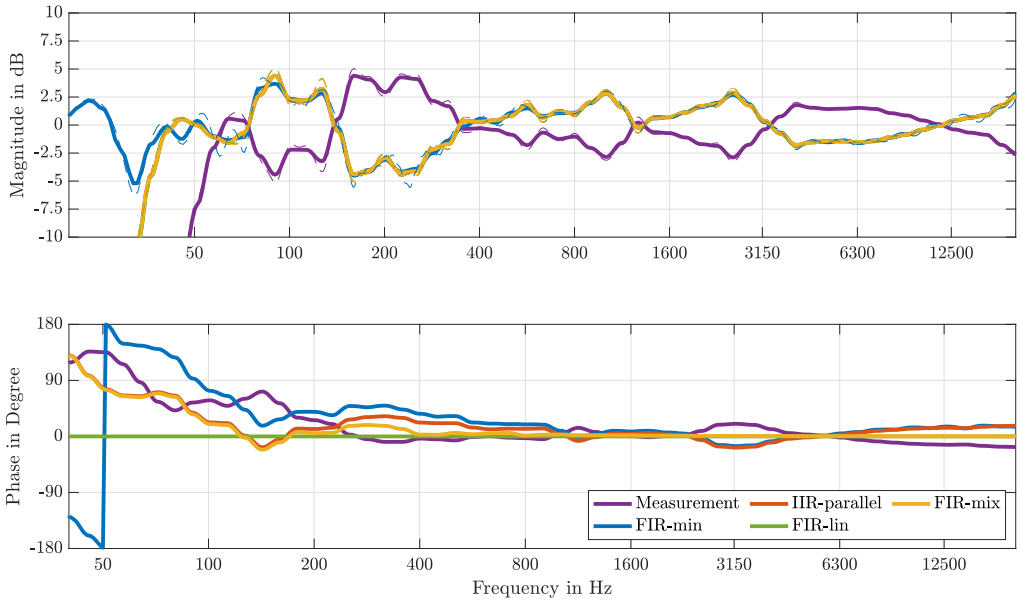


Figure 5.5: Comparison of different filter implementations in magnitude and phase

same measured IRs $h_i[h]$ of 15 observation locations in a small studio around the listening position. The averaged magnitude response $H_{\text{av}}[k]$ of this measurement is smoothed with an $1/6-$octave filter bank towards $H_{\text{av,s}}[k]$ and the corresponding microphone correction is applied. As the target for all algorithms, a Dirac delta filtered with a 56 Hz high-pass of order six is used. For the parallel IIR filter (IIR-parallel), a pole resolution of three poles per octave in the range from $30\,\text{Hz} - 20\,\text{kHz}$ was chosen. The minimum phase FIR filter (FIR-min) has a length of 4096 samples. To maintain frequency resolution, the filter length of the linear phase (FIR-lin) is doubled with 8193 samples. The linear part of the mixed phase filter (FIR-mix) contains 256 samples while the whole filter is truncated to 4096 samples.

The similarities in the magnitude of the different algorithms are unambiguous. Except for the minimum FIR filter, all filter magnitudes behave nearly the same throughout the whole frequency range. At very low frequencies, the minimum phase filter cuts less. The main differences between the designs are in the phase responses. The minimum phase and parallel IIR show an inverted phase response compared to measurement in the high frequency. As expected, the linear phase filter does not change the phase as well as the mixed phase filter in the frequencies above 2 kHz. The mixed-phase filter shows a similar phase at lower frequencies with minor deviations as the parallel IIR filter.

6 Conclusion

In this seminar report, a prototype application (Automatic Loudspeaker Equalization) was developed in MATLAB and published on Git[1], using parallel, FIR, and linear-phase filters, showing how to equalize loudspeakers in the playback room in an automated way. This prototype application can be used as a template for developing an automatic EQ plugin.

A perceptually matched frequency response equalization of the loudspeaker in the measured playback room can be achieved by n-band smoothing and averaging over multiple listening positions. The automation is done by an automatable sweep measurement routine and its deconvolution, the processing of the measured impulse responses by third-octave band smoothing and local averaging, as well as minimum phasing, and the formation of an equalization frequency response with Balazs Bank's parallel filter technique, which allows a freely parameterizable complexity/precision of the equalization filters. An FIR and linear-phase FIR filter calculation is also implemented for purposes beyond real-time processing.

As output, the software provides all IR measurements, the average IR, the suitable equalizing IR, and the filter coefficients for a parallel filter bank. A prototype `.vst`-plugin imports the resulting parallel-filter coefficients and performs a filtering process in the time domain. This prototype is only capable of stereo applications and expects two sets of coefficients for the left and right audio channels, respectively.

The example application demonstrates how the subtasks with all their parameters can be compactly operable, and the resulting frequency responses remain verifiable simultaneously. The app, which was presented in this work, can be improved in several steps. The following aspects are not covered within this work: evaluation of the filter quality within a listening experiment, further implementation of a multichannel version (stereo and ambisonics) with multi-sweep measurement, consideration of a target-response (customized EQ-curve) as well as a full realization as `.vst`-plugin.

[1] `https://github.com/simonwindtner/ALE`

7 Appendix

7.1 Derivation of Exponentially Decreasing Sinusiods

Using a partial fraction expansion to eq. (3.7) we get

$$H_k[z] = \frac{1}{(1-p_k z^{-1})(1-p_k^* z^{-1})} = \frac{C}{(1-p_k z^{-1})} + \frac{C^*}{(1-p_k^* z^{-1})}\,. \tag{7.1}$$

In order to obtain C we multiply 7.1 with $(1-p_k^* z^{-1})(1-p_k z^{-1})$

$$1 + 0z^{-1} = C(1-p_k^* z^{-1}) + C^*(1-p_k z^{-1}) = C + C^* - z^{-1}(Cp_k^* + C^* p_k)\,, \tag{7.2}$$

and compare the coefficients such that

$$1 = C + C^* \qquad \text{and} \qquad 0 \;\; = (Cp_k^* + C^* p_k) \qquad \longrightarrow \qquad C = \frac{p_k}{p_k - p_k^*}\,. \tag{7.3}$$

Knowing C, we are able to determine the IR $h_k[n]$ directly from $H_k[z]$ as

$$h_k[n] = C\, p_k^n\, u[n] + C^*\, {p_k^*}^n u[n]\,, \qquad \text{with unit-step-function } \; u[n]\,. \tag{7.4}$$

By using the expression for the poles as in eq. (3.6a) we define

$$p_k = e^{\frac{-\Delta\theta_k}{2}} e^{j\theta_k} = e^{\frac{-\Delta\theta_k}{2} + j\theta_k} \qquad \text{and} \qquad p_k^* = e^{\frac{-\Delta\theta_k}{2}} e^{-j\theta_k} = e^{\frac{-\Delta\theta_k}{2} - j\theta_k}\,. \tag{7.5}$$

The constant C and C^* become

$$C = \frac{p_k}{p_k - p_k^*} = \frac{e^{\frac{-\Delta\theta_k}{2}} e^{j\theta_k}}{e^{\frac{-\Delta\theta_k}{2}}(e^{j\theta_k} - e^{-j\theta_k})} = \frac{e^{j\theta_k}}{2j\sin(\theta_k)} \qquad \text{and} \qquad C^* = -\frac{e^{-j\theta_k}}{2j\sin(\theta_k)}\,. \tag{7.6}$$

Updating $h_k[n]$ from eq (7.4) by using C and the exponential expression for the poles p_k we get

$$\begin{aligned}
h_k[n] &= \frac{e^{j\theta_k}}{2j\sin(\theta_k)} \left(e^{\frac{-\Delta\theta_k}{2}} e^{j\theta_k}\right)^n u[n] - \frac{e^{-j\theta_k}}{2j\sin(\theta_k)} \left(e^{\frac{-\Delta\theta_k}{2}} e^{-j\theta_k}\right)^n u[n] \\
&= \frac{1}{2j\sin(\theta_k)} \left(e^{j\theta_k} e^{\frac{-\Delta\theta_k n}{2}} e^{j\theta_k n}\right) u[n] - \frac{1}{2j\sin(\theta_k)} \left(e^{-j\theta_k} e^{\frac{-\Delta\theta_k n}{2}} e^{-j\theta_k n}\right) u[n] \\
&= \frac{e^{\frac{-\Delta\theta_k n}{2}}}{2j\sin(\theta_k)} \left(e^{j\theta_k} e^{j\theta_k n} - e^{-j\theta_k} e^{-j\theta_k n}\right) u[n] = \frac{e^{\frac{-\Delta\theta_k n}{2}}}{2j\sin(\theta_k)} \left(e^{j\theta_k(n+1)} - e^{-j\theta_k(n+1)}\right) u[n] \\
&= \frac{e^{\frac{-\Delta\theta_k n}{2}}}{2j\sin(\theta_k)} 2j\sin(\theta_k(n+1))u[n] = \frac{e^{\frac{-\Delta\theta_k n}{2}}}{\sin(\theta_k)} \sin(\theta_k(n+1))u[n]\,.
\end{aligned} \tag{7.7}$$

As we can observe, the resulting IRs from the all-pole-filter are exponentially decreasing sinusoids, which are scaled with $\frac{1}{\sin(\theta_k)}$. From this, it follows that the low-frequency sinusoids are drastically raised in amplitude. Fig. 7.1 show some exemplary and normalized $h_k[n]$'s and its magnitude.

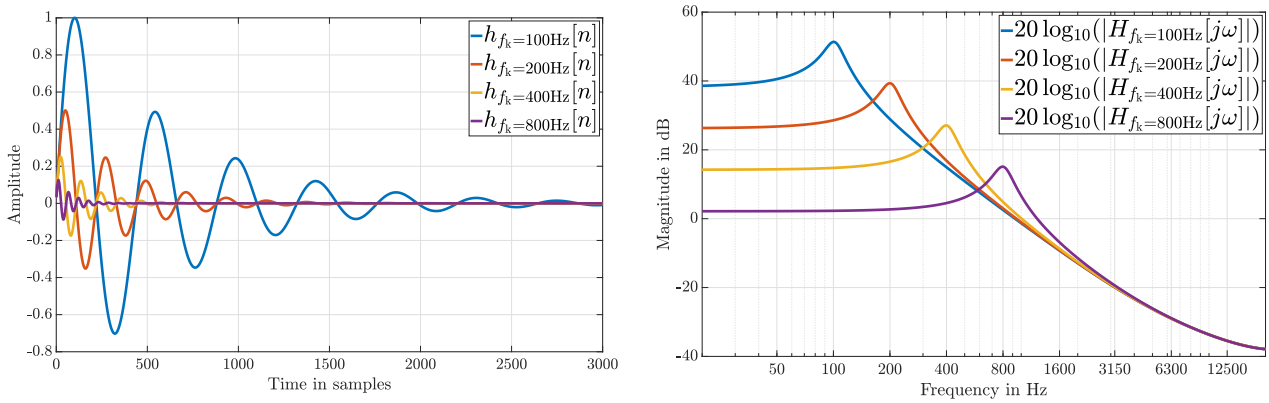


Figure 7.1: Exemplary exponentially decreasing sinusoids in the time- and frequency domain

7.2 Additional Plots

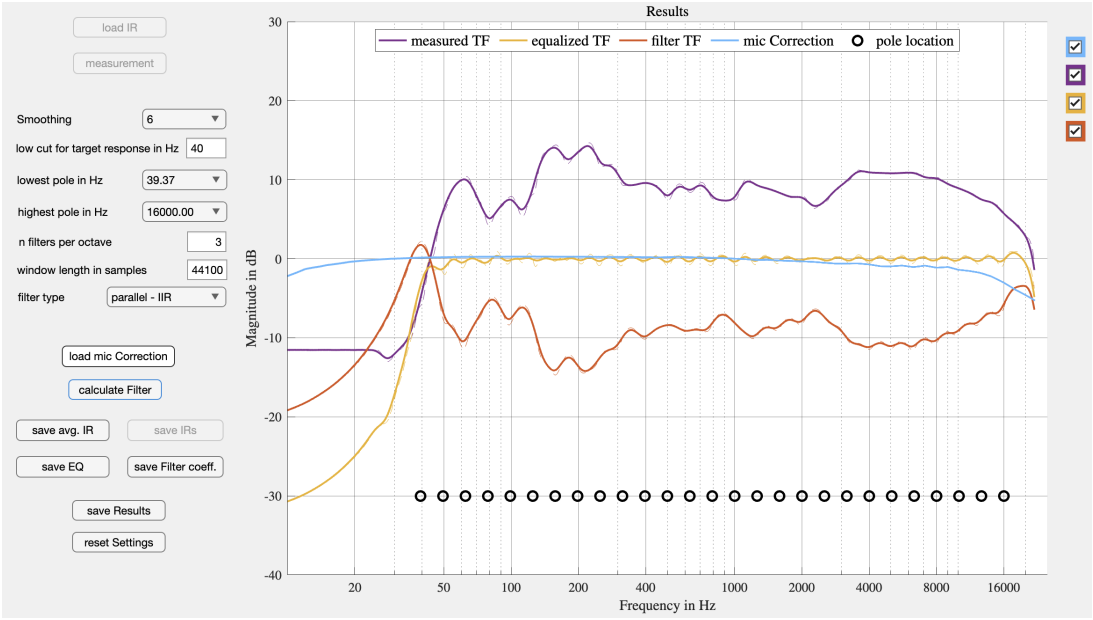


Figure 7.2: Calculation from the app for 3 poles/octave; All smoothed to 1/6 octaves

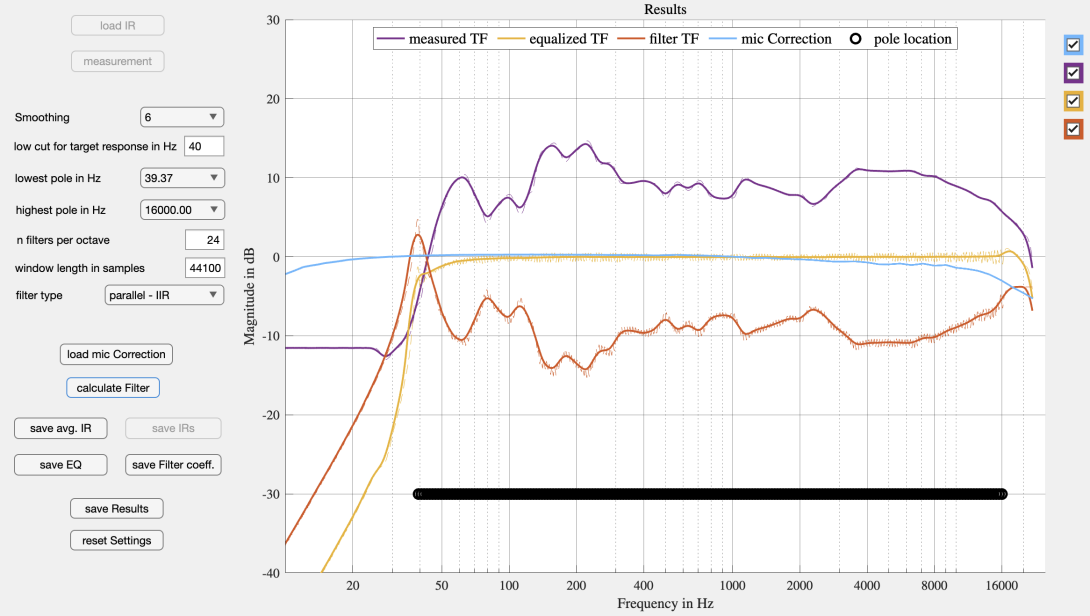


Figure 7.3: Calculation from the app for 24 poles/octave; All smoothed to 1/6 octaves

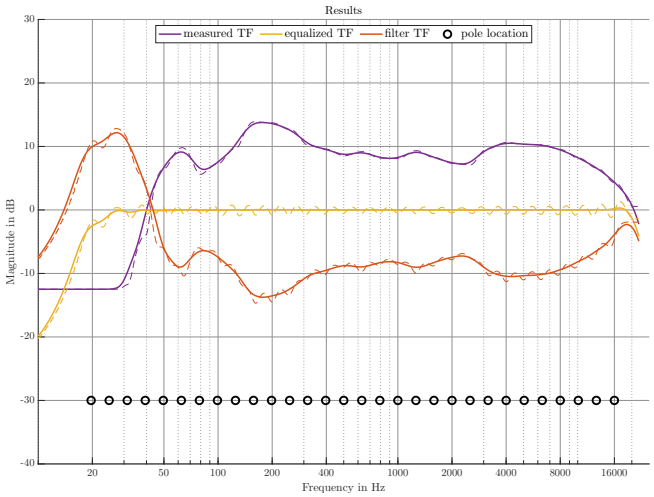


Figure 7.4: Parallel filter

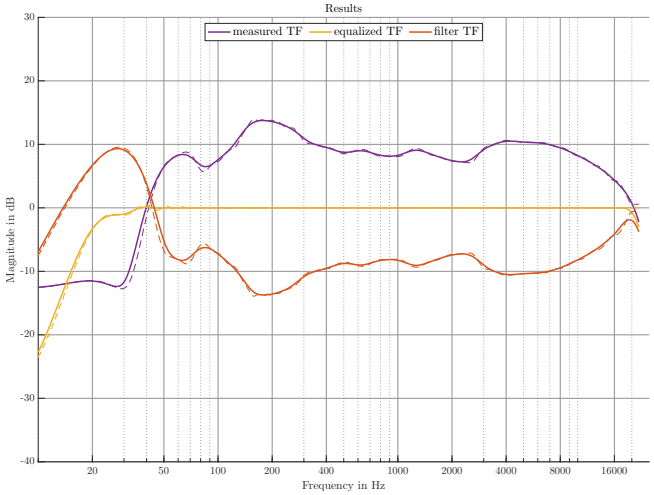


Figure 7.5: FIR filter

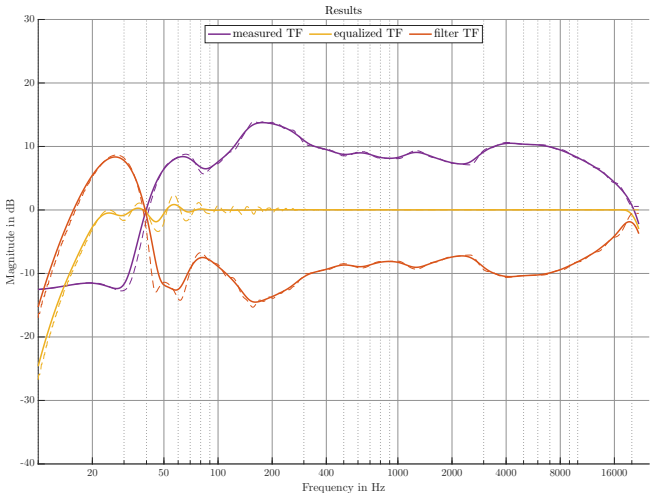


Figure 7.6: Linear-phase filter

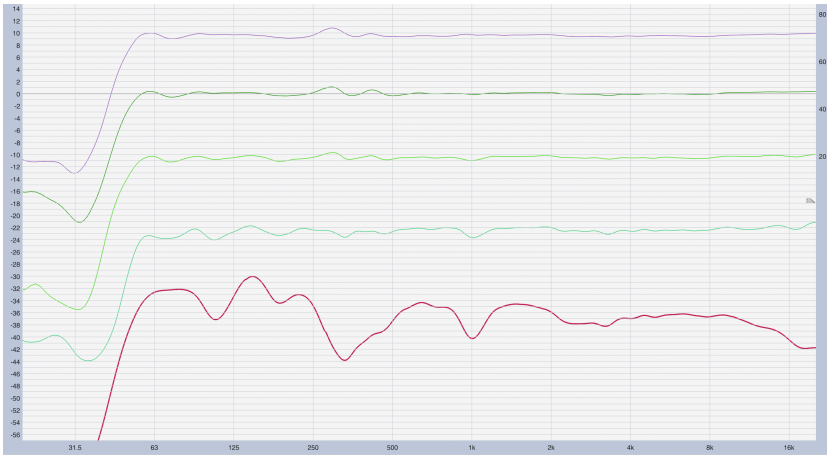


Figure 7.7: Validation measurement with Smaart v8. red: measurement, blue: corrected with 3 poles/octave, orange: corrected with six poles/octave, green: corrected with 12 poles/octave, black: corrected with 24 poles/octave. All smoothed to 1/3 octaves

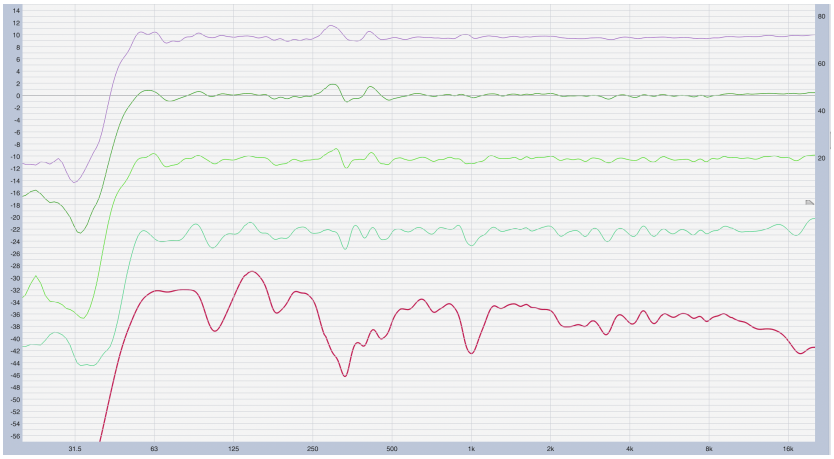


Figure 7.8: Validation measurement with Smaart v8. red: measurement, blue: corrected with 3 poles/octave, orange: corrected with 6 poles/octave, green: corrected with 12 poles/octave, black: corrected with 24 poles/octave. All smoothed to 1/6 octaves

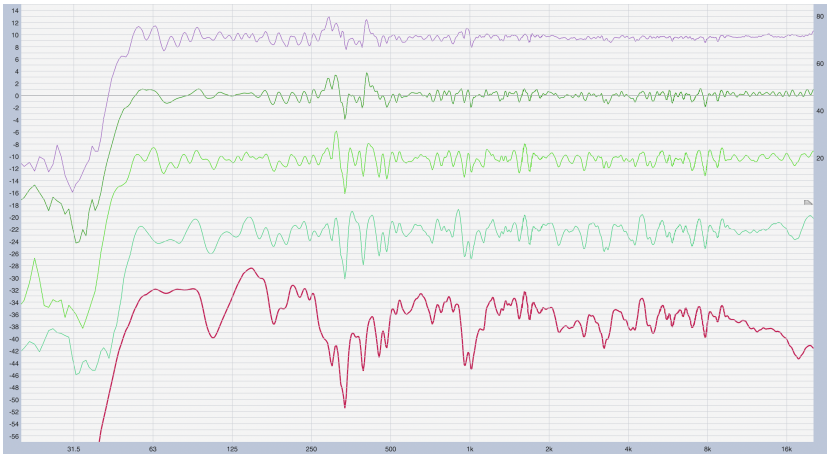


Figure 7.9: Validation measurement with Smaart v8. red: measurement, blue: corrected with 3 poles/octave, orange: corrected with 6 poles/octave, green: corrected with 12 poles/octave, black: corrected with 24 poles/octave. All smoothed to 1/24 octaves

Bibliography

[Ban08] B. Bank, "Perceptually motivated audio equalization using fixed-pole parallel second-order filters," vol. 15, pp. 477 – 480, 2008.

[ban13a] b. bank, "audio equalization with fixed-pole parallel filters: an efficient alternative to complex smoothing," *journal of the audio engineering society*, vol. 61, no. 1/2, pp. 39–49, january 2013.

[Ban13b] B. Bank, "Loudspeaker and room response equalization using parallel filters: Comparison of pole positioning strategies," in *Audio Engineering Society Conference: 51st International Conference: Loudspeakers and Headphones*, Aug 2013. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=16884

[HCZ09] M. Holters, T. Corbach, and U. Zölzer, "Impulse response measurement techniques and their applicability in the real world," in *Proceedings of the 12th International Conference on Digital Audio Effects, DAFx 2009*, 2009.

[ISO15] ISO Central Secretary, "Cinematography — b-chain electro-acoustic reponse of motion-picture control rooms and indoor theatres — specifications and measurements," International Organization for Standardization, Geneva, CH, Standard ISO 2969:2015, 2015. [Online]. Available: https://www.iso.org/standard/43646.html

[KP06] M. Karjalainen and T. Paatero, "Equalization of audio systems using kautz filters with log-like frequency resolution," in *Audio Engineering Society Convention 120*, 2006. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=13571

[MBL07] P. Majdak, P. Balazs, and B. Laback, "Multiple exponential sweep method for fast measurement of head-related transfer functions," vol. 55, no. 7, pp. 623–637, 2007. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=14190

[OSB99] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Prentice-hall Englewood Cliffs, 1999.

[PCR+11] A. Primavera, S. Cecchi, L. Romoli, P. Peretti, and F. Piazza, "A low latency implementation of a non uniform partitioned overlap and save algorithm for real time applications," in *Audio Engineering Society Convention 131*, Oct 2011. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=16047

[Too09] F. Toole, *Sound Reproduction - The Acoustics and Psychoacoustics of Loudspeakers and Rooms*. Boca Raton, Fla: CRC Press, 2009.

[ZG22] F. Zotter and L. Gölles, "Exponentially swept sinusoid with zero-crossing in its beginning and end," https://git.iem.at/zotter/IR-Measurement-JupyterNotebook/-/blob/master/01-generate-zero-crossing-sweep.ipynb, 2022, online; accessed 14-12-2023.

[Zö11] U. Zölzer, *DAFX - Digital Audio Effects*. New York: John Wiley Sons, 2011.