

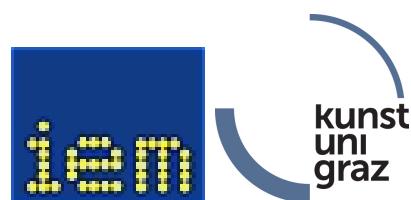
SE Algorithms in Acoustics and Computer Music 02
LV-Nr.17.0033

Automatic Loudspeaker Equalization

Leon Merkel, 11745386
Simon Windtner, 11727158

Lecturer:

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



kunst
uni
graz

Institute for Electronic Music and Acoustics
University of Music and Performing Arts
October 17, 2023

Equalization is a crucial tool contributing to high audio quality for playback loudspeakers in their acoustic environment. In our work, we implement filters that automatically calculate its coefficients to flatten the loudspeaker's magnitude over an averaged listening area. Different spectral resolutions and filter designs are evaluated by using the prototype implementation. The parallel filters impress with the very 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	4
1.1	Motivation	4
1.2	Structure of this Work	5
2	Theory	6
2.1	System Identification	6
2.1.1	Impulse Response	6
2.1.2	Exponential Sweep	6
2.2	Post-Processing of the Measurements	7
2.2.1	Microphone Correction and Magnitude and Phase Averaging	7
2.2.2	Magnitude Smoothing	8
2.2.3	Minimum-Phase Version of IR	9
2.3	Parallel Filter calculation	9
2.3.1	General Filter Structure	9
2.3.2	Pole Positioning	10
2.3.3	Derivation of Exponentially Decreasing Sinusoids	10
2.3.4	Problem Formulation in Matrix Notation	11
2.3.5	Weight Estimation using Least-Squares Algorithm	12
2.4	Minimum-Phase FIR Filter	13
2.5	Linear-Phase FIR Filter	14
3	Software Implementation	16
3.1	Structure	16
3.2	Operation Procedure	17
4	Results	20
4.1	Window Length	20
4.2	Validation Measurement	20
4.3	Pole Resolution	20
4.4	Parallel, Minimum-Phase FIR, and Linear-Phase FIR Filter	21
5	Conclusion	22
6	Appendix	23

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

Regarding sound reproduction, F. Toole asks himself this question in his book [Too09] and attempts to answer it concerning several topics. He discusses the influence of loudspeakers [chp. 17 & 18] and their surrounding acoustical environment [chp. 16] on the resulting human auditory perception [chp. 19]. In [chp. 22], he describes how to equalize loudspeakers due to certain requirements and summarizes that sound reproduction in space can be improved by compensating acoustical and electrical effects.

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.

Using parametric filters is a common approach for equalizing an arbitrary frequency response to obtain a flat or specifically desired magnitude. Compared to low- and high-pass filters, which manipulate the audio spectrum below or above a particular cut-off frequency, a parametric equalizer shapes the spectrum by affecting certain frequency bands without touching other frequency bands. Those equalizers are built mainly by four to six serially connected first and second-order shelving and peak filters with individual coefficients. Shelving filters enhance or attenuate low or high-frequency bands, whereas peak filters are used to enhance or attenuate mid-frequency bands [Zö11]. The easy implementation and coefficient calculation stand in contrast to the small number of bands and degrees of freedom. Increasing the number of bands leads to a higher frequency resolution and extends the filter length. From that, a slightly higher delay occurs, but the computational complexity increases due to the multiplication of each band and the essential coefficient precision for IIR filters.

Another approach to flatten a loudspeaker response is the equalization using Kautz filters. This equalization scheme utilizes a particular infinite impulse response (IIR) filter configuration called Kautz filters as generalizations of finite impulse response (FIR) filters and their warped counterparts. The desired frequency resolution allocation, in this case, the logarithmic one, is attained by a chosen set of fixed pole positions that define the particular Kautz filter. The frequency resolution mapping is characterized by the all-pass part of the Kautz filter, which is interpreted as a formal generalization

of the warping concept. The second step in the filter design is assigning the Kautz filter tap-output weights, which is then more or less a standard least-square configuration. The proposed method is demonstrated in [KP]. Here, the same drawback as using the parametric filters occurs, with the advantage of a more straightforward coefficient calculation due to fewer degrees of freedom.

Solutions like Sonaworks¹ - 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.

Avoiding the discussed disadvantages of high latency and computational complexity, the presented realization uses parallel second-order IIR filters after Balázs Bank [Ban]. Compared to serially connected IIR filters, parallel filters benefit from fewer multiplications and significantly less delay.

1.2 Structure of this Work

The first chapter gives an overview of the theory of the used equalization scheme. Furthermore, the theoretical fundamentals are described in detail, such as sweep calculation, room impulse response (RIR) measurements, averaging of the RIR, band-limited pole calculation, and filter calculation for parallel filters. The implementation is presented in the second chapter. The third chapter shows several filter results of the implemented tool and their performance in the field. The fourth chapter gives a conclusion and an outlook of the discussed work.

¹www.sonaworks.com

²www.sonible.com/smarteq3

2 Theory

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. [MBL].

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 [HCZ], 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

$$\sum_{k=0}^{N-1} x[k]x_{\text{inv}}[n-k] = C\delta[n_0 - n]. \quad (2.1)$$

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

$$C h[n_0 - n] = \sum_{k=0}^{N-1} y[k]x_{\text{inv}}[n-k]. \quad (2.2)$$

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

$$H[k] = \mathcal{F}\{y[n]\}\mathcal{F}\{x_{\text{inv}}[n]\}. \quad (2.3)$$

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

$$h[n] = \mathcal{F}^{-1}\{H[k]\}. \quad (2.4)$$

2.1.2 Exponential Sweep

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

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

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 $\omega[n]$

$$\omega[n] = ce^{bn}. \quad (2.6)$$

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

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

Therefore $x[n]$ can be calculated by

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

In order to start and end the exponential sweep with zero phase, a gradient descent algorithm finds the optimal starting frequency ω_0 . The resulting normalized sweep in frequency/time representation and its convolution with its inverse are depicted in Fig. 2.1 and Fig. 2.2.

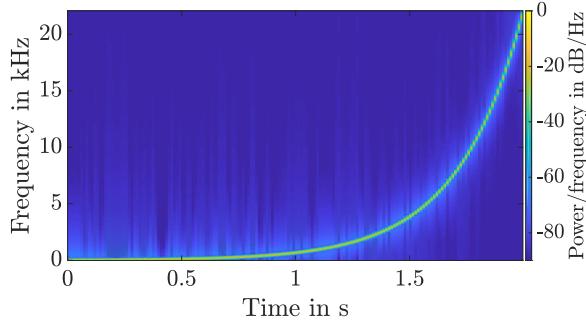


Figure 2.1: exponential sweep

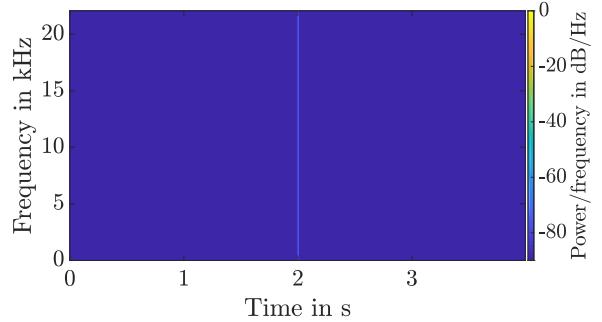


Figure 2.2: convolution of $x[n]$ and $x_{\text{inv}}[n]$

2.2 Post-Processing of the Measurements

2.2.1 Microphone Correction and Magnitude and Phase Averaging

After measuring the exponential sine sweeps, the IRs $h_i[n]$ per microphone position i are calculated as described in section 2.1. To accomplish a flat frequency response on average across an entire audience area, the magnitude-squared frequency response,

$$|H_i[k]|^2 = |\mathcal{F}\{h_i[n]\}|^2 \quad (2.9)$$

measured at several locations within the audience can be averaged. In order to compensate the magnitude of the measurement microphone, a calibration file can be added. 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. The normalized sum of the squared absolute values of the Fourier transformation of each i measured IR results in the spectral average $H_{\text{av}}[k]$ with k frequency bins

$$H_{\text{av}}[k] = \frac{1}{N} \sum_{i=1}^N \left(|H_i[k]| |H_{\text{mic}}[k]|^{-1} \right)^2 . \quad (2.10)$$

The phase relationship of several RIRs can also be averaged by considering the phase angles of each measurement. Therefore, the averaged phase angles $\varphi_{\text{av}}[k]$ can be obtained by calculating the normalized mean of the exponential phase angle $e^{j\varphi_i[k]}$ over all i measurements, as shown in equation 2.11

$$\varphi_{\text{av}}[k] = \frac{\frac{1}{N} \sum_{i=1}^N e^{j\varphi_i[k]}}{\left| \frac{1}{N} \sum_{i=1}^N e^{j\varphi_i[k]} \right|} . \quad (2.11)$$

The averaged IR is determined by performing the inverse Fourier transformation of the composed version of the averaged magnitude and phase

$$h_{\text{av}}[n] = \Re \left\{ \mathcal{F}^{-1} \{ H_{\text{av}}[k] \varphi_{\text{av}}[k] \} \right\}. \quad (2.12)$$

As further calculations require the minimal phase version of the IR, the phase averaging step can be skipped. Subsequent calculations are performed with $H_{\text{av}}[k]$ from eq. (2.10). It is also worth noting that averaging over several acoustic measurements leads to an expected value close to zero at high frequencies because minimal path differences result in significant phase changes, which are summarized nearly to zero.

2.2.2 Magnitude Smoothing

A common technique to adopt human hearing perception is reducing spectral resolution. The averaged magnitude $H_{\text{av}}[k]$ is filtered with several n-octave band-pass filters. Thus, the volatile behavior of the magnitude will be smoothed in particular frequency bands. For every n -octave center frequency k_c we calculate $H_{\text{av},s}^{(k_c)}[k]$ with

$$H_{\text{av},s}^{(k_c)} = \sum_{k=0}^{\text{NFFT}/2} W^{(k_c)}[k] H_{\text{av}}[k], \quad (2.13)$$

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

$$W^{(k_c)}[k] = \frac{\widetilde{W}^{(k_c)}[k]}{\sum_m \widetilde{W}^{(k_c)}[m]} \quad \text{with} \quad \widetilde{W}^{(k_c)}[k] = \cos^2 \left[\frac{\pi}{2} \text{Clip} \left\{ \text{ld} \left(\frac{k}{k_c} n_{\text{oct}} \right), -1, 1 \right\} \right]. \quad (2.14)$$

The resulting filter bank for $n_{\text{oct}} = 3$ is shown in Fig. 2.3. In order to achieve the previous frequency resolution of $\text{NFFT}/2$, necessary for the further inverse Fourier Transformation, the smoothed values are multiplied with the whole filter bank consisting of N_{oct} frequency bands, such that

$$H_{\text{av},s}[k] = \sqrt{\sum_{k_c=k_{c,0}}^{k_{c,N_{\text{oct}}}} W^{(k_c)}[k] H_{\text{av},s}^{(k_c)}}. \quad (2.15)$$

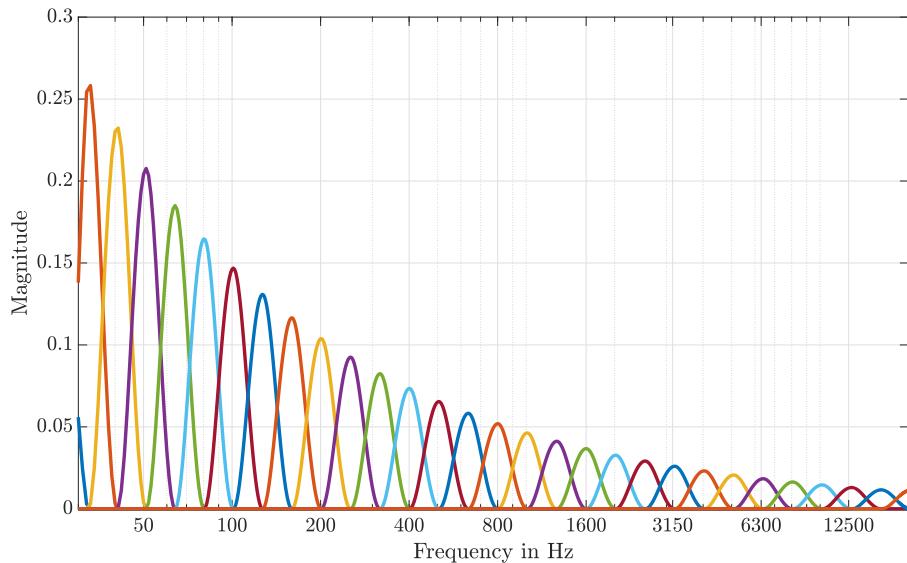


Figure 2.3: $W^{(k_c)}[k]$: Normalized \cos^2 third-octave filterbank for frequency-response smoothing.

2.2.3 Minimum-Phase Version of IR

The averaged IR is calculated using the complex cepstrum of the averaged and smoothed magnitude $H_{av,s}[k]$. The following steps lead to the averaged minimal phase version of the IR $h_{min}[n]$, which is written as $h_s[n]$ for simplicity in the next chapter:

$$h[n] = \mathcal{F}^{-1}\{\ln(H_{av,s}[k])\}$$

$$h_c[n] = \begin{cases} h[n] & \text{for } n = 0 \\ 2h[n] & \text{for } 1 \leq n \leq \frac{NFFT}{2} - 1 \\ h[n] & \text{for } n = \frac{NFFT}{2} \\ 0 & \text{for } \frac{NFFT}{2} + 1 \leq n \leq NFFT \end{cases} \quad (2.16)$$

$$h_{min}[n] = \mathcal{F}^{-1}\{e^{\mathcal{F}\{h_c[n]\}}\}$$

2.3 Parallel Filter calculation

Implementing infinite impulse response (IIR) filters in parallel second-order sections has been chosen due to the computational efficiency and the short delay times in the filtering procedure. The positional independent averaging process leads to gentle magnitudes, so no steep filters are necessary to match the target response. Compared to a finite impulse response (FIR) filter, the parallel design of IIR filters requires only two sample delays with total frequency resolution. This is especially advantageous in the low-frequency range, where FIR filters need many coefficients to achieve the exact resolution, which again is computationally effortful and costs much delay in real-time processing.

2.3.1 General Filter Structure

The general form of a second-order parallel filter can be denoted as

$$H(z) = \sum_{k=1}^K \frac{d_{k,0} + d_{k,1}z^{-1}}{1 + a_{k,1}z^{-1} + a_{k,2}z^{-2}} + \sum_{m=0}^M b_m z^{-m}, \quad (2.17)$$

where K is the number of second-order IIR sections, and M is the number of M -order finite impulse response (FIR) sections. The filter coefficients d_k , a_k , and b_m have to be determined such that the filter reaches the desired behavior. The structure is shown in fig. 2.4. In the following approach after

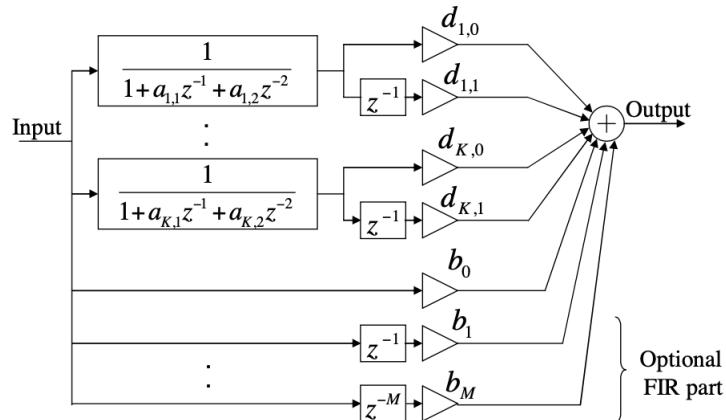


Figure 2.4: structure of parallel filter [Ban13b]

B. Bank, the idea of designing a parallel filter that minimizes the error between the final equalized

response $h_{\text{eq}}[n]$ and the target frequency response $h_t[n]$. It leads to a system identification problem with output error minimization: the input of the parallel filter is the system response $h_s[n]$, and the tool should estimate the filter parameters; thus, the output $h_{\text{eq}}[n]$ matches best with the target response $h_t[n]$. Therefore, the transfer functions from eq. (2.17) can rewritten as

$$H_{\text{eq}}(z) = H(z)H_s(z) = \sum_{k=1}^K \frac{d_{k,0} + d_{k,1}z^{-1}}{a_{k,0} + a_{k,1}z^{-1} + a_{k,2}z^{-2}} H_s(z) + \sum_{m=0}^M b_m z^{-m} H_s(z). \quad (2.18)$$

2.3.2 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

$$f_k = [100, 125, 160, 200, 250, 315, 400, 500, 630, 800, 1000, 1250, 1600, 2000]^T. \quad (2.19)$$

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

$$\Delta\vartheta_k = \frac{\vartheta_k}{Q_k}, \quad \text{where} \quad \vartheta_k = \frac{2\pi f_k}{f_s} \quad \text{with sampling frequency } f_s. \quad (2.20)$$

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

$$\Delta\vartheta_k = \left(\frac{\vartheta_{k+1} - \vartheta_{k-1}}{2} \right). \quad (2.21)$$

Then, the corresponding poles p_k are determined by

$$p_k = e^{\frac{-\Delta\vartheta_k}{2}} e^{\pm j\vartheta_k}, \quad (2.22a)$$

with the pole frequencies ϑ_k in radians [ban13a]. From filter theory, we can derive 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 conjugated version p_k^* , the transfer function reads as

$$H_k(z) = \frac{1}{(1 - p_k z^{-1})(1 - p_k^* z^{-1})} = \frac{1}{1 + 2\Re\{p_k\} z^{-1} + |p_k|^2 z^{-2}}, \quad (2.23)$$

so that the filter coefficients result in

$$a_{k,0} = 1, \quad a_{k,1} = 2\Re\{p_k\}, \quad a_{k,2} = |p_k|^2. \quad (2.24)$$

2.3.3 Derivation of Exponentially Decreasing Sinusoids

Using a partial fraction expansion to eq. (2.23) we get

$$H_k[z] = \frac{1}{(1 - p_k z^{-1})(1 - p_k^* z^{-1})} = \frac{C}{(1 - p_k z^{-1})} + \frac{C^*}{(1 - p_k^* z^{-1})}. \quad (2.25)$$

In order to obtain C we multiply 2.25 with $(1 - p_k^* z^{-1})(1 - p_k z^{-1})$

$$1 + 0z^{-1} = C(1 - p_k^* z^{-1}) + C^*(1 - p_k z^{-1}) = C + C^* - z^{-1}(Cp_k^* + C^*p_k), \quad (2.26)$$

and compare the coefficients such that

$$1 = C + C^* \quad \text{and} \quad 0 = (C p_k^* + C^* p_k) \quad \rightarrow \quad C = \frac{p_k}{p_k - p_k^*}. \quad (2.27)$$

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

$$h_k[n] = C p_k^n u[n] + C^* p_k^{n-1} u[n], \quad \text{with unit-step-function } u[n]. \quad (2.28)$$

By using the expression for the poles as in eq. (2.22a) we define

$$p_k = e^{-\frac{\Delta\vartheta_k}{2}} e^{j\vartheta_k} = e^{-\frac{\Delta\vartheta_k}{2} + j\vartheta_k} \quad \text{and} \quad p_k^* = e^{-\frac{\Delta\vartheta_k}{2}} e^{-j\vartheta_k} = e^{-\frac{\Delta\vartheta_k}{2} - j\vartheta_k}. \quad (2.29)$$

The constant C and C^* become

$$C = \frac{p_k}{p_k - p_k^*} = \frac{e^{-\frac{\Delta\vartheta_k}{2}} e^{j\vartheta_k}}{e^{-\frac{\Delta\vartheta_k}{2}} (e^{j\vartheta_k} - e^{-j\vartheta_k})} = \frac{e^{j\vartheta_k}}{2j\sin(\vartheta_k)} \quad \text{and} \quad C^* = -\frac{e^{-j\vartheta_k}}{2j\sin(\vartheta_k)}. \quad (2.30)$$

Updating $h_k[n]$ from eq (2.28) by using C and the exponential expression for the poles p_k we get

$$\begin{aligned} h_k[n] &= \frac{e^{j\vartheta_k}}{2j\sin(\vartheta_k)} \left(e^{-\frac{\Delta\vartheta_k}{2}} e^{j\vartheta_k} \right)^n u[n] - \frac{e^{-j\vartheta_k}}{2j\sin(\vartheta_k)} \left(e^{-\frac{\Delta\vartheta_k}{2}} e^{-j\vartheta_k} \right)^n u[n] \\ &= \frac{1}{2j\sin(\vartheta_k)} \left(e^{j\vartheta_k} e^{-\frac{\Delta\vartheta_k n}{2}} e^{j\vartheta_k n} \right) u[n] - \frac{1}{2j\sin(\vartheta_k)} \left(e^{-j\vartheta_k} e^{-\frac{\Delta\vartheta_k n}{2}} e^{-j\vartheta_k n} \right) u[n] \\ &= \frac{e^{-\frac{\Delta\vartheta_k n}{2}}}{2j\sin(\vartheta_k)} \left(e^{j\vartheta_k} e^{j\vartheta_k n} - e^{-j\vartheta_k} e^{-j\vartheta_k n} \right) u[n] = \frac{e^{-\frac{\Delta\vartheta_k n}{2}}}{2j\sin(\vartheta_k)} \left(e^{j\vartheta_k(n+1)} - e^{-j\vartheta_k(n+1)} \right) u[n] \\ &= \frac{e^{-\frac{\Delta\vartheta_k n}{2}}}{2j\sin(\vartheta_k)} 2j\sin(\vartheta_k(n+1)) u[n] = \frac{e^{-\frac{\Delta\vartheta_k n}{2}}}{\sin(\vartheta_k)} \sin(\vartheta_k(n+1)) u[n]. \end{aligned} \quad (2.31)$$

As we can observe, the resulting IRs from the all-pole-filter are exponentially decreasing sinusoids, which are scaled with $\frac{1}{\sin(\vartheta_k)}$. From this, it follows that the low-frequency sinusoids are drastically raised in amplitude. Fig. 2.5 show some exemplary and normalized $h_k[n]$'s and its magnitude.

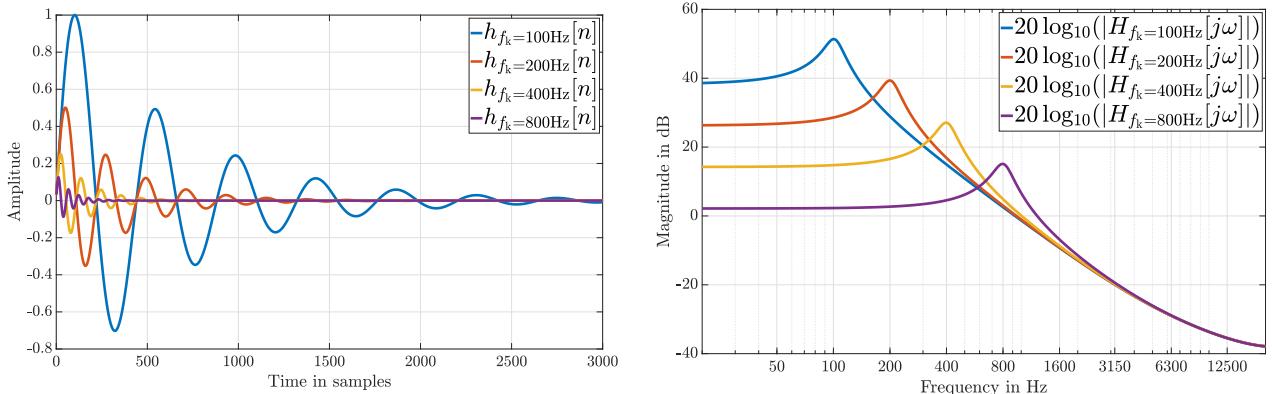


Figure 2.5: Exemplary exponentially decreasing sinusoids in the time- and frequency domain

2.3.4 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 2.17, in matrix notation can be written as

$$\mathbf{h}_{\text{eq}} = \mathbf{M}\mathbf{d}, \quad (2.32)$$

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

$$\mathbf{d} = [d_{1,0}, d_{1,1}, \dots, d_{K,0}, d_{K,1}, b_0, \dots, b_M]^T, \quad (2.33)$$

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

$$\begin{aligned} s_k[n] &= \sum_{i=0}^N h_k[i] h_s[n-i] = h_k[n] * h_s[n], \\ S_k[z] &= H_k[z] H_s[z] = \sum_{k=1}^K \frac{1}{a_{k,0} + a_{k,1}z^{-1} + a_{k,2}z^{-2}} H_s(z). \end{aligned} \quad (2.34)$$

In order to fulfill eq. (2.18) 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. (2.18) to

$$H_{\text{eq}}(z) = \sum_{k=1}^K (d_{k,0} + d_{k,1}z^{-1}) S_k[z] + \sum_{m=0}^M b_m z^{-m} H_s(z). \quad (2.35)$$

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

$$\mathbf{M} = \left[\begin{array}{ccccccccc} s_0[0] & 0 & s_1[0] & 0 & \dots & s_K[0] & 0 & h_s[0] & 0 \\ s_0[1] & s_0[0] & s_1[1] & s_1[0] & \dots & s_K[1] & s_K[0] & h_s[1] & h_s[0] \\ s_0[2] & s_0[1] & s_1[2] & s_1[1] & \dots & s_K[2] & s_K[1] & h_s[2] & h_s[1] \\ \vdots & s_0[2] & \vdots & s_1[2] & \dots & \vdots & s_K[2] & \vdots & h_s[2] \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ s_0[N] & s_0[N-1] & s_1[N] & s_1[N-1] & \dots & s_K[N] & s_K[N-1] & h_s[N] & h_s[N-1] \end{array} \right]. \quad (2.36)$$

2.3.5 Weight Estimation using Least-Squares Algorithm

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

$$\mathbf{h}_t = [h_t[0], \dots, h_t[N]]^T = [1, 0, \dots, 0]^T. \quad (2.37)$$

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

$$H_{\text{HP},4}[z] = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}, \quad \text{and} \quad H_{t,\text{HP}}[z] = H_t[z] H_{\text{HP},4}[z]. \quad (2.38)$$

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

$$\begin{aligned} e_{\text{LS}} &= \sum_{n=0}^N |h_{\text{eq}}[n] - h_{t,\text{HP}}[n]|^2 = (\mathbf{h}_{\text{eq}} - \mathbf{h}_{t,\text{HP}})^H (\mathbf{h}_{\text{eq}} - \mathbf{h}_{t,\text{HP}}) = (\mathbf{M}\mathbf{d} - \mathbf{h}_{t,\text{HP}})^H (\mathbf{M}\mathbf{d} - \mathbf{h}_{t,\text{HP}}) \\ &= \mathbf{d}^H \mathbf{M}^H \mathbf{M}\mathbf{d} - 2\mathbf{d}^H \mathbf{M}^H \mathbf{h}_{t,\text{HP}} + \mathbf{h}_{t,\text{HP}}^H \mathbf{h}_{t,\text{HP}}, \end{aligned} \quad (2.39)$$

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

$$\begin{aligned}\nabla_{\mathbf{d}}(e_{\text{LS}}) &\stackrel{!}{=} 0 \\ 2\mathbf{M}^H \mathbf{M} \mathbf{d}_{\text{opt}} - 2\mathbf{M}^H \mathbf{h}_{t,\text{HP}} &= 0 \\ \mathbf{M}^H \mathbf{M} \mathbf{d}_{\text{opt}} &= \mathbf{M}^H \mathbf{h}_{t,\text{HP}} \\ \mathbf{d}_{\text{opt}} &= (\mathbf{M}^H \mathbf{M})^{-1} \mathbf{M}^H \mathbf{h}_{t,\text{HP}}.\end{aligned}\quad (2.40)$$

The resulting well-known least squares (LS) solution contains the Moore-Penrose pseudo-inverse $\mathbf{M}^+ = (\mathbf{M}^H \mathbf{M})^{-1} \mathbf{M}^H$ so that we can simplify eq. (2.40) to

$$\mathbf{d}_{\text{opt}} = \mathbf{M}^+ \mathbf{h}_{t,\text{HP}}. \quad (2.41)$$

The resulting filter-coefficients $d_{k,0}, d_{k,1}, a_{k,0}, a_{k,1}, a_{k,2}, b_m$ can then be used to filter directly in real-time using a parallel filter algorithm or compute an equalizing IR $h[n]$ by the filtering of a unit Dirac-delta pulse $h_t[n]$ with the parallel filter as

$$a_{k,0} h[n] = \sum_{k=1}^K (d_{k,0} h_t[n] + d_{k,1} h_t[n-1] - a_{k,1} h[n-1] - a_{k,2} h[n-2]) + \sum_{m=0}^M b_m h_t[n-m]. \quad (2.42)$$

2.4 Minimum-Phase FIR Filter

A straightforward minimum-phase FIR implementation is performed using a similar procedure to compare the results. Here, the free FIR coefficients d_k collected in $\mathbf{d}_{\text{opt,FIR}} = [d_{0,\text{FIR}}, d_{1,\text{FIR}}, \dots, d_{K,\text{FIR}}]^T$ can directly calculated, by solving the least square error as

$$\mathbf{d}_{\text{opt,FIR}} = \mathbf{M}_{\text{FIR}}^+ \mathbf{h}_{t,\text{HP}}, \quad (2.43)$$

where \mathbf{M}_{FIR} contains just the delayed measured and averaged IR $h_s[n]$ so that

$$\mathbf{M}_{\text{FIR}} = \begin{bmatrix} h_s[0] & 0 & 0 & 0 & 0 \\ h_s[1] & & \vdots & \vdots & \vdots \\ h_s[2] & \ddots & h_s[N] & h_s[N-1] & h_s[N-2] \\ \vdots & & 0 & h_s[N] & h_s[N-1] \\ 0 & 0 & 0 & 0 & h_s[N] \end{bmatrix}. \quad (2.44)$$

As far as the FIR coefficients directly correspond to the IR, the resulting vector $\mathbf{d}_{\text{opt,FIR}}$ contains the equalizing IR $h_{\text{FIR}}[n]$. The number of L coefficients bound the frequency resolution. Using a sampling frequency of $f_s = 44\,100\,\text{Hz}$ a $L = 4096$ tap length filter results in a frequency resolution of

$$\Delta f_{\text{FIR}} = \frac{f_s}{L} = \frac{44\,100\,\text{Hz}}{4096} = 10.77\,\text{Hz}, \quad (2.45)$$

which almost corresponds to the third-octave resolution at low frequencies ($\Delta f_{3\text{th}/\text{Oct},\text{LF}} = 40\,\text{Hz} - 31.5\,\text{Hz} = 8.5\,\text{Hz}$). This filter length results in a minimal processing delay Δt_p of

$$\Delta t_{p,\text{FIR}} = \frac{L}{f_s} = \frac{4096}{44\,100\,\text{Hz}} = 92.88\,\text{ms}. \quad (2.46)$$

Fig. 2.6 compares the $L = 4096$ tap FIR filter with a $K = 28$ poles parallel IIR filters with $M = 1$, both filtered with an exemplary $h_s[n]$. Besides a slight offset in magnitude, after filtering with the initial measured IR, the resulting filter and magnitude show almost the same behavior. Viewing

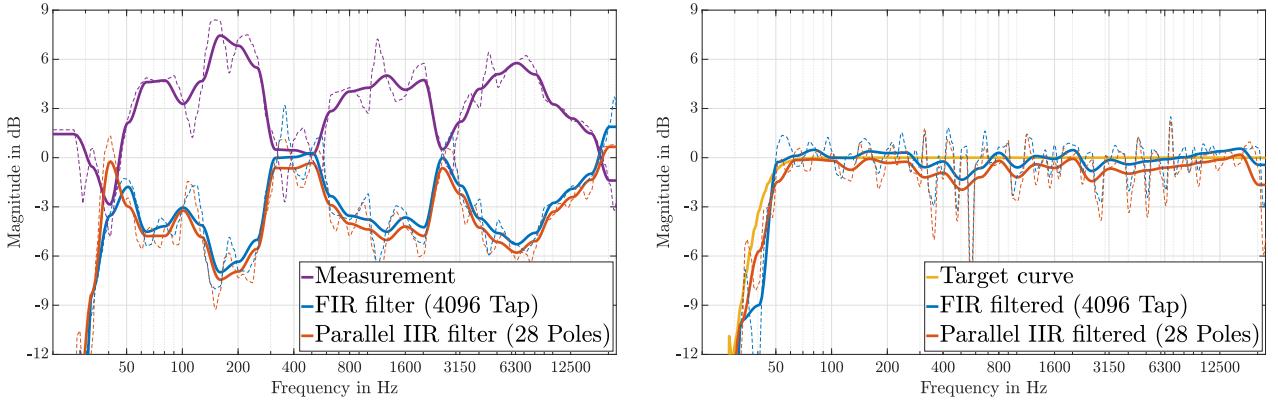


Figure 2.6: Comparison of 4096 Tap FIR filter and 28 Pole Parallel IIR filter

on eq. (2.47), the computational effort using a $L = 4096$ tap FIR filter is expectably higher than the $K = 28$ poles parallel IIR filters with $M = 1$ FIR sections, while the frequency resolution remains equal. Furthermore, the delay of $\Delta t_{p,FIR} = 92.88$ ms is not suitable for live-monitored real-time computations compared to the significantly smaller delay of two samples ($\Delta t_{p,IIR} = 0.045$ ms) implementing the parallel IIR filters, which can be assumed as real-time capable.

$$\begin{aligned} \text{FIR : } & \begin{cases} L - 1 = 4095 & \text{Additions per sample} \\ L = 4096 & \text{Multiplications per sample} \end{cases} \\ \text{Parallel IIR : } & \begin{cases} 4K - 1 + (M + 1) = 4 \times 28 - 1 + 2 = 113 & \text{Additions per sample} \\ 5K + (M + 1) = 5 \times 28 + 2 = 142 & \text{Multiplications per sample} \end{cases} \end{aligned} \quad (2.47)$$

2.5 Linear-Phase FIR Filter

Another common demand for equalizing loudspeakers in a room, preserving the initial phase, are linear-phase FIR filters. Performing a similar procedure using the LMS algorithm as based in eq. (2.32) is implemented by using the inverse magnitude of the measurement as a target response and a \mathbf{M} -matrix consisting of several sinusoids, representing the frequency bins. In the frequency domain, a general representation of a filtering transfer function $H(e^{j\omega})$ can be expressed with

$$H(e^{j\omega_k}) = \sum_{l=-L/2}^{L/2} h_l e^{-j\omega_k l} \quad k = 0, 1, 2, \dots, \text{NFFT} - 1. \quad (2.48)$$

Designing a filter with linear phase, all complex filtering parts (sinus terms) that contain the phase information can be neglected by using the symmetry property ($h_n = h_{-n}$) as well as shifting the impulse response by $L/2$ to make it causal, leads to

$$H(\omega_k) = \sum_{l=-L/2}^{L/2} h_l (\cos(\omega_k l) + j \sin(\omega_k l)) = h_0 \underbrace{\cos(\omega_k (l=0))}_{=1} + 2 \sum_{l=1}^{L/2} h_l \cos(\omega_k l). \quad (2.49)$$

Having the target response of $H(\omega_k)$ and an unknown number of L filter coefficients h_l , the expression in the frequency domain with linear phase can be obtained with

$$H(\omega_k) = h_0 + 2 \sum_{l=1}^{L/2} h_l \cos(\omega_k l). \quad (2.50)$$

Concerning this filtering procedure, our general solution for achieving an equalizing filter from a magnitude response reads similar to eq. (2.43), where $\mathbf{h}_{t,HP}$ is replaced with \mathbf{H}_t , the inverse of the

magnitude $|\mathcal{F}\{\mathbf{h}_s\}|^{-1}$ of the measurement \mathbf{h}_s filtered with the high-pass filtered target response $\mathbf{h}_{t,HP}$ such that,

$$\mathbf{H}_t = \left| \frac{\mathcal{F}\{\mathbf{h}_{t,HP}\}}{\mathcal{F}\{\mathbf{h}_s\}} \right|. \quad (2.51)$$

In matrix notation eq. (2.50) can be rewritten with L as the number of FIR-taps and ω_k as the k -th frequency bin in range of $[0, 2\pi]$

$$\begin{aligned} \mathbf{H}_t &= \mathbf{M}_{\text{linFIR}} \quad \mathbf{d}_{\text{opt,linFIR}} \\ \begin{bmatrix} H(\omega_0) \\ H(\omega_1) \\ \vdots \\ H(\omega_k) \end{bmatrix} &= \begin{bmatrix} 1 & 2\cos(\omega_0) & 2\cos(2\omega_0) & \dots & 2\cos(\frac{L}{2}\omega_0) \\ 1 & 2\cos(\omega_1) & 2\cos(2\omega_1) & \dots & 2\cos(\frac{L}{2}\omega_1) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 2\cos(\omega_k) & 2\cos(2\omega_k) & \dots & 2\cos(\frac{L}{2}\omega_k) \end{bmatrix} \begin{bmatrix} h_{\text{lin}}[0] \\ h_{\text{lin}}[1] \\ \vdots \\ h_{\text{lin}}[\frac{L}{2}] \end{bmatrix} \end{aligned} \quad (2.52)$$

Solving the set of equations as before, using the solution obtained by the LMS algorithm, we achieve

$$\mathbf{d}_{\text{opt,linFIR}} = \mathbf{M}_{\text{linFIR}}^+ \mathbf{H}_t. \quad (2.53)$$

In order to fulfill the requirements of symmetry of a linear phase filter, we assemble the filter as

$$h_{\text{linFIR}}[n] = \begin{cases} h_{\text{lin}}[\frac{L}{2} - n] & \text{for } 0 < n < \frac{L}{2} - 1 \\ h_{\text{lin}}[0] & \text{for } n = \frac{L}{2} \\ h_{\text{lin}}[n - \frac{L}{2}] & \text{for } \frac{L}{2} + 1 < n < L \end{cases}. \quad (2.54)$$

Fig. 2.7 compares the impulse, magnitude, and phase responses of a linear phase FIR filter of 8192 samples with a FIR filter of length 4096 samples. 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 filters show the same trace, while the phase response of the linear-

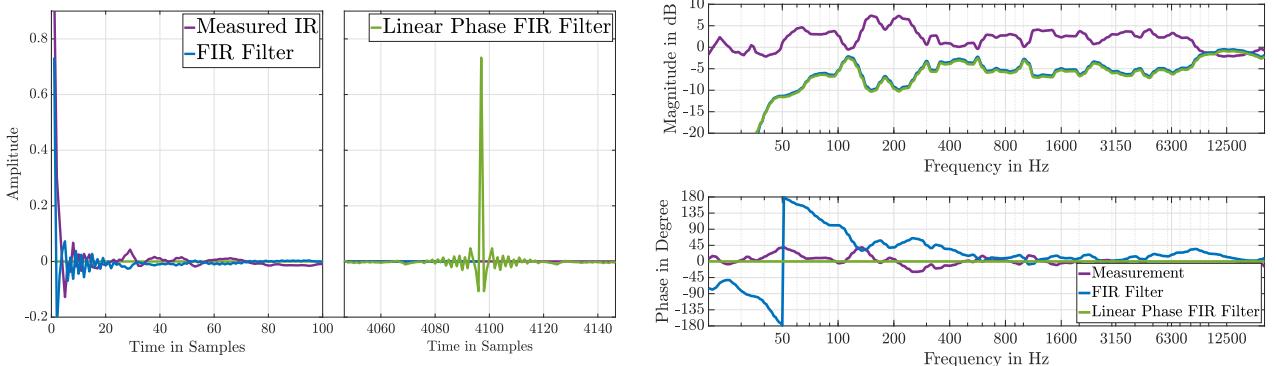


Figure 2.7: Comparison of linear-phase FIR with normal FIR filter

phase filter is exceptionally zero. The disadvantage of the linear-phase filter is the significantly higher latency and pre-ringing effects. 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 Software Implementation

3.1 Structure

The current chapter gives an overview of the implementation structure, shown in Figure 3.1. The implementation was done in two steps. First, a general implementation was realized in *MATLAB*. In

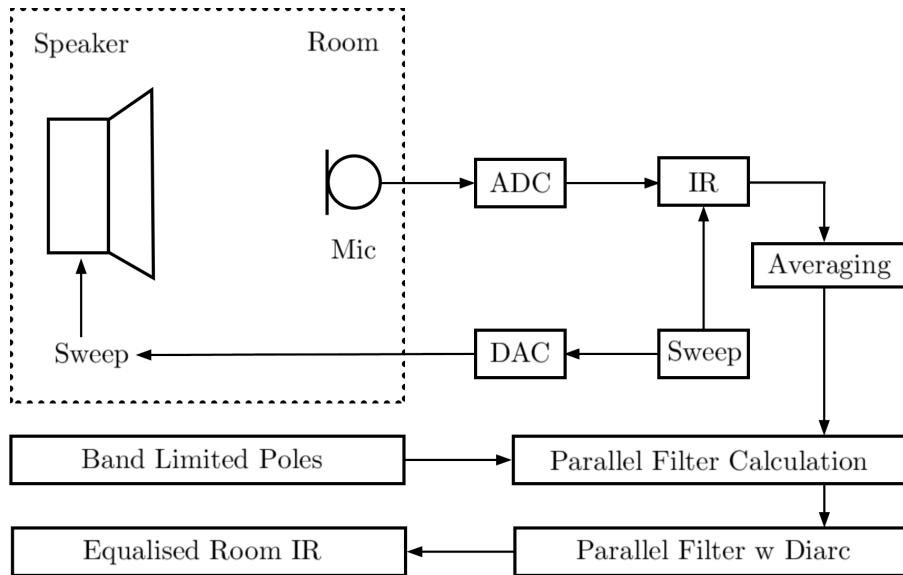


Figure 3.1: Implementation structure

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. 2.3.2.
5. The modeling matrix \mathbf{M} comprises the all-pole filtered average measured IR calculated corresponding to sec. 2.3.4.
6. The LMS algorithm calculates the optimal nominator weights as described in sec. 2.3.5.
7. Optionally, a Dirac impulse $\delta[n]$ can be filtered with the parallel filter, resulting in an IR with rectifying properties.

3.2 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. 3.2 shows the app's graphical user interface (GUI).

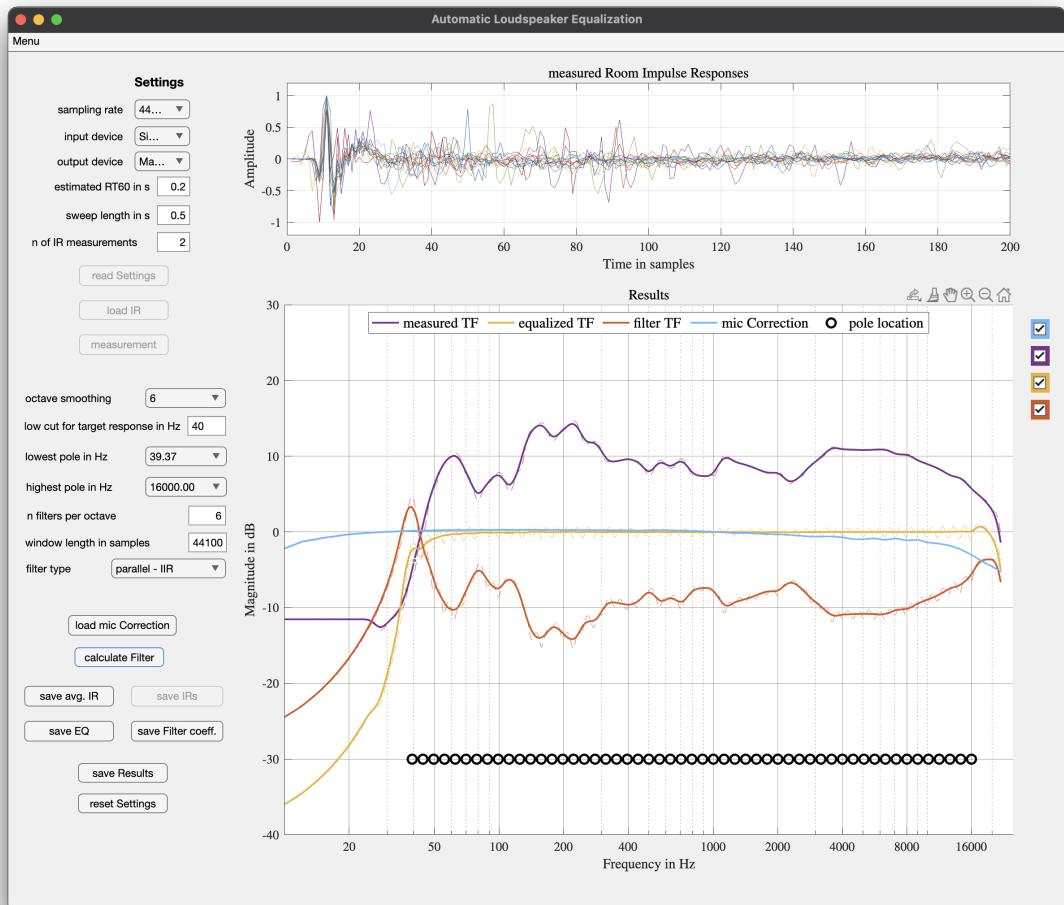


Figure 3.2: App overview

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 ($f_{\text{low}}, f_{\text{high}}$) 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 [KP]. Twenty-four pole frequencies per octave are pre-calculated between an interval $f_k \in [15.625, 32000]\text{Hz}$, diminished and adjusted as the user specifies. Fig. 3.3 shows an exemplary logarithmic pole distribution in the entire interval with $n_{\text{oct}} = 3$.

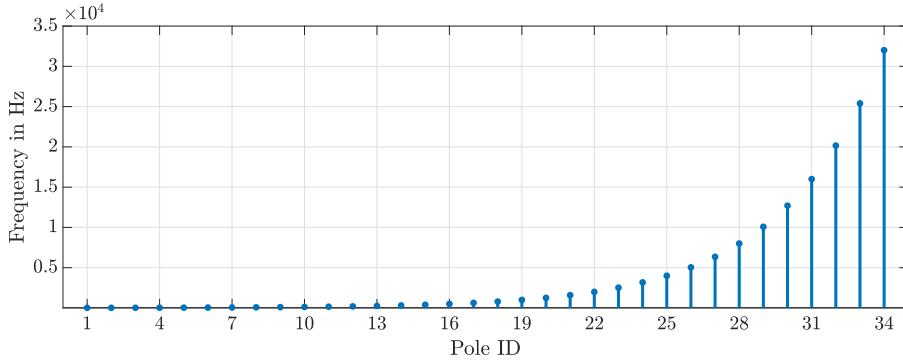


Figure 3.3: logarithmic pole distribution.

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. 3.1, 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. (2.38) in sec. 2.3.5. 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. (2.46).

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.

4 Results

4.1 Window Length

First, the impact of the window length is evaluated. Fig. 4.3 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.

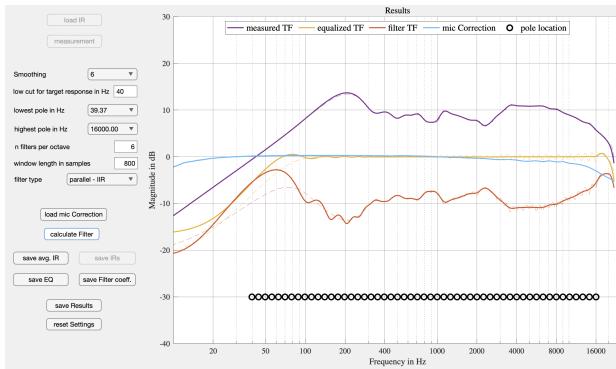


Figure 4.1: 800 samples truncation

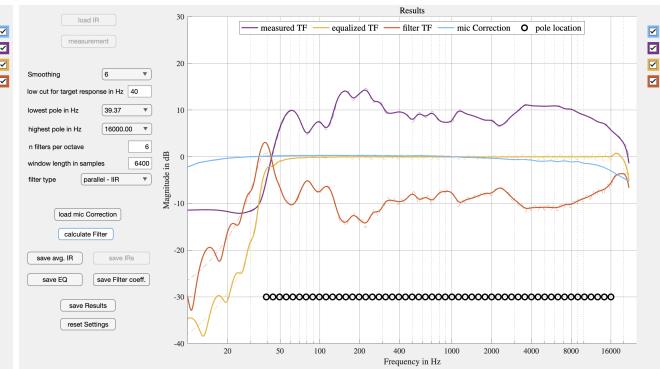


Figure 4.2: 6400 samples truncation

Figure 4.3: Calculation with pole distribution with $n = 6$, and truncation of measured IRs

4.2 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 4.4 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. 4.4, 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.

4.3 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. 4.4, which is the

¹<http://www.rationalacoustics.de/smaart-v8.html>

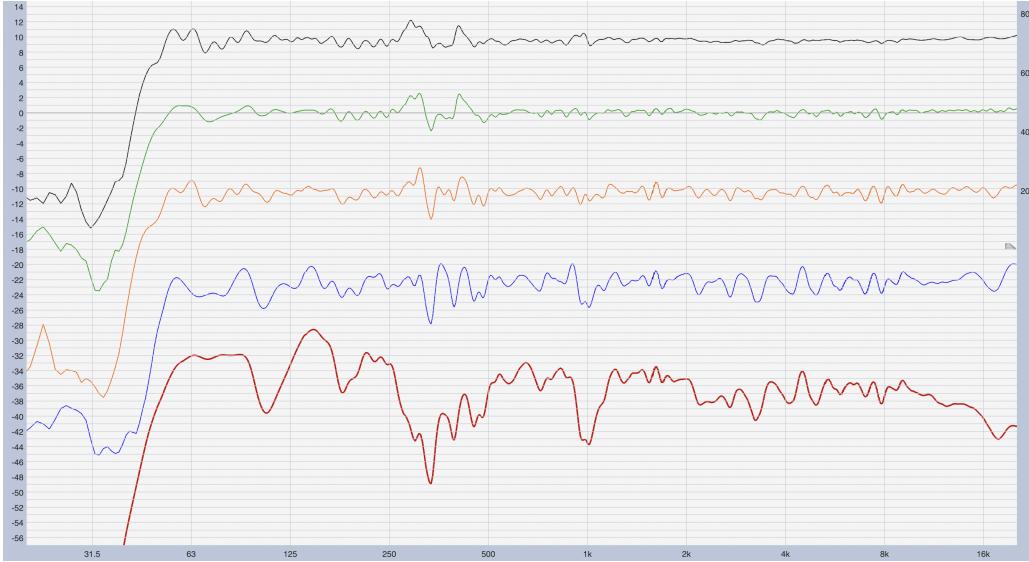


Figure 4.4: Validation measurement with Smaart v8. red: measurement, blue: corrected with three poles/octave, orange: corrected with six poles/octave, green: corrected with 12 poles/octave, black: corrected with 24 poles/octave. All smoothed to 1/12 octaves

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.

4.4 Parallel, Minimum-Phase FIR, and 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 Figures 6.3, 6.4, 6.5, shown in appendix 6, the three filter magnitudes are depicted. 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

$$\Delta f_{lp,FIR} = \frac{f_s}{L/2} = \frac{44.1 \text{ kHz}}{4097/2} = 21.53 \text{ Hz}, \quad (4.1)$$

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

5 Conclusion

In this seminar report, a prototype application (Automatic Loudspeaker Equalization) was developed in MATLAB and published on Git¹, 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>

6 Appendix

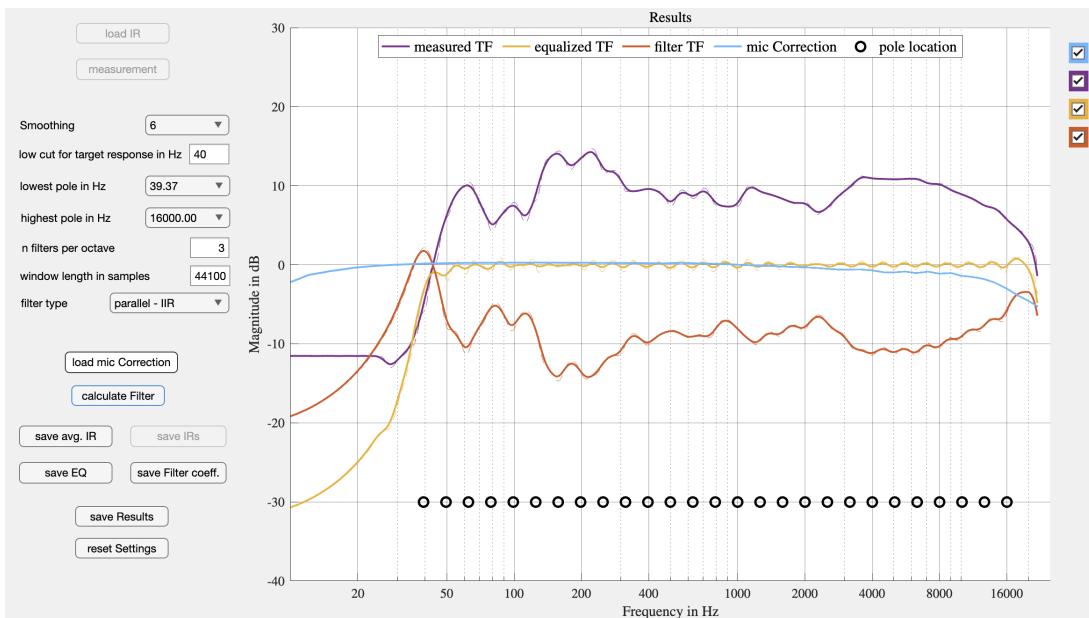


Figure 6.1: Calculation from the app for 3 poles/octave; All smoothed to 1/6 octaves

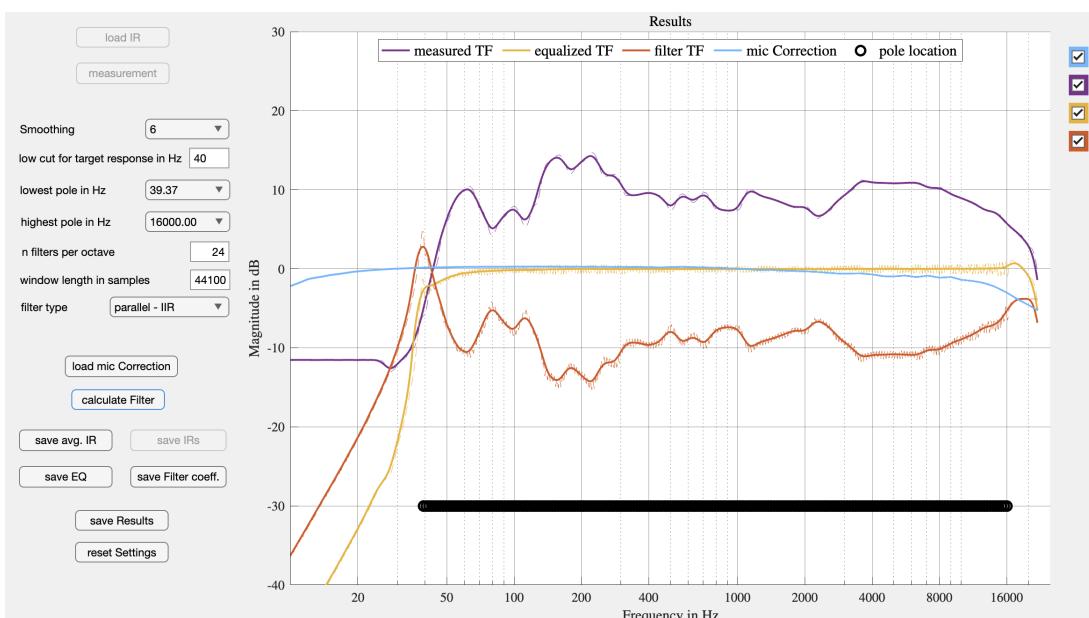


Figure 6.2: Calculation from the app for 24 poles/octave; All smoothed to 1/6 octaves

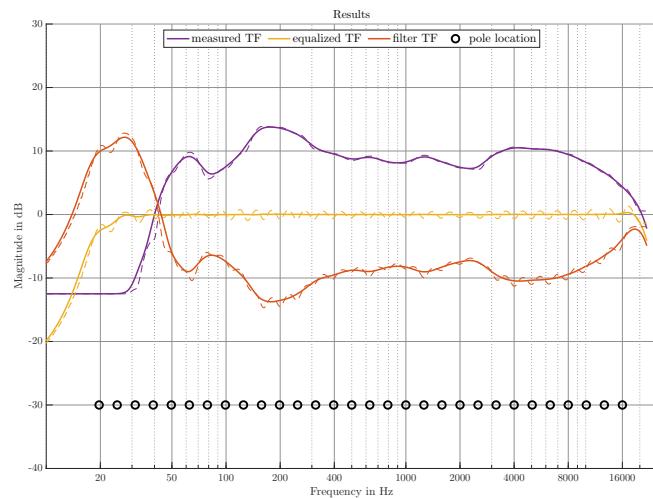


Figure 6.3: Parallel filter

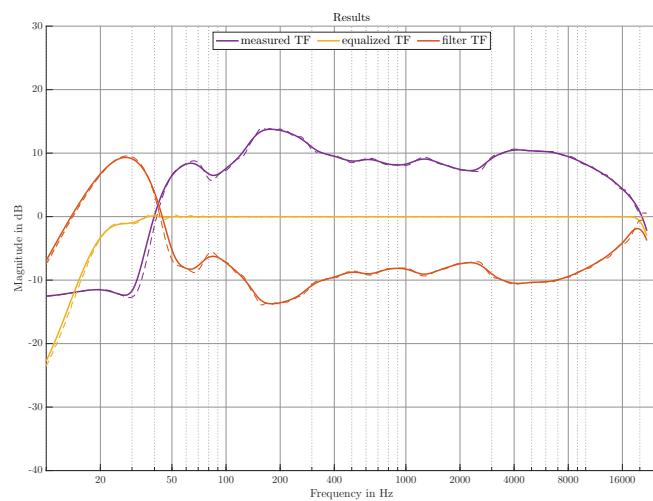


Figure 6.4: FIR filter

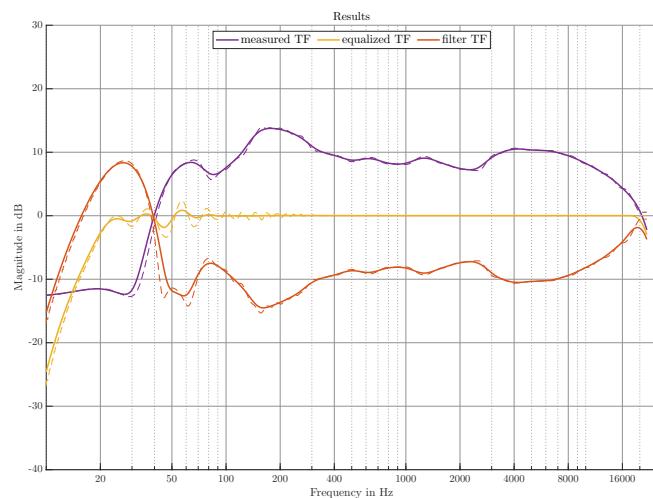


Figure 6.5: Linear-phase filter

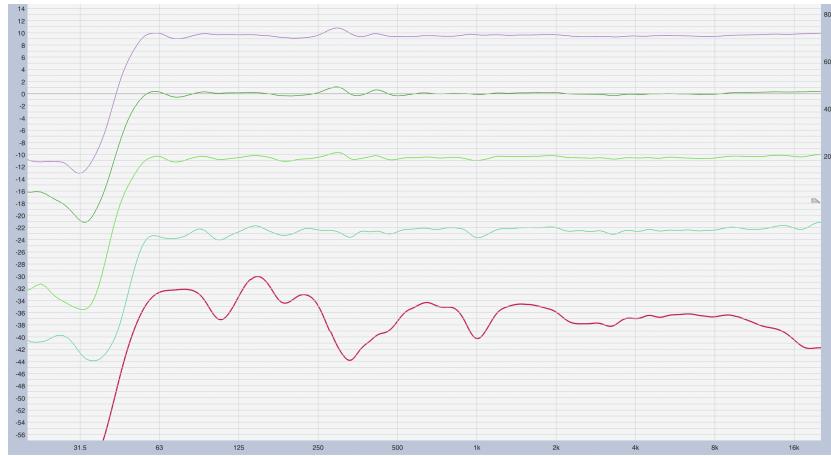


Figure 6.6: Validation measurement with Smaart v8. red: measurement, blue: corrected with 3 poles/octave, orange: corrected with six poles/octave, green: corrected with 12 poles/octave, black: corrected with 24 poles/octave. All smoothed to 1/3 octaves



Figure 6.7: 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

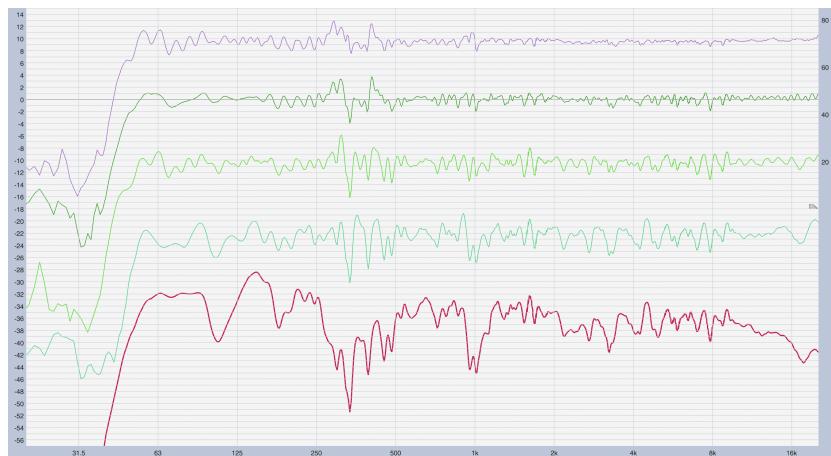


Figure 6.8: Validation measurement with Smaart v8. red: measurement, blue: corrected with 3 poles/octave, orange: corrected with 6 poles/octave, green: corrected with 12 poles/octave, black: corrected with 24 poles/octave. All smoothed to 1/24 octaves

Bibliography

- [Ban] B. Bank, “Perceptually motivated audio equalization using fixed-pole parallel second-order filters,” vol. 15, pp. 477 – 480.
- [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>
- [HCZ] 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*.
- [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>
- [KP] M. Karjalainen and T. Paatero, “Equalization of audio systems using kautz filters with log-like frequency resolution,” in *Audio Engineering Society Convention 120*. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=13571>
- [MBL] 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. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=14190>
- [Too09] F. Toole, *Sound Reproduction - The Acoustics and Psychoacoustics of Loudspeakers and Rooms*. Boca Raton, Fla: CRC Press, 2009.
- [Zö11] U. Zölzer, *DAFX - Digital Audio Effects*. New York: John Wiley Sons, 2011.