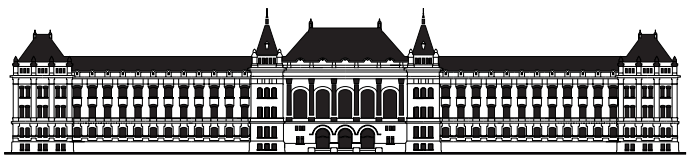
Logarithmic Frequency Resolution Filter Design for Audio Applications

Balázs Bank

Logarithmic Frequency Resolution Filter Design for Audio Applications

Balázs Bank



Budapest University of Technology and Economics
Department of Measurement and Information Systems
2021

Budapest University of Technology and Economics
Department of Measurement and Information Systems
H-1117 Budapest, XI. Magyar tudósok körútja 2.
Homepage: http://www.mit.bme.hu/∼bank
Email: bank@mit.bme.hu

Szüleimnek.

Preface

The main purpose of this thesis is to summarize my research contributions for the fulfillment of the requirements of the Doctoral Degree of the Hungarian Academy of Sciences. This is reflected by the writing style where I try to be as clear as possible which are my own contributions and how they are related to prior research. In addition, my main results are listed in the form of scientific statements at the end of the chapters outlining my own contributions.

My secondary goal with this work is to provide a coherent view on the various filter design techniques that I have developed with the hope that it can be useful for the broad signal processing community. Hence the choice of English language, and the textbook-like presentation with introductory chapters. Compared to the conference or journal papers where my results have been originally published, this format allows a more thorough description of the techniques, the inclusion of additional proofs, and finally sharing my thoughts on how these methods relate to each other and to the ones proposed in the literature.

My Ph.D. research has been mostly dealt with the physics-based synthesis of the piano. That topic has also required the development of some specific filter design techniques. However, I have turned my attention to general filter design only after the completion of my Ph.D. thesis. In year 2007, with the help of an EU Marie Curie grant, I have spent a year in the Acoustics Laboratory of Helsinki University of Technology (now Aalto University), where, while still working on sound synthesis topics, I have got acquainted with the work of Matti Karjalainen and Tuomas Paatero using Kautz structures for audio filter design. At that time the Kautz filter was the most efficient approach for designing IIR filters with a flexible resolution. In particular, it allowed the design of filters with logarithmic frequency resolution desirable for audio applications, which has been unrealizable with earlier techniques. Fascinated with the favorable properties of the Kautz filter, but discontented by its relatively complicated structure, an idea came to my mind to substitute the Kautz structure with that of ordinary parallel second-order filters, while keeping the fixed-pole design methodology. Thus, the fixed-pole parallel filter was born, allowing the same design accuracy as with the Kautz filter but using a filter structure that requires less arithmetic operations, fully parallelizable, and much easier to understand. With this moment I actually started my new research line of developing filter and equalizer design algorithms for audio applications. My goal was to reach better filter approximation compared to earlier methods, and to do so by keeping the relative simplicity of the approaches. I believe that keeping the algorithms simple is a key to make them easy to implement and thus most useful both for the academic community and the industry.

Both due to the required page limit of the thesis and to keep the work as coherent as possible, only my most significant results related to fixed-pole parallel filter design are included in this work and in the list of scientific statements. Most of the papers connected to these statements were published with my sole authorship. With the few publications where I had a coauthor I include only that part of the results which is clearly my own contribution. As for my other

research not included in the list of scientific contributions, but still belonging to the audio filter design field, I will provide a short summary in Sec. 10.1. (My less related contributions, even if they were done after my Ph.D. thesis, will not be outlined here, and the reader is referred to the list of my publications.)

In the following, I give the outline of this thesis. To start with the motivation of my research, Chapter 1 gives the reasons for designing filters with logarithmic frequency resolution and demonstrates that this is not possible by using traditional FIR and IIR design techniques. The idea of logarithmic frequency resolution is strongly related to fractional-octave smoothing, a long-time tradition in the audio field for displaying frequency responses, therefore this topic cannot be omitted from our discussion (Chapter 3). Prior techniques aiming at logarithmic frequency resolution filter design will be discussed in Chapters 4 and 5. I start outlining my own contributions in Chapter 6 by introducing the fixed-pole design of parallel second-order filters, giving the basic design techniques, demonstrating its equivalent approximation to Kautz filters and discussing its relation to transfer function smoothing. The choice of pole positions of the parallel filter has a decisive role, therefore Chapter 7 presents the pole positioning methods I have developed. It also includes a comparison to earlier techniques showing that with the proposed methodology improved performance can be achieved for the same filter order. Chapter 8 extends some of the design ideas to the multichannel case with the application to passive admittance modeling and to the equalization of MIMO systems. Finally, Chapter 9 introduces the idea of the delayed parallel filter with improved numerical properties compared to the traditional parallel form and proposes an alternative method for converting direct form IIR filters to parallel structures. Chapter 10 concludes my work by briefly outlining my other contributions related to filter design and discusses the significance and applications of the methods presented in this thesis.

At this point I would also like to express my gratitude to everyone who have helped along the way of reaching these results. To my coauthors, most of whom had become good friends, Federico Fontana, Julius Smith, late Matti Karjalainen, Stefano Zambon, and Vesa Välimäki, to name a few. I am also thankful to the colleagues at BME MIT, especially to my former Ph.D. supervisor, László Sujbert for highly useful comments about this manuscript, and to the head of the department, Tamás Dabóczi for his support and for periodically asking how I am progressing. Finally, I am most grateful to my family for creating the atmosphere and circumstances to become a researcher. I dedicate this thesis to my mother and to my late father, who had planted the seeds of curiosity and critical thinking already in my early childhood.

Contents

Chapter 1

Introduction

Modeling or equalizing a given transfer function is one of the most often used applications of digital filters in the field of audio. The most typical example is to correct the non-ideal frequency response of loudspeakers, ranging from speakers in mobile devices, computer speakers, car audio, to large-scale public address systems. Often the transfer function of the room is also equalized together with the loudspeaker, this is termed loudspeaker–room equalization. Other applications include the modeling of the same systems for simulation purposes, the modeling and equalization of headphones, the modeling of head related transfer functions for 3D audio, and the modeling of instrument body responses for physics-based sound synthesis, to name a few. For all of these applications a digital filter has to be designed that achieves the best sound quality at a given computational cost.

As in audio the final judge of quality is the human ear, it seems logical to take into account some of the properties of the auditory system during filter design. One such property that is used since the early times of audio is the logarithmic-like frequency resolution of hearing. Audio transfer functions are almost always displayed in a logarithmic frequency scale and they are often smoothed to some fractional octave resolution (see Chap. 3 for details). This long tradition is also justified by the study of Olive [2004a,b] demonstrating that the evenness of the log-scale smoothed magnitude response has a strong correlation with the subjective quality values of loudspeakers obtained by listening tests.

Also, graphic equalizers used to manually tune the response of an audio system have bands with center frequencies evenly distributed in the logarithmic scale. As a connection to music, it is interesting to note that the frequency distribution of the musical scale used in the western tradition (chromatic scale) is exactly logarithmic. When designing a digital filter for audio purposes it seems logical to design the filter so that the error is distributed evenly in the logarithmic scale, and indeed, this is the approach most often taken in the literature when designing filters for audio applications.

Note that various auditory frequency scales exist, like the Bark, mel, or ERB [Smith and Abel 1999; Zwicker and Fastl 1990] scale (which are actually all quite close to the logarithmic scale) that are sometimes used for audio applications. Since in audio the logarithmic scale is far the most commonly used, therefore the examples of this thesis will use the logarithmic scale. Using one type of frequency scale only has the benefit that the figures of the thesis will be comparable to each other. Nevertheless, since the new methods proposed here do not rely on the assumption of a logarithmic scale, they can be used with other type of perceptual frequency scales (or with any kind of user-defined frequency scale) if needed. In particular, the examples in Appendix A.4 will demonstrate the complete freedom in controlling the frequency resolution

by using the proposed techniques.

1.1 Limitations of general purpose filter design techniques

We must note that the term *filter design* is often used in a strict sense for designing low-pass, high-pass, band-pass or band-reject filters with a certain passband ripple and stopband attenuation [Parks and Burrus 1987; Oppenheim et al. 1999]. On the contrary, this work uses the term *filter design* as a general method for approximating an arbitrary impulse response or frequency response.

Modeling or equalizing a transfer function by an FIR or IIR filter is a common task in other fields of digital signal processing as well. Therefore, a wide range of filter design techniques exist that at first glance seem to be also appropriate for audio applications. However, these general filter design methods have an important property which makes them less than optimal for audio. Namely, they have linear frequency resolution, meaning that the error of the filter is distributed evenly in the linear frequency scale. Linear frequency resolution is inherent in FIR filters, since their frequency response is given as the DFT of their impulse response, leading to a transfer function vector with linearly spaced frequency bins. Thus, the frequency resolution Δf is directly determined by the length of the filter N and the sampling frequency f_s, given as the resolution of the DFT: $\Delta f = f_\mathrm{s}/N$.

Also, many IIR filter design algorithms (e.g., Prony [Parks and Burrus 1987], Steiglitz-McBride [Steiglitz and McBride 1965]) minimize the error between the target impulse response and the filter response in the mean squared sense, which, due to Parseval's theorem, is equivalent to minimizing the mean squared error between the target and filter frequency response in a linear frequency scale. Some frequency-domain IIR filter design methods allow the use of a weighting function (e.g., the frequency-domain Steiglitz-McBride algorithm [Jackson 2008] or the `invfreqz` function in MATLAB), or their target frequency scale can be made logarithmic instead of linear. So in theory it should be feasible to achieve a logarithmic frequency resolution by these methods. However, the logarithmic scale is so distorted compared to the linear one that this is not working in practice [Waters and Sandler 1993].

The following examples will demonstrate the difficulties of achieving logarithmic frequency resolution with general IIR filter deign methods. The example case is modeling a loudspeaker–room response. In Fig. 1.1 a 100th order IIR filter is designed by the frequency-domain Steiglitz-McBride algorithm [Jackson 2008]. The target is a minimum-phase loudspeaker–room response, and the target points are linearly distributed in frequency: 10000 points from 0 Hz to half of the sample rate $f_\mathrm{s}/2 = 22050$ Hz. Note that the sample rate $f_\mathrm{s} = 44100$ Hz is the one most commonly used in audio, and this sample rate will be used for all the examples of this work, unless otherwise noted. Due to the linear distribution of target frequency points, we expect that the error will be linearly distributed in frequency. This is indeed visible in Fig. 1.1 (a) where the filter response (thick solid line) follows the target (dashed line) by the same accuracy for all frequencies, which is also indicated by the fact that the frequencies of the filter poles are evenly distributed in the linear frequency scale (see the crosses in Fig. 1.1). When we plot the same curves in a logarithmic scale in Fig. 1.1 (b), we see a different picture and the lack of modeling ability at low frequencies is immediately apparent.

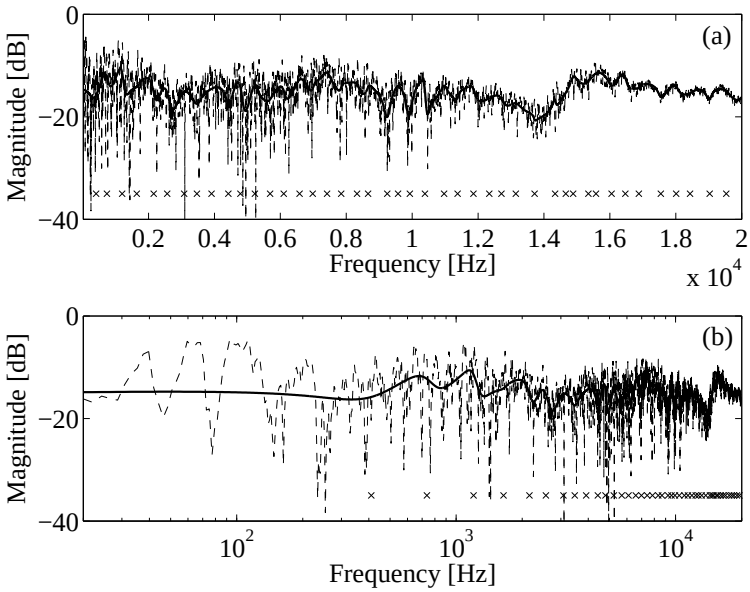


Figure 1.1: 100th order IIR filter design by the frequency-domain Steiglitz-McBride algorithm. The minimum-phase loudspeaker–room response is displayed by dashed line, and the modeled response by a thick solid line. The target frequency points are linearly spaced. The same responses are plotted in (a) linear and in (b) logarithmic frequency scale. The crosses indicate the pole frequencies of the filter.

The next step is to apply frequency weighting. In the example of Fig. 1.2 a weighting of $W_{\mathrm{t}}(f) = 1/f^2$ is used, otherwise the design is the same as for Fig. 1.1. (Note that $W_{\mathrm{t}}(f)$ is limited at $f = 20$ Hz to avoid the very large weights at the otherwise irrelevant infrasonic frequencies.) It can be seen in Fig. 1.2 (b) that the fit gets only slightly improved at low frequencies, which is quite surprising since the lowest frequencies get 10^6 times larger weight in the error compared to the highest ones.

As a next trial, the filter design is based on a target response whose frequency points are logarithmically spaced. This will inherently mean that the error is minimized along the logarithmic scale. In Fig. 1.3 the target (dashed line) was resampled to a logarithmic scale with 100 bins per octave from 20 Hz to $f_{\mathrm{s}}/2 = 22050$ Hz, giving 1011 specification points. This is also visible when comparing the dashed lines of Fig. 1.3 to Figs. 1.1 and 1.2. No weighting is used, since the resampling of the frequency grid already assures that the error is minimized in the logarithmic scale. Strangely enough, the low-frequency modeling accuracy visible in Fig. 1.3 (b) is again practically the same as for the first design of Fig. 1.1 (b), so no improvements are found.

From the examples it appears that while in theory both the weighted design and the design based on a logarithmically spaced specification should result in a filter with logarithmic frequency resolution, this is not happening in practice. The examples used only the frequency-domain Steiglitz-McBride algorithm [Jackson 2008], but the same is happening with other

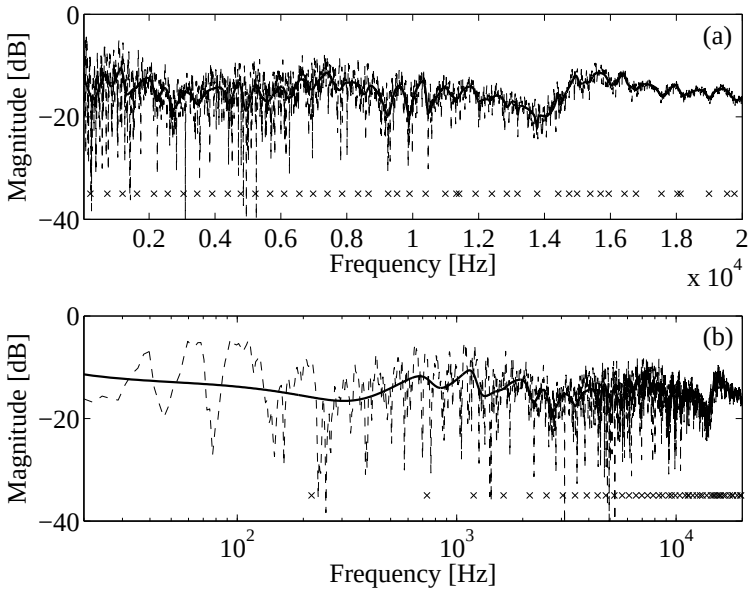


Figure 1.2: 100th order IIR filter design by the frequency-domain Steiglitz-McBride algorithm. The minimum-phase loudspeaker–room response is displayed by dashed line, and the modeled response by a thick solid line. The target frequency points are linearly spaced, and a $1/f^2$ type weighting is used. The same responses are plotted in (a) linear and in (b) logarithmic frequency scale. The crosses indicate the pole frequencies of the filter.

methods, like `invfreqz` in MATLAB. This complies with the findings of [Waters and Sandler 1993] who have found that traditional IIR design techniques using weighted least squares minimization fail to converge when the target frequency points are logarithmically spaced.

The reasons for this are most probably of numerical nature. Logarithmic frequency resolution would require a very high pole density at low frequencies with poles near the unit circle, and this cannot be implemented by direct form IIR filters even at double precision. (This can be easily tested by converting the logarithmic frequency resolution filters discussed later in this work to a direct form filter: the resulting filter is almost always unstable and its frequency response is very different from the original due to numerical errors.) Traditional IIR filter design algorithms estimate the parameters of a direct form IIR filter, they thus cannot give such a set of coefficients which would lead to the desired high pole density at low frequencies, since such a set of coefficients do not exist in the space of available numbers due to finite precision.

1.2 Specialized filter structures for achieving non-uniform frequency resolution

By recognizing these limitations, specialized IIR filter design techniques have been developed that allow a more flexible distribution of modeling detail as a function of frequency. It is interesting to note that all of these methods use special filter structures instead of the direct (rational) form used for general purpose IIR filters and the filters are designed directly in these special forms.

This work, after introducing warped [Härmä et al. 2000] and Kautz [Paatero and Karjalainen 2003] filters, will focus on fixed-pole parallel filters, a methodology allowing the design of IIR

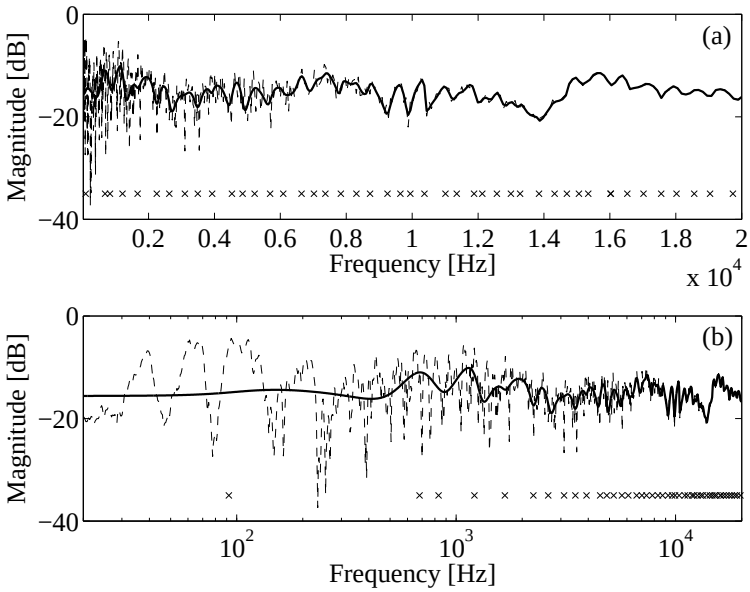


Figure 1.3: 100th order IIR filter design by the frequency-domain Steiglitz-McBride algorithm. The minimum-phase loudspeaker–room response is displayed by dashed line, and the modeled response by a thick solid line. The difference from Fig. 1.1 is that target frequency points are logarithmically spaced. The same responses are plotted in (a) linear and in (b) logarithmic frequency scale. The crosses indicate the pole frequencies of the filter.

filters at arbitrary frequency resolution profiles, while still leading to a simple filter structure with low computational complexity.

It has to be mentioned that another way to address the problem of logarithmic frequency resolution comes from manually tuned graphic or parametric equalizers that exist from the early times of audio. Automatic techniques have been developed that iteratively tune the parameters (center frequency, Q-factor, gain) of such equalizers by a nonlinear optimization process (see, e.g., [Ramos and López 2006; Behrends et al. 2011; Vairetti et al. 2018]). This approach is advantageous when there is a need to manually fine-tune the filter response by listening, since the parameters of the equalization filters are perceptually meaningful and well understood by sound engineers. Also, interpolation between different filter settings is relatively easily achieved. A drawback that the special form of the filter sections limits the degrees of freedom (3 parameters instead of 4 for a general second-order section) and thus results in lower accuracy for the same filter order (see [Bank and Ramos 2011] Fig. 3 for a comparison example). In addition, these techniques allow magnitude equalization only, while warped, Kautz or parallel filters are also able to model or equalize the phase behavior if desired. Finally, for the above parametric equalizer approach a more complex optimization process is necessary, as opposed to the relatively simple parameter estimation techniques required for warped, Kautz, or parallel filters. It can be thus said that while aiming at a similar goal, the automatic tuning of parametric equalizers is a different line of research and therefore it will not be further discussed here. For an overview, the reader is referred to [Vairetti et al. 2018].

Chapter 2

Basic concepts and methods

Here a quick summary of the basic concepts used in this thesis is provided, which will hopefully also help in understanding the notations used in the following chapters.

2.1 Approximating a target response by a digital filter

Modeling a system by a digital filter requires that the filter response is close to the system response. In the time domain this means that the error between the target impulse response $h_\mathrm{t}(n)$ and the filter impulse response $h(n)$ is minimal

$$||h(n) - h_\mathrm{t}(n)|| \to \min \tag{2.1}$$

for the sample range $[0, 1, \ldots, N-1]$, where $||x(n)||$ is some kind of error norm.

In the frequency domain, the error between the target frequency response $H_\mathrm{t}(\vartheta_n)$ and the filter frequency response $H(\vartheta_n)$ is minimized

$$||H(\vartheta_n) - H_\mathrm{t}(\vartheta_n)|| \to \min \tag{2.2}$$

for a finite set of angular frequencies $[\vartheta_1, \vartheta_2, \ldots, \vartheta_N]$. The angular frequencies are related to the analog frequencies f_n as

$$\vartheta_n = 2\pi \frac{f_n}{f_\mathrm{s}}, \tag{2.3}$$

where f_s is the sampling frequency.

Note that the target frequency points do not have to be evenly distributed. By the proper distribution of the frequency points ϑ_n we can control the modeling detail: those regions which have more target points will have a larger weight in the total error, thus, they will force the optimization procedure to obtain a better fit in that region. In audio usually a logarithmically spaced ϑ_n is employed and this should in theory lead to an even distribution of errors in the logarithmic frequency scale. However, as we have seen in Chap. 1, this alone is not enough to achieve a logarithmic frequency resolution.

As for the norms$||x(n)||$ used in Eqs. (2.1) and Eq. (2.2), usually the L_p norm is applied which is defined as

$$||x(n)||_p = \left(\sum_{n=0}^{N-1} |x(n)|^p \right)^{\frac{1}{p}}, \tag{2.4}$$

where the $p = \infty$ (minimax) and $p = 2$ (least squares or LS) are the most typical choices. For classic lowpass, highbass, bandpass, or bandreject filters, usually the L_∞ norm is utilized (e.g., by the Chebyshev I, II and elliptic IIR filters or by the Parks-McClellan algorithm for FIR filters [Parks and Burrus 1987]), while for designing filters with an arbitrary target response, the L_2 norm is the usual choice. Complying with this common practice, this work also employs the L_2 norm in the proposed algorithms. The rationale for this choice is described in the Appendix A.1 in more detail. It must be noted however that the majority of the proposed algorithms can be modified for applying other norms, most simply by introducing an iterative weighting into the LS design as described in [Vargas and Burrus 2001; Kobayashi and Imai 1990].

2.2 Equalizing a system response

Besides modeling, a common task in signal processing is to design a filter that equalizes a given system. This means that when the equalizer is connected in series with the system, then the total response should be close to the target. Connecting in series means, for example, for loudspeaker–room equalization that the digital signal (music) is first filtered by the equalizer filter, and then converted to the analog domain, amplified and fed to the loudspeaker. In contrast, equalizing the response of a microphone would mean that the filter comes afterwards: the digitized signal of the microphone is filtered by the equalizer. Since in this thesis we assume that the systems are linear and time-invariant (in other words, we model or equalize only the linear aspects), the sequence of the system and the equalizer does not matter, and the two cases can be handled jointly.

2.2.1 Direct equalizer design

In the time domain, the impulse response of the equalized system $h(n)$ is the convolution of the system impulse response $h_\mathrm{s}(n)$ and the equalizer response $h_\mathrm{eq}(n)$, that is, $h(n) = h_\mathrm{s}(n) * h_\mathrm{eq}(n)$, and this has to be close to the target impulse response $h_\mathrm{t}(n)$. Mathematically, this is expressed as

$$||h(n) - h_\mathrm{t}(n)|| = ||[h_\mathrm{s}(n) * h_\mathrm{eq}(n)] - h_\mathrm{t}(n)|| \rightarrow \min. \tag{2.5}$$

In the frequency domain, the frequency response of the equalized system $H(\vartheta_n)$ is the product of the system frequency response $H_\mathrm{s}(\vartheta_n)$ and the equalizer frequency response $H_\mathrm{eq}(\vartheta_n)$, that is, $H(\vartheta_n) = H_\mathrm{s}(\vartheta_n)H_\mathrm{eq}(\vartheta_n)$, and this has to be close to the target frequency response $H_\mathrm{t}(\vartheta_n)$. Mathematically, this is expressed as

$$e = ||H(\vartheta_n) - H_\mathrm{t}(\vartheta_n)|| = ||H_\mathrm{s}(\vartheta_n)H_\mathrm{eq}(\vartheta_n) - H_\mathrm{t}(\vartheta_n)|| \rightarrow \min. \tag{2.6}$$

This procedure is termed *direct equalizer design*, since we design the equalizer directly. Note also that this is similar to a system identification problem where the task is to estimate the parameters of a digital filter so that at its input we have the system response and at the output we obtain the target response. Thus, we might also call the method *equalizer design based on system identification*.

2.2.2 Filter design based on the inverted system response

Another common choice for designing an equalizer is to define a new target $H'_{\mathrm{t}}(\vartheta_n)$ as the ratio of the target response and system response

$$H'_{\mathrm{t}}(\vartheta_n) = \frac{H_{\mathrm{t}}(\vartheta_n)}{H_{\mathrm{s}}(\vartheta_n)} \tag{2.7}$$

and then use the optimization form developed for filter design Eq. (2.2). Since this involves the inverse of the system response $1/H_{\mathrm{s}}(\vartheta_n)$, it will be termed *inversion-based equalizer design*.

Note this procedure can also be done in the time domain where $h'_{\mathrm{t}}(n)$ is computed by deconvolving the target impulse response $h_{\mathrm{t}}(n)$ with the system response $h_{\mathrm{s}}(n)$. Deconvolution can be either done by a least squares optimization [Mourjopoulos et al. 1982] or in the frequency domain as $h'_{\mathrm{t}}(n) = \mathrm{IFFT}\{H_{\mathrm{t}}(\vartheta_n)/H_{\mathrm{s}}(\vartheta_n)\}$. Then the equalizer is designed by Eq. (2.1) for the new target.

The advantage of the inversion-based equalizer design is that the simpler filter design forms Eqs. (2.1) and (2.2) can be used. Indeed, some filter algorithms are only available in the filter design form, and have no system identification option. For example, while `stmcb` in MATLAB can estimate a system between input and output, `prony` and `invfreqz` cannot. In the latter case, we can only use the inversion-based approach.

However, it should be clear that the direct and inversion-based methods are not equivalent, since they minimize different errors. Writing the error in the L_2 sense for the inversion-based equalizer gives

$$e = \sum_{n=1}^{N} \left| H_{\mathrm{eq}}(\vartheta_n) - \frac{H_{\mathrm{t}}(\vartheta_n)}{H_{\mathrm{s}}(\vartheta_n)} \right|^2 = \sum_{n=1}^{N} \frac{1}{|H_{\mathrm{s}}(\vartheta_n)|^2} |H_{\mathrm{s}}(\vartheta_n) H_{\mathrm{eq}}(\vartheta_n) - H_{\mathrm{t}}(\vartheta_n)|^2, \tag{2.8}$$

which is the same as the direct-equalzier design of Eq. (2.6) with the additional weighting term $1/|H_{\mathrm{s}}(\vartheta_n)|^2$. This means that wherever $H_{\mathrm{s}}(\vartheta_n)$ is small, the optimization will try to equalize it more accurately compared to other target points. It is especially problematic if $H_{\mathrm{s}}(\vartheta_n)$ is close to zero at some frequencies, because this will mean very large weights at those points. This of course can also be seen from Eq. (2.7) showing that whenever $H_{\mathrm{s}}(\vartheta_n)$ is close to zero, the new target $H'_{\mathrm{t}}(\vartheta_n)$ will be close to infinity.

Therefore, whenever possible, the direct equalizer design approach should be used. If this cannot be done, then some kind of preprocessing or regularization has to be applied before computing Eq. (2.7) that removes the dips of the system response. This may include limiting or smoothing the system response, or both (see, e.g. [Pedersen and Thomsen 2007]). In addition, a weighting with $|H_{\mathrm{s}}(\vartheta_n)|^2$ can be used to counteract the $1/|H_{\mathrm{s}}(\vartheta_n)|^2$ weighting inherent in Eq. (2.8), whenever the filter design algorithm offers this option.

2.3 Solving linear-in-parameter problems by the least squares (LS) method

Many filter design methods are based on solving a linear-in-parameter problem, where the transfer function or impulse response is a linear function of the filter coefficients. FIR filter design is linear in parameters both in the time- and frequency domain, and so is IIR filter design in equation error setting [Parks and Burrus 1987]. The impulse responses and transfer functions

of Kautz filters (Chap. 5) and fixed-pole parallel filters (Chap. 6) are also linear in their free parameters.

For all of these cases, the output $\mathbf{h}$ (a column vector) is a linear combination of the basis functions $\mathbf{m}_k$ with weights p_k, written as

$$\mathbf{h} = \sum_{k=1}^{K} p_k \mathbf{m}_k = \mathbf{M}\mathbf{p}, \tag{2.9}$$

where $\mathbf{M}$ is the modeling matrix, whose columns contain the vectors of the basis functions $\mathbf{m}_k$, and $\mathbf{p}$ is a column vector composed of the weights p_k. The p_k parameters should be set in such a way that the linear combination of the basis functions $\mathbf{h}$ is the closest to the target vector $\mathbf{h}_\mathrm{t}$, that is, $||\mathbf{h} - \mathbf{h}_\mathrm{t}||$ is minimal. When the L_2 norm is used, the solution can be found in one step in closed form, as opposed to more complicated (usually iterative) methods required for other norms.

Minimizing the error between the output vector $\mathbf{h}$ and the target vector $\mathbf{h}_\mathrm{t}$ in the mean squared sense can be written as

$$e_\mathrm{LS} = \sum_{n=1}^{N} |h_n - h_{\mathrm{t},n}|^2 = (\mathbf{h} - \mathbf{h}_\mathrm{t})^H (\mathbf{h} - \mathbf{h}_\mathrm{t}) = \\ (\mathbf{M}\mathbf{p} - \mathbf{h}_\mathrm{t})^H (\mathbf{M}\mathbf{p} - \mathbf{h}_\mathrm{t}) = \mathbf{p}^H \mathbf{M}^H \mathbf{M}\mathbf{p} - 2\mathbf{p}^H \mathbf{M}^H \mathbf{h}_\mathrm{t} + \mathbf{h}_\mathrm{t}^H \mathbf{h}_\mathrm{t}, \tag{2.10}$$

where $\mathbf{x}^H$ is the conjugate transpose of $\mathbf{x}$. The absolute value sign, and thus the conjugation is needed because this way the equations are also valid for complex vectors. Since Eq. (2.10) is quadratic in $\mathbf{p}$, it has a unique minimum, which can be found by taking the derivative of Eq. (2.10) with respect to $\mathbf{p}$, and setting it to zero [Schnell 1998; Parks and Burrus 1987]:

$$\frac{\partial e_\mathrm{LS}}{\partial \mathbf{p}} = 2\mathbf{M}^H \mathbf{M}\mathbf{p} - 2\mathbf{M}^H \mathbf{h}_\mathrm{t} = 0. \tag{2.11}$$

This means that we need to solve the following system of linear equations (the so called "normal equations") for $\mathbf{p}$:

$$(\mathbf{M}^H \mathbf{M})\mathbf{p} = (\mathbf{M}^H \mathbf{h}_\mathrm{t}), \tag{2.12}$$

where $(\mathbf{M}^H \mathbf{M})$ is a $K \times K$ full rank (invertible) square matrix, assuming that the K modeling signals $\mathbf{m}_k$ are independent. Both $(\mathbf{M}^H \mathbf{h}_\mathrm{t})$ and $\mathbf{p}$ are length K column vectors.

Thus, the optimal set of parameters $\mathbf{p}_\mathrm{opt}$ is given as

$$\mathbf{p}_\mathrm{opt} = \mathbf{M}^+ \mathbf{h}_\mathrm{t}, \tag{2.13a}$$
$$\mathbf{M}^+ = (\mathbf{M}^H \mathbf{M})^{-1} \mathbf{M}^H, \tag{2.13b}$$

where $\mathbf{M}^+$ is the Moore-Penrose pseudoinverse.

While Eq. (2.13) is useful for problems with a modest (few hundred) number of parameters, there are faster and numerically more robust alternatives to solve least-squares problems: this includes solving the normal equations Eq. (2.12) via Cholesky factorization or obtaining the solution by the QR decomposition of matrix $\mathbf{M}$ [Golub and Loan 2013].

2.3.1 Weighted least squares design

It is also possible to add a specific weight to certain elements of $\mathbf{h}$ when we sum the errors as

$$e_{\mathrm{WLS}} = \sum_{n=1}^{N} W_n |h_n - h_{\mathrm{t},n}|^2 = (\mathbf{h} - \mathbf{h}_\mathrm{t})^H \mathbf{W} (\mathbf{h} - \mathbf{h}_\mathrm{t}), \tag{2.14}$$

where W_n is the weight for the n-th element of the output vector $\mathbf{h}$, and $\mathbf{W}$ is the weighting matrix having W_n in its diagonal and zeros elsewhere. The minimum is obtained by the weighted-least-squares (WLS) solution [Schnell 1998; Parks and Burrus 1987]:

$$\mathbf{p}_{\mathrm{opt}} = (\mathbf{M}^H \mathbf{W} \mathbf{M})^{-1} \mathbf{M}^H \mathbf{W} \mathbf{h}_\mathrm{t}. \tag{2.15}$$

Note however that it is impractical to compute $\mathbf{p}_{\mathrm{opt}}$ as in Eq. (2.15), because it involves the multiplication with an almost empty matrix $\mathbf{W}$. By substituting $\mathbf{W} = \mathbf{W}'\mathbf{W}'$ into Eq. (2.15) where $\mathbf{W}'$ is a diagonal matrix containing $\sqrt{W_n}$, it is easy to see that we simply have to multiply the n-th element of $\mathbf{h}_\mathrm{t}$ and the n-th row of $\mathbf{M}$ by $\sqrt{W_n}$ before using Eq. (2.13) to obtain the same solution as with Eq. (2.15), but with less computations.

2.4 Minimum-phase filter design and equalization

IIR filters can be both minimum-phase and non-minimumphase, depending on the location of their zeros. Practical experience shows that minimum-phase systems can be modeled easier by IIR filters (i.e., with a lower filter order for the same magnitude accuracy) than their non-minimumphase counterparts. Let us take a look into this issue in the following.

For that we first note that if the order of the IIR filter was the same as that of the system to be modeled, there would be no complications in approximating non-minimumphase responses. However, in audio we are usually modeling or equalizing high-order systems by lower order filters, so the situation is different.

Minimum-phase systems have many important properties: if they have a rational transfer function, then their zeros also lie within the unit circle (thus, their inverse is stable), their phase response is linked to the logarithm of their magnitude response by the Hilbert transform [Oppenheim and Schafer 1975], and the energy in their impulse response is most concentrated to the beginning part of the response (minimum energy-delay property [Oppenheim et al. 1999]). This last property translates to the fact that their impulse response has the most decaying character for a given magnitude response.

For all of the perceptually motivated filter design techniques discussed in this work (warped, Kautz, and parallel filters) the net transfer function is proper (has the same numerator and denominator order). Causal IIR filters with a proper transfer function compose their impulse response as a sum of decaying exponentials as coming from partial fraction expansion; thus, they are most suited to model decaying impulse responses. This means that minimum-phase responses are easy targets for all of these methods due to their minimum energy-delay property.

Slightly non-minimumphase targets are usually also well modeled by IIR filters, but highly non-minimumphase responses, such as a room impulse responses with multiple reflections, or far-field instrument body responses where the main peak of the response comes after a few hundred samples, cannot be easily followed by the decaying response of IIR filters.

The equalization of non-minimumphase transfer functions by IIR filters is even more problematic than modeling them, since their zeros outside the unit circle would require unstable

poles in the equalizer for perfect equalization. One solution to this problem is to use a forward-backward filtering scheme and implement the acausal part of the IIR filter in the backward filtering block [Powel and Chau 1991; Maeng and Lee 1995; Kurosu et al. 2003]. However, this requires a complicated block-based processing and depending on the implementation, it can lead to harmonic distortion due to the variable-length truncation of the backward filter response. Thus, instead of the backward IIR filter it is both simpler and safer to use an additional FIR path in the filter for the early (acausal) part of the impulse response [Paatero and Karjalainen 2006; Karjalainen and Paatero 2007; Bank 2007] (using an FIR filter for the rising part of the response will be demonstrated for the parallel filter in Sec. 9.1). Note that in both cases the filter must be of course causal, actually the delayed version of the theoretically acausal filter. This additional delay might be problematic in some applications such as live sound reinforcement, and the rising part of the equalizer response can also cause audible artifacts termed *pre-echo* or *pre-ringing* [Karjalainen et al. 2005; Cecchi et al. 2018].

A straightforward choice to avoid these complications is to convert the system response to minimum-phase before filter or equalizer design [Karjalainen et al. 2005; Cecchi et al. 2018]. While using a minimum-phase target might seem to be a limitation at first glance, it is a common choice in audio signal processing. A physical motivation for minimum-phase filter design is that many systems in audio (microphones and one-way loudspeakers, the modal range of room frequency responses, analog parametric and graphic equalizers) have a minimum-phase response anyway. Another motivation is coming from perceptual principles: while there is evidence that the phase response of electroacoustic systems can be heard on certain special program material (see, e.g., [Liski et al. 2018]), it is generally accepted that the magnitude response has much larger importance in the observed quality of audio systems. Since minimum-phase responses are easier modeled or equalized by IIR filters, this means that by giving up the perceptually less important phase accuracy, lower filter orders are sufficient for the same precision in magnitude modeling or equalization. This explains why using a minimum-phase target is so common in the audio field.

Complying with this usual procedure, most of the design examples of this work apply a minimum-phase target specification. To make this clear to the reader, this will be always mentioned in the captions of the corresponding figures. Nevertheless, it is worth emphasizing that IIR filters can of course be used for modeling or equalization of non-minimumphase responses if required (see, e.g., Fig. 7.1 for non-minimumphase warped and parallel filters), albeit with less efficiency, that is, typically requiring higher filter order compared to their minimum-phase version.[1]

[1] As an intermediate case between minimum-phase and non-minimumphase filter design, I have developed a general method (usable with any filter design technique, not only by fixed-pole parallel filters) that aims at achieving magnitude and phase accuracy in those frequency regions where this is possible, and prioritizes the magnitude over the phase whenever this cannot be done. Since for this thesis I have decided to include only those results that are closely related to fixed-pole parallel filter design, the detailed description of *magnitude-priority filter design* is not given here. However, a quick summary and my related publications are listed in Sec 10.1.2.

Chapter 3

Fractional-octave smoothing

The quasi-logarithmic frequency resolution of human hearing is also reflected in how transfer functions are displayed in the audio field. From the earliest times, a logarithmic frequency scale is used, and often the magnitude response is smoothed at a fractional-octave (e.g., third octave) resolution. The motivation behind fractional-octave smoothing is that the original transfer function is too detailed for evaluation: the effects of small variations in practice cannot be heard, and the raggedness of the response might actually prevent us from observing the more important global shape of the response. In other words, the smoothed version of the transfer function gives a better estimate of the perceived timbre. While smoothing stems from analog signal analyzers, typically all current digital audio spectrum analyzers offer this option.

3.1 Magnitude- or power-smoothing

Traditionally, smoothing has only been applied for the magnitude response of audio systems. In the time of analog signal analyzers, a smoothed transfer function was either obtained with exciting the system with pink noise and measuring the output voltage of bandpass filters (e.g., having a Q value corresponding to a third-octave resolution), or with heterodyne spectrum analyzers the Y signal of the cathode ray tube representing the magnitude value was low-pass filtered. The same procedure was applied with plotters when the write speed in the Y direction was decreased by lowpass filtering.

In digital analyzers, the starting point is usually a spectrum computed via FFT, thus, a spectrum in linear frequency scale. The smoothed transfer function is then computed by averaging the squared magnitude response values around the frequency of interest. Averaging the magnitude itself is also an option, though less often used, and the term *magnitude smoothing* usually actually refers to power smoothing in the literature. Since we are aiming at a fractional-octave (constant Q) resolution, the width of the averaging window will be linearly proportional to frequency [Lipschitz et al. 1985]. This is equivalent to convolving the transfer function (actually, the discrete transfer function vector) with a window function whose size increases with frequency. In the simplest case, simple averaging is applied, corresponding to a rectangular window, but weighted averaging can also be used. For example, a Hann window is a straightforward choice to weight the elements.

Smoothing is not only useful for displaying transfer functions, but also for a preprocessing step before filter design, as often applied in loudspeaker and room response equalization (see, e.g., [Ramos and López 2006; Craven and Gerzon 1992; Pedersen and Thomsen 2007]).

Smoothing in equalizer design is motivated by two reasons. First, the systems to be equalized are typically of very high order (e.g., due to the high modal density in room responses), and correction of all this detail would lead to very high order filters. As the final judge in sound quality is the human ear, it is more efficient to equalize only those aspects that lead to an audible error, and for that the smoothed response is a much better starting point. Besides efficiency and perceptual aspects, there is also a physical reason for applying logarithmic or logarithmic-like frequency resolution in equalizer design. Namely, an audio system often has multiple outputs, like multiple listening positions in a room, and the equalizer should maintain or improve the sound quality at all positions. Transfer functions measured at different points in space have more similarity at low frequencies than at high frequencies, due to the different wavelengths of sound. Therefore, an overly precise correction computed from one measurement point would actually worsen the response at other positions, especially at high frequencies [Craven and Gerzon 1992; Karjalainen et al. 2005; Cecchi et al. 2018]. This means that as frequency increases, it is more and more true that only the macro properties of the transfer function should be equalized, and equalization based on fractional octave smoothed responses does exactly this.

Figure 3.1 solid lines display a loudspeaker room response and its smoothed versions up to third-octave smoothing. Smoothing at $1/\beta$th-octave resolution was done by weighting the squared magnitudes by a Hann window having the full width of $2/\beta$-octave (i.e., its half width, or, the distance of its 0.5 points is $1/\beta$th-octave). This complies with the results of analog third-octave analyzers, where the half-power points of the bandpass filters have a third-octave distance. It can be seen that with increased smoothing the small details of the magnitude response disappear and the macro properties become more prominent. Note that the amount of smoothing required depends on the application, and also on the available filtering resources. That is, if we know that we will model or equalize the response with a low-order filter, it is advisable to smooth the response more strongly so that the filter will be forced to concentrate on the overall response, rather than modeling some sharp peaks and dips that are perceptually less relevant.

3.2 Complex smoothing

A drawback of traditional magnitude or power smoothing is that it only considers the magnitude response, thus, the phase information and the corresponding time structure of the impulse response is lost. This is acceptable for visualization, and also for magnitude-only loudspeaker or room response equalization, when usually a minimum-phase specification is computed based on the smoothed magnitude.

However, in digital processing, it is also possible to smooth the complex transfer function so that the phase information and the time structure of the impulse response can also be reconstructed [Hatziantoniou and Mourjopoulos 2000]. This has the important benefit that this way the time-domain aspects (and thus, the phase response) of the measured system can also be modeled or equalized.

Complex smoothing is very similar to power smoothing, but now the complex transfer function (and not the squared magnitude) is convolved with a real smoothing function $W(f)$. This is basically the same as smoothing the real and imaginary parts of the transfer function separately. The interesting property of such a processing is that it is equivalent to multiplying the impulse response by a time-domain window function $w(t)$ where the length of the window is

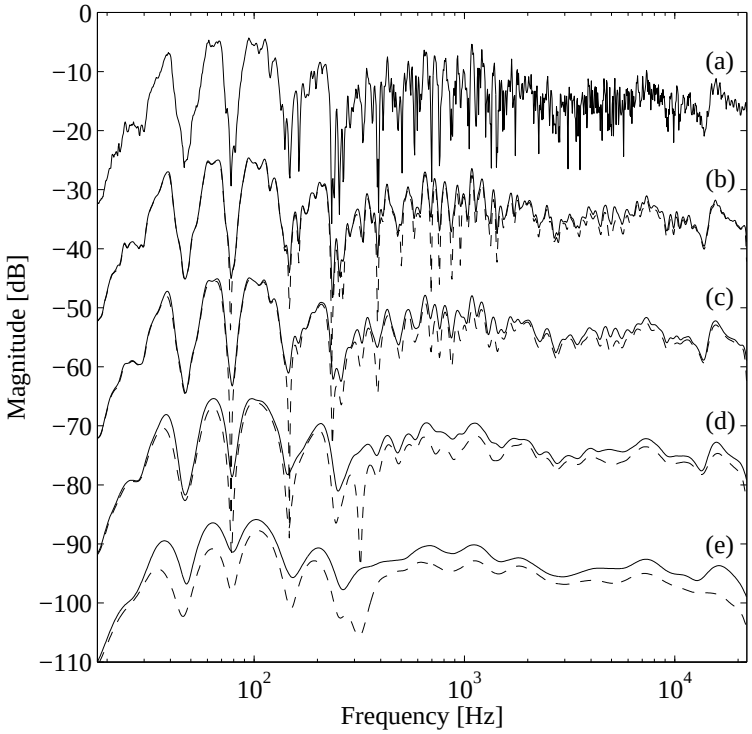


Figure 3.1: Fractional-octave smoothing: (a) original loudspeaker–room response, and the responses smoothed to (b) 24th-octave, (c) 12th-octave, (d) 6th-octave, and (e) 3rd-octave resolution. The solid lines show the power-smoothed curves, while the dashed lines display the magnitude of the complex-smoothed responses. The curves are offset for clarity.

frequency dependent [Hatziantoniou and Mourjopoulos 2000]. Note that such equivalence cannot be made for power smoothing due to the nonlinear operation of squaring the magnitude. Frequency-dependent signal windowing, the time-domain equivalent of complex smoothing, has also been proposed in [Karjalainen and Paatero 2001].

Smoothing functions $W(f)$ with a Hann window shape having the width of 50, 100, and 200 Hz are displayed in Fig. 3.2 (a). By taking the inverse Fourier transform of $W(f)$, the corresponding time-domain window functions $w(t)$ are obtained. These are displayed in Fig. 3.2 (b). It can be seen that when the width of the frequency-domain smoothing function depends on frequency, the operation corresponds to multiplying the impulse response by a window function whose length is frequency dependent. Note that in complex smoothing, the smoothing function $W(f)$ is chosen to be a real (zero phase) function [Hatziantoniou and Mourjopoulos 2000], which leads to a corresponding time window that is symmetric around $t = 0$, that is, $w(-t) = w(t)$. Since we are interested in smoothing causal responses ($h(t) = 0$ for $t < 0$), it is sufficient to multiply the impulse response $h(t)$ with the right half of the window function when the frequency-dependent windowing operation is performed.

For obtaining a logarithmic frequency resolution, wider smoothing functions have to be used at high frequencies compared to the low ones. This means that the original impulse response is windowed to shorter length at high frequencies compared to the low ones. Naturally, not only fixed fractional-octave (logarithmic), but arbitrary smoothing resolution can be applied, including those corresponding to Bark or ERB scales [Hatziantoniou and Mourjopoulos 2000].

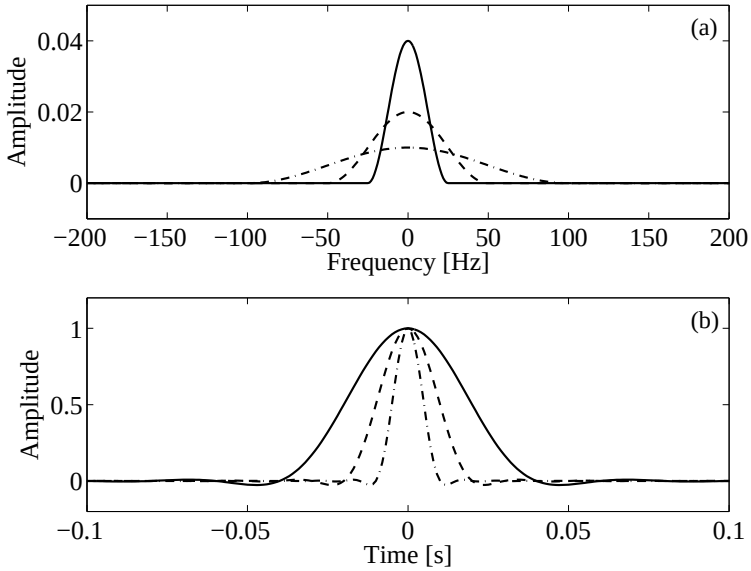


Figure 3.2: Complex smoothing and frequency-dependent windowing: (a) smoothing windows $W(f)$ with a Hann shape, having the width of 50 Hz (solid line), 100 Hz (dashed line), and 200 Hz (dash-dotted line). (b) The corresponding time-domain window functions $w(t)$ computed by the inverse Fourier transform of $W(f)$ are displayed by the same line types.

Figure 3.1 dashed-lines display the magnitudes of the complex-smoothed transfer functions. The smoothing was done by weighting the complex transfer function values by a Hann window having the full width of $2/\beta$-octave (i.e., its half width, or, the distance of its 0.5 points is $1/\beta$th-octave). The power-smoothed responses are displayed in Fig. 3.1 by solid lines. It can be seen that for mild values of smoothing, the two curves are mostly in line, while for heavier smoothing the complex-smoothed transfer functions (dashed lines) are below the power-smoothed ones (solid lines). This is because power smoothing retains all the energy of the original transfer function, since it averages the power spectrum. On the other hand, complex-smoothing is equivalent to frequency-dependent windowing of the impulse response, and heavier smoothing leads to shorter windows, thus, more energy loss. This is actually an advantage in room equalization: this way the position-dependent high-frequency reflections of the room are eliminated from the system response, thus, they are not equalized, which helps the equalizer to provide an improvement in a larger area of the room [Mourjopoulos and Hatziantoniou 2004].

3.3 Equivalent complex smoothing

Nevertheless, if we wish to avoid the "energy loss" of complex smoothing seen in Fig. 3.1 and discussed above, Hatziantoniou and Mourjopoulos [2000] proposes an alternative solution where the magnitude response is corrected so that it matches the power-smoothed response. This is called *equivalent complex smoothing*, which basically means computing both the complex-smoothed and power-smoothed responses, and combining them in such a way that the magnitude is taken from power smoothing, and the phase from complex smoothing. A similar idea is proposed by [Panzer and Ferekidis 2004] where the magnitude and phase responses are smoothed separately.

Chapter 4

Warped filters

The earliest perceptually motivated design technique is based on frequency warping. The idea can be traced back to the paper of Constantinides [1970] which proposes the substitution of unit delays in digital filters with allpass filters in order to change the filter type. Such a transformation can be lowpass–lowpass (change in cutoff frequency), lowpass–highpass, lowpass–bandpass, lowpass–band-reject, similarly to the spectral transformations used in the design of analog filters.

The first application of the allpass transform as a means of obtaining nonlinear frequency resolution was proposed by Oppenheim et al. [1971], where a non-uniform DFT was obtained by passing the input signal through an allpass chain and using the outputs of the stages as the input of an ordinary FFT operation. Strube [1980] has applied frequency warping for approximating the frequency resolution of the human auditory system in linear predictive coding.

The use of frequency warping as a means of approximating logarithmic frequency resolution for IIR filters was proposed in [Smith 1983; Waters and Sandler 1993], and the most extensive overview on the subject was presented in [Härmä et al. 2000].

4.1 The effect of warping

The basic idea of warped filters is that the unit delay z^{-1} of traditional FIR or IIR filters is replaced by an allpass filter

$$z^{-1} \leftarrow D(z) = \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}}. \tag{4.1}$$

The transformation of the frequency axis is related to the phase response of the first-order allpass. The substitution results in the frequency mapping

$$\tilde{\vartheta} = \nu(\vartheta) = \arctan \frac{(1 - \lambda^2)\sin(\vartheta)}{(1 + \lambda^2)\cos(\vartheta) - 2\lambda}, \tag{4.2}$$

where ϑ is the original and $\tilde{\vartheta}$ is the warped angular frequency in radians [Härmä et al. 2000]. This transformation is displayed for various λ values in Fig. 4.1.

Accordingly, a filter originally having the transfer function of $H(\vartheta)$ will have the transfer function of $H(\nu(\vartheta))$ after substituting its delay elements by the first order allpass of Eq. (4.1). For increasing the resolution at low frequencies, which is required for achieving a logarithmic scale, positive λ values are used. It can be seen in Fig. 4.1 that in this case the region around

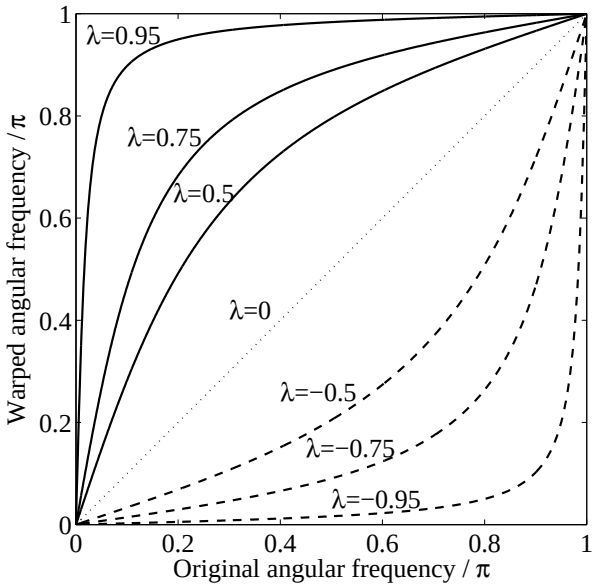


Figure 4.1: The frequency mapping function $\nu(\vartheta)$ of Eq. (4.2) for various warping parameters.

zero frequency will span a much larger frequency region in the warped domain, which means increased resolution.

The change of the frequency resolution is related to the slope of the mapping function in Fig. 4.1, that is, the derivative of Eq. (4.2). The higher the slope, the larger is the increase in frequency resolution. Accordingly, if a FIR or IIR filter has a local resolution $\Delta f(f)$, then its warped variant will have the resolution $\Delta\tilde{f}(f)$

$$\Delta\tilde{f}(f) = \frac{1+\lambda^2-2\lambda\cos(\vartheta)}{1-\lambda^2}\Delta f, \tag{4.3}$$

where $\vartheta = 2\pi f/f_\mathrm{s}$, with f_s being the sampling frequency [Ramos et al. 2009]. For FIR filters, $\Delta f = f_\mathrm{s}/N$, where N is the filter order. For IIR filters such an exact number cannot be computed, nevertheless, Eq. (4.3) still shows how the resolution is mapped when the IIR filter is implemented using allpasses instead of unit delays.

The relative resolution is shown in Fig. 4.2 (a) for various λ values, and the dotted line $\lambda = 0$ corresponds to no warping. The plot was computed using Eq. (4.3) with $\Delta f = 1$, and the sampling frequency was $f_\mathrm{s} = 44.1$ kHz. It can be seen in Fig. 4.2 (a) that the warped implementations with $\lambda > 0$ increase the resolution (decrease $\Delta\tilde{f}$) at low frequencies, at the expense of lower resolution (larger $\Delta\tilde{f}$) at high frequencies. This tradeoff is understandable since the degrees of freedom in the filter are unchanged, thus, an improvement at a specific frequency band will lead to poorer performance at another band.

In audio applications we usually try to approximate logarithmic frequency resolution, therefore it makes sense to plot the same curves divided by frequency, given by

$$\frac{\Delta\tilde{f}(f)}{f} = \frac{1+\lambda^2-2\lambda\cos(\vartheta)}{(1-\lambda^2)}\frac{\Delta f}{f}. \tag{4.4}$$

In Fig. 4.2 (b) $\Delta\tilde{f}(f)/f$ is plotted for $\Delta f = 1$. Again, the dotted line with $\lambda = 0$ shows what happens with an ordinary (not warped) FIR or IIR filter, and a logarithmic frequency resolution

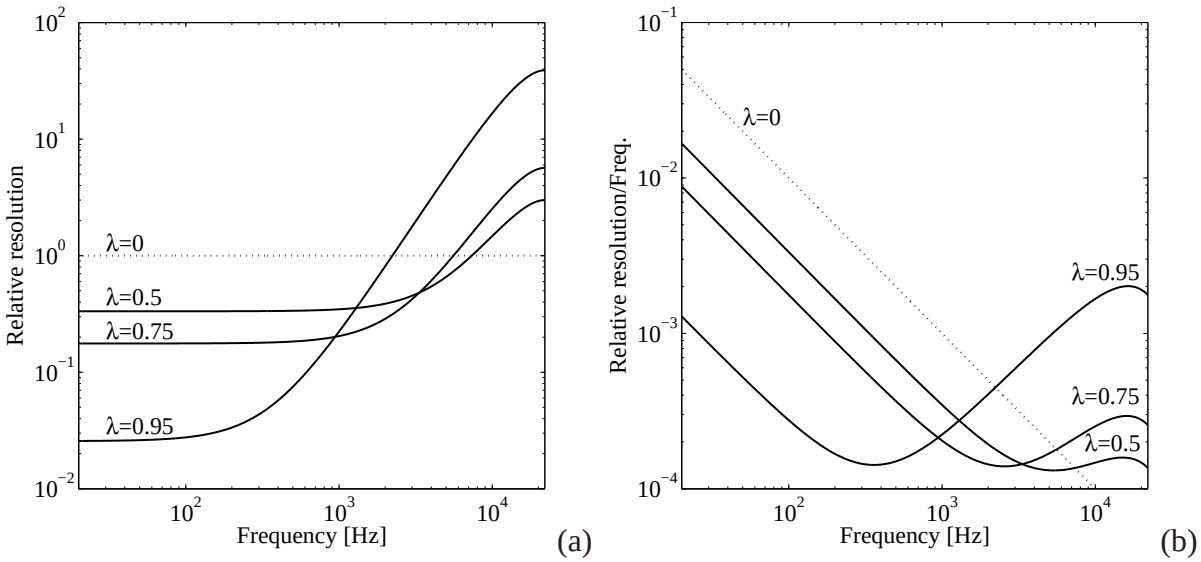


Figure 4.2: The change of frequency resolution due to warping: (a) $\Delta\tilde{f}$ according to Eq. (4.3) for various warping parameters, and (b) "logarithmic resolution" $\Delta\tilde{f}/f$ computed by Eq. (4.4) The resolution prior to warping is $\Delta f = 1$ in both figures. Smaller values correspond to a higher resolution.

would correspond to a horizontal line in the figure. It can be seen that while none of the λ values achieve fully logarithmic resolution (none of the curves are horizontal lines), there are some frequency bands for each λ where this is relatively well achieved. Moreover, the warped curves (solid lines) span a significantly smaller vertical range compared to no warping (dotted line), meaning that we are closer to the desired distribution of frequency resolution.

The relation of warping to psychoacoustic scales (ERB, Bark, Greenwood) is discussed in [Härmä et al. 2000]. In addition, Smith and Abel [1999] has given an analytical expression for λ as a function of sampling rate to match the Bark scale. However, similarly to the logarithmic scale, an exact match is not possible due to the limited degrees of freedom (a single warping parameter).

The warping effect is demonstrated in Fig. 4.3, where (a) displays the original frequency response of an arbitrary FIR filter having random coefficients, while (b) and (c) display the filter response when the unit delays are exchanged for first-order allpass filters with $\lambda = 0.5$ and $\lambda = 0.75$, respectively. It can be seen in (a) that the original FIR filter has even (linear) resolution, that is, the detail is evenly distributed in the linear frequency scale. However, when frequency warping is applied, the transfer function gradually shifts towards lower frequencies with increasing λ, meaning that the level of detail is higher at low frequencies compared to high frequencies.

4.2 Filter design

The basic idea of warped filter design is that the filter specification in the time- or frequency-domain is predistorted with the inverse of the warping effect of the filter. Then any traditional filter design technique can be used to design an FIR or IIR filter, and finally when the filter is implemented by using first-order allpass elements, the filter response gets to the right place. A very appealing property of warped filters is that the embedded filter design step is the same as

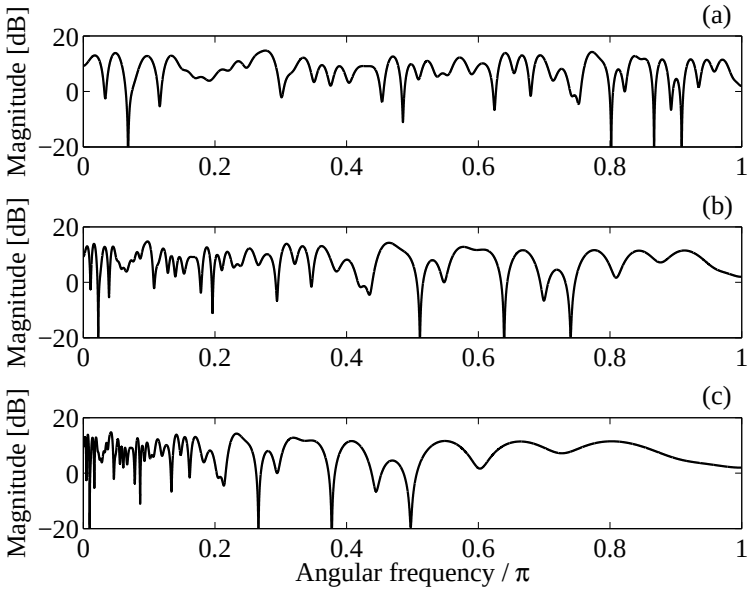


Figure 4.3: The frequency response of an 100th order FIR filter with random coefficients (a), and the same filter having its unit delays replaced by allpass filters of the form Eq. (4.1) with (b) $\lambda = 0.5$ and (c) $\lambda = 0.75$.

for ordinary FIR or IIR filters.

4.2.1 Frequency-domain design

Let us define the inverse mapping $\nu^{-1}(\tilde{\vartheta})$ so that $\vartheta = \nu^{-1}(\nu(\vartheta))$. If the mapping function was computed by using λ in Eq. (4.2), then the inverse mapping can be obtained by using $-\lambda$ in the same function Eq. (4.2) [Härmä et al. 2000].

The steps of filter design are the following:

1. Prewarping of the target frequency response. The filter specification is transformed by the inverse mapping function $\nu^{-1}(\tilde{\vartheta})$. Mathematically, this mapping is described by

$$\tilde{H}_{\mathrm{t}}(\tilde{\vartheta}_n) = H_{\mathrm{t}}(\nu^{-1}(\tilde{\vartheta}_n)). \tag{4.5}$$

In practice this can be done by some suitable interpolation, where we have direct control over the density of the target frequency points in the warped domain. A simpler alternative solution is to move the original specification points $H_{\mathrm{t}}(\vartheta_n)$ to the frequencies $\tilde{\vartheta}_n = \nu(\vartheta_n)$, but leaving their magnitude and phase values unchanged.

2. Frequency-domain filter design. An FIR or IIR filter is designed based on the pre-warped target $\tilde{H}_{\mathrm{t}}(\tilde{\vartheta})$ by any of the available filter design methods, just as with ordinary FIR or IIR filters. This leads to the filter $\tilde{H}(\tilde{\vartheta})$.

3. Filter implementation. This is actually not part of the design process, but when the warped filter is implemented by substituting the unit delays of the FIR or IIR filter designed in Step 2 by the first-order allpass filters, the frequency-scale of the filter will be mapped

back to match that of the original specification. Thus, the transfer function is automatically shifted from $\tilde{H}(\tilde{\vartheta})$ to $\tilde{H}(\vartheta)$ so that

$$H(\vartheta) = \tilde{H}(\nu(\vartheta)). \tag{4.6}$$

4.2.2 Time-domain design

The steps of the design are the following:

1. **Prewarping of the target impulse response.** In the time domain, the design of warped filters starts with warping the target impulse response $h_{\mathrm{t}}(n)$ by the use of an allpass chain with $-\lambda$ [Härmä et al. 2000].

2. **Time-domain filter design.** Warped FIR (WFIR) filters can be simply obtained by truncating or windowing the warped target response $\tilde{h}_{\mathrm{t}}(n)$ just as what would be done when modeling an infinite impulse response with a finite length filter. Similarly, warped IIR filters are designed by traditional filter design algorithms (e.g., LPC, Prony, Steiglitz-McBride) using this warped $\tilde{h}_{\mathrm{t}}(n)$.

3. **Filter implementation.** When the filter designed in step 2 is implemented with first-order allpasses in place of the unit delays, its impulse response is automatically mapped back to be in accordance with the original (unwarped) target impulse response.

4.3 Filter design examples

Figure 4.4 shows various warped filter designs based on the same loudspeaker–room response where the standard IIR filter design methods failed to provide logarithmic frequency resolution in Chap. 1. The first three curves (a)–(c) correspond to warped FIR filters with various λ values designed by truncating the warped impulse response. It can be seen that increasing λ shifts the region with detailed modeling in accordance with the resolution curves of Fig. 4.2 (b). It can also be seen that none of the λ values provide even distribution of the modeling accuracy in the logarithmic scale.

Figure 4.4 (d)–(f) displays warped IIR filters designs using the Steiglitz-McBride method [Steiglitz and McBride 1965] with the same λ values as for the WFIR filters. Here again increasing λ shifts the region of accurate modeling to low frequencies. Compared to the WFIR examples, the WIIR filters provide a better fit since they can redistribute the modeling detail by their poles. Coming from this, the region of accurate modeling is wider than for WFIR filters. However, there is still no such λ value where the accuracy is evenly distributed in the full audio bandwidth.

4.4 Implementation

4.4.1 Implementation with special filter structures

The WFIR filters have a similar structure as FIR filters, but the unit delays are replaced by the allpass filter $D(z)$. That is, the WFIR filter is an allpass chain, where the signals between the first-order allpass blocks are tapped and weighted by the FIR coefficients b_k. On the contrary

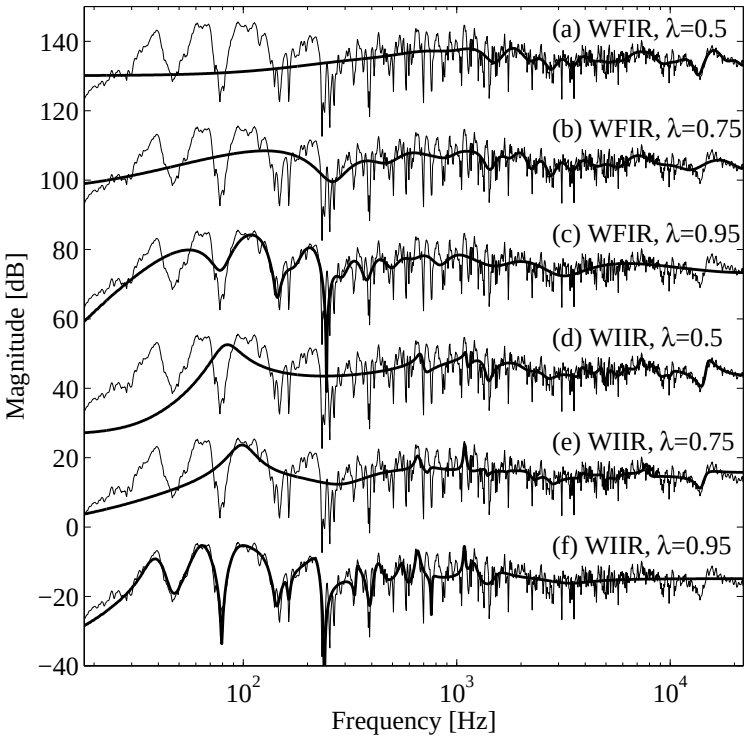


Figure 4.4: Modeling a minimum-phase loudspeaker–room response with (a)–(c) 32nd order warped FIR filters and (d)–(f) 32nd order warped IIR filters using various warping parameters $\lambda = 0.5, 0.8, 0.95$. The target response is displayed by thin lines, while the filter responses are shown by thick lines. The curves are offset for clarity.

to WFIR filters, the implementation of WIIR filters is less straightforward. This is because for WIIR filters the replacement of unit delays by $D(z)$ leads to delay-free loops, and the filter structure has to be modified for practical implementation [Johnson 1979; Steiglitz 1980; Ahuja and Roy 1980; Karjalainen et al. 1997]. Because of the specialized filter structures, WFIR and WIIR filters typically require 2–4 times higher computational time compared to normal FIR and IIR filters of the same order [Härmä et al. 2000].

4.4.2 Dewarping to direct-form filters

Instead of implementing warped filters directly, it is also possible to simply substitute the first-order allpass transfer function Eq. (4.1) into the transfer function of the FIR or IIR filter, which will result in an IIR filter with the same order for the numerator and the denominator [Johnson 1979; Smith 1983; Härmä et al. 2000]. The disadvantage of dewarping the filter to a direct-form realization is that it can be done only up to filter orders around 20 due to numerical problems coming from pole clustering at low frequencies [Härmä et al. 2000].

4.4.3 Dewarping to cascade or parallel sections

Another, numerically better behaving option is to dewarp the filter to a cascade or parallel second-order structure. The idea is first break up the transfer function of the warped filter to series or parallel second-order sections, and then dewarp the sections separately [Ahuja and Roy

1980]. Finally, the filter is implemented in this series or parallel form.

While [Ahuja and Roy 1980] presents the formulas of dewarping a second-order section, in [Tyril et al. 2001] the problem is addressed by first finding the poles $\tilde{p}_k$ and zeros $\tilde{m}_k$ of the warped IIR filter, then dewarping them by the expression

$$p_k = \frac{\tilde{p}_k + \lambda}{1 + \lambda\tilde{p}_k}, \quad m_k = \frac{\tilde{m}_k + \lambda}{1 + \lambda\tilde{m}_k}. \tag{4.7}$$

Finally, the filter is implemented as a series of second-order sections, computed from the dewarped (linear frequency-scale) poles p_k and zeros m_k [Tyril et al. 2001].

Dewarping to second-order sections is a very effective way of implementing warped filters since it can be used with high filter orders (while dewarping to direct form cannot). For the series second-order case, the number of multiplications remains the same as for traditional IIR filters of the same order, requiring 4 multiplications per second-order section. This leads to 4 multiplications per second-order section. On the contrary, when the warped filter is dewarped to parallel second-order sections, there is an increase in the number of multiplications. This is because the decomposition of the warped filter to parallel second-order sections leads to sections whose numerator order is 1, and denominator order is 2 (4 multiplications per section). When this section is dewarped, the numerator order also becomes 2 (a zero is placed at λ), requiring 5 multiplications per section, as opposed to 4 for the series case.

4.5 Extensions of basic warping techniques

We have seen in Fig. 4.2 (b) that there is no single λ value which would result in a constant resolution in the logarithmic scale. On the contrary, each λ focuses the resolution around a certain frequency region. Therefore, a straightforward development of the warped filter concept is to use different warping parameters for the different frequency regions.

4.5.1 Combination with linear filters

A special case of multi-band warping is when one of the bands is a normal FIR or IIR filter (that is, for that band $\lambda = 0$). This is motivated by the fact that straightforward FIR and IIR filters can be implemented more efficiently compared to their warped counterparts. Since the resolution of normal FIR and IIR filters is linear, they are best suited for modeling or equalization of the high-frequency region of the transfer function. Such a combined warped and linear equalizer was presented in [Wang et al. 2000a]. The frequency band is split to two by a crossover network, which includes a lowpass and a highpass filter. The high-frequency part of the signal is processed by an FIR filter, while the low-frequency part by a warped FIR filter. A similar approach is presented in [Ramos et al. 2009] with the important difference that the FIR and warped FIR filters are in cascade, eliminating the need for a crossover network.

4.5.2 Multiple warped filters

The first paper using multiple warped filters is [Wang et al. 2000b], which proposes the use of a three-band equalizer, where different λ values are chosen in the three branches to maximize the warping effect for each warped FIR filter. In addition, the middle band incorporates decimation and interpolation so that the processing is done at a reduced sample rate to maximize efficiency.

The three bands are separated by a crossover network composed of a low-pass, band-pass and a high-pass filter.

I have developed a multi-band warping technique for computing the pole positions of the fixed-pole parallel filter [Bank and Ramos 2011], and the first part of the method can be directly used as a warped IIR filter design. In this method the transfer function is split to two bands, and two warped IIR filters are designed with such λ values that lead to maximal modeling resolution in the center of their corresponding bands. Then the two WIIR filters are combined (connected in series). The details of this procedure will be outlined in Sec. 7.3.3.

4.5.3 Custom warping

So far we have been using the frequency warping function Eq. (4.2), which gives a limited freedom due to a single parameter λ. Improved results were obtained by combining multiple warped filters with different λ values. However, a question arises if it is possible to use a logarithmic frequency mapping prior to filter design, since that would result in an even distribution of approximation errors in the logarithmic scale, and thus, to a truly logarithmic frequency resolution. One solution to the problem is to design a parallel set of all-pass filters of increasing order, based on the phase response of the logarithmic mapping [Härmä and Paatero 2001]. However, a disadvantage of the method is its very heavy computational load making it impractical for real-time applications. I have developed a more efficient approach that does not increase the filter order. The basic idea of [Bank 2011c] is that the warped filter designed based on the logarithmically mapped specification is implemented by pole-zero dewarping, and not by the usual allpass substitution. Since the filter is implemented in the parallel second-order form, it can also be considered as a pole positioning technique for fixed-pole parallel filters. Accordingly, the details will be presented in Sec. 7.3.4.

4.6 Final remarks on the various warping techniques

Warped FIR filters represent the simplest method for obtaining filters with logarithmic-like frequency resolution. They can be designed in the time-domain by truncating or windowing the pre-warped target impulse response, similarly how one would model an infinite impulse response system by a finite impulse response filter. The implementation of warped FIR filters is also straightforward: the unit delays of the FIR filter are replaced by first-order allpass filters.

Warped IIR filters result in more accurate models for the same filter order compared to WFIR filters. However, the implementation of WIIR filters is more complicated: to avoid the problem of delay-free loops, a modified filter structure has to be used with recomputed filter coefficients. Alternatively, the WIIR filter can be dewarped to series or parallel second-order sections, leading to a more efficient implementation with a simpler filter structure requiring less additions and multiplications.

Normal WIIR filters already provide a reasonable modeling if not the entire audio bandwidth is used (for example, imagine modeling or equalizing a small computer speaker having the bandwidth of [200 Hz, 15 kHz]). If the full audio range has to be modeled or equalized, then multi-band or custom warping techniques can be used to improve accuracy. However, as we shall see, there are even more flexible methodologies for obtaining filters with arbitrary frequency resolution: these are the Kautz and parallel filters discussed in the next chapters.

Chapter 5

Kautz filters

We have seen in Chap. 4 that warped filters (especially WIIR filters) provide a much better approximation to logarithmic frequency resolution compared to straightforward IIR filters shown in Chap. 1. However, when the full audio band from 20 Hz to 20 kHz has to be modeled or equalized, a single λ parameter is insufficient, as either the high-, or the low frequencies will lack modeling detail (see Fig. 4.4). This can be improved using multiple λ values, as it was outlined in Sec. 4.5.

As a generalization, the question arises if it would be possible to construct the WFIR and WIIR filters in such a way that all the allpasses have different λ values. This question has been investigated by Tyril et al. [2001], who has proposed the use of WizFIR (warped individual z FIR) filters. No systematic procedure has been given for choosing the different λ parameters for the various sections, rather, they were set by trial and error. The authors have found that the performance is slightly improved compared to normal WFIR filters. However, this comes at a price of more complicated parameter estimation, since now the procedure of prewarping and the use of traditional filter design techniques as shown in Sec. 4.2 is unfeasible. This is because no common mapping function exists. Instead, the parameters of the WizFIR filter have been obtained with a least squares fit where the basis functions are the impulse responses computed at the various stages of the allpass chain. Tyril et al. [2001] have concluded that for the same computational complexity, WizFIR filters are outperformed by WIIR filters. Thus, the use of WizFIR filters is not encouraged. The authors have also added that the use of individual λ-s in WIIR filters is doubtful since WIIR filters can have different poles anyway.

A mathematically better founded alternative to WizFIR filters is the use of Kautz filters, which are indeed very similar to a warped FIR filter structure with different λ values, with an additional feature of orthonormal basis functions [Paatero and Karjalainen 2003]. Before seeing how Kautz filters can be used to achieve logarithmic frequency resolution, we first review the history of the method.

5.1 Laguerre and Kautz models

Traditionally, Laguerre and Kautz models were proposed for system identification. These models reconstruct the system response as a linear combination of orthonormal basis functions. In those times orthonormal functions were essential since that way the parameters of the models are obtained by a scalar product, which could be computed even on the earliest computers.

In the case of Laguerre models [Oliveira e Silva 1995], the orthonormalization procedure is

started from identical first-order lowpass transfer functions $1/(1-pz^{-1})$ with a pole at p. This gives the following set of orthonormal functions:

$$L_k(z) = \frac{\sqrt{1-p^2}}{1-pz^{-1}} \left(\frac{z^{-1}-p}{1-pz^{-1}} \right)^{k-1}, \tag{5.1}$$

for $k = 1, 2, \ldots K$.

The term in the parenthesis corresponds to an allpass filter: indeed, Laguerre models lead to the same filter structure as warped FIR filters, and the only difference is the normalization term $\sqrt{1-p^2}/(1-pz^{-1})$, which is simply a lowpass filter at the input of the allpass backbone.

A straightforward generalization of Laguerre filters is when the orthonormalization process is started from first-order lowpasses having different poles (note that the poles p_k can also be complex). For continuous-time systems the concept was introduced by Kautz [1954], while the corresponding discrete-time orthonormal sequences were first presented by Broome [1965]. The orthonormal polynomials take the following form [Paatero and Karjalainen 2003]:

$$G_k(z) = \frac{\sqrt{1-p_k\overline{p}_k}}{1-p_kz^{-1}} \prod_{j=1}^{k-1} \frac{z^{-1}-\overline{p}_j}{1-p_jz^{-1}}, \tag{5.2}$$

for $k = 1, \ldots K$, where $\overline{p}_k$ is the complex conjugate of p_k. (Note that in [Paatero and Karjalainen 2003] the indexing starts from $k = 0$.) Again, the filter can be implemented as a tapped allpass backbone, but now the poles of the filter are different. Therefore, the first-order lowpass normalization terms $\sqrt{1-p_k\overline{p}_k}/(1-p_kz^{-1})$ have to be implemented separately after the tapping points of the backbone. Note that this is similar to the WizFIR filter of [Tyril et al. 2001] with the added first-order lowpasses at the tap outputs.

Equation (5.2) results in complex sequences (impulse responses) for complex poles. Also, such a model would result in a filter with complex coefficients. However, usually we are interested in modeling real impulse responses, and the use of filters with real coefficients. For such systems complex poles always appear in complex conjugate pairs p_i and $\overline{p}_i$ and the complex pole pairs can be combined to form second-order sections. For a pole pair p_i and $\overline{p}_i$ we obtain a pair of real valued basis functions $G_i^+(z)$ and $G_i^-(z)$ as follows [Broome 1965; Paatero and Karjalainen 2003]:

$$A_i(z) = \frac{1}{(1-p_1z^{-1})(1-\overline{p}_1z^{-1})} \prod_{j=2}^{i} \frac{(z^{-1}-p_{j-1})(z^{-1}-\overline{p}_{j-1})}{(1-p_jz^{-1})(1-\overline{p}_jz^{-1})} \tag{5.3a}$$

$$G_i^+(z) = C_i^+(1+z^{-1})A_i(z) \tag{5.3b}$$

$$G_i^-(z) = -C_i^-(1-z^{-1})A_i(z), \tag{5.3c}$$

for $i = 1, 2, \ldots I$. In Eq. (5.3) C_i^+ and C_i^- are normalization constants computed from the pole set p_i [Broome 1965; Paatero and Karjalainen 2003]. Note that since

$$A_i(z) = A_{i-1}(z)\frac{(z^{-1}-p_{i-1})(z^{-1}-\overline{p}_{i-1})}{(1-p_iz^{-1})(1-\overline{p}_iz^{-1})}, \tag{5.4}$$

thus, each $A_i(z)$ can be implemented by filtering the previous term $A_{i-1}(z)$ with a second-order filter, meaning that the $A_i(z)$ part can be implemented as a backbone composed of second-order stages. Then the signal is tapped between the sections and filtered by the first-order

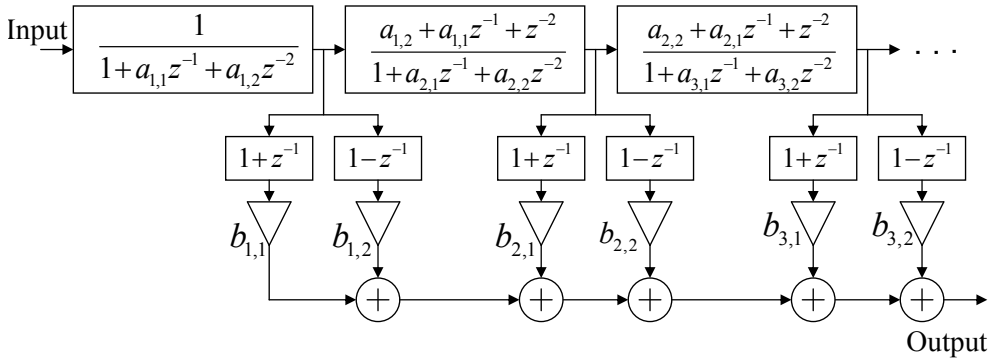


Figure 5.1: The structure of the Kautz filter with real coefficients.

numerator terms $C_i^+(1 + z^{-1})$ and $-C_i^-(1 - z^{-1})$ to obtain the outputs $G_i^+(z)$ and $G_i^-(z)$ [Broome 1965]. The total filter response is the linear combination of these outputs with weights w_k. The block diagram of the Kautz structure in its efficient real form is displayed in Fig. 5.1, where $a_{i,1} = -2\mathrm{Re}\{p_i\}$, $a_{i,2} = |p_i|^2$, $b_{i,1} = w_{2i-1}C_i^+$, and $b_{i,2} = -w_{2i}C_i^-$.

Both for Laguerre and Kautz models the impulse response of a system $h_\mathrm{t}(n)$ is modeled as a linear combination of basis functions $x_k(n)$, which are obtained as the inverse–z transform of $L_k(z)$ for Laguerre and $G_i^+(z)$ and $G_i^-(z)$ for Kautz filters:

$$h(n) = \sum_{k=1}^{K} w_k x_k(n), \tag{5.5}$$

where w_k are the weights. The goal is to estimate w_k such that the model response $h(n)$ is closest to the target $h_\mathrm{t}(n)$. Since the $x_k(n)$ sequences are orthonormal, the optimal solution in the mean-squared sense is given by the scalar product

$$w_k = \sum_{n=0}^{N} h_\mathrm{t}(n)x_k(n), \tag{5.6}$$

requiring much less computations compared to solving the usual LS equations required for non-orthogonal basis functions. This complexity can be decreased even more by noting that the scalar product of Eq. (5.6) is equivalent to convolving the time-reversed target $h_\mathrm{t}(-n)$ with $x_k(n)$ and taking the output for $n = 0$ [Broome 1965; Paatero and Karjalainen 2003]. Convolution with $x_k(n)$ is actually done by filtering $h_\mathrm{t}(-n)$ with the same recursive Laguerre or Kautz filter structure which is used for modeling, and w_k are simply obtained by reading the outputs at time $n = 0$. This property leads to very low computational complexity for weight estimation.

Nowadays solving a linear least squares problem is considered as one of the simplest optimization problems, thus, the orthonormality of Laguerre and Kautz basis functions has lost some of its attractiveness. However, for some cases such as adaptive filtering orthonormality is still highly beneficial since it leads to faster convergence [Salama and Cousseau 1998].

5.2 Kautz filters for audio applications

The use of Kautz filters as a means of controlling the frequency resolution of filter design was proposed by Paatero and Karjalainen [2003]: they note the equivalence of warped FIR filters and

Laguerre models, and show that for Kautz filters with individual poles the frequency resolution can be directly controlled by the pole values.

For obtaining a logarithmic resolution, the poles should be set according to the following formulas [Paatero and Karjalainen 2003; Karjalainen and Paatero 2007]:

$$\vartheta_k = \frac{2\pi f_k}{f_s} \tag{5.7}$$

$$p_k = R^{\vartheta_k/\pi} e^{\pm j\vartheta_k}, \tag{5.8}$$

where ϑ_k are the pole frequencies in radians determined by the logarithmic frequency series f_k and the sampling frequency f_s. The pole magnitudes form an exponentially damping sequence approximating a constant Q resolution. The pole magnitude at the Nyquist rate is set by the damping parameter R. As a rule of thumb, R should be set so that the magnitude responses of the neighboring taps cross at their -3 dB point, but it is said that the precise value of the damping parameter is not critical [Karjalainen and Paatero 2007].

Another way of finding the poles suggested in [Paatero and Karjalainen 2003] is by the Brandenstein-Unbehauen method [Brandenstein and Unbehauen 1998], which is an iterative least-squares optimization technique similar to the Steiglitz-McBride algorithm [Steiglitz and McBride 1965]. It is suggested that for audio applications, a warped version of the Brandenstein-Unbehauen technique should be used, where the algorithm is run on the warped impulse response, and then the poles are dewarped by Eq. (4.7) [Paatero and Karjalainen 2003]. The practical performance is similar to warped filters, with the added flexibility of complete control over the frequency resolution of the design.

Recently a new pole positioning technique have been presented in [Vairetti et al. 2015, 2016, 2017] for the application of sparse approximation of highly reverberant room impulse responses. Instead of aiming at modeling all the details of the room impulse, the proposed technique tries to fit the most prominent resonances by iteratively testing a set of candidate pole positions and adding the one to the final pole set that decreases the approximation error the most. The examples demonstrate that the method provides a stable approximation of room impulse responses with high filter orders ($N > 1000$). However, the pole frequencies and damping factors cannot be estimated as accurately as with the Brandenstein-Unbehauen technique since they are chosen from a predefined pole set [Vairetti et al. 2017]. Therefore, for general filter or equalizer design tasks where we are not aiming at a sparse approximation of very high order systems, the Brandenstein-Unbehauen method [Brandenstein and Unbehauen 1998; Paatero and Karjalainen 2003] can be still considered as the state of the art for finding the poles of the Kautz filter.

The mathematical equivalence of Kautz and parallel filters will be shown in Sec. 6.4, meaning that Kautz and parallel filters provide the same approximation for a given pole set. Therefore, no design examples are presented here and the reader is referred to the parallel filter examples in Chap. 6. Note that coming from this equivalence, all the pole positioning methods I have developed for parallel filters (see Chap. 7) can be used for Kautz filters as well, with improved results compared to the above Brandenstein-Unbehauen method.

We may conclude that Kautz filters provide an attractive way for constructing logarithmic frequency resolution filters. However, the combined cascade-parallel nature of the Kautz filter visible in Fig. 5.1 requires more computation compared to filters implemented in direct or cascade form. For a practical DSP implementation every second-order section of the Kautz filter requires 6 multiply-and-accumulate (MAC) operations and 2 additions (ADD), while a direct-from or cascade IIR filter needs only 4 MAC instructions per second-order section.

Chapter 6

Fixed-pole parallel filters

It has been discussed in Chap. 5 that Kautz filters provide more flexibility in the distribution of frequency resolution compared to warped filters, because the resolution is controlled by the entire pole set and not only by the single parameter λ. However, a drawback of Kautz filters that they require a complicated series-parallel filter structure leading to higher computational complexity compared to normal IIR filters of the same order. When working in the Acoustics Laboratory in the Helsinki University of Technology in 2007, the question came to my mind whether it is possible to find such a filter structure that retains the modeling flexibility of Kautz filters, while enables more efficient implementation. Since the Kautz basis functions were generated by orthonomalizing a set of damped exponentials, it seemed logical to test whether the control of frequency resolution remains the same even with this non-orthogonal set of basis functions. The answer was yes, and the idea of fixed-pole parallel filters was born where the poles of the sections are predetermined and the numerator coefficients are estimated by a least-squares procedure [Bank 2007], as will be shown in Sec. 6.2. Later I have also developed the mathematical equivalence of the Kautz and parallel designs in terms of the resulting transfer function [Bank 2013a], which will be outlined in Sec. 6.4. Similarly to Kautz filters, for complex conjugate pole pairs the complex basis functions can be combined, thus the filter structure reduces to a parallel set of second-order filters. The advantage compared to Kautz filters is that the required number of multiplications is reduced from 6 to 4 per section, while the number of additions is reduced from 8 to 4. The simpler filter structure predicts an even larger increase in efficiency in practical implementations compared to what comes from the number of arithmetic operations, especially in the view of parallel computing architectures gaining more and more popularity.

The idea of implementing IIR filters in the form of parallel second-order sections is of course not new, and have been used traditionally because its better quantization noise performance compared to direct-form filters, similarly to series biquads [Rabiner and Gold 1975; Oppenheim et al. 1999; Chen 1996]. The parameters of the second-order sections are determined from the direct form IIR filters, by, e.g., the partial fraction expansion or a similar algorithm [Rabiner and Gold 1975; Oppenheim et al. 1999; Price et al. 1996].

The novelty of the methodology I have developed lies in the fact that instead of converting from a direct-form IIR filter, the parallel second-order filter-bank is designed directly, and that by the suitable choice of the pole frequencies, we gain direct control over the frequency resolution of the design.

Similarly to Kautz filters, the poles are set according to the desired frequency resolution. This leads to a linear-in-parameter model for the zeros of the second-order sections. Since

we have given up the orthonormality of the basis functions, the scalar product of Eq. (5.6) cannot be used for parameter estimation. This added computational complexity for filter design compared to Kautz filters is not anymore a problem with current computers. Actually, linear least squares problems are considered as the simplest optimization problems since they have a closed-form solution, as we have seen in Sec. 2.3. Thus, the increased design complexity is far outweighed by the simpler filter structure leading to significant computational savings during real-time filtering.

6.1 Filter structure

Every transfer function of the form $H(z) = B(z)/A(z)$ can be rewritten in the form of partial fractions:

$$H(z) = \sum_{i=1}^{P} c_i \frac{1}{1 - p_i z^{-1}} + \sum_{m=0}^{M} f_m z^{-m}, \tag{6.1}$$

where p_i are the poles, either real valued or forming conjugate pairs, if the system has a real impulse response. The second sum in Eq. (6.1) is the FIR filter part of order M. If the orders of $A(z)$ and $B(z)$ are the same, the FIR part reduces to a constant coefficient f_0, while for transfer functions having more poles than zeros, the FIR part vanishes completely. Note that in the case of pole multiplicity, terms of higher order also appear in Eq. (6.1) [Smith 2007].

Now let us assume that we are trying to fit the IIR filter $H(z)$ to a target response $H_\mathrm{t}(z)$ so that the poles of the filter are predefined. In this case Eq. (6.1) becomes linear in its free parameters c_i and f_m, thus, they can be estimated by a linear least squares fit to match the required response, as proposed in [Bank 2007].

The resulting filter can be implemented directly as in Eq. (6.1), forming parallel first-order complex filters. However, it is more practical to combine the complex pole pairs to a common denominator resulting in a parallel set of second-order sections with real valued coefficients. Those fractions of Eq. (6.1) that have real poles can be combined with other real poles to form second-order IIR filters, yielding a canonical structure. Thus, the transfer function becomes

$$H(z) = \sum_{k=1}^{K} \frac{b_{k,0} + b_{k,1} z^{-1}}{1 + a_{k,1} z^{-1} + a_{k,2} z^{-2}} + \sum_{m=0}^{M} f_m z^{-m}, \tag{6.2}$$

where K is the number of second-order sections. The filter structure is depicted in Fig. 6.1. In the case of multiple poles, higher order sections would also be required. However, as the pole set is given by the designer, pole multiplicity can be avoided, and therefore will not be discussed further.

For most modeling or equalization tasks, there is no need for the FIR part. On the other hand, for non-decaying responses where the peak of the target response is not in the beginning, using the FIR path for the early, rising part of the response improves modeling accuracy for a given computational complexity [Bank 2007]. (See also the discussion in Sec. 2.4).

For determining the parameters, one option is to estimate c_i and f_m in Eq. (6.1) and then combine the complex conjugate pairs, as shown in [Bank 2007]. However, since the second-order form is still linear in its free parameters, $b_{k,0}$, $b_{k,1}$, and f_m, the filter can be designed in this form directly [Bank 2008]. This path will be taken in the following.

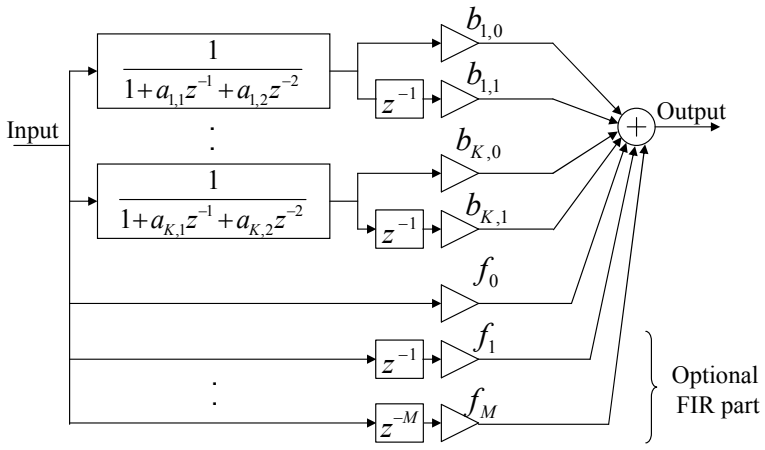


Figure 6.1: Structure of the parallel second-order filter.

6.2 Filter design

Again, let us assume that the pole set of the filter is known (various methods for choosing the poles will be outlined later in Chap. 7). In this case the parallel filter weights (numerator parts) $b_{k,0}$, $b_{k,1}$ and the FIR coefficients f_m can be obtained either from a target impulse response $h_\mathrm{t}(n)$ [Bank 2008] or from a target frequency response $H_\mathrm{t}(\vartheta_n)$ [Bank 2011b]. Both of these will be presented here, together with a discussion of their differences.

6.2.1 Time-domain design

The impulse response of the parallel filter is given by

$$h(n) = \sum_{k=1}^{K} b_{k,0}u_k(n) + b_{k,1}u_k(n-1) + \sum_{m=0}^{M} f_m\delta(n-m), \tag{6.3}$$

where $u_k(n)$ is the inverse z-transform of the transfer function $1/(1+a_{k,1}z^{-1}+a_{k,2}z^{-2})$, which is an exponentially decaying sinusoidal function, and $\delta(n)$ is the discrete-time unit impulse.

Naturally, Eq. (6.3) is linear in parameters, similarly to its z-transform counterpart Eq. (6.2). Writing Eq. (6.3) in matrix form yields

$$\mathbf{h} = \mathbf{M}\mathbf{p}, \tag{6.4}$$

where $\mathbf{p} = [b_{1,0}, b_{1,1}, \ldots b_{K,0}, b_{K,1},\ f_0 \ldots f_M]^T$ is a column vector composed of the free parameters. The columns of the modeling signal matrix $\mathbf{M}$ contain the modeling signals, which are $u_k(n)$ and their delayed counterparts $u_k(n-1)$, and for the FIR part, the unit impulse $\delta(n)$ and its delayed versions up to $\delta(n-M)$. Finally, $\mathbf{h} = [h(0) \ldots h(N)]^T$ is a column vector composed of the resulting impulse response. Now the question is how to obtain $\mathbf{p}$ if the modeling signal matrix $\mathbf{M}$ and the target impulse response $\mathbf{h}$ are known. The problem reduces to finding the optimal parameters $\mathbf{p}_\mathrm{opt}$ such that $\mathbf{h} = \mathbf{M}\mathbf{p}_\mathrm{opt}$ is closest to the target response $\mathbf{h}_\mathrm{t}$. If the error function is evaluated in the mean squares sense,

$$e_\mathrm{LS} = \sum_{n=1}^{N} |h(n) - h_\mathrm{t}(n)|^2 = (\mathbf{h}-\mathbf{h}_\mathrm{t})^H(\mathbf{h}-\mathbf{h}_\mathrm{t}), \tag{6.5}$$

the optimum is found by the least-squares (LS) equations, as already outlined in Sec. 2.3:

$$\mathbf{p}_{\text{opt}} = \mathbf{M}^{+}\mathbf{h}_{\text{t}}, \tag{6.6a}$$
$$\mathbf{M}^{+} = (\mathbf{M}^{H}\mathbf{M})^{-1}\mathbf{M}^{H}, \tag{6.6b}$$

where $\mathbf{M}^{+}$ is the Moore-Penrose pseudoinverse, and $\mathbf{M}^{H}$ is the conjugate transpose of $\mathbf{M}$. Note that usually we are interested in modeling real impulse responses, and in this case the conjugate transpose is equivalent to a simple transposition.

If the pole set and thus the modeling matrix $\mathbf{M}$ is fixed, the pseudo-inverse $\mathbf{M}^{+}$ can be precomputed and stored, so the parameter estimation reduces to a matrix multiplication according to Eq. (6.6a). This is especially useful for designing multiple sets of filters with the same frequency resolution (e.g., for modeling MIMO systems with common poles, see Sec. 8.1.1), since in this case Eq. (6.6b) has to be computed only once. Another application when this becomes handy is when the target response of the filter may be varied in real-time while the required frequency resolution is fixed, as in the case of a graphic equalizer where the user can change the target curve by sliders. Actually this property makes the graphic equalizer design method proposed in [Bank et al. 2017] highly efficient.

A minimum-phase loudspeaker–room response modeling example is shown for a 32nd order parallel filter (16 second-order sections) in Fig. 6.2 thick line. The pole positions are set according to a logarithmic frequency scale as will be discussed later in Sec. 7.2. The thin line displays the target response, the thick line the filter response, and the separate transfer functions of the second-order sections are also visualized by dashed lines. It can be seen that the parallel filter approximates the given response as a combination of bell-shaped transfer functions, similarly to a graphic equalizer. However, here not only the gains, but also the phases of the different "bands" are free parameters. The effect of this is especially noticeable around the dips of the filter response: since there the basis functions have a large magnitude, these dips must be produced by cancellations of the basis functions, meaning that they have opposing phase at the dip frequency. Also note that the transfer function (thick line) follows the local average of the target (thin line), that is, the filter design performs some kind of smoothing. The theoretical reasons for this will be discussed in Sec. 6.5.

6.2.2 Frequency-domain design

Substituting $z^{-1} = e^{-j\vartheta_n}$ into Eq. (6.2) for a finite set of ϑ_n angular frequencies yields

$$H(\vartheta_n) = \sum_{k=1}^{K} \frac{b_{k,0} + b_{k,1}e^{-j\vartheta_n}}{1 + a_{k,1}e^{-j\vartheta_n} + a_{k,2}e^{-j2\vartheta_n}} + \sum_{m=0}^{M} f_m e^{-jm\vartheta_n}, \tag{6.7}$$

which is again written in a matrix form

$$\mathbf{h} = \mathbf{M}\mathbf{p}, \tag{6.8}$$

where $\mathbf{p} = [b_{1,0}, b_{1,1}, \ldots b_{K,0}, b_{K,1}, f_0 \ldots f_M]^T$ is a column vector composed of the free parameters. The columns of the modeling matrix $\mathbf{M}$ contain the transfer functions of the second-order sections $1/(1 + a_{k,1}e^{-j\vartheta_n} + a_{k,2}e^{-j2\vartheta_n})$ and their delayed versions $e^{-j\vartheta_n}/(1 + a_{k,1}e^{-j\vartheta_n} + a_{k,2}e^{-j2\vartheta_n})$ for the ϑ_n angular frequencies. The last columns of $\mathbf{M}$ are the transfer functions of

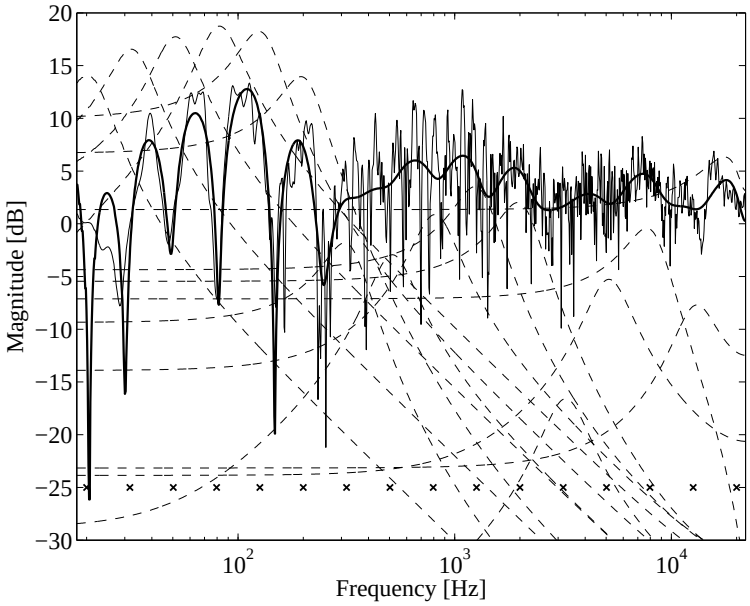


Figure 6.2: 32nd order parallel filter design with logarithmic pole positioning: the minimum-phase target response (thin solid line), the magnitude response of the parallel filter (thick solid line) and magnitude responses of the second-order sections (dashed lines). The crosses indicate the pole frequencies.

the FIR part $e^{-jm\vartheta_n}$ for $m = [0 \dots M]$. Finally, $\mathbf{h} = [H(\vartheta_1) \dots H(\vartheta_N)]^T$ is a column vector composed of the resulting frequency response.

Now the task is to find the optimal parameters $\mathbf{p}_{\text{opt}}$ such that $\mathbf{h} = \mathbf{M}\mathbf{p}_{\text{opt}}$ is closest to the target frequency response $\mathbf{h}_\text{t} = [H(\vartheta_1)_\text{t} \dots H_\text{t}(\vartheta_N)]^T$. If the error is evaluated in the mean squares sense

$$e_{\text{LS}} = \sum_{n=1}^{N} |H(\vartheta_n) - H_\text{t}(\vartheta_n)|^2 = (\mathbf{h} - \mathbf{h}_\text{t})^H(\mathbf{h} - \mathbf{h}_\text{t}), \tag{6.9}$$

the minimum of Eq. (6.9) is found by the least-squares (LS) solution, similarly to the time-domain case:

$$\mathbf{p}_{\text{opt}} = \mathbf{M}^+\mathbf{h}_\text{t}, \tag{6.10a}$$
$$\mathbf{M}^+ = (\mathbf{M}^H\mathbf{M})^{-1}\mathbf{M}^H, \tag{6.10b}$$

where $\mathbf{M}^+$ is the Moore-Penrose pseudoinverse, and $\mathbf{M}^H$ is the conjugate transpose of $\mathbf{M}$. Similarly to the time-domain design, $\mathbf{M}^+$ can be precomputed if the pole set is fixed.

Note that Eq. (6.10) assumes a filter specification $H_\text{t}(\vartheta_n)$ given for the full frequency range $\vartheta_n \in [-\pi, \pi]$. Thus, the design can be used for obtaining filters with complex coefficients, since the frequency specification is not constrained to be conjugate-symmetric. However, in most of the cases we are interested in filters with real coefficients: in this case we have to ensure that $H_\text{t}(-\vartheta_n) = \overline{H}_\text{t}(\vartheta_n)$, where $\overline{H}_\text{t}$ is the complex conjugate of H_t (an alternative approach using a one-sided specification is outlined in [Bank 2011c; Rämö et al. 2014]).

A clear benefit of designing the fixed-pole parallel filter in the frequency-domain is that this allows adding different weights to the different frequency points. In this case, the error becomes

$$e_{\text{WLS}} = \sum_{n=1}^{N} W(\vartheta_n)|H(\vartheta_n) - H(\vartheta_n)_\text{t}|^2 = (\mathbf{h} - \mathbf{h}_\text{t})^H\mathbf{W}(\mathbf{h} - \mathbf{h}_\text{t}), \tag{6.11}$$

where $W(\vartheta_n)$ is the weight for the ϑ_n frequency, and $\mathbf{W}$ is the weighting matrix having $W(\vartheta_n)$ in its diagonal and zeros elsewhere. The minimum is obtained by the weighted-least-squares (WLS) solution (see Sec. 2.3.1):

$$\mathbf{p}_{\text{opt}} = (\mathbf{M}^H\mathbf{W}\mathbf{M})^{-1}\mathbf{M}^H\mathbf{W}\mathbf{h}_{\text{t}}. \tag{6.12}$$

6.2.3 Magnitude-only design

In some applications only the magnitude of the target frequency response is specified, and the phase of the filter can be arbitrary. In this case the magnitude error

$$e_{\text{magn}} = \sum_{n=1}^{N}(|H(\vartheta_n)| - |H_{\text{t}}(\vartheta_n)|)^2, \tag{6.13}$$

should be minimized instead of the complex transfer function error of Eq. (6.9). This is a much more complicated problem since Eq. (6.13) is a nonlinear function of the filter weights $b_{k,0}, b_{k,1}, f_m$ as opposed to the previous cases. To overcome this difficulty, I have proposed an iterative least squares technique in [Bank 2011c] inspired by the method of [Jackson 2008]. The method is based on the fact that minimizing the complex transfer function error of Eq. (6.9) corresponds to magnitude error minimization if the phase of the specification $\varphi\{H_{\text{t}}(\vartheta_n)\}$ equals with the phase of the filter $\varphi\{H(\vartheta_n)\}$, since in this case we have

$$\begin{aligned} e_{\text{LS}} = |H(\vartheta_n) - H_{\text{t}}(\vartheta_n)|^2 = \left||H(\vartheta_n)|e^{j\varphi\{H(\vartheta_n)\}} - |H_{\text{t}}(\vartheta_n)|e^{j\varphi\{H(\vartheta_n)\}}\right|^2 = \\ \left|e^{j\varphi\{H(\vartheta_n)\}}\left(|H(\vartheta_n)| - |H_{\text{t}}(\vartheta_n)|\right)\right|^2 = (|H(\vartheta_n)| - |H_{\text{t}}(\vartheta_n)|)^2 = e_{\text{magn}}. \end{aligned} \tag{6.14}$$

The condition of $\varphi\{H_{\text{t}}(\vartheta_n)\} = \varphi\{H(\vartheta_n)\}$ is assured by an iterative procedure, where the phase of the specification is adjusted to match the phase of the filter obtained in the previous iteration by setting $\varphi\{H_{\text{t},i}(\vartheta_n)\} = \varphi\{H_{i-1}(\vartheta_n)\}$, while its magnitude is kept unchanged. Then, a new filter $H_i(\vartheta_n)$ is designed based on this updated specification until convergence is reached. Jackson [2008] uses the above idea for his frequency-domain Steiglitz-McBride algorithm and starts the iteration with a zero-phase target $H_{\text{t}}(\vartheta_n)$. However, I suggest to use a minimum-phase target specification $H_{\text{t},0}(\vartheta_n)$ since that can be followed by IIR filters much more easily (see Sec. 2.4). The target phase of the first iteration $\varphi\{H_{\text{t},1}(\vartheta_n)\}$ is thus obtained from the magnitude specification $|H_{\text{t}}(\vartheta_n)|$ based on the Hilbert-transform relation of magnitude and phase of minimum-phase transfer functions [Oppenheim et al. 1999]. This makes the convergence of the procedure significantly faster compared to using zero initial phase, requiring five-ten iterations in practice [Bank 2011c].

6.2.4 Comparison of time- and frequency-domain filter design

The time-domain and frequency-domain versions of parallel-filter design provide the same result if the ϑ_n frequencies are distributed evenly according to a linear frequency scale and the grid is dense enough. This is due to Parseval's theorem: if the energy of the estimation error is minimal in the time-domain, so it is in the frequency-domain.

However, if the ϑ_n frequencies are given at a logarithmic frequency scale, different results could be expected, because now the error is minimized over a logarithmic frequency grid as

opposed to the linear-grid equivalent of the time-domain design. It is actually quite surprising to see that the differences of the two designs are practically negligible if the time-domain specification is long enough (lasts until the target impulse response has faded out completely). The reasons can be understood by looking at the elementary transfer functions (dashed lines) in Fig. 6.2: they are localized in frequency so that only the target points having their frequencies close to the peak of the basis functions will contribute to the weights of the corresponding second-order sections. Even if the linear frequency grid corresponding to the time domain design means an overly dense specification at high frequencies when plotted in the logarithmic frequency scale, the high-frequency target points have almost no effect on the low-frequency basis functions since the basis functions die out relatively fast as a function of frequency, as can be seen in Fig. 6.2.

This is displayed in Fig. 6.3 for the same loudspeaker–room response target as for Fig. 6.2. It can be seen that if the time-domain LS fit is performed for a long enough target response ($N = 130000$ in this case), the time-domain design (b) provides practically the same fit as the frequency-domain design using 1000 logarithmically spaced target points (a), with the exception of some inaccuracies below 40 Hz. On the other hand, if the target impulse response is truncated ($N = 2000$ in this example), it can lead to unacceptable fitting errors as shown in (c), thick dashed line. The low frequency boost of Fig. 6.3 (c) actually comes from the fact that the error is minimized in Eq. (6.5) only up to sample N and for $n > N$ $h(n)$ has actually a "don't care" region. In some cases having a good fit for the samples $0 \geq n \geq N$ can lead to extremely large response in the latter part $n > N$ since there is actually no penalty in the error function for such behavior. For the time-domain design this can be avoided by using the entire target impulse response until full decay, or, if that is not available, zero-padding the target to force the LS design to make the filter response decay fast. Of course this means a much larger target vector, and thus much longer design time. For room responses, the frequency-domain method requires 100 times fewer specification points and thus around two orders of magnitude smaller design time compared to the time-domain design for the same accuracy. Therefore, for such long impulse responses as that of a room response, it is advised to convert them to the frequency domain by the help of an FFT and then resample it to logarithmic frequency scale so that frequency-domain design variant can be used. On the other hand, for shorter impulse responses, such as the anechoic response of loudspeakers typically lasting a few thousand samples, the advantage of the frequency-domain method vanishes.

6.3 Direct equalizer design

Equalizing a system (such as a loudspeaker) by the parallel filter can be done by inverting the system response as described in Sec. 2.2.2 and designing the parallel filter as outlined in the previous section. In the frequency-domain this can be done by dividing the desired target response $H_\mathrm{t}(\vartheta_n)$ (e.g., a bandpass response) by the system response $H_\mathrm{s}(\vartheta_n)$ and designing a parallel filter for this $H_\mathrm{t}(\vartheta_n)/H_\mathrm{s}(\vartheta_n)$ specification according to Sec. 6.2. However, the narrow dips of $H_\mathrm{s}(\vartheta_n)$ result in sharp peaks in $H_\mathrm{t}(\vartheta_n)/H_\mathrm{s}(\vartheta_n)$ because of the division, biasing the filter design. While the problems of division can be reduced by regularization, a more appropriate way of designing an equalizer is to minimize the time- or frequency-domain error between the final, equalized response and the target response, as discussed in Sec. 2.2.1. This is similar to a system identification problem with output error minimization: the input of the parallel filter

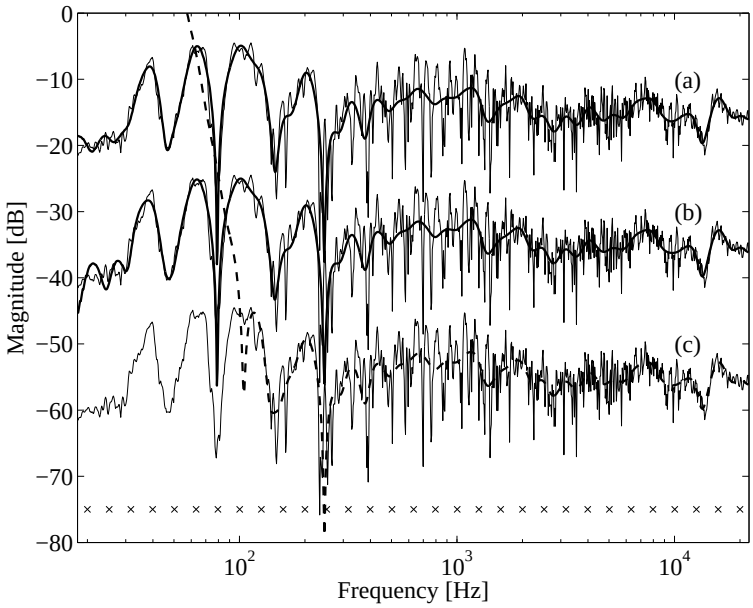


Figure 6.3: 62nd order parallel filter design with logarithmic pole positioning: (a) frequency-domain design with a logarithmic frequency scale specification having 1000 target frequency points, (b) time-domain design with a target impulse response length of 130000 samples, and (c) time-domain design with a target impulse truncated to 2000 samples. The curves are offset for clarity. Thin lines: minimum-phase target response, thick lines: filter responses. The crosses indicate the pole frequencies.

is the system response and we should estimate the filter parameters such that its output best matches the target response.

6.3.1 Time-domain equalizer design

Designing an equalizer requires that the resulting response $h(n)$, which is the convolution of the equalizer response $h_{\mathrm{eq}}(n)$ and the system response $h_{\mathrm{s}}(n)$, is close to the target response $h_{\mathrm{t}}(n)$ (which can be a unit impulse, for example). In our case, this means that the input of the parallel filter is the system response $h_{\mathrm{s}}(n)$ and its output $h(n)$ should match the target response $h_{\mathrm{t}}(n)$. The output of the parallel filter is computed as

$$\begin{aligned} h(n) = h_{\mathrm{eq}}(n) * h_{\mathrm{s}}(n) = \\ \sum_{k=1}^{K} b_{k,0} u_k(n) * h_{\mathrm{s}}(n) + b_{k,1} u_k(n-1) * h_{\mathrm{s}}(n) + \\ \sum_{m=0}^{M} f_m \delta(n-m) * h_{\mathrm{s}}(n) = \\ \sum_{k=1}^{K} b_{k,0} s_k(n) + b_{k,1} s_k(n-1) + \sum_{m=0}^{M} f_m h_{\mathrm{s}}(n-m), \end{aligned} \quad (6.15)$$

where $*$ denotes convolution. The signal $s_k(n) = u_k(n) * h_{\mathrm{s}}(n)$ is the system response $h_{\mathrm{s}}(n)$ filtered by $1/(1 + a_{k,1}z^{-1} + a_{k,2}z^{-2})$. It can be seen that Eq. (6.15) has the same structure as Eq. (6.3). Therefore, the parameters $b_{k,0}$, $b_{k,1}$, and f_m can be estimated in the same way as

presented in the previous section. Similarly, writing this in a matrix form yields

$$\mathbf{h} = \mathbf{M}_{\mathrm{eq}}\mathbf{p}, \tag{6.16}$$

where the columns of the new signal modeling matrix $\mathbf{M}_{\mathrm{eq}}$ contain $s_k(n)$, $s_k(n-1)$, and finally the system response $h_{\mathrm{s}}(n)$ and its delayed versions up to $h_{\mathrm{s}}(n-M)$. The samples of the target impulse response are forming the column vector $\mathbf{h}_{\mathrm{t}}$. The optimal set of parameters is again obtained by

$$\mathbf{p}_{\mathrm{opt}} = (\mathbf{M}_{\mathrm{eq}}^H\mathbf{M}_{\mathrm{eq}})^{-1}\mathbf{M}_{\mathrm{eq}}^H\mathbf{h}_{\mathrm{t}}. \tag{6.17}$$

6.3.2 Frequency-domain equalizer design

Since the input of the parallel filter is the system response $H_{\mathrm{s}}(\vartheta_n)$, we should estimate the filter parameters such that its output, which is the product of the system response and the frequency response of the parallel equalizer $H(\vartheta_n) = H_{\mathrm{s}}(\vartheta_n)H_{\mathrm{eq}}(\vartheta_n)$, best matches the target response $H_{\mathrm{t}}(\vartheta_n)$.

Accordingly, the equalized response is given by

$$H(z) = H_{\mathrm{eq}}(z)H_{\mathrm{s}}(z) = \sum_{k=1}^{K}\frac{b_{k,0}+b_{k,1}z^{-1}}{1+a_{k,1}z^{-1}+a_{k,2}z^{-2}}H_{\mathrm{s}}(z) + \sum_{m=0}^{M} f_m z^{-m}H_{\mathrm{s}}(z). \tag{6.18}$$

Writing this in a matrix form for a finite set of ϑ_n angular frequencies with $z^{-1} = e^{-j\vartheta_n}$ again yields

$$\mathbf{h} = \mathbf{M}_{\mathrm{eq}}\mathbf{p}, \tag{6.19}$$

where $\mathbf{p} = [b_{1,0}, b_{1,1}, \ldots b_{K,0}, b_{K,1}, f_0 \ldots f_M]^T$ is a column vector composed of the free parameters of the parallel equalizer. The columns of the equalizer modeling matrix $\mathbf{M}_{\mathrm{eq}}$ are obtained from the modeling matrix $\mathbf{M}$ constructed in Sec. 6.2.2 by multiplying them with the system frequency response $H_{\mathrm{s}}(\vartheta_n)$. For example, instead of $1/(1+a_{k,1}e^{-j\vartheta_n}+a_{k,2}e^{-j2\vartheta_n})$ we simply have $H_{\mathrm{s}}(\vartheta_n)/(1+a_{k,1}e^{-j\vartheta_n}+a_{k,2}e^{-j2\vartheta_n})$. Finally, $\mathbf{h} = [H(\vartheta_1)\ldots H(\vartheta_N)]^T$ is a column vector composed of the resulting final frequency response. Since Eq. (6.19) has the same structure as Eq. (6.8), the optimal set of parameters are obtained in the same way as in Sec. 6.2.2 by Eq. (6.10).

Naturally, frequency weighting can also be used as described in Sec. 6.2.2, and the discussion about the advantages of the frequency-domain method over the time-domain one given in Sec. 6.2.4 is valid for equalizer design as well.

6.4 Relation of Kautz and parallel filters

6.4.1 The equivalence of approximation properties

Figure 6.4 (a) and (b) show a Kautz and parallel filter design using the same pole set having 31 pole pairs distributed uniformly in the logarithmic scale (both the Kautz and the parallel filter weights have been estimated in the time-domain). As can be seen, the same filter response arises for both filters. This is explained by the fact that the Kautz basis functions are the orthonormalized versions of decaying complex exponentials, which are the basis functions of the

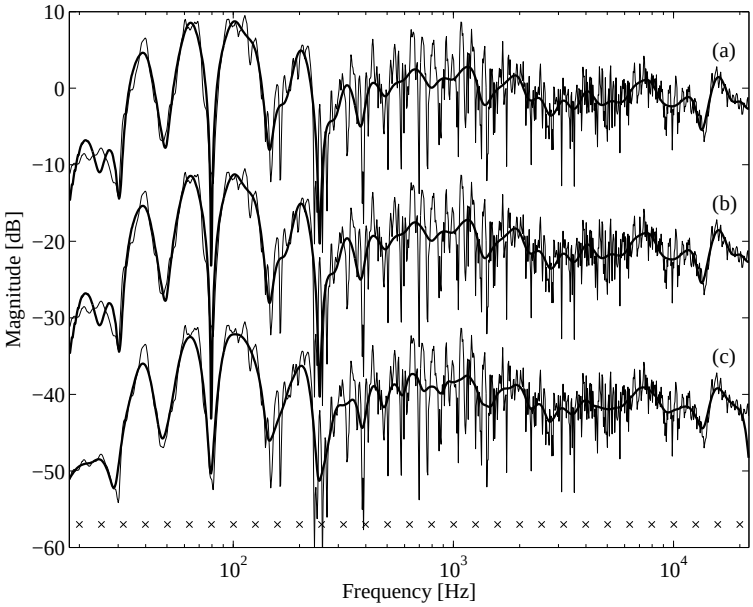


Figure 6.4: Modeling a loudspeaker–room response with a 62nd order Kautz filter (a) and a 62nd order parallel filter (b) having the same set of poles (three poles per octave from 20 Hz to 20 kHz). The pole frequencies are indicated by crosses in the bottom of the figure. The thin lines show the minimum-phase target response, while the thick lines the filter response. In (c) the thin line is the sixth-octave smoothed version of the target (thin line).

parallel filter. I have developed a formal proof for the equivalence of the two structures [Bank 2013a]. The derivation is based on the partial fraction expansion of the complex Kautz basis functions Eq. (5.2):

$$G_k(z) = \frac{\sqrt{1-p_k\overline{p}_k}}{1-p_k z^{-1}} \prod_{j=1}^{k-1} \frac{z^{-1}-\overline{p}_j}{1-p_j z^{-1}} = \sqrt{1-p_k\overline{p}_k} \prod_{j=1}^{k} \frac{1}{1-p_j z^{-1}} \prod_{j=1}^{k-1} (z^{-1}-\overline{p}_j), \qquad (6.20)$$

which are kth order filters ($k = 1, 2, \ldots K$). In the case of no pole multiplicity, which is easily satisfied when the pole set is predetermined, Eq. (6.20) can be written in a partial fraction form

$$G_k(z) = \sum_{i=1}^{k} c_{k,i} \frac{1}{1-p_i z^{-1}}, \qquad (6.21)$$

where the k coefficients $c_{k,i}$ are found by the usual procedure of partial fraction expansion [Rabiner and Gold 1975] with the Heaviside cover-up method:

$$c_{k,i} = (1-p_i z^{-1}) G_k(z)|_{z=p_i} = \sqrt{1-p_k\overline{p}_k} \prod_{j=1, j\neq i}^{k} \frac{1}{1-p_j/p_i} \prod_{j=1}^{k-1} (1/p_i - \overline{p}_j), \qquad (6.22)$$

which finally results in

$$c_{k,i} = \sqrt{1-p_k\overline{p}_k} \prod_{j=1, j\neq i}^{k} \frac{1}{p_i - p_j} \prod_{j=1}^{k-1} (1-\overline{p}_j p_i). \qquad (6.23)$$

By noting that the partial fraction form of Eq. (6.21) is the same as the complex form of the parallel filter Eq. (6.1) without the FIR part ($M = 0$), it is clear that the Kautz basis functions

can be reconstructed by the parallel filter exactly. As the Kautz filter response is the linear combination of the Kautz basis functions $G_k(z)$, it is straightforward to convert a Kautz filter into a parallel filter. If the parameters of the Kautz filter are given in a vector $\mathbf{w} = [w_1, \ldots, w_K]^T$, the parameter vector of the parallel filter $\mathbf{p} = [c_1, \ldots, c_K]^T$ can be obtained by the matrix multiplication

$$\mathbf{p} = \mathbf{K}\mathbf{w}, \tag{6.24}$$

where the conversion matrix $\mathbf{K}$ is given as

$$\begin{aligned} K_{i,k} &= \sqrt{1 - p_k\overline{p}_k} \prod_{j=1, j\neq i}^{k} \frac{1}{p_i - p_j} \prod_{j=1}^{k-1}(1 - \overline{p}_j p_i) \quad \text{for } i \leq k, \\ K_{i,k} &= 0 \quad \text{for } i > k. \end{aligned} \tag{6.25}$$

The matrix $\mathbf{K}$ is triangular, and such matrices are nonsingular if none of the diagonal elements are zero. This is satisfied since $|p_i| < 1$ for all the poles (we assume that the Kautz filter is asymptotically stable). As a result, the inverse matrix $\mathbf{K}^{-1}$ can be computed that can be used to convert the parallel filter parameters to Kautz parameters ($\mathbf{w} = \mathbf{K}^{-1}\mathbf{p}$).

Basically, this proves that the basis functions of the parallel and Kautz filters span the same approximation space, and converting between the two filters is merely a change of basis. Therefore, approximating a target response using any error norm (e.g., the L_2 norm in least-squares design) will lead to exactly the same filter response in both cases for a given pole set p_k. This is shown for the LS design in the Appendix A.2 with some further discussion on the conditions of this equivalence.

While we have related the complex forms of the Kautz and parallel filters, since the real forms Eqs. (5.3) and (6.2) are mathematically equivalent to the complex ones, the results are valid for the more practical real forms as well. This means that pole positioning techniques developed for the parallel filter can also be used for the Kautz filter and vice versa. Also, the smoothing properties derived for the parallel filter in Sec. 6.5 will be valid for the Kautz filter as well.

Besides its theoretical importance, the relation between the two filter structures allows a computationally more efficient design of the parallel filter. Namely, first a Kautz filter is designed by the scalar product of Eq. (5.6), then the parameters are converted by Eqs. (6.24) and (6.25). While this seems to be conceptually more complicated, the number of required arithmetic operations is reduced compared to the LS design of Eq. (6.6), so it is a useful alternative for high (> 100) filter orders. Unfortunately, in the case of direct equalizer design of Sec. 6.3 this cannot be done, since for that case the scalar product of Eq. (5.6) cannot be used and also the Kautz filter has to be designed by a LS equation [Karjalainen and Paatero 2007]. Also note that when the parallel filter is designed in the frequency-domain as in Sec. 6.2.2, the benefit of the conversion-based design is reduced, since the frequency-domain design is already more efficient compared to the time-domain one due to the smaller number of required specification points as discussed in Sec. 6.2.4.

6.4.2 Computational complexity

It can be deduced from Fig. 5.1 that the Kautz backbone requires 4 MAC (multiply and accumulate) operations per second-order section, similarly to series biquads (except the first section that needs 2 MAC). Then the $(1 + z^{-1})$ and $(1 - z^{-1})$ terms need 2 additions (ADD), while

the weights $b_{i,1}$, $b_{i,2}$ and the corresponding output summing require 2 MAC operations. For the parallel filter every second-order section requires 4 MAC operations, including the output summing (see Fig. 6.1). For DSPs where the MAC and ADD instructions take one operation cycle, this means that using the parallel filter structure instead of the Kautz filter reduces the number of arithmetic operations to the half (from 8 to 4). Note that additional operations are also required (reading from the memory to the registers and vice versa), but these can usually be performed in parallel with the arithmetic operations, depending on the processor architecture.

On general purpose processors where MAC instructions are not available, each second-order section for the parallel filter requires 4 additions and 4 multiplications, while for the Kautz filter we need 8 additions and 6 multiplications.[1]

Coming from the fact that the fixed-pole parallel filter has a fully parallel structure, even more significant computational savings can be achieved on parallel architectures such as graphic processing units (GPUs) [Belloch et al. 2014] compared to the series backbone of the Kautz filter that cannot be implemented in parallel.

6.5 Connections to complex smoothing

Figure 6.4 (c) thick line shows the sixth-octave complex smoothed version of the thin line, which is a measured loudspeaker–room response. By observing the results of parallel and Kautz filters with a logarithmic pole distribution in Fig. 6.4 (a) and (b), it is apparent that the effect of filter design is similar to that of fractional-octave complex-smoothing of transfer functions. This can be intuitively understood by looking at Fig. 6.2 showing that the total transfer function of the parallel filter (thick line) is a combination of smooth basis functions (dashed lines). I have systematically analyzed this smoothing behavior in [Bank 2010, 2013a] that will be given in the following. While only the case of the parallel filter is discussed here, since it results in exactly the same filter response as the Kautz filter (as it was proven in Sec. 6.4), the observations are valid for the Kautz filter as well.

6.5.1 Uniform pole distribution

We start our analysis with the simplest case, where the K poles of the parallel filter are distributed uniformly on a circle of radius $R < 1$. Then the complex form of the parallel filter becomes

$$H(z) = \sum_{k=1}^{K} \frac{c_k}{1 - p_k z^{-1}} = \sum_{k=1}^{K} \frac{c_k}{1 - Re^{j2\pi k/K} z^{-1}}, \tag{6.26}$$

which, after cross-multiplying all the denominator terms, takes the form

$$H(z) = \prod_{k=1}^{K} \frac{1}{1 - Re^{j2\pi k/K} z^{-1}} \times \sum_{k=1}^{K} c_k \prod_{i=1, i\neq k}^{K} (1 - Re^{j2\pi k/K} z^{-1}), \tag{6.27}$$

where the first part of the product equals $1/(1 - z^{-K}R^K)$, and the second is a $(K-1)$th order polynomial of z^{-1}. Therefore, the transfer function of the parallel filter with uniform pole

[1] Note that in [Bank 2008] I have erroneously underestimated the computational complexity of the Kautz filter by assuming 6 additions per second-order section instead of 8. Therefore the computational benefits of the parallel filter are actually larger than I have stated in [Bank 2008] and in my subsequent publications.

distribution is actually equivalent to a feedback comb filter and a $(K-1)$th order FIR filter in series:

$$H(z) = \frac{1}{1 - z^{-K} R^K} \sum_{k=0}^{K-1} f_k z^{-k}, \tag{6.28}$$

where the FIR coefficients f_k arise as the linear combinations of the parallel filter weights c_k according to the second term of the product in Eq. (6.27). The feedback comb filter repeats the FIR response with an attenuation of R^K in each round, therefore the first K samples of the impulse response of the parallel filter are thus $h(n) = f_n$, the second K samples are $h(n+K) = R^K f_n$, the third K samples are $h(n+2K) = R^{2K} f_n$ for $n = [0 \ldots K-1]$, and so on.

For practical pole sets $R^K \ll 1$, therefore, the impulse response $h(n)$ for $n \geq K$ may be neglected, and thus the transfer function can be approximated by the FIR part only:

$$H(z^{-1}) \approx \sum_{k=0}^{K-1} f_k z^{-k}. \tag{6.29}$$

When designing the parallel filter according to Eq. (6.6), the mean-squared error between the target impulse response $h_\mathrm{t}(n)$ and the filter response $h(n)$ is minimized. This error is minimal if the parallel filter coefficients c_k are set in a way that the equivalent FIR coefficients f_k are equal to the first K samples of the target impulse response $f_k = h_\mathrm{t}(k)$. As a result, the filter impulse response $h(n)$ is the truncated version of the target response $h_\mathrm{t}(n)$, which is equivalent to multiplying the target response by a rectangular window $w(n)$ of length K:

$$h(n) = w(n) h_\mathrm{t}(n), \tag{6.30}$$

where

$$\begin{aligned} w(n) &= 1 \quad \text{for } \ 0 \leq n \leq K-1, \\ w(n) &= 0 \qquad \text{elsewhere.} \end{aligned} \tag{6.31}$$

Note that this window is defined only for positive times $n \geq 0$ (it is a half window), in contrast to the symmetric windows used in complex smoothing (see Chap. 3). Since for causal impulse responses $h(n) = 0$ for $n < 0$ anyway, we may think of symmetrically extending the window $w(n)$ to negative times by setting $w(-n) = w(n)$, without influencing the product $h(n) = h_\mathrm{t}(n) w(n)$. This has the advantage that now the results will be directly comparable with those of complex smoothing.

Accordingly, designing a parallel filter with a uniform pole distribution is equivalent to multiplying the target impulse response by a symmetric rectangular window of total length $2K-1$. In the frequency domain, this corresponds to convolving the target transfer function $H_\mathrm{t}(\vartheta)$ with a sinc-like (periodic sinc) function:

$$H(\vartheta) = H_\mathrm{t}(\vartheta) * \frac{\sin\left(\frac{2K-1}{2}\vartheta\right)}{\sin\left(\frac{1}{2}\vartheta\right)} \tag{6.32}$$

which is clearly a form of transfer function smoothing. Actually, it corresponds to "filtering" the transfer function with an ideal lowpass filter, eliminating all those components that have a periodicity smaller than $4\pi/(2K-1) \approx 2\pi/K$. Since the smoothed transfer function has become band-limited, if it is sampled at a periodicity of π/K, all the information is still retained, thus, we can say that it has a π/K resolution, which actually equals the half of the pole frequency distance $\Delta\theta/2$.

Next we relate the behavior of the design with complex smoothing discussed in Chap. 3. The central lobe of the periodic sinc function in Eq. (6.32) has a somewhat similar shape to that of a Hann window most often used in transfer function smoothing. The width of the main lobe is $4\pi/(2K-1) \approx 2\pi/K$ that actually equals to the pole distance $\Delta\theta$. Therefore, the smoothing behavior will be comparable to smoothing the transfer function with a Hann window whose width equals the pole distance $\Delta\theta$. According to Chap. 3, smoothing with a $\Delta\theta$ wide Hann window corresponds to $\Delta\theta/2$ frequency resolution.

6.5.2 Stepwise uniform pole distribution

Next, let us consider a more interesting case where the pole density is different in the various regions of the frequency range. Since I was not able to derive the smoothing function in a closed form as in Sec. 6.5.1, I have suggested an alternative approach [Bank 2013a] described in the following.

Since the parameters of the parallel filter are determined by a linear LS design, the superposition principle holds. This means that if we compose the target response $h_{\mathrm{t}}(n)$ as a weighted sum of some test functions, the filter response $h(n)$ will be equal to the weighted sum of filter responses designed for the test functions separately. Since we would like to gain some insight to the frequency-dependent nature of smoothing, a natural choice for such a test function is the basis function of the Fourier transform $e^{-j\vartheta_0 n}$, where ϑ_0 is the angular frequency of the complex exponential. In the frequency domain, this is equivalent to $\delta(\vartheta - \vartheta_0)$, which is a Dirac delta function at position ϑ_0. Accordingly, in the frequency domain, we are computing the "impulse response" of the smoothing operation, that is, we obtain the smoothing function directly.

We can assume that if the overlap of the basis functions of the parallel filter is not too large, our test function $e^{-j\vartheta_0 n}$ will be approximated by parallel sections whose center frequencies are near to ϑ_0, while the contribution of the other sections will be negligible. Therefore, we expect that the width of the smoothing function in the frequency domain, and the length of the corresponding window function in the time domain will only depend on the *local* pole density near ϑ_0. From Sec. 6.5.1 we expect that if the distance of the poles is $\Delta\theta$ in some frequency region, that region will be smoothed corresponding to $\Delta\theta/2$ resolution.

A practical example is presented in Figs. 6.5 and 6.6, displaying a parallel filter design with 30 poles (15 pole pairs) around the unit circle. The pole frequencies are chosen in such a way that 20 poles are distributed evenly in the lower half of the frequency range $|\vartheta| \leq \pi/2$, while 10 poles are spread in the upper range $\pi/2 < |\vartheta| \leq \pi$. The pole radius is $R = 0.8$ for all the poles. The dotted vertical lines show the pole frequencies in Fig. 6.6.

In Fig. 6.5 the target impulse responses $e^{-j\vartheta_0 n}$ are displayed by dashed lines, and the resulting parallel filter responses by solid lines. Note that the target and filter responses are complex, here only the real parts of the signals are shown, but the imaginary parts have a similar behavior. (In another interpretation, the figures show how the parallel filter approximates the $\mathrm{Re}\{e^{-j\vartheta_0 n}\} = \cos(\vartheta_0 n)$ function). Figure 6.5 (a) shows a case where the frequency of the exponential test function is in the high pole density region, while in (c) the frequency is in the low pole density region of the filter. As expected, the resulting impulse response (solid line) is

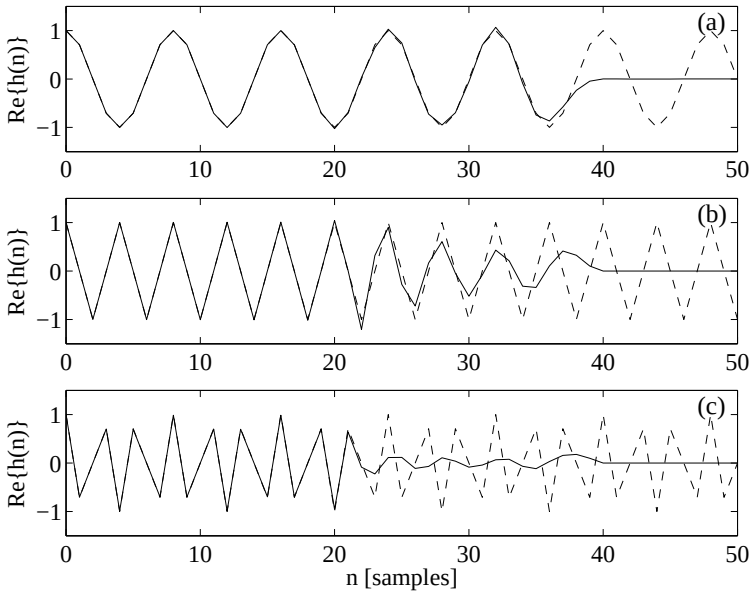


Figure 6.5: Modeling a complex exponential $e^{-j\vartheta_0 n}$ by a parallel filter having stepwise uniform pole frequency set: (a) $\vartheta_0 = 1/4\pi$, (b) $\vartheta_0 = 1/2\pi$, and (c) $\vartheta_0 = 3/4\pi$. The dashed line is the target response $e^{-j\vartheta_0 n}$ and the solid line is the resulting parallel filter impulse response. Only real parts of the signals are shown.

"windowed" to a longer length in the first case compared to the second case. The theoretically computed half window length $2\pi/\Delta\theta$ is 40 and 20 for (a) and (c), which is in a good agreement with what can be observed in practice. Figure 6.5 (b) displays an intermediate case when the frequency of the test signal is exactly at the boundary of the two pole density regions. This results in a more mild windowing, where the window length is somewhere in between the (a) and (c) cases.

The same phenomenon can also be observed in the frequency domain in Fig. 6.6, for the same cases. The solid lines display the transfer functions of the resulting filters trying to approximate the test function $e^{-j\vartheta_0 n}$, again with (a) $\vartheta_0 = 1/4\pi$, (b) $\vartheta_0 = 1/2\pi$, and (c) $\vartheta_0 = 3/4\pi$. Note that the frequency responses were computed by first extending the parallel filter responses to negative times $h(-n) = \overline{h}(n)$, to comply with the symmetric windows used in complex smoothing. Accordingly, the frequency responses displayed in Fig. 6.6 are real (zero phase) functions and can be directly compared to the smoothing functions used in complex smoothing. In Fig. 6.6 the dotted vertical lines show the pole frequencies of the parallel filter. It can be seen in (a) and (c) that the width of the main lobe equals to the pole distance $\Delta\theta$ in that region, and so is the periodicity. Locally, the smoothing function has a sinc-like shape, similarly to the case of the uniform pole distribution of Sec. 6.5.1. The dashed lines show the theoretical $1/|\vartheta - \vartheta_0|$ envelopes of the sinc functions. Again, (b) is a borderline case where the envelope still follows that of a regular sinc function, but the periodicity is different at the left and right sides, coming from the different pole densities.

The results are in line with the expectation that the "width" of the frequency-domain smoothing function (and the length of the corresponding time-domain window) depends only on the local pole density around the frequency of interest. This phenomenon can be effectively utilized for obtaining different resolution (variable amount of smoothing) in different frequency regions by setting the pole density appropriately.

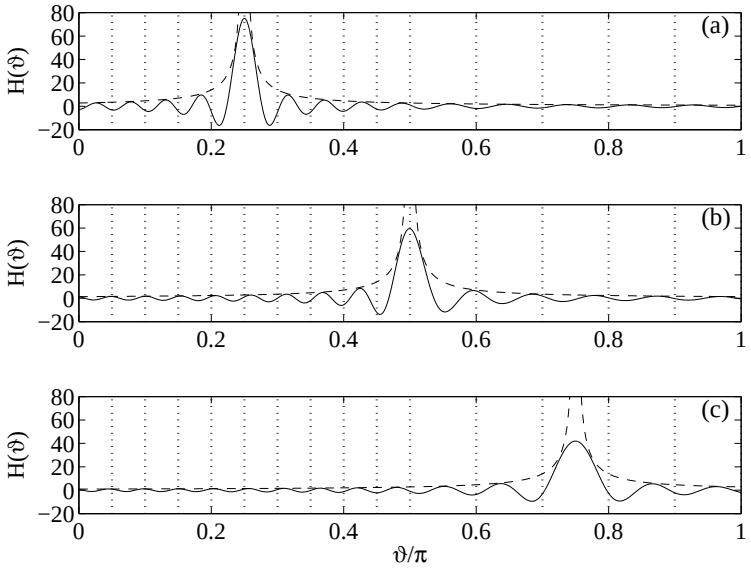


Figure 6.6: Smoothing functions corresponding to the cases of Fig. 6.5 (a)-(c). The dotted vertical lines display the pole frequencies of the parallel filter. The dashed lines show the $1/|\vartheta - \vartheta_0|$ envelopes of the smoothing functions.

6.5.3 Logarithmic pole distribution

As a particularly important case for audio, let us consider the case of the logarithmic pole frequency set. In the following example the parallel filter has three pole pairs in each octave, having all together 31 pole pairs from 20 to 20480 Hz. The test function is again a complex exponential, with $\vartheta_0 = 2\pi f_0/f_\mathrm{s}$, where $f_0 = 1050$ Hz is the frequency of the exponential, and $f_\mathrm{s} = 44.1$ kHz is the sampling frequency. The time domain (real part) and frequency domain responses are displayed in Fig. 6.7 (a) and (b), respectively. Note the linear frequency axis in (b).

It can be seen in Fig. 6.7 (a) that now the target function is "windowed" quite mildly and it has a low frequency tail. In the frequency domain (Fig. 6.7 (b)) the density of the notches of the sinc-like function (solid line) follow that of the pole distribution (dotted vertical lines). However, the envelope of the smoothing function still shows the $1/\vartheta$ behavior, corresponding to the envelope of a regular sinc function.

Figure 6.8 solid line displays the same frequency response on a logarithmic frequency scale. Now it is easy to notice that the periodicity of the window function is exactly logarithmic. It is interesting that while the periodicity (sine part) is related to the logarithm of the frequency, the envelope is related to linear frequency. Based on these observations, I have constructed a "logsinc" function

$$S(\vartheta) = C\frac{\sin(2\pi\alpha\log_2(\frac{\vartheta}{\vartheta_0}))}{\vartheta - \vartheta_0}, \tag{6.33}$$

where C is a positive constant, and α is the pole per octave density (in our case, $\alpha = 3$). This function is displayed by a dashed line in Fig. 6.8 to test if the underlying assumptions

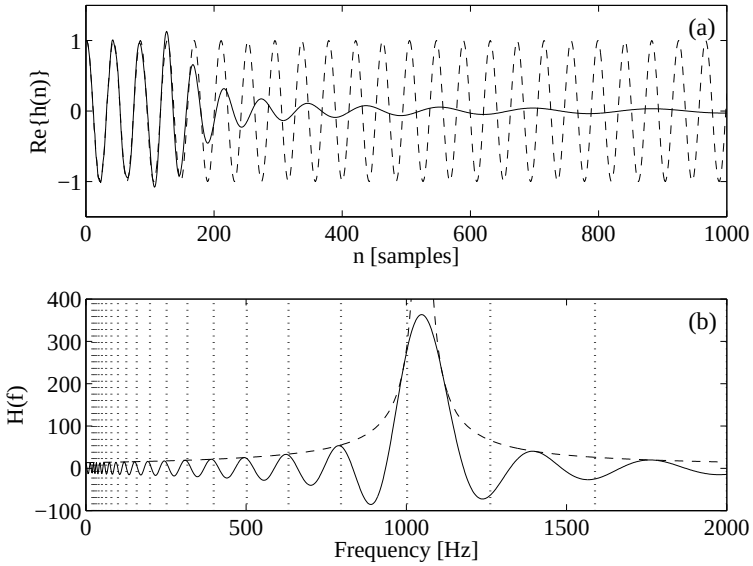


Figure 6.7: Modeling an exponential test function $e^{-j\vartheta_0 n}$ with a parallel filter having a logarithmically distributed pole set: (a) the real parts of the target impulse response (dashed line) and the parallel filter response (solid line), (b) the smoothing function of the parallel filter (solid line) and its $1/|\vartheta - \vartheta_0|$ envelope. The vertical dotted lines display the pole frequencies.

were right. Indeed, the two curves match very precisely with some differences only visible at low frequencies. The match is similarly accurate also for other ϑ_0 test frequencies, meaning that Eq. (6.33) can be used to describe the smoothing behavior of the parallel filter with a logarithmic pole set.

The most important consequence is that the periodicity of the sinc function and the width of its central lobe again follow the pole distances. Since the pole distances linearly increase as a function of frequency due to the logarithmic pole frequency set, the behavior should be similar to fractional octave smoothing.

For example, having third-octave pole distances is comparable to fractional octave smoothing by a third-octave wide Hann window. This corresponds to 6th-octave resolution, as discussed in Chap. 3. The similarity of the filter design to that of fractional octave smoothing can be observed in Fig. 6.4, where the solid lines in (a) and (b) show Kautz and parallel filter designs having three poles per octave, while (c) displays the sixth-octave smoothed target. Some difference is visible between the filter responses (a), (b) and the smoothed response (c), which is due to the fact that the response has been smoothed by a Hann window in (c) while the inherent smoothing of the filter design (a), (b) corresponds to smoothing by a sinc-like function. Nevertheless, their behavior is close enough so that it can be concluded that a logarithmic pole set with $1/\alpha$ octave pole distances is comparable to complex smoothing with $1/(2\alpha)$ octave resolution.

Naturally, further examples could be presented with, e.g., stepwise logarithmic pole distribution, or that of following the Bark or ERB scales [Smith and Abel 1999; Zwicker and Fastl 1990], but according to the above examples, we already have an intuition about the smoothing behavior of the parallel filter.

As an important consequence, by the suitable choice of pole frequencies the frequency resolution of the design can be taken under control. This will be utilized when designing parallel filters with a predetermined pole set in Sec. 7.2.

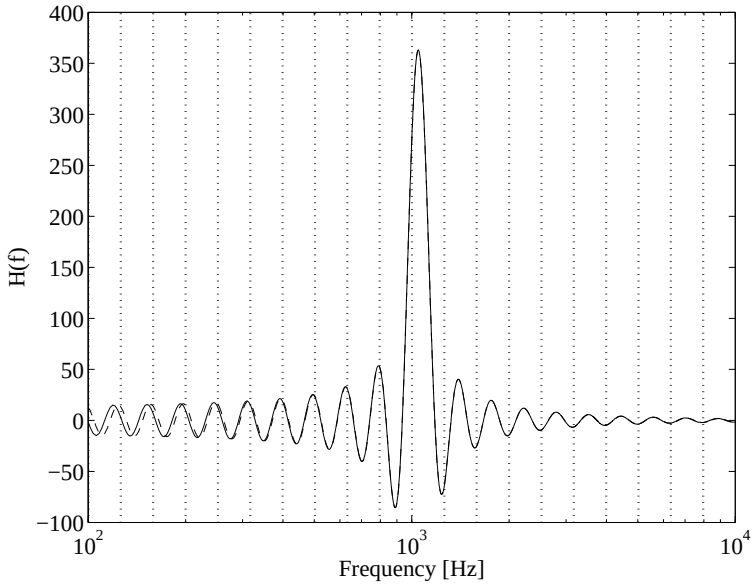


Figure 6.8: The smoothing function of Fig. 6.7 (b) displayed in a logarithmic frequency axis (solid line). The dashed line shows the approximating "logsinc" function Eq. (6.33). The vertical dotted lines display the pole frequencies.

6.6 Scientific contributions

Statement 1: I have created a methodology for the fixed-pole design of parallel second-order filters and demonstrated that the new method can be used for realizing filters with a flexible allocation of frequency resolution similarly to Kautz filters, albeit with a significantly reduced computational complexity.

1.1: I have developed filter design methods for fixed-pole parallel filters both in the time- and frequency-domain: after choosing the poles, the numerator coefficients are computed by a least-squares fit in both cases. In addition, I have proposed an iterative least-squares technique for magnitude-only specifications, and I have also given a direct equalization design method both in the time- and frequency-domain.

1.2: By using partial fraction expansion over the basis functions of the Kautz filter I have shown that the Kautz and the fixed-pole parallel filters lead to the same approximation if their pole set is identical, while the fixed-pole parallel filter requires the half amount of arithmetic operations on DSPs. I have also given a method to convert the Kautz parameters to that of the parallel filter.

1.3: I have shown that the fixed-pole design of second-order parallel filters leads to a filter transfer function similar to complex smoothing the target response, where the frequency resolution (amount of smoothing) is directly controlled by the pole density. A local pole frequency distance $\Delta\theta(\vartheta)$ at ϑ frequency leads to $\Delta\theta(\vartheta)/2$ frequency resolution in terms of smoothing. A logarithmic pole frequency set with $1/\alpha$ octave pole distances is thus comparable to complex smoothing with $1/(2\alpha)$ octave resolution. Since the Kautz filter leads to the same approximation for the same pole set (see 1.2 above), the results are also applicable to the Kautz filter.

The results related to these contributions have been published in three journal papers [Bank 2008, 2011b, 2013a] and in one conference paper [Bank 2007].

Chapter 7

Pole positioning

We have seen in Sec. 6.2 that the parallel filter weights are determined by a least-squares procedure leading to a unique optimum for a given set of poles. Therefore, for a given filter order the accuracy of the parallel filter design depends only on the choice of pole frequencies. The various pole positioning techniques can be put into two main categories, depending on the relation of the system and model order, or in other words, if we wish to model or equalize the system precisely or only approximately. The approximate modeling approach has two subcategories. Let us take a look into these categories first before proceeding to the detailed description of the pole positioning methods.

- **1. Accurate modeling: Pole positioning based on the system response**

  If we aim to model a system precisely, the order of the modeling filter is approximately the same as that of the system. In this case the filter poles should correspond to system poles for best accuracy, and the LS fit discussed in Sec. 6.2 will do its best to match the zeros of the transfer function as well. This is in a way related to the field of system identification since the parameters of the model have a direct connection to physical reality. In this case a straightforward approach is that an IIR filter is designed based on the target response, and the poles of this IIR filter are found as the roots of its denominator. Then these poles are used as the poles of the parallel filter. Here the accuracy is determined by how well the original IIR filter design approximates the target response. This approach will be outlined in Sec. 7.1.

- **2a. Approximate modeling: Predetermined pole set**

  On the other hand, in audio we often aim at modeling only the most important features of the transfer function (the features that can actually be heard), which is better described by the logarithmically smoothed version of the measured response (see Chap. 3). In this case the model order is significantly smaller compared to the order of the system. A typical example is the case of equalizing a loudspeaker–room response, whose order is in the range of hundred-thousand or more ($N > 100000$), but even for a relatively low order system (such as an anechoic loudspeaker response with $N \approx 100$) we may decide to use a lower order filter to fit the available computational resources. In this case the previous idea based on IIR filter design will often not work because the IIR design algorithm may pick and model a few resonant peaks while the others are not modeled (we have seen this behavior for warped IIR design in Fig. 4.4). In other words, the model implements a subset of the original system poles instead of modeling the general trend of the response.

To avoid this, we can take advantage of the fact that the resolution of the design can be directly controlled by the pole density, as seen in Sec. 6.5. Therefore, instead of using poles that have some relation to the poles of the system, we use a predetermined (e.g., logarithmically distributed) set of poles based on the modeling resolution we wish to achieve. Since in this case the poles of the filter, and thus the zeros of the filter are unrelated to the original system, this is more like a nonparametric approach. This method is outlined in Sec. 7.2.

- **2b. Approximate modeling: Pole positioning based on the smoothed response**

  While using a predetermined (e.g., logarithmic) pole set works very well for modeling higher order systems with low order filters, it is still possible to increase the filter design accuracy at the price of increased design complexity. This is because some regions of the system response may be smoother than others, and a strictly logarithmic frequency resolution may waste some computational resources at these already smooth regions. For example, a typical loudspeaker-room response has much larger ripples at low frequencies compared to the high ones after smoothing (see Fig. 3.1 (d) and (e)). This can be taken into account in a simple way by setting higher pole density for low frequencies manually; however, this requires some user interaction. It would be more desirable to have such a method that automatically recognizes which regions need higher resolution. The basic idea is that the target response is smoothed to the required resolution (which again can be different in the various frequency regions), and then this smoothed response is used to determine the optimal pole set of the filter. The pole positions can be either determined based on the raggedness of the smoothed transfer function as in Sec. 7.3.1, or by designing a warped IIR filter based on the smoothed response as in Secs. 7.3.2, 7.3.3, and 7.3.4. Here again the poles and the zeros of the filter will not have a direct connection with the system poles and zeros.

7.1 Pole positioning based on the system response

When the goal is to model a given transfer function accurately, that is, the order of the filter is in the same range as that of the system, it makes sense to use the poles of the system for the design. For estimating the poles I have proposed to use a warped IIR filter design in [Bank 2007], similarly to the case of Kautz filters, where the Brandenstein-Unbehaunen method was used in the warped domain [Paatero and Karjalainen 2003].

As discussed in Chap. 4, for designing a warped IIR filter any IIR filter design or system identification technique can be used: e.g., Prony [Parks and Burrus 1987], the method of Steiglitz and McBride [1965], balanced model reduction [Beliczynski et al. 1992], or the method of Brandenstein and Unbehauen [1998]. The Steiglitz-McBride method with no iterations is equivalent to the Prony method, and while its stability is not guaranteed, in practice the iterations almost always improve over the results of the Prony method. It is stated in [Brandenstein and Unbehauen 1998] that their method gives basically the same filter responses as that in [Beliczynski et al. 1992], albeit with much lower computational cost. I have made various tests using the Brandenstein-Unbehaunen technique and found that it practically gives the same approximation error as the Steiglitz-McBride method. On the other hand, the Steiglitz-McBride method can be used for estimating an IIR filter from a given input to a given output required for designing equalizer filters, which is not available for the Brandenstein-Unbehaunen technique.

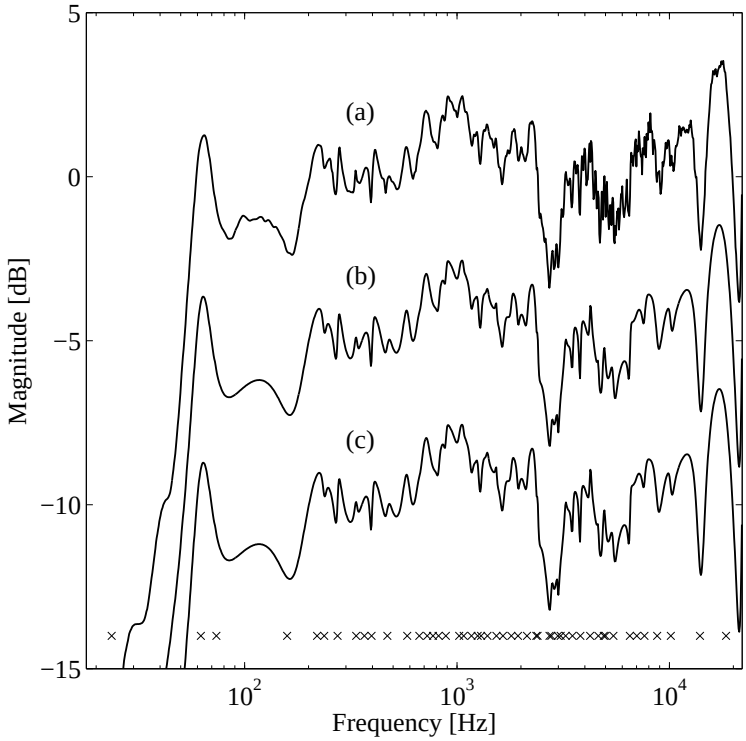


Figure 7.1: Anechoic loudspeaker response (a), modeled by a 50th order warped IIR filter using $\lambda = 0.9$ (b) and by a 100th order parallel filter based on the poles of the warped IIR filter (c). The curves are offset for clarity.

Based on these observations and its availability in MATLAB (`stmcb` function), I have decided to use the Steiglitz-McBride method for designing the underlying IIR filters used for pole positioning. However, the methods developed here are not relying on the actual IIR estimation technique, and the embedded IIR filter design step can be arbitrary, including those techniques that minimize a norm other than L_2.

In the example of Fig. 7.1, first a 100th order warped IIR filter is identified by the Steiglitz-McBride method [Steiglitz and McBride 1965] based on the warped version of a loudspeaker response (a). In this example, a warping parameter $\lambda = 0.9$ is used and the resulting filter response is displayed in Fig. 7.1 (b). Then, the poles $\tilde{p}_k$ of the identified IIR filter are obtained, and dewarped by Eq. (4.7). These dewarped poles are used for fixed-pole parallel filter design, shown in Fig. 7.1 (c).

It can be seen in Fig. 7.1 that the warped IIR filter (b) follows the target response (a) very well on the logarithmic scale. However, it is also apparent that the small ripples of the transfer function are not followed exactly above 3 kHz. While the practical need for such a detailed modeling is questionable, the phenomenon still shows that the modeling detail is concentrated to a specific region (approximately between 200 Hz and 3 kHz), and not spread completely evenly on the logarithmic scale, as already discussed in Sec. 4.3. This shortcoming will be addressed in the pole positioning methods of Secs. 7.3.3 and 7.3.4.

Figure 7.1 also shows that the frequency response of the parallel filter (c) and that of the warped IIR filter from where the poles originate (b) are practically identical. However, there is a difference in filter implementation. Parallel filters are implemented as parallel second-order sections, while for warped IIR filters there are different possibilities, as discussed in Sec. 4.4.

Table 7.1: Pole densities, number of pole frequencies and filter orders for typical values of fractional-octave resolutions using a predetermined pole set. The pole frequencies span 10 octaves from 20 Hz to 20480 Hz.

Fractional octave resolution	1	1/3	1/6	1/12	1/24
Pole frequencies per octave	1/2	3/2	3	6	12
Total no. pole freqencies	6	16	31	61	121
Filter order	12	32	62	122	242

The most efficient possibility is to dewarp them to series second-order sections, but in that case the warped implementation will have higher quantization noise than that of the parallel filter as demonstrated in [Bank and Horváth 2017a] using floating point arithmetic. Another possibility for implementing the warped IIR filter is to dewarp it to parallel second-order sections. However, that leads to second-order numerators as discussed in Sec. 4.4.3, while when the parallel filter is designed using the WIIR poles, the numerators are of first order, leading to lower computational complexity. To sum up, while both the parallel filter Fig. 7.1 (b) and the warped IIR filter Fig. 7.1 (c) have the same frequency response, the parallel filter is more advantageous from the implementation point of view in terms of computational complexity and quantization noise.

7.2 Pole positioning using a predetermined pole set

As already discussed, for high-order systems, such as a loudspeaker–room response we usually design a modeling filter or equalizer whose order is significantly lower than that of the system.

It has been observed for Kautz filters that a logarithmic pole set produces logarithmic frequency resolution, and the equations for obtaining such a logarithmic pole set have been given in [Paatero and Karjalainen 2003] (see Eqs. (5.7) and (5.8)), which can of course be used for the parallel filter as well. However, no direct connection between the resolution of the design and the number of poles have been given in [Paatero and Karjalainen 2003]. After finding the relation between complex smoothing and the approximation properties of parallel filters, I have given the connection of the pole density and filter resolution which allow an accurate control of the modeling detail of the filter [Bank 2013a], already discussed in Sec. 6.5.

We have seen in Sec. 6.5 that a $\Delta\vartheta$ pole frequency distance corresponds to a smoothing with $\Delta/2$ resolution. Turning this around, if the desired resolution is $\Delta\vartheta$ in some frequency region, the pole frequency distances should be the double, that is, $2\Delta\vartheta$ in that region.

For logarithmic frequency resolution, if we wish to achieve a result similar to $1/\beta$ octave smoothing, we need to have $2/\beta$ octave pole frequency distances. In other words, this means that we need to have a pole density of $\beta/2$ poles per octave. Table 7.1 displays the pole densities required to achieve the most typical fractional-octave resolutions used in transfer function smoothing. It also lists the number of pole frequencies $K = 10(\beta/2) + 1$ assuming a design with the ten octaves of the full audio bandwidth, and the total filter order, which is $2K$ in this case.

The equations for obtaining a logarithmic pole set has been given in [Paatero and Karjalainen 2003] (see Eqs. (5.7) and (5.8)) for the Kautz filter that can also be used for parallel filter design. However, the difficulty in their approach lies in manually setting the damping

parameter R in Eq. (5.8); Paatero and Karjalainen [2003] suggest that it should be tuned so that the individual Kautz outputs cross at their -3 dB point. Some examples are also presented by using manually chosen pole frequencies for the Kautz filter in [Paatero and Karjalainen 2003], but in that case tuning the damping factors of each pole manually creates a tedious task.

Therefore I have developed a more general set of pole equations which can determine the pole radii for an arbitrary set of pole frequencies [Bank 2013a]. The pole radii $R_k = |p_k|$ are computed based on the bandwidths $\Delta\theta_k$ as

$$R_k = e^{-\frac{\Delta\theta_k}{2}}. \tag{7.1}$$

Accordingly, the poles p_k become

$$\theta_k = \frac{2\pi f_k}{f_s} \tag{7.2a}$$

$$p_k = e^{-\frac{\Delta\theta_k}{2}} e^{j\theta_k}, \tag{7.2b}$$

where θ_k are the pole frequencies in radians given by the predetermined analog frequency series f_k and the sampling frequency f_s. For a filter having a real impulse response, the poles must be in complex-conjugate pairs, therefore the pole set needs to be extended by the complex-conjugate version $\overline{p}_k$ of the poles p_k.

The bandwidth of the kth second-order section $\Delta\theta_k$ is computed from the neighboring pole frequencies

$$\begin{aligned} \Delta\theta_k &= \tfrac{\theta_{k+1}-\theta_{k-1}}{2} \qquad \text{for } k = [2, .., K-1] \\ \Delta\theta_1 &= \theta_2 - \theta_1 \\ \Delta\theta_K &= \theta_K - \theta_{K-1}. \end{aligned} \tag{7.3}$$

Equation (7.1) sets the pole radii $|p_k|$ in such a way that the transfer functions of the parallel sections cross approximately at their -3 dB points. While the exact value of where the transfer functions cross is not critical (anything in the range of -2 and -4 dB works perfectly) -3 dB is a safe value assuring that there is sufficient overlap between the basis functions, and also complies with the usual choice for defining filter bandwidths. Too small or no overlap means that the target response cannot be approximated between the pole frequencies and this results in a peaky filter frequency response. For a linear pole frequency set in this case we are violating the assumption of $R^K \ll 1$ in Eq. (6.28), meaning that the comb filter part cannot be neglected, indeed leading to a peaky response. Having a large overlap between the basis functions should in theory cause no problems. However, in practice a too large overlap decreases the independence of the basis functions, which spoils the condition number of the normal equations. This can cause numerical errors in the parameter estimation.

The proof of Eq. (7.1) setting the pole radii corresponding to the -3 dB points of the basis functions is presented in Appendix A.3.

In the example of Fig. 7.2, first a third-octave resolution design (a) and then a sixth-octave resolution design (b) is presented, the target is a minimum-phase loudspeaker–room response. The filter orders are 32 and 62, respectively. In (c) the pole frequencies correspond to sixth octave resolution below 300 Hz and to a third-octave resolution above, resulting in 22 pole frequencies, and thus a 44th order parallel filter. It can be seen in (c) that the results below 300 Hz are equivalent to the strictly logarithmic sixth-octave case (b), and to the strictly logarithmic third-octave case above (a), showing that it is indeed possible to apply different frequency resolutions in the different regions of the transfer functions with a smooth transition in between.

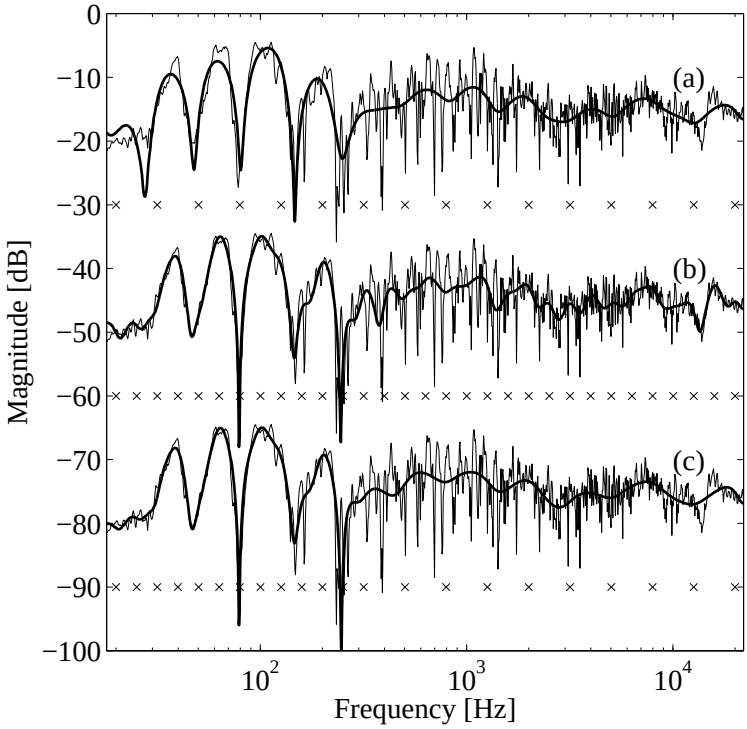


Figure 7.2: Modeling a minimum-phase loudspeaker–room response by fixed-pole parallel filters having a predetermined pole set: (a) third-octave resolution design by using 3/2 pole frequencies per octave (filter order is 32), (b) sixth-octave resolution design by applying 3 poles per octave (filter order is 62). In (c) the pole density is 3 poles per octave below 300 Hz and 3/2 poles per octave above (total filter order is 44). Thin lines: target response, thick lines: filter responses. The pole frequencies are indicated by crosses. The curves are offset for clarity.

Additional examples are displayed in Appendix A.4 including that of manually chosen pole frequencies by the help of an interactive MATLAB script I have developed [Parallel filter homepage 2021], and a synthetic target response in a linear frequency scale design. The examples confirm the complete flexibility in controlling the frequency resolution of the design, not only for the logarithmic case used in most of the examples of this work.

7.3 Pole positioning based on the smoothed system response

Similarly to the predetermined pole positioning, these methods are also related to modeling or equalizing a system with a filter whose order is significantly lower compared to that of the system. However, instead of directly controlling the resolution of the design, in the following methods a different approach is taken: first the system response is smoothed to the desired resolution via complex smoothing (see Chap. 3), and then the poles are determined based on this smoothed response. While this increases the complexity of filter design, it results in better approximation for the same filter order compared to the predetermined pole positioning, as we will see in the comparison later in Sec. 7.4.

7.3.1 Ripple-density based pole positioning

While it is in principle possible to set the pole frequencies based on the properties of the system response (e.g., insert more poles in the problematic or important frequency regions) as demonstrated in Appendix A.4, this requires some expertise from the user. For some applications it would be thus desirable to have such a pole positioning method that does this automatically for a given filter order. This could be included in automatic measurement and equalization systems, for example. To address this, I have developed a computationally efficient method that chooses the pole frequencies based on the smoothed system response, without any prior knowledge on the resolution of smoothing applied in the various regions [Bank 2013c].

The basic idea of the method is that in those regions where there are more ripples in the transfer function, more poles are needed. The steps of the algorithm are explained by using Fig. 7.3. The sixth-octave smoothed version of the loudspeaker–room response is shown in Fig. 7.3 (a), given at logarithmically spaced frequency points. The first step of the algorithm is computing the absolute differences of the adjacent frequency response points in dB scale. The level of this function will be proportional to the raggedness of the transfer function, so it is called ripple density. The result is displayed in Fig. 7.3 (b). The goal is now to divide this ripple density function in as many equal areas as many poles we would like to use, and the borders of these areas will be chosen as pole frequencies. This is done by integrating (in practice cumulating) the density function giving the ripple distribution function. Then this is scaled so that it goes from zero to the number of pole frequencies minus one, displayed in Fig. 7.3 (c). Finally, whenever this distribution function is integer (crosses a horizontal line in Fig. 7.3 (c)) a pole frequency is obtained, displayed by crosses. The same pole frequencies are also displayed in Fig. 7.3 (d), showing that indeed more poles are placed in those regions where the target response (thin line) has larger variation. Once the pole frequencies are obtained, the pole radii are determined by Eqs. (7.2) and (7.3) in the same way as with the predetermined pole set. Figure 7.3 (d) thick line shows the frequency response of the parallel filter designed using the pole set obtained from the ripple density function (the smoothed target is displayed by a thin line). It can be seen that the method recognizes the problematic low-frequency region and increases the frequency resolution by placing more poles in that region. Naturally, the method can also be applied in exactly the same way for obtaining arbitrary (non-logarithmic) frequency resolution if the target response is smoothed according to the frequency resolution profile we wish to achieve.

7.3.2 Pole positioning based on a warped IIR filter design

The next step in terms of design complexity is to determine the pole positions of the parallel filter by estimating a warped IIR filter based on the smoothed system response [Bank 2013c]. The design procedure is similar to accurately modeling the system response by a warped IIR filter as discussed in Sec. 7.1, with the difference that now the complex-smoothed version of the system response is used as a target. Therefore, the details are omitted here, and the performance of the method for smoothed responses will be presented when comparing the various techniques in Sec. 7.4.

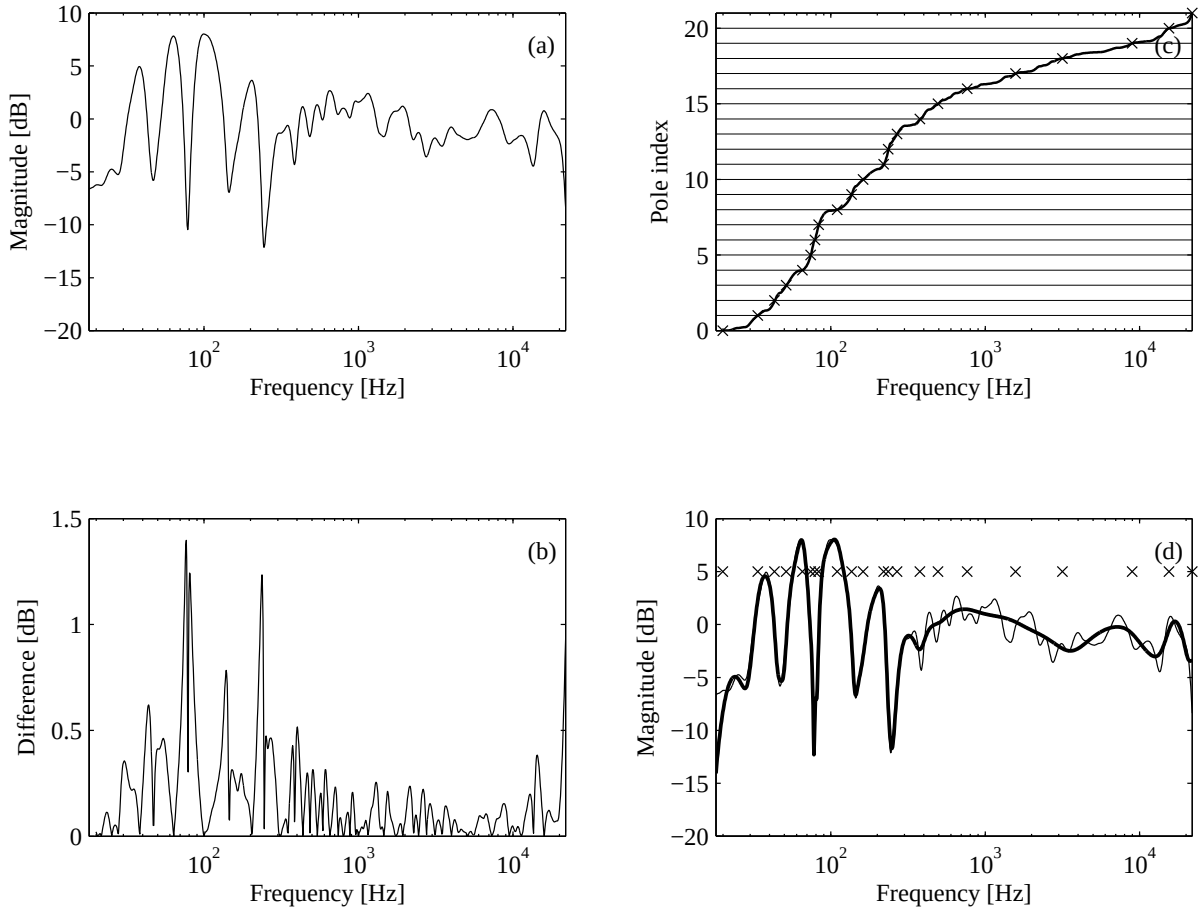


Figure 7.3: Steps of ripple-density based pole positioning: (a) the sixth-octave smoothed magnitude response, (b) the ripple density (the differences between the adjacent points of the smoothed response), and (c) the ripple distribution (scaled cumulative sum of ripple density). In (c) horizontal lines show the pole indices, and the pole frequencies are displayed by crosses. In (d) these pole frequencies are used to design a parallel filter (thick line) to fit the minimum-phase smoothed target response (thin line). The pole frequencies are marked by crosses. The filter order is 44, the same as in Fig. 7.2 (c).

7.3.3 Pole positioning based on multi-band warping

We have seen in Chap. 4 that a warped IIR filter design with a single λ parameter cannot spread the frequency resolution evenly in the logarithmic scale, and that there is always a certain frequency region where the resolution is maximal ($\Delta\tilde{f}/f$ is minimal in Fig. 4.2 (b)). This means that using the poles of such a warped filter cannot have a logarithmic resolution either.

To address this problem, I have developed a technique based on multi-band warping [Bank and Ramos 2011]. In this method the transfer function is divided into separate frequency bands, and different warping parameters are used in each band so that the warping effect is maximized in each region. Then, separate warped IIR filters are designed for the different regions of the smoothed response, their poles are dewarped, and finally the pole sets are united and used for parallel filter design.

The first step of filter design is splitting the frequency region to separate bands and choosing appropriate λ values for the bands. In its simplest form, two bands are used with a crossover frequency of $f_\mathrm{c} = 500$ Hz, which is approximately in the middle of the [20 Hz, 20 kHz]

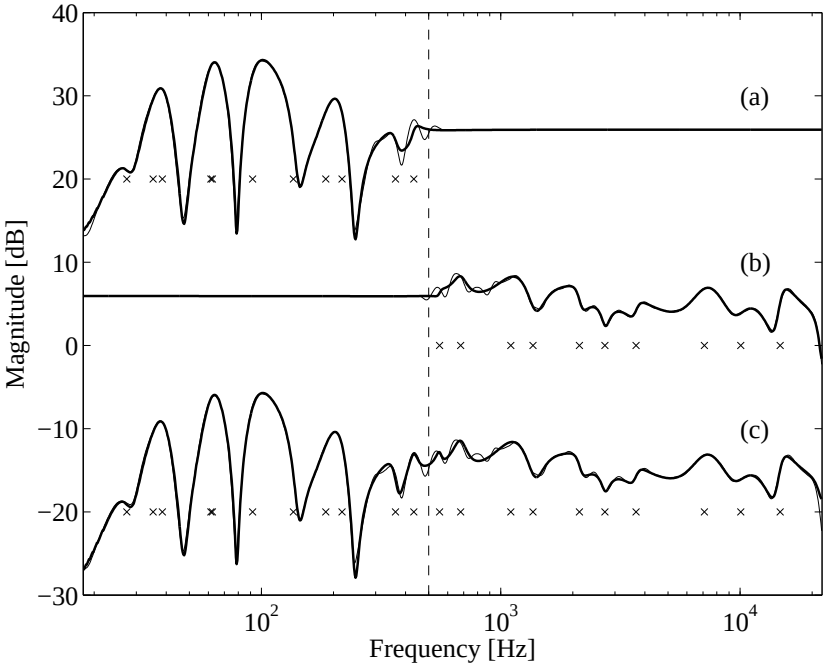


Figure 7.4: Dual-band warped IIR filter design based on a sixth-octave smoothed loudspeaker–room response: (a) the minimum-phase specification of the low-frequency warped IIR filter (thin line) and the filter response (thick line), (b) the minimum-phase specification of the high-frequency warped IIR filter (thin line) and the filter response (thick line), and (c) the complete minimum-phase target response (thin line) and the final filter response (thick line) of the parallel filter using the united pole set. The pole frequencies of the filters are displayed by crosses. The vertical dashed line indicates the crossover frequency $f_\mathbf{c} = 500$ Hz of the two warped IIR designs. The curves are offset for clarity.

audio band in a logarithmic scale. This is displayed in Fig. 7.4 for a sixth-octave smoothed loudspeaker–room response. The low-frequency part of the specification is displayed by a thin line in Fig. 7.4 (a), while the high-frequency part by a thin line in Fig. 7.4 (b). As can be seen in Fig. 7.4, the out-of-band parts of the transfer functions are crossfaded to a constant gain. This is done to assure that the warped IIR filters will not waste any resources (poles) outside their respective bands.

Next, two warped IIR filters are designed, one for the low-frequency and one for the high-frequency target. The λ values are chosen so that the warped filters have the maximal logarithmic resolution (minimal $\Delta f/f$) in the middle of their respective bands by finding such λ values where the minimum of the logarithmic resolution curves Eq. (4.4) is at $f = \sqrt{20 \times 500} = 100$ Hz for the low-frequency band and $f = \sqrt{500 \times 20000} = 3160$ Hz for the high-frequency band, giving $\lambda_{\mathrm{LF}} = 0.986$ and $\lambda_{\mathrm{HF}} = 0.65$, respectively. The corresponding resolution curves are displayed in Fig. 7.5.

The warped filters are designed by the frequency-domain Steiglitz–McBride algorithm [Jackson 2008] based on the prewarped transfer functions. The frequency response of the low-frequency warped IIR filter (filter order is 22) is displayed by thick solid line in Fig. 7.4 (a), while the frequency response of the high-frequency warped IIR filter (filter order is 22) is shown by thick solid line in Fig. 7.4 (b). The pole frequencies of the filters are displayed by crosses, showing that indeed there are no poles outside the respective bands. Once the two warped IIR filters are designed, their poles are dewarped using the corresponding λ_{LF} and λ_{HF} values in

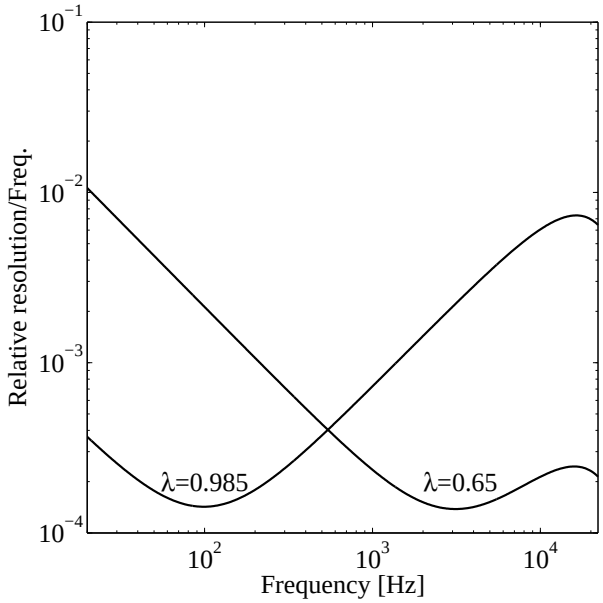


Figure 7.5: The logarithmic frequency resolution for the two bands of multi-band pole positioning: Eq. (4.4) computed with $\Delta f = 1$ for $\lambda_{\mathrm{LF}} = 0.986$ and $\lambda_{\mathrm{HF}} = 0.65$.

Eq. (4.7) and the two pole sets are united, giving a total filter order of 44 (the same as for the stepwise logarithmic case of Fig. 7.2 (c) and as for the ripple-density based pole positioning Fig. 7.3). It can be seen in Fig. 7.4 (c) that the filter response (thick line) follows the smoothed target (thin line) very accurately and the modeling detail is spread evenly in the logarithmic scale.

7.3.4 Pole positioning based on custom warping

So far we have been using the allpass frequency mapping function Eq. (4.2), which gives a limited freedom due to a single parameter λ. Improved results were obtained by combining multiple warped filters with different λ values, but this still does not result in exactly logarithmic frequency resolution. I have developed a custom warping method that can be used with arbitrary warping profiles, including logarithmic [Bank 2011c]. The basic idea is that a warped filter designed based on the logarithmically mapped specification is dewarped using the logarithmic function, and not by the usual allpass substitution.

The method starts with defining a custom frequency mapping function that determines the allocation of frequency resolution. Here a logarithmic frequency transformation is used, leading to logarithmic frequency resolution, but it is emphasized that any other monotonic, smooth function can be used to have a different kind of distribution of the frequency resolution. We need to map the original angular frequencies $\vartheta \in [0, \pi]$ to the warped frequencies $\tilde{\vartheta} = \nu(\vartheta) \in [0, \pi]$ by a continuously differentiable (C^1) function. That is, $\nu(\vartheta)$ and its first derivative $\nu'(\vartheta) = d\nu(\vartheta)/d\vartheta$ should be continuous, because these functions will be later used for pole dewarping. Since we would like to map $\vartheta = 0$ to $\tilde{\vartheta} = 0$, the logarithmic curve of $\nu(\vartheta)$ is exchanged for a linear curve very low frequencies ($\vartheta < \vartheta_c$), as also suggested in [Härmä and Paatero 2001] for their high-order allpass approximation. This basically means that the frequency resolution is constant below ϑ_c, while logarithmic above ϑ_c.

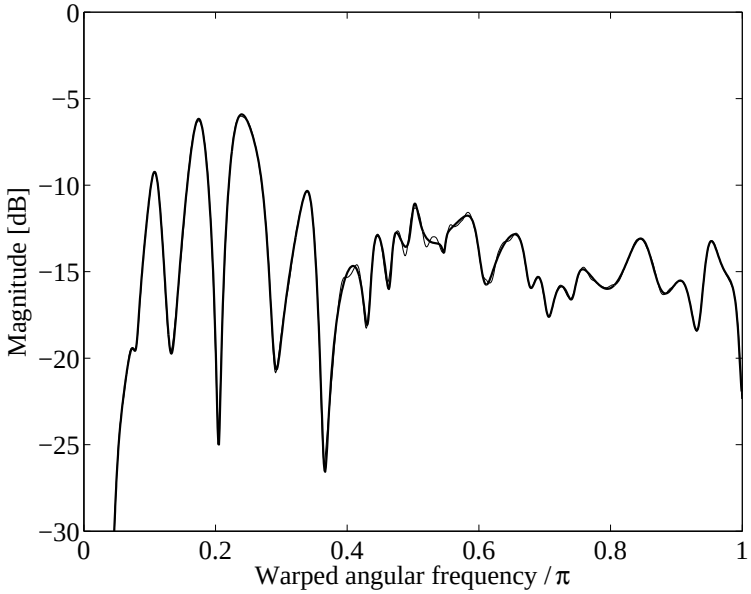


Figure 7.6: Logarithmically warped target specification (sixth-octave smoothed minimum-phase loudspeaker–room response, displayed by thin line) and a 44th order IIR filter designed by `invfreqz` in MATLAB (thick line). Note that the warped angular frequency is displayed in a linear scale.

Then, the filter specification is transformed by this mapping function so that the original specification points $H_\mathrm{t}(\vartheta_n)$ are moved to the frequencies $\tilde{\vartheta}_n = \nu(\vartheta_n)$. Mathematically, this mapping is described by $\tilde{H}_\mathrm{t}(\tilde{\vartheta}_n) = H_\mathrm{t}(\nu^{-1}(\tilde{\vartheta}_n))$. The target specification used for illustrating the design steps is the usual loudspeaker–room response, which is smoothed to a sixth-octave resolution. The logarithmically warped filter specification is displayed in Fig. 7.6 thin line. Note that the warped angular frequency is plotted in a linear scale.

At this step an IIR filter is designed to the warped specification $\tilde{H}_\mathrm{t}(\tilde{\vartheta})$ by any of the traditional filter design methods. Here the `invfreqz` command of MATLAB is used to design a 44th order IIR filter. The resulting response is shown in Fig. 7.6 thick line. The log-warped specification is made minimum-phase prior to filter design: practical experience shows that this helps to guarantee the stability of the designed IIR filter.

Then, the poles of this warped filter $\tilde{H}(\tilde{\vartheta})$ are found and mapped back to the original frequency scale. For complex poles, we first compute the pole frequencies $\tilde{\theta}_k = \varphi\{\tilde{p}_k\}$ and radii $\tilde{R}_k = |\tilde{p}_k|$. The pole frequencies are mapped so that the peaks of the filter get back to the original peak frequencies of the target by using the inverse mapping $\nu^{-1}(\tilde{\vartheta})$, leading to

$$\theta_k = \nu^{-1}(\tilde{\theta}_k). \tag{7.4}$$

We also need to map the pole radii in such a way that the bandwidths of the resonances of the dewarped filter $\Delta\theta_k$ match the bandwidths of the peaks of the target response. If the mapping function can be considered linear at $\tilde{\theta}_k$ in the range of $\Delta\tilde{\theta}_k$ then the new bandwidth after mapping will be proportional to the derivative of the mapping function at $\tilde{\theta}_k$ pole frequency. Accordingly, we apply the inverse mapping $\nu^{-1}(\tilde{\theta}_k)$, so the bandwidth will be

$$\Delta\theta = \nu^{-1'}(\tilde{\theta}_k)\Delta\tilde{\theta}_k, \tag{7.5}$$

where $\nu^{-1'}(\tilde{\vartheta})$ is the derivative of the inverse mapping function $\nu^{-1}(\tilde{\vartheta})$. Similarly to the case of

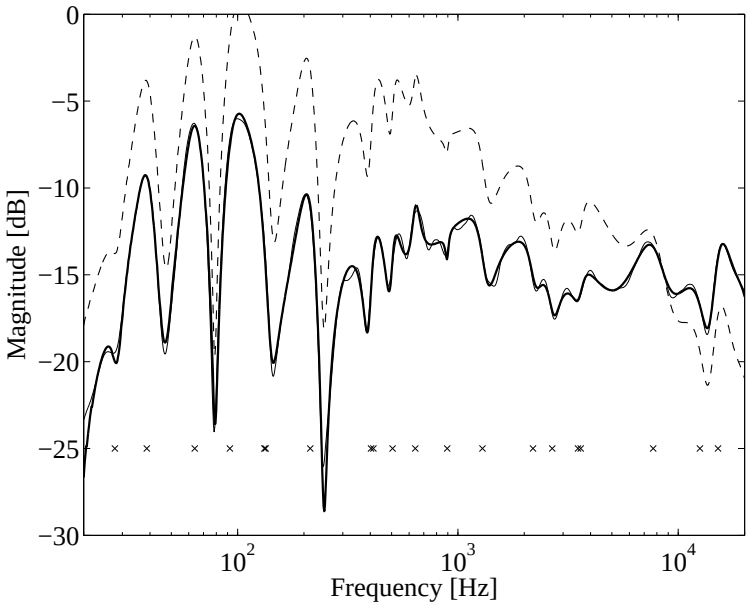


Figure 7.7: Minimum-phase loudspeaker–room response modeling: original specification (thin line), 44th order series second-order filter after pole-zero dewarping (dashed line), and a 44th order parallel second-order filter designed using the dewarped poles (thick solid line). The pole frequencies are displayed by crosses.

predetermined pole positioning, we again use the approximation Eq. (7.1) for the pole radii:

$$\tilde{R}_k = e^{-\frac{\Delta\tilde{\theta}_k}{2}} \Leftrightarrow \ln(\tilde{R}_k) = -\frac{\Delta\tilde{\theta}_k}{2}. \tag{7.6}$$

For the dewarped radii we obtain:

$$R_k = e^{-\frac{\Delta\theta_k}{2}} = e^{-\frac{\Delta\tilde{\theta}_k}{2}\nu^{-1'}(\tilde{\theta}_k)} = e^{\ln(\tilde{R}_k)\nu^{-1'}(\tilde{\theta}_k)} = \tilde{R}_k^{\nu^{-1'}(\tilde{\theta}_k)}, \tag{7.7}$$

that is, the pole radii $\tilde{R}_k$ of the warped filter need to be raised to the to the power of $\nu^{-1'}(\tilde{\theta}_k)$ to obtain the dewarped pole radii R_k. Finally we compose the pole from its angle and radii as $p_k = R_k e^{j\theta_k}$.

The equations can be summarized as follows:

$$\theta_k = \nu^{-1}(\tilde{\theta}_k), \tag{7.8a}$$

$$R_k = \tilde{R}_k^{\nu^{-1'}(\tilde{\theta}_k)}, \tag{7.8b}$$

$$p_k = R_k e^{j\theta_k}. \tag{7.8c}$$

For real poles we compute their frequencies (the -3dB point of their transfer functions) and remap them by $\nu^{-1}(\tilde{\vartheta})$.

While we will use only the poles of the warped filter, it is interesting to stop for a moment and take a look of the response that arises when both the poles and zeros are dewarped and paired based on Eq. (7.8) to form a filter composed of a series of second-order sections. The resulting response is displayed in Fig. 7.7 dashed line, together with the specification (thin solid line) in the original frequency scale. (Note the logarithmic frequency axis in Fig. 7.7 as opposed

to the linear one in Fig. 7.6.) It can be seen that the resulting filter response (dashed line) is tilted compared to the specification (thin solid line). This comes from the inaccuracies of pole-zero remapping: one reason is that in deriving Eq. (7.5) we were assuming relatively narrow pole bandwidths so it is inaccurate for highly damped poles. Additionally, when a pole is dewarped in normal warped filters, a zero arises at λ, while when a zero is dewarped, a pole arises at λ. Since λ is constant in traditional warped filters, these additional poles and zeros cancel out. However, in our case every dewarping corresponds to a different equivalent λ value, and the effects of these not-implemented poles and zeros accumulate and bias the response.

However, we are not intending to implement the filter as a cascade structure, but only use the dewarped poles for fixed-pole parallel filter design. The result is displayed in Fig. 7.7 thick line. It can be seen that now the filter response matches the target specification quite precisely because the LS optimization of the zeros of the transfer function compensates the inaccuracies of pole dewarping.

7.4 Pole positioning comparison

Here the various pole positioning methods presented in this chapter will be compared on three different loudspeaker equalization applications (one in-room and two anechoic measurements). In the detailed explanation of the methods a modeling example was used since it made explaining the techniques easier. Nevertheless, they can all be used for equalizer design as well. In equalizer design the goal is to estimate a filter which, when applied to the system response, equalizes the transfer function so that it matches the target response (see Sec. 2.2). Since the target is usually a flat response within the corner frequencies of the loudspeaker, displaying the equalized transfer functions will give us a clearer picture on the distribution of the approximation error, something much harder to see in the modeling case.

7.4.1 Loudspeaker–room response equalization

Figure 7.8 (a) thick line shows the sixth-octave smoothed version of a loudspeaker–room response. The target frequency response we wish to achieve by equalization is a forth-order high-pass response with a corner frequency of 30 Hz, displayed by a thin line. In all the following examples the same target curve is used, and the parallel filter is designed by the frequency-domain direct equalizer design method presented in Sec. 6.3.2. However, they will differ in how the poles are obtained.

Figure 7.8 (b) shows the equalization using a predetermined pole set with stepwise logarithmic pole positioning. The total number of pole pairs is 20, that is, the filter has 20 second-order sections (the filter order is 40). The method gives an acceptable performance since the ripples of the equalized transfer function are within ± 1 dB around the target response. There are only a few sharp dips at 80 Hz and 250 Hz that are not equalized: for some applications, such as in room response correction this is actually an advantage since these sharp dips are usually position dependent, and it is generally accepted that they should not be equalized [Craven and Gerzon 1992; Cecchi et al. 2018]. This is because that would require a sharp peak in an equalizer, which, at other positions in the room where the dip is not counteracting it, actually produces a disturbing ringing sound. Failing to equalize these dips is coming from the “automatic” smoothing behavior of the parallel filter with a predetermined pole set demonstrated in Sec. 6.5. Since the modeling resolution is fixed, the equalizer cannot counteract anything narrower than the

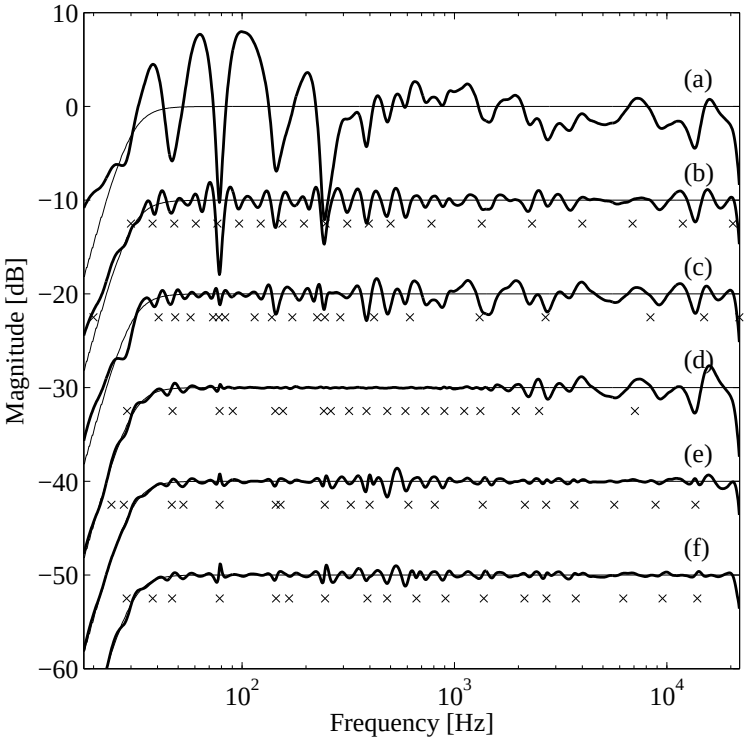


Figure 7.8: Comparison of different pole positioning techniques for minimum-phase loudspeaker–room response equalization. The sixth-octave smoothed room response is displayed in (a). The pole positioning techniques used are (b) predetermined pole set, pole positioning based on (c) ripple density, (d) standard warped IIR filter design, (e) multi-band warping, and (f) custom warping. The target response is shown by thin lines. The pole frequencies are displayed by crosses. The number of pole-pairs (or, second-order sections) is 20 in all cases, giving a total filter order 40. The curves are offset for clarity.

desired resolution. Note that for the rest of the pole positioning methods proposed here if we wish to avoid the equalization of sharp dips, they need to be removed from the system response manually or by an appropriate automatic limiting process [Craven and Gerzon 1992; Pedersen and Thomsen 2007]. Also note that now the parallel filter was designed using the sixth-octave smoothed response for making the comparison with other pole positioning approaches more coherent. However, for cost-effective applications the predetermined pole set has an additional advantage besides its simplicity that it does not require the smoothing of the measured transfer function as it performs smoothing “automatically” based on its pole density.

The equalization using a ripple-density based pole set (Sec. 7.3.1) is shown in Fig. 7.8 (c), leading to a similar performance to the predetermined pole set in this case, but now without the need for deciding in which regions the density of the poles should be larger. An important difference is that now the narrow dips of the response are also equalized by placing more poles in that region (see Fig. 7.8 (c) around 80 Hz). As already discussed, equalizing the sharp dips of loudspeaker–room responses is not recommended, and this can be avoided by removing such dips from the system response in a preprocessing step as suggested in [Craven and Gerzon 1992; Pedersen and Thomsen 2007]. Nevertheless, if we consider the method as a general equalization technique, it is actually an advantage that we may counteract the sharp dips if we wish so.

In the example of Fig. 7.8 (d), a 40th order IIR filter is identified by the frequency-domain Steiglitz-McBride method [Jackson 2008] based on the warped version of the smoothed system

Table 7.2: Errors of equalization examples presented in Figs. 7.8, 7.9, and 7.10. The order of the methods correspond to those in the figures: no equalizaztion (a), predetermined pole set (b), ripple-density based pole positioning (c), poles obtained from a warped IIR design (d), from a dual-band warped design (e), and via custom warping (f). The smallest errors are marked with bold.

Room eq. Fig. 7.8	No EQ (a)	Predet. (b)	Ripple d. (c)	Warping (d)	Dual-bd. (e)	Custom wp. (f)
Mean squared error	0.482	0.122	0.116	0.059	0.043	**0.040**
Mean abs. dB error	2.506 dB	0.691 dB	0.646 dB	0.324 dB	0.238 dB	**0.215 dB**
Loudsp. eq. Fig. 7.9	No EQ (a)	Predet. (b)	Ripple d. (c)	Warping (d)	Dual-bd. (e)	Custom wp. (f)
Mean squared error	0.322	0.052	0.054	0.038	**0.022**	0.027
Mean abs. dB error	1.033 dB	0.291 dB	0.304 dB	0.211 dB	**0.125 dB**	0.147 dB
Loudsp. eq. Fig. 7.10	No EQ (a)	Predet. (b)	Ripple d. (c)	Warping (d)	Dual-bd. (e)	Custom wp. (f)
Mean squared error	0.419	0.087	0.054	0.028	0.028	**0.023**
Mean abs. dB error	1.993 dB	0.489 dB	0.301 dB	0.156 dB	0.159 dB	**0.132 dB**

response as the input and the warped version of the target as the output. In this example, a warping parameter $\lambda = 0.95$ is used since that λ resulted in the smallest error. It can be seen in Fig. 7.8 (d) that the warping-based pole positioning provides more accurate equalization compared to the first two methods, and its only shortcoming that the accuracy is concentrated in the middle frequency range.

In Fig. 7.8 (e) dual-band warping is applied. The split frequency is 500 Hz. Similarly to Sec. 7.3.3, $\lambda_{\mathrm{LF}} = 0.986$ and $\lambda_{\mathrm{HF}} = 0.65$ are used. The filter orders are 20 in both the low and high bands, giving a total filter order of 40. Figure 7.8 (e) shows that the frequency resolution is now spread much more evenly on the logarithmic scale and thus an excellent equalization performance is achieved for the same filter order as with the previous methods. The only larger ripples that can be seen are in the transition region of the two bands (around 500 Hz), the rest of the ripples are within ± 0.5 dB of the target, which can be considered negligible. The price to pay compared to the simple warping of Sec. 7.1 is the additional task of separating the response to different regions and the need of designing multiple filters. However, the total order of filters remains the same, so the computational complexity of the design is not increased significantly. A small shortcoming is that the user has to manually balance between the number of poles used in the two frequency bands.

The equalization using poles obtained by custom warping is displayed in Fig. 7.8 (f). The cutoff frequency where the linear mapping is changed to logarithmic is set to 50 Hz. The performance is very similar to the multi-band warping case, the only difference is that the ripples are smaller at 500 Hz. The computational complexity of the design is similar to the other two warped designs. A benefit compared to the multi-band warping is that only the total filter order has to be given by the user, and not the orders in the different bands as for multi-band warping.

The equalization errors for the five different pole positioning methods can be compared in Table 7.2. The rows with "Mean squared error" are computed as the square root of the mean squared difference of the equalized and the target frequency responses, thus, they show the error of the complex transfer function. This kind of error measure is useful since it also shows that the phase is correctly approximated (not shown in the figures), and it is directly related to the error minimized during filter design. The averaging is done between the corner frequencies of the target response (30 Hz and 20 kHz in this case), and over a logarithmic frequency grid with

100 points per octave resolution.

To show a different kind of error sometimes also used in the audio field, the second rows display the “Mean abs. dB error”: here the magnitudes of the equalized and target responses are computed in dB, and their absolute differences are averaged on a logarithmic frequency grid, as suggested in [Ramos and López 2006]. This kind of error is more related to what we actually see in Fig. 7.8 since that is in dB scale as well.

It can be seen in Table 7.2 that the two kinds of errors are strongly correlated: if a certain method is better than an other with respect to “Mean squared error” it is also better in “Mean abs. dB error”. For the loudspeaker–room response equalization example of Fig. 7.8, the dual-band warping (e) and custom-warping (f) has the lowest errors in both measures, the latter performing slightly better.

7.4.2 Loudspeaker equalization

Here additional equalization examples are presented to give a broader view on the behavior of the different methods. First the anechoic response of the same two-way floorstanding loudspeaker is equalized which was used in a room in the previous example. Now the measured anechoic frequency response is smoothed to 12th-octave resolution. This is shown in Fig. 7.9 (a) thick line. The target response is a forth-order highpass with a cutoff frequency of 50 Hz.

For the predetermined pole set, while 12th-octave resolution would require higher pole density to perfectly equalize the response (six pole pairs per octave actually leads to 51 pole pairs), we use 20 pole pairs between 50 Hz and 20 kHz, since for the more sophisticated methods this filter order is sufficient to obtain practically perfect equalization, as we shall see later. The results are acceptable, the ripples are within ± 1 dB, as can be seen in Fig. 7.9 (b). The ripple-density based pole positioning leads to similar results as shown in Fig. 7.9 (c).

For this example $\lambda = 0.85$ gives the best results for warping-based pole positioning, shown in Fig. 7.9 (d). Similarly to Fig. 7.8 (d), the accuracy is concentrated to the middle range, while the band edges are less accurately equalized. This is again overcome by the multi-band (Fig. 7.9 (e)) and custom warping (Fig. 7.9 (f)). The better performance of these two methods manifests also in their smaller error values in Table 7.2.

The next example is a two-way bookshelf speaker having some distinct spikes in its frequency response. The response is smoothed to 12th-octave resolution prior to equalizer design, displayed in Fig. 7.10 (a). The target is the series connection of a forth-order highpass with a cutoff frequency of 70 Hz and a second-order lowpass with a corner frequency of 18 kHz.

First, a predetermined pole set is used with 20 pole pairs logarithmically distributed between 70 Hz and 20 kHz. The equalizer has a poor performance as seen in Fig. 7.10 (b), because the pole density is too low in the problematic regions. On the other hand, the ripple-density based pole positioning can show its power: equalization improves significantly, since now the poles are concentrated in the ragged parts of the frequency response (see Fig. 7.10 (c)).

In this case $\lambda = 0.8$ is used for warping-based pole positioning in Fig. 7.10 (d). On the contrary to the previous examples, even simple warping can provide almost perfect equalization (with the exception of a small discrepancy around 5 kHz). This is because the design bandwidth is smaller (note in Fig. 7.10 (a) that the response has ripples only above 400 Hz), thus, we do not face with the problems we had at the band edges for Fig. 7.8 (d) and Fig. 7.9 (d).

Since we are not using the full audio band, for the multiband warping of Fig. 7.10 (e) a crossover frequency of of 2 kHz and lambdas of $\lambda_{\mathrm{LF}} = 0.8$ and $\lambda_{\mathrm{HF}} = 0.3$ are used. Note however that this does not decrease the error of the equalization (see Tab. 7.2). For the custom

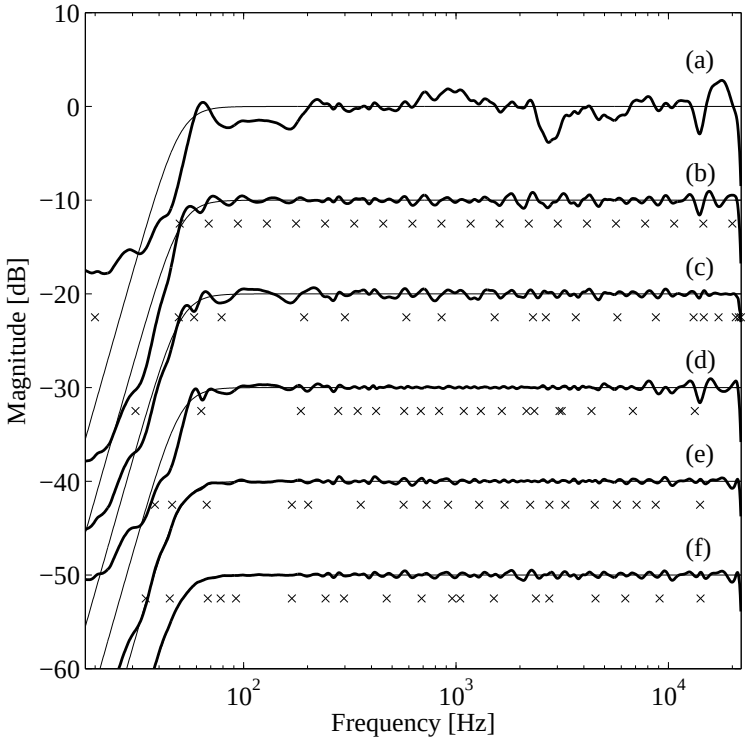


Figure 7.9: Comparison of different pole positioning techniques for a minimum-phase loudspeaker response equalization. The 12th-octave smoothed anechoic loudspeaker response of a two-way loudspeaker floorstanding is displayed in (a). The pole positioning techniques used are (b) predetermined pole set, pole positioning based on (c) ripple density, (d) standard warped IIR filter design, (e) multi-band warping, and (f) custom warping. The target response is shown by thin lines. The pole frequencies are displayed by crosses. The number of pole-pairs (or, second-order sections) is 20 in all cases, giving a total filter order 40.

warping method, the best results are obtained when ϑ_c is set so that the resolution changes from constant to logarithmic at 1 kHz. While custom warping (f) has the lowest equalization error according to Tab. 7.2, the relatively minor improvement over the results of single-band warping might not actually worth the additional design complexity.

7.5 Discussion of pole positioning techniques

After the discussion of the pole positioning strategies for parallel filter design and seeing a couple of examples, some conclusions can be drawn about the benefits of the various methods.

Predetermined pole set: This is the simplest method, which works relatively well for modeling or equalizing high-order systems (such as loudspeaker–room response) with a lower order filter, since in this case there are so many peaks and valleys in the original response that waviness of smoothed response is determined by the smoothing itself. However, when the raggedness is unevenly distributed in the response, such as in Fig. 7.10 (a), a logarithmic pole positioning is not adequate, and regions with different pole density should be selected manually, which is sometimes not practical, e.g., in the case of an automatic equalizer system. Note however that in certain design applications the possibility of manual intervention can be beneficial, e.g., an

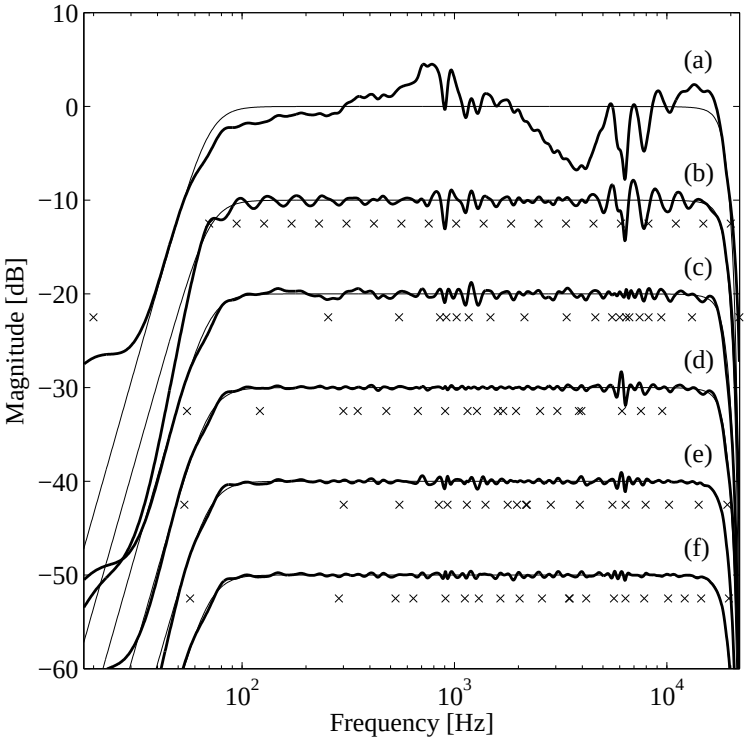


Figure 7.10: Comparison of different pole positioning techniques for minimum-phase loudspeaker response equalization. The 12th-octave smoothed anechoic response of a bookshelf loudspeaker is displayed in (a). The pole positioning techniques used are (b) predetermined pole set, pole positioning based on (c) ripple density, (d) standard warped IIR filter design, (e) multi-band warping, and (f) custom warping. The target response is shown by thin lines. The pole frequencies are displayed by crosses. The number of pole-pairs (or, second-order sections) is 20 in all cases, giving a total filter order 40.

expert user can manually fine-tune the frequency resolution by moving around the poles and listening to the result (the interactive demo `parfeqdemo` downloadable from [Parallel filter homepage 2021] allows the manual positioning of the poles and designs the equalizer immediately). Bedsides its simplicity, an additional advantage of the predetermined pole set is the automatic smoothing behavior: there is actually no need for transfer function smoothing prior to equalizer design, so the equalizer can be obtained from the measured response directly.

Ripple-density based pole positioning: This method, while simple enough, works robustly in all cases since it allocates the frequency resolution in accordance with the raggedness of the response. It shows its main benefits compared to the predetermined pole set when the response has some specific problematic regions, as in Fig. 7.10 (c).

Warping-based pole positioning: Since this method is based on a warped IIR filter design, it is able to actually take into account both the frequencies and Q factors of the peaks and valleys of the smoothed response. When the system is not using the full audio bandwidth (such as in small bookshelf or multimedia speakers, see Fig. 7.10 (d)) it provides almost perfect equalization. Systems having full audio bandwidth will suffer from low- and high-frequency inaccuracies when using this pole positioning method (see Figs. 7.8 (d) and 7.9 (d)).

Pole positioning using multi-band warping: This method overcomes the bandwidth limitation of straightforward warping, thus, it can be used for the full audio bandwidth from 20 Hz to 20 kHz with excellent results. The only slight shortcoming is that the user has to balance

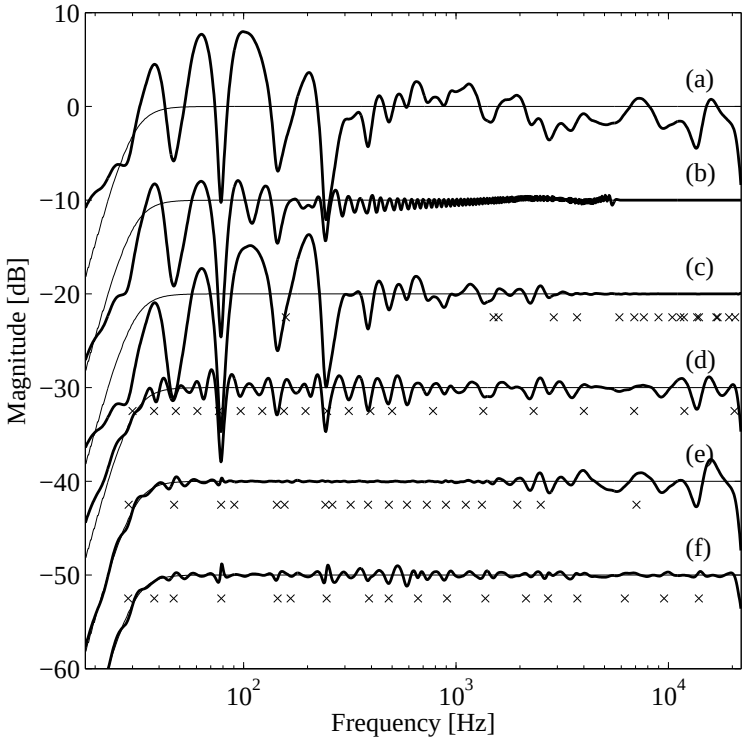


Figure 7.11: Comparison of different filter design methods for loudspeaker–room response equalization. The smoothed room response is displayed in (a). The response equalized by an 1000th-order FIR filter (b), a 40th-order standard IIR filter (c), a 40th-order parallel filter with stepwise logarithmic pole frequencies (d), a 40th-order warped IIR filter using $\lambda = 0.95$ (e), and a 40th-order parallel with poles obtained by custom warping (f). The target response is shown by thin lines. The pole frequencies are displayed by crosses.

between the filter orders of the two frequency bands by trial and error.

Pole positioning using custom warping: This method provides excellent equalization in the full audio frequency range similarly to multi-band warping. A benefit over multi-band warping is that only the total filter order has to be specified by the user.

7.6 Comparison with previous filter design methods

While we have discussed in Chap. 1 that standard FIR and IIR filter design techniques are not well suited to filter- or equalizer design on a logarithmic frequency scale, it is instructive to see how they actually perform on the same loudspeaker-room equalization example as of Fig. 7.8.

The first example is frequency-domain least squares equalization by a 1000th-order FIR filter, shown in Fig. 7.11 (b): it can be seen that even such a long FIR filter cannot provide enough resolution at low frequencies to equalize the response below 200 Hz. Next, a 40th-order IIR filter is designed by the frequency-domain Steiglitz-McBride method [Jackson 2008] where the specification is given on a logarithmic frequency grid so that the error should be minimized on the logarithmic scale. However, as can be seen in Fig. 7.11 (c), the equalization is unusable below 300 Hz, which is also expected from the pole frequencies of the filter, shown by crosses.

On the contrary, a fixed-pole parallel filter requiring the same computational resources pro-

Table 7.3: Errors of equalization examples presented in Fig. 7.11. The marking of the methods correspond to those in the figure: 1000th-order FIR filter (b), 40th-order IIR filter (c), 40th-order parallel filter with predetermined poles (d), 40th-order warped IIR design (e), 40th-order parallel filter with custom warping (f).

Room eq. Fig. 7.11	FIR 1000 (b)	IIR (c)	par. log. (d)	warped IIR (e)	par. custom wp. (f)
Mean squared error	0.154	0.288	0.122	0.059	0.040
Mean abs. dB error	0.990 dB	1.725 dB	0.691 dB	0.324 dB	0.215 dB

vides a significantly better performance even with a predetermined set of poles, as shown in Fig. 7.11 (d). It follows that compared to straightforward FIR or IIR filter design techniques, even the simplest form of parallel filter design (logarithmically positioned poles) provides a drastically improved performance when the accuracy is evaluated on a logarithmic frequency scale. The equalization error is also displayed in Table 7.3.

Let us also compare the fixed-pole parallel filter to logarithmic frequency resolution filter design methods that were the state of the art at the time of the publication of my results. Fig. 7.11 (e) shows a 40th-order warped IIR equalization example designed with a warping parameter $\lambda = 0.95$. The response of a Kautz filter with warping-based pole positioning would be also the same, so it serves as a good comparison with Kautz filters as well. When using these poles, parallel filters result in the same response as Fig. 7.11 (e), but with lower computational complexity compared to both warped and Kautz filters and a completely parallel filter structure. This means that at this point parallel filter design is already more efficient in terms of accuracy/complexity ratio.

This efficiency is improved even more by the more sophisticated pole positioning methods I have developed. For example, Fig. 7.11 (f) demonstrates that by using custom warping a significantly improved equalization is achieved compared to the previous state of the art for the same filter order.

In the audio signal processing field filter design algorithms are usually compared based on real-world (measured) system responses as was done in Fig. 7.11. However, publications about general filter design techniques often use synthetic example cases. The reader is invited to take a look at such an example in Appendix A.5 where the filters are designed based on a target which looks like a square wave in the logarithmic frequency scale. While this kind of a target is very different from that of a loudspeaker–room response, the same conclusions can be drawn from the results. This is also true for the other synthetic and real-world examples that I have tested, confirming the superiority of the proposed methods compared to earlier techniques.

It also worth mentioning that Maestre et al. [2016] have further developed the pole positioning method based on warped IIR design by using an iterative procedure where the poles obtained from a single-band warped IIR design using the Prony or the Steiglitz-McBride technique are post-optimized with a gradient descent algorithm. Compared to the warped Steiglitz-McBride technique of Sec. 7.1, the improvement is relatively minor, as can be seen in Fig. 4 of [Maestre et al. 2016]. In my opinion, such an improvement might not worth the additional complexity of a complicated optimization algorithm. Nevertheless, for aiming at the best possible modeling accuracy, initializing the post-optimization technique of Maestre et al. [2016] with the poles obtained using multi-band or custom warping methods proposed in Secs. 7.3.3 and 7.3.4 could be a promising topic for future research.

7.7 Scientific contributions

Statement 2: I have developed new pole positioning methods that can be used not only for the fixed-pole design of parallel filters, but also for Kautz filter design, and lead to more accurate approximation compared to earlier methods developed for Kautz filters. As a result, the proposed methodology leads to better filter approximation on the logarithmic scale compared to earlier techniques (traditional IIR, warped and Kautz filters) for the same filter order.

2.1: As an efficient alternative to manual pole positioning, I have developed a technique based on the "ripple density" of the target response. The method sets the pole frequencies according to the ripples of the target, and thus results in more poles and better resolution in the problematic regions of the transfer function. I have also given the formulas for the pole radii that can be used to compute the complex poles from an arbitrary set of pole frequencies.

2.2: I have developed a dual-band warping-based pole positioning method where the warped IIR filters are designed with different λ values for the lower and upper bands. This results in a frequency resolution closer to logarithmic, which manifests in a smaller approximation error computed on the logarithmic scale compared to straightforward, single-band warping.

2.3: I have developed a "custom warping" method where the warped IIR filter is designed using an arbitrary (smooth and monotonic) frequency mapping, as opposed to the allpass transform of traditional warped filters. The poles obtained by such a design are used as the poles of the parallel filter, and the method – when a logarithmic mapping is used – results in a smaller approximation error on the logarithmic scale compared to single-band warping.

The results have been published in two journal papers [Bank 2013a] and [Bank and Ramos 2011][1] and in two conference papers [Bank 2011c, 2013c].

[1] While the paper was published together with Prof. German Ramos, the technique was developed by myself. Prof. Ramos has helped in the comparison with his earlier technique [Ramos and López 2006] and in the preparation of the manuscript.

Chapter 8

MIMO extensions to the fixed-pole design of parallel filters

After developing single-channel algorithms, a natural choice of my research direction was to provide their multichannel extensions. Multiple-input multiple-output (MIMO) filtering is used in various fields of signal processing. In audio the most straightforward example is modeling or equalizing a room having multiple listeners and multiple loudspeakers. Modern car audio systems are also designed in such a way that all passengers have optimal listening experience, and do so by applying a large number (typically between 10 and 20) of loudspeakers. As a two-by-two example, binaural crosstalk cancellation aims at independently controlling the signals reaching the left and right ears of the listener by the use of two loudspeakers. The directivity of loudspeakers containing multiple speaker elements can be also controlled by a MIMO filter, and the same is true for microphone arrays. In sound synthesis, both the direction-dependent radiation pattern of a musical instrument and the mechanical admittance of its various parts can be described by MIMO filters.

The relatively trivial extensions of the single channel case will be outlined in Sec. 8.1 that will also show how the computational complexity can be decreased by using a common pole set, a choice often made in modeling acoustic systems.

In Sec. 8.2 the idea of parallel filters will be applied to admittance matrix modeling: besides being a MIMO problem, here the passivity of the transfer function must also be guaranteed since we aim to model a passive physical system. When such an admittance model is coupled to other structures (such as other parts of a musical instrument model), failing to guarantee passivity may actually lead to the spurious generation of energy and to the instability of the complete model. Besides an efficient passive design formulation, the estimation of common poles using a warped common-denominator all-pole model will also be presented that can be used for other applications where a common pole set can be motivated by physical or perceptual considerations. At the time of the publication of the results [Bank and Karjalainen 2008, 2010], this was the first technique that allowed the design of admittance filters for the full audio frequency range while guaranteeing passivity. Later my technique has been further developed by Maestre et al. [2015, 2017] that provide improved modeling accuracy at the price of applying a more complicated optimization procedure.

Finally, Sec. 8.3 provides the extension of the direct equalizer design approach of Sec. 6.3 to the MIMO case. While MIMO parallel filter design is trivial in the sense that the elements of the filter matrix can be estimated in the same way as for the single channel case, this is not true for equalizer design. Therefore I have developed a matrix formulation for the equalization

case, which shows that by taking advantage of fixed poles, the design of an IIR equalizer can be equally simple as that of the commonly used FIR equalizer and a closed-form solution can be obtained when the problem is formulated in the least-squares sense. The importance of the results lies in the fact that for MIMO equalization almost always FIR filters have been employed in the literature probably due to the fact that extending IIR filter design techniques to the MIMO case is non-trivial and the implementation of iterative methods can be problematic for the high number of parameters that needs to be optimized for MIMO IIR filters. However, these difficulties are now removed since the numerators of parallel filters can be estimated by a linear LS fit similarly as for FIR filters, so the methods developed for MIMO FIR equalizer design can be easily adapted to the parallel filter case.

8.1 MIMO parallel filters

Let the input of the MIMO filter be a column vector $\mathbf{x}(z) = [x_1(z), \ldots, x_I(z)]^T$ containing all the entries corresponding to the I input channels, and the output a column vector $\mathbf{y}(z) = [y_1(z), \ldots, y_O(z)]^T$ with the entries corresponding to the O output channels. Thus, the MIMO filtering can be written as

$$\mathbf{y}(z) = \mathbf{H}(z)\mathbf{x}(z), \tag{8.1}$$

with the transfer function $\mathbf{H}(z)$ containing the elements $H_{o,i}(z)$. When the separate elements are independent parallel filters, the transfer function becomes

$$H_{o,i}(z) = \sum_{k=1}^{K^{(o,i)}} \frac{b_{k,0}^{(o,i)} + b_{k,1}^{(o,i)} z^{-1}}{1 + a_{k,1}^{(o,i)} z^{-1} + a_{k,2}^{(o,i)} z^{-2}} + \sum_{m=0}^{M^{(o,i)}} f_m^{(o,i)} z^{-m}, \tag{8.2}$$

where i is the index of the input channel and o is the index of the output channel of the MIMO filter. The design of such a MIMO parallel filter is done by estimating separate SISO parallel filters for each element of the target matrix $\mathbf{H}_\mathrm{t}(z)$, thus, it does not lead to any additional complications compared to the single channel case outlined in Sec. 6.2.

8.1.1 Common-pole parallel filters

In comparison to the general form Eq. (8.2), significant computational savings can be achieved when the pole set (and thus the filter order) is the same across the entries of $\mathbf{H}(z)$, since this allows sharing the denominator parts across the different channels.

Common-denominator IIR filters are often used for modeling acoustic transfer functions [Haneda and Kaneda 1994; Liu and Hsieh 1988]. This is motivated by the fact that for a single physical system that has multiple input and output points (like multiple loudspeakers and microphones in a room, or multiple force inputs of a mechanical system) the poles of all transfer functions are the same by theory since they correspond to the vibrating modes of the same system.

In addition to the physical reasons, often we would like to model or equalize the elements of the transfer function matrix with the same frequency resolution, again leading to the same pole set when the predetermined pole positioning of Sec. 7.2 is used.

One option is to use the same pole set, and thus the same denominators for all the outputs

	Multiplications
General case	$4OIK$
Common denominators for a specific input	$2(O+1)IK$
Common denominators for a specific output	$2O(I+1)K$

Table 8.1: Number of multiplications and additions required for a MIMO parallel filter having I inputs, O outputs, and K second-order sections in each element of the filter matrix.

coming from a specific input channel i, leading to

$$H_{o,i}(z) = \sum_{k=1}^{K^{(i)}} \frac{b_{k,0}^{(o,i)} + b_{k,1}^{(o,i)} z^{-1}}{1 + a_{k,1}^{(i)} z^{-1} + a_{k,2}^{(i)} z^{-2}} + \sum_{m=0}^{M^{(i)}} f_m^{(o,i)} z^{-m}. \tag{8.3}$$

This leads to the reduction of required numerical operations because now it is possible to filter a particular input i with the denominator parts $1/(1 + a_{k,1}^{(i)} z^{-1} + a_{k,2}^{(i)} z^{-2})$ of the K sections only once, and use this common set of filtered signals as the inputs of the numerator parts $b_{k,0}^{(o,i)} + b_{k,1}^{(o,i)} z^{-1}$ that are different for the various outputs. A special case of this formulation is the single-input multiple-output case, where only one set of denominator parts are needed.

Another option is to use the same pole set, and thus the same denominators for all the inputs going to a specific output channel o, leading to

$$H_{o,i}(z) = \sum_{k=1}^{K^{(o)}} \frac{b_{k,0}^{(o,i)} + b_{k,1}^{(o,i)} z^{-1}}{1 + a_{k,1}^{(o)} z^{-1} + a_{k,2}^{(o)} z^{-2}} + \sum_{m=0}^{M^{(o)}} f_m^{(o,i)} z^{-m}. \tag{8.4}$$

In this case the computational complexity is reduced compared to the general form of (8.2) because now it is possible to first compute the numerator parts $b_{k,0}^{(o,i)} + b_{k,1}^{(o,i)} z^{-1}$ of the second-order filters, sum those that correspond to the same output channel o, and then filter with the common denominators $1/(1 + a_{k,1}^{(o)} z^{-1} + a_{k,2}^{(o)} z^{-2})$. A special case is the multiple-input single-output scenario, where only one set of denominators are implemented.

Of course it is also possible to use the same pole set for all the elements of $\mathbf{H}(z)$, however, this provides no computational advantage over the two cases outlined so far. The number of multiplications required by running a MIMO parallel filter having K second-order sections, I inputs, O outputs are shown in Table 8.1, without the optional FIR part (for the FIR part no savings can be achieved). It can be seen that if either the number of input- or output channels is significantly larger than 1, the computational complexity can be reduced roughly to the half compared to the general case by using the shared denominator approach.

All the pole positioning techniques presented in Chap. 7 can be extended for finding common-poles in a straightforward manner. As for the predetermined pole set of Sec. 7.2, no changes are necessary: we simply prescribe the same resolution, thus, the same pole frequencies for all the transfer function paths, and this will result in the same pole set. For pole positioning based on ripple density described in Sec. 7.3.1, the ripple densities computed for the individual transfer functions are simply averaged before numerical integration and segmentation. The rest of the techniques discussed in Chap. 7 are all based on a (typically warped) IIR filter design and using the poles of the IIR filter. A common-pole extension to these techniques includes finding a common-denominator IIR filter based on the set of impulse responses or transfer functions.

Common-denominator IIR models are often used in acoustics [Haneda and Kaneda 1994; Liu and Hsieh 1988], and the related technique for estimating common poles will be outlined in Sec. 8.2.4 for the case of passive admittance matrix modeling.

8.2 Passive admittance matrix modeling using fixed-pole parallel filters

In physics-based sound synthesis, the sound of an instrument is generated by modeling the instrument behavior rather than modeling the sound itself. Therefore the model blocks correspond to the main parts of the instrument (for an overview, see [Välimäki et al. 2006]). Depending on the modeling paradigm, these models can be parameterized in various ways.

For example, it is possible to parameterize parts of the instrument model by a measured mechanical or acoustical immittance (impedance or admittance). As an example, the effect of an immittance (e.g., the instrument bridge) connected to a string is that it changes the modal frequencies and decay times of the string compared to a rigid termination, and provides coupling between the horizontal, vertical, and longitudinal polarizations of the string. Note that the application used here is a mechanical admittance, but the treatment is equally applicable to other passive (e.g., acoustical, electrical) systems and to impedances instead of admittances.

The starting point of such a parameterization is a mechanical admittance measurement of the given part of the instrument (e.g., the bridge). Naturally, all parts of acoustical instruments are passive, that is, they can only dissipate energy that is introduced by the player. Therefore, instead of straightforward filter design that may result in non-passive filters due to measurement or approximation errors, we need such a design technique where the passivity of the result is guaranteed. In [Smith 2010], passive admittance filters are constructed by manually tuning the modal frequencies and decay times of second-order resonators to produce a function similar to the guitar admittance, and a similarly simplified guitar bridge model is presented in [Evangelista and Raspaud 2009] by connecting the passive admittance to a scattering junction. In [Lambourg and Chaigne 1993], the 2D mechanical admittance of a guitar bridge up to 3 kHz is modeled by a set of mass-spring-damper elements (second-order resonators), and the matrix pencil method is used for parameter estimation. In the frequency-domain guitar model of [Woodhouse 2004], a standard modal analysis technique (circle fitting) is used up to 1.4 kHz, and above that a random number generator is applied to produce a statistically similar modal behavior as in the measured response.

Accordingly, at the time of my research, no methods existed that could fit a discrete-time passive admittance filter to measurement data in the full audio frequency range. To overcome this limitation, in [Bank and Karjalainen 2008] I have proposed an admittance filter design method using a modification of the fixed-pole design of parallel second-order filters where the admittance transfer function is constructed as a weighted sum of passive (positive real) second-order transfer functions. In [Bank and Karjalainen 2010] I have extended the method to the modeling of admittance matrices, and this multidimensional case will be presented in the following.

8.2.1 Passivity and positive realness

A system is passive if it cannot produce energy, and for passive systems, immittances are positive real (PR) [Anderson and Vongpanitlerd 1973]. For rational functions of s that do not have

a pole on the closed right-half plane (that is, for asymptotically stable systems), the transfer function matrix $\mathbf{H}(s)$ is PR if and only if

$$\mathrm{Re}\left\{\mathbf{H}(j\omega)\right\} = \frac{1}{2}\left(\mathbf{H}(j\omega) + \overline{\mathbf{H}}(j\omega)\right) \geq 0 \tag{8.5}$$

for all real ω [Anderson and Vongpanitlerd 1973]. Here $\overline{\mathbf{A}}$ means complex conjugate of $\mathbf{A}$, and $\mathbf{A} \geq 0$ means that $\mathbf{A}$ is positive semidefinite.

The PR condition for a digital transfer function $\mathbf{H}(z) = \mathbf{H}(e^{-j\vartheta})$ in a rational form with poles in the open unit disk (asymptotically stable systems) is similar to that for the continuous case [Jury 1974]:

$$\mathrm{Re}\left\{\mathbf{H}(e^{-j\vartheta})\right\} = \frac{1}{2}\left(\mathbf{H}(e^{-j\vartheta}) + \overline{\mathbf{H}}(e^{-j\vartheta})\right) \geq 0. \tag{8.6}$$

That is, it is enough to check positive realness on the unit circle, by looking at the frequency response.

8.2.2 Modal framework

First, let us define the admittance matrix $\mathbf{Y}$:

$$\mathbf{v}(\omega) = \mathbf{Y}(\omega)\mathbf{f}(\omega) \tag{8.7}$$

where $\mathbf{f} = [F_1, \ldots, F_I]^T$ is a column vector composed of the forces exciting the structure at positions $1, \ldots, K$, and $\mathbf{v} = [v_1, \ldots, v_I]^T$ is a column vector composed of the velocities of the points $1, \ldots, K$.

In modal analysis, the general assumption is that the structure can be described as a set of masses that are connected by linear springs and linear dampers [Ewins 1992]. Then, the vibration of the structure can be decomposed to a sum of K normal modes with different modal frequencies ω_k, decay rates σ_k and modal shapes $\mathbf{\Phi}_k$. It is a common assumption in modal analysis that the damping is viscous and it is distributed proportionally to the mass and stiffness elements, referred as proportional damping in the literature. In this case the modal shapes $\mathbf{\Phi}_k$ are real and the mechanical admittance (mobility) matrix of the system can be written as

$$\mathbf{Y}(\omega) = \sum_{k=1}^{K} \mathbf{\Phi}_k^T \mathbf{\Phi}_k \frac{j\omega}{m_k(\omega_k^2 - \omega^2 + 2j\sigma_k\omega_k\omega)}, \tag{8.8}$$

where m_k is the effective mass of mode k, and $\mathbf{\Phi}_k^T\mathbf{\Phi}_k$ is a rank 1 size I square matrix which is positive semidefinite, since the elements of $\mathbf{\Phi}_k$ are real numbers [Marshall 1984]. The scalar transfer functions in Eq. (8.8) are PR because their phase span from $-\pi/2$ to $\pi/2$. Thus, the real part of $\mathbf{Y}$ will be positive semidefinite for all ω frequencies, since $\mathbf{Y}$ is a linear combination of positive real semidefinite matrices $\mathbf{\Phi}_k^T\mathbf{\Phi}_k$ with positive real weights.

A straightforward approach for modeling a given (measured) admittance is to use standard modal analysis tools [Ewins 1992] to fit a modal model of Eq. (8.8) to the measured data, and implement a discretized version of Eq. (8.8). However, there are two related problems which prevent us from doing so. First, standard modal analysis techniques work only in such regions of the transfer function where the modal overlap is low (modes are well separated). Therefore, accurate modal parameters could be obtained for the low frequency region of instrument bridges only. In addition, in the case of sound synthesis applications, the model order is significantly

smaller compared to the order of the system, which means that the assumptions used to derive Eq. (8.8) are no longer true. For example, the poles of the model do not necessarily correspond to the poles of the system, and the "modal shapes" of the model should approximate the gross behavior of all the system modes having modal frequencies near to the corresponding pole frequency of the model.

8.2.3 The passive admittance model

Motivated by the fact that the admittance can be described as a parallel set of analog second-order transfer functions Eq. (8.8), I have developed a passive variant for parallel filter design for the one dimensional [Bank and Karjalainen 2008] and multidimensional [Bank and Karjalainen 2010] cases. Here only the more general, multidimensional case will be outlined. The key idea is a modification to the modal model of Eq. (8.8) by interchanging the $\mathbf{\Phi}_k^T\mathbf{\Phi}_k$ rank 1 matrices with general (full rank) symmetric $\mathbf{Y}_k$ matrices, giving more degrees of freedom in modeling. This actually corresponds to allowing maximum I modal shapes for each pole-pair of the model instead of a single mode. As a result, the admittance is modeled as

$$\mathbf{Y}(z) = \sum_{k=1}^{K} \mathbf{Y}_k H_k(z) \tag{8.9a}$$

$$H_k(z) = \frac{1 - z^{-2}}{(1 - p_k z^{-1})(1 - \overline{p}_k z^{-1)}}, \tag{8.9b}$$

where $H_k(z)$ are the bilinearly transformed discrete-time versions of the second-order functions of Eq. (8.8). If a positive real function $H(s)$ is converted to a discrete-time function $H(z)$ by the bilinear transform, it remains positive real [Smith 1983]. Therefore, $H_k(z)$ are PR. A sufficient condition for the admittance model $\mathbf{Y}(z)$ to be PR is that all the $\mathbf{Y}_k$ matrices are positive semidefinite, because in this case we have

$$\mathrm{Re}\{\mathbf{Y}(z)\} = \mathrm{Re}\{\sum_{k=1}^{K} \mathbf{Y}_k H_k(z)\} = \sum_{k=1}^{K} \mathbf{Y}_k \mathrm{Re}\{H_k(z)\} \geq 0, \tag{8.10}$$

since the linear combination of positive semidefinite matrices $\mathbf{Y}_k$ with nonnegative scalar weights $\mathrm{Re}\{H_k(z)\}$ is also positive semidefinite.

Let us now take a look how the parameters of the admittance model Eq. (8.9) are obtained from a measured admittance matrix $\mathbf{Y}_\mathrm{m}(z)$.

8.2.4 Finding a common set of poles

The measured admittance $\mathbf{Y}_\mathrm{m}(z)$ contains K^2 transfer functions, of which $K(K+1)/2$ are independent, due to symmetry. The task is to find a common-denominator model that best describes all the $K(K+1)/2$ transfer functions, since the poles are the same for each transfer function in the model of Eq. (8.9). This can be done by various common-denominator algorithms used in modal analysis [Ewins 1992; Woodhouse 2004]. Here we will fit a discrete-time all-pole model, similarly as it was done for acoustic transfer functions in [Haneda and Kaneda 1994]. The all-pole design problem is essentially the same as equation error IIR filter design (Prony's method) [Parks and Burrus 1987] without the numerator coefficients. As a notation, let us define $\mathbf{Y}[n]$ as the element-wise inverse z transform of $\mathbf{Y}(z)$, which is actually the impulse

response of the admittance matrix. Accordingly, $\mathbf{Y}_\mathrm{m}[n]$ is the measured admittance impulse response. Then, the output error for the $Y_{\mathrm{m},i,j}[n]$ element of the matrix $\mathbf{Y}_\mathrm{m}[n]$ can be written in the mean-squared sense as

$$E_{i,j} = \sum_{n=0}^{N} \left(Y_{\mathrm{m},i,j}[n] + \sum_{l=1}^{L} a_l Y_{\mathrm{m},i,j}[n-l] \right)^2 , \tag{8.11}$$

where L is the order of the denominator, and N is the length of the measured admittance impulse response $Y_{\mathrm{m},i,j}[n]$. Since the problem is linear in its free parameters a_l, the error is quadratic in a_l, thus, the solution is given in a closed form (see Sec. 2.3).

If the estimation was done independently for all the i, j elements of the impulse response matrix, we would obtain different a_l parameters in each case. Since we want to force the denominator coefficients a_l to be the same for all the i, j elements in Eq. (8.11), the task is to minimize all errors jointly such that the total error

$$e = \sum_{i=1}^{K} \sum_{j=1}^{i} E_{i,j} \tag{8.12}$$

is minimal [Haneda and Kaneda 1994]. This is essentially the same linear least-squares problem, but it has a larger dimensionality.

As in audio applications filter design should have better accuracy at low frequencies compared to the high ones, the above common-denominator model is estimated in the warped domain, similarly to the pole positioning based on warped IIR filters for the single channel case (see Sec. 7.1). For that, all the measured impulse responses are frequency warped with parameter λ, and the common-denominator all-pole model is estimated based on this warped data. Then, the roots $\tilde{p}_k$ of the common denominator are found and "dewarped" by Eq. (4.7). The dewarped poles p_k are then used for constructing the second-order functions $H_r(z)$ according to Eq. (8.9b).

Note that the other pole positioning methods discussed in Chap. 7, such as multi-band or custom warping, can also be used instead of the straightforward warping discussed above with minor modifications to the algorithms.

8.2.5 Weight matrix estimation

The final step is to estimate the weight matrices $\mathbf{Y}_k$, which is a linear-in-parameter problem with the positive-semidefiniteness constraints $\mathbf{Y}_k \geq 0$.

The time-domain error for one matrix element in the mean-squared sense is

$$E'_{i,j} = \sum_{n=0}^{N} \left(Y^k_{i,j} h_k[n] - Y_{\mathrm{m},i,j}[n] \right)^2 \tag{8.13}$$

where $Y^k_{i,j}$ is the i, j element of $\mathbf{Y}_k$ (thus, the superscript k is not a power but an index), and $h_k[n]$ is the inverse z transform of $H_k(z)$.

The optimal set of parameters $\mathbf{Y}_k$ are obtained by solving

$$e' = \sum_{i=1}^{I} \sum_{j=1}^{i} E'_{i,j} \to \min \tag{8.14a}$$

$$\text{subject to} \quad \mathbf{Y}_k \geq 0. \tag{8.14b}$$

Instead of solving Eq. (8.14) by constrained optimization techniques, I have proposed a simple and computationally efficient solution to the problem [Bank and Karjalainen 2010]. The first step of the method involves finding the $\tilde{\mathbf{Y}}_k$ matrices without the constraint of Eq. (8.14b). Since now the elements of $\tilde{\mathbf{Y}}_k$ become independent, the total error is minimal if all $E'_{i,j}$ are minimal. Thus, the problem reduces to minimizing Eq. (8.13) for all $E'_{i,j}$ independently, which are separate linear least-squares problems with a closed-form solution.

As a second step, the resulting $\tilde{\mathbf{Y}}_k$ matrices are "converted" to positive semidefinite matrices. This last step involves finding the nearest positive semidefinite matrix to $\tilde{\mathbf{Y}}_k$. The solution to the problem becomes relatively simple if the distance between the original $\tilde{\mathbf{Y}}_k$ and the positive semidefinite $\mathbf{Y}_k$ matrices is evaluated in terms of the Frobenius norm. Formally, this can be written as the optimization problem

$$||\mathbf{Y}_k - \tilde{\mathbf{Y}}_k||_F = \sqrt{\sum_{i,j}(Y^k_{i,j} - \tilde{Y}^k_{i,j})^2} \rightarrow \min \tag{8.15a}$$

$$\text{subject to} \quad \mathbf{Y}_k \geq 0. \tag{8.15b}$$

It turns out that the optimal $\mathbf{Y}_k$ is obtained by computing the spectral decomposition of $\tilde{\mathbf{Y}}_k$, discarding the negative eigenvalues and their eigenvectors, and reconstructing the matrix from the remaining positive eigenvalues and corresponding eigenvectors to obtain $\mathbf{Y}_k$ [Higham 1988].

While the above two-step solution is suboptimal compared to solving the two lines of Eq. (8.14) jointly, we will see in the example of Sec. 8.2.6 that it provides reasonable accuracy without the need of complicated parameter estimation.

8.2.6 Design example

The example is a two-dimensional admittance modeling of an acoustic guitar bridge (Gibson, from 1960's) near the lowest (E) string. The bridge was excited by the wire breaking technique [Woodhouse 2004; Bank and Karjalainen 2010] and the movement of the bridge was measured by a miniature accelerometer. The bridge was excited with the wire breaking in the direction perpendicular (y direction) and parallel to the body (z direction). The acceleration was also measured in these two directions. This gave a 2 by 2 admittance impulse response matrix

$$\mathbf{Y}_\mathrm{m}[n] = \begin{bmatrix} Y_{\mathrm{m},yy}[n] & Y_{\mathrm{m},yz}[n] \\ Y_{\mathrm{m},zy}[n] & Y_{\mathrm{m},zz}[n] \end{bmatrix} \tag{8.16}$$

where the "m" subscript indicates that these are measured values, which are then approximated by the admittance model impulse response $\mathbf{Y}[n]$. Note that in theory $Y_{\mathrm{m},yz}[n] = Y_{\mathrm{m},zy}[n]$, but there are always some differences due to measurement errors. However, for model fitting, a symmetric $\mathbf{Y}_\mathrm{m}[n]$ matrix is assumed (see Sec. 8.2.3). This is most easily satisfied either by averaging the two responses, or by using only one of them (e.g., the less noisy one).

The parameters of the admittance model were estimated in the time-domain by the parameter estimation procedure outlined in Secs. 8.2.4 and 8.2.5. The results of the parameter estimation for an admittance model having 100 second-order filters ($K = 100$) are shown in Fig. 8.2.6. The thin black lines in Fig. 8.2.6 show the measured responses, while the thick gray lines display the parallel filter responses based on $\tilde{\mathbf{Y}}_k$, before they are converted to the passive versions $\tilde{\mathbf{Y}}_k$ according to Sec. 8.2.5, following the measured response quite accurately. When the passivity constraints are enforced, the filter responses still follow the structure of the admittance, but become shifted (thick black line vs. gray line). Note that this would not happen if the original $\tilde{\mathbf{Y}}_k$

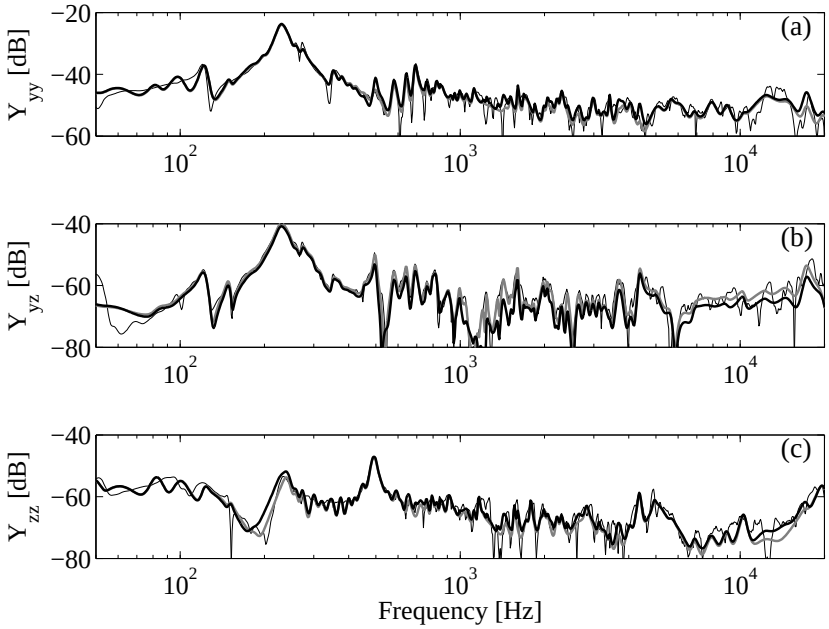


Figure 8.1: Modeling a measured guitar bridge admittance by the proposed passive admittance model with 100 second-order filters for the (a) yy, (b) yz, and (c) zz elements of the admittance matrix. Thin line: measured, thick gray line: non-passive, thick black line: passive parallel filter responses.

responses were passive: since they follow the measurement reasonably well, it can be assumed that the measurement results themselves correspond to a non-passive admittance. This is most probably due to the fact that the wire-breaking provides only approximately constant excitation force as a function of frequency.

As already noted, based on my ideas, an improved parameter estimation method have been later proposed by Maestre et al. [2015, 2017]. Their technique uses the same filter structure composed of positive semidefinite matrices and passive second-order transfer functions (Eq. (8.9)). However, instead of common-pole filter design, in [Maestre et al. 2015] the poles are obtained by sequential quadratic programming, while in [Maestre et al. 2017] by a peak selection algorithm. Finally, the $\mathbf{Y}_k$ matrices are obtained by semidefinite programming enforcing passivity. The comparison in [Maestre et al. 2017] demonstrates that a significant improvement is obtained in modeling accuracy compared to the simpler technique presented here and first published in [Bank and Karjalainen 2010], at the cost of a more complex optimization process.

8.3 MIMO equalization using fixed-pole parallel filters

Equalizing a MIMO transfer function is a common task in audio or acoustic signal processing. The process is often called multichannel inversion or deconvolution, and it basically means designing a set of digital filters given by the matrix $\mathbf{H}(z)$ which modifies the transfer function matrix of the system $\mathbf{S}(z)$ such that the resulting transfer function is close to the target response matrix $\mathbf{Q}_\mathrm{t}(z)$. Crosstalk cancellation is mathematically the same problem. The practical difference as opposed to general equalization is that instead of a full target matrix $\mathbf{Q}_\mathrm{t}(z)$ only certain transfer function paths of the target response (typically the diagonal elements of $\mathbf{Q}_\mathrm{t}(z)$) are nonzero.

The filter matrix $\mathbf{H}(z)$ can be applied both before and after the acoustic system: when

the signals are filtered before they are sent to the loudspeakers, the equalized transfer function becomes $\mathbf{Q}(z) = \mathbf{S}(z)\mathbf{H}(z)$ (this is shown in Fig. 8.2), while when they are filtered after being picked up by the microphones, it is $\mathbf{Q}(z) = \mathbf{H}(z)\mathbf{S}(z)$.

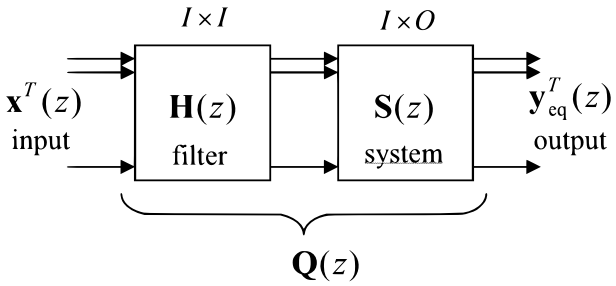


Figure 8.2: The block diagram of MIMO equalization.

Applications of such processing in the audio field include the equalization of room transfer functions [Miyoshi and Kaneda 1988; Sarris et al. 2004; Santillian et al. 2007; Brännmark and Ahlén 2015; Cecchi et al. 2018], crosstalk cancellation for single [Huang et al. 2007; Mertins et al. 2021] and multiple [Huang et al. 2008; Hollebon et al. 2021] listeners, wave-field synthesis with room compensation applied [Gauthier and Berry 2007; Fuster et al. 2005], directivity control of loudspeaker arrays [Zotter et al. 2008; Zotter 2009] and microphone array processing [Benesty et al. 2007]. The implementation of personal sound zones also requires MIMO filtering [Betlehem et al. 2015; Vindrola et al. 2019, 2020; Ebri et al. 2020].

The FIR equalization filters can be designed in the time-domain [Miyoshi and Kaneda 1988; Sarris et al. 2004; Santillian et al. 2007; Huang et al. 2007, 2008; Kirkeby and Nelson 1999] by solving a least-squares problem for approximating the required impulse responses of the transfer function paths. A more efficient alternative (often termed frequency-domain deconvolution) is inverting the transfer function matrix for all frequencies separately, and then computing the equalization filter impulse responses via IFFT [Kirkeby et al. 1998]. This reduces the design time significantly, but results in a sub-optimal solution since the various frequencies are treated independently. As a result, the time-domain least-squares approach requires lower order filters for the same accuracy [Kirkeby and Nelson 1999; Fuster et al. 2005].

Interestingly, the elements of the equalization filter matrix $\mathbf{H}(z)$ are almost always FIR filters. The most probable reason for using FIR filters is that in this case the problem is linear in parameters, allowing a simple mathematical formulation, as opposed to a general IIR filter design case. The only example I am aware of when not FIR filters are used is two-channel crosstalk cancellation using warped FIR filters [Kirkeby et al. 1999; Jeong et al. 2005]. Warped FIR filters are designed similarly to normal FIR filters once all the impulse responses are warped, but they act as IIR filters when being implemented (see Chap. 4). It is shown in [Jeong et al. 2005] that compared to a straightforward FIR equalizer, frequency warping leads to better channel separation in the important low frequency region.

We know from single-channel IIR filter design that IIR filters allow modeling or equalizing physical systems at lower filter order for the same accuracy compared to FIR filters, and this is especially true for audio applications where the inherently linear frequency resolution of FIR filters does not match the logarithmic frequency resolution of hearing. Therefore, the aim of my research at this point was to generalize the fixed-pole design of parallel filters to the MIMO equalization case [Bank 2018b].

8.3.1 Two-step equalizer design

The most straightforward way of designing a MIMO equalizer using parallel filters is a two-step procedure. First, an equalizer filter matrix $\mathbf{H}_{\mathrm{FIR}}(z)$ is estimated in the FIR form either in the frequency domain [Kirkeby et al. 1998; Vindrola et al. 2019] or in the time-domain [Miyoshi and Kaneda 1988; Sarris et al. 2004; Santillian et al. 2007; Huang et al. 2007, 2008; Kirkeby and Nelson 1999] by the readily available methods. Second, for each element of $\mathbf{H}(z)$ a single-channel parallel filter is designed, which consists in finding the suitable poles by one of the methods of Chap. 7, then estimating the filter numerators by an LS fit as described in Secs. 6.2. However, this is mathematically not optimal, since the error is minimized in two separate steps: first, an intermediate FIR equalizer is designed that minimizes the error between the target response and equalized system response matrices, and next separate parallel filters are estimated where the errors between the impulse responses of the FIR and parallel filters are minimized.

8.3.2 Direct equalizer design

To improve the performance compared to the above two-step process, I have proposed a direct equalizer design procedure in [Bank 2018b] that is the generalization of the single-channel case described in Sec. 6.3. The poles of the parallel filters are determined by one of the methods presented in Chap. 7 based on the FIR filter matrix $\mathbf{H}_{\mathrm{FIR}}(z)$, similarly to the two-step case of Sec. 8.3.1. The difference lies in the fact that here the numerators of the second-order sections are not estimated based on the intermediate FIR equalizer $\mathbf{H}_{\mathrm{FIR}}(z)$, but obtained in a single step, where the error between the target response matrix and the equalized system response is minimized.

In the MIMO equalization problem we have a system with the matrix transfer function $\mathbf{S}(z)$ which gives the output signal vector

$$\mathbf{y}(z) = \mathbf{S}(z)\mathbf{x}(z) \tag{8.17}$$

as a response to the input column vector $\mathbf{x}(z)$.

The response is equalized by filtering the input signals $\mathbf{x}(z)$ with a MIMO filter $\mathbf{H}(z)$ prior to sending them to the system, giving the equalized output

$$\mathbf{y}_{\mathrm{eq}}(z) = \mathbf{S}(z)\mathbf{H}(z)\mathbf{x}(z) = \mathbf{Q}(z)\mathbf{x}(z), \tag{8.18}$$

where $\mathbf{Q}(z)$ is the transfer function matrix of the equalized MIMO system (see Fig. 8.2). Note that in the alternative configuration, the MIMO filter is applied to the output of the system, giving basically the same equations with interchanged order of the matrices.

In equalizer design our goal is to optimize the parameters of the filter $\mathbf{H}(z)$ such that the resulting equalized transfer function $\mathbf{Q}(z) = \mathbf{S}(z)\mathbf{H}(z)$ is the closest to the target $\mathbf{Q}_{\mathrm{t}}(z)$. The problem can be separated into independent SIMO subproblems by splitting $\mathbf{Q}(z)$ and $\mathbf{H}(z)$ to their columns as $\mathbf{Q}(z) = [\mathbf{q}_1(z) \dots \mathbf{q}_I(z)]$ and $\mathbf{H}(z) = [\mathbf{h}_1(z) \dots \mathbf{h}_I(z)]$, and thus

$$\mathbf{Q}(z) = \mathbf{S}(z)\mathbf{H}(z) \;\Rightarrow\; \mathbf{q}_i(z) = \mathbf{S}(z)\mathbf{h}_i(z) \tag{8.19}$$

for all $i = [1 \dots I]$. This basically corresponds to obtaining the equalized transfer functions from a specific input i to all outputs, given in vector $\mathbf{q}_i(z)$. Now this should be close to the ith

column of $\mathbf{Q}_\mathrm{t}(z)$ given as $\mathbf{q}_{\mathrm{t},i}(z)$, which actually specifies the desired transfer functions to all outputs if only input i is excited.

For the sake of clarity but without the loss of generality we write the equalization problem (8.19) for a system having two inputs and two outputs. For a specific input i (8.19) becomes:

$$q_{1,i}(z) = S_{1,1}(z)h_{1,i}(z) + S_{1,2}(z)h_{2,i}(z), \tag{8.20a}$$
$$q_{2,i}(z) = S_{2,1}(z)h_{1,i}(z) + S_{2,2}(z)h_{2,i}(z). \tag{8.20b}$$

Writing this in the time-domain yields

$$q_{1,i}(n) = s_{1,1}(n) * h_{1,i}(n) + s_{1,2}(n) * h_{2,i}(n), \tag{8.21a}$$
$$q_{2,i}(n) = s_{2,1}(n) * h_{1,i}(z) + s_{2,2}(n) * h_{2,i}(n), \tag{8.21b}$$

where $*$ denotes convolution.

The impulse responses $h_{1,i}(n)$ and $h_{2,i}(n)$ of the equalizer filters can be written as a linear combination of the same basis functions as for the single channel case, that is, the decaying responses of second-order denominators, and the delayed impulses of the FIR part (see Sec. 6.2.1). To simplify the notations, here we will call all modeling signals $m_k(n)$, which means that $m_1(n) = u_1(n)$, $m_2(n) = u_1(n-1)$ representing the decaying response of the first denominator and its delayed version, then $m_3(n) = u_2(n)$, $m_4(n) = u_2(n-1)$, and so on. In general, $m_{2k-1}(n) = u_k(n)$ and $m_{2k}(n) = u_k(n-1)$, for $k = 1 \ldots K$. Finally, the last m_k signals contain the delayed unit pulses corresponding to the optional FIR path. Since the poles of the filters can be different, two different sets of basis functions are needed: $m_k(n)^{(1,i)}$ for $h_{1,i}(n)$ and $m_k(n)^{(2,i)}$ for $h_{2,i}(n)$.

Thus, the equalized impulse response of the first output channel becomes

$$q_{1,i}(n) = \sum_{k=1}^{R} r_k^{(1,i)} m_k(n)^{(1,i)} * s_{1,1}(n) + \sum_{k=1}^{R} r_k^{(2,i)} m_k(n)^{(2,i)} * s_{1,2}(n), \tag{8.22}$$

and for the second output channel it is

$$q_{2,i}(n) = \sum_{k=1}^{R} r_k^{(1,i)} m_k(n)^{(1,i)} * s_{2,1}(n) + \sum_{k=1}^{R} r_k^{(2,i)} m_k(n)^{(2,i)} * s_{2,2}(n), \tag{8.23}$$

where $r_k^{(1,i)}$ contain the filter weights (numerator coefficients and the parallel FIR coefficients) of the first filter $h_{1,i}(n)$ and $r_k^{(2,i)}$ contain the filter weights of the second filter $h_{2,i}(n)$.

Now we need to jointly optimize $r_k^{(1,i)}$ and $r_k^{(2,i)}$ such that $q_{1,i}(n)$ and $q_{2,i}(n)$ will be close to the targets $q_{\mathrm{t},1,i}(n)$ and $q_{\mathrm{t},2,i}(n)$, respectively. For this, we write Eqs. (8.22) and (8.23) in a joint matrix form

$$\begin{bmatrix} \mathbf{q}_{1,i} \\ \mathbf{q}_{2,i} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{1,i} * \mathbf{s}_{1,1} & \mathbf{M}_{2,i} * \mathbf{s}_{1,2} \\ \mathbf{M}_{1,i} * \mathbf{s}_{2,1} & \mathbf{M}_{2,i} * \mathbf{s}_{2,2} \end{bmatrix} \begin{bmatrix} \mathbf{r}_{1,i} \\ \mathbf{r}_{2,i} \end{bmatrix}, \tag{8.24}$$

where $\mathbf{q}_{o,i}$ contains the impulse response of the equalized system from input i to output o, $\mathbf{s}_{o,i}$ contains the system impulse response from input i to output o, $\mathbf{M}_{o,i}$ are the same type of modeling matrices as for the single-channel case (see Sec. 6.2.1) for the filter $H_{o,i}(z)$, and finally $\mathbf{r}_{o,i}$ contain the free parameters of the parallel filters. Writing (8.24) in a compact form by composing a single matrix and two column vectors yields

$$\mathbf{q}_\mathrm{c} = \mathbf{M}_\mathrm{c}\mathbf{r}_\mathrm{c}, \tag{8.25}$$

which should be close to

$$\mathbf{q}_{\mathrm{t,c}} = \begin{bmatrix} \mathbf{q}_{\mathrm{t},1,i} \\ \mathbf{q}_{\mathrm{t},2,i} \end{bmatrix} \tag{8.26}$$

in the mean square sense, where $\mathbf{q}_{\mathrm{t},1,i}$ is the target impulse response from input i to the first output, and $\mathbf{q}_{\mathrm{t},2,i}$ is the target impulse response from input i to the second output.

The solution can be again given in a closed form:

$$\mathbf{r}_{\mathrm{c,opt}} = \mathbf{M}_{\mathrm{c}}^{+}\mathbf{q}_{\mathrm{t,c}}, \tag{8.27a}$$

$$\mathbf{M}_{\mathrm{eq}}^{+} = (\mathbf{M}_{\mathrm{c}}^{T}\mathbf{M}_{\mathrm{c}})^{-1}\mathbf{M}_{\mathrm{c}}^{T}. \tag{8.27b}$$

While Eq. (8.27) is appealing due to its compact form, it is important to note that computing the pseudoinverse for large matrices, such as the ones which usually occur in MIMO parallel filter design is numerically problematic. As already noted in Sec. 2.3, numerically robust methods for solving large least-squares problems include the Cholesky decomposition or QR factorization [Golub and Loan 2013]. In MATLAB the later is implemented by the `mldivide` function.

8.3.3 Design examples

As an illustrative example for the IIR MIMO equalizer, here a multichannel crosstalk cancellation application is presented. Crosstalk cancellation is particularly well suited to compare the effectiveness of MIMO equalization techniques due to its simple target function: the specification is constant response in a few transfer function paths, and zero in all the others. This allows the easy inspection of the equalization performance.

The plant to be equalized is a spherical loudspeaker array having 20 speaker elements. Spherical loudspeaker arrays allow the separate excitation of each speaker element, and thus a sound source with controlled directivity (beamforming) can be constructed. Such speakers have been used in acoustic measurements [Warusfel et al. 1997; Zotter and Höldrich 2007; Neal and Vigeant 2020], local sound control (noise cancellation) [Rafaely 2009], and in contemporary music performances [Zotter et al. 2017].

The desired directivity of spherical speaker is usually given in terms of spherical harmonics. A decoder matrix transforms the spherical harmonic signals into loudspeaker signals. Accordingly, directivity control requires the individual control of each loudspeaker element. Therefore a crosstalk canceller must be applied before the loudspeaker signals are actually sent to the speaker elements [Zotter 2009; Zotter et al. 2017]. The crosstalk canceller is usually implemented as an FIR filter, in the examples below we will investigate the possibilities of applying fixed-pole IIR filters instead.

The transfer functions of the speaker elements have been measured by exciting the speakers separately with logarithmic sweep signals and measuring the velocity of each loudspeaker cap using a laser-vibrometer [Zotter 2009]. Since meaningful beamforming can be achieved by such a large array only up to a few kHz due to spatial aliasing, the sampling rate was limited to $f_{\mathrm{s}} = 11.025$ kHz. The measured responses are windowed to 2048 taps where the impulse responses have already died out. This gave a 20×20 transfer function matrix, ideally having nonzero elements only in its diagonal. However, since the speaker elements share the same air volume in the enclosure, there is a significant crosstalk between the channels. This is displayed in Fig. 8.3 (a), showing the first column of the transfer function matrix $\mathbf{S}(z)$, corresponding to the case when loudspeaker No. 1 is excited and all speakers are measured. It can be seen in Fig. 8.3 (a) that neither the transfer function of the diagonal entry is flat (gray thick line), nor

the off-diagonal entries are zero (thin lines). Other input channels have a similar (though not completely identical) behavior, since all the 20 loudspeakers are of the same type, but slightly different due to manufacturing tolerances.

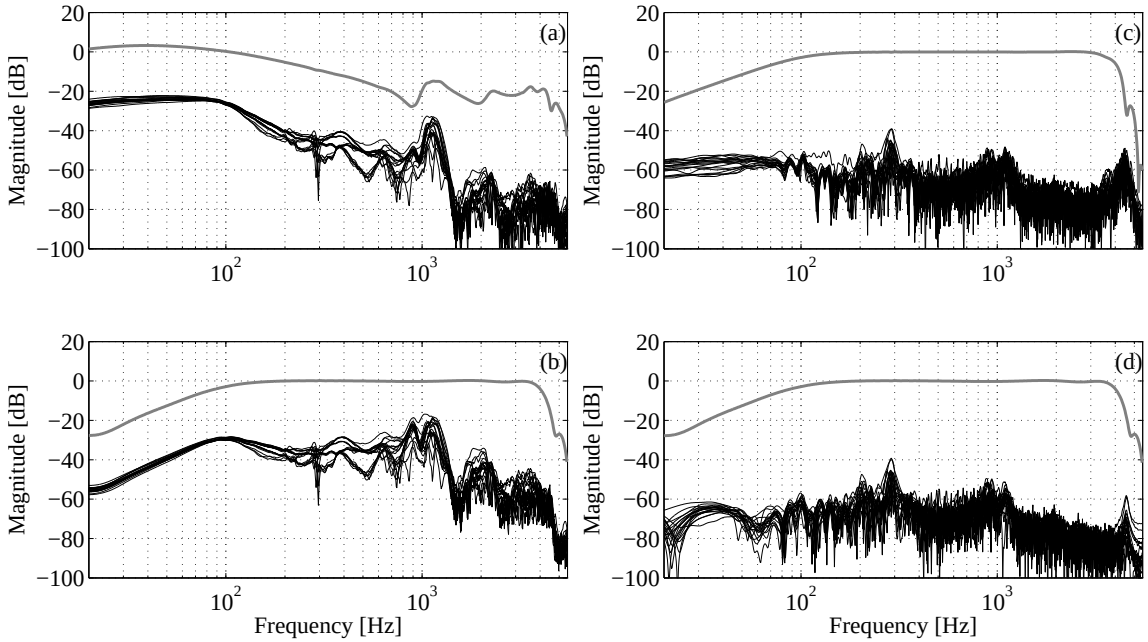


Figure 8.3: Velocity responses of a 20 channel spherical loudspeaker array when only speaker No. 1 is excited: (a) original, equalized by (b) a single-channel 256 tap FIR filter, (c) by a 256 tap MIMO FIR filter designed by the fast deconvolution method of [Kirkeby et al. 1998], and (d) by a 256 tap MIMO FIR filter designed in the time domain by a least squares fit [Kirkeby and Nelson 1999; Huang et al. 2007]. Gray thick line: diagonal element, thin line: crosstalk products.

The first approach to equalization is designing a single-channel 256 tap FIR filter that flattens the frequency response of the diagonal: the effect is shown in Fig. 8.3 (b) using a band-pass target response with the corner frequencies of 100 Hz and 4 kHz. Naturally, a filter applied to input channel 1 only will not change the relative values of the responses; therefore the relative crosstalk remains unaffected, whose worst case is around -20 dB at 1 kHz.

Next, different MIMO equalizers are designed that feed all the 20 loudspeaker channels. Here we will only display the first SIMO subproblem aiming at producing a band-pass response for speaker 1 (the corner frequencies are 100 Hz and 4 kHz), while minimal (ideally, zero) velocity output for speakers 2–20. Thus, only the first column of the equalized response matrix $\mathbf{Q}(z)$ will be shown. The other input channels have an almost identical behavior due to the symmetry of the loudspeaker arrangement.

The first example is the case of frequency-domain deconvolution [Kirkeby et al. 1998]. The full length (2048 taps) impulse responses obtained from deconvolution (termed *long Kirkeby inverse* here) are windowed to 256 taps and applied as a MIMO FIR equalizer. (Note that "256 tap" means 256 tap FIR filters in each element of the filter matrix.) The results are shown in Fig. 8.3 (c). It can be seen in Fig. 8.3 (c) that a 256 tap impulse response reduces the maximal crosstalk from -20 dB to -40 dB compared to Fig. 8.3 (d). However, a peak in the crosstalk terms at 300 Hz arises in Fig. 8.3 (c): this corresponds to the problematic area having sharp notches around 300 Hz in the original transfer function (see Fig. 8.3 (a)). For eliminating this,

a longer impulse response, and thus finer frequency resolution would be needed in that region.

The time-domain FIR LS design [Kirkeby and Nelson 1999; Huang et al. 2007] leads to a slightly better performance for the same filter length due to the fact that now the filter is optimized for the available 256 taps, and not obtained by windowing a long optimal filter, as for the frequency-domain deconvolution case. This is shown in Fig. 8.3 (d), and the difference is mostly visible at low frequencies compared to Fig. 8.3 (c).

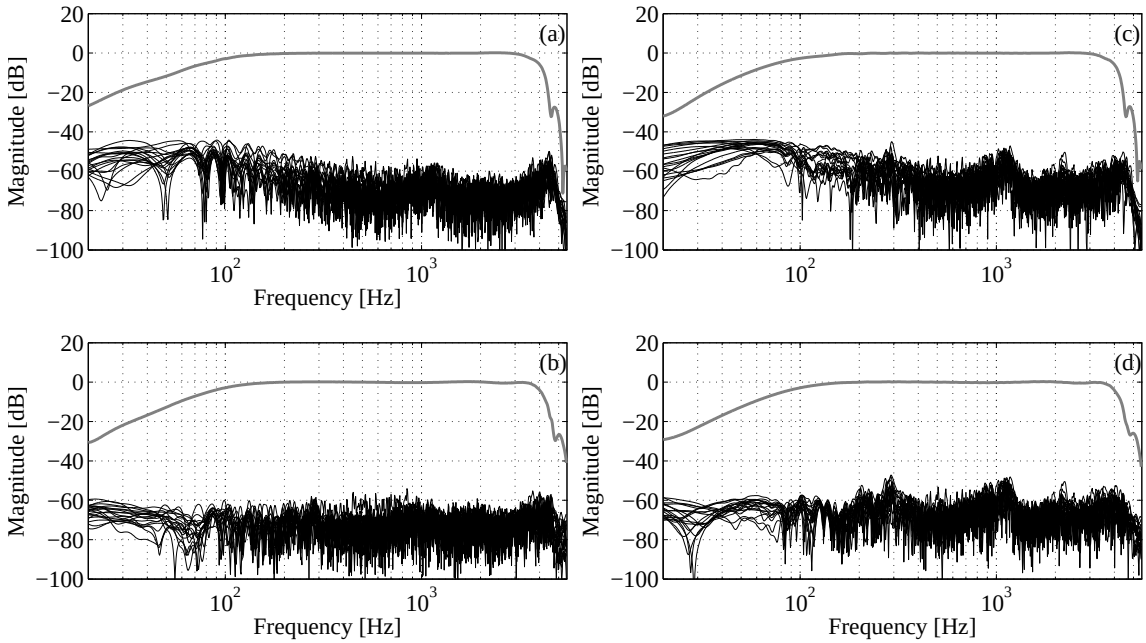


Figure 8.4: Velocity responses of the loudspeaker array equalized (a) by a 128th order MIMO parallel filter obtained by the two-step design and (b) by a 128th order MIMO parallel filter obtained by the direct-design approach. The same examples with lower filter order: the response equalized by (c) a 64th order MIMO parallel filter obtained by the two-step design and (d) by a 64th order MIMO parallel filter obtained by the direct-design approach. Gray thick line: diagonal element, thin line: crosstalk products.

Finally, two parallel filter designs are presented. Figure 8.4 (a) shows an example where each filter path contains a 128th order IIR filter (64 second-order IIR sections and no parallel FIR path) which needs the same amount of multiplication and addition operations as the 256 tap FIR filter. The first filter is obtained as described in Sec. 8.3.1, that is, separate fixed-pole parallel filters are designed based on the *long Kirkeby inverse*. The poles are obtained by a warped IIR design with $\lambda = 0.6$, as described in Sec. 7.1. As can be seen in Fig. 8.4 (a), the performance is significantly improved around the problematic region near 300 Hz compared to the FIR equalizers of Fig. 8.3 (c) and (d). On the other hand, the crosstalk at low frequencies is increased.

Next, a 128th order MIMO parallel filter is obtained by the direct design method proposed in Sec. 8.3.2. The poles of the filter are obtained from the *long Kirkeby inverse* via a warped IIR deign with a warping parameter $\lambda = 0.6$ similarly to the previous example. However, the numerator coefficients are estimated by the LS procedure proposed in Sec. 8.3.2. According to Fig. 8.4 (b) the channel separation is improved significantly compared to the FIR equalizer case of Fig. 8.3 (d): while for the FIR case the largest crosstalk is -40 dB in the operating range of the loudspeaker, this is reduced to -60 dB with the parallel filter by using the same amount of

arithmetic operations.

Fig. 8.4 (c) and (d) shows a different type of comparison: now the order of the parallel filter is 64, meaning that the number of required arithmetic operations is the half compared to the FIR examples. On the other hand, the parallel equalizer obtained by the direct design method (Fig. 8.4 (d)) still outperforms the FIR equalizers of Fig. 8.3 (c) and (d), since the maximal crosstalk is -50 dB instead of -40 dB of the FIR equalizers. The direct design of Fig. 8.4 (d) leads to a significantly improved low-frequency performance compared to the two-step procedure of Fig. 8.4 (c), similarly to the previous example.

To sum up, this section has generalized the fixed-pole design of parallel second-order filters for the equalization of MIMO systems. By taking advantage of fixed poles, the design of an IIR equalizer is equally simple as that of the commonly used FIR equalizer and a closed-form solution can be obtained when the problem is formulated in the least-squares sense. The significance of this result is that in many MIMO equalization applications parallel filters can be used instead of FIR filters without major modifications to the methods and algorithms, with higher flexibility in frequency resolution that can lead to better equalization performance for the same number of arithmetic operations.

8.4 Scientific statements

Statement 3: I have given the multichannel extension of the fixed-pole parallel filter and shown that significant computational savings can be achieved with a common pole set. I have given the extension of single-channel design techniques for passive admittance matrix modeling and MIMO equalizer design.

3.1: I have developed a filter structure and a corresponding design method that can be used for modeling MIMO passive systems in the full audio frequency range. The key idea is to construct the filter matrix as a linear combination of frequency-independent positive semidefinite matrices and single-channel positive real second-order filters. The poles are chosen based on warped common-pole all-pole modeling, and the positive semidefinite matrices are found by an unconstrained LS design and then finding the closest positive semidefinite matrix.

3.2: I have extended the direct equalizer design method for the equalization of MIMO systems and showed that the problem remains linear in its parameters also for the MIMO case. This means that the least squares method commonly used for designing FIR MIMO equalizers remains applicable for the parallel filter, with the added flexibility of the arbitrary frequency resolution achievable by fixed-pole parallel filters.

The related results have been published in one journal paper [Bank 2018b] and in one conference paper [Bank and Karjalainen 2010][1].

[1]The publication has been written together with late Prof. Matti Karjalainen. I have developed the filter design algorithm presented in this thesis, while his contribution was about converting the admittance filters to reflection filters so that they can be directly connected to digital waveguide string models. His part of the work has not been included in this thesis, nor in the above list of scientific statements.

Chapter 9

The delayed parallel filter

The original parallel filter structure of Fig. 6.1 stems from the form used in partial fraction expansion Eq. (6.1), where the FIR part is in parallel with the IIR part. However, later I have discovered that in the case of a long FIR part a dynamic range problem arises which can cause numerical issues when the filter is implemented in a DSP: the magnitude response of the FIR part and the second-order sections are often significantly larger than that of the resulting filter, requiring the downscaling of the input signal or the numerator coefficients to avoid overflow, which leads to a loss of useful bitdepth [Bank and Smith 2014]. I have realized that this is because of the time-domain overlap of the FIR and IIR parts, and I have proposed a solution to the problem by using a delayed parallel structure where IIR response starts after the FIR part [Bank and Smith 2014]. This will be presented in Sec. 9.1.

Perhaps even more importantly, I have demonstrated that exactly the same problem arises when direct form IIR filters are converted to parallel second-order sections by the usual procedure of partial fraction expansion. As a solution, I suggested to use the delayed parallel form for implementation in [Bank and Smith 2014; Bank 2018a]. In addition, I have demonstrated that the usual partial fraction expansion can be numerically problematic for high ($N > 100$) filter orders, and proposed a more robust least-squares direct-to-parallel conversion method [Bank 2018a]. These results will be presented in Sec. 9.2.

9.1 The delayed parallel filter for fixed-pole design

Since causal IIR filters with a proper transfer function compose their impulse response as a sum of decaying exponentials, they are most suited to model decaying impulse responses. Modeling minimum-phase systems are thus optimal targets for IIR filter design, since in that case the energy of the impulse response is concentrated to the beginning of the response, as already discussed in Sec. 2.4. However, many systems have non-minimumphase behavior, and thus a rising part in the beginning of their impulse response. In such cases it is more efficient to model the first part up to the highest peak of the impulse response by an FIR filter, and the decaying part with an IIR filter. Coming this observation, I have suggested the use of an FIR path in parallel with the second-order sections in [Bank 2007], to remain compatible with the usual parallel form obtained by partial fraction expansion. At that time I did not realize that this can lead to numerical problems especially when the FIR part has a high order. However, later it had turned out that this parallel FIR part can lead to a dynamic range limitation of the filter. This is described in the following.

9.1.1 The dynamic range problem

Let us first look at an example that illustrates the problem. For this, a fixed-pole parallel filter is designed to model a piano soundboard response with 50 second-order sections. The target impulse response is highly nonminimum-phase, as can be seen in Fig. 9.2 (a), thin line (notice that the peak of the response is at around $n = 200$ and not at around $n = 0$). Since the second-order sections cannot efficiently model the rising part of the response, a 200th order parallel FIR part is added, as I have suggested in [Bank 2007]. As can be seen in the Fig. 9.1 (a) thick line and Fig. 9.2 left column (a) thick line, the filter follows the specification quite well. However, when looking at the individual responses of the sections, we see a striking picture: 130 dB difference arises between the peaks of some second-order sections (thin lines) and the peak of the final transfer function (thick line), as can be seen in Fig. 9.1 (a). In a fixed-point implementation this would require the heavy downscaling of the input signal or the numerator coefficients to avoid overflow in the sections, and this 130 dB difference would thus actually lead to the loss of 21 bits precision. In floating point, no downscaling is necessary since the number format takes care of that automatically. However, the loss of useful bits is still the same in that case.

I have given an explanation for this behavior in [Bank and Smith 2014], which will be outlined here. According to Sec. 6.2.1, in time-domain design of parallel filters a linear least squares (LS) fit is performed where the basis functions are delayed impulses for the FIR part and exponentially decaying sinusoids (and their one-sample delayed versions) for the second-order IIR sections. The weights are set by the LS design so that the filter impulse response best approximates the target impulse response. Since an Mth order FIR part gives complete freedom for setting the first $M + 1$ samples of the filter response, the LS design can set the FIR coefficients in such a way that the first $M + 1$ samples ($n = [0..M]$) are matched perfectly. This means that the numerators of the second-order sections will depend only on the samples after the FIR part ($n > M + 1$). This can be shown by partitioning the matrices involved in the least-squares design, as outlined in Appendix A.6.

If the FIR part is sufficiently long, the decaying sinusoids corresponding to the denominators $1/(1 + a_{1,k}z^{-1} + a_{2,k}z^{-1})$ have already a low level for these samples $n > M + 1$, which is counteracted by the LS design by increasing the initial amplitudes of these sinusoids, and thus the numerator coefficients of the second-order filters. In return, this will mean a large signal at the beginning of the response, overlapping the FIR part, as can be seen in Fig. 9.2 left column (b). Actually, the FIR coefficients arise as the difference of the target response and the response of the IIR sections for the first $M + 1$ samples (see Eq. (A.29) in the Appendix), thus, besides setting the initial sample values, another role of the FIR coefficients is to cancel the response of the IIR part in the first $M + 1$ samples. This is shown in Fig. 9.2 left column (b) and (c); here also note the different amplitude scale compared to (a).

Once understood, this numerical problem can be solved in a simple way: the parallel IIR part must be delayed so that there is no overlap between the FIR and IIR parts:

$$H(z) = z^{-(M+1)} \sum_{k=1}^{K} \frac{\tilde{b}_{k,0} + \tilde{b}_{k,1}z^{-1}}{1 + a_{k,1}z^{-1} + a_{k,2}z^{-2}} + \sum_{m=0}^{M} \tilde{f}_m z^{-m}, \tag{9.1}$$

The first $M + 1$ samples of the impulse response are now determined solely by the the Mth order FIR part, and the rest of the impulse response by the IIR part.

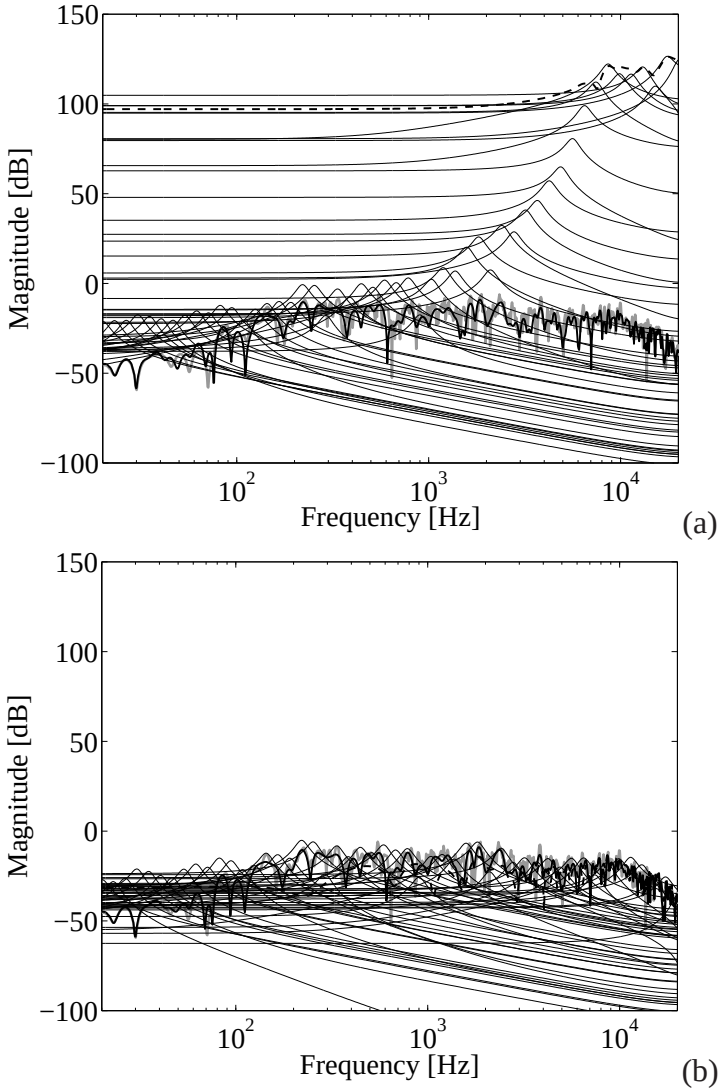


Figure 9.1: Fixed-pole parallel filter implementations aiming to model a piano soundboard response with the original, non-delayed parallel filter (a) and the delayed form (b). The filters contain 50 second-order IIR sections and a 200 tap FIR path. Thick gray line: target frequency response, thick line: filter response, dashed line: frequency response of the FIR part, thin lines: the individual responses of the second-order sections.

9.1.2 Conversion from the original form to the delayed form

The parameters of the delayed filter structure can be obtained from the original parallel structure of Eq. (6.2) with an Mth order FIR part as follows: the first $M+1$ samples of the filter impulse response $h(n)$ are computed, and these samples are directly used as the new FIR coefficients:

$$\tilde{f}_m = h(m) \text{ for } m = [0, 1, \ldots M]. \tag{9.2}$$

For the parallel IIR sections, the denominators remain the same and the numerators are set in such a way that the decaying exponentials of the delayed form have the same amplitude and phase at sample $n = 0$ as at sample $n = M+1$ with the original sections. First the second-order sections are decomposed to a pair of complex first-order IIR filters to obtain the complex form

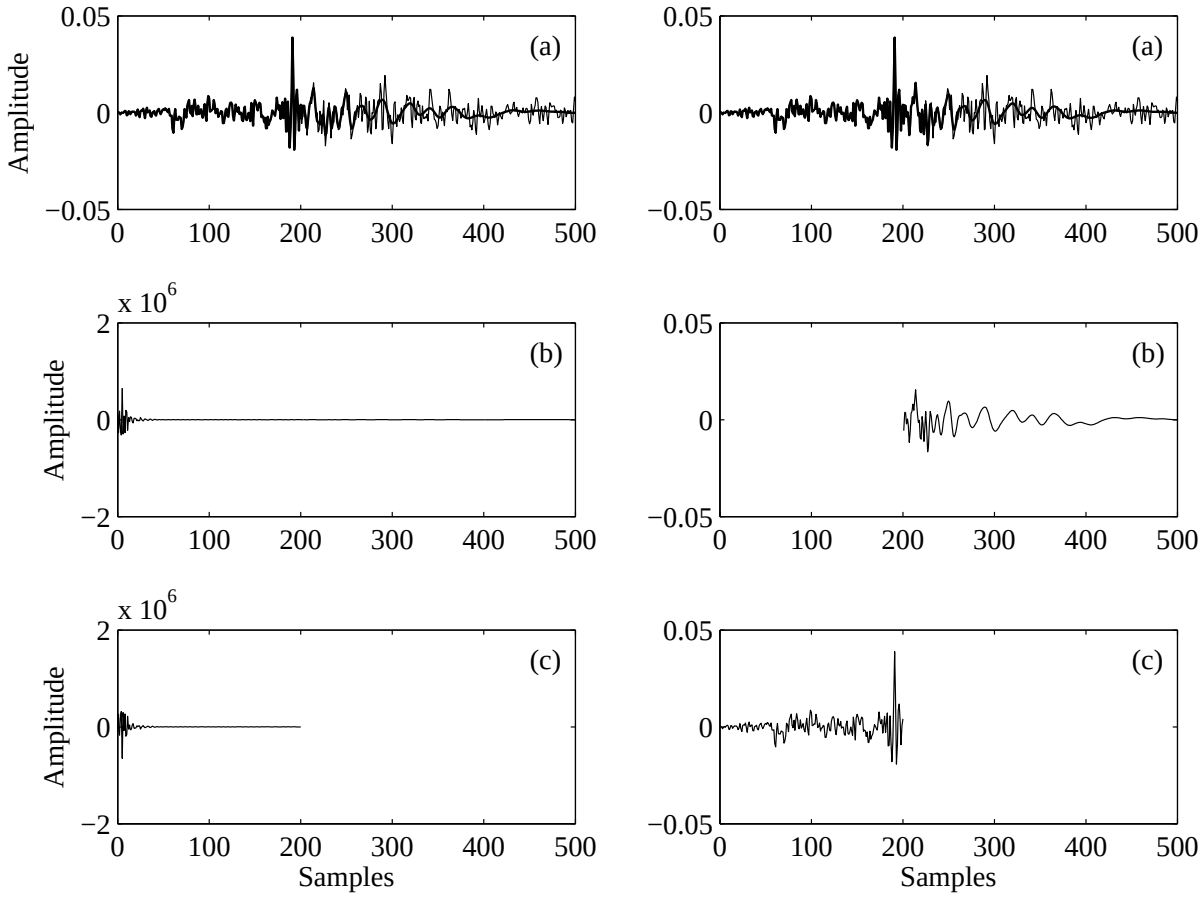


Figure 9.2: Time-domain responses of the parallel filter shown in Fig. 9.1. Left column: the original, non-delayed form, right column: delayed form. In both columns: (a) target impulse response (thin line) and filter impulse response (thick line), (b) the response of the second-order sections, and (c) the response of the 200th order FIR part.

of Eq. (6.1). Then the modified residues $\tilde{c}_i$ are obtained as

$$\tilde{c}_i = c_i p_i^{M+1}, \tag{9.3}$$

and finally the first-order filters are combined to form second-order sections having real coefficients.

9.1.3 Design in the delayed form

It is also possible to design the parallel filter in the delayed form of Eq. (9.1) directly, instead of converting from the original form. In this case, we choose the FIR coefficients $\tilde{f}_m$ equal to the first $M+1$ samples of the target impulse response $h_\mathrm{t}(n)$:

$$\tilde{f}_m = h_\mathrm{t}(m) \text{ for } k = [0, 1, \ldots M], \tag{9.4}$$

where M is the order of the FIR part.

Then, the remaining part of the target will be used as a specification

$$\tilde{h}_\mathrm{t}(n) = h_\mathrm{t}(n + M + 1) \tag{9.5}$$

for designing a FIR-less parallel filter. Thus, the modeling matrix $\tilde{\mathbf{M}}$ will contain only the responses corresponding to the second-order sections, but no unit pulses.

Besides avoiding the conversion, this has an added benefit of decreasing the computational complexity of the design, since now the normal equations estimate $M + 1$ fewer parameters. In addition, the design problem becomes numerically better conditioned, since in the original design the same $M + 1$ samples are determined both by the FIR and the IIR parts, which is now avoided.

Note that when the delayed parallel filter is designed based on a frequency response specification, we cannot take advantage of the fact that the FIR coefficients are the same as the early part of the impulse response. In this case all free parameters (FIR and numerator coefficients) have to be computed by the LS solution, similarly to Sec. 6.2.2. The only difference is that the frequency responses of the second-order sections in the frequency-domain modeling matrix $\mathbf{M}$ used in Sec. 6.2.2 are multiplied by $z^{-(M+1)} = e^{-j\vartheta_n(M+1)}$ in accordance with Eq. (9.1).

This is also the case for direct equalizer design, where the modeling matrix $\mathbf{M}_{\text{eq}}$ must be modified so that the responses corresponding to second-order parts are delayed by $M + 1$ samples. For time-domain design, this simply means shifting down the corresponding columns of $\mathbf{M}_{\text{eq}}$ by $M + 1$ samples and inserting zeros for the first $M + 1$ elements, while for frequency-domain design this achieved by a multiplication with $z^{-(M+1)} = e^{-j\vartheta_n(M+1)}$.

9.1.4 Delayed parallel filter example

To show that the delayed form of the parallel filter eliminates the dynamic range problem of Fig. 9.1 (a), a delayed parallel filter is designed based on the same target response and same pole positions as for Fig. 9.1 (a). The filter is designed directly in the delayed form in the time-domain, as discussed in Sec. 9.1.3. It can be seen in Fig. 9.1 (b) that now the gains of the individual sections (dotted lines) and the FIR part (dashed line) are in the same range as that of the total transfer function: the highest peak of a second-order section is now only 2 dB larger than the peak of the overall response. The same responses are displayed in the time-domain in Fig. 9.2 right column, showing how the delayed IIR (b) and FIR (c) parts are combined to form the total impulse response (a). The filter impulse response obtained by the this modified design (Figs. 9.1 (b) and 9.2 (a) right column) is the same as that of the original parallel filter (Figs. 9.1 (a) and 9.2 (a) left column) up to numerical precision, and the same would also be true if the delayed parallel filter was obtained from the original parallel filter by conversion according to Sec. 9.1.2. However, now the need for downscaling and thus the dynamic range reduction is avoided.

9.2 Obtaining parallel filters from direct form IIR filters

So far we have used the parallel second-order IIR structure in a fixed-pole design context for gaining control over the frequency resolution of the design. A more traditional and thus more widespread use of the parallel second-order structure is to use it as an alternative implementation form for IIR filters designed in direct form, as already mentioned in the introduction of Chap. 6. This is because a theoretically stable IIR filter might become unstable when implemented with finite coefficient precision due to coefficient rounding. The problem becomes pronounced when the filter has high order and/or has poles near the unit circle. As a remedy, IIR filters are

often implemented as a series or parallel combination of (typically, second-order) subfilters [Oppenheim et al. 1999; Rabiner and Gold 1975; Chen 1996].

Traditionally, the series connection of second-order filters has been more common. However, nowadays the parallel implementation is gaining more and more interest since it provides several advantages compared to series biquads: it has lower quantization noise [Chen 1996], and even more importantly, it leads to a significant speedup in modern multi-core processors that can take advantage of the fully parallel filter structure [Belloch et al. 2014].

While alternative methods are available for direct-to-parallel conversion [Price et al. 1996; Krukowski et al. 1996], by far the most common way of converting filters to parallel form is based on partial fraction expansion [Oppenheim et al. 1999]. Here the first step is converting the rational transfer function to the residue form

$$H(z^{-1}) = \frac{r_1}{1 - p_1 z^{-1}} + \frac{r_2}{1 - p_2 z^{-1}} + \ldots + \frac{r_{N\text{den}}}{1 - p_{N\text{den}} z^{-1}}, \tag{9.6}$$

where r_n are the residues corresponding to the poles p_n, and N_{den} is the order of the numerator. The usual way of determining r_n is the Heaviside cover-up method, which can be formulated mathematically as

$$r_n = (1 - z^{-1} p_n) H(z)\big|_{z=p_n}. \tag{9.7}$$

Note that (9.6) and (9.7) is general only if poles p_n are distinct. In the case of pole multiplicity, higher order terms also appear [Smith 2007].

The partial fraction expansion requires that the transfer function $H(z)$ is strictly proper, that is, the order of the denominator is larger than the order of the numerator ($N_{\text{den}} > N_{\text{num}}$). If this is not the case, polynomial long division is performed to result in a FIR part $F(z^{-1})$ and a strictly proper IIR part $B'(z^{-1})/A(z^{-1})$ as

$$H(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{B'(z^{-1})}{A(z^{-1})} + f_0 + f_1 z^{-1} + \ldots + f_M z^{-M}, \tag{9.8}$$

where $M = N_{\text{num}} - N_{\text{den}}$ is the order of the FIR part. Then the partial fraction expansion is applied to the strictly proper $B'(z^{-1})/A(z^{-1})$. The last step of the conversion is combining the complex-conjugate pairs to second-order sections having real coefficients:

$$H(z) = \sum_{l=1}^{L} \frac{b_{0,l} + b_{1,l} z^{-1}}{1 + a_{1,l} z^{-1} + a_{2,l} z^{-2}} + \sum_{m=0}^{M} f_m z^{-m}. \tag{9.9}$$

9.2.1 Partial fraction expansion examples

I have demonstrated in [Bank and Smith 2014; Bank 2018a] that due to the overlapping FIR and IIR parts exactly the same dynamic range problem arises for the partial fraction expansion as with the fixed-pole design shown in Sec. 9.1.1. This can be of course explained by the fact that the filter structure is the same in both cases.

In Fig. 9.3 (a) a direct-form IIR filter is designed by the Steiglitz-McBride method [Steiglitz and McBride 1965] to model a measured anechoic loudspeaker response. This is shown in Fig. 9.3 (a) by a thick gray line, and then converted to a parallel set of second-order sections plus a FIR part by the usual polynomial division partial fraction expansion [Oppenheim et al. 1999]. The orders of the numerator and the denominator are both 20, with the short notation: (20/20). This results in 10 second-order sections plus a constant gain section in parallel. The

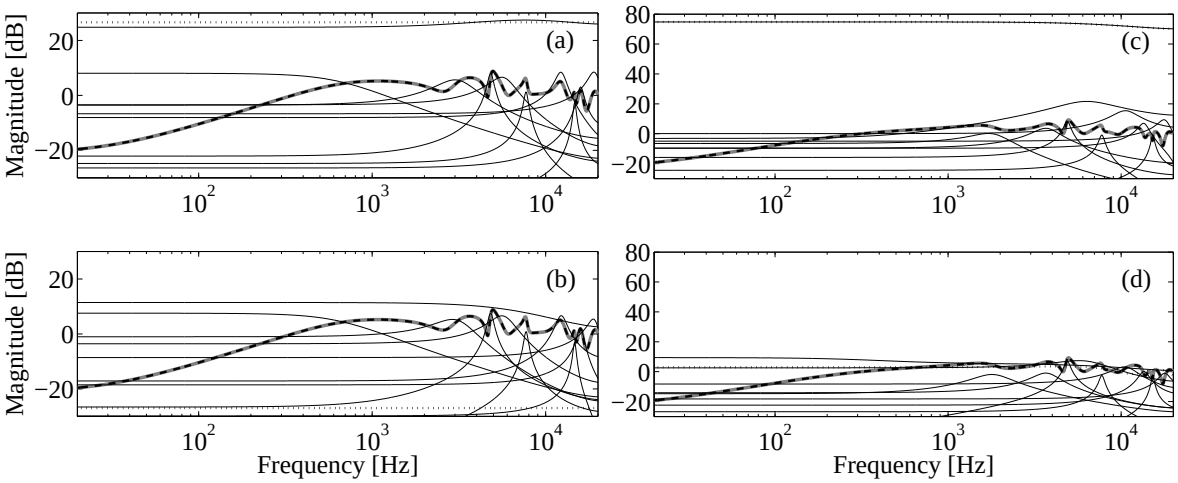


Figure 9.3: Parallel implementation of a (20/20) transfer function: (a) traditional parallel form, (b) delayed parallel form, and parallel implementation of a (25/20) transfer function: (c) traditional parallel form, (d) delayed parallel form. The thick gray line shows the original transfer function and the thick black dashed line displays the net response of the parallel implementation. The thin lines display the magnitude responses of the second-order sections, and the dotted line shows the magnitude response of the FIR path (constant gain in the case of the (20/20) transfer function).

thick black dashed line shows the net transfer function, overlapping the direct-form transfer function (thick gray line) perfectly. The dotted line displays the transfer function of the FIR part $F(z^{-1})$, which is now a constant gain, while the thin lines correspond to the magnitude responses of the individual second-order sections.

It can be seen in Fig. 9.3 (a) that some of the individual transfer functions (in this case, the constant gain part displayed by dotted line and one second-order transfer function displayed by a thin line) are significantly larger than the net transfer function. Figure 9.3 (a) shows only magnitude responses, therefore it cannot be seen that these two upper curves have almost opposite phase, and the required net response is a result of the phase cancellation of these individual responses. This demonstrates that even one sample overlap of the FIR and IIR parts (that is, the constant gain path that arises when converting proper transfer functions having the same numerator and denominator orders) can cause dynamic range limitations.

The problem becomes even more pronounced if the order of the numerator N_{num} is larger than that of the denominator N_{den}, since in that case more samples overlap. This is illustrated in Fig. 9.3 (c) for numerator and denominator orders of 25 and 20, respectively (25/20). It can be seen in Fig. 9.3 (c) that now the fifth-order FIR part (dotted line) is around 70 dB larger than the net transfer function, decreasing the signal-to-noise ratio by 70 dB due to the required downscaling. There is a thin line very close to the dotted line that again corresponds to a second-order transfer function with almost opposite phase compared to the parallel FIR part.

9.2.2 The delayed parallel form

As we have seen Sec. 9.1.4, the dynamic range problem can be completely eliminated if we do not allow any overlap of the FIR and IIR parts by using the delayed parallel form of Eq. (9.1). The parameters of the delayed form can be obtained from the results of the partial fraction expansion using the conversion presented in Sec. 9.1.2.

However, it is also possible to use a different form of partial fraction expansion which gives the parameters of the delayed parallel form directly [Bank and Smith 2014; Bank 2018a]. In [Smith 2007] an alternative partial fraction expansion method has been presented, where the FIR part $\tilde{F}(z^{-1})$ and the strictly proper numerator $\tilde{B}'(z^{-1})$ are computed by performing polynomial long division over the reversed numerator coefficients with the reversed order denominator. (Note that at that time it was not yet known that this alternative variant has more favorable numerical properties and this was first demonstrated in [Bank and Smith 2014].)

The alternative long division results in

$$H(z^{-1}) = z^{-(M+1)} \frac{\tilde{B}'(z^{-1})}{A(z^{-1})} + \tilde{f}_0 + \tilde{f}_1 z^{-1} + \ldots + \tilde{f}_M z^{-M}. \tag{9.10}$$

Finally, the transfer function $\tilde{B}'(z^{-1})/A(z^{-1})$ is expanded to partial fractions by the Heaviside cover up method as in Eq. (9.7) and combined to second-order sections to obtain the delayed parallel form as

$$H(z) = z^{-(M+1)} \sum_{k=1}^{K} \frac{\tilde{b}_{k,0} + \tilde{b}_{k,1} z^{-1}}{1 + a_{k,1} z^{-1} + a_{k,2} z^{-2}} + \sum_{m=0}^{M} \tilde{f}_m z^{-m}. \tag{9.11}$$

Figure 9.3 (b) shows the same (20/20) transfer function as in Fig. 9.3 (a), but now the IIR part is delayed by one sample so that there is no time-domain overlap with the constant gain part. Now the individual transfer functions are only around 5 dB larger than the net transfer function, which is 15 dB smaller compared to that of the traditional parallel form of Fig. 9.3 (a).

Even more pronounced is the difference for the (25/20) transfer function of Fig. 9.3 (c): with the delayed form displayed in Fig. 9.3 (d), the need for downscaling by 70 dB is completely eliminated, leading to a drastic improvement in signal-to-noise ratio.

9.2.3 Obtaining the parallel form by a least-squares fit

For moderate filter orders (< 100), obtaining the parallel form of IIR filters by partial fraction expansion is the most practical option. However, for higher filter orders the conversion can lead to numerical errors, as I have demonstrated in [Bank 2018a].

This is displayed in Fig. 9.4 (a), where thick gray line shows a 200th order IIR filter (200/200) designed by the Steiglitz-McBride method [Steiglitz and McBride 1965] to model a measured room response. The black dashed line is the net transfer function of a delayed parallel filter obtained by the procedure outlined in Sec. 9.2.2, that is, performing polynomial long division on the reversed numerator polynomial and then partial fraction expansion. It can be seen that the magnitude response of the converted filter does not match that of the original, which is due to numerical errors. This is most probably due to the fact that partial fraction expansion involves finding the roots of a polynomial, and root finding is known to be numerically sensitive.

To allow the conversion of high order IIR filters, I have developed a least squares method [Bank 2018a]. This procedure gives the parameters of the delayed parallel form directly and is robust even for very large ($N > 1000$) filter orders. The method is inspired by the fixed-pole design of parallel filters. First the roots of denominator $A(z^{-1})$ are found that are used to form the denominator polynomials of the second-order sections $A_l(z^{-1})$. Next, the numerators of the sections are obtained via a least squares fit such that the difference between the impulse responses of the original and parallel structures is minimized.

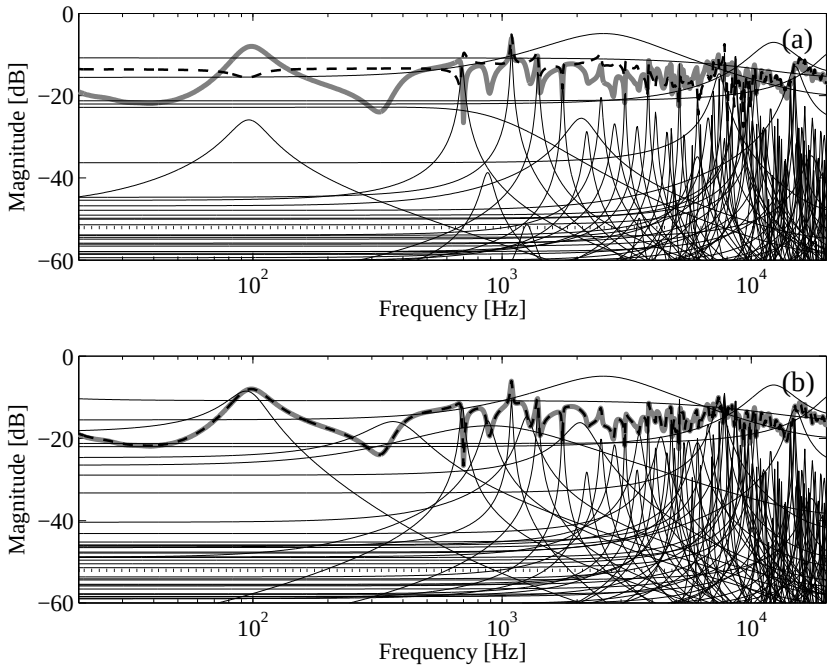


Figure 9.4: Delayed parallel implementation of a (200/200) transfer function: conversion done (a) by partial fraction expansion and (b) by a least squares fit. The thick gray line is the original transfer function and the black dashed line is the net transfer function of the delayed parallel form. The thin lines show the responses of the individual second-order sections and the dotted line displays the transfer function of the constant gain.

While the procedure is also applicable to the traditional, non-delayed parallel form, it will be illustrated for the numerically better performing delayed version. The procedure is outlined for the case of no pole multiplicity. In the case of repeated poles, terms of higher than second-order must also be included, similarly to the case of partial fraction expansion.

The steps of the conversion are the following:

1. Compute the roots p_n of the denominator $A(z^{-1})$, flip the unstable poles $|p_n| > 1$ inside the unit circle by replacing them with $1/p_n$, find the complex-conjugate pairs and recombine the denominators of the second-order sections $A_l(z^{-1})$.

2. Compute the impulse response $h(i)$ of the filter $H(z^{-1}) = B(z^{-1})/A(z^{-1})$ for samples $i = 0 \ldots I$.

3. For $N_{\text{num}} \geq N_{\text{den}}$, the coefficients of the FIR part equal to the first $M = N_{\text{num}} - N_{\text{den}} + 1$ samples of the filter impulse response, that is, $\tilde{f}_k = h(k)$ for $k = 0 \ldots M$.
   For $N_{\text{num}} < N_{\text{den}}$, there is no FIR part.

4. Compute the impulse responses $m_l(i)$ of the numerators $1/A_l(z^{-1}) = 1/(1 + a_{l,1}z^{-1} + a_{l,2}z^{-2})$. either analytically by the inverse z-transform, or by simply "running" the filters by using the recursion

$$m_l(i) = -a_{l,1}m_l(i-1) - a_{l,2}m_l(i-2) + \delta(i), \tag{9.12}$$

   where $\delta(i)$ is the discrete-time unit pulse.

Filter order	Error with PFE conv.	Error with LS conv.
(50/50)	2.36×10^{-10} dB	3.86×10^{-10} dB
(100/100)	7.97×10^{-4} dB	5.52×10^{-8} dB
(200/200)	2.84 dB	6.78×10^{-8} dB
(500/500)	4.45 dB	7.02×10^{-8} dB
(1000/1000)	NaN	1.70×10^{-7} dB
(4000/4000)	NaN	9.17×10^{-7} dB
(8000/8000)	NaN	8.09×10^{-5} dB

Table 9.1: Mean absolute dB errors of transfer functions converted to parallel form by partial fraction expansion and by the proposed least squares method.

5. Find the numerator coefficients $\tilde{b}_{l,0}$, $\tilde{b}_{l,1}$ by a least squares fit such that the resulting impulse response

$$\tilde{h}(i) = \sum_{l=1}^{L} \tilde{b}_{l,0} m_l(i) + \tilde{b}_{l,1} m_l(i-1) \tag{9.13}$$

is closest to the impulse response of the original filter $h(i)$ starting from sample $i = M+1$ (for $N_{\text{num}} < N_{\text{den}}$, starting from sample $i = 0$).

Since (9.13) is linear in its free parameters $\tilde{b}_{l,0}$, $\tilde{b}_{l,1}$, it can be written in a matrix form

$$\tilde{\mathbf{h}} = \mathbf{M}\tilde{\mathbf{p}}, \tag{9.14}$$

where $\mathbf{M}$ contains the impulse responses $m_l(i)$ and their delayed versions $m_l(i-1)$ in its columns, and $\tilde{\mathbf{p}}$ is a column vector composed of the corresponding $\tilde{b}_{l,0}$ and $\tilde{b}_{l,0}$ values. Now the resulting impulse response vector $\tilde{\mathbf{h}}$ should be the closest possible to the target $\mathbf{h}$ vector containing the samples $h(i)$ from $i = M+1$ in the least squares sense. This is again a standard linear least squares problem (outlined in Secs. 2.3 and 6.2.1), with a closed-form solution

$$\tilde{\mathbf{p}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\tilde{\mathbf{h}}. \tag{9.15}$$

Figure 9.4 (b) shows the net transfer function of the delayed parallel form when the conversion is done by the above least-squares fit. As can be seen, now the conversion is much more accurate compared to the one obtained by using partial fraction expansion shown in Fig. 9.4 (a). As for the size of the least-squares problem, the impulse response fit was made for $I = 400$ samples. In general, it is a good practice to choose I such a way that it contains all the important parts of the impulse response, so that $h(i)$ for $i > I$ is negligible.

Table 9.1 lists the mean absolute dB errors computed between the original and converted transfer functions in the range of 20 Hz and 22.05 kHz for various filter orders, including the (200/200) example of Fig. 9.4. The significantly better accuracy of the LS procedure is apparent starting from order 100. For the orders of 1000, 4000, and 8000, some of the extracted poles p_n are outside the unit circle, thus, the PFE based method leads to an unstable filter. On the other hand, the proposed procedure still produces accurate results since it starts with stabilizing the poles by flipping them inside the unit circle in Step 1.

The reason for the significantly better performance compared to partial fraction expansion is that the numerical errors in finding the poles are compensated by the numerators of the second-order sections: the least-squares fit will give the best possible impulse response match for the given (slightly inaccurate) denominators.

It is also worth noting that an apparent disadvantage of the least squares method proposed here over the traditional partial fraction expansion is that it requires significantly more arithmetic operations. However, the dominant source of computational time in direct-to-parallel conversion is finding the roots of $A(z^{-1})$, and compared to that the computational time difference of the two methods becomes negligible.

In some situations the transfer function we are converting to a parallel form is given as a series of second-order sections, or, equivalently, in pole-zero form. Examples include classic low-pass, band-pass, etc. filters such as Butterworth, Chebyshev, etc. [Oppenheim et al. 1999]. (The `butter`, `cheby1`, etc. commands in MATLAB/Octave can give the pole-zero versions of the filters making the implementation possible for such a high-order/low cutoff frequency filters where the direct form implementation is unfeasible due to numerical reasons.) Other examples can be series graphic or parametric equalizers [Välimäki and Reiss 2016] and equalizer filters iteratively designed directly in the series form [Ramos and López 2006] or obtained from a warped IIR design [Tyril et al. 2001].

I have demonstrated in [Bank 2018a] that the above least squares method can also be used to convert from series second-order sections or pole-zero form to the parallel form. It is important that the target impulse response $h(i)$ is computed by running the series version of the filter, and we are not converting the series or pole-zero form to direct form. Also, in this case the numerically problematic root finding is avoided, since either the poles, or the second-order denominators are already known.

9.3 Scientific contributions

Statement 4: I have demonstrated that the delayed parallel filter, where the IIR part is delayed so that there is no overlap with the FIR part results in better numerical properties compared to the traditional parallel form. The results are applicable both for the fixed-pole design of parallel filters and for parallel second-order filters obtained from direct form IIR filters (rational transfer functions) by expansion. For the latter, I have also developed a numerically more robust conversion method.

4.1: I have demonstrated that parallel filters with a FIR path result in a dynamic range limitation due to the need of downscaling of the input signal because of the overlapping FIR and IIR parts, both in the case of fixed-pole design and when obtained by partial fraction expansion. I have shown that this can be avoided by delaying the IIR part. Besides outlining the design of the delayed parallel filter, I have also given the formulas for converting the traditional (non-delayed) parallel form to the delayed form.

4.2: I have developed a method for converting direct form IIR filters to parallel second-order form based on a least-squares design that is numerically better behaving than the partial fraction expansion method and thus allows the conversion of very high order IIR filters ($N > 1000$), which was previously not possible using the common partial fraction expansion technique.

The results have been published in one journal paper [Bank 2018a] and in one conference paper [Bank and Smith 2014][1].

[1] This publication was written together with Prof. Julius Smith. Prof. Smith has been writing the part about partial fraction expansion and has helped me in the general preparation of the manuscript. The new scientific results of the paper related to the scientific statements above and outlined in this Chapter are my own contributions.

Chapter 10

Conclusion

Besides showing my main research contributions, this work has also aimed to give an overview of filter design methods that are targeted towards audio applications. All these methods are able to design filters with nonlinear frequency resolution, and also a common property is that they all use special filter structures instead of the traditional direct-form IIR or FIR filters.

Warped FIR and IIR filters (Chap. 4) arise by substituting the unit delays of normal FIR and IIR filters by a first order allpass, leading to a frequency mapping (warping) which is tweaked by the choice of the allpass pole λ [Härmä et al. 2000]. An advantageous property of warped filters is that they can be designed by common FIR and IIR design techniques, the only difference is that the target response have to be prewarped before filter design.

Kautz filters (Chap. 5) can be seen as the generalization of warped FIR filters where the allpasses can have different poles [Paatero and Karjalainen 2003]. Due to the orthonormality of Kautz basis functions, time-domain design is much simplified, however, this results in a relatively complicated filter structure. Fixed-pole parallel filters (Chap. 6) have a simpler structure composed of parallel second-order sections, while provide the same filtering accuracy as Kautz filters. This comes at the price of giving up the orthonormality of the basis functions, but this is nowadays not a real drawback since a linear least-squares design can be performed fast in today's computers and microprocessors. Also, the orthonormality of Kautz functions can be utilized only in time-domain filter design, while for frequency-domain design and in direct equalizer design a least squares approach should be used for the Kautz filters as well. Orthonormality can be still useful in adaptive filtering [Salama and Cousseau 1998], but for general filtering or equalization tasks the fixed-pole parallel filter is more advantageous due to the fact that it only requires 50 % arithmetic operations on DSPs compared to Kautz filters.

The equivalence of the net transfer functions for Kautz and parallel filters (Sec. 6.4) implies that the pole positioning strategies developed for one of them can also be used for the other. From these, the simplest approach is to set the poles according to the required resolution: for example, a logarithmic set of pole frequencies will result in a logarithmic frequency resolution, and the transfer function of the filter will resemble to the fractional-octave smoothed version of the target frequency response (Sec. 6.5). The fit can be improved at a given filter order by the use of more complex pole positioning strategies based on single-band, dual-band or custom warped IIR filter design, or by automatically placing the poles according to the ripple density of the smoothed transfer function (Chap. 7).

The reader might wonder which method should be used among the various possibilities. Of course there is no single answer. However, I believe that a good starting point is the design of parallel filters with predetermined (e.g., logarithmic) pole set, which already achieves signifi-

cantly better performance compared to traditional FIR and IIR filter design approaches, while still being very simple both in its concept and implementation. The design accuracy can be improved even more by the use of more sophisticated pole positioning techniques using multiband or custom warped design. This of course leads to a somewhat more complicated design procedure, therefore, it can be considered as a next step in audio filter or equalizer design.

Besides outlining the single-channel approaches, this work also extended some of the methods to the MIMO case (Chap. 8). The passive design variant allows the design of positive semidefinite admittance matrices that can be used to model measured responses in the full audio frequency range while still guaranteeing the passivity of the model. A direct design approach was also presented for the MIMO equalization case allowing the design of an IIR MIMO equalizer in an equally simple way as the common FIR equalizers, but allowing the use of arbitrary frequency resolution similarly to the single-channel parallel filter.

Finally, we have seen in Chap. 9 that when there is an FIR path in parallel with the second-order sections, the individual transfer functions can be much larger than the net transfer function. In practical implementations this leads to the loss of useful bitdepth due to the downscaling required to avoid overflow. The same problem arises both in the fixed-pole design of parallel filters and when the second-order sections are obtained from direct-form filters by partial fraction expansion. This limitation can be simply avoided by the use of the delayed parallel form where there is no overlap between the FIR part and the response of the second-order sections. In addition, a numerically robust conversion technique was also presented that allows the direct to parallel conversion of filters of with orders in the range of thousand which was not possible previously.

10.1 Further results in the field of audio filter design

I have developed several additional methods in the field of audio filter design that are not included in the list of my scientific statements, although often apply the techniques presented in the statements or otherwise closely related to the topic of the thesis. Some of them are a result of collaboration with other researchers; this will be clearly marked by using the article “we” and can also be seen from the corresponding references containing other names than “Bank”. Besides audio filter design, my other research field is sound synthesis; these less related contributions are not listed in the following, but the interested reader is referred to the list of my publications.

10.1.1 Implementation aspects of fixed-pole parallel filters

Some of the publications are directly related to fixed-pole parallel filters: in [Bank and Horváth 2017a] we have compared the quantization noise performance of warped IIR and fixed-pole parallel filters. For the warped IIR filter we have considered both the special filter structure and the more efficient series second-order implementation (these have been discussed in Sec. 4.4). We have demonstrated with simulation examples that while the series second-order implementation of warped IIR filters has the same computational complexity as that of fixed-pole parallel filters, parallel filters have a significantly lower quantization noise.

To decrease the quantization noise of fixed-pole parallel filters, in [Bank and Horváth 2017b] we have proposed using a special warped structure for the most problematic low-frequency second-order sections. The warping parameter λ is different for each section and it is set so that

it minimizes the quantization noise produced by that section. For choosing λ a simple analytical formula was also given that approximates the numerically optimal λ values very accurately.

In [Horváth and Bank 2019] we have systematically analyzed a handful set of second-order sections in terms of numerical performance and computational complexity. Besides the usual direct form and transposed structures, we have tested the Gold & Rader [Rader and Gold 1967], the Kingsbury [Kingsbury 1972], the Chamberlin [Chamberlin 1985; Smith III 2019], the Zölzer [Zölzer 1994] and the above optimized warped IIR (WIIR) second-order structure [Bank and Horváth 2017b]. We have also proposed an extension to the Chamberlin state variable filter so that it can be used as a general IIR filter. It had turned out that exactly this filter has the best performance in terms of quantization noise for the problematic low-frequency poles. Therefore, we have proposed the use of the extended Chamberlin structure for the low frequency poles, while staying with the more efficient direct form 1 (DF1) structure for the rest. In our test case this has increasesed the signal-to-noise ratio by a quite significant 40 dB at the price of 50% increase in total computational cost.

Still related to the implementation of fixed-pole parallel filters, in [Belloch et al. 2014] we have shown that the inherently parallel nature of the structure translates to a very high efficiency in parallel architectures (such as Graphic Processing Units), on the contrary to direct form or series IIR filters. By using an Nvidia TeslaK20c GPU more than thousand filter paths at a filter order of 256 could be run in real-time. This can find its application in the equalization of large MIMO systems, e.g., in the full equalization of wave field synthesis [Fuster et al. 2005; Gauthier and Berry 2007].

10.1.2 Magnitude-priority filter design

As discussed in Sec. 2.4, in audio an accurate magnitude modeling or equalization is more important than that of the phase, and this is often reflected by the choice of minimum-phase filter design generally resulting in better magnitude accuracy for the same filter order. However, this also means giving up phase modeling all together. In [Bank 2012b, 2014] I have proposed an alternative approach called “magnitude-priority filter design” where the filter follows both the magnitude and phase response in those frequency regions where it can, while where this is not possible, it gives priority to the magnitude. I have developed two variants, one updating the phase of the target response, while the other updating the magnitude response in each iteration. The method is applicable to any filter design technique that works in the time-domain or uses a complex target response in the frequency-domain. While [Bank 2012b] contains only fixed-pole parallel filter design examples, in [Bank 2014] I have also demonstrated the technique using a windowed FIR filter design and a warped IIR design.

For rectangular rooms with symmetric loudspeaker arrangements, full room equalization can be achieved at low frequencies by generating a plane wave that propagates along the room [Santillian 2001; Santillian et al. 2007]. However, often the room is not rectangular, and/or a symmetric loudspeaker setup cannot be assured, leading to a deteriorated equalization performance. I have addressed this problem in [Bank 2012a] with the multichannel extension of the above “magnitude-priority” approach where the magnitude specification is kept constant in the control points, while the phase is determined by an iterative optimization process starting from the plane wave solution.

10.1.3 Combined quasi-anechoic and in-room equalization of loudspeakers

Equalization of a loudspeaker response based on anechoic measurements is a relatively simple task and it can be used to improve both the time- and frequency-domain response of the loudspeaker. When this speaker is put in a room, such an equalization improves the direct sound first reaching the listener. Equalizing the response of the loudspeaker based on in-room measurements is much more problematic due to the fact that such responses are highly non-minimumphase and position dependent (see also the discussion in Sec. 3.1). To address this problem, in [Bank 2013b] I have proposed a combined approach where the loudspeaker is first equalized based on the quasi-anechoic response obtained from a windowed in-room measurement and then by a room equalizer addressing the more problematic low-frequency region of room responses and correcting the errors of windowing. For this latter part I have used an equalizer based on the fixed-pole parallel filter. Later we have also extended the approach to the case of multipoint equalization in [Cecchi et al. 2014].

10.1.4 Highly accurate graphic equalizers using parallel second-order filters

Graphic equalizers are important tools in audio that are used not only by sound engineers and music producers but also by end users since they provide simple means of modifying the loudness of various frequency components. Traditional graphic equalizers are composed of a set of second-order filters with logarithmically spaced center frequencies and fixed Q values, where the gain of the sections is varied by the user. A difficulty in both analog and digital graphic equalizer design is that due to the interaction of the neighboring bands, the net response will be different from what is actually set by the user. To overcome this, in [Rämö et al. 2014] we have proposed an alternative approach where a precise magnitude target curve is composed based on the user settings with a suitable interpolation and a fixed-pole parallel filter is designed using this target response. With this approach a much more accurate equalizer response was achieved compared to earlier approaches. This came at a price of increased design complexity which can be a limitation if we wish to follow the user changing the sliders continuously. To eliminate this drawback, we have developed an efficient design variant in [Bank et al. 2017] where both the target magnitude and phase response are generated as a linear combination of elementary minimum-phase functions decreasing the design time drastically. Another part of speed-up comes from an alternate formulation of the weighted least-squares problem that we have also published in [Belloch et al. 2017]. Finally, we have converted the cascade form graphic equalizer of [Välimäki and Liski 2017] to the delayed parallel form in [Liski et al. 2019], and also provided an efficient method for converting any filter available in the cascade second-order form to the numerically more advantageous delayed parallel form.

10.1.5 Modeling of nonlinear systems

So far we have assumed the linearity of the systems that are equalized or modeled. However, in some cases the nonlinearity cannot be neglected: in the audio field such devices include tube amplifiers, distortion circuits or speakers driven to their nonlinear range as often done by guitar players. In [Yeh et al. 2008] we have been modeling the nonlinear behavior of a guitar cabinet by using a static nonlinearity applied to the estimated loudspeaker displacement, followed by a

linear filter implemented using fixed-pole parallel filters. A common-pole single-input multiple-output filter structure was used (outlined in Sec. 8.1.1) to allow interpolation between various microphone positions.

In black-box modeling Wiener and Hammerstein models are commonly used since they can be indentified relatively easily compared to more complex models (e.g., Wiener–Hammerstein or Volterra models). Polynomial Hammerstein models compose their response as a sum of parallel branches each having a static nonlinearity (a polynomial) and a linear filter in series. In [Bank 2011a] I have proposed a highly efficient implementation of such models by the use of multiple-input single-output parallel filters with common poles (see Sec. 8.1.1) and applied it to the nonlinear modeling of a loudspeaker. We have made an extensive comparison of the technique with a polynomial Hammerstein model using FIR filters in [Romoli et al. 2014]. We have demonstrated that the proposed approach has a significantly reduced computational complexity for similar modeling performance compared to the common FIR approach.

10.1.6 Other related results

For testing the MIMO equalization approach using parallel filters shown in Sec. 8.3.3, I have used the multichannel measurement of a spherical loudspeaker. During the measurements it had turned out that the results are highly sensitive to the positioning of the loudspeaker that cannot be controlled as accurately as desired due to physical constraints. To overcome this, we have proposed a calibration technique in [Zotter and Bank 2012] that estimates the positioning and rotation errors solely by using the measured responses and applies a suitable compensation to the results, thus, removes the effects of geometric inaccuracies in the measurement.

In [Ramos et al. 2017] we have applied a special form of the parallel second-order structure to the modeling of head-related transfer functions (HRTFs). Coming from the special structure, the parameters were estimated using an iterative optimization scheme instead of the usual least-squares design. The benefit of the approach compared to the general parallel filter is that by this way HRTFs can be accurately interpolated for those angles where measurements are not available.

10.2 Significance and applications

Here I give a brief summary of the significance and applications of the results included in the set of my scientific statements and discussed in this thesis in detail.

10.2.1 Fixed-pole design of parallel filters

As I see it, the fixed-pole design of parallel filters is a very efficient methodology to design IIR filters at arbitrary (non-uniform) frequency resolution, of which logarithmic scale has been used throughout this thesis since that is the one most often used in audio applications. Compared to the quasi-logarithmic frequency resolution filter design methods proposed previously (warped and Kautz filters), the efficiency of the approach comes from two factors. First, by the choice of a simpler filter structure, the computational complexity is reduced for the same filter order. Second, when used with the pole positioning methods I have developed, the parallel filter results in lower approximation or equalization error compared to earlier approaches for the same filter order. I find it important to emphasize that this improved efficiency is not coming at the price of

a highly complicated structure or design method: on the contrary, the filter structure is actually simplified and the design still uses the least-squares minimization common in the literature.

Besides introducing the basic technique, by showing the smoothing properties, equivalence to Kautz filters, and developing pole positioning methods I have developed a complete toolset that can be used for a wide range of applications.

Most applications that I have developed myself or with other researchers are related to the equalization of loudspeaker–room responses [Bank 2008, 2011b; Bank and Ramos 2011; Bank 2011c, 2013a,b,c; Ramos and Bank 2013] and to the modeling or equalization of anechoic loudspeaker responses [Yeh et al. 2008; Bank 2013c]. Recently, Kamaris et al. [2021] have applied the fixed pole parallel filter design to the equalization of mobile earphones.

Another field with several application examples that I have developed is sound synthesis. This includes the modeling of the radiation response of the piano soundboard [Bank 2007; Bank et al. 2010], modeling the one-dimensional admittance of a guitar bridge [Bank and Karjalainen 2008], simulation of the sustain pedal effect for the piano [Zambon et al. 2008], and the parametric resynthesis of piano sounds for listening tests [Bank and Lehtonen 2010].

Graphic equalizers are important tools in music production and in consumer audio: recently we have been applying the fixed-pole design of parallel filters to obtain a highly accurate but still efficient third-octave graphic equalizer [Rämö et al. 2014; Bank et al. 2017], as already mentioned in Sec. 10.1.4

I have applied the MIMO extension of parallel filters to the modeling of the 2D admittance matrix of a guitar bridge for sound synthesis [Bank and Karjalainen 2010] as discussed in Sec. 8.2, and the same filter structure with an improved parameter estimation algorithm has been applied to the modeling of the violin, viola, and cello in [Maestre et al. 2013, 2017]. In [Maestre et al. 2021] the warped common-pole estimation technique of Sec. 8.2.4 has been used to parameterize a state space radiation model.

I have also applied the multichannel design of parallel filters to the computationally efficient modeling of nonlinear loudspeaker behavior [Bank 2011a; Romoli et al. 2014] mentioned in Sec. 10.1.5 and to the common-pole modeling of piano soundboard response first proposed in [Bank 2007; Bank et al. 2010], and further developed by Zambon [2013]; Gabrielli et al. [2015]. The direct equalizer design for MIMO systems was used to equalize the velocity responses of a 20 channel spherical loudspeaker array in [Bank 2018b], as shown in Sec. 8.3.3.

Coming from both the simplicity and efficiency of the approach, parallel filters have also found industrial applications. Where I have also taken part of is the soundboard modeling in the Physis piano of the Viscount corporation [Bank et al. 2010; Zambon et al. 2016]. Since companies rarely publish the methods they are using, I mostly know from informal discussions at conferences or by email inquiries that my methods have been used in some products. Applications that are already developed and working include the Python script for room equalization using fixed-pole parallel filters [Green 2012] and dual-band warping used to calibrate a loudspeaker response in the Audio Precision APx500 measurement system [Kite 2013]. From email inquiries it turned out that Antelope audio was interested in using parallel filters for the modeling of microphone transfer functions and guitar loudspeaker responses, although I have no information if they have finally came out with a product using the techniques I have developed [Levin 2014].

Instead of listing possible further applications, it can be safely said that the methods I have developed can be used for various audio applications very effectively. This is because most audio related filter or equalizer design or tasks benefit from using logarithmic frequency resolution, and I believe that the fixed-pole parallel filter design is the most efficient logarithmic

frequency resolution methodology at present date.

Since I have been working in the field of audio signal processing the applications I can mostly think of are related to audio. Nevertheless, the non-uniform frequency resolution achievable by parallel filters could most likely be used also in other fields. For example, in frequency-domain system identification the measured frequency response is often available in logarithmic frequency scale [Pintelon et al. 1994], where some of the methods I have developed could possibly be also applied. However, this is left for future research.

10.2.2 Converting IIR filters to parallel form

While the fixed-pole design of parallel filters have been developed with the requirements of audio signal processing in mind, my results about converting direct form or series second-order filters to the parallel form are useful for the broad signal processing community. For the conversion the partial fraction expansion using the Heaviside cover-up method is the most commonly used and actually this is the only one covered in the DSP textbooks I am aware of. Therefore I believe that showing that this common method can lead to dynamic range limitations in practical implementations and providing a solution to the problem is my most widely applicable result. In addition, the proposed LS method allows the direct to parallel conversion of very high ($N > 1000$) order transfer functions that was not possible previously.

I have applied the technique to loudspeaker response modeling, piano soundboard modeling and implementing a Butterworth high-pass in the parallel form in [Bank 2018a]. We are also using the delayed parallel form in a recent graphic equalizer design [Liski et al. 2019]. In [Kereliuk et al. 2018] the delayed parallel form is used to model room impulse responses, and in [Schlecht and Habets 2019; Schlecht 2020] for the computation of the residues of a feedback delay network (FDN). In addition, the FDN toolbox [Schlecht 2020] applies the proposed LS method for partial fraction expansion.

As for industrial applications, I am aware from an email inquiry that the world's largest pro-audio company, Music Tribe Inc., has been testing my method for converting series form transfer function to parallel form with very satisfying results [Christensen 2018]. Implementing IIR filters in the parallel form rather than in series is gaining more and more popularity due to the availability of processors capable of parallel processing (such as GPU-s), thus, I expect more and more actual applications to come.

10.2.3 MATLAB/Octave code related to the presented algorithms

MATLAB/Octave codes for designing parallel filters both in the time- and frequency-domain can be downloaded from `http://www.mit.bme.hu/~bank/parfilt`. The page also includes scripts for direct equalizer design. Perhaps the most interesting files are `parfiltdemo.m` and `parfeqdemo.m` that are interactive applications where the parallel filter is designed in real-time according to the poles positioned by the user.

As for the delayed parallel filter, the scripts are downloadable from `http://www.mit.bme.hu/~bank/parconv`. These include both the fixed-pole design of the parallel filter and the codes required to convert direct-form IIR filters to the parallel form. Besides the alternative partial fraction expansion variant `tf2delparf.m` the code for the numerically more robust LS conversion is also provided `tf2delparf_ls.m`.

Bibliography

My own contributions

Bank, B. (2007). Direct design of parallel second-order filters for instrument body modeling, *Proc. Int. Computer Music Conf.*, Copenhagen, Denmark, pp. 458–465.

Bank, B. (2008). Perceptually motivated audio equalization using fixed-pole parallel second-order filters, *IEEE Signal Process. Lett.* **15**: 477–480.

Bank, B. (2010). Audio equalization with fixed-pole parallel filters: An efficient alternative to complex smoothing, *Proc. 128*th *AES Conv., Preprint No. 7965*, London, UK.

Bank, B. (2011a). Computationally efficient nonlinear Chebyshev models using fixed-pole parallel filters with the application to loudspeaker modeling, *Proc. 130*th *AES Conv., Preprint No. 8416*, London, UK.

Bank, B. (2011b). Logarithmic frequency scale parallel filter design with complex and magnitude-only specifications, *IEEE Signal Process. Lett.* **18**(2): 138–141.

Bank, B. (2011c). Warped IIR filter design with custom warping profiles and its application to room equalization, *Proc. 130*th *AES Conv., Preprint No. 8415*, London, UK.

Bank, B. (2012a). Full room equalization at low frequencies with asymmetric loudspeaker arrangements, *Proc. 132*nd *AES Conv., Preprint No. 8593*, Budapest, Hungary.

Bank, B. (2012b). Magnitude-priority filter design for audio applications, *Proc. 132*nd *AES Conv., Preprint No. 8591*, Budapest, Hungary.

Bank, B. (2013a). Audio equalization with fixed-pole parallel filters: An efficient alternative to complex smoothing, *J. Audio Eng. Soc.* **61**(1/2): 39–49.

Bank, B. (2013b). Combined quasi-anechoic and in-room equalization of loudspeaker responses, *Proc. 134*nd *AES Conv., Preprint No. 8826*, Rome, Italy.

Bank, B. (2013c). Loudspeaker and room equalization using parallel filters: Comparison of pole positioning strategies, *Proc. 51*st *AES Conf. on Loudspeakers and Headphones*, Helsinki, Finland.

Bank, B. (2014). Magnitude-priority filter design, *J. Audio Eng. Soc.* **62**(7/8): 485–492.

Bank, B. (2018a). Converting infinite impulse response filters to parallel form, *IEEE Signal Process. Mag.* **35**(3): 124–130.

Bank, B. (2018b). Multichannel equalization and crosstalk cancellation using fixed-pole IIR filters, *J. Audio Eng. Soc.* **66**(11): 901–909.

Bank, B. and Horváth, K. (2017a). Quantization noise of warped and parallel filters using floating point arithmetic, *Proc. 142*nd *AES Conv., eBrief No. 337*.

Bank, B. and Horváth, K. (2017b). Warped implementation of parallel second-order filters with optimized quantization noise performance, *Proc. 142*nd *AES Conv., eBrief No. 338*.

Bank, B. and Karjalainen, M. (2008). Passive admittance synthesis for sound synthesis applications, *Proc. Acoustics'08 Paris Conf.*, Paris, France.

Bank, B. and Karjalainen, M. (2010). Passive admittance matrix modeling for guitar synthesis, *Proc. Conf. on Digital Audio Effects*, Graz, Austria, pp. 3–7.

Bank, B. and Lehtonen, H.-M. (2010). Perception of longitudinal components in piano string vibrations, *J. Acoust. Soc. Am. Exp. Lett.* **128**(3): EL117–EL128.

Bank, B. and Ramos, G. (2011). Improved pole positioning for parallel filters based on spectral smoothing and multi-band warping, *IEEE Signal Process. Lett.* **18**(5): 299–302.

Bank, B. and Smith, J. O. (2014). A delayed parallel filter structure with an FIR part having improved numerical properties, *Proc. 136*th *AES Conv., Preprint No. 9084*, Berlin, Germany.

Bank, B., Belloch, J. A. and Välimäki, V. (2017). Efficient design of a parallel graphic equalizer, *J. Audio Eng. Soc.* **65**(10): 817–825.

Bank, B., Zambon, S. and Fontana, F. (2010). A modal-based real-time piano synthesizer, *IEEE Trans. Audio, Speech, and Lang. Process.* **18**(4): 809–821. URL: `http://home.mit.bme.hu/ ∼bank/publist/taslp-piano/`.

Belloch, J. A., Bank, B., Igual, F. D., Quintana-Ortí, E. S. and Vidal, A. M. (2017). Solving weighted least squares (WLS) problems on ARM-based architectures, *The Journal of Supercomputing* **73**(1): 530–542.

Belloch, J. A., Bank, B., Savioja, L., Gonzalez, A. and Välimäki, V. (2014). Multi-channel IIR filtering of audio signals using a GPU, *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, Florence, Italy, pp. 6692–6696.

Cecchi, S., Romoli, L., Piazza, F., Bank, B. and Carini, A. (2014). A novel approach for prototype extraction in a multipoint equalization procedure, *Proc. 136*th *AES Conv., Preprint No. 9048*, Berlin.

Horváth, K. and Bank, B. (2019). Optimizing the numerical noise of parallel second-order filters in fixed-point arithmetic, *J. Audio Eng. Soc.* **67**(10): 763–771.

Liski, J., Bank, B., Smith, J. O. and Välimäki, V. (2019). Converting series biquad filters into delayed parallel form: Application to graphic equalizers, *IEEE Trans. Signal Process.* **67**(14): 3785–3795.

Parallel filter homepage (2021). MATLAB/Octave scripts for the fixed-pole design of parallel filters. URL: `http://www.mit.bme.hu/∼bank/parfilt`.

Ramos, G. and Bank, B. (2013). Low computational cost equalization and modeling of audio systems, *Proc. TecniAcustica*, Valladolid, Spain.

Ramos, G., Cobos, M., Bank, B. and Belloch, J. A. (2017). A parallel approach to HRTF approximation and interpolation based on a parametric filter, *IEEE Signal Process. Mag.* **24**(10): 1507–1511.

Rämö, J., Välimäki, V., and Bank, B. (2014). High-precision parallel graphic equalizer, *IEEE Trans. Audio, Speech, and Lang. Process.* **22**: 1894–1904.

Romoli, L., Cecchi, S., Bank, B., Gasparini, M. and Piazza, F. (2014). Application of common-pole parallel filters to nonlinear models based on orthogonal functions, *Proc. 136*$^{\text{th}}$ *AES Conv., Preprint No. 9068*, Berlin.

Yeh, D., Bank, B. and Karjalainen, M. (2008). Nonlinear modeling of a guitar loudspeaker cabinet, *Proc. Conf. on Digital Audio Effects*, Espoo, Finland, pp. 89–96.

Zambon, S., Giordani, E., Fontana, F. and Bank, B. (2016). System to reproduce the sound of a stringed instrument. US Patent 9,293,126 B2.

Zambon, S., Lehtonen, H.-M. and Bank, B. (2008). Simulation of piano sustain-pedal effect by parallel second-order filters, *Proc. Conf. on Digital Audio Effects*, Espoo, Finland, pp. 199–204.

Zotter, F. and Bank, B. (2012). Geometric error estimation and compensation in compact spherical loudspeaker array calibration, *Proc. Int. Instrumentation and Meas. Conf. (IMTC12)*, Graz, Austria, pp. 2710–2715.

Other references

Adams, J. W. and Sullivan, J. L. (1998). Peak-constrained least-squares optimization, *IEEE Trans. Signal Process.* **16**(2): 306–321.

Ahuja, S. S. and Roy, S. C. D. (1980). Variable digital filters, *IEEE Trans. Circ. Syst.* **CAS-27**(9): 836–838.

Anderson, B. D. O. and Vongpanitlerd, S. (1973). *Network Analysis and Synthesis*, Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Behrends, H., von dem Knesebeck, A., Bradinal, W., Neumann, P. and Zölzer, U. (2011). Automatic equalization using parametric IIR filters, *J. Audio Eng. Soc.* **59**(3): 102–109.

Beliczynski, B., Kale, J. and Cain, G. D. (1992). Approximation of FIR by IIR filters: An algorithm bases on balanced model reduction, *IEEE Trans. Signal Process.* **40**: 532–542.

Benesty, J., Chen, J., Huang, Y. A. and Dmochowski, J. (2007). On microphone-array beamforming from a MIMO acoustic signal processing perspective, *IEEE Trans. Audio, Speech, and Lang. Process.* **15**(3): 1053–1065.

Betlehem, T., Zhang, W., Poletti, M. A. and Abhayapala, T. D. (2015). Personal sound zones: Delivering interface-free audio to multiple listeners, *IEEE Signal Process. Mag.* **32**(2): 81–91.

Brandenstein, H. and Unbehauen, R. (1998). Least-squares approximation of FIR by IIR digital filters, *IEEE Trans. Signal Process.* **46**(1): 21–30.

Brännmark, L.-J. and Ahlén, A. (2015). Multichannel room correction with focus control, *J. Audio Eng. Soc.* **63**(1/2): 21–30.

Broome, P. W. (1965). Discrete orthonormal sequences, *J. of the Association for Computing Machinery* **12**(2): 151–168.

Cecchi, S., Carini, A. and Spors, S. (2018). Room response equalization – A review, *Appl. Sci.* article 16.

Chamberlin, H. (1985). *Musical Applications of Microprocessors*, Hayden Books, USA.

Chen, W. (1996). Performance of cascade and parallel IIR filters, *J. Audio Eng. Soc.* **44**(3): 148–158.

Christensen, K. B. (2018). Personal communication (email). Music Tribe Inc.

Constantinides, A. G. (1970). Spectral transformations for digital filters, *Proc. IEE* **117**: 1585–1590.

Craven, P. G. and Gerzon, M. A. (1992). Practical adaptive room and loudspeaker equalizer for hi-fi use, *Proc. 92nd AES Conv., Preprint No. 3346*, Vienna, Austria.

Ebri, M., Strozzi, N., Fazi, F. M., Farina, A. and Cattani, L. (2020). Individual listening zone with frequency-dependent trim of measured impulse responses, *Proc. 149rd AES Conv., Preprint No. 10409*, Presented Online.

Evangelista, G. and Raspaud, M. (2009). Simplified guitar bridge model for the displacement wave representation in digital waveguides, *Proc. Conf. on Digital Audio Effects*, Como, Italy, pp. 484–491.

Ewins, D. J. (1992). *Modal Testing: Theory and Practice*, Research Studies Press, Taunton, UK.

Fuster, L., López, J. J., González, A. and Faus, P. (2005). Time and frequency domain room compensation applied to wave field synthesis, *Proc. Conf. on Digital Audio Effects*, Madrid, Spain, pp. 7–12.

Gabrielli, L., Zambon, S. and Fontana, F. (2015). Parallel digital signal processing for efficient piano synthesis, *Proc. 23th Eur. Sign. Proc. Conf. (EUSIPCO)*, Nice, France.

Gauthier, P.-A. and Berry, A. (2007). Adaptive wave field synthesis for sound field reproduction: Theory, experiments, and future perspectives, *J. Audio Eng. Soc.* **55**(12): 1107–1124.

Golub, G. H. and Loan, C. F. V. (2013). *Matrix Computations*, 4th edn, The John Hopkins University Press, Baltimore, MD, USA.

Green, M. A. (2012). Python open room correction. Open source Python project for miniDSP OpenDRC, URL: https://www.minidsp.com/applications/advanced-tools/python-open-room-correction.

Haneda, Y. and Kaneda, Y. (1994). Common acoustical pole and zero modeling of room transfer functions, *IEEE Trans. Speech Audio Process.* **9**(2): 320–328.

Hatziantoniou, P. D. and Mourjopoulos, J. N. (2000). Generalized fractional-octave smoothing of audio and acoustic responses, *J. Audio Eng. Soc.* **48**(4): 259–280.

Higham, N. J. (1988). Computing a nearest symmetric positive semidefinite matrix, *Elsevier Lin. Alg. and Appl.* **103**: 103–118.

Hollebon, J., Fazi, F. M. and Simón Gálvez, M. F. (2021). A multiple listener crosstalk cancellation system using loudspeaker dependent regularization, *J. Audio Eng. Soc.* **69**(3): 191–203.

Härmä, A. and Paatero, T. (2001). Discrete representation of signals on a logarithmic frequency scale, *Proc. IEEE Workshop Appl. of Signal Process. to Audio and Acoust.*, New Paltz, NY, USA.

Härmä, A., Karjalainen, M., Savioja, L., Välimäki, V., Laine, U. K. and Huopaniemi, J. (2000). Frequency-warped signal processing for audio applications, *J. Audio Eng. Soc.* **48**(11): 1011–1031.

Huang, Y., Benesty, J. and Chen, J. (2007). On crosstalk cancellation and equalization with multiple loudspeakers for 3-D sound reproduction, *IEEE Signal Process. Lett.* **4**(10): 649–652.

Huang, Y., Benesty, J. and Chen, J. (2008). Generalized crosstalk cancellation and equalization using multiple loudspeakers for 3D sound reproduction at the ears of multiple listeners, *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, Las Vegas, NV, USA, pp. 405–408.

Jackson, L. B. (2008). Frequency-domain Steiglitz-McBride method for least-squares filter design, ARMA modeling, and periodogram smoothing, *IEEE Signal Process. Lett.* **15**: 49–52.

Jeong, J., Lee, J., Park, Y., Kim, J. and Youn, D. (2005). Design and implementation of IIR crosstalk cancellation filters approximating frequency warping, *Proc. 118$^{\text{th}}$ AES Conv., Preprint No. 6490*, Barcelona, Spain.

Johnson, D. H. (1979). Variable digital filters having a recursive filter structure, *IEEE Trans. Acoust. Speech Signal Process.* **1**: 98–99.

Jury, E. I. (1974). *Inners and Stability of Dynamic Systems*, John Wiley and Sons, New York, USA.

Kamaris, G., Zachos, P. and Mourjopoulos, J. (2021). Low filter order digital equalization for mobile device earphones, *J. Audio Eng. Soc.* **69**(5): 297–308.

Karjalainen, M. and Paatero, T. (2001). Frequency-dependent signal windowing, *Proc. IEEE Workshop Appl. of Signal Process. to Audio and Acoust.*, New Paltz, NY, USA, pp. 35–38.

Karjalainen, M. and Paatero, T. (2007). Equalization of loudspeaker and room responses using Kautz filters: Direct least squares design, *EURASIP J. on Advances in Sign. Proc., Spec. Iss. on Spatial Sound and Virtual Acoustics* **2007**: 13. Article ID 60949, doi:10.1155/2007/60949.

Karjalainen, M., Härmä, A. and Laine, U. K. (1997). Realizable warped IIR filters and their properties, *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, Vol. 3, Munich, Germany, pp. 2205–2208.

Karjalainen, M., Paatero, T., Mourjopoulos, J. N. and Hatziantoniou, P. D. (2005). About room response equalization and dereverberation, *Proc. IEEE Workshop Appl. of Signal Process. to Audio and Acoust.*, New Paltz, NY, USA, pp. 183–186.

Kautz, W. H. (1954). Transient synthesis in the time domain, *IRE Trans. on Circuit Theory* **1**(3): 29–39.

Kereliuk, C., Herman, W., Wedelich, R. and Gillespie, D. J. (2018). Modal analysis of room impulse rsponses using subband ESPRIT, *Proc. Conf. on Digital Audio Effects,* Aveiro, Portugal.

Kingsbury, N. (1972). Second-order recursive digital-filter element for poles near the unit circle and the real z axis, *Elec. Lett.* **8**(6): 155–156.

Kirkeby, O. and Nelson, P. A. (1999). Digital filter design for inversion problems in sound reproduction, *J. Audio Eng. Soc.* **47**(7/8): 583–595.

Kirkeby, O., Nelson, P. A., Hamada, H. and Orduna-Bustamante, F. (1998). Fast deconvolution of multichannel systems using reguralization, *IEEE Trans. Speech Audio Process.* **6**(2): 189–194.

Kirkeby, O., Rubak, P., Johansen, L. G. and Nelson, P. A. (1999). Implementation of cross-talk cancellation networks using warped FIR filters, *Proc. AES 16*$^{\text{th}}$ *Int. Conf.*, Rovaniemi, Finland, pp. 358–365.

Kite, T. (2013). Personal communication (email). Audio Precision Inc.

Kobayashi, T. and Imai, S. (1990). Design of IIR digital filters with arbitrary log magnitude function by WLS techniques, *IEEE Trans. Acoust. Speech Signal Process.* **38**(2): 247–252.

Krukowski, A., Kale, I. and Cain, G. D. (1996). Decomposition of IIR transfer functions into parallel arbitrary-order IIR subfilters, *Proc. IEEE Nordic Signal Processing Symposium (NORSIG'96)*, Espoo, Finland.

Kurosu, A., Miyase, S. and Tomiyama, S. (2003). A technique to truncate IIR filter impulse response and its application to real-time implementation of linear-phase filters, *IEEE Trans. Signal Process.* **51**(5): 1284–1292.

Lambourg, C. and Chaigne, A. J. (1993). Measurements and modeling of the admittance matrix at the bridge in guitars, *Proc. Stockholm Music Acoust. Conf.*, Stockholm, Sweden, pp. 448–453.

Lavandier, M., Meunier, S. and Herzog, P. (2008). Identification of some perceptual dimensions underlying loudspeaker dissimilarities, *J. Acoust. Soc. Am.* **123**(6): 4186–4198.

Levin, I. (2014). Personal communication (email). Antelope Audio Inc.

Lipschitz, S. P., Scott, T. C. and Vanderkooy, J. (1985). Increasing the audio measurement capability of FFT analyzers by microprocessor postprocessing, *J. Audio Eng. Soc.* **33**(9): 626–648.

Liski, J., Mäkivirta, A. and Välimäki, V. (2018). Audibility of loudspeaker group-delay characteristics, *Proc. 144*th *AES Conv., Preprint No. 10008*, Milano, Italy.

Liu, C. J. and Hsieh, S. F. (1988). Common-acoustic-poles/zeros approximation of head-related transfer functions, *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, Vol. 5, Salt Lake City, UT, USA, pp. 3341–3344.

Maeng, S. J. and Lee, B. G. (1995). A design of linear-phased IIR nyquist filters, *IEEE Trans. on Selected Areas in Communications* **13**(1): 167–175.

Maestre, E., Scavone, G. P. and Smith, J. O. (2013). Digital modeling of bridge driving-point admittances from measurements on violin-family instruments, *Proc. Stockholm Music Acoust. Conf.*, Stockholm, Sweden, pp. 101–108.

Maestre, E., Scavone, G. P. and Smith, J. O. (2015). Digital modeling of string instrument bridge reflectance and body radiativity for sound synthesis by digital waveguides, *Proc. IEEE Workshop Appl. of Signal Process. to Audio and Acoust.*, New Paltz, NY, USA, pp. 1–5.

Maestre, E., Scavone, G. P. and Smith, J. O. (2016). Design of recursive digital filters in parallel form by linearly constrained pole optimization, *IEEE Signal Process. Lett.* **23**(11): 1547–1550.

Maestre, E., Scavone, G. P. and Smith, J. O. (2017). Joint modeling of bridge admittance and body radiativity for efficient synthesis of string instrument sound by digital waveguides, *IEEE Trans. Audio, Speech, and Lang. Process.* **25**(5): 1128–1139.

Maestre, E., Scavone, G. P. and Smith, J. O. (2021). State-space modeling of sound source directivity: An experimental study of the violin and the clarinet, *J. Acoust. Soc. Am.* **149**(4): 2768–2781.

Marshall, K. D. (1984). Modal analysis of a violin, *J. Acoust. Soc. Am.* **77**(2): 695–709.

Mertins, A., Maass, M. and Katzberg, F. (2021). Room impulse response reshaping and crosstalk cancellation using convex optimization, *IEEE Trans. Audio, Speech, and Lang. Process.* **29**: 489–502.

Miyoshi, M. and Kaneda, Y. (1988). Inverse filtering of room acoustics, *IEEE Trans. Acoust. Speech Signal Process.* **36**(2): 145–152.

Mourjopoulos, J. N. and Hatziantoniou, P. D. (2004). Real-time room equalization based on complex smoothing: Robustness results, *Proc. 116*th *AES Conv., Preprint No. 6070*.

Mourjopoulos, J. N., Clarkson, P. M. and Hammond, J. K. (1982). A comparative study of least-squares and homomorphic techniques for the inversion of mixed phase signals, *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, pp. 1858–1861.

Neal, M. T. and Vigeant, M. C. (2020). A compact spherical loudspeaker array for efficiently recreating instrument directivities, *J. Audio Eng. Soc.* **68**(11): 796–809.

Olive, S. E. (2004a). A multiple regression model for predicting loudspeaker preference using objective measurements: Part I - Listening test results, *Proc. 117th AES Conv., Preprint No. 6113*, San Francisco, USA.

Olive, S. E. (2004b). A multiple regression model for predicting loudspeaker preference using objective measurements: Part II - Development of the model, *Proc. 117th AES Conv., Preprint No. 6190*, San Francisco, USA.

Oliveira e Silva, T. (1995). Laguerre filters – An introduction, *Revista do Detua* **1**(3): 237–248.

Oppenheim, A., Johnson, D. and Steighlitz, K. (1971). Computation of spectra with unequal resolution using the fast fourier transform, *Proc. IEEE* **59**: 299–301.

Oppenheim, A. V. and Schafer, R. W. (1975). *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Oppenheim, A. V., Schafer, R. W. and Bruck, J. R. (1999). *Discrete-Time Signal Processing*, Prentice-Hall, Upper Saddle River, New Jersey, USA.

Paatero, T. and Karjalainen, M. (2003). Kautz filters and generalized frequency resolution: Theory and audio applications, *J. Audio Eng. Soc.* **51**(1–2): 27–44.

Paatero, T. and Karjalainen, M. (2006). Equalization of audio systems using Kautz filters with log-like frequency resolution, *Proc. 120th AES Conv., Preprint No. 6767*, Paris, France.

Panzer, J. and Ferekidis, L. (2004). The use of continuous phase for interpolation, smoothing and forming mean values of complex frequency response curves, *Proc. 116th AES Conv., Preprint No. 6005*, Berlin, Germany.

Parks, T. W. and Burrus, C. S. (1987). *Digital Filter Design*, John Wiley and Sons, USA.

Pedersen, J. A. and Thomsen, K. (2007). Fully automatic loudspeaker–room adaptation – The RoomPerfect system, *Proc. AES 32nd Int. Conf. "DSP for Loudspeakers"*, Hillerød, Denmark, pp. 11–20.

Pintelon, R., Guillaume, P., Rolain, Y., Schoukens, J. and Hamme, H. V. (1994). Parametric identification of transfer functions in the frequency domain-a survey, *IEEE Trans. Automatic Control* **39**(11): 2245–2260.

Powel, S. R. and Chau, P. M. (1991). A technique for realizing linear phase IIR filters, *IEEE Trans. Signal Process.* **39**(11): 2425–2435.

Price, M., Holden, S. and Sandler, M. (1996). Accurate parallel form filter synthesis, *IEE Electronics Letters* **32**(22): 2066–2067.

Rabiner, L. R. and Gold, B. (1975). *Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Rader, C. M. and Gold, B. (1967). Effects of parameter quantization on the poles of a digital filter, *Proc. IEEE* **55**(5): 688–689.

Rafaely, B. (2009). Spherical loudspeaker array for local active control of sound, *J. Acoust. Soc. Am.* **125**(5): 3006–3017.

Ramos, G. and López, J. J. (2006). Filter design method for loudspeaker equalization based on IIR parametric filters, *J. Audio Eng. Soc.* **54**(12): 1162–1178.

Ramos, G., López, J. J. and Pueo, B. (2009). Cascaded warped-fir and fir filter structure for loudspeaker equalization with low computational cost requirements, *Digit. Signal Process.* **19**: 393–409.

Salama, L. and Cousseau, J. E. (1998). Comparison of orthonormal adaptive FIR and IIR filter realizations, *Proc. IEEE Int. Symp. on Advances in Digital Filtering and Signal Processing*, pp. 77–81.

Santillian, A. O. (2001). Spatially extended sound equalization in rectangular rooms, *J. Acoust. Soc. Am.* **110**(4): 1989–1997.

Santillian, A. O., Pedersen, C. S. and Lydolf, M. (2007). Experimental implementation of a low-frequency global sound equalization method based on free field propagation, *J. Appl. Acoust.* **68**: 1063–1085.

Sarris, J. C., Stefanakis, N. J. and Cambourakis, G. E. (2004). Signal processing techniques for robust multichannel sound equalization, *Proc. 116*$^{\text{th}}$ *AES Conv., Preprint No. 6087*, Berlin, Germany.

Schlecht, S. J. (2020). FDNTB: The deedback delay network toolbox, *Proc. Conf. on Digital Audio Effects*, Vienna, Austria, pp. 211–218.

Schlecht, S. J. and Habets, E. A. P. (2019). Modal decomposition of feedback delay networks, *IEEE Trans. Signal Process.* **67**(20): 5340–5351.

Schnell, L. (ed.) (1998). *Jelek és Rendszerek Méréstechnikája I. (in Hungarian)*, M?egyetemi Kiadó, Budapest, Hungary.

Smith III, J. O. (2019). Digital state-variable filters. URL: `https://ccrma.stanford.edu/~jos/svf/`, Accessed: 16. Feb. 2019.

Smith, J. O. (1983). *Techniques for Digital Filter Design and System Identification with Application to the Violin*, PhD thesis, Stanford University, California, USA.

Smith, J. O. (2007). *Introduction to Digital Filters with Audio Applications*, Center for Computer Research in Music and Acoustics, Stanford University, W3K Publishing, USA. URL: `http://ccrma.stanford.edu/~jos/filters/`.

Smith, J. O. (2010). *Physical Audio Signal Processing for Virtual Musical Instruments and Audio Effects*, Center for Computer Research in Music and Acoustics, Stanford University, W3K Publishing, USA. On-line book, URL: `http://ccrma.stanford.edu/~jos/pasp`, Accessed: March 2010.

Smith, J. O. and Abel, J. S. (1999). Bark and ERB bilinear transform, *IEEE Trans. Speech Audio Process.* **7**(6): 697–708.

Steiglitz, K. (1980). A note on variable recursive filters, *IEEE Trans. Acoust. Speech Signal Process.* **1**: 111–112.

Steiglitz, K. and McBride, L. E. (1965). A technique for the indentification of linear systems, *IEEE Trans. Autom. Control* **AC-10**: 461–464.

Strube, H. W. (1980). Linear prediction on a warped frequency scale, *J. Acoust. Soc. Am.* **68**(4): 1071–1076.

Tyril, M., Pedersen, J. A. and Rubak, P. (2001). Digital filters for low-frequency equalization, *J. Audio Eng. Soc.* **29**(1–2): 36–43.

Vairetti, G., De Sena, E., Catrysse, M., Jensen, S. H., Moonen, M. and van Waterschoot, T. (2016). Multichannel identification of room acoustic systems with adaptive filters based on orthonormal basis functions, *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, Shanghai, China, pp. 16–20.

Vairetti, G., De Sena, E., Catrysse, M., Jensen, S. H., Moonen, M. and van Waterschoot, T. (2017). A scalable algorithm for physically motivated and sparse approximation of room impulse responses with orthonormal basis functions, *IEEE Trans. Audio, Speech, and Lang. Process.* **25**(7): 1547–1561.

Vairetti, G., De Sena, E., Catrysse, M., Jensen, S. H., Moonen, M. and van Waterschoot, T. (2018). An automatic design procedure for low-order IIR parametric equalizers, *J. Audio Eng. Soc.* **66**(11): 935–952.

Vairetti, G., De Sena, E., Moonen, M., van Waterschoot, T., Catrysse, M., Kaplanis, N. and Jensen, S. H. (2015). A physically motivated parametric model for compact representation of room impulse respones based on orthonormal basis functions, *Proc. 10th EuroNoise Conf.*, Maastricht, The Nethwerlands, pp. 149–154.

Välimäki, V. and Liski, J. (2017). Accurate cascade graphic equalizer, *IEEE Signal Process. Lett.* **24**(2): 176–180.

Välimäki, V. and Reiss, J. D. (2016). All about audio equalization: Solutions and frontiers, *Appl. Sci.* **6**(5): 129.

Vargas, R. A. and Burrus, C. S. (2001). On the design of L_p IIR filters with arbitrary frequency response, *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, Vol. 6, Salt Lake City, Utah, USA, pp. 3829–3832.

Vindrola, L., Melon, M., Chamard, J.-C. and Gazengel, B. (2020). Pressure matching with forced filters for personal sound zones application, *J. Audio Eng. Soc.* **68**(11): 832–842.

Vindrola, L., Melon, M., Chamard, J.-C., Gazengel, B. and Plantier, G. (2019). Personal sound zones: A comparison between frequency and time domain formulations in a transportation context, *Proc. 147*$^{\text{th}}$ *AES Conv., Preprint No. 10216*, New York, USA.

Välimäki, V., Pakarinen, J., Erkut, C. and Karjalainen, M. (2006). Discrete-time modelling of musical instruments, *Reports on Progress in Physics* **69**(1): 1–78.

Wang, P., Ser, W. and Zhang, M. (2000a). A dual-band equalizer for loudpseakers, *J. Audio Eng. Soc.* **48**(10): 917–921.

Wang, P., Wee, S. and Ming, Z. (2000b). Multiband warped filter equalizer design for loudspeaker systems, *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, Vol. 2, Istanbul, Turkey, pp. II913–II916.

Warusfel, O., Derogis, P. and Caussé, R. (1997). Radiation synthesis with digitally controlled loudspeakers, *Proc. 103rd AES Conv., Preprint No. 4577*, Paris, France.

Waters, M. and Sandler, M. B. (1993). Least squares IIR filter design on a logarithmic frequency scale, *Proc. IEEE Int. Symp. on Circuits and Syst.*, pp. 635–638.

Woodhouse, J. (2004). Plucked guitar transients: Comparison of measurements and synthesis, *Acta Acust. – Acust.* **90**(5): 945–965.

Zambon, S. (2013). Distributed piano soundboard modeling with common-pole parallel filters, *Proc. Stockholm Music Acoust. Conf.*, Stockholm, Sweeden, pp. 641–647.

Zölzer, U. (1994). Roundoff error analysis of digital filters, *J. Audio Eng. Soc.* **42**(4): 232–244.

Zotter, F. (2009). *Analysis and Synthesis of Sound-Radiation with Spherical Loudspeaker Arrays*, PhD thesis, University of Music and Performing Arts, Graz, Austria.

Zotter, F. and Höldrich, R. (2007). Modeling radiation synthesis with spherical loudspeaker arrays, *Int. Congress on Acoustics, September*, Madrid.

Zotter, F., Pomberger, H. and Schmeder, A. (2008). Efficient directivity pattern control for spherical loudspeaker arrays, *Proceedings of the 2nd ASA-EAA joint conference, ACOUSTICS08*, Paris.

Zotter, F., Zaunschirm, M., Frank, M. and Kronlachner, M. (2017). A beamformer to play with wall reflections: The icosahedral loudspeaker, *Computer Music J.* **41**(13): 50–68.

Zwicker, E. and Fastl, H. (1990). *Psychoacoustics: Facts and Models*, Springer-Verlag, Heidelberg, Germany.

Appendix

A.1 Error norms

In filter design, when computing the distance of the target and the filter response, usually the L_p norm is used which is defined as

$$||x(n)||_p = \left(\sum_{n=0}^{N-1} |x(n)|^p \right)^{\frac{1}{p}}. \tag{A.1}$$

Except for $p = \infty$, minimizing $||x||_p$ is equivalent to minimizing its pth power, which means that the outer function $()^{1/p}$ can be dropped:

$$||x(n)||_p^p = \sum_{n=0}^{N-1} |x(n)|^p, \tag{A.2}$$

leading to a simpler problem [Vargas and Burrus 2001]. Typical choises of p include $p = 1$ for minimizing the mean absolute error, $p = \infty$ for minimax (or Chebyshev) minimization where the maximum of the error is minimized, and $p = 2$ for least squares (LS) filter design. From these the L_2 norm has a great advantage that it leads to a particularly simple optimization problem for linear-in-parameter models (see Sec. 2.3).

For designing linear-phase FIR filters both the L_2 (least squares) and L_∞ norms (Parks–McClellan or Remez algorithm) are used [Oppenheim et al. 1999; Parks and Burrus 1987]. For FIR filters with an arbitrary phase specification (complex target response), the L_∞ norm leads to a nonlinear optimization problem due to minimizing the maximal values of absolute vales, while the L_2 norm still results in a simple closed-form solution [Parks and Burrus 1987]. Note that the combined use of L_2 (least squares) and L_∞ norms is also possible in the form of peak-constrained least-squares (PCLS) design. Such a problem can be either solved by the simple, but slow iteratively weighted LS algorithm, or by the more complex, but faster generalized exchange algorithm proposed in [Adams and Sullivan 1998].

For IIR filter design, as far as classic lowpass, highpass, bandpass, bandreject filters are concerned, the most common method is to convert an analog prototype to the digital domain. Chebyshev I, Chebyshev II, and elliptic filters all minimize the maximum error in their pass-band, stopband, or both, thus, they are optimal in the L_∞ sense [Parks and Burrus 1987; Oppenheim et al. 1999]. However, for designing IIR filters with arbitrary specifications, the minimization of the L_∞ becomes much more complicated and therefore the mathematically better tractable L_2 norm is used by most algorithms. In general, L_p norms with $p \neq 2$ would require gradient descent methods, but they can also be implemented by L_2 minimization when the weights are iteratively updated in a weighted least squares design (WLS). Examples of such

IIR filter design methods include L_p minimization for arbitrary p [Vargas and Burrus 2001] and minimizing the log-magnitude error in least-squares or minimax sense [Kobayashi and Imai 1990].

For designing low-pass, high-pass, etc. filters, generally the L_∞ norm is the preferred choice since it guarantees an upper bound on the error both in the passband and in the stopband [Parks and Burrus 1987]. However, as far as audio applications are concerned, this preference cannot be justified since an infinitely narrow frequency region where the approximation has a large error leads to the same large error in the minimax (L_∞) sense, but its effect is most probably inaudible. This is because what we hear is connected to the logarithmically-smoothed transfer function (see the discussion in Chap. 1 and 3), and such a smoothing eliminates the narrow peaks by averaging. For a peak or valley to be audible, it should be both wide and large enough, so the audibility is better related to the area of the irregularity rather than its amplitude. This means that norms with $p < \infty$ are better choices for audio.

As for the literature, there is not too much discussion on what kind of norm would be ideal for audio filter design. There are some arguments both for minimizing the L_2 mean-squared [Vairetti et al. 2018] and mean log-magnitude errors [Ramos and López 2006], but no systematic studies or psychoacoustic tests exist on the subject. The only conclusion that can be taken by analyzing the related literature is that L_2 minimization is by far the most common choice in the field of audio filter design. This was probably also motivated by the simplicity and efficiency of least-squares method in the early days of filter design, but it is reasonable to expect that if the L_2 norm was unsuitable for audio, it would not be so widespread at present day. (On the contrary, the unsuitability of minimizing the error in the linear scale for audio led to the prevalent use of the error computed on the logarithmic frequency scale).

In line with this common practice, throughout this work least squares (L_2) optimization will be used, but it must be noted that the proposed algorithms could be straightforwardly modified for applying other norms. This could be most simply done by using WLS techniques along the lines of [Vargas and Burrus 2001; Kobayashi and Imai 1990]. However, this extension is out of the scope of the present work.

Taking a broader view, there are some auditory models that try to asses the audibility of differences between transfer functions: in [Olive 2004b] a multiple regression model about the preference rating of loudspeakers is created based on listening tests, while in [Lavandier et al. 2008] the perceptual similarities of loudspeakers were related to acoustic measurements. In theory such auditory models could be used in the (definitely nonlinear) optimization process for finding the coefficients of a filter or equalizer: this can be an interesting topic for future reseach.

A.2 Equivalent LS approximation using Kautz and parallel filters

It has been shown in 6.4 that the Kautz basis functions are the linear combinations of the basis functions of the parallel filter and that the parallel filter parameters $\mathbf{p}$ are obtained from the Kautz weights $\mathbf{w}$ by a multiplication with the triangular matrix $\mathbf{K}$. While already from this it should be clear that the two methods will result in the same filter response, it is still instructive to develop the equations for the least-squares case used for parallel filter design.

According to Eqs. (6.4) and (6.6), the impulse response of the fixed-pole parallel filter is

$$\mathbf{h} = \mathbf{M}\mathbf{p} = \mathbf{M}\mathbf{M}^{+}\mathbf{h}_\mathrm{t} = \mathbf{M}(\mathbf{M}^H\mathbf{M})^{-1}\mathbf{M}^H\mathbf{h}_\mathrm{t}, \tag{A.3}$$

where $\mathbf{h}_\mathrm{t}$ is the target impulse response, $\mathbf{M}$ is the modeling matrix containing the basis functions of the parallel filter in its columns, and $\mathbf{p}$ contains the optimal set of parallel filter parameters.

We can write the same for Kautz filters

$$\tilde{\mathbf{h}} = \mathbf{G}\mathbf{w} = \mathbf{G}\mathbf{G}^{+}\mathbf{h}_\mathrm{t} = \mathbf{G}(\mathbf{G}^H\mathbf{G})^{-1}\mathbf{G}^H\mathbf{h}_\mathrm{t}, \tag{A.4}$$

where $\tilde{\mathbf{h}}$ is the impulse response of the Kautz filter, $\mathbf{G}$ is the modeling matrix containing the basis functions of the Kautz filter in its columns, and $\mathbf{w}$ contains the optimal set of Kautz parameters. The Kautz basis functions are the linear combinations of the basis functions of the parallel filter:

$$\mathbf{G} = \mathbf{M}\mathbf{K}. \tag{A.5}$$

Substituting Eq. (A.5) into Eq. (A.4) gives

$$\begin{aligned}\tilde{\mathbf{h}} = \mathbf{M}\mathbf{K}(\mathbf{K}^H\mathbf{M}^H\mathbf{M}\mathbf{K})^{-1}\mathbf{K}^H\mathbf{M}^H\mathbf{h}_\mathrm{t} = \\ = \mathbf{M}\mathbf{K}\mathbf{K}^{-1}(\mathbf{M}^H\mathbf{M})^{-1}(\mathbf{K}^H)^{-1}\mathbf{K}^H\mathbf{M}^H\mathbf{h}_\mathrm{t} = \\ = \mathbf{M}(\mathbf{M}^H\mathbf{M})^{-1}\mathbf{M}^H\mathbf{h}_\mathrm{t},\end{aligned} \tag{A.6}$$

which is the same as Eq. (A.3), showing that the impulse responses of the parallel and Kautz filters $\mathbf{h}$ and $\tilde{\mathbf{h}}$ are indeed the same when using the same pole set p_i and the same target response $\mathbf{h}_\mathrm{t}$.

The complete equivalence is valid when the Kautz filter is designed by fitting the linear combinations of the Kautz signals to the target response by an LS fit as in Eq. (A.4). However, by taking advantage of the orthogonality of the Kautz responses $\mathbf{G}^H = \mathbf{G}^{-1}$, the Kautz parameters are usually computed by the scalar product of Eq. (5.6), that is,

$$\mathbf{w} = \mathbf{G}^H\mathbf{h}_\mathrm{t}. \tag{A.7}$$

Since the orthonormality of the Kautz responses holds only for the entire (infinitely long) basis functions, these two forms of Kautz design (LS and scalar product) are equivalent only if $\mathbf{G}$ is long enough so that all Kautz responses have decayed to a negligible value. This usually holds since it is in general a good practice to specify a target impulse response that is at least as long as the effective length of the Kautz basis functions.

Note that when designing the Kautz filter with Eq. (A.7) with a target response $\mathbf{h}_\mathrm{t}$ that is shorter than the Kautz basis functions (say, N sample long), it is basically equivalent to zero padding the target $\mathbf{h}_\mathrm{t}$ to infinity and fitting the infinitely long Kautz responses. On the other hand, the LS design of Eq. (A.4) will only take into account the first N samples of the target, and the rest will be a "don't care" region. The same results for the parallel filter when designed with the LS method.

A.3 Pole radius for a pretedermined pole set

Using a predetermined (e.g., logarithmic) pole set is the simplest choice for parallel filter design. In this case the pole frequencies are set by the user, but the pole radii have to be determined by a suitable formula. As also suggested in [Paatero and Karjalainen 2003] for the Kautz filter, we will set them so that the transfer functions of the sections cross at their -3 dB point. Smaller overlap (higher Q factor) would mean that there are "empty" parts between the peaks of the

individual sections, and larger overlap (lower Q factor) means that the numerical conditioning of the parameter estimation gets worse since more and more sections will have a meaningful contribution for each target frequency point. I have found by experiments that the exact value is not critical, but -3 dB is a reasonable choice.

Let us consider the frequency response of a complex first-order IIR filter:

$$H(\vartheta) = \frac{1}{1 - pz^{-1}} = \frac{1}{1 - pe^{-j\vartheta}} = \frac{1}{1 - Re^{j(\vartheta_0 - \vartheta)}}, \tag{A.8}$$

where the pole position $p = Re^{j\vartheta_0}$ is defined by the radius R and angle ϑ_0. Equation (A.8) is actually the basis function of the complex form of the parallel filter (see Eq. (6.1)), and in the real form of the filter shown in Eq. (6.2) the second-order sections are composed of two such terms with complex conjugate poles. In the following we will analyze the complex basis function of Eq. (A.8) and assume that the response coming from the complex-conjugate pole will have a no effect on the frequencies of the -3 dB points of the peak of interest.

The maximum gain of Eq. (A.8) is $1/(1 - R)$ at $\vartheta = \vartheta_0$, thus, at the -3 dB point we have

$$\left| \frac{1}{1 - Re^{j(\vartheta_0 - \vartheta)}} \right| = \frac{1}{\sqrt{2}} \frac{1}{1 - R}. \tag{A.9}$$

Taking the square and reciprocal gives

$$\begin{aligned} \left(1 - Re^{j\alpha}\right)\left(1 - Re^{-j\alpha}\right) &= 2(1 - R^2) = & \text{(A.10)} \\ 1 + R^2 - R\left(e^{j\alpha} + e^{-j\alpha}\right) &= 2 + 2R^2 - 4R, & \text{(A.11)} \end{aligned}$$

where $\alpha = \vartheta_0 - \vartheta$. By using Euler's formula we obtain

$$1 + R^2 + R(2\cos(\alpha) - 4) = 0, \tag{A.12}$$

which needs to be solved for R for the given α distance of the pole frequency and the angular frequency where we wish to have -3 dB magnitude response.

The solution is obtained as

$$R = 2 - \cos(\alpha) - \sqrt{(2 - \cos(\alpha))^2 - 1}. \tag{A.13}$$

While Eq. (A.13) is not overly complicated, I have been looking for a simpler formula, even if it will be an approximation, since the filter design performance will not be affected if we slightly depart from the -3 dB value. This I have obtained using the Taylor-series approximations $\cos(x) \approx 1 - x^2$ and $\sqrt{1 + x} \approx 1 + x/2$.

With some algebraic manipulations we then obtain

$$R \approx 2 - 1 + \frac{\alpha^2}{2} - \sqrt{\left(2 - 1 + \frac{\alpha^2}{2}\right)^2 - 1} \approx 1 - \alpha + \frac{\alpha^2}{2} - \frac{\alpha^3}{8}. \tag{A.14}$$

This is very similar to the Taylor-series approximation of the exponential function:

$$e^{-\alpha} \approx 1 - \alpha + \frac{\alpha^2}{2} - \frac{\alpha^3}{6} + \frac{\alpha^4}{24} + \ldots. \tag{A.15}$$

Thus, we may use the simpler formula

$$R = e^{-\alpha} = e^{-\frac{\Delta\theta}{2}} \tag{A.16}$$

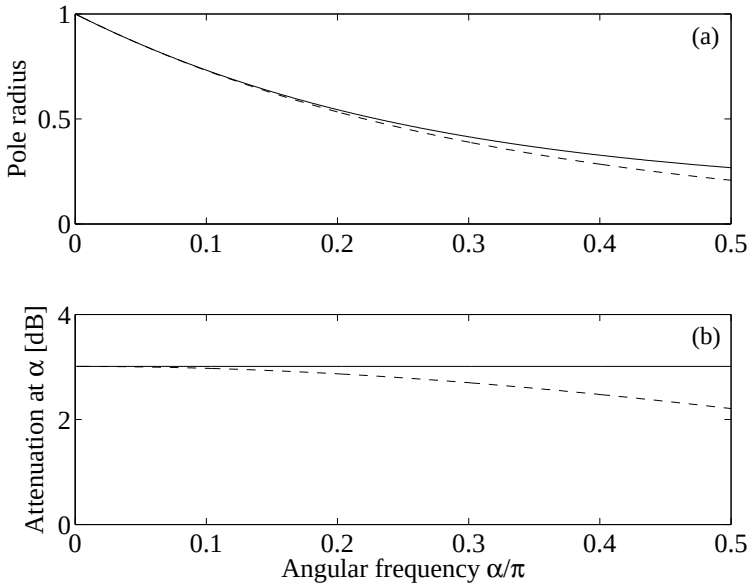


Figure A.1: Accurate (solid line) and approximate (dashed line) pole radii to obtain -3 dB at $\alpha = \vartheta - \vartheta_0$ frequency distance from the pole angle ϑ_0 (a), and the actual attenuation value at this frequency (b).

as an approximation to Eq. (A.13), where $\Delta\theta$ is the total frequency distance of the two -3 dB points used in Eq. (7.1), while α is the distance of one -3 dB point from the peak of the response.

The approximation is very accurate for $\alpha = \Delta\theta/2 < 0.1\pi$, as can be seen in Fig. A.1 (a), and reasonable up to $\alpha = \Delta\theta/2 = \pi/2$. The actual attenuation compared to the peak of the response is very close to 3 dB for $\alpha < 0.1\pi$ and thus $\Delta\theta < 0.2\pi$, while it is a still acceptable 2.2 dB for $\alpha = \pi/2$ corresponding to $\Delta\theta = \pi$, as displayed in Fig. A.1 (b). Since a pole distance $\Delta\theta < 0.2\pi$ is always fulfilled expect for very low pole densities (and $\Delta\theta < \pi$ is fulfilled in each case since that would only arise if one pole is at zero frequency and the other is at half the sample rate), Eq. (A.16), and thus Eq. (7.1) can be considered as a simple but sufficiently accurate approximation.

A.4 Additional examples of using the predetermined pole set for parallel filters

In Sec. 7.2 we have already seen that by applying a logarithmically spaced pole frequency set, a logarithmic frequency resolution filter design can be achieved and that by using different pole densities in the different regions, the frequency resolution will be also different. By the help of Fig. A.2 we will take a look at some additional examples for the same loudspeaker–room response. The first case of Fig. A.2 (a) employs pole frequencies in the low frequency-region only, and indeed it is visible that the target response is modeled in that region only. As a practical application, in an equalizer design setting this would mean equalizing the most problematic low-frequency region of the response, while leaving other parts intact. Figure A.2 (b) uses a denser set of pole frequencies in the middle-frequency range, showing again that the modeling accuracy concentrated in a limited frequency range. Probably the most interesting case is that of Fig. A.2 (c) where the pole frequencies are set arbitrarily. They were actually chosen by the

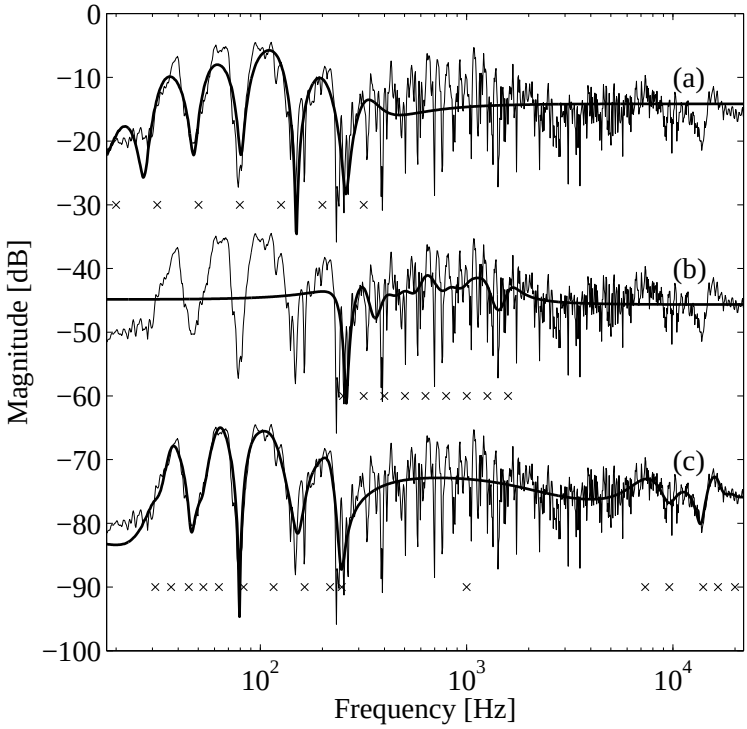


Figure A.2: Modeling a minimum-phase loudspeaker–room response by fixed-pole parallel filters having a predetermined pole set: (a) seven pole frequencies logarithmically spaced from 20 Hz to 300 Hz (filter order is 14), (b) nine pole frequencies logarithmically spaced from 250 Hz to 1500 Hz (filter order is 18). In (c) the 16 pole frequencies were selected manually by the help of the `parfiltdemo` MATLAB script (total filter order is 32). The pole frequencies are indicated by crosses. The curves are offset for clarity.

help of my interactive MATLAB script `parfiltdemo` where the poles can be positioned by the click of a mouse and the resulting filter response can be seen immediately. Thus it is possible to manually finetune the approximation according to the desire of the user. The script can be downloaded from [Parallel filter homepage 2021] and runs both in MATLAB and Octave.

To show something different from logarithmic scale, the next example will display a linear frequency scale plot where the accuracy of modelling is again controlled by the density of the pole frequencies. The target response is a synthetic case where the magnitude response is the sum of two sinusoids to obtain a target where detail is present at two levels. The phase response is calculated by the Hilbert transform from the log. magnitude [Oppenheim and Schafer 1975] so that we have a minimum-phase specification. It can be seen in Fig. A.3 (a) that with a low-density pole set only the general trend of the response (the low-frequency sinusoid) is modelled. As expected, using a denser set of poles in a certain region in Fig. A.3 (b) leads to modelling the fine detail in that region only. Figure A.3 (c) shows a combined case where the fine detail is approximated only in the angular frequency range of 0.2π to 0.4π, while neglected for the rest of the frequencies. These examples attest the complete freedom in setting the frequency resolution of the design when using fixed-pole parallel filters.

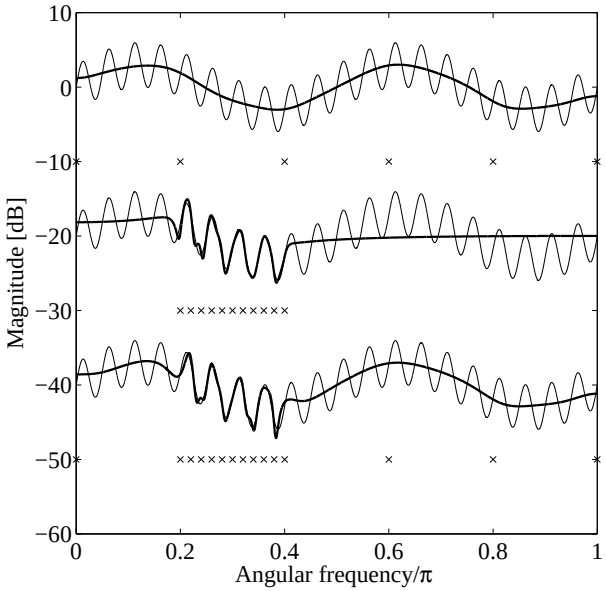


Figure A.3: Modeling a minimum-phase synthetic target response by fixed-pole parallel filters having a predetermined pole set: (a) six pole frequencies linearly spaced from 0 to π angular frequencies, (b) eleven pole frequencies linearly spaced from 0.2π to 0.4π, and (c) the combination of the above two pole sets. Note the linear frequency scale in radians as opposed to the logarithmic scale of other figures. The pole frequencies are indicated by crosses. The curves are offset for clarity.

A.5 Further comparisons of various filter design techniques

While in the audio signal processing community it is more customary to provide the comparison of various filter design approaches using real-world examples as was done in Chap. 7, here the various pole positioning techniques proposed for the parallel filter will be compared on a synthetic target response. In addition, a comparison with previous filter design techniques will also be provided. We will see that the synthetic target response will provide some additional insight on how the modeling detail of the various techniques is distributed as a function of frequency.

The target magnitude response has linear segments that jump between +4 and -4 dB at each octave, assuring that the "detail" is distributed evenly in the logarithmic scale, as can be seen in Figs. A.4 and A.5, thin line. The phase is computed by the help of the Hilbert transform so that a minimum-phase response is obtained. The two figures show the same filter design techniques (filter responses displayed by thick lines), the only difference is that the filter order is double in Fig. A.5 compared to Fig. A.4.

It can be seen in both figures that the FIR filters (a) obtained by windowing the target impulse response are able to follow only the high frequency part of the responses, and this is significantly improved by the IIR filters (b) designed using the Steiglitz-McBride method, but the mid- and low frequencies are still poorly modeled.

The warped FIR filters (c) designed by windowing the warped target response with $\lambda = 0.9$ are able to better distribute their modeling ability on the logarithmic scale, but the modeling is confined to a limited frequency range. In addition, this technique is unable to follow the sharp transitions of the target response.

Curve (d) shows a fixed-pole parallel filter design with logarithmic pole positioning. Now

the distribution of the modeling detail is even in the logarithmic scale, since all frequency regions are modeled with equal accuracy. It can also be seen that the filter response is the smoothed (or low-pass filtered) version of the target frequency response, as anticipated from Sec. 6.5. It can be seen that already this very simple approach gives significantly improved results compared to traditional FIR, IIR and warped FIR filters. While not shown, the Kautz filter with a logarithmic pole set gives exactly the same filter response as in curve (d), albeit with double amount of arithmetic operations due to its more complicated filter structure.

The magnitude response of a fixed-pole parallel filter using ripple-density based pole positioning is displayed by curve (e). For the lower filter order example of Fig. A.4 (e), the response is the same as that with a strictly logarithmic pole set, since the limited number of pole frequencies does not allow concentrating them to the edges of the response, but results in an almost logarithmic pole frequency series shown by crosses in Fig. A.4 (e). On the contrary, at double filter order displayed in Fig. A.5 (e) the ripple-density based method improves the performance compared to logarithmic pole positioning by assigning more poles to the transitions of the target response. While it is hard to see in the figure, actually there are two distinct poles close to each other at each edge.

The response of a fixed-pole parallel filter using warped IIR based pole positioning is shown by curve (f). The pole set was obtained by designing a warped IIR filter with $\lambda = 0.9$ based on the target response, finding and dewarping the poles, then using them as the poles of the parallel filter. The frequency response of the parallel filter is practically the same as that of the warped IIR filter, and if the same poles are used for the Kautz filter, the response would be again identical. Basically, this curve shows what has been possible using prior approaches. It can be seen in Fig. A.4 (f) that a filter order of 24 is not sufficient to provide appropriate modeling at low and high frequencies, and the modeling detail is concentrated in the middle range of the response, similarly to warped FIR filters. At double filter order shown in Fig. A.4 (f) the filter response becomes quite reasonable, with some loss of detail at very low and very high frequencies.

Parallel filters obtained with dual-band warping and custom warping are shown by curves (g) and (h). It is especially apparent in Fig. A.4 (g) and (h) that the approximation error is more evenly distributed in the logarithmic scale compared to the single-band warping of Fig. A.4 (f). The two methods provide a similar performance: for the lower order case of Fig. A.4, custom warping (h) seems to provide a better behaving response, while for the higher order case of Fig. A.5, dual-band warping (g) has a some benefit. In any case, they both provide better modeling accuracy compared to earlier approaches, the improvement is especially apparent for using lower order filters as in Fig. A.4. Turning this around, it can also be stated that a similar modeling accuracy on the logarithmic scale can now be achieved at reduced filter order, and the computational complexity is decreased even further by the simpler filter structure compared to warped or Kautz filters.

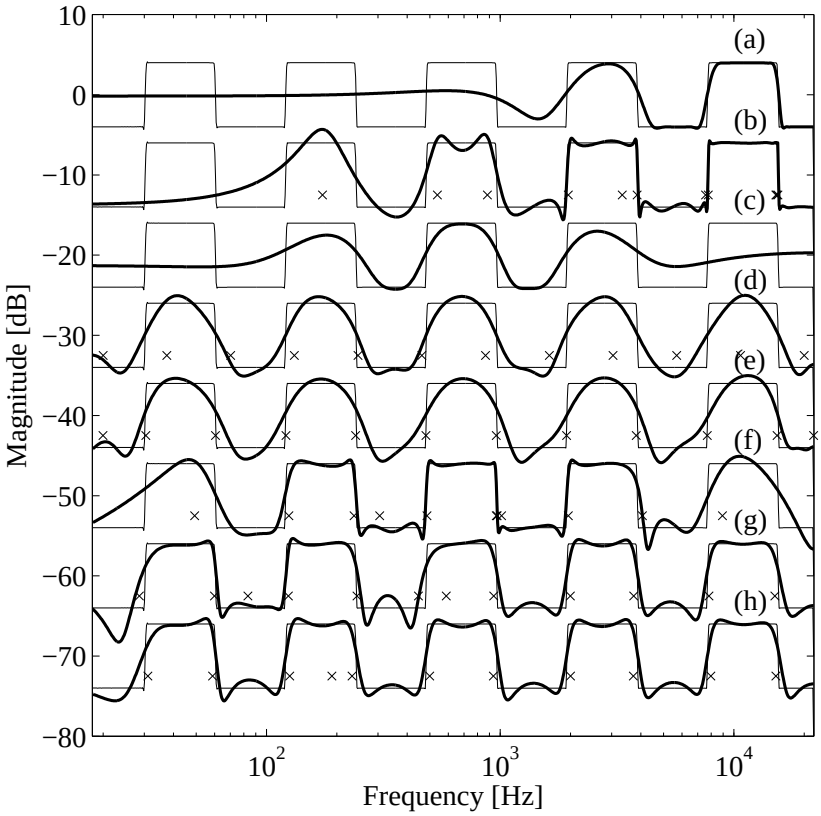


Figure A.4: Comparison of various filter design methods using a filter order of 24 for the IIR filters (b)-(h), and 48 for the FIR filter (a). The thin lines show the minimum-phase target response, while the solid lines display the filter responses. The first three curves display (a) FIR, (b) IIR, and (c) warped FIR filters. The next ones show parallel filters with (d) strictly logarithmic pole frequency series, poles obtained by the (e) ripple-density method, (f) single-band warping, (g) dual-band warping, and (h) custom-warping. The pole frequencies are marked by crosses, and the curves are offset for clarity.

A.6 Least-squares design for the non-delayed parallel filter

According to Sec. 6.2.1, finding the parameters of the fixed-pole parallel filter for a time-domain target response involves finding the parameter vector $\mathbf{p}$ such that the resulting impulse response

$$\mathbf{h} = \mathbf{M}\mathbf{p} \tag{A.17}$$

is the closest to the target impulse response $\mathbf{h}_\mathrm{t}$ in the mean squared sense.

Let us first rewrite the modeling matrix $\mathbf{M}$ in such a way that its first part contains the modeling signals for the IIR part $\mathbf{P}$ (N by P matrix where N is the number of samples, and P is the number of IIR weights) and the second part contains the modeling signals for the FIR part $\mathbf{F}$ (N by $M+1$ matrix where $M+1$ is the length of the FIR path). This gives

$$\mathbf{M} = \begin{bmatrix} \mathbf{P} & \mathbf{F} \end{bmatrix}. \tag{A.18}$$

The FIR part $\mathbf{F}$ contains the delayed unit pulses $\delta(n-m)$ for $m = [0, \ldots, M]$, thus, it can be partitioned to an $M+1$ by $M+1$ identity matrix $\mathbf{E}$ and an $N-M-1$ by $M+1$ zero matrix $\mathbf{0}$ as

$$\mathbf{F} = \begin{bmatrix} \mathbf{E} \\ \mathbf{0} \end{bmatrix}. \tag{A.19}$$

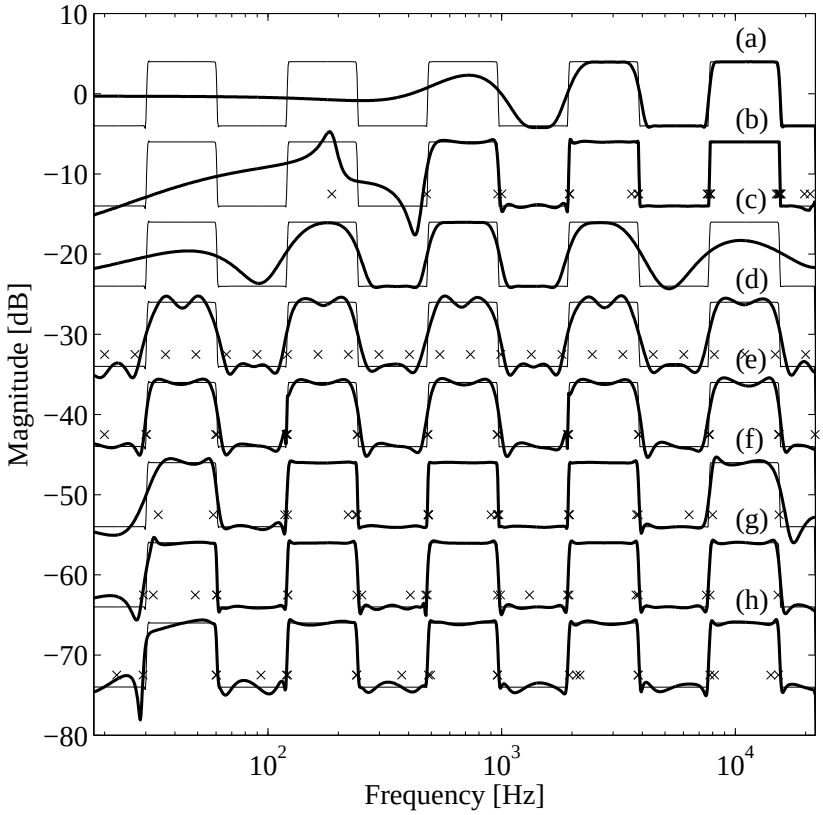


Figure A.5: Comparison of various filter design methods using a filter order of 48 for the IIR filters (b)-(h), and 96 for the FIR filter (a). The thin lines show the minimum-phase target response, while the solid lines display the filter responses. The first three curves display (a) FIR, (b) IIR, and (c) warped FIR filters. The following ones show parallel filters with (d) strictly logarithmic pole frequency series, poles obtained by the (e) ripple-density method, (f) single-band warping, (g) dual-band warping, and (h) custom-warping. The pole frequencies are marked by crosses, and the curves are offset for clarity.

We may make a similar partitioning to the IIR part $\mathbf{P}$ where the first $M+1$ samples of the modeling signals are contained in $\mathbf{P}_1$ and the rest are in $\mathbf{P}_2$, giving

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix}. \tag{A.20}$$

Thus, the entire modeling matrix $\mathbf{M}$ is partitioned to four parts:

$$\mathbf{M} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{E} \\ \mathbf{P}_2 & \mathbf{0} \end{bmatrix}. \tag{A.21}$$

We also partition the parameter vector $\mathbf{p}$ to the parameter vector of IIR part (numerator coefficients) $\mathbf{b}$ and of the FIR part $\mathbf{f}$ as

$$\mathbf{p} = \begin{bmatrix} \mathbf{b} \\ \mathbf{f} \end{bmatrix} \tag{A.22}$$

and the resulting impulse response $\mathbf{h}$ to the first $M+1$ samples as $\mathbf{h}_1$ and the following $N-M-1$ samples as $\mathbf{h}_2$ as

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix}. \tag{A.23}$$

Finding the optimal parameter vector $\mathbf{p}$ in Eq. (A.17) in the least squares sense involves computing the products $\mathbf{M}^T\mathbf{M}$ and $\mathbf{M}^T\mathbf{h}_\mathrm{t}$, since we actually need to solve

$$\mathbf{M}^T\mathbf{M}\mathbf{p} = \mathbf{M}^T\mathbf{h}_\mathrm{t} \tag{A.24}$$

for $\mathbf{p}$. The matrices are:

$$\mathbf{M}^T\mathbf{M} = \begin{bmatrix} \mathbf{P}_1^T & \mathbf{P}_2^T \\ \mathbf{E} & \mathbf{0}^T \end{bmatrix} \times \begin{bmatrix} \mathbf{P}_1 & \mathbf{E} \\ \mathbf{P}_2 & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1^T\mathbf{P}_1 + \mathbf{P}_2^T\mathbf{P}_2 & \mathbf{P}_1^T \\ \mathbf{P}_1 & \mathbf{E} \end{bmatrix}, \tag{A.25}$$

and

$$\mathbf{M}^T\mathbf{h}_\mathrm{t} = \begin{bmatrix} \mathbf{P}_1^T & \mathbf{P}_2^T \\ \mathbf{E} & \mathbf{0} \end{bmatrix} \times \begin{bmatrix} \mathbf{h}_{\mathrm{t},1} \\ \mathbf{h}_{\mathrm{t},2} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1^T\mathbf{h}_{\mathrm{t},1} + \mathbf{P}_2^T\mathbf{h}_{\mathrm{t},2} \\ \mathbf{h}_{\mathrm{t},1} \end{bmatrix}. \tag{A.26}$$

Substituting these into Eq. (A.24) gives

$$\begin{bmatrix} \mathbf{P}_1^T\mathbf{P}_1 + \mathbf{P}_2^T\mathbf{P}_2 & \mathbf{P}_1^T \\ \mathbf{P}_1 & \mathbf{E} \end{bmatrix} \times \begin{bmatrix} \mathbf{b} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1^T\mathbf{h}_{\mathrm{t},1} + \mathbf{P}_2^T\mathbf{h}_{\mathrm{t},2} \\ \mathbf{h}_{\mathrm{t},1} \end{bmatrix}. \tag{A.27}$$

The second "row" of Eq. (A.27) is

$$\mathbf{P}_1\mathbf{b} + \mathbf{f} = \mathbf{h}_{\mathrm{t},1}, \tag{A.28}$$

from which the FIR coefficients $\mathbf{b}$ are obtained as

$$\mathbf{f} = \mathbf{h}_{\mathrm{t},1} - \mathbf{P}_1\mathbf{b}, \tag{A.29}$$

where $\mathbf{P}_1\mathbf{b}$ actually contains the first $M+1$ samples of the IIR response. The first "row" of Eq. (A.27) is

$$\mathbf{P}_1^T\mathbf{P}_1\mathbf{b} + \mathbf{P}_2^T\mathbf{P}_2\mathbf{b} + \mathbf{P}_1^T\mathbf{f} = \mathbf{P}_1^T\mathbf{h}_{\mathrm{t},1} + \mathbf{P}_2^T\mathbf{h}_{\mathrm{t},2}. \tag{A.30}$$

By substituting Eq. (A.29) into Eq. (A.30), many terms cancel out and we obtain

$$\mathbf{P}_2^T\mathbf{P}_2\mathbf{b} = \mathbf{P}_2^T\mathbf{h}_{\mathrm{t},2}, \tag{A.31}$$

which is solved as

$$\mathbf{b} = (\mathbf{P}_2^T\mathbf{P}_2)^{-1}\mathbf{P}_2^T\mathbf{h}_{\mathrm{t},2}. \tag{A.32}$$

Equation (A.32) actually shows that the weights of the parallel filter contained in vector $\mathbf{d}$ depend only on that part of the target impulse response vector $\mathbf{h}_\mathrm{t}$ which does not overlap with the FIR part (this is $\mathbf{h}_{\mathrm{t},2}$), as anticipated in Sec. 9.1.1.

The FIR part is beneficial for modeling non-minimumphase responses (see Sec. 9.1 for a practical example), and it is used for modeling the response before the main peak. That part of the target response is contained in $\mathbf{h}_{\mathrm{t},1}$. The decaying part after the main peak is thus in $\mathbf{h}_{\mathrm{t},2}$. Looking at Eq. (A.32), if there is a long FIR part (M is large), the basis functions of the IIR part have already decayed to a small value at sample $M+1$, thus the samples in $\mathbf{P}_2$ will be small, but $\mathbf{h}_{\mathrm{t},2}$ will be still significant since it is just after the main peak of the target impulse response. This leads to large coefficients in $\mathbf{b}$. On the contrary, $\mathbf{P}_1$ contains the first part of the IIR basis functions that have not yet decayed to a small value, thus, the FIR coefficients $\mathbf{f} = \mathbf{h}_{\mathrm{t},1} - \mathbf{P}_1\mathbf{b}$ also need to be large in magnitude, typically much larger than the target response $\mathbf{h}_{\mathrm{t},1}$ itself because of the large magnitude of $\mathbf{P}_1\mathbf{b}$. This explains why the IIR and FIR parts look like a mirror image to each other in Fig. 9.2 (b) and (c), both much larger than the resulting impulse response (a). This requires the downscaling of the input signal (or the numerator coefficients) to avoid overflow in the FIR and IIR branches, reducing the usable dynamic range significantly.