Digital Audio Filter Design

Using Frequency Transformations

by

Chalee Asavathiratham

Submitted to the Department of Electrical Engineering and

Computer Science

in Partial Fulfillment of the Requirements for the Degrees of

Bachelor of Science in Electrical Engineering and Computer Science

and Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 23, 1996

Author.....
Department of Electrical Engineering and Computer Science
May 23, 1996

Certified by.....
Alan V. Oppenheim
Distinguished Professor of Electrical Engineering
Thesis Supervisor

Accepted by.....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Digital Audio Filter Design

Using Frequency Transformations

by

Chalee Asavathiratham

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 1996, in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Electrical Engineering and Computer Science
and
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis involves designing discrete-time filters for modifying the spectrum of audio signals. The main contribution of this research is the significant reduction in the order of the filters. To achieve this, we have combined a technique called frequency transformation, or frequency warping, with an effective Finite Impulse Response (FIR) filter design algorithm. Both techniques exploit some properties of audio filters which allow us to relax the design specifications according to human auditory perception.

We show several properties of frequency transformations and explain their importance in designing audio filters. We incorporate this technique into design procedures and test them with sample filters. We study the signal flowgraphs of warped filters and evaluate their computational requirements and performance in the presence of quantization noise.

Thesis supervisor: Alan V. Oppenheim
Title: Distinguished Professor of Electrical Engineering

Acknowledgments

I am very grateful to Dr. Paul Beckmann of Bose Corporation, whose guidance, and encouragement have been a tremendous help for me to finish this thesis. It was simply a privilege for me to work with him. I am also very thankful to Professor Alan Oppenheim for giving me the opportunity to work on this problem, for teaching me everything about signal processing, and for taking me into the Digital Signal Processing Group, where I get to meet with wonderful new friends. My gratitude also extends to the Bose Corporation whose generosity has supported this research for the past one and a half years.

Finally, I would like to thank my family and friends for the all their love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis deals with the design of discrete-time filters for modifying the spectrum of audio signals. The main contribution of this research is the significant reduction in the memory and computational requirements of the filters. To achieve this, we have combined a technique called frequency transformation, or frequency warping, with an effective Finite Impulse Response (FIR) filter design algorithm.

Frequency transformation is a technique which allows us to produce a filter whose frequency response, $G(e^{j\omega})$, is equal to that of another filter, $H(e^{j\omega})$, except that the frequency axis is rescaled. The ability to design a filter on a rescaled frequency axis suits the problem of audio equalization very properly. Since human ears have better frequency resolution at lower frequency than at higher frequency, filter design algorithms must pay attention to the fine details in the low frequencies. As the low-frequency details becomes finer, the required filter length becomes longer. By exploiting the relationship between the frequency response $H(e^{j\omega})$ and the frequency-transformed response $G(e^{j\omega})$, we can design a short-length FIR filter $h[n]$ and choose a transformation such that $G(e^{j\omega})$ has not only the desired audio equalization, but also a low order like that of $h[n]$. Note that due to the frequency transformation, the filter $G(e^{j\omega})$ will be IIR (Infinite Impulse Response) even though

$h[n]$ is FIR. Thus, we are able to draw on the wealth of FIR filter design algorithms and use them to design IIR filters.

Further reduction in filter length is achieved by designing FIR filters based on a model of human audio perception. Specifically, the deviation in the response of a filter from the desired one is weighted against a perceptual bound. This bound represents the threshold at which a typical listener can detect an error in the frequency response. This model allows us to relax the FIR filter design constraint, from designing a filter that must precisely fit a desired response to designing a filter that must lie within upper and lower magnitude bounds.

The two FIR filter design algorithms which we used were the well-known Parks-McClellan algorithm and the lesser-known CONRIP algorithm. Traditionally, the Parks-McClellan algorithm has been widely used to design lowpass and bandpass filters. Here we will use it to fit an arbitrary curve with a weighting function derived from our perceptual model. Though largely overlooked, the CONRIP algorithm suits our application very appropriately and returns the shortest possible FIR filter that fits our error model. However, experimental results in Chapter 5 showed that these two algorithms yield the same minimal order almost all the time. Both of these design algorithms required high numeric precision and failed to converge when standard double-precision arithmetic was used. To overcome this problem, we wrote a library of arbitrary precision mathematical functions and used them during filter design.

Chapter 2 presents an overview of audio equalizer design including frequency magnitude specifications and error bands based on psychoacoustic measures. This section also discusses the problems encountered when standard FIR and IIR filter design methods are used to design audio equalizers. Chapter 3 reviews the idea of allpass transformations and proves several important properties that they possess. It also summarizes the design steps involved in applying frequency transformations. Chapter 4 discusses the Parks-McClellan and CONRIP design algorithms with emphasis on CONRIP. Chapter 5 presents results from filter design experiments and Chapter 6 investigates the noise performance when implementing (not designing) frequency warped filter using finite precision arithmetic. Chapter 7 summarizes the main results of the thesis and suggests directions for future works.

Chapter 2

Audio Filter Specification and Design

Digital filters can be used in a wide variety of ways to modify audio signals. These modifications can be grouped into three main categories: spectral, temporal, and spatial. Spectral changes are perceived as changes in the frequency response of the audio signal. Temporal changes are perceived in the time domain and examples include echoes and reverberation. Spatial changes modify the perceived location of a sound. All of these audio signal modifications can be achieved through digital filters – digital linear time-invariant systems – and thus there may not always be a clear distinction between them. For example, a digital filter designed for spectral modifications may also cause some perceptible time-domain distortion.

In this thesis, we focus on designing digital filters for spectrally modifying an audio signal. This is probably the most common use of an audio filter and there are many applications. The simplest application is the implementation of tone controls (bass, mid-range, treble) commonly found on audio equipment. Another application might be to compensate for errors in the frequency response of a loudspeaker which result from shortcomings in the transducer. A further application is to compensate for the

constrained speaker position in an automobile or for the damping of the response due to the car interior.

In many cases, we want to be able to change the frequency response in the field, rather than in the factory. In order to accomplish this, the filter design algorithm must be able to operate without supervision. That is, we need a algorithm that is robust enough to return the optimal filter without human intervention once inputs are specified.

In this chapter, we discuss specific features of audio equalizers which affect their design and implementation. It will be shown that most traditional IIR design techniques and FIR implementations are not suitable for use in audio equalizers.

2.1 Logarithmic Frequency and Magnitude Specifications

Experiments have shown that the human auditory system has better resolution at low frequencies than at high frequencies [1]. For example, it is quite easy to distinguish between 100 and 110 Hz, but extremely difficult to distinguish between 10000 and 10010 Hz. Due to this property, audio equalizers require much higher resolution at low frequencies than at high frequencies. A good model which approximates this behavior is to assume that the frequency specifications are uniform when viewed on a logarithmic frequency scale.

Frequency specifications on a log scale are given in fixed multiplicative increments. One standard unit of such increments is an octave. A k-octave specification is the

one whose frequency response is given on the set of frequencies $f_0, 2^k f_0, 2^{2k} f_0, 2^{3k} f_0$, and so on. For example, a one-third octave specification shall contain the frequency response at the following frequencies: $f_0, 2^{1/3} f_0, 2^{2/3} f_0, 2f_0$, etc.

The auditory system can also perceive signals over an enormous dynamic range; from a pin drop to a jet plane. The most appropriate manner in which to represent this range of loudness is in decibels (dB) which is defined as 20 $\log_{10}(x)$. dB is also the standard scale to use when discussing changes in magnitude such as in a frequency response.

Loudness is also perceived on a dB scale. For example, increasing a signal level from 10 to 20 dB roughly doubles its perceived loudness. Doubling again from 20 to 40 dB doubles it again.

2.2 Insignificance of Phase Response

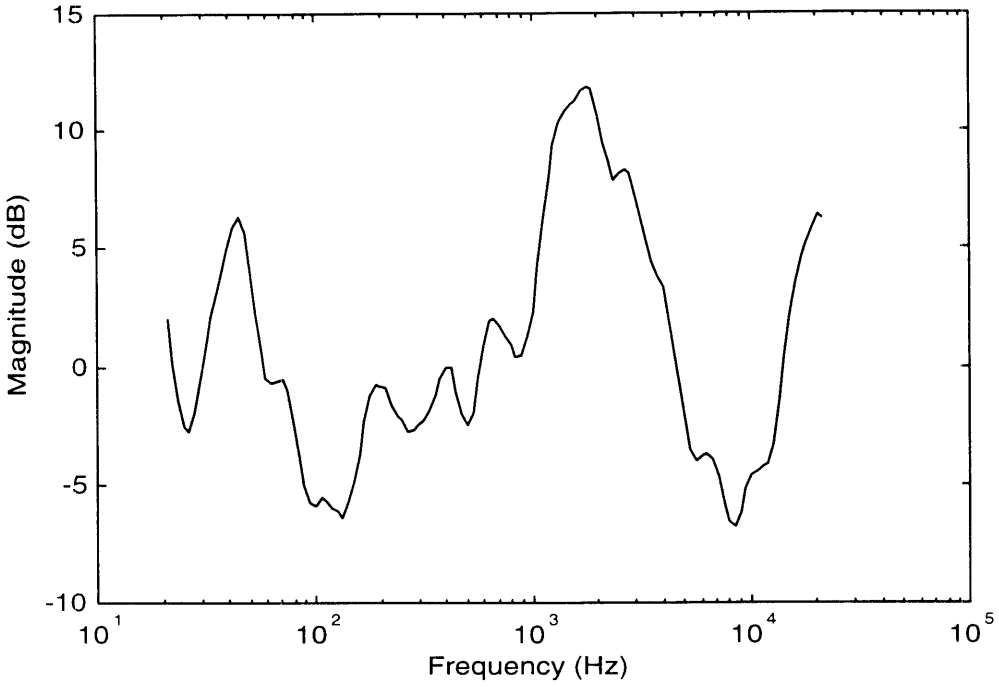
The specifications of an audio filter are usually given in terms of its magnitude response. For the most part, the human ear is insensitive to small variations in phase, and there is quite a bit of flexibility in selecting the phase response of a filter. Phase only becomes an issue when it is perceived as a time-domain effect such as a delay, early or late echoes, or significant smearing of the audio signal caused by uneven group delays across the frequencies. One way to minimize the audible effect of uneven group delays is to restrict the maximum group delay of the filter. This can be accomplished by designing the audio filter to be minimum-phase.

To achieve minimum-phase, the technique of spectral factorization can be included into the filter design procedure. As each filter design algorithm in this thesis returns FIR filters with linear-phase, we can spectrally factor the output into maximum- and minimum-phase parts. Spectral factorization reduces the order an FIR filter into roughly half. For FIR filters with zeros on the unit circle, certain manipulations need to be performed on the input before the factoring. These procedures are described in detail by Schussler [10, pg. 468].

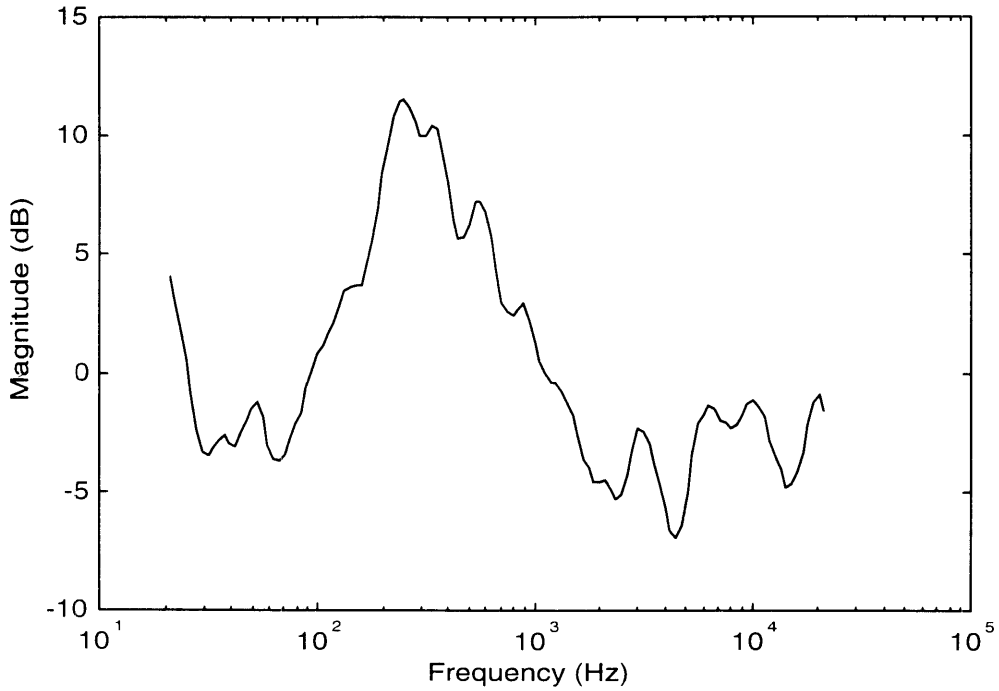
Since spectral factorization reduces the magnitude of both parts to only the square root of the specification, we must compensate by squaring the magnitude of the specifications before designing the target filter.

2.3 Sample Frequency Responses

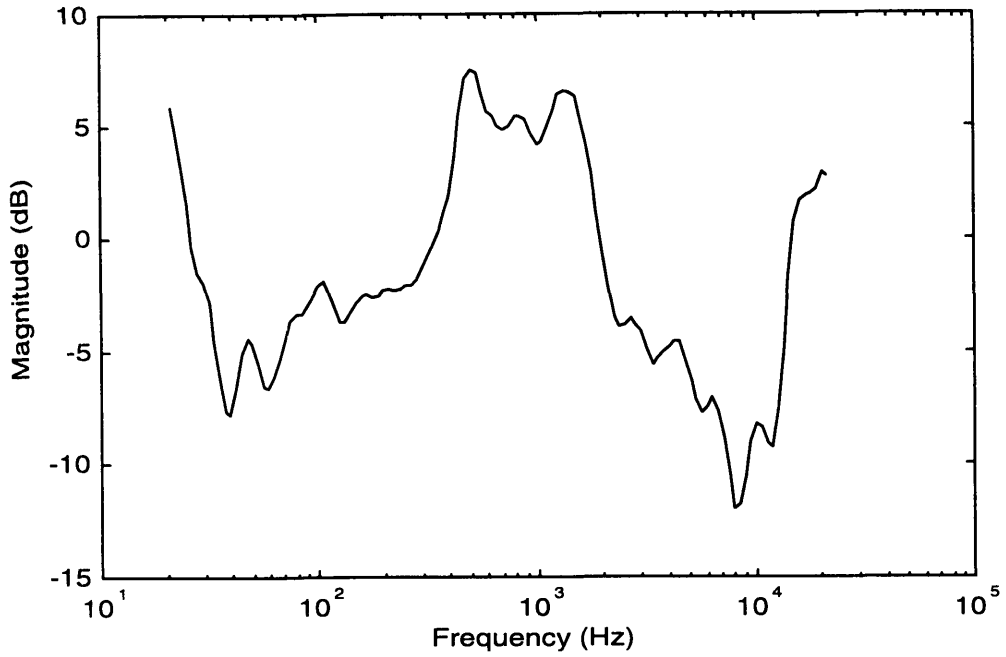
The filter design algorithm presented in this thesis was tested using a set of 20 prototypical frequency responses. Several of these responses are shown in Figure 2-1 (a) through (d).


15
10
5
0
-5
-10
Magnitude (dB)
10^1
10^2
10^3
10^4
10^5
Frequency (Hz)

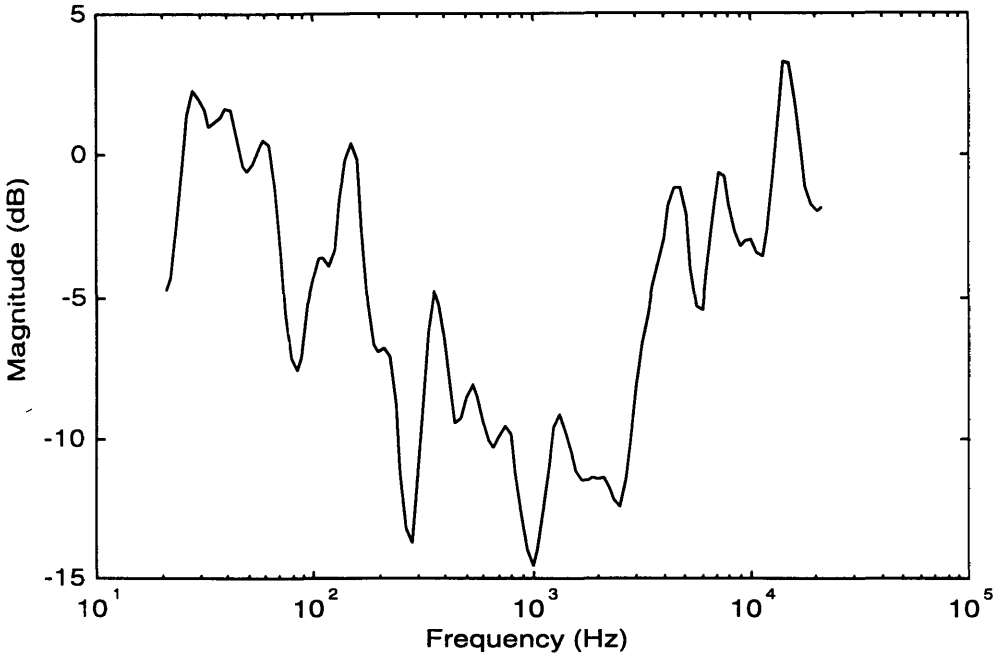
(a)


15
10
5
0
-5
-10
Magnitude (dB)
10^1
10^2
10^3
10^4
10^5
Frequency (Hz)

(b)



(c)



(d)

Figure 2-1 (a)–(d) Examples of Frequency Response Specifications

The responses were designed to model the equalization needed to flatten a speaker's measured frequency response in typical listening environments. These responses have a frequency resolution of 1/3 octave. That is, the details of the frequency response are spaced at roughly 1/3 octave.

This data is based on measurements made at Bose Corporation for a variety of speakers and listening environments, and represents averages over many positions within the same room. The target responses are a reasonable test of the performance of a filter design algorithm. Designing filters to actual measured data would produce similar results.

2.4 Just Noticeable Difference Curves

In order to optimally approximate the target frequency responses by a digital filter, we must have some error measure. Standard error measures which are frequently used, such as mean squared error, are inappropriate in this case, because they do not take into account the behavior of the auditory system.

We will take a slightly different approach to approximating the desired response. Due to limitations in the auditory system, the human ear is insensitive to small changes in frequency response. Thus, there is a whole set of filters which sounds indistinguishable from the desired filter. Our goal is to define this set and identify a filter which lies within.

The sensitivity of the human auditory system can be described using Just-Noticeable Differences, or JNDs. The JND for loudness is a frequency dependent

quantity. It has been experimentally determined through a set of subject-based listening tests [1] .

The listening tests proceed as follows. The listener is first presented with pink noise, and then with the same noise filtered by the shelf filter shown in Figure 2-2.

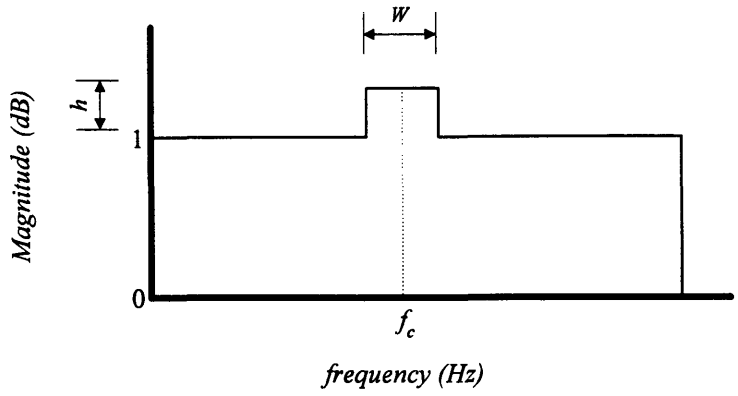


Figure 2-2 Frequency response of the filter used in obtaining the Just Noticeable Difference Curve.

The user is asked if the two signals presented were distinct or indistinguishable. The results of the experiment depend upon the details of the shelf filter: center frequency f_c, bandwidth W in octaves, and gain h.

The standard method of probing the limits of the auditory system is to keep f_c and W fixed, and then vary the gain h until the signals are "just noticeably different." The center frequency is changed, and the experiment repeated. This produces a curve which is a function of frequency and represents the level in dB at which the listener can detect a W-octave change in level. These curves are plotted in Figure

2-3 for W=1/3 octave, 1 octave, and 3 octaves, and represent results averaged over many subjects.

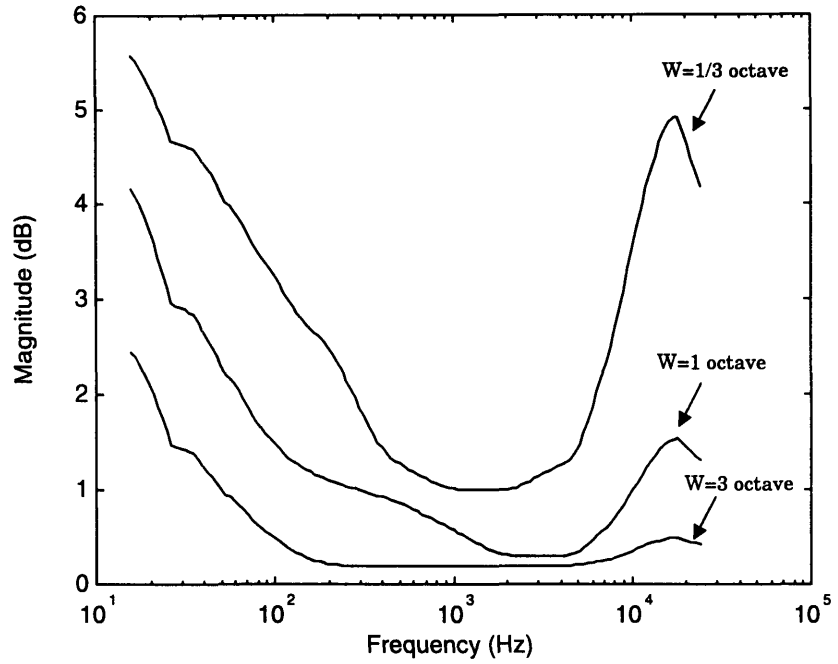


Figure 2-3 JND curves for *W*=1/3 octave, 1 octave and 3 octave.

As can be seen, the auditory system is most sensitive to frequencies between 2 and 5 kHz. Also, the larger *W*, the easier it is for subjects to detect differences.

Unfortunately, these results are difficult to apply to the problem of determining general constraints on a filter such that it sounds indistinguishable from another. Recall that the results are for a single perturbation of width *W*, and do not apply to simultaneous perturbations at multiple frequencies. In fact, no results from psychoacoustics have immediate application to our problem.

We will extend the results for loudness JND in a manner similar to that used to extend audio masking data [1]. An audio signal A can mask another signal B if it is sufficiently loud in level and close in frequency to B. Experiments were conducted for the case of a single sinusoid A masking a bandpass noise B. These results were then extended — without serious experimental evidence — to the case of multiple sinusoids masking multiple noises. It was shown that as long as the original masking criterion is obeyed at every frequency, the collection of sinusoids appropriately masks the collection of noises. This principle has been widely applied in the design of audio coders.

We will make a similar assumption in this thesis. For example, suppose that we choose the 1-octave JND curve. Given a desired frequency response, we derive upper and lower bounds by adding and subtracting (in dB) the JND curve. We will assume that any filter which falls completely between the upper and lower curves is audibly indistinguishable from the desired response.

There is one clear difficulty with this assumption. Suppose that the designed filter falls within the upper and lower bounds of the 1-octave JND curve. However, instead of having a response close to our desired curve which is vertically centered within the bounds, the designed filter has a response which is close to, say, the upper bound for most of the frequencies. Then this means that the width (in frequency) of the deviation from desired response can be wider than one octave. For example, suppose that this "ripple" is 2.5 octaves wide. In order to be inaudible, the height of this ripple must satisfy the tighter 3-octave JND bound instead of the 1-octave bound. There is no way to avoid this a priori. Instead, after the filter has

been designed, we will verify that the width of its ripples satisfy the 1-octave JND curve. Experimentally, we have found that this is always satisfied.

2.5 Limitations of Standard FIR and IIR techniques

There are several shortcomings of designing FIR filters to specifications on a broad frequency range. As specifications are given on log scales, the high density of the details in the low frequency region cause the FIR filter to be too long and too costly to implement. Moreover, the large order of the resulting filters require high-precision arithmetic operations in the design algorithm. Standard double precision would be insufficient so that the design algorithm would not converge at all. FFT-based algorithm such as overlap-add or overlap-save may be employed to reduce the computational complexity. Since these algorithms are block-based, they introduce substantial latency which may be inappropriate for some real-time applications.

The problem of existing IIR design techniques is that they cannot handle the required detail of the frequencies response. They often fail to converge to an acceptable solution and require constant supervision from the filter designer as IIR design procedures are usually not completely autonomous.

Chapter 3

Frequency Transformations

This chapter introduces the mathematical definition of a frequency transformation as well as some of its properties. To date, the technique of applying frequency transformations in audio filter design has not been used extensively, although the transformation has been recognized for over 20 years by Oppenheim and Johnson [2]. Classically, this technique was used in the design of filters. We will use it both in the design of filters and in their implementation. As a result, we are able to reduce filter lengths by a significant amount, typically by a factor of 80 or so. Results from filter designs using frequency transformation are summarized in Chapter 5.

3.1 Allpass Transformation

A *frequency transformation* or *frequency warping* in its most general sense is any mapping Θ of the z-plane, i.e., $\Theta(z^{-1})$ maps the complex plane to itself. A filter $g[n]$ is a frequency transformed version of another filter $h[n]$ if their z-transforms are related by a substitution of variables, or

$$G(z^{-1}) = H(\Theta(z^{-1})).^{+} \tag{3.1}$$

[+] In this chapter, we choose to write "$\Theta(z^{-1})$ is a rational function of z^{-1}" instead of "$\Theta(z)$ is a rational function of z" because the former sentence provides us with more

We will refer to $h[n]$ as the prototype filter and $g[n]$ as the transformed or warped filter.

We are only interested in frequency transformations that satisfy the following properties:

1. $\Theta(z^{-1})$ is a rational function of z^{-1}.
2. $\Theta(z^{-1})$ maps the unit circle to itself.
3. $\Theta(z^{-1})$ maps the interior of the unit circle to itself.

The first constraint ensures that the function $G(z^{-1})$ is a rational transform if $H(z^{-1})$ is rational. This is important because only filters with rational transforms may be realized. The second constraint ensures that the frequency response $G(e^{-j\omega})$ of the transformed filter is related to $H(e^{-j\omega})$ through a warping of the frequency axis. One way to visualize the response $G(e^{-j\omega})$ is to think of the frequency response $H(e^{-j\omega})$ but with the frequency axis rescaled, so that the ω–axis is "stretched out" in some region and "squeezed in" in some other. The third constraints ensures that $G(z^{-1})$ will be stable provided that $H(z^{-1})$ is, because all the poles will stay inside the unit circle with the warping.

A new idea developed in this thesis is to use the frequency transformation not only in the design phase, but also in the implementation of a filter. Just as the

---

understanding of the physical implementation; it states that we can substitute every delay with another filter $\Theta(z^{-1})$.

transformed and prototype filters are related by a substitution of variables, $G(z^{-1}) = H(\Theta(z^{-1}))$, the system structure for $G(z^{-1})$ can be derived by a direct substitution of

$$z^{-1} \rightarrow \Theta(z^{-1}) \tag{3.2}$$

into the system structure for $H(z^{-1})$. This is illustrated in Figure 3-1.

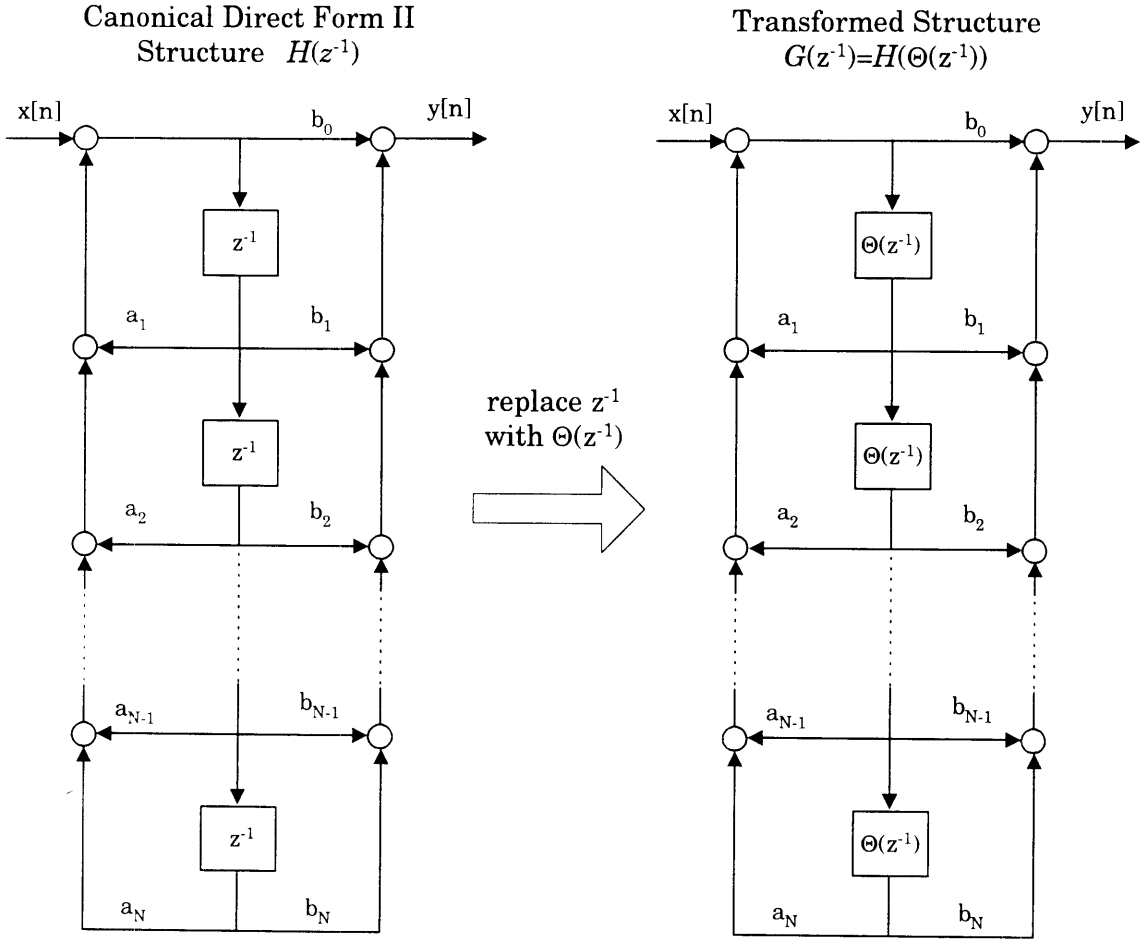


Figure 3-1 Derivation of $G(z^{-1})$ based on a substitution of variables in $H(z^{-1})$.

It has been shown in [3, pg. 432] and [4] that the most general form of $\Theta(z^{-1})$ that satisfies the above three properties is

$$\Theta(z^{-1}) = \pm\prod_{1}^{N}\frac{z^{-1}-\alpha_k}{1-\alpha_k z^{-1}} \quad \text{for } |\alpha_k| < 1. \tag{3.3}$$

In other words, $\Theta(z^{-1})$ is a cascade of first-order allpass filters.

We can constrain the choice of $\Theta(z^{-1})$ by requiring that the mapping from $H(e^{-j\omega})$ to $G(e^{-j\omega})$ be one-to-one. Otherwise, $G(e^{-j\omega})$ would contain multiple compressed copies of $H(e^{-j\omega})$ and would greatly limit the types of frequency responses which can be obtained. The constraint that $\Theta(z^{-1})$ be one-to-one limits the choice of frequency transformations to first-order, real allpass functions

$$\Theta_a(z^{-1}) = \frac{z^{-1}-a}{1-az^{-1}}, \text{ where } a \text{ is real and } |a| < 1. \tag{3.4}$$

We will call this type of transformation an *allpass transformation*. Since, by choice, this transformation maps the unit circle to itself, we can relate a frequency θ of the prototype filter with a frequency ω of the warped filter by

$$e^{j\theta} = \frac{e^{-j\omega}-a}{1-ae^{-j\omega}} \text{ for real } \theta, \omega, a \text{ and } |a| < 1. \tag{3.5}$$

from which it follows that

$$\omega = \arctan\left[\frac{(1-\alpha^2)\sin\theta}{2\alpha+(1+\alpha^2)\cos\theta}\right] \tag{3.6}$$

or equivalently,

$$\omega = \omega + 2\arctan\left[\frac{a\sin\theta}{1-a\cos\theta}\right]. \tag{3.7}$$

Notice that the function $\Theta_a(z^{-1})$ is bijective and its inverse is simply another allpass filter whose warping parameter is $-a$. That is,

$$\Theta_a^{-1}(z^{-1}) = \frac{z^{-1} + a}{1 + az^{-1}} = \Theta_{-a}(z^{-1}). \tag{3.8}$$

To summarize, a function $G(z^{-1})$ is a frequency transformation of $H(z^{-1})$ if

$$G(z^{-1}) = H(\Theta_a(z^{-1})) = H(\frac{z^{-1} - a}{1 - az^{-1}}). \tag{3.9}$$

For rational filters, $G(z^{-1})$ can be obtained from $H(z^{-1})$ by replacing every delay with an allpass filter. The parameter a in $\Theta_a(z^{-1})$ is called the *warping factor*. It is a free parameter and gives us some control between the warping from $H(e^{-j\omega})$ to $G(e^{-j\omega})$. An example of the function $\Theta_a(e^{-j\omega})$ for a few values of a is plotted below.

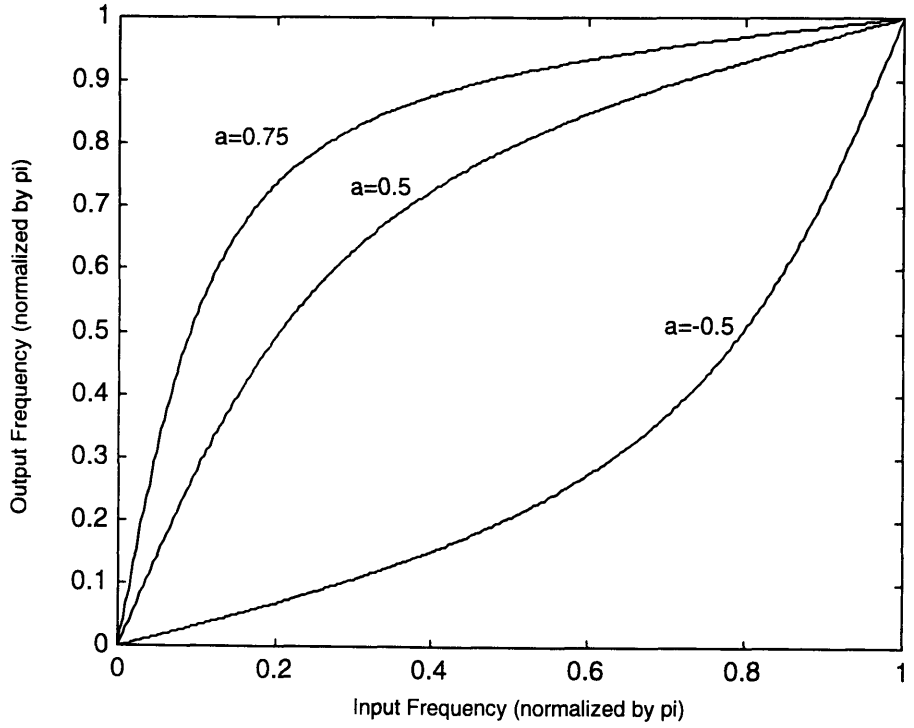


Figure 3-2 Frequencies warped by Allpass frequency transformations with different parameters.

For $0 < a < 1$, the effect of the allpass transformation is to stretch the low frequencies of $H(e^{-j\omega})$ and compress the high frequencies. Similarly, for $-1 < a < 0$, the low frequencies are compressed and the high frequencies stretched.

This stretching and compressing of the frequency axis is the key benefit of frequency warping and yields a substantial reduction in filter order.

As discussed earlier, an audio filter is often specified on a logarithmic scale due to human auditory perception. If we plot out a typical response specification on a linear scale, we would find that the filter detail is very dense in the low-frequency region, and sparse (or monotonic) in the high-frequency. It is conjectured that the narrow features of the frequency response down in the low-frequency range are the main cause for long FIR filters. An intuitive understanding of this conjecture can be obtained by considering the design of a bandpass filter. The narrower the passband we require, the longer the FIR filter will be. Since an arbitrary audio filter can be approximated as many passband filters connected in parallel, the minimum order of an FIR filter that meets the requirement is dictated by the narrowest passband. Hence, by applying a frequency transformation, we hope to increase the width of the narrow features of the frequency response and thereby decrease the overall order of the filter. This conjecture has been confirmed experimentally with the filter design results in Chapter 5.

3.2 Properties of Allpass Transformations

In this section, we show several properties of allpass transformation which are of interest to audio applications. We will assume that $h[n]$ is an FIR filter with z-

transform $H(z^{-1})$. The filter $G(z^{-1})$ is defined to be the allpass frequency-transformed (or *warped*) version of $H(z^{-1})$ with some warping parameter a as in (3.9).

3.2.1 Order Preserving

Suppose that the FIR filter $h[n]$ has N+1 filter coefficients, or equivalently, N zeros. Then after allpass transformation, the warped filter $G(z^{-1}) = H(\Theta_a(z^{-1}))$ shall be a rational IIR filter, with N zeros and N poles. Moreover, all the poles occur at a, the warping factor. This property can shown by direct substitution,

$$\begin{aligned} H(z^{-1}) &= h[0] + h[1]z^{-1} + h[2]z^{-2} + \ldots + h[N]z^{-N} \\ G(z^{-1}) &= H\left(\frac{z^{-1}-a}{1-az^{-1}}\right) \\ &= h[0] + h[1]\frac{z^{-1}-a}{1-az^{-1}} + h[2]\left(\frac{z^{-1}-a}{1-az^{-1}}\right)^2 + \ldots + h[N]\left(\frac{z^{-1}-a}{1-az^{-1}}\right)^N \\ &= \frac{\tilde{h}[0] + \tilde{h}[1]z^{-1} + \tilde{h}[2]z^{-2} + \ldots + \tilde{h}[N]z^{-N}}{(1-az^{-1})^N}. \end{aligned} \tag{3.10}$$

Thus, although allpass transformations changes the filter from FIR to IIR, it preserves its order.

3.2.2 Allpass Property Preserving

It is surprising that if $H(z^{-1})$ is an allpass filter, then $G(z^{-1})$ will be an allpass filter too, since the allpass transformation maps the unit circle to itself. However, for completeness, we have included a proof here.

The most general form of an allpass filter $H(z^{-1})$ is

$$H(z^{-1}) = \prod_{k=1}^{Mr} \frac{z^{-1}-d_k}{1-d_k z^{-1}} \prod_{k=1}^{M_c} \frac{(z^{-1}-e_k^*)(z^{-1}-e_k)}{(1-e_k z^{-1})(1-e_k^* z^{-1})} \tag{3.11}$$

where the d_k's are the real poles and the e_k's are the complex poles of $H(z^{-1})$.

Then by warping the filter $H(z^{-1})$ with parameter a, we get

$$\begin{aligned} G(z^{-1}) &= H\left(\frac{z^{-1}-a}{1-az^{-1}}\right) \\ &= \prod_{k=1}^{Mr} \frac{\dfrac{z^{-1}-a}{1-az^{-1}} - d_k}{1 - d_k \dfrac{z^{-1}-a}{1-az^{-1}}} \prod_{k=1}^{M_c} \frac{\left(\dfrac{z^{-1}-a}{1-az^{-1}} - e_k^*\right)\left(\dfrac{z^{-1}-a}{1-az^{-1}} - e_k\right)}{\left(1 - e_k \dfrac{z^{-1}-a}{1-az^{-1}}\right)\left(1 - e_k^* \dfrac{z^{-1}-a}{1-az^{-1}}\right)}. \end{aligned} \tag{3.12}$$

To help manipulate the complicated expressions in (3.12), we consider the effect of warping on the real and the complex sections separately. For each real section with a pole at d_k, the warped filter can be reduced to

$$\begin{aligned} \frac{\dfrac{z^{-1}-a}{1-az^{-1}} - d_k}{1 - d_k \dfrac{z^{-1}-a}{1-az^{-1}}} &= \frac{z^{-1}(1+ad_k) - (a+d_k)}{(1+ad_k) - (a+d_k)z^{-1}} \\ &= \frac{z^{-1} - \dfrac{a+d_k}{1+ad_k}}{1 - \dfrac{a+d_k}{1+ad_k} z^{-1}}. \end{aligned} \tag{3.13}$$

Similarly, each section corresponding to complex-conjugate poles e_k and e_k^* can be reduced to

$$\frac{\frac{z^{-1}-a}{1-az^{-1}} - e_k^*}{1 - e_k \frac{z^{-1}-a}{1-az^{-1}}} \cdot \frac{\frac{z^{-1}-a}{1-az^{-1}} - e_k}{1 - e_k^* \frac{z^{-1}-a}{1-az^{-1}}} = \frac{z^{-1}(1+ae_k^*) - (a+e_k^*)}{(1+ae_k) - (a+e_k)z^{-1}} \cdot \frac{z^{-1}(1+ae_k) - (a+e_k)}{(1+ae_k^*) - (a+e_k^*)z^{-1}}$$

$$= \frac{1+ae_k^*}{1+ae_k} \cdot \frac{z^{-1} - \frac{a+e_k^*}{1+ae_k^*}}{1 - \frac{a+e_k}{1+ae_k} z^{-1}} \cdot \frac{1+ae_k}{1+ae_k^*} \cdot \frac{z^{-1} - \frac{a+e_k}{1+ae_k}}{1 - \frac{a+e_k^*}{1+ae_k^*} z^{-1}} \quad (3.14)$$

$$= \frac{z^{-1} - \frac{a+e_k^*}{1+ae_k^*}}{1 - \frac{a+e_k}{1+ae_k} z^{-1}} \cdot \frac{z^{-1} - \frac{a+e_k}{1+ae_k}}{1 - \frac{a+e_k^*}{1+ae_k^*} z^{-1}}.$$

Substituting (3.13) and (3.14) into (3.12), the warped filter $G(z^{-1})$ can be expressed as

$$G(z^{-1}) = \prod_{k=1}^{Mr} \frac{z^{-1} - r_k}{1 - r_k z^{-1}} \prod_{k=1}^{M_c} \frac{(z^{-1} - s_k^*)(z^{-1} - s_k)}{(1 - s_k z^{-1})(1 - s_k^* z^{-1})} \quad (3.15)$$

where

$$r_k = \frac{a + d_k}{1 + ad_k}$$

$$s_k = \frac{a + e_k}{1 + ae_k}$$

are the new sets of poles. Since the form of $G(z^{-1})$ in (3.15) is that of an allpass filter, we conclude that allpass transformations preserve the allpass property of a filter.

The allpass preserving property of the allpass transformation is not directly useful for designing audio filters. However, it is useful for the proof of the next property which is much more significant: the minimum-phase preserving property.

3.2.3 Minimum-phase Preserving

The allpass transformation also preserves the minimum-phase property of a filter. Namely, if $H(z^{-1})$ is a minimum-phase filter, then so is the warped version $G(z^{-1})$. As discussed earlier, this property is important to audio filters because a minimum-phase filter possesses the minimum energy delay property. The easiest way to see why this is true is to notice that the poles and zeros of $H(z^{-1})$ which are inside the unit circle will always stay inside after warping. This is because, by choice, $\Theta_a(z^{-1})$ must map the interior of the unit circle to itself.

Another intuitive argument of why an allpass transformation preserves the minimum-phase property relies on the invertibility of the allpass transformation $\Theta_a(z^{-1})$. Given a filter $H(z^{-1})$, we can factor it into minimum-phase and the allpass components as shown

$$H(z^{-1}) = H_{ap}(z^{-1}) \cdot H_{min}(z^{-1}). \tag{3.16}$$

On the other hand, the transformed filter $G(z^{-1}) = H(\Theta_a(z^{-1}))$ can also be factored in the same way,

$$G(z^{-1}) = G_{ap}(z^{-1}) \cdot G_{min}(z^{-1}). \tag{3.17}$$

Since we have already shown that the transformation preserves the allpass property, $G_{ap}(z^{-1})$ must contain the transformation of $H_{ap}(z^{-1})$, which is an allpass filter, plus possibly some extra allpass filter $K_{ap}(z^{-1})$. That is,

$$G_{ap}(z^{-1}) = H_{ap}(\Theta_a(z^{-1})) \cdot K_{ap}(z^{-1}). \tag{3.18}$$

Conversely, since $\Theta(z^{-1})$ is invertible, we can transform $G_{ap}(z^{-1})$ by $\Theta_a^{-1}(z^{-1})$ to obtain another allpass filter. Moreover, $G_{ap}(\Theta_a^{-1}(z^{-1}))$ must be included in $H_{ap}(z^{-1})$ because, by definition, $H_{ap}(z^{-1})$ encompasses the entire allpass portion of $H(z^{-1})$. Thus,

$$H_{ap}(z^{-1}) = G_{ap}(\Theta_a^{-1}(z^{-1}))\, C_{ap}(z^{-1}) \tag{3.19}$$

where $C_{ap}(z^{-1})$ is any extra term that $H_{ap}(z^{-1})$ might contain. Then by replacing every z^{-1} in (3.19) with $\Theta_a(z^{-1})$, we have a transformation of $H_{ap}(z^{-1})$ again.

$$H_{ap}(\Theta_a(z^{-1})) = G_{ap}(z^{-1})\, C_{ap}(\Theta_a(z^{-1})) \tag{3.20}$$

This implies that $H_{ap}(\Theta_a(z^{-1}))$ contains at least every term of $G_{ap}(z^{-1})$. However, (3.18) also says that $G_{ap}(z^{-1})$ contains at least every term of $H_{ap}(\Theta_a(z^{-1}))$. Therefore,

$$G_{ap}(z^{-1}) = H_{ap}(\Theta_a(z^{-1})) \tag{3.21}$$

which implies that

$$G_{min}(z^{-1}) = H_{min}(\,\Theta_a(z^{-1})). \tag{3.22}$$

A more formal proof is done by showing that each singularity (pole and zero) of $G(z^{-1})$ is within the unit circle as long as the corresponding singularity of $H(z^{-1})$ is within the unit circle. Let us assume $H(z^{-1})$ is a rational, minimum-phase filter with complex poles and zeros,

$$H(z^{-1}) = \frac{(1 - s_1 z^{-1})\,(1 - s_2 z^{-1}) \quad \cdots \quad (1 - s_n z^{-1})}{(1 - p_1 z^{-1})\,(1 - p_2 z^{-1}) \quad \cdots \quad (1 - p_n z^{-1})}. \tag{3.23}$$

Since $H(z^{-1})$ is minimum-phase, there is an equal number of poles and zeros. Furthermore, $|s_i| < 1$ and $|p_i| < 1$. Let us transform $H(z^{-1})$ with parameter a so that

$$
\begin{aligned}
G(z^{-1}) &= H(\Theta(z^{-1})) \\
&= H\left(\frac{z^{-1}-a}{1-az^{-1}}\right) \\
&= \frac{(1-s_1\frac{z^{-1}-a}{1-az^{-1}})(1-s_2\frac{z^{-1}-a}{1-az^{-1}}) \quad \cdots \quad (1-s_n\frac{z^{-1}-a}{1-az^{-1}})}{(1-p_1\frac{z^{-1}-a}{1-az^{-1}})(1-p_2\frac{z^{-1}-a}{1-az^{-1}}) \quad \cdots \quad (1-p_n\frac{z^{-1}-a}{1-az^{-1}})}.
\end{aligned}
\tag{3.24}
$$

Each term in the numerator and denominator can be simplified as

$$
\begin{aligned}
1-s_i\frac{z^{-1}-a}{1-az^{-1}} &= \frac{1-az^{-1}-s_iz^{-1}+s_ia}{1-az^{-1}} \\
&= \frac{1+s_ia-(a+s_i)z^{-1}}{1-az^{-1}} \\
&= (1+s_ia)\left(\frac{1-\frac{a+s_i}{1+s_ia}z^{-1}}{1-az^{-1}}\right).
\end{aligned}
\tag{3.25}
$$

By substituting expression (3.25) into (3.24), we obtain

$$
G(z^{-1}) = K\frac{(1-c_1z^{-1})(1-c_2z^{-1}) \quad \cdots \quad (1-c_nz^{-1})}{(1-d_1z^{-1})(1-d_2z^{-1}) \quad \cdots \quad (1-d_nz^{-1})} \tag{3.26}
$$

where

$$
\begin{aligned}
K &= \prod_{i=1}^{n}\frac{1+s_ia}{1+p_ia}, \\
c_i &= \frac{a+s_i}{1+s_ia} \text{ and } d_i = \frac{a+p_i}{1+p_ia}.
\end{aligned}
\tag{3.27}
$$

With this new expression for $G(z^{-1})$, the question is whether the new poles and zeros are inside the unit circle. That is, we must show that $|c_i| < 1$ as well as $|d_i| < 1$. To see why this is true, let us rewrite the magnitude squared of the numerator of c_i as

$$
\begin{aligned}
|a+s_i|^2 &= (a+s_i)(a+\bar{s}_i) \\
&= a^2 + as_i + a\bar{s}_i + |s_i|^2 \\
&= a^2 + 2a\,\mathrm{Re}\{s_i\} + |s_i|^2.
\end{aligned}
\tag{3.28}
$$

The denominator can be written as

$$\begin{aligned} \left|1+as_i\right|^2 &= (1+as_i)(1+a\bar{s}_i) \\ &= 1+as_i+a\bar{s}_i+a^2\left|s_i\right|^2 \\ &= 1+2a\,\mathrm{Re}\{s_i\}+a^2\left|s_i\right|^2. \end{aligned} \tag{3.29}$$

Since $0 \le |a| < 1$, and $0 \le |s_i| < 1$, we can write them as

$$\begin{aligned} a^2 &= 1-\varepsilon \quad \text{for } 0<\varepsilon\le 1 \\ \left|s_i\right|^2 &= 1-\sigma \quad \text{for } 0<\sigma\le 1. \end{aligned} \tag{3.30}$$

By substituting (3.30) into the expression for the numerator c_i in (3.28), we get

$$\begin{aligned} \left|a+s_i\right|^2 &= (1-\varepsilon)+2a\,\mathrm{Re}\{s_i\}+(1-\sigma) \\ &= 2+2a\,\mathrm{Re}\{s_i\}-\varepsilon-\sigma. \end{aligned} \tag{3.31}$$

Similarly, the denominator of c_i in (3.29) becomes

$$\begin{aligned} \left|1+as_i\right|^2 &= 1+2a\,\mathrm{Re}\{s_i\}+(1-\varepsilon)(1-\sigma) \\ &= 2+2a\,\mathrm{Re}\{s_i\}-\varepsilon-\sigma+\varepsilon\sigma \\ &= \left|a+s_i\right|^2+\varepsilon\sigma \\ &> \left|a+s_i\right|^2. \end{aligned} \tag{3.32}$$

Therefore,

$$\left|c_i\right| = \left|\frac{a+s_i}{1+as_i}\right| < 1. \tag{3.33}$$

By repeating the argument above with s_i replaced with p_i, we can conclude that

$$\left|d_i\right| = \left|\frac{a+p_i}{1+ap_i}\right| < 1. \tag{3.34}$$

Thus, all of the singularities of $G(z^{-1})$ are within the unit circle and therefore, $G(z^{-1})$ is a minimum-phase filter.

3.3 Design Procedure with Frequency Transformations

This section is an overview of the entire design procedure when the frequency transformation technique is used. It also discusses how one chooses a warping factor so that the filter order might be minimized.

3.3.1 Summary of Design Steps

First we are given a target frequency response $H_d(f)$ on a logarithmic scale. We apply the Just-Noticeable Difference (JND) curve to obtain the upper and lower tolerance bounds, $U(f)$ and $L(f)$ as shown below.

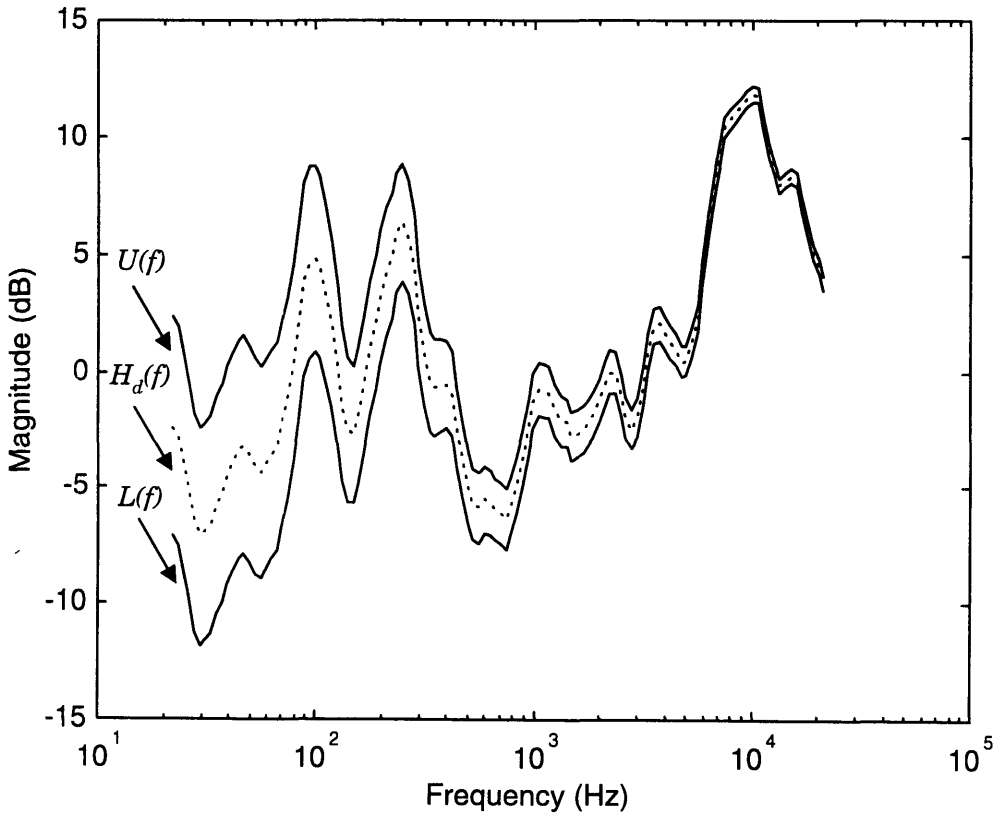


Figure 3-3 A typical target Frequency Response $H_d(f)$ with Just-Noticeable Difference bounds $U(f)$ and $L(f)$.

Notice that the magnitude specification above is in dB or $20\log_{10}$(Magnitude), and that the frequency spans roughly the entire audible range (20 Hz to 20 kHz). On a linear scale normalized by sampling frequency, the upper and lower bounds, $U(e^{-j\omega})$ and $L(e^{-j\omega})$ have a very dense low-frequency specification as shown in Figure 3-4.

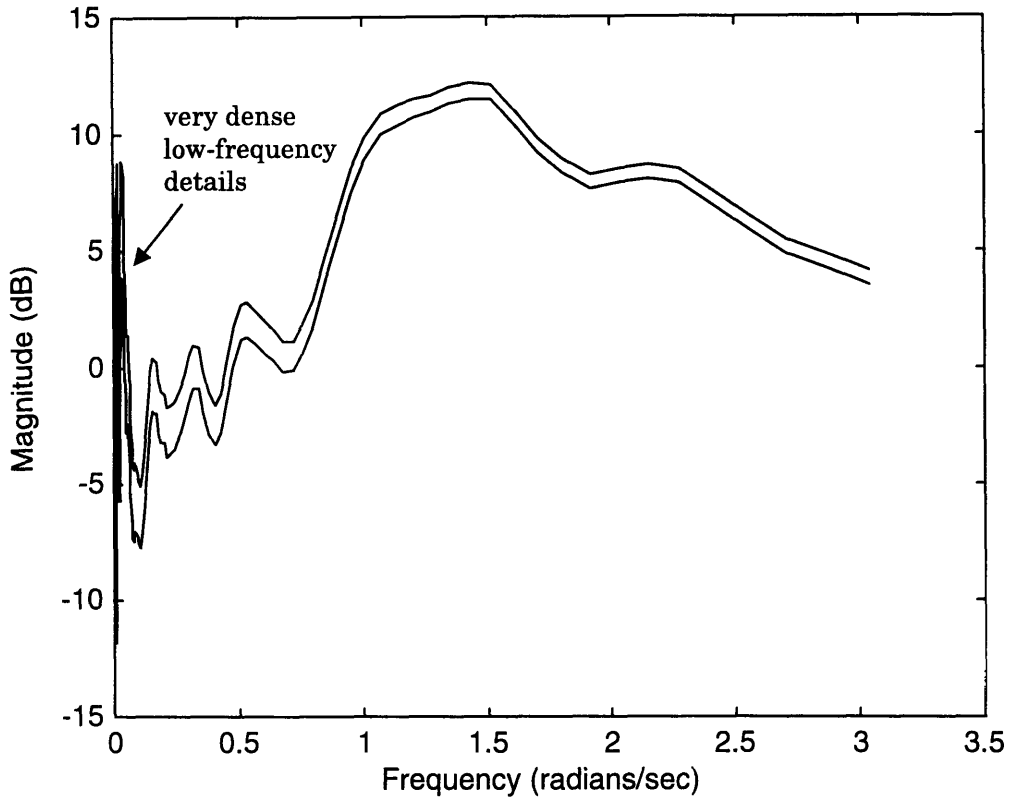


Figure 3-4 JND bounds $U(e^{-j\omega})$ and $L(e^{-j\omega})$ plotted on a linear, sampling-frequency normalized scale.

Next, we use the allpass frequency transformation to "stretch out" the crowded low-frequency region and simultaneously "squeeze in" the high-frequency region. How we determine the parameter a in the transformation is explained in the next section. Suppose we choose the warping factor a to be 0.93. The frequency warped

filter $H_d(\Theta_a(e^{-j\omega}))$ now looks much more similar to its original log-scale specification, as in Figure 3-5.

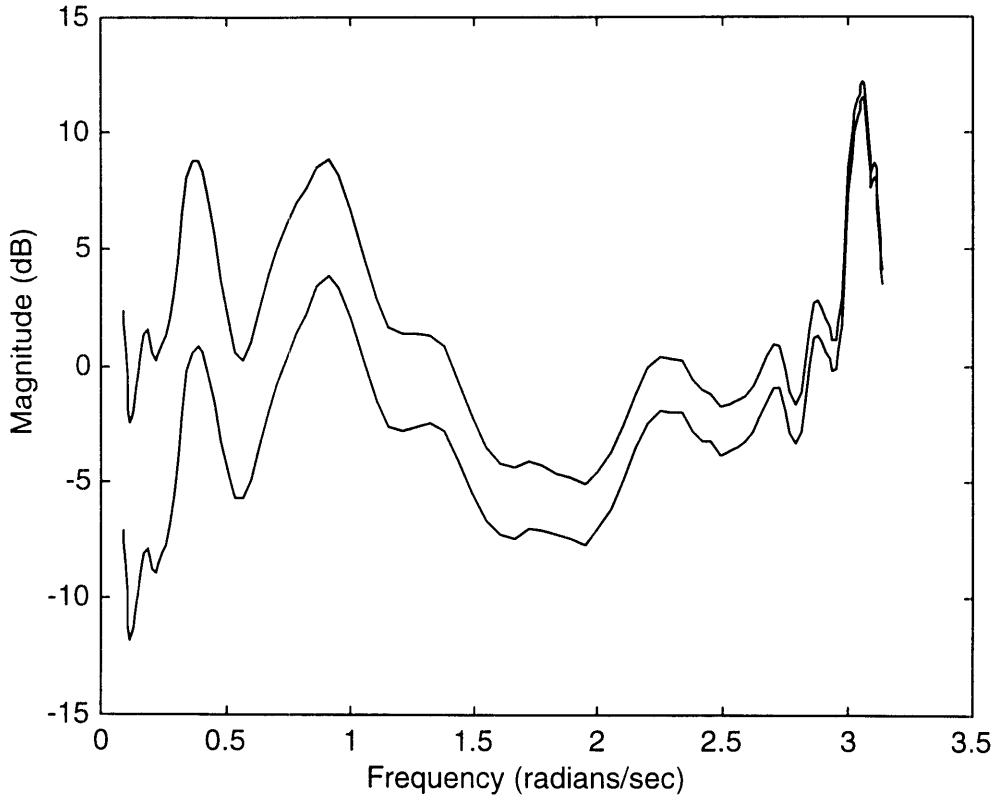


Figure 3-5 Warped target response $H_d(\Theta_a(e^{-j\omega}))$.

We now set off to design a filter that lies within those error bounds. We choose to design an FIR filter instead of IIR because of the availability of many design programs. For this example with a=0.93, the shortest FIR filter has a length of 68 points. Figure 3-6 below confirms that its magnitude meets our constraints.

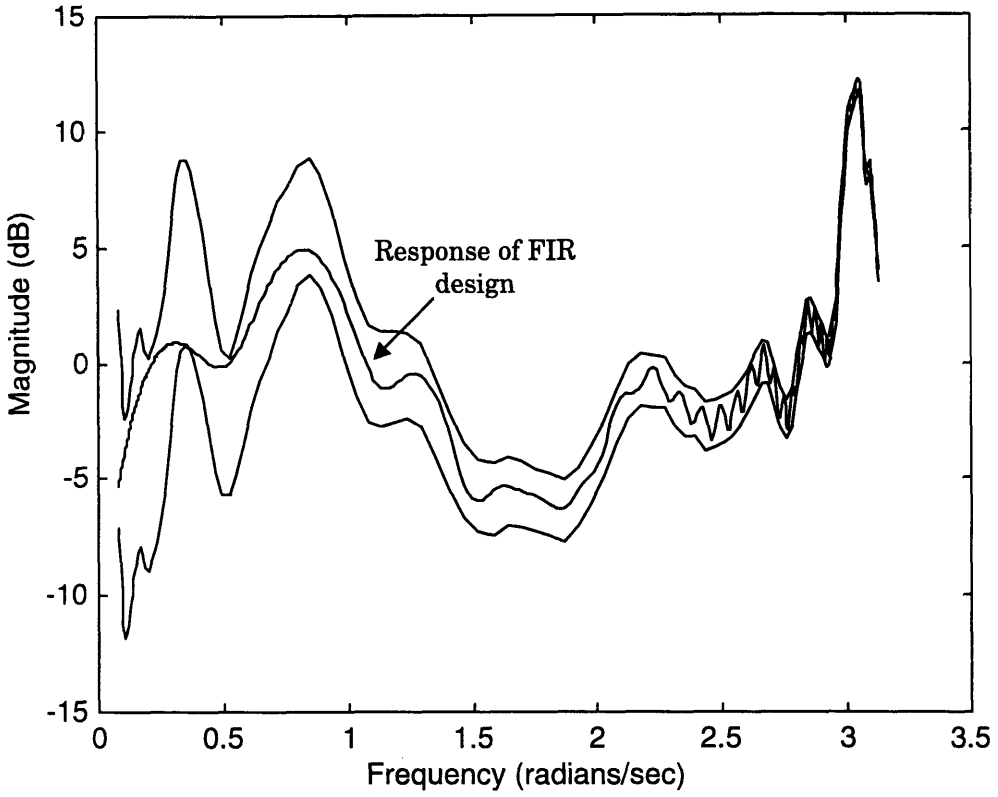


Figure 3-6 FIR filter design results

To implement the actual filter, we simply build the FIR filter network and inverse-warp it by replacing all the delays with allpass filters with parameter a= –0.93. The final frequency response of the filter implemented using frequency warping is shown below.

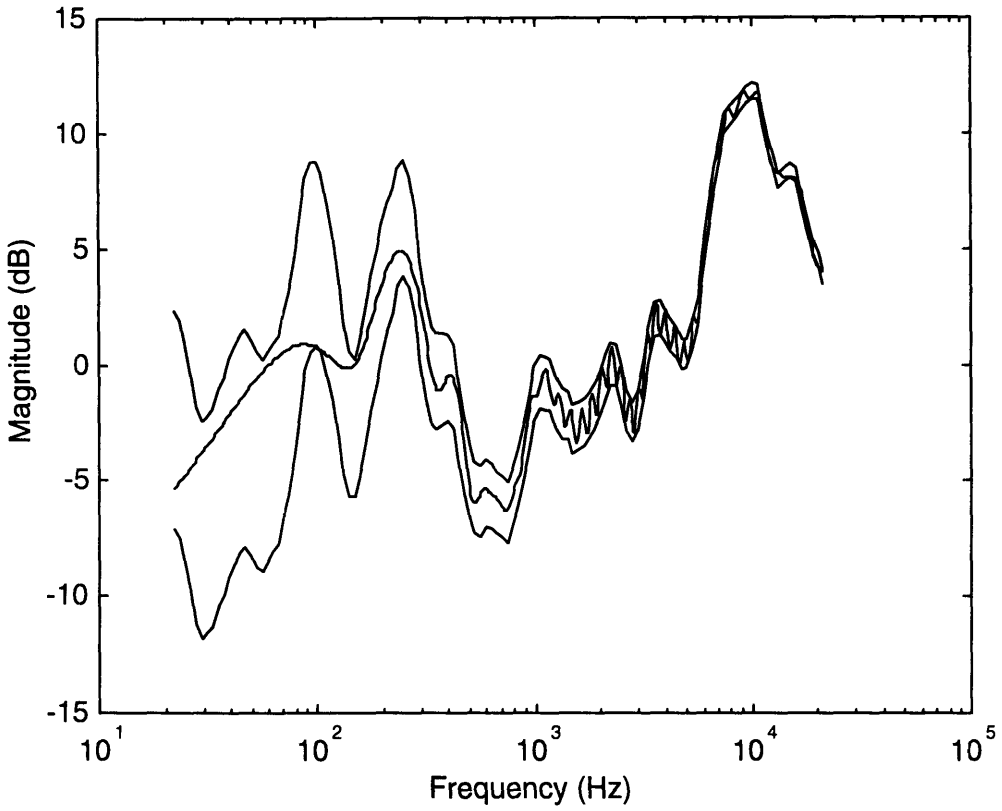


Figure 3-7 Effective frequency response of the final system

As described in section 2.2, in general we can reduce the order of the FIR filter into half by spectrally factoring out the minimum-phase part before the final warping. However, since spectral factorizations reduce the magnitude of filter to only its square root, we must compensate by squaring the magnitude of the input upper and lower bounds before the first warping. The entire design procedure is summarized into the following diagram.

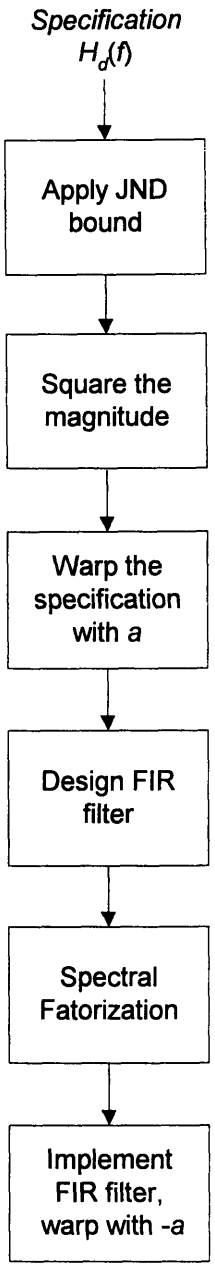


Figure 3-8 Summary of Design steps.

3.4 Choosing A Warping Factor

As mentioned earlier, it is only a conjecture that the frequency warping technique is beneficial to digital filter design when the desired response is given on a logarithmic

scale. We made no attempt in proving it. Nor did we mathematically characterize the "shape" of the frequency response that would yield the low filter order. However, experiments have shown such positive results (savings of nearly two orders of magnitude in filter order) that we believe it is a rather plausible assumption and we state it loosely here.

Assumption 1 (*Even Spread*): Given a fixed level of error, the order of the (rational) filter that meets a desired frequency response is lowest when the details of the desired response are spread out, or warped, as evenly as possible in frequency. This is when considering the response on a linear frequency scale.

The next question is naturally in the choice of the allpass transformation that achieves the above assumption. In other words, we must pick the warping parameter a in the allpass transformation

$$\Theta_a(z^{-1}) = \frac{z^{-1} - a}{1 - az^{-1}} \tag{3.35}$$

such that the desired frequency response is spread out as evenly as possible when plotted on a linear scale. From Figure 3-1, we know that the optimal value of a must be positive because positive values of a magnify the low-frequency region where the details of a specification are most crowded. We predict that the relation between the values of a (in the range [–1,1]) and the corresponding minimal FIR filter is bitonic in a; the order first decreases monotonically until it reaches a minimum and then increases monotonically, as shown.

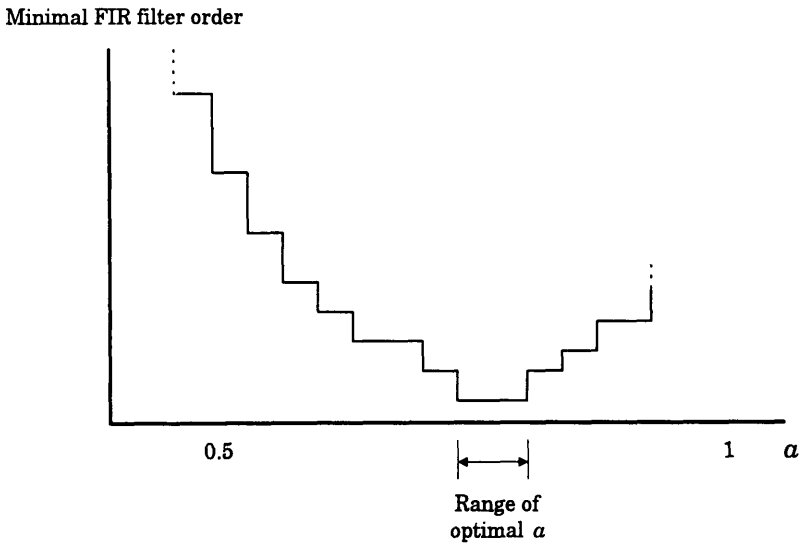


Figure 3-9 Prediction of the relation between the minimal possible filter order and the warping factor.

From the figure, it is worth noticing that the because the filter order is discrete, the plot exhibits a stair-like shape. Secondly, because slight variations in a need not change the order of the filter, the optimal warping parameter is not single-valued, but lies in a range of values.

In practice, it would be much more convenient to have an a priori estimate of the optimal warping factor instead of searching by trial-and-error for all a between 0 and 1. In order to obtain a closed-form solution for such an estimate, we formulate the problem as follows.

Assume that the desired frequency response takes the form of N+1 piecewise constant, logarithmically spaced bands, called band 0 through band N. Let the *log-scale center* (in radians/sec) of the ith band be

$$B_i = B_0 r^i \quad \text{for } 0 \le i \le N \text{ and for some } r > 1 \tag{3.36}$$

where B_0 is the log-scale center of the lowest band. r controls the spacing between adjacent bands. For example, suppose that a sample of 44.1 kHz is used, the first band is centered at 25 Hz, and the band spacing is one-third octave, Then

$$B_0 = \frac{2\pi \cdot 25}{44100} = 3.56 \times 10^{-3} \text{ rad / sec} \tag{3.37}$$

and

$$r = 2^{1/3}. \tag{3.38}$$

We will assume the ith band occupies the frequency in the range

$$\left[B_i r^{-1/2}, B_i r^{1/2}\right). \tag{3.39}$$

Therefore, the width of ith band is

$$W_i = B_i r^{\frac{1}{2}} - B_i r^{-\frac{1}{2}}. \tag{3.40}$$

Define the *linear-scale center* of the ith band to be the average of the upper and lower boundary, i.e.,

$$C_i = \tfrac{1}{2}(B_i r^{\frac{1}{2}} + B_i r^{-\frac{1}{2}}). \tag{3.41}$$

For convenience, we will now replace the notation $\Theta_a(e^{-j\omega})$ with $\Theta_a(\omega)$. After frequency transformation, the width of the ith band will become

$$\hat{W}_{a,i} = \Theta_a(B_i r^{\frac{1}{2}}) - \Theta_a(B_i r^{-\frac{1}{2}}). \tag{3.42}$$

Let the *log-optimal warping factor* $\hat{a}$ be the one which maximizes the narrowest band after warping, or more formally,

$$\hat{a} = \arg\max_{-1<a<1} \min_{0 \le i \le N-1} \hat{W}_{a,i}. \tag{3.43}$$

This minimax problem can be solved numerically using an exhaustive search by a computer. It might seem at first difficult to perform an exhaustive search over the variable a since it is a continuous variable. However, experience has showed that $\hat{a}$ is usually in the range between 0.8 and 1. This allows us to search through a discretized set of a within a restricted range.

As an example, suppose we design an equalizer for the entire audible frequency range which extends from 20 Hz to 20 kHz. There are 30 bands (band 0 through 29) and each band is spaced apart at one-third octave ($r=2^{1/3}$). The first log-scale band center is fixed at 25 Hz. Assuming that the sampling rate is at 44.1 kHz, then by a searching through the values of a discretized to multiples of 0.001, we find that $\hat{a} = 0.933$. We will use this value of the warping factor in the designs in Chapter 5.

Chapter 4

Polynomial Fitting within Upper and Lower Bounds

As described earlier, our specification for an audio filter will take the form of two real boundary functions, the upper and lower bound. We look for filters whose magnitude responses are vertically bounded by those two functions, *and* have a maximum ripple width less than some fixed bandwidth.

Our design procedure is as follows. First, we ignore the second requirement about the maximum ripple width for a moment, and only search for filters that fall within the prescribed magnitude bounds. We will call this the *Constrained Ripple* problem. Then, after obtaining such filters, we will verify that they also satisfy the ripple width constraint. This chapter concerns only the first part: designing filters that fit within the upper and lower bounds.

We will concentrate only on Type I filters. That is, a filter $h[n]$ with an odd length $M+1$ that satisfies $h[n]=h[M-n]$. From Oppenheim and Schafer [3, pg. 465], a type I linear-phase FIR filter can be transformed to a polynomial, and vice versa. By applying this polynomial transformation to the upper and lower bounds, we effectively get two polynomials in the range [–1,1]. We will refer to the upper and lower bound as $U(x)$ and $L(x)$ respectively. Our job now is to find a polynomial that

lies in between these polynomials. Therefore, our discussion in this chapter will only be based on a polynomial in the variable x. Once we find that polynomial, we can inverse-transform it back to an FIR filter and then warp $h[n]$ to obtain the audio equalizer $g[n]$.

In this chapter, we describe two design algorithms which return such polynomials: Parks-McClellan and CONRIP. We will emphasize the lesser known algorithm discovered by M. T. McCallig [5], [6]. The algorithm CONRIP (CONstrained RIPple) iteratively finds a polynomial that meets the upper and lower bound constraints. Moreover, the algorithm guarantees that the polynomials found has the minimum possible length of all valid polynomials.

We chose not to analyze the performance of the windowing design method, because this method cannot be performed without supervision. In Parks-McClellan and CONRIP, the algorithms always perform a solution search as exhaustively as they can for each given filter order. If the algorithms fail to find a solution, the only way to proceed is to keep increasing the order until a solution is found. On the other hand, when a windowing design method fails for a given filter order, it is unknown whether the correct approach is to increase the filter order or to modify the target response. This is due to the lack of direction in choosing the target frequency response, which is allowed to lie anywhere within the vertical bounds. Certainly, we may take an arbitrary approach in increasing the order every time windowing fails, but this will yield a solution with unnecessarily large order.

4.1 Parks-McClellan (Remez) Algorithm

The algorithm of Parks-McClellan [7] adapted the second algorithm of Remez [8, pg. 95] to find a polynomial whose weighted error is minimized. Although Parks-McClellan has been traditionally used in designing filters with piecewise constant or piecewise linear response mixed with don't-care regions, there is nothing intrinsic about the underlying theory that prevents one from designing a polynomial to fit an arbitrary curve. The set of classes of functions to which the Remez exchange algorithm can apply is very broad. Precise necessary conditions on the desired function $H_d(x)$ can be found in [8].

There are only three inputs to the Parks-McClellan algorithm: a filter order, a desired function $H_d(x)$ and a weighting function $W(x)$. The Parks-McClellan algorithm returns the Mth-order polynomial $A(x)$ which minimizes the maximum weighted approximation error. That is, it determines a set of coefficients $\{p_0, p_1, \ldots, p_M\}$ such that for

$$A(x) = \sum_{i=}^{M} p_i x^i , \tag{4.1}$$

and for a closed subset F_p consisting of disjoint union of closed subsets of the real axis x, the maximum weighted error

$$\max_{x \in F_p} \left| W(x)\big(H_d(x) - A(x)\big) \right| \tag{4.2}$$

is minimized [3, pg. 468]. Note that the weighting function $W(x)$ does not give us direct control over the absolute sizes of the approximation errors. Rather, $W(x)$ controls the ratio of the ripple sizes. Thus, for a given set of tolerances, it is

necessary to apply the Parks-McClellan iteratively with various filter orders until the specifications are met.

In the audio filter design problem, we are given upper and lower bounds $U(x)$ and $L(x)$ (where $U(x) > L(x)$), but no desired response $H_d(x)$. In order to adapt the audio filter design problem to use the Parks-McClellan algorithm, we propose using

$$H_d(x) = \tfrac{1}{2}\big(U(x) + L(x)\big) \tag{4.3}$$

and

$$W(x) = \frac{1}{\tfrac{1}{2}\big(U(x) - L(x)\big)}. \tag{4.4}$$

If we apply the Parks-McClellan algorithm on this set of inputs and increment the filter order until the weighted error is less than 1, then we have found a solution to the constrained ripple problem as well. More precisely, if $A(x)$ has a weighted error less than 1 then,

$$\begin{aligned} 1 &> \Big|W(x)\big(H_d(x) - A(x)\big)\Big|, \quad \forall x \in F_p \\ &= \left|\frac{\tfrac{1}{2}\big(U(x) + L(x)\big) - A(x)}{\tfrac{1}{2}\big(U(x) - L(x)\big)}\right|. \end{aligned} \tag{4.5}$$

which makes

$$\tfrac{1}{2}\big(U(x) - L(x)\big) \;>\; \tfrac{1}{2}\big(U(x) + L(x)\big) - A(x) \;>\; -\tfrac{1}{2}\big(U(x) - L(x)\big). \tag{4.6}$$

This reduces to

$$\begin{aligned} A(x) &> L(x) \\ \text{and } A(x) &< U(x) \end{aligned} \tag{4.7}$$

as desired.

An intuitive argument is as follows. $H_d(x)$ is the middle curve between the two bounds, and $W(x)$ the inverse of one-half of the gap width. So at any point x, if the error is larger than one-half the width of the gap, then it must have exceeded the bound. The conjecture in setting $H_d(x)$ to be the midline allows the resulting polynomial to have as large a vertical "swing" as we can afford, which, in turn, should keep the polynomial order low.

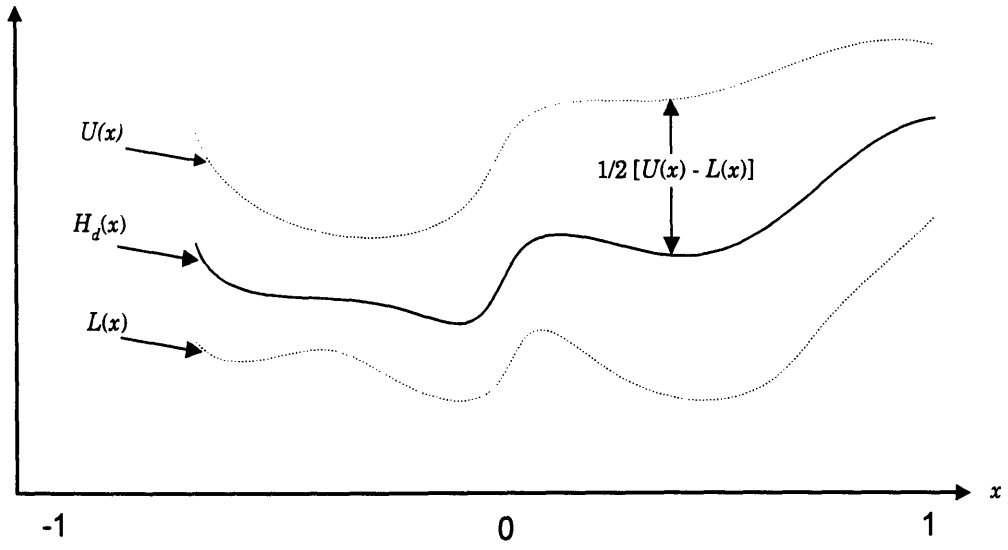


Figure 4-1 ***Polynomials of upper and lower bound U(x) and L(x) on the interval [-1,1]. The middle curve $H_d(x)$ is the average magnitude of U(x) and L(x).***

4.1.1 Automatic Order Increase

Another advantage of using the constant 1 as the threshold for rejecting or accepting a design is the ability to detect early on if a given polynomial order is too small. This capability allows the filter designer to increase the order without carrying the algorithm to convergence. The detection can be done as follows. Suppose the Remez algorithm is currently searching for an optimal polynomial at

order L. In each iteration, we must compute an approximating polynomial based on a certain set of interpolation points $\{ x_i, y_i \}$ for $0 \le i \le L$. The algorithm states that the y-coordinate of this set of points must be chosen to be as close to $H_d(x)$ as possible. Specifically, the weighted error between $H_d(x)$ and the interpolating polynomial when evaluated at $x_o, x_1, x_2, \ldots, x_L$ must be minimized. Note that this does not mean that the weighted error evaluated at other points would be minimized as well. By the alternation theorem, this produces a unique polynomial whose weighted error at the points x_i are of equal magnitude and alternating sign:

$$\delta, -\delta, \delta, -\delta, \ldots, (-1)^{L+1}\delta.$$

In the proof of convergence of this algorithm, Cheney [8, pg. 98] has shown that the absolute value of the weighted error, $|\delta|$, forms a bounded, monotonically increasing sequence with each iteration. This means that we can increment the polynomial order L as soon as $|\delta| > 1$, instead of waiting until $|\delta|$ converges.

Unfortunately, even if we try every order n starting from $n=2$ as the polynomial order, until the weighted error is less than unity, we still cannot be certain if the filter order we have arrived is the minimum taken over all polynomials which are the solutions to the constrained ripple problem; it is only the minimum order taken from the set of solutions from Remez algorithm when given the above $H_d(x)$ and $W(x)$ as inputs.

4.2 CONRIP algorithm

Although the method above describes a way in which we can arrive at a solution, it does not guarantee that the solution has minimal order. In his thesis [6], McCallig

proved that a filter resulting from his algorithm will always have the minimal length among the valid filters meeting the upper and lower constraints. Also, he showed that if his algorithm fails to find such a polynomial, then no polynomial exists which meets the constraints.

To date, there have been very few references to his thesis. Hence, we would like to reiterate some of the highlights of the theories that he developed. Although some theorems have been proven for generalized polynomials (or *Chebyshev Systems*), our presentation here is in terms of ordinary polynomials 1, x, x^2,..., x^n since they are of main interests to us.

Definition (P_n): Given continuous functions $U(x)$ and $L(x)$ on [–1,1] such that $U(x) > L(x)$ for each x, let P_n be the set of all polynomials $p(x)$ of order n,

$$p(x) = \sum_{i=0}^{n} p_i x^i \tag{4.8}$$

such that $L(x) \le p(x) \le U(x)$ for $-1 \le x \le 1$. By the Weierstrass Approximation Theorem [8, pg. 66], P_n is non-empty for sufficiently large n.

Before introducing the next definition, it is worthwhile to point out that in order to specify a polynomial of degree n uniquely, we need exactly $n+1$ coefficients $p_0, p_1,\ldots,p_n$. Alternatively, we can also specify it with $n+1$ interpolation points $(x_0,y_0), (x_1,y_1), \cdots, (x_n,y_n)$ where $x_i \neq x_j$ for $i \neq j$.

Definition (P_u): Given $U(x)$ and $L(x)$, let P_u be the set of nth-order polynomials which can be specified by interpolation points alternately on $U(x)$ and $L(x)$, the first

point being on $U(x)$. The abscissas of the interpolation points are arranged in increasing order, i.e., $x_0 < x_1 < \cdots < x_n$.

Definition (P_l): The set P_l has the same definition as P_u except that the first interpolation point is on $L(x)$.

Note that a polynomial could be in P_u and P_l. For example, consider a polynomial $g(x)$ from P_u. At x_0, $g(x_0)=U(x_0)$ by definition. However, in the region where $x < x_0$, we cannot predict how $g(x)$ behaves. It might cross $L(x)$ at, say, x_{-1}. This means we can also specify $g(x)$ by interpolating through

$$\left(x_{-1}, L(x_{-1})\right),\ \left(x_0, y_0\right),\ \left(x_1, y_1\right),\ \cdots,\ \left(x_{n-1}, y_{n-1}\right).$$

Therefore, $g(x) \in P_u \cap P_l$.

Theorem 4.1 (*Existence of p_+ and p_-*): If n is large enough so that P_n is not empty, then

I. $P_n \cap P_u$ contains a single polynomial p_+

II. $P_n \cap P_l$ contains a single polynomial p_-

Proof: The proof may be found in [9, pg. 72].

Theorem 4.1 is fundamentally important to CONRIP. It states that if, for a fixed order, there exists any polynomial lying within the bounds at all, then there must also exist two polynomials which touches the upper and lower boundaries

alternately. This is the reason that CONRIP directs all its effort only into searching those two particular polynomials: p_+ and p_-.

4.2.1 Conditions on the Minimal Order of the Solutions

McCallig also proved the following two theorems which are very useful in determining if a given order n is the minimal order.

Theorem 4.2 (Conditions on n being too small): P_n is empty if and only if $P_u \cap P_l$ is not empty.

Theorem 4.3 (Conditions on n being too large): The coefficients of the term x^n of p_+ and p_- are of opposite signs if and only if the polynomial order n is not minimal.

Proof: The proof can be found in [6].

4.2.2 Description of the Algorithm

As an important part of his thesis, McCallig presented a very efficient algorithm that finds p_+ and p_-. The CONRIP algorithm is strikingly similar to the Remez exchange algorithm. With each passing iteration, the state of the algorithm takes the form of $n+1$ interpolation points (x_i, y_i). During each iteration, the interpolation points would produce a unique polynomial, from which we obtain the new set of interpolation points. If n is sufficiently large so that P_n is non-empty, then the algorithm converges to p_+, and, with some modification in the algorithm, p_-. If n is too small, the algorithm would eventually produce a polynomial which is in $P_u \cap P_l$. Therefore, we need not carry the algorithm to convergence in order to realize that

the order is too small, because we can monitor if the current polynomial has the characteristics of both P_u and P_l, and if so, we must increase the order n.

4.2.3 Solution Polynomials: p_+ and p_-

In his thesis, McCallig also showed that except in some very special cases, these two polynomials are different. By definition, these two polynomials are different in the manner in which they touch the boundaries. The polynomial p_+ touches the upper bound before the lower bound, while for p_-, the opposite is true. Experience has shown that these two curves often "move" in opposite directions. That is, when p_+ is near $U(x)$, p_- would be near $L(x)$ and vice versa. Moreover, because p_+ and p_- are both solutions to the constrained ripple problem, any weighted sum

$$\alpha p_+ + (1-\alpha)p_- \quad \text{for } 0 \le \alpha \le 1 \tag{4.9}$$

would be a valid solution as well. This suggests that we can use a combination of p_+ and p_- to reduce the size of the ripples in the final solution.

4.3 Comparisons on Practical Issues

From an implementation point of view, the Parks-McClellan and CONRIP algorithms are nearly equal. Both algorithms involve evaluating a polynomial on a discrete x-axis, and finding the maxima and minima of errors.

The characterization of the polynomial of the minimal order n is only available in CONRIP but not in Remez. By merely observing the polynomial coefficients, it allows us to detect if the current filter order n is too large and may be reduced. However, the ability to detect early on that n is too small can be done in both algorithms. This option is especially valuable when we are searching for the

minimal order n starting from a small value. It enables us to stop iterating before the polynomial has converged and move on to a higher order.

4.4 Precision Requirements in Filter Design

Both CONRIP and the Remez exchange algorithm produce intermediate polynomial coefficients. We have found that for high order designs ($n \geq 30$) the precision needed to represent these coefficients exceeded that of standard double precision. In order to solve this problem, we created a custom arithmetic system in C with variable precision. With each iteration, the precision is adjusted dynamically according to the numerical error in the interpolated data. Fortunately, after the transformation from Chebyshev polynomials to Type I FIR filters, the coefficients of the final FIR filters usually have a dynamic range of about two to three orders of magnitude, which is conducive to actual finite-precision implementation.

Chapter 5

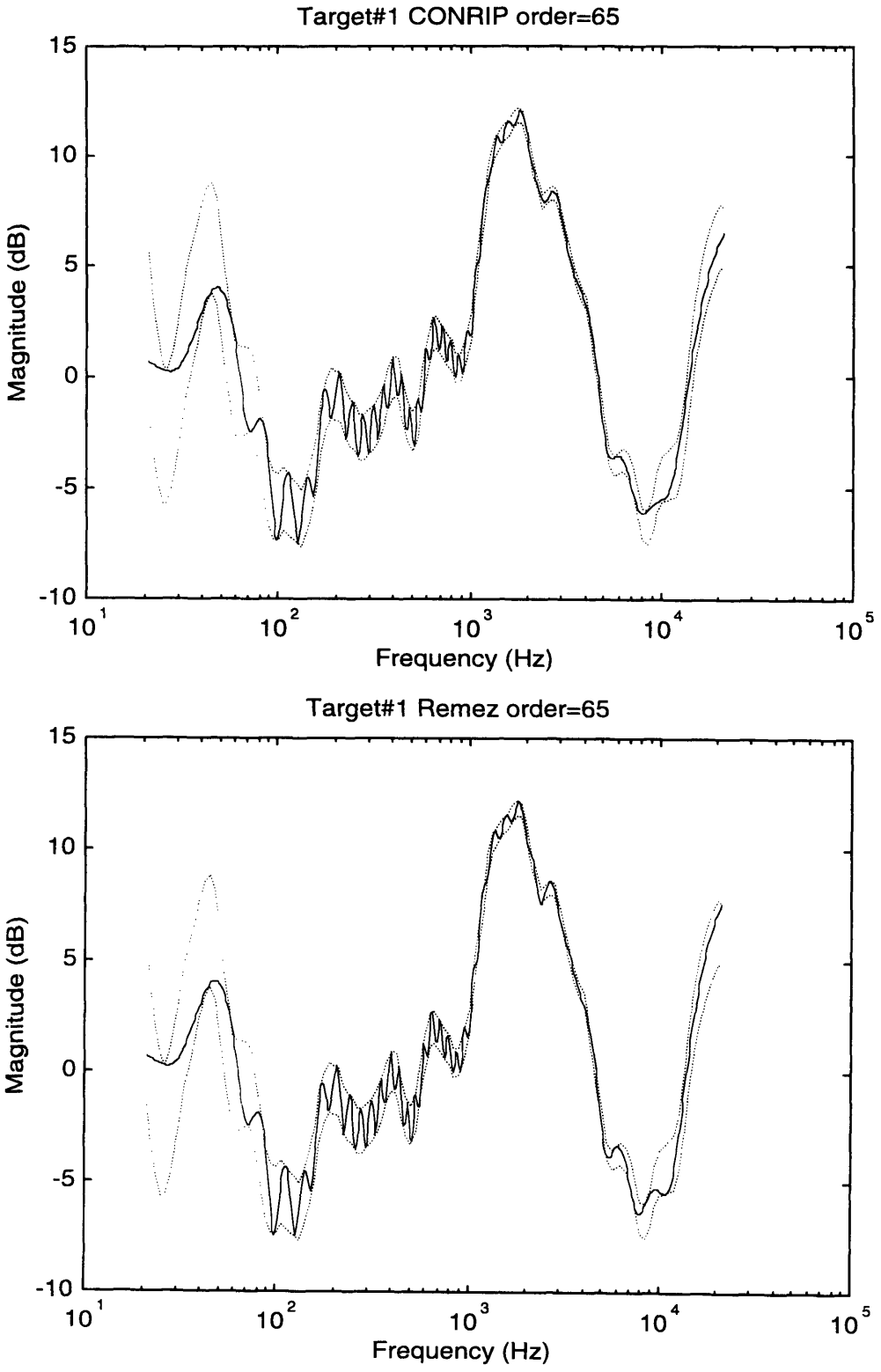
Results From Filter Design Experiments

We tested an overall design procedure using the target frequency response curves that were described in section 2.3. We examined 20 designs and compared the resulting filter orders that result from using Remez and CONRIP. We also discuss the experimentally obtained optimal warping parameter and compare it with the theoretical results.

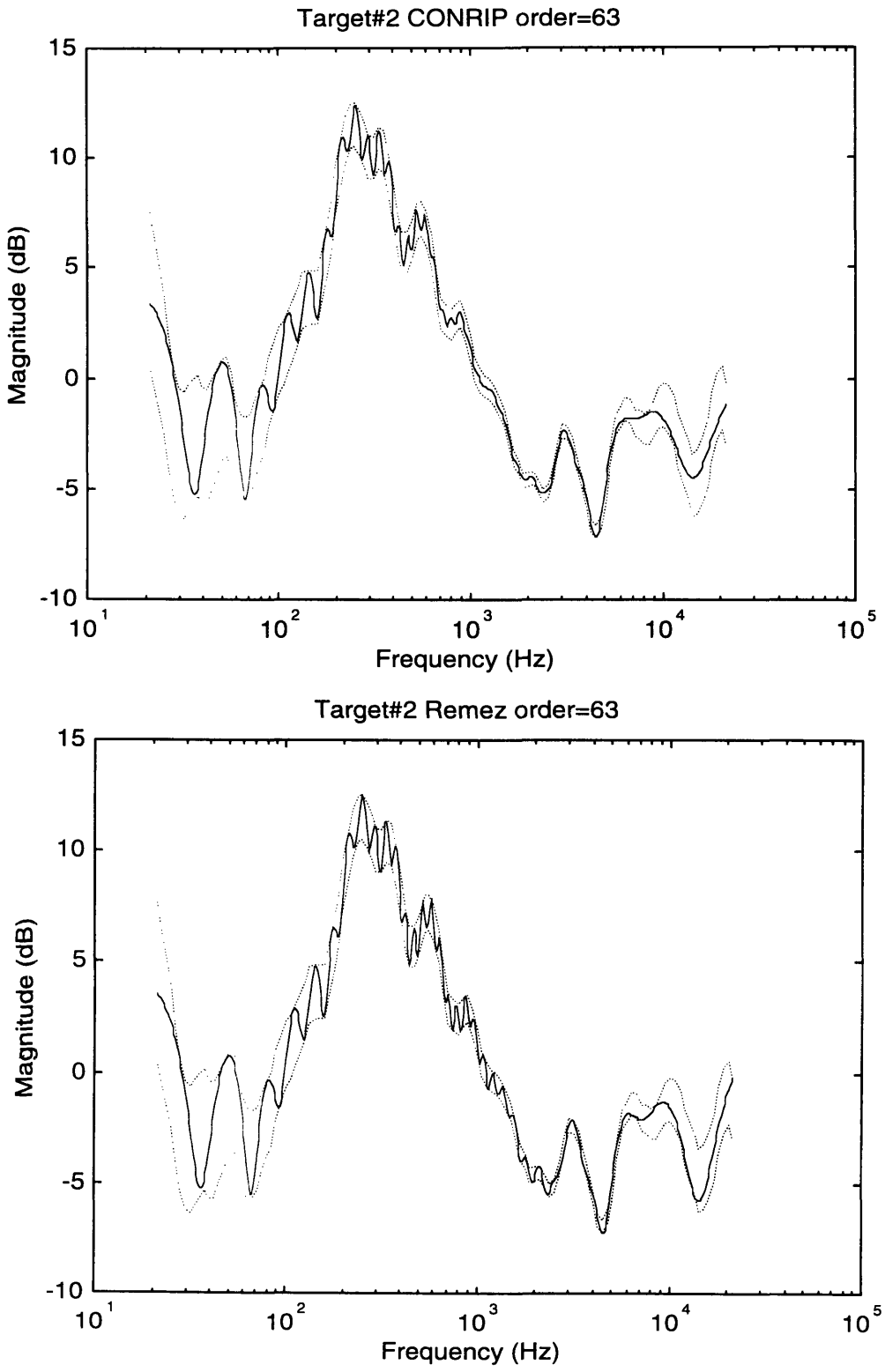
5.1 Filter Orders and Comparison with Remez Algorithm

Recall that in obtaining a truly minimal filter order for a given target frequency response, there are two successive minimization processes. First, we must warp the target response with an optimal warping factor. Second, we must design an FIR filter of the minimal order to fit the warped response. In the first minimization, we first find the solution in (3.43) by a computer search to obtain an approximate warping factor. Then we search in the (discretized) neighborhood of that estimate. For each neighboring warping factor, we use both CONRIP and Remez to design FIR filters with orders as low as the algorithm can find. As described in section 3.3, the FIR filter is then implemented with warping so that the effective response meets the target.

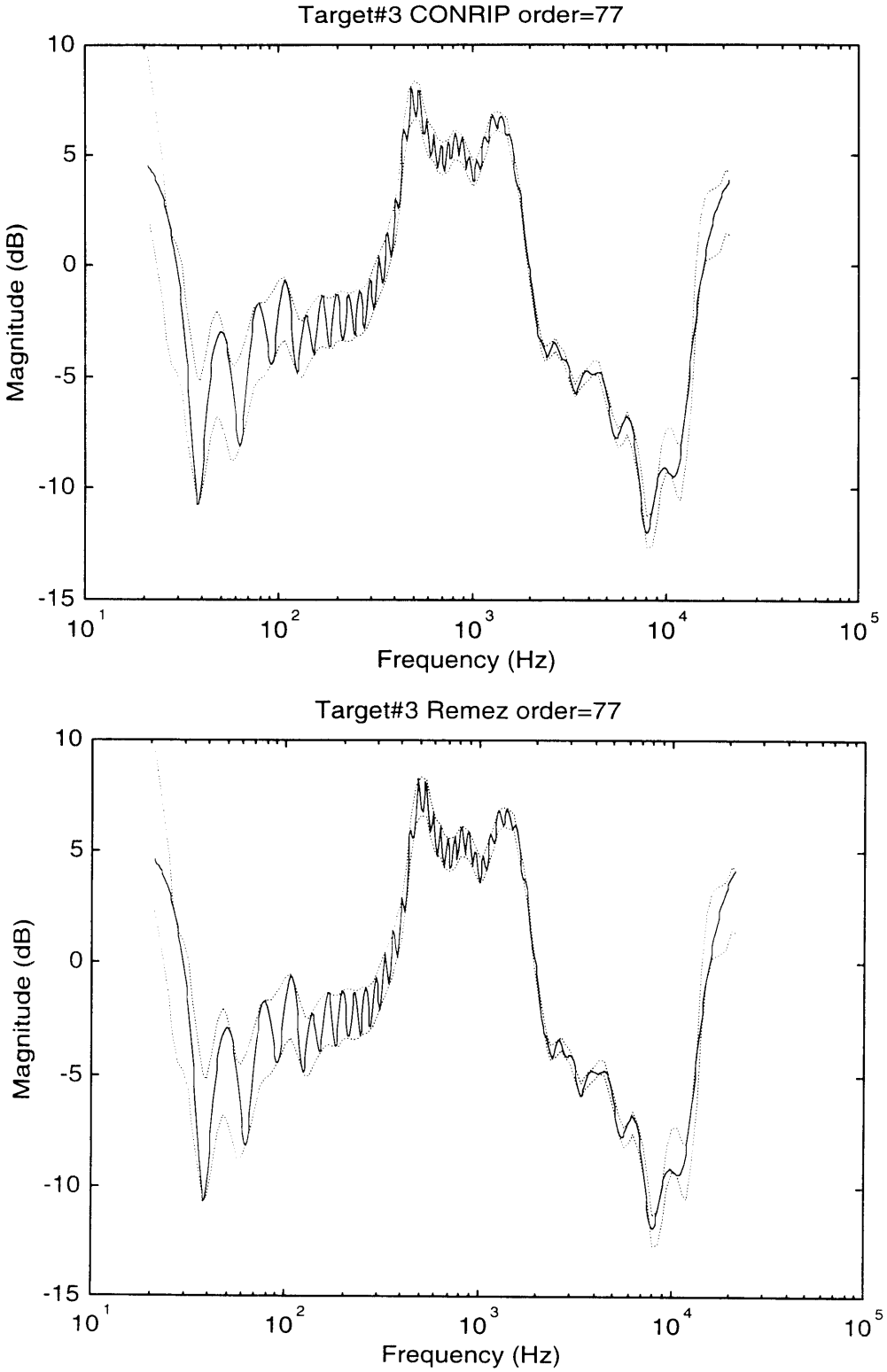
Some of the final design results of both algorithms follow. In the figures below, the effective filter response is shown with the Just-Noticeable Difference bound. All the results from CONRIP are for the average response $(p_+ + p_-)/2$.


Target#1 CONRIP order=65
Magnitude (dB)
Frequency (Hz)
Target#1 Remez order=65
Magnitude (dB)
Frequency (Hz)

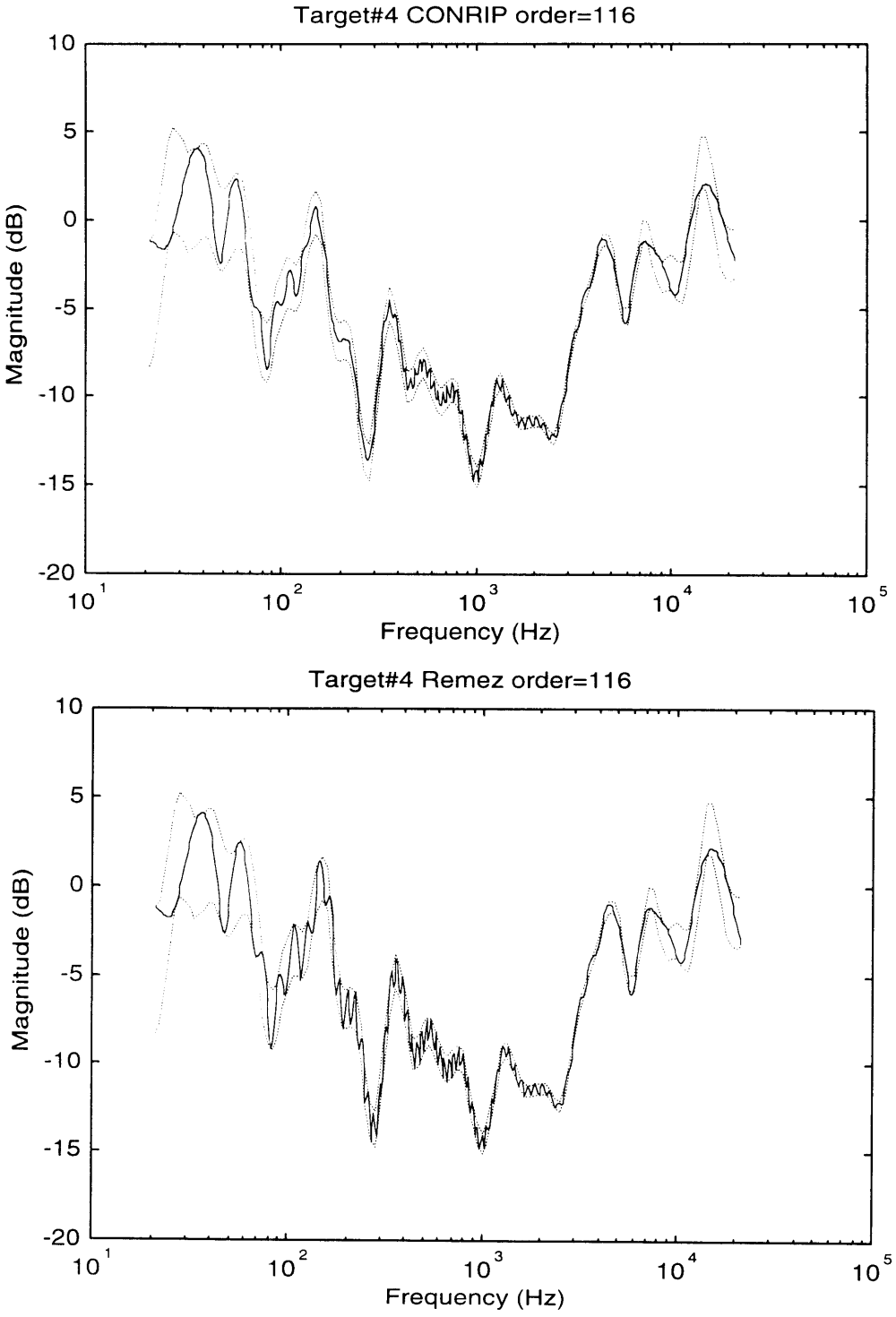
(a)


Target#2 CONRIP order=63
Magnitude (dB)
Frequency (Hz)
Target#2 Remez order=63
Magnitude (dB)
Frequency (Hz)

(b)


Target#3 CONRIP order=77
Magnitude (dB)
Frequency (Hz)
Target#3 Remez order=77
Magnitude (dB)
Frequency (Hz)

(c)



(d)

Figure 5-1 (a)-(d) Sample design results.

The following table summarizes the minimal filter orders from each algorithm.

Target #	CONRIP order	Remez order	Remez-CONRIP
1	65	65	0
2	63	63	0
3	77	77	0
4	116	116	0
5	60	60	0
6	63	62	–1
7	63	63	0
8	72	72	0
9	60	61	1
10	71	71	0
11	91	91	0
12	67	67	0
13	98	98	0
14	49	50	1
15	62	62	0
16	44	44	0
17	64	64	0
18	64	64	0
19	> 130	> 130	N/A
20	> 130	> 130	N/A

Table 5-1 *Minimal orders of FIR filters.*

Several observations can be made from the above table. First, the minimal order from the two algorithms are nearly always equal. There are three filters in which the minimal filter orders of the two algorithms are different. It is surprising that Remez even outperforms CONRIP in target#6, because the theory behind CONRIP guarantees the minimal order of its solution. We believe that this phenomenon can be explained by the way the design algorithms are coded on the computer. Specifically, we believe that the minimal order for target#6 was on the borderline between order 62 and 63. Because both programs detect the convergence based on

some reasonably small, yet arbitrary constant, CONRIP might have decided prematurely that the algorithm has converged but does not meet the boundary conditions; if it had continued a few more iterations at order 62, it might have succeeded without having to increase the order to 63.

The other thing we noticed from Table 5-1 is that the last two target designs did not reach a result, although attempts were made to design filters of order up to 130. To design a filter of such a high order requires very high precision arithmetic. At the time we aborted these design processes, the computer was using roughly 33 bytes to store the mantissa of each variable. This is equivalent to about 80 decimal digits, which is five times as accurate as standard double precision. The polynomial interpolation procedure required too much time and was aborted. This suggests that without warping, these algorithms would have required such a long mantissa representation that a computer might either run out of memory or take an unreasonably long time to design.

5.2 Experimentally Obtained Optimal Warping Factors

Based on the numerical solution to (3.43), the optimal warping factor is estimated to be $\hat{a}=0.933$. In the design experiments, we search for the optimal warping factor in the neighborhood of that estimate. By discretizing the values of the warping factor a to multiples of 0.01, we find the minimal filter order (according to each algorithm) for each value of a until we find a local minimum. As described in section 3.4, we conjecture that there is only one minimum for all a, which implies that the local minimum found is also the global minimum. The experimentally obtained optimal warping factor according to each algorithm is shown in the following table.

Target #	CONRIP order	Remez order
1	0.92	0.92
2	0.92	0.92
3	0.91	0.91
4	0.92	0.92
5	0.92	0.92
6	0.91	0.91
7	0.91	0.91
8	0.92	0.92
9	0.92	0.92
10	0.92	0.92
11	0.92	0.92
12	0.91	0.91
13	0.91	0.91
14	0.92	0.92
15	0.92	0.92
16	0.91	0.91
17	0.92	0.92
18	0.91	0.91
19	N/A	N/A
20	N/A	N/A

Table 5-2 *Experimentally obtained warping factors.*

In Table 5-2, we notice that Remez and CONRIP always achieve minimal order at the same warping factor. This shows even further that the two algorithms are roughly equivalent in terms of performance for filter orders within our interests. Moreover, the experimentally obtained warping factor is always below the predicted $\hat{a}$=0.933, which is greater than the observed optimal warping factor in Table 5-2. Recall from Figure 3-2 that a warping factor with higher values corresponds to stretching out the low frequencies even further. This is reasonable when one takes into account the form of the JND curves in Figure 2-3. Due to human perception's better resolution in low frequencies, JND curves have bigger values in the low-frequency range, which means bigger errors are allowed. Since the problem statement from which $\hat{a}$ is derived assumes no partiality in the low-frequency

range, it predicts a value that would "help" the low-frequency range more than it really needs to.

Chapter 6

Implementation Issues

Thus far, the frequency warping technique seems like an attractive method of equalizing signals due to the considerable reduction in the filter orders when compared to non-warped methods. In this chapter, we address other issues concerning the actual implementation of this technique. We analyze the signal flow graph of typical implementations and calculate the error due to finite numerical effects. We also summarize the number of arithmetic operations and number of registers required per output sample, and compare them to some other filtering methods.

6.1 Quantization Noise Analysis in Fixed-Point Arithmetic

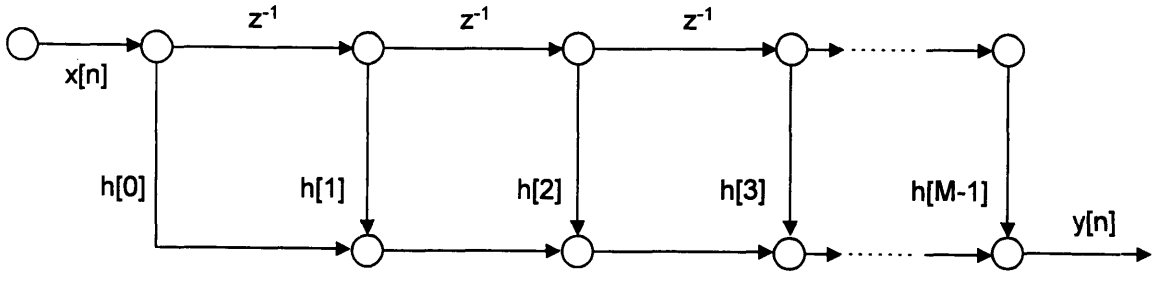
Several questions naturally arise concerning the behavior of warped filter in the presence of coefficient quantization and roundoff noise. For a given transfer function $H(z)$, there are several different implementations (e.g., direct form, cascade, parallel, etc.) which are theoretically equivalent but behave differently in the presence of numerical errors. We will focus on system structures derived by direct replacement of delays by first order allpasses. We will analyze structures based on direct form FIR and transposed direct form FIR filters.

Direct replacement is used without further simplification for a number of reasons:

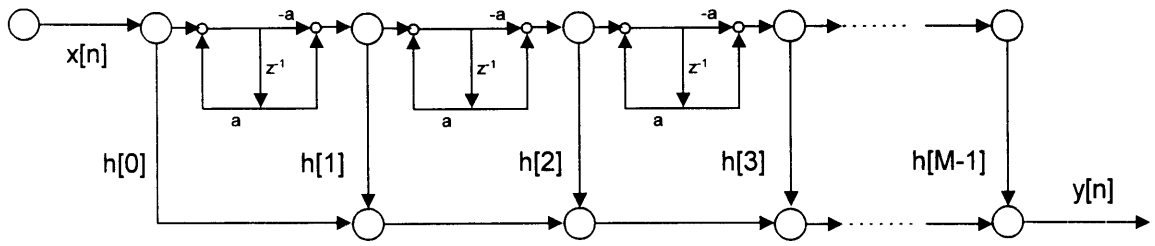
1. It is well-known that structures based on a cascade or sum of low-order sections are more robust than high-order structures [3, pg. 337].
2. First order allpasses remain allpass regardless of the quantization of a, as long as the coefficients in the numerator and the denominator are quantized the same way. Thus, we will always realize a true frequency warping. The quantization of a is of no real concern because we can limit a to realizable values during the filter design procedures.
3. There is no penalty in memory over higher order implementations. As for computation, the number of multiplications is 1.5 times more than that required by high-order implementations.

6.1.1 Warped Direct-Form FIR Implementation

This section analyzes implementations based on direct form FIR systems. These are the more apparent choice of implementation and we describe them first. A direct form FIR is shown in Figure 6-1 (a) and the corresponding warped version in Figure 6-1 (b).



(a)



(b)

Figure 6-1 (a) Direct Form FIR Implementations. (b) Warped, Direct Form FIR Implementations.

Since every multiplication in the above flowgraph is performed with finite-precision fixed-point arithmetic, there is some error introduced after each multiplication. We model that error as a noise source injected into the signal, as shown in Figure 6-2.

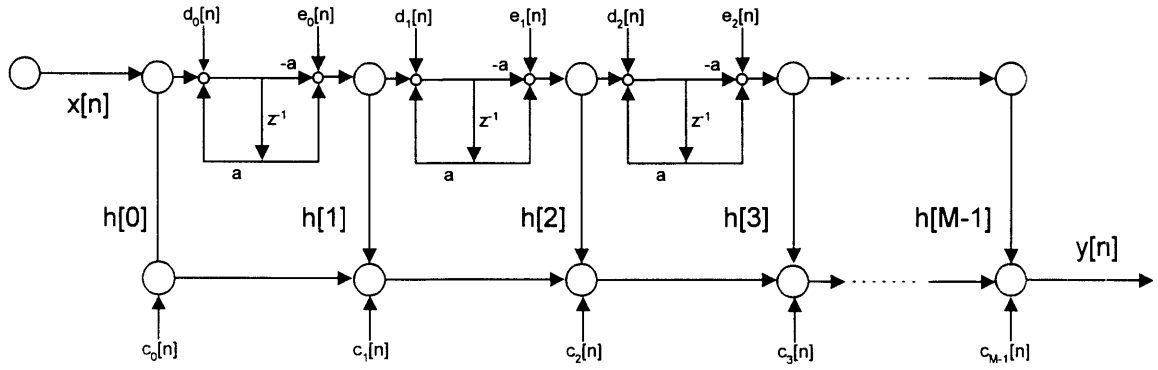


Figure 6-2 Warped, Direct Form FIR implementation with noise source injected after each multiplication.

We will make the same assumptions about the each quantization noise source as do Oppenheim and Schafer [3, pg. 353]. Those assumptions are:

1. Each quantization noise source $c_i[n]$, $d_i[n]$, and $e_i[n]$ is a wide-sense stationary white-noise process.

2. Each noise source has a uniform distribution of amplitude over one quantization interval.
3. Each quantization noise source is uncorrelated with the input to the corresponding quantizer, all other quantization noise sources, and the input to the system.

Assuming that we use B-bit arithmetic in all of the above operations, each quantization noise source would be uniformly distributed in the range $\pm \frac{1}{2} 2^{-B}$. Thus the variance of each noise source is

$$\sigma_{c_i}^2 = \sigma_{d_i}^2 = \sigma_{e_i}^2 = \frac{2^{-2B}}{12}. \tag{6.1}$$

To calculate the output of this system due to the noise sources, we first notice that we can combine all the sources $c_i[n]$ to a single source $C[n]$ which is injected right into the output $y[n]$ with a variance M times as high as that of each individual $c_i[n]$. Namely,

$$\sigma_C^2 = M\sigma_{c_i}^2 = M\frac{2^{-2B}}{12}. \tag{6.2}$$

Next, in each of the allpass sections, the noise $d_i[n]$ is filtered by an allpass filter before combining with $e_i[n]$ of the same section. However, since allpass filters have a unity gain and leave the variance of the filtered noise $d_i[n]$ unchanged, we can combine $e_i[n]$ and the allpass-filtered version of $d_i[n]$ into a noise source $D_i[n]$ whose variance is

$$\begin{aligned} \sigma_{D_i}^2 &= \sigma_{e_i}^2 + \sigma_{d_i}^2 \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|H_{ap}(e^{j\omega})\right|^2 d\omega \\ &= \sigma_{e_i}^2 + \sigma_{d_i}^2 \\ &= 2\frac{2^{-2B}}{12}. \end{aligned} \tag{6.3}$$

With the new, lumped noise sources $C[n]$ and $D_i[n]$, the system now looks as shown in Figure 6-3.

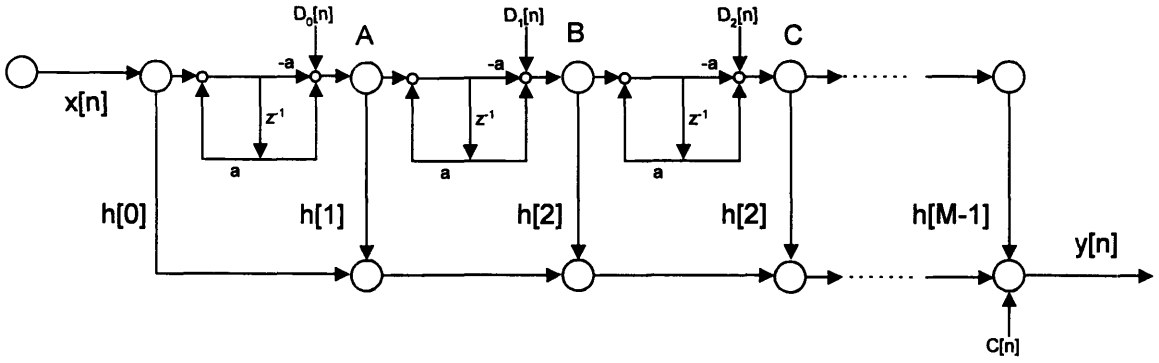


Figure 6-3 Warped, Direct Form FIR implementation with the noise sources $c_i[n]$ combined into $C[n]$, and $d_i[n]$ and $e_i[n]$ into $D_i[n]$.

Let us now analyze the path from each noise source $D_i[n]$ to the output $y[n]$ as well as any amplification occurring along it. At each junction A, B, C,... in Figure 6-3, the signal $D_i[n]$ may branch out two ways; it may continue to flow horizontally to the right through a series of allpasses, or it may travel down a vertical branch where it is multiplied by some filter coefficient $h[n]$ before being accumulated to the output. Since allpass filters have a unity gain, the variance of $D_i[n]$ is unchanged until it is multiplied by $h[n]$. The variance of the output due to each noise source $D_i[n]$ is

$$\begin{aligned} \text{Output variance due to } D_i[n] &= \sum_{n=i+1}^{M-1} \left| h[n] \right|^2 \sigma_{D_i}^2 \\ &\le (M-i-1)\sigma_{D_i}^2 h_{\max}^2 \end{aligned} \tag{6.4}$$

where

$$h_{\max} = \max_{0 \le n \le M-1} \left| h[n] \right|.$$

The variance of the total noise due to numerical roundoff is therefore

$$\text{Output var. from } C[n] + \sum_{i=0}^{M-2} \text{Output var. from } D_i[n] \le M\frac{2^{-2B}}{12} + \sum_{i=0}^{M-2}(M-i-1)\sigma_{D_i}^2 h_{\max}$$

$$= \frac{2^{-2B}}{12}M + h_{\max}^2\frac{M^2}{2}2\cdot\frac{2^{-2B}}{12}) \qquad (6.5)$$

$$= \frac{2^{-2B}}{12}(M + h_{\max}^2 M^2).$$

To measure the effect of the roundoff noise in number of bits, we calculate the ratio between the noise standard deviation and the amplitude of the least significant bit, and then take the log of that quantity. Assuming that all operations are done in B-bit arithmetic, the number of output bits that are "roughly" corrupted by the roundoff noise in the direct form FIR implementation is

$$N_D = \log_2\left[\frac{\text{standard deviation of total roundoff noise}}{\text{magnitude of least significant bit}}\right]$$

$$\le \log_2 \frac{\left(\frac{2^{-2B}}{12}(M + h_{\max}^2 M^2)\right)^{1/2}}{2^{-B}} \qquad (6.6)$$

$$= \frac{1}{2}\log_2\left(M + h_{\max}^2 M^2\right) - \log_2 12$$

$$\approx \frac{1}{2}\log_2\left(M + h_{\max}^2 M^2\right) - 1.8 \text{ bits.}$$

Suppose there were *no* warping, then the output noise would be due to $C[n]$ alone. The amplitude of this noise when measured in bits would be

$$N_0 = \log_2\left[\frac{\text{standard deviation of } C[n]}{\text{magnitude of least significant bit}}\right]$$

$$= \log_2 \frac{\left(M2^{-2B}/12\right)^{1/2}}{2^{-B}} \qquad (6.7)$$

$$\approx \frac{1}{2}\log_2 M - 1.8 \text{ bits.}$$

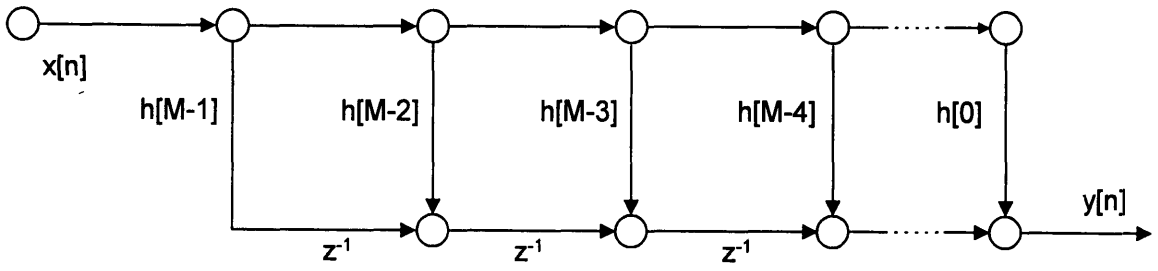
Therefore, by warping, we have increased the number of corrupted bits by

$$N_D - N_0 = \left(\frac{1}{2}\log_2\left(M + h_{\max}^2 M^2\right) - 1.8\right) - \left(\frac{1}{2}\log_2 M - 1.8\right)$$
$$= \frac{1}{2}\log_2\left(1 + h_{\max}^2 M\right) \text{ bits.} \tag{6.8}$$

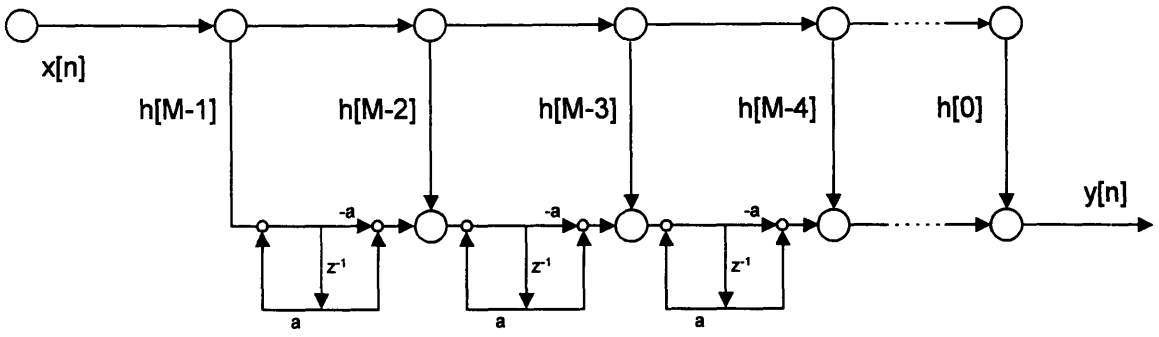
As an example, a typical value of M for a warped filter designed to equalize the audible range spectrum is about 2^7. If we let $h_{\max}$ be 1, then the roundoff error of the warped implementation will cost us an extra $N_D - N_0 \approx \frac{1}{2} \log_2 2^7 = 3.5$ bits in addition to the inevitable $N_0 = \frac{1}{2} \log_2 M - 1.8 = 1.7$ bits. This can be a serious penalty especially when considering that most systems today handle audio signals at no more than 16 bits. Moreover, the number of additional corrupted bits due to warping, $N_D - N_0$, grows as a function of M. This shortcoming makes warping from a direct form FIR structure a rather unattractive implementation, despite all its simplicity.

6.1.2 Warped, Transposed, Direct-Form FIR Implementation

We now analyze the transposed direct form implementation. Surprisingly, the effect of numerical noise is not as severe when we start with a transposed structure. Consider a transposed system and its corresponding warped version below.



(a)



(b)

Figure 6-4 (a) A Transposed Direct Form implementation of an FIR filter. (b) A warped version of the transposed form.

With the same assumptions as in the previous section, we introduce roundoff error noise after each multiplication, as shown in Figure 6-6.

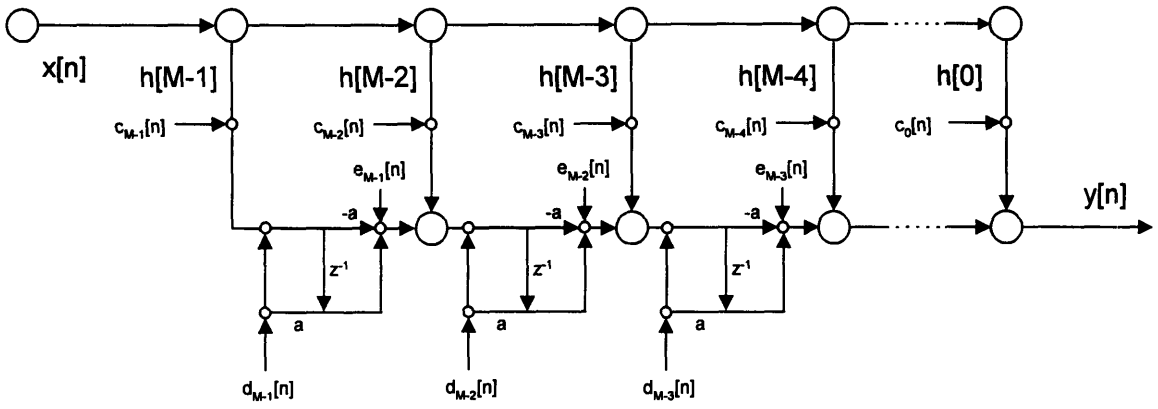


Figure 6-5 Warped, Transposed Direct Form FIR with noise sources.

From the above diagram, we observe that we can combine noise sources which are injected to the same node into a new noise source. Namely, we can define

$$\begin{aligned} r_{M-1}[n] &= c_{M-1}[n] + d_{M-1}[n] \\ r_i[n] &= c_i[n] + d_i[n] + e_{i+1}[n] \quad \text{for } 1 \le i \le M-2 \\ \text{and} \quad r_0[n] &= c_0[n] + e_1[n]. \end{aligned} \tag{6.9}$$

so that the system looks as follows.

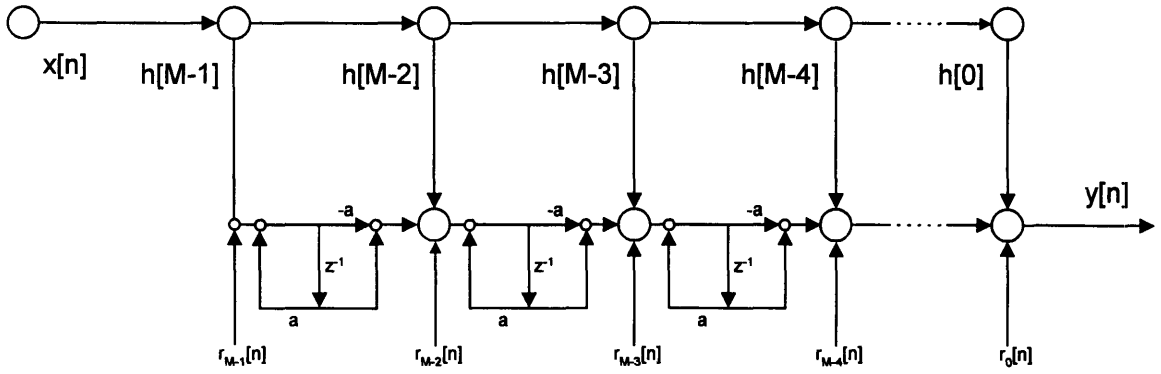


Figure 6-6 Warped Direct Form FIR with noise source combined

The variance of the noise $r_i[n]$ is now

$$\sigma_r^2 \le \sigma_c^3 + \sigma_d^3 + \sigma_e^3 = 3\frac{2^{-2B}}{12} = 2^{-2B-2}. \tag{6.10}$$

Note that the equality holds except at endpoints h[M-1], and h[0].

Let $R[n]$ be the output due to $r_i[n]$ summed over all i. Since an allpass filter has unity amplitude gain, the variance of $R[n]$ can be expressed simply as

$$\sigma_R^2 \le M\sigma_r^2 = M2^{-2B-2}. \tag{6.11}$$

Finally, to measure the effect of the noise in terms of bits, we calculate the log of the ratio between the noise standard deviation and the magnitude of the least significant bit to be

$$\begin{aligned} N_T &= \log_2\left[\frac{\sigma_R}{\text{magnitude of LSB}}\right] \\ &= \log_2 \frac{\left(M \cdot 2^{-2B-2}\right)^{1/2}}{2^{-B}} \\ &= \frac{1}{2}\log_2 M - 1. \end{aligned} \tag{6.12}$$

Without warping, the output due to roundoff would be due to all the sources $c_i[n]$ alone. Therefore, the number of corrupted bits of a transposed, unwarped, direct form is equal to that obtained from an untransposed, unwarped implementation, or N_0 in (6.7). The number of extra corrupted bits that warping incurs is then

$$\begin{aligned} N_T - N_0 &= \left(\frac{1}{2}\log_2 M - 1\right) - \left(\frac{1}{2}\log_2 M - 1.8\right) \\ &= 0.8 \text{ bits} \end{aligned} \tag{6.13}$$

which is independent of M.

As an example, for a typical value of M as 2^7, the number of corrupted bits of a warped, transposed structure is N_T = ½ $\log_2 2^7 - 1 = 2.5$ bits, whereas that of an unwarped implementation is N_0 = ½ $\log_2 2^7$ - 1.8 = 1.7 bits. This shows that warping from a transposed structure gains us the numerical accuracy without any additional computation cost, and is a more desirable method of implementation than warping from an untransposed form.

6.2 Memory and Computational Requirements

Given a warped implementation of a length M FIR filter as above, the following table summarizes the number of operations and storage required per output sample when implemented using the warped, transposed direct form FIR implementation. It also compares these requirements to those required by unwarped transposed direct form FIR, as well as by N-point overlap-save Fast Fourier Transform (FFT). In creating this table, we have assumed that one complex multiplication requires four real multiplications and two real additions. Also, we assume that the FFT coefficients of the FIR filter is computed off-line.

Number of operations	Warped, transposed direct form FIR (Typically $M \sim 70$)	Unwarped, direct form FIR (Typically $M \sim 4{,}000$)	Overlap-Save N-point Fast Fourier Transform (FFT) (Typically $M \sim 4{,}000$)
Real Multiplication	$3(M-1)+1 = 3M-2$	M	$\frac{4N\log_2 N + 2N}{N-M+1}$
Real Addition	$3(M-1) = 3M-3$	$M-1$	$\frac{6N\log_2 N + N}{N-M+1}$
Registers	$M-1$	$M-1$	$7N$

Table 6-1 Computation requirements.

Let us substitute in typical values in the table above to obtain a rough comparison among the three techniques. Without warping, the order of the target FIR filter will be much larger than that designed through warping. From design results in Chapter 5 and our experience, the orders of the FIR filters designed with and without warping are on the order of 70 and 4,000 respectively. The unwarped, direct form FIR is clearly the most inefficient method of convolution and can be eliminated from our comparison first. A reasonable value of N, the FFT length, is twice as big as M. Therefore, assuming N to be 8,000, we find that the total number of operations (additions and multiplications) per output sample of overlap-save is about 270, while that of warping is 420. This shows that the FFT-based overlap-save is more computationally efficient than warping, at least when measured in number of arithmetic operations.

Still, warped filtering has a few advantages over overlap-save method from a practical point of view. First, the latency of the warping method is essentially zero, since each input sample affects the output instantaneously. Overlap-save, on the

other hand, requires the buffering of up to N-M+1 samples before the convolution of each block. Typically, this latency would be unacceptable in real-time audio applications. The reason is that the longest latency that our ears can tolerate is approximately 30 ms. This limits N to be no more than (44.1 kHz)·(30 ms) = 1323 for a sampling rate of 44.1 kHz. Therefore, either M must be significantly smaller than N which is 1323, or we would lose the efficiency of FFT. Clearly, such a low order of M is not realizable with dense, unwarped low-frequency details. Another advantage of warping is the low memory requirement. For M=70, the registers required by warping is only about 70, as opposed to $7N$=7·(8,000)=56,000 required by overlap-save.

Chapter 7

Conclusions and Suggestions for Further Research

The goal of this research was to apply the technique of frequency transformations to the design of practical audio equalizers. We have discussed some properties of the allpass transformation, and described how it is incorporated into filter design processes. As experimental results in Chapter 5 show, frequency transformations can reduce the order of the result significantly. We also made use of the JND bounds by constraining our filters within the prescribed error bounds.

The two filter design algorithms that we used, Parks-McClellan (Remez) and CONRIP, have demonstrated nearly equivalent performance in terms of design order. This is somewhat surprising since we anticipated that CONRIP would clearly outperform Remez especially when we consider CONRIP's many rigorous theorems supporting its optimality. A typical audio filter with 1/3-octave resolution would have an order of about 70, which is substantially lower than that of a typical FIR filter without warping. Finally we showed that a fixed-point implementation of warped filters has satisfactorily low quantization error, and is therefore, feasible to implement.

An open question one might pursue is to determine a more efficient implementation of warped filters. In Chapter 6, we showed how warping might be done by replacing delays in a direct form FIR structure with allpasses. This essentially changes the filter type from FIR to IIR, which means we can no longer use the efficient FFT to convolve audio signals. However, the class of IIR filters to which warped filters belong is so special that it suggests an FFT-like implementation. Specifically, although warped filters have poles, all poles occur at a, the warping factor. However, the question of efficient forms of implementation is not as serious when we consider that typical warped filters have orders of less than 100 points, which are usually low enough that most designers might be indifferent about using the FFT.

The problem of finding an optimal warping factor $\hat{a}$ for a given specification also remains. In this thesis, we have shown how to obtain an estimate of $\hat{a}$ which is generally an upper bound of the true $\hat{a}$, as discussed in section 5.3. Having a closed-form solution for $\hat{a}$ or an estimate that takes into account the JND curve would save the time to search in the neighborhood of the estimate.

Finally, we need to conduct listening experiments to test if our assumptions about the JNDs are correct. As elaborated in section 2.4, the test filter from which the JND curves are derived contains only one ripple in frequency, rather than multiple ripples as in our design results. It still remains to be tested if this extension of the JND bounds causes the listener to detect any difference in the audio signals.

Bibliography

[1] E. C. Carterette and M. P. Friedman, *Handbook of Perception*, New York: Academic Press, 1978.

[2] A. V. Oppenheim and D. H. Johnson, "Discrete representation of signals," *Proceedings of the IEEE*, vol. 60, No. 6, pp. 681-691, Jun. 1972.

[3] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, New Jersey: McGraw-Hill, 1989.

[4] A. G. Constantinides, "Spectral transformations for digital filters," *Proc. IEEE*, vol. 117, No. 8, pp. 1585-1590, Aug. 1970.

[5] M. T. McCallig and B. Leon, "Constrained ripple design of FIR digital filters," *IEEE Transactions on Circuits and Systems*, vol. CAS-25, pp.893-902, Nov. 1978.

[6] M. T. McCallig, "Design of nonrecursive digital filters to meet maximum and minimum frequency response constraints," Ph.D. dissertation, Purdue University, West Lafayette, IN.

[7] T. W. Parks, and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Transactions on Circuit Theory*, vol. CT-19, No. 2, Mar. 1972.

[8] E. W. Cheney, *Introduction to Approximation Theory*, New York: McGraw-Hill, 1966, pp. 72-100.

[9] S. Karlin, and W. J. Studden, *Tchebycheff Systems: With Applications in Analysis and Statistics*, New York: Wiley, 1966.

[10] J. S. Lim, and A. V. Oppenheim, Ed., *Advanced Topics in Signal Processing*, New Jersey: Prentice Hall, 1988.