Design of the Ultraspherical Window Function and Its Applications

by

Stuart William Abe Bergen
B.Sc., University of Calgary, 2000

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Supervisor: Dr. A. Antoniou

ABSTRACT

Window functions are used to reduce Gibbs' oscillations resulting from the truncation of a Fourier series and they are employed in a variety of signal processing applications including power spectral estimation, beamforming, and digital filter design. In this dissertation, the application of window functions based on the ultraspherical window is explored.

First, two methods for evaluating the coefficients of the ultraspherical window are presented. An efficient formulation for one of the methods is proposed which requires significantly less computation than that required for the Kaiser window.

Next, a method for selecting the three independent parameters of the ultraspherical window so as to achieve prescribed spectral characteristics is proposed. The method can be used to achieve a specified ripple ratio and either a main-lobe width or null-to-null width along with a user-defined side-lobe pattern. The side-lobe pattern in other known two-parameter windows cannot be controlled as in the proposed method. Applications of the proposed method in digital beamforming and image processing are explored.

A closed-form method for the design of nonrecursive digital filters using the ultraspherical window is developed. The method can be used to design lowpass, highpass, bandpass, and bandstop filters as well as digital differentiators and Hilbert transformers that would satisfy prescribed specifications. The method yields lower-order filters relative to designs obtained with other windows such as the Kaiser, Saramäki, and Dolph-Chebyshev windows. Alternatively, for a fixed filter length, the ultraspherical window can provide reduced passband ripple and increased stopband attenuation. In addition, it entails reduced computational complexity which renders it suitable for applications where the design must be carried out in real or quasi-real time.

An efficient closed-form method for the design of M-channel cosine-modulated filter banks using the ultraspherical window that would yield prescribed stopband attenuation in

the subbands and channel overlap is proposed. On the average, the method yields prototype filters with the shortest length and least design computational complexity while the Kaiser window yields filter banks with the smallest reconstruction error. When compared with other methods, the proposed method yields filter banks that have prototype filters of the same length, increased average maximum amplitude error, and the same average aliasing error and average total aliasing error.

The dissertation also considers the application of the ultraspherical window along with the short-time discrete Fourier transform method for gene identification based on the well known period-three property. The ultraspherical window is employed to suppress spectral noise originating from noncoding regions in the DNA sequence. A method for tailoring the independent parameters of the ultraspherical window for the identification of a particular gene is proposed. Comparisons show that the ultraspherical, Kaiser, and Saramäki windows yield values for a gene-identification measure that are approximately the same, and that they are 13.72% better than that achieved when using the rectangular window.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AF	Array factor
CAT	Computerized tomography
CCD	Charge-coupled device
CMFB	Cosine modulated filter bank
CPU	Central processing unit
CR	Contrast ratio
DD	Digital differentiator
DFT	Discrete Fourier transform
DNA	Deoxyribonucleic acid
DSP	Digital signal processing
FFT	Fast Fourier transform
MPEG	Moving pictures experts group
P-3	Period-three
SAR	Synthetic aperture radar
SNR	Signal-to-noise
STDFT	short-time discrete Fourier transform
TMUX	Transmultiplexer

Acknowledgement

First and foremost I wish to express gratitude towards my supervisor Dr. Andreas Antoniou for his unwavering encouragement and support during the research and writing of this dissertation. Without his superb mentorship and willingness to help, this work would not have been possible.

Second, I am indebted to my committee members for their time and effort in reviewing this dissertation.

Third, I wish to thank the graduate students of the Digital Signal Processing Group for their enlightening and spirited conversations work-related and otherwise. Special thanks go to Rajeev Nongpiur, Brad Riel, Nanyan Wang, and Sabbir Ahmad.

Finally, I wish to thank the staff of the Department of Electrical and Computer Engineering for their timely responses to my degree-related enquires and requests. Special thanks go to Vicky Smith and Lynne Barrett.

Dedication

To My Family and Friends

Chapter 1

Introduction

1.1 Background

Signal processing is a technology used in a wide range of disciplines. It is most prevalent in fields of physical science and engineering such as communications and control systems; however, it also finds uses in non-tradition fields such as medicine and bioinformatics. Systems using signal processing include everyday consumer products such as the television or compact-disc player as well as some highly specialized applications such as military radar and tracking systems.

Signal processing is used to represent, transform, and manipulate signals and the information they contain and can be performed on both continuous- and discrete-time signals. Prior to the 1960s, signal processing algorithms were implemented primarily with continuous-time systems using analog circuitry and even mechanical devices. At that time, computers lacked the processing capability to make discrete-time systems practical. Discrete-time systems were initially used to perform classical numerical analysis techniques such as interpolation, differentiation, and integration. The roots of digital filtering occurred in this respect because these operations represent a manipulation of the frequency spectrum of a signal. In subsequent years, many sophisticated algorithms were formulated by researchers in academic institutions as well as in industry to perform digital filtering tasks. However, it was not until 1965 when the fast Fourier transform (FFT) was introduced that digital signal processing (DSP) began to gain acceptance for practical applications.

Today DSP algorithms and digital filters are widely used. They can be implemented in hardware or software and can process both real-time and off-line (recorded) signals. Digital hardware now routinely performs tasks that were almost exclusively performed by analog systems in the past. Likewise, software programs have been developed such as MATLAB that enable users to implement complex DSP algorithms with simple function calls. Such advancements present DSP-based systems as an easy-to-use and flexible alternative to analog systems. Advancements in hardware design, software design, and algorithm development will continue to fuel the adoption of DSP in new disciplines and tasks previously restricted to analog systems.

1.2 Fourier Series

The Fourier series of a periodic function $x(t)$ with period T is a representation of $x(t)$ in terms of an infinite sum of sine and cosine functions of the form

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k \cos k\omega_0 t + b_k \sin k\omega_0 t) \tag{1.1}$$

where $\omega_0 = 2\pi/T$ is called the fundamental frequency and $k\omega_0$ is its kth harmonic. The coefficients of the Fourier series are given by [1]

$$a_k = \frac{1}{\pi}\int_{-T/2}^{T/2} x(t)\cos k\omega_0 t\, dt \quad \text{and} \quad b_k = \frac{1}{\pi}\int_{-T/2}^{T/2} x(t)\sin k\omega_0 t\, dt \tag{1.2}$$

1.3 Gibbs' Oscillations and Early Smoothing

In practice it is usually required to truncate an infinite Fourier series; however, truncation of a Fourier series causes so-called Gibbs' oscillations (also known as ringing) which are most pronounced near jump discontinuities. For example, the truncated Fourier series of

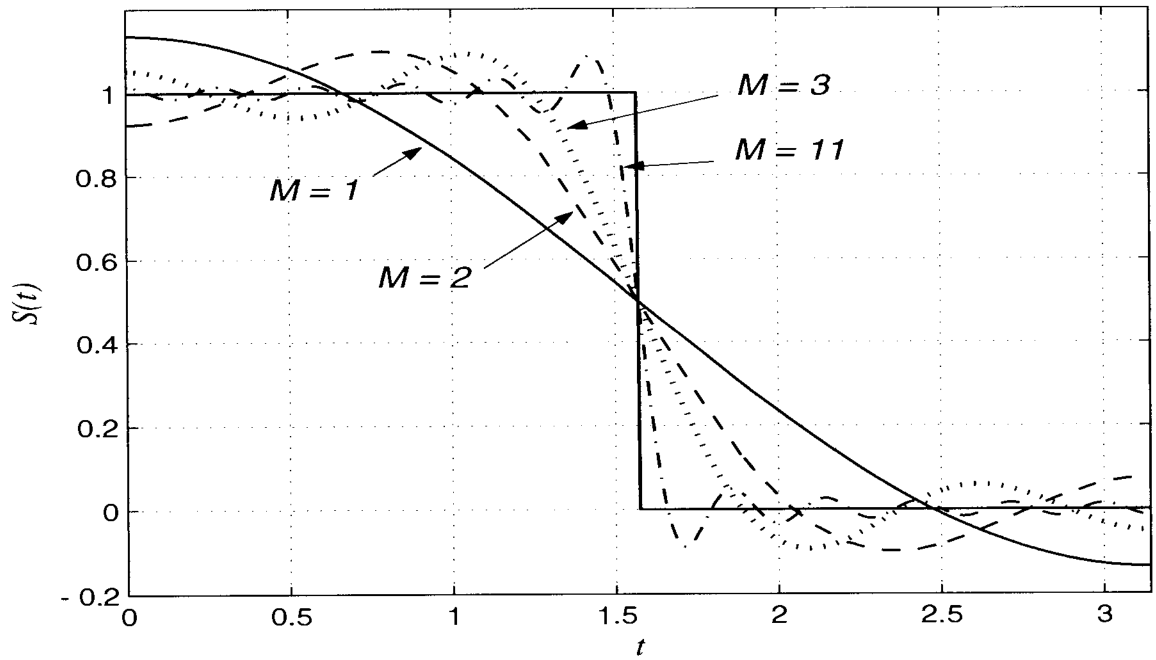


Figure 1.1. *Truncated Fourier series with* $M = 1$ *(solid line),* 2 *(dashed line),* 3 *(dotted line), and* 11 *(dashed-dotted line) terms.*

the signal

$$x(t) = \begin{cases} 0 & \text{for } -\pi \le t < -\pi/2 \\ 1 & \text{for } -\pi/2 \le t \le \pi/2 \\ 0 & \text{for } \pi/2 < t \le \pi \end{cases} \tag{1.3}$$

can be expressed as

$$S(t) = \frac{1}{2} + \frac{2}{\pi}\sum_{k=1}^{M}(-1)^k \frac{\cos(2k+1)t}{(2k+1)} \tag{1.4}$$

where M is the number of terms retained. This is illustrated in Fig. 1.1 for $M = 1, 2, 3$, and 11. As M increases, the amplitude of the oscillations near the discontinuity tends to remain approximately constant. These oscillations were explained mathematically by Gibbs and thus became known as Gibbs' oscillations [2].

The performance offered by a truncated Fourier series is often objectionable for practical applications and ways must be sought for the reduction of Gibbs' oscillations. One of the first approaches for smoothing out Gibbs' oscillations was offered by Fejer who suggested averaging a number of truncated Fourier series [3]. This process, which is sometimes referred to as Fejer averaging, can be implemented by applying the multiplicative factor

$$A(M,k) = \frac{M-k}{M} \tag{1.5}$$

to a truncated Fourier series as follows:

$$S(t) = \frac{a_0}{2} + \sum_{k=1}^{M} A(M,k)[a_k \cos k\omega_0 t + b_k \sin k\omega_0 t] \tag{1.6}$$

Another smoothing approach for Gibbs' oscillations was proposed by Lanczos who observed that the amplitude of the oscillations of a truncated Fourier series have approximately the same period as either the first term neglected or the last term kept in the series [4]. He argued that smoothing the truncated Fourier series over this period would reduce the amplitude of the oscillations. This process is called Lanczos smoothing and can be implemented by applying the multiplicative factor (sometimes called the sigma factor)

$$A(M,k) = \frac{\sin \pi k/M}{\pi k/M} \tag{1.7}$$

in Eq. (1.6). Figure 1.2 shows plots of a truncated Fourier series after applying the Fejer averaging and Lanczos smoothing techniques. As can be seen, Lanczos smoothing yields a better approximation than Fejer averaging, which is rarely used in practical applications.

A function with only one jump discontinuity has been examined here; however, Gibbs' oscillations and the performance obtained by using smoothing factors are characteristic of any truncated Fourier series regardless of the number of discontinuities or their locations.

1.4 Window Functions

A more comprehensive view of the truncation and smoothing operations is in terms of window functions (or windows for short). The truncated Fourier series can be obtained by assigning

$$c_n = 0 \quad \text{for } |n| > M \tag{1.8}$$

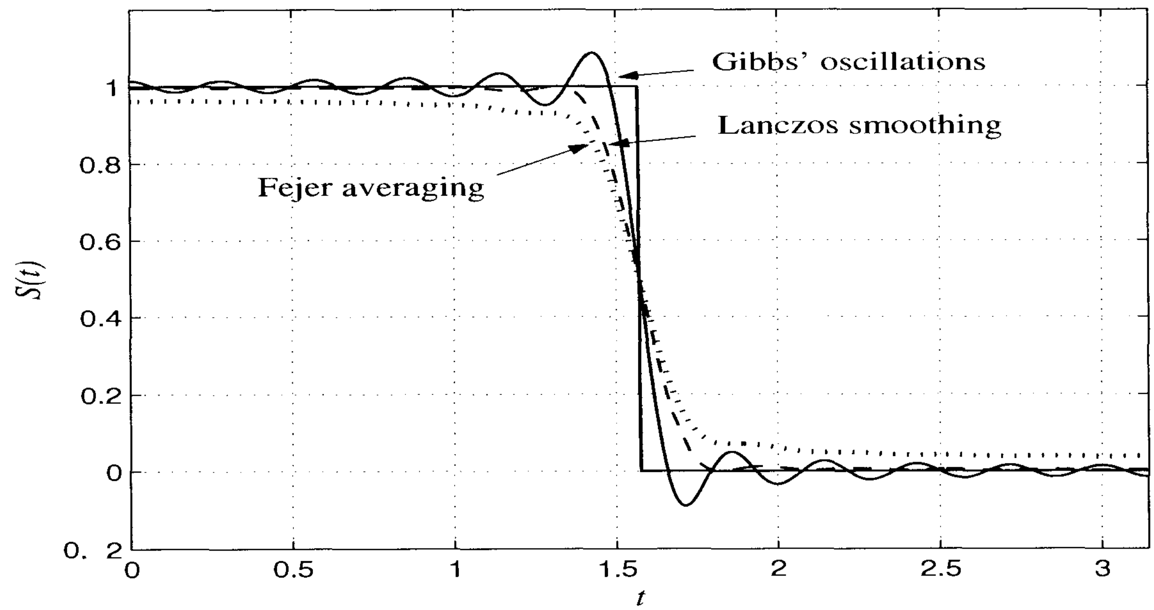


Figure 1.2. *Truncated Fourier series for $M = 11$ using no smoothing (solid line), Fejer averaging (dotted line), and Lanczos smoothing (dashed line).*

in the exponential Fourier series given by

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \quad \text{where} \quad c_k = \frac{1}{2\pi}\int_{-T/2}^{T/2} x(t)e^{-jk\omega_0 t}\,dt \tag{1.9}$$

Alternatively, the truncated Fourier series can be obtained by using the multiplicative factor

$$w_R(nT) = \begin{cases} 1 & \text{for } |n| \leq M \\ 0 & \text{otherwise} \end{cases} \tag{1.10}$$

which can be referred to as the *rectangular window* for obvious reasons. The windowing operation is illustrated in Fig. 1.3.

Windows are frequently compared and classified in terms of their spectral characteristics. The spectral representation of a window $w(nT)$ of length $N = 2M + 1$ defined over the range $-M \leq n \leq M$ is given by the z transform of $w(nT)$ evaluated on the unit-circle of the z plane, i.e.,

$$W(e^{j\omega T}) = \sum_{n=-M}^{M} w(nT)e^{-j\omega nT} \tag{1.11}$$

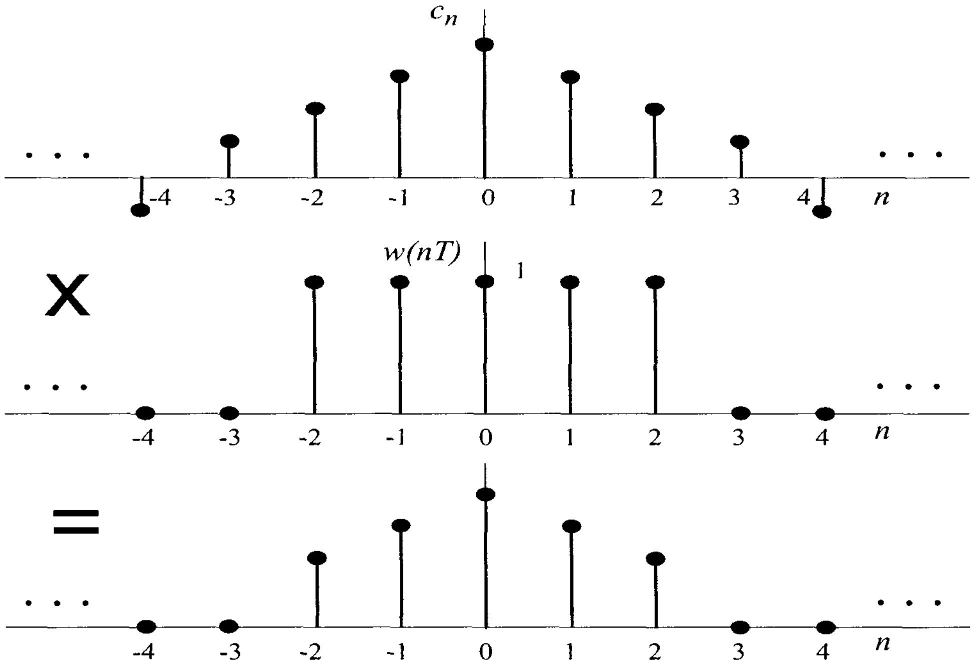


Figure 1.3. *Windowing operation - the pointwise multiplication of the signal's Fourier coefficients by the window coefficients.*

The frequency spectrum of a window is given by

$$W(e^{j\omega T}) = e^{-j\omega MT} W_0(e^{j\omega T}) \tag{1.12}$$

where $W_0(e^{j\omega T})$ is called the amplitude function. The amplitude and phase spectrums of a window are given by $A(\omega) = |W_0(e^{j\omega T})|$ and $\theta(\omega) = -\omega MT$, respectively, and $|W_0(e^{j\omega T})|/W_0(e^0)$ is a normalized version of the amplitude spectrum. A typical window's normalized amplitude spectrum and some common spectral characteristics are depicted in Fig. 1.4.

Two parameters of windows in general are the null-to-null width B_n and the main-lobe width B_r. These quantities are defined as $B_n = 2\omega_n$ and $B_r = 2\omega_r$, where ω_n and ω_r are the half null-to-null and half main-lobe widths, respectively, as shown in Fig. 1.4. An

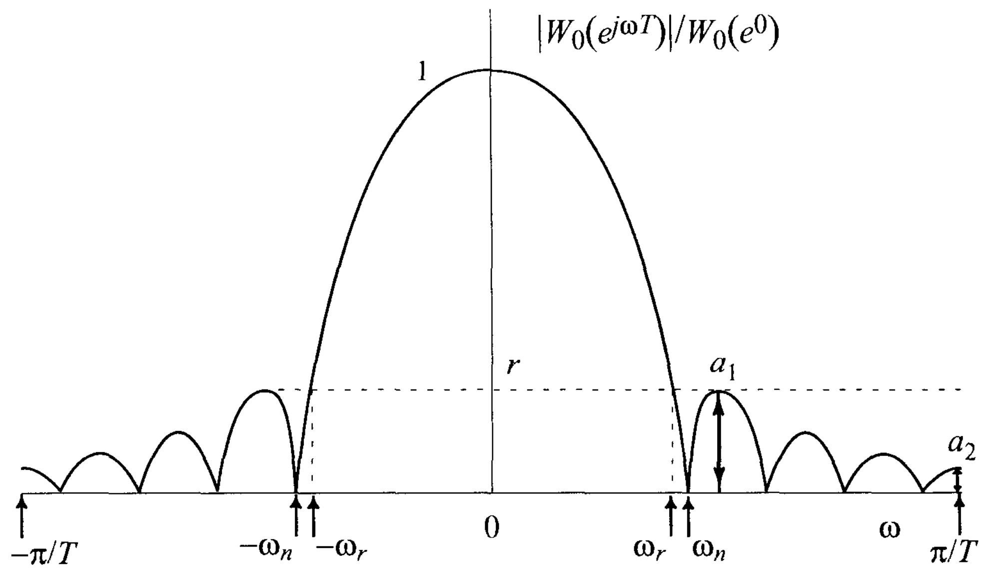


Figure 1.4. *Amplitude spectrum and some common spectral characteristics of a typical normalized window.*

important window parameter is the ripple ratio r which is defined as

$$r = \frac{\text{maximum side-lobe amplitude}}{\text{main-lobe amplitude}} \tag{1.13}$$

(see Fig. 1.4). The ripple ratio is a small quantity less than unity and, in consequence, it is convenient to work with the reciprocal of r in dB, i.e.,

$$R = 20\log(1/r) \tag{1.14}$$

where R can be interpreted as the minimum side-lobe attenuation relative to the main lobe and $-R$ is the ripple ratio in dB. Another parameter used to describe the side-lobe pattern of a window is the side-lobe roll-off ratio, s, which is defined as

$$s = a_1/a_2 \tag{1.15}$$

where a_1 and a_2 are the amplitudes of the side lobe nearest and furthest, respectively, from the main lobe (see Fig. 1.4). If S is the side-lobe roll-off ratio in dB, then s is given by

$$s = 10^{S/20} \tag{1.16}$$

For the side-lobe roll-off ratio to have meaning, the envelope of the side-lobe pattern should be monotonically increasing or decreasing.

These spectral characteristics are important performance measures for windows. When analyzing bandlimited signals, such as sinusoids, weak signals can easily be obscured by nearby strong signals. The width characteristics provide a resolution measure between adjacent signals while the ripple ratio determines the worst-case scenario for detecting weak signals in the presence of strong signals. The side-lobe roll-off ratio provides a description of the distribution of energy throughout the side lobes, which can be of importance if prior knowledge of the location of an interfering signal is known. Further explanation of the usefulness of these spectral characteristics can be found in [5].

The windowing operation is equivalent to the pointwise multiplication of two discrete-time signals at each instant in time. The z transform of these two discrete-time signals is equal to the complex convolution of the z transforms of the two signals. Evaluating the complex convolution on the unit circle of the z plane yields

$$X_w(e^{j\omega T}) = \frac{T}{2\pi} \int_0^{2\pi/T} X(e^{j\varpi T}) W(e^{j(\omega-\varpi)T})\, d\varpi \tag{1.17}$$

which is the convolution of the frequency spectrums of the window and the signal. The effects of a window on a signal can be illustrated by considering a signal $x(t)$ with the frequency spectrum

$$X(e^{j\omega T}) = \begin{cases} 1 & \text{for } -\pi/2 \leq \omega \leq \pi/2 \\ 0 & \text{otherwise} \end{cases} \tag{1.18}$$

and a window with spectrum $W(e^{j\omega T})$ similar to that depicted in Fig. 1.4. The complex convolution is illustrated in Fig. 1.5. The side lobes in the spectrum of the window cause ripples in $X_w(e^{j\omega T})$ whose amplitude is proportional to the ripple ratio. Further, the width of the transition bands in $X_w(e^{j\omega T})$ is proportional to the main-lobe width of the window.

The convolution process reveals two distinct changes in the frequency spectrum of a signal resulting from the windowing process.

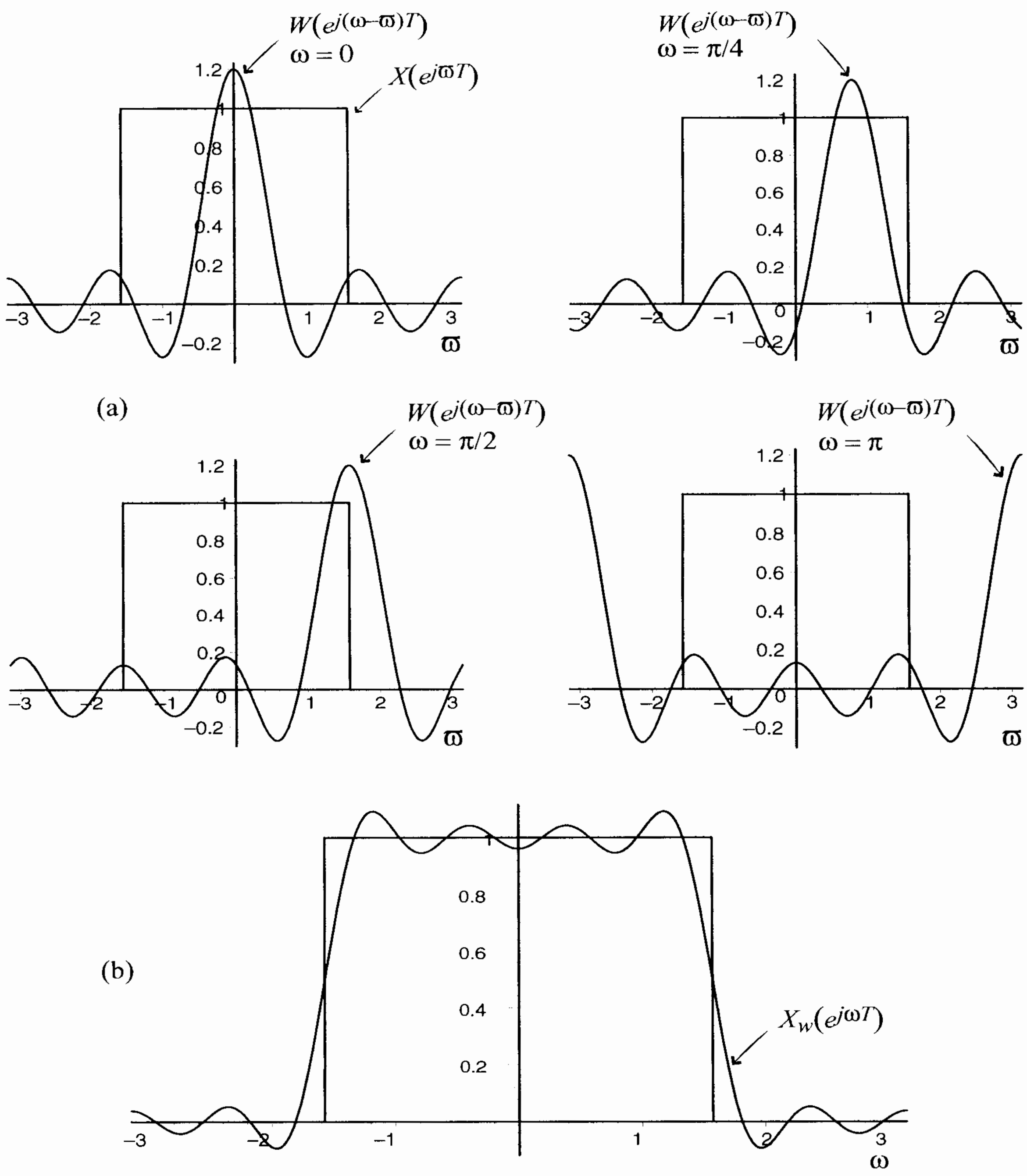


Figure 1.5. *Effect of windowing in the frequency domain. (a) The complex convolution process. (b) The response of the resulting signal.*

1. *Spectral spreading* occurs at jump discontinuities in $X(e^{j\omega T})$ resulting in gradual transitions for one level to the next instead of a sudden switch.

2. *Spectral leakage* occurs in the form of Gibbs' oscillations in zero values of the signal, i.e., although $X(e^{j\omega T})$ is bandlimited, $X_w(e^{j\omega T})$ is not.

Both effects cause the loss of spectral resolution. Spectral spreading (or smearing) causes loss of resolution between adjacent spectral lines and is directly proportional to the main-lobe width of the window. Increased smearing occurs with wider main-lobe widths, which usually correspond to shorter window lengths. Conversely, decreased smearing occurs with narrower main-lobe widths, which usually correspond to longer window lengths. On the other hand, spectral leakage determines the worst-case scenario for detecting weak spectral lines in the presence of strong spectral lines nearby and is proportional to the ripple ratio of the window.

1.5 Some Prominent Windows

In practice, spectral spreading and leakage are opposing effects and the improvement in one inevitably leads to the deterioration of the other. To accommodate varied spectral requirements, a number of windows have been proposed over the years which can be broadly categorized as either fixed or adjustable [6]. Fixed windows have only one independent parameter, namely, the window length which controls the main-lobe width and thus spectral spreading. Some of the more popular fixed windows in addition to the rectangular include the triangular (Fejer averaging), von Hann, Hamming, and Blackman windows (expressions and explanations can be found in [5]). Unfortunately, fixed windows do not permit adjustable ripple ratios and thus provide no control over spectral leakage. Conversely, adjustable windows have two or more independent parameters, namely, the window length, as in fixed windows, and one or more additional parameters that can control other window characteristics. Some of the more popular adjustable windows include the Kaiser and Saramäki windows [7], [8], which have two parameters and achieve close approximations to discrete prolate functions that have maximum energy concentration in the main lobe. Another popular window is the Dolph-Chebyshev window [9] which has two parameters and produces the minimum main-lobe width for a specified maximum side-lobe level. The Kaiser, Saramäki, and Dolph-Chebyshev windows can control the amplitude of the side lobes relative to that of the main lobe and thus can provide control over both spectral spreading and leakage. Figure 1.6 illustrates the weighting functions and spectral representations of the rectangular, Kaiser and Dolph-Chebyshev windows of length $N = 51$. Of the three windows, the rectangular window offers the smallest main-lobe width, however it also possesses the largest ripple ratio. On the other hand, both the Kaiser and Dolph-Chebyshev windows provide smaller ripple ratios but at the expense of an increased main-lobe width. Furthermore, if we are to compare the distribution of energy in the side lobes (the side-lobe patterns), all of the windows provide significantly different results. Obviously no one window is best for all situations but rather superior only for particular situations that arise from different applications. The Kaiser window has an important advantage over other parametric windows. It can be used to design filters that satisfy prescribed specifications [1], [7].

Windows are used to reduce Gibbs' oscillations and they are employed in a variety of signal processing applications such as power spectral estimation, beamforming, and digital filter design. Despite their maturity, windows continue to find new roles in the applications of today. Very recently, windows have been used to facilitate the detection of irregular and abnormal heartbeat patterns in patients in electrocardiograms [10], [11]. Medical imaging systems, such as ultrasound, have shown enhanced performance when windows are used to improve the contrast resolution of the system [12]. Windows have also been employed to aid in the classification of cosmic data [13], [14] and to improve the reliability of weather prediction models [15]. With such a large number of applications for windows available that span a variety of disciplines, window flexibility becomes a key concern.

Another parametric window is the ultraspherical window which has three independent parameters for controlling its properties [16]. Through the proper choice of these parame-

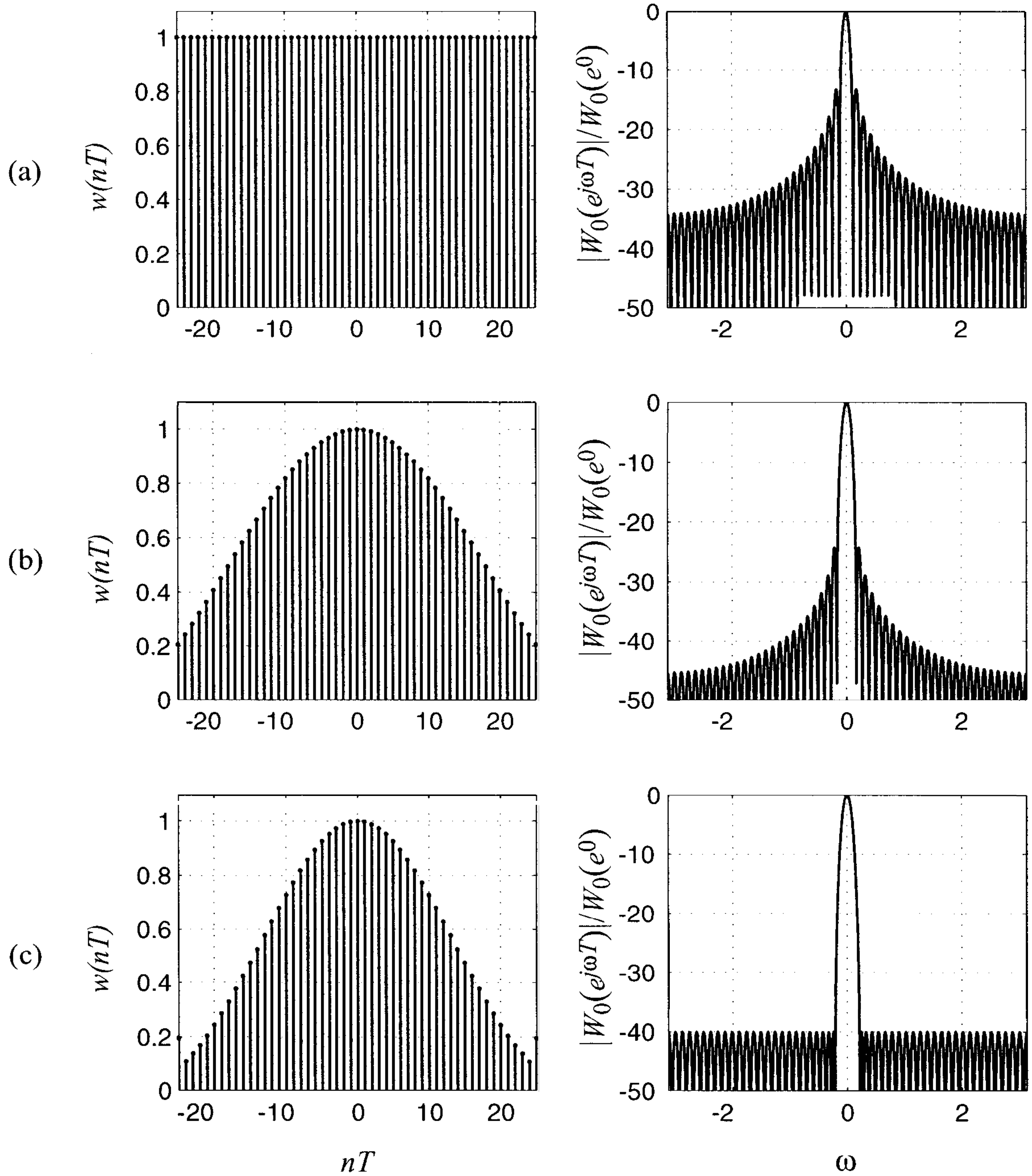


Figure 1.6. *Windows and their spectral representations. (a) Rectangular window. (b) Kaiser window ($\alpha = 3$). (c) Dolph-Chebyshev window ($R = 40$).*

ters, the amplitude of the side lobes relative to that of the main lobe can be controlled as in the Kaiser, Saramäki, and Dolph-Chebyshev windows; and in addition, a variety of side-lobe patterns can be achieved. To facilitate the application of the ultraspherical window to the diverse range of applications alluded to earlier, practical and efficient design methods are required that can utilize its inherent flexibility.

1.6 Nonrecursive Digital Filter Design

Many methods for nonrecursive digital-filter design have been proposed and a comprehensive review of state-of-the-art methods can be found in [1]. Two of the more popular methods are the window and weighted-Chebyshev methods. The window method is based largely on closed-form solutions and, as a result, it is straightforward to apply and entails a relatively insignificant amount of computation. Unfortunately, the window method usually yields suboptimal designs whereby the filter order required to satisfy a given set of specifications is not the lowest that can be achieved. On the other hand, multivariable optimization algorithms for nonrecursive digital-filter design, e.g., the weighted-Chebyshev method of Parks and McClellan [17], [18] and the more recent generalized Remez method of Shpak and Antoniou [19] yield optimal designs with respect to some error criterion; however, these algorithms generally require a large amount of computation and are, therefore, unsuitable for real or quasi-real time applications like portable multimedia devices where on-the-fly designs that adapt to changing environmental conditions such as battery power and quality-of-service issues are required. Since each method has advantages and disadvantages, it is important for filter designers to consider the application at hand when selecting a filter-design method.

1.7 Scope of Thesis

Parametric windows find uses in many applications due to their simplicity, low computational complexity, and closed-form solutions; and they are easily modified by adjusting their independent parameters. A limitation of two-parameter windows is that they cannot adjust the side-lobe pattern. In this dissertation, the application of window functions based on the ultraspherical window is explored. Other parametric windows such as the Kaiser, Saramäki, and Dolph-Chebyshev windows are used throughout the dissertation for the sake of comparison.

In Chapter 2, two methods for evaluating the coefficients of the ultraspherical window are presented. The first method corresponds to a concise exposition of Streit's method [16]. The second is a new method that involves equating an ultraspherical window's frequency-domain representation to a Fourier series from which the coefficients are readily found. The two methods yield the same coefficients for the same independent parameters. The computational complexity associated with the two methods is compared and an efficient formulation for the evaluation of the coefficients is proposed. The new formulation constitutes a computational complexity of $O(N)$ as compared with $O(N^2)$ for the previous formulation. Alternatively, the amount of computation of the new formulation is on the average 4.49% of that required for the previous formulation and 9.27% of that required for the evaluation of the Kaiser window coefficients. Aspects of the ultraspherical window's frequency spectrum and its equivalence to other windows are also considered.

In Chapter 3, a method for selecting the three independent parameters of the ultraspherical window so as to achieve prescribed spectral characteristics is proposed. As discussed in Section 1.4, the spectral characteristics of a window are important performance measures for window applications such as power spectral estimation. The width characteristics provide a resolution measure between adjacent signals, the ripple ratio determines the worst-case scenario for detecting weak signals in the presence of strong signals, and the side-lobe roll-off ratio provides a description of the distribution of energy throughout the side lobes. The method comprises a collection of techniques that can be used to achieve a specified ripple ratio and either a main-lobe width or null-to-null width along with a user-defined side-lobe pattern. The side-lobe pattern in other known two-parameter windows cannot be controlled as in the proposed method. In addition, an expression is provided that can be used to judge how much ripple ratio is sacrificed to attain a given side-lobe pattern when compared to the Dolph-Chebyshev pattern. This is useful for antenna array designers who may need to trade-off between side-lobe pattern and ripple ratio for the application at hand. The proposed method can also be used to increase the contrast ratio in imaging systems that construct images by using two-dimensional windowed inverse DFTs on spatial frequency-domain data such as synthetic aperture radar (SAR), computerized tomography (CAT scans), and charge-coupled device (CCD)-based X-rays.

In Chapter 4, a closed-form method for the design of nonrecursive digital filters using the ultraspherical window and the proposed efficient formulation for evaluating its coefficients is developed. The method can be used to design lowpass, highpass, bandpass, and bandstop filters as well as digital differentiators and Hilbert transformers that satisfy prescribed specifications. The ultraspherical window yields lower-order filters relative to designs obtained using other windows yielding on the average a reduction of 3.07% relative to the Kaiser window, 2.86% relative to the Saramäki window, and 5.30% relative to the Dolph-Chebyshev window. Alternatively, for a fixed filter length, the ultraspherical window increases the stopband attenuation relative to the other windows achieving on the average an increase of 2.61 dB relative to the Kaiser window, 2.42 dB relative to the Saramäki window, and 4.49 dB relative to the Dolph-Chebyshev window. On the other hand, the weighted-Chebyshev method increases the stopband attenuation relative to the ultraspherical window by about 2.76 dB on the average; however, the computational complexity associated with the weighted-Chebyshev method is far greater than that required by the proposed method.

In Chapter 5, an efficient closed-form method for the design of M-channel cosine-modulated filter banks using the ultraspherical window that would yield prescribed stopband attenuation in the subbands and channel overlap is proposed. The design of the prototype filter is based on the proposed method for the design of lowpass filters described in Chapter 4. On the average, use of the Kaiser window yields filter banks with the smallest reconstruction error achieving an average percentage decrease in error over the Saramäki and ultraspherical windows of, respectively, 11.69% and 12.17% for the maximum amplitude error in the filter bank, 1.34% and 26.51% for the maximum aliasing error in the filter bank, and 2.11% and 34.65% for the maximum total aliasing error in the filter bank. On the other hand, use of the ultraspherical window yields filter banks with the least amount of design computational complexity (due to the efficient formulation proposed in Chapter 2) and prototype filters with the shortest length (as described in Chapter 4). When compared with two other window-based optimization design methods, the proposed method increased the average maximum amplitude error by 9.53% and 1.52%, respectively, provided almost no change in the average aliasing error and the average total aliasing error, and produced prototype filters of the same length. The computational effort required by the proposed design method is a small fraction, less than 2%, of that required by the other two methods which require solutions to one-dimensional optimization problems. When compared to a filter-bank design method that employs the weighted-Chebyshev method for the prototype filter design, the proposed method requires significantly less computation and can be used to achieve the prescribed specifications; the other method cannot be used to achieve the prescribed specifications and requires a huge amount of computation due to the repeated use of the Remez exchange algorithm within an optimization routine.

In Chapter 6, the application of the ultraspherical window along with the short-time discrete Fourier transform method for gene identification based on the well known period-three property is explored. The ultraspherical window is employed to suppress spectral noise originating from noncoding regions in the DNA sequence. A method for tailoring the independent parameters of the ultraspherical window for the identification of a particular gene is proposed. When the method was applied to gene F56F11.4 of the *C.elegans* organism, a signal-to-noise (SNR)-based measure for gene identification was increased by

13.72% relative to that achieved when using the rectangular window. Comparisons show that the ultraspherical, Kaiser, and Saramäki windows yield approximately the same SNR values when their parameters are optimized. The Dolph-Chebyshev window yields an SNR value that is 0.28% smaller than that of the other windows.

Chapter 2

The Ultraspherical Window Function

2.1 Introduction

Not long ago, Streit [16] explored the use of ultraspherical polynomials (also known as Gegenbauer polynomials) [20] to produce weighting functions with a variety of side-lobe patterns for use in symmetric equally-spaced broadside antenna arrays. These weighting functions can be considered as window functions with three parameters thereby introducing an extra degree of freedom relative to two-parameter windows such as the Kaiser, Saramäki, and Dolph-Chebyshev windows. After Streit's work, Soltis [21], [22], and Saèd et al. [23] used ultraspherical polynomials to further investigate antenna arrays and used them in wavelet analysis. Later, Deczky [24] used the ultraspherical window to provide a proof-of-concept example for nonrecursive digital-filter design.

In this chapter, methods for evaluating the coefficients of the ultraspherical window are developed. The chapter is structured as follows. Section 2.2 explores two methods for evaluating the coefficients of the ultraspherical window. Section 2.3 describes the spectral properties and characterizations of the ultraspherical window. Section 2.4 proposes an efficient formulation for evaluating the coefficients of the ultraspherical window.

2.2 Window Coefficients

The coefficients of a right-sided ultraspherical window can be calculated explicitly for an even or odd length N as [16]

$$w(nT) = \frac{A}{p-n}\binom{\mu+p-n-1}{p-n-1} \cdot \sum_{m=0}^{n} \binom{\mu+n-1}{n-m}\binom{p-n}{m} B^m \tag{2.1}$$

for $n = 0, 1, \ldots, N-1$, where [20]

$$\binom{\alpha}{p} = \frac{\alpha(\alpha-1)\cdots(\alpha-p+1)}{p!} \quad \text{for } p \geq 1 \tag{2.2}$$

with $\binom{\alpha}{0} = \binom{\alpha}{\alpha} = 1$ because $\binom{n}{k} = \binom{n}{n-k}$. T is the interval between samples and

$$A = \begin{cases} \mu x_\mu^p & \text{for } \mu \neq 0 \\ x_\mu^p & \text{for } \mu = 0 \end{cases} \tag{2.3}$$

$$B = 1 - x_\mu^{-2} \tag{2.4}$$

$$p = N - 1 \tag{2.5}$$

In Eq. (2.1), μ, x_μ, and N are independent parameters and $w[(N-n-1)T] = w(nT)$, i.e., the window is symmetrical. A normalized window is obtained as $\hat{w}(nT) = w(nT)/w(DT)$ where

$$D = \begin{cases} (N-1)/2 & \text{for odd } N \\ N/2 - 1 & \text{for even } N \end{cases} \tag{2.6}$$

A second method for the computation of the window coefficients involves equating an ultraspherical window's frequency-domain representation to a Fourier series. To start with, we take a lead from Stegen [25] where he notes that a sum

$$F(x) = \sum_{m=0}^{r} (a_m \cos mx + b_m \sin mx) \tag{2.7}$$

can be found that furnishes the best possible representation of a function $u(x)$ that takes the values $u_0, u_1, u_2, \ldots, u_{n-1}$, when x takes the values $0, 2\pi/n, 4\pi/n, \ldots, 2(n-1)\pi/n$, respectively, where $n \geq 2r + 1$. The coefficients in Eq. (2.7) are given by

$$a_0 = \frac{1}{n} \sum_{k=0}^{n-1} u_k \tag{2.8}$$

$$a_m = \frac{2}{n} \sum_{k=0}^{n-1} u_k \cos \frac{2k\pi m}{n} \tag{2.9}$$

and

$$b_m = \frac{2}{n} \sum_{k=0}^{n-1} u_k \sin \frac{2k\pi m}{n} \tag{2.10}$$

If we set $r = (N-1)/2$ and $n = 2r + 1$, the values $u_k = u(x_k)$ in Eqs. (2.8), (2.9), and (2.10) are found by setting

$$u(x) = C_{N-1}^{\mu}\left(x_\mu \cos \frac{x}{2}\right) \tag{2.11}$$

where $C_n^{\lambda}(x)$ is the ultraspherical polynomial of degree n and order λ, and subsequently finding $u(x_s)$ at N points distributed over x given by

$$x_s = \frac{2\pi}{N} s \quad \text{for } s = 0, 1, ..., N-1 \tag{2.12}$$

where

$$u_s = u(x_s) = C_{N-1}^{\mu}\left(x_\mu \cos \frac{\pi s}{N}\right) \tag{2.13}$$

The ultraspherical polynomial can be calculated using the recurrence relationship [20]

$$C_r^{\lambda}(x) = \frac{1}{r}\left[2x(r + \lambda - 1)C_{r-1}^{\lambda}(x) - (r + 2\lambda - 2)C_{r-2}^{\lambda}(x)\right] \tag{2.14}$$

for $r = 2, 3, ..., n$, where $C_0^{\lambda}(x) = 1$ and $C_1^{\lambda}(x) = 2\lambda x$. With $b_m = 0$, the expressions for coefficients a_0 and a_m become

$$\begin{aligned} a_0 &= \frac{1}{N} \sum_{s=0}^{N-1} u_s \\ &= \frac{1}{N}\left(u_0 + \sum_{s=1}^{(N-1)/2} u_s + \sum_{s=\frac{N-1}{2}+1}^{N-1} u_s\right) \end{aligned} \tag{2.15}$$

and

$$a_m = \frac{2}{N}\sum_{s=0}^{N-1} u_s \cos\frac{2s\pi m}{N}$$
$$= \frac{2}{N}\left(u_0 + \sum_{s=1}^{(N-1)/2} u_s \cos\frac{2s\pi m}{N} + \sum_{s=\frac{N-1}{2}+1}^{N-1} u_s \cos\frac{2s\pi m}{N}\right) \qquad (2.16)$$

Now in an effort to simplify the above expressions, we note that $C_{N-1}^{\mu}\left(x_\mu \cos x/2\right)$ is of degree $N-1$ in $X = x_\mu \cos x/2$. As such, it is an even function of X implying that $C_{N-1}^{\mu}\left(X\right) = C_{N-1}^{\mu}\left(-X\right)$. Using this property, Eqs. (2.15) and (2.16) yield

$$a_0 = \frac{1}{N}\left[C_{N-1}^{\mu}(x_\mu) + 2\sum_{s=1}^{(N-1)/2} C_{N-1}^{\mu}\left(x_\mu \cos\frac{\pi s}{N}\right)\right] \qquad (2.17)$$

and

$$a_m = \frac{2}{N}\left[C_{N-1}^{\mu}(x_\mu) + 2\sum_{s=1}^{(N-1)/2} C_{N-1}^{\mu}\left(x_\mu \cos\frac{\pi s}{N}\right)\cdot\cos\frac{2\pi sm}{N}\right] \qquad (2.18)$$

We can now express the window coefficients for the ultraspherical window of odd length as

$$w(nT) = \frac{1}{N}\left[C_{N-1}^{\mu}(x_\mu) + 2\sum_{s=1}^{(N-1)/2} C_{N-1}^{\mu}\left(x_\mu \cos\frac{\pi s}{N}\right)\cdot\cos\frac{2\pi s(n-\frac{N-1}{2})}{N}\right] \qquad (2.19)$$

and for even length as

$$w(nT) = \frac{1}{N}\left[C_{N-1}^{\mu}(x_\mu) + 2\sum_{s=1}^{N/2-1} C_{N-1}^{\mu}\left(x_\mu \cos\frac{\pi s}{N}\right)\cdot\cos\frac{2\pi s(n-\frac{N}{2})}{N}\right] \qquad (2.20)$$

for $n = 0, 1, \ldots, N-1$. A normalized window is obtained as $\widehat{w}(nT) = w(nT)/w(DT)$. This method of computation of the ultraspherical window coefficients produces the same results as Eq. (2.1) given the same set of independent parameters μ, x_μ, and N.

Figure 2.1 shows a comparison of the computation time associated with Eqs. (2.1) and (2.19) for increasing values of N. The high computational complexity in Eq. (2.19) is primarily due to the repeated calculation of $C_n^{\mu}\left(x\right)$ by the recurrence relationship given in Eq. (2.14). Evidently, Eq. (2.1) offers reduced computational complexity. The computation time was measured using the MATLAB stopwatch commands *tic* and *toc* which return the total CPU time used to execute the code between the two commands.

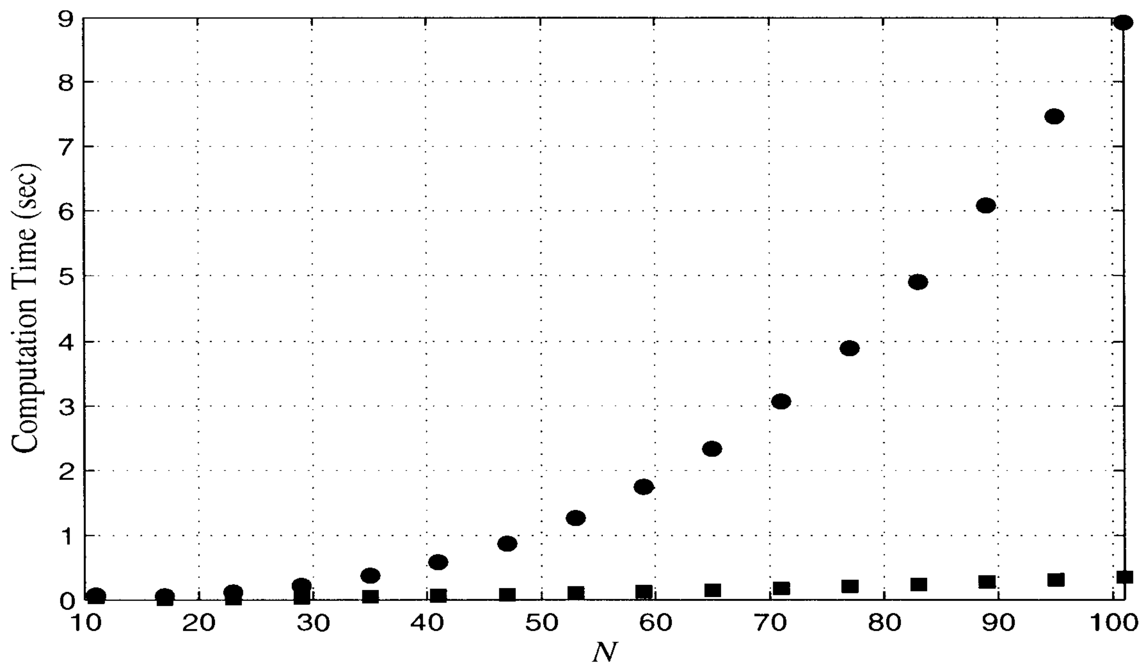


Figure 2.1. *Computation time associated with Eq. 2.1 (squares) and Eq. 2.19 (circles) vs. the window length N.*

2.3 Spectral Characterizations

The amplitude function of the ultraspherical window is given by

$$W_0(e^{j\omega T}) = C_{N-1}^{\mu}\left[x_\mu \cos(\omega T/2)\right] \tag{2.21}$$

The independent parameter x_μ can be expressed as

$$x_\mu = \frac{x_{N-1,1}^{(\mu)}}{\cos(\beta\pi/N)} \tag{2.22}$$

where $\beta \geq 1$ and $x_{N-1,1}^{(\mu)}$ is the largest zero of the ultraspherical polynomial $C_{N-1}^{\mu}(x)$. The new independent parameter β in Eq. (2.22) is the so-called *shape parameter* and can be used to set the null-to-null width of a window to $4\beta\pi/N$, i.e., β times that of the rectangular window [8]. Throughout this work, $x_{n,l}^{(\lambda)}$ is used to denote the lth zero of the ultraspherical polynomial $C_n^\lambda(x)$. Unfortunately, closed-form expressions for the zeros of this polynomial do not exist but the zeros can be found quickly using the following iterative algorithm which is valid for $l = 1$ and $\text{rnd}(n/2)$ yielding the largest and smallest zeros, respectively. The rounding operator is defined as

$$\text{rnd}(x) = \text{int}(x + 0.5) \tag{2.23}$$

where $\text{int}(y)$ is the integer part of y and is also known as the floor operator. Due to the symmetry relation $C_n^\mu(-x) = (-1)^n C_n^\mu(x)$, only the positive zeros need be considered.

Algorithm 2.1 ***l*****th zero of** $C_n^\lambda(x)$.

Step 1

Input l, λ, n, and ε.

If $\lambda = 0$, then output $x^* = \cos[\pi(l - 1/2)/n]$ and stop.

If $\lambda = 1$, then output $x^* = \cos[l\pi/(n+1)]$ and stop.

Set $k = 1$, and compute

$$y_1 = \frac{\sqrt{n^2 + 2(n-1)\lambda - 1}}{n + \lambda} \cos \frac{(l-1)\pi}{n-1} \tag{2.24}$$

Step 2

Compute

$$y_{k+1} = y_k - \frac{C_n^\lambda(y_k)}{2\lambda C_{n-1}^{\lambda+1}(y_k)}. \tag{2.25}$$

The values of $C_n^\lambda(x)$ can be calculated using Eq. (2.14). The denominator in Eq. (2.25) can be calculated quickly using the recurrence relationship [20]

$$2\lambda C_{r-1}^{\lambda+1}(x) = \frac{2\lambda + r - 1}{1 - x^2} C_{r-1}^\lambda(x) - (rx) C_r^\lambda(x) \tag{2.26}$$

which uses some of the intermediate calculations from Eq. (2.14).

Step 3

If $|y_{k+1} - y_k| \le \varepsilon$, then output $x^* = y_{k+1}$ and stop.

Set $k = k + 1$, and repeat from Step 2.

In this algorithm, ε is the termination tolerance. A good choice is $\varepsilon = 10^{-6}$ which would cause the algorithm to converge in 3 to 6 iterations. Equation (2.24) in Step 1 represents the lowest upper bound for the zeros of the ultraspherical polynomial [26]. In Step 2, the Newton-Raphson method can be used to obtain the next estimate of the zero.

The Dolph-Chebyshev window is a special case of the ultraspherical window and can be obtained by letting $\mu = 0$ in Eq. (2.1), which results in

$$W_0(e^{j\omega T}) = T_{N-1}\left[x_\mu \cos(\omega T/2)\right] \tag{2.27}$$

where

$$T_n(x) = \cos(n \cos^{-1} x) \tag{2.28}$$

is the Chebyshev polynomial of the first kind. In the Dolph-Chebyshev window, the side-lobe pattern is fixed, i.e., (1) all side lobes have the same amplitude and (2) a minimum main-lobe width is achieved for a specified side-lobe level. Hence this window is usually designed to yield a specified ripple ratio r. To design a Dolph-Chebyshev window, x_μ is calculated using the relation [9]

$$x_\mu = x_0 = \cosh\left(\frac{1}{N-1}\cosh^{-1}\frac{1}{r}\right) \tag{2.29}$$

Alternatively, the Dolph-Chebyshev window can be designed to yield a specified null-to-null width β times that of the rectangular window. This can be accomplished by using Eq. (2.22) where $x_{N-1,1}^{(\mu)} = x_{N-1,1}^{(0)}$ is the largest zero of the Chebyshev polynomial of the first kind $T_{N-1}(x)$, which is given by

$$x_{N-1,1}^{(0)} = \cos\left[\frac{\pi}{2(N-1)}\right] \tag{2.30}$$

The Saramäki window is a special case of the ultraspherical window and can be obtained by letting $\mu = 1$ in Eq. (2.1), which results in

$$W_0(e^{j\omega T}) = U_{N-1}\left[x_\mu \cos(\omega T/2)\right] \tag{2.31}$$

where

$$U_n(x) = \frac{\sin[(n+1)\cos^{-1} x]}{\sin(\cos^{-1} x)} \tag{2.32}$$

is the Chebyshev polynomial of the second kind. The Saramäki window, like the Kaiser window, leads to close approximations of the discrete prolate functions and is designed to yield a null-to-null width β times that of the rectangular window. This can be accomplished by using Eq. (2.22) where $x_{N-1,1}^{(\mu)} = x_{N-1,1}^{(1)}$ is the largest zero of the Chebyshev polynomial of the second kind $U_{N-1}(x)$, which is given by

$$x_{N-1,1}^{(1)} = \cos(\pi/N) \tag{2.33}$$

Another special case of interest is the case where $\mu = 0.5$ in Eq. (2.1), which results in

$$W_0(e^{j\omega T}) = P_{N-1}\left[x_\mu \cos(\omega T/2)\right] \tag{2.34}$$

where $P_n(x)$ is the Legendre polynomial. These polynomials can be calculated using the recurrence relationship

$$P_r(x) = \frac{1}{r}\Big[x(2r-1)P_{r-1}(x) - (r-1)P_{r-2}(x)\Big] \quad \text{for } r = 2, 3, ..., n \tag{2.35}$$

where $P_0(x) = 1$ and $P_1(x) = x$.

Figure 2.2 shows the normalized amplitude spectrum for ultraspherical windows with different values of the window length, shape parameter, and parameter μ. As can be seen, the shape parameter controls the null-to-null width while parameter μ controls the side-lobe pattern. As discussed in Section 1.4, the width parameters affect the resolution between adjacent signals while the side-lobe pattern affects the distribution of energy throughout the side lobes.

2.4 Efficient Formulation for Window Coefficients

A reduction in the computational complexity associated with windowing operations can be achieved by reducing the amount of computation required to generate the window coefficients. For the ultraspherical window, the primary computational bottleneck in Eq. (2.1) is due to the recursive evaluation of the binomial coefficients using Eq. (2.2). In its current

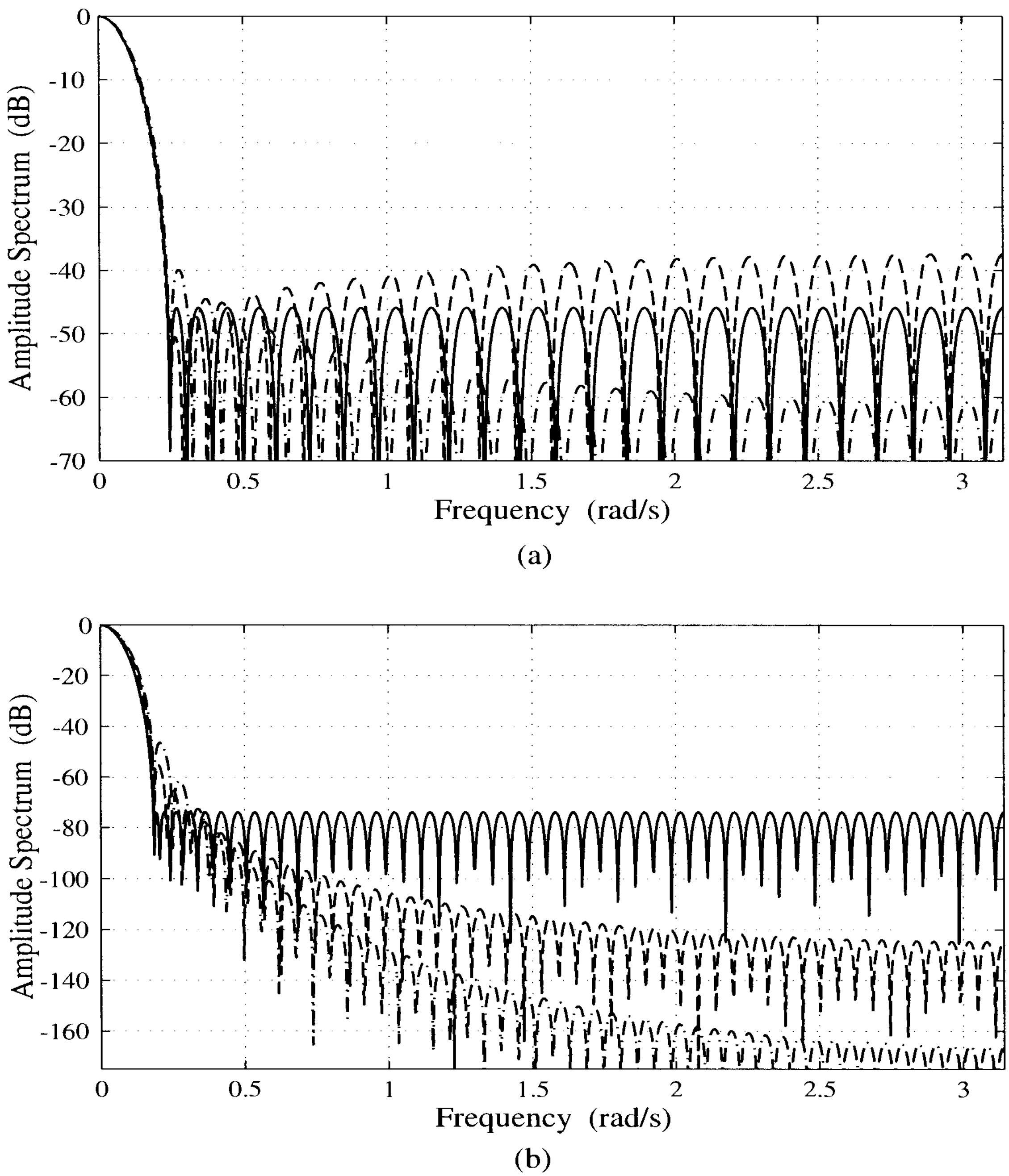


Figure 2.2. *Normalized amplitude spectrum for the ultraspherical window. (a) Length* $N = 51$ *designed with* $\beta = 2$ *and* $\mu = -0.5$ *(dashed),* 0 *(solid), and* 1 *(dashed-dotted). (b) Length* $N = 101$ *designed with* $\beta = 3$ *and* $\mu = 0$ *(solid),* 3 *(dashed), and* 6 *(dashed-dotted).*

form, Eq. (2.1) requires the evaluation of $(D+1)+\sum_{n=0}^{D}\sum_{m=0}^{n} 2 = D^2+4D+3$ binomial coefficients where D is given by Eq. (2.6). By exploiting certain redundancies in Eq. (2.1), the number of binomial-coefficient evaluations can be reduced quite significantly and the computational complexity associated with the ultraspherical window can be reduced. To begin with, the first binomial-coefficient expression in Eq. (2.1) can be expressed as

$$v_0(n) = \binom{\mu+p-n-1}{p-n-1} = \binom{\alpha_0-n}{p_0-n} \tag{2.36}$$

where $\alpha_0 = \mu + p - 1$ and $p_0 = p - 1$. Using the identity [20]

$$\binom{a}{b} = \frac{a!}{(a-b)!b!} \tag{2.37}$$

$v_0(n)$ can be represented as

$$\begin{aligned} v_0(n) &= \frac{(\alpha_0-n)!}{(\alpha_0-p_0)!(p_0-n)!} \\ &= \frac{p_0-n+1}{\alpha_0-n+1}\cdot\frac{p_0-n+2}{\alpha_0-n+2}\cdots\frac{p_0-1}{\alpha_0-1}\cdot\frac{p_0}{\alpha_0}\cdot\binom{\alpha_0}{p_0} \quad \text{for } n \geq 1 \end{aligned} \tag{2.38}$$

which leads to the recurrence relationships

$$v_0(0) = \binom{\alpha_0}{p_0}; \quad v_0(n) = \frac{p_0-n+1}{\alpha_0-n+1}v_0(n-1) \quad \text{for } n \geq 1 \tag{2.39}$$

In this formulation, the evaluation of one binomial coefficient replaces the evaluation of $D+1$ binomial coefficients thereby providing a savings of D binomial-coefficient evaluations.

Next, let us express the second binomial-coefficient expression in Eq. (2.1) as

$$v_1(n,m) = \binom{\mu+n-1}{n-m} = \binom{\alpha_1}{g} \tag{2.40}$$

where $\alpha_1 = \mu + n - 1$ and $g = n - m$. Observing that $v_1(n,n) = \binom{\alpha_1}{0} = 1$ and using the recursive identity [20]

$$\binom{a}{0} = 1; \quad \binom{a}{b} = \frac{a-b+1}{b}\binom{a}{b-1} \quad \text{for } b \geq 1 \tag{2.41}$$

$v_1(n,m)$ can be represented as

$$v_1(n,m) = \binom{\alpha_1}{g} = \frac{\alpha_1 - g + 1}{g} \cdot \frac{\alpha_1 - g + 2}{g-1} \cdots \frac{\alpha_1 - 1}{2} \cdot \alpha_1 \cdot \binom{\alpha_1}{0} \tag{2.42}$$

This analysis leads to the recurrence relationships

$$v_1(n,n) = 1; \quad v_1(n,m) = \frac{\mu + m}{n - m} v_1(n, m+1) \quad \text{for } 0 \le m < n \tag{2.43}$$

This formulation is equivalent to the evaluation of D binomial coefficients replacing the evaluation requirements of $\sum_{n=0}^{D} \sum_{m=0}^{n} 1 = \frac{1}{2}D^2 + \frac{3}{2}D + 1$ binomial coefficients, which would result in a savings of $\frac{1}{2}D^2 + \frac{1}{2}D + 1$ binomial-coefficient evaluations.

Finally, let us express the third binomial-coefficient expression in Eq. (2.1) as

$$v_2(n,m) = \binom{p-n}{m} = \binom{\alpha_2}{m} \tag{2.44}$$

where $\alpha_2 = p - n$. Observing that $v_2(n,0) = \binom{\alpha_2}{0} = 1$ and using the recursive identity in Eq. (2.41), $v_2(n,m)$ can be represented as

$$v_2(n,m) = \binom{\alpha_2}{m} = \frac{\alpha_2 - m + 1}{m} \cdot \frac{\alpha_2 - m + 2}{m-1} \cdots \frac{\alpha_2 - 1}{2} \cdot \alpha_2 \cdot \binom{\alpha_2}{0} \tag{2.45}$$

This leads to the recurrence relationships

$$v_2(n,0) = 1; \quad v_2(n,m) = \frac{\alpha_2 - m + 1}{m} v_2(n, m-1) \quad \text{for } 1 \le m \le n \tag{2.46}$$

This formulation is equivalent to the evaluation of D binomial coefficients replacing the evaluation of $\sum_{n=0}^{D} \sum_{m=0}^{n} 1 = \frac{1}{2}D^2 + \frac{3}{2}D + 1$ binomial coefficients, which provides a savings of $\frac{1}{2}D^2 + \frac{1}{2}D + 1$ binomial-coefficient evaluations.

Using the above expressions, the coefficients of the right-sided ultraspherical window of length N can be calculated using the formulation

$$w(nT) = \frac{A}{p-n} v_0(n) \cdot \sum_{m=0}^{n} v_1(n,m) v_2(n,m) B^m \quad \text{for } n = 0, 1, ..., N-1 \tag{2.47}$$

where $v_0(n)$, $v_1(n,m)$, and $v_2(n,m)$ are calculated using the recurrence relationships provided by Eqs. (2.39), (2.43), and (2.46), respectively, and A, B, and p are given by Eqs. (2.3), (2.4), and (2.5), respectively. This method requires the recursive evaluation of $2D+1$ binomial coefficients, which constitutes a computational complexity of $O(N)$ as compared with the evaluation of $D^2 + 4D + 3$ binomial coefficients required by Eq. (2.1), which constitutes a computational complexity of $O(N^2)$. In this way, an overall savings of D^2+2D+2 binomial-coefficient evaluations can be achieved. Figure 2.3 shows the computation time required to evaluate the coefficients of the ultraspherical window using Eqs. (2.1) and (2.47) vs. the window length. The computation time includes that required to calculate to largest zero of the ultraspherical polynomial $x_{N-1,1}^{(\mu)}$ using Algorithm 2.1 and was measured using the MATLAB stopwatch commands *tic* and *toc*. The time to compute the coefficients of the Kaiser window is included for comparison. The zeroth-order modified Bessel function of the first kind $I_0(x)$ was evaluated to an accuracy of $\epsilon = 10^{-10}$. The amount of computation of Eq. (2.47) is on the average 4.49% of that required by Eq. (2.1) and 9.27% of that required for the evaluation of the Kaiser window coefficients.

2.5 Conclusions

Two methods for evaluating the coefficients of the ultraspherical window were explored. The two methods yield the same coefficients for the same independent parameters μ, x_μ, and N. Economies in computation are achieved through an efficient formulation for the window coefficients which entails a computational complexity of $O(N)$ as compared with $O(N^2)$ for Streit's method. Alternatively, the amount of computation of the new formulation is on the average 4.49% of that required for Streit's method and 9.27% of that required for the evaluation of the Kaiser window coefficients. In addition, a method for setting the null-to-null width of the ultraspherical window to $4\beta\pi/N$, i.e., β times that of the rectangular window, was introduced. The chapter has also shown that the Dolph-Chebyshev and Saramäki windows are special cases of the ultraspherical window and can be obtained by setting $\mu = 0$ and 1, respectively.

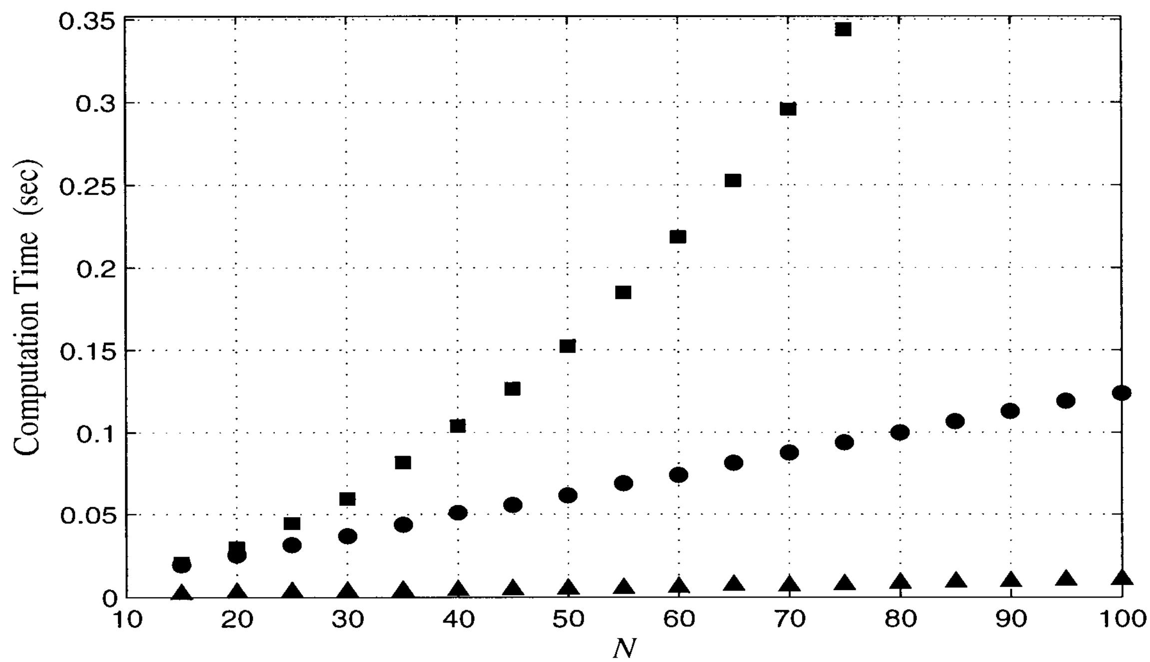


Figure 2.3. *Computation time associated with Eq. 2.1 (squares), Eq. 2.47 (triangles), and for the Kaiser window (circles) vs. the window length N.*

Chapter 3

Design of the Ultraspherical Window with Prescribed Spectral Characteristics

3.1 Introduction

Window selection has been a complicated task in the past due to the varied spectral characteristics required for different applications. For instance, if a signal contains a strong interference source whose frequency differs quite significantly from the frequency of interest, then a window with large side-lobe roll-off ratio, i.e., $s > 1$, should be considered. On the other hand, if a strong interference source is near the frequency of interest, then a window with a rather small ripple ratio and/or small side-lobe roll-off ratio, i.e., $s < 1$ is desirable. Further, for a sinusoidal interference source in which the focus is on amplitude accuracy rather than precise frequency location, a window with a wide main lobe is recommended. Window design methods should be flexible and should provide the designer the ability to tailor the window to account for varied application requirements.

Many of the available windows were obtained by exploiting certain characteristics of well-known polynomials and special functions to satisfy a particular criterion best. For instance, the Kaiser and Saramäki windows employ modified Bessel functions and Chebyshev polynomials of the second kind, respectively, and are close approximations to discrete prolate functions that have maximum energy concentration in the main lobe. The Dolph-Chebyshev window employs Chebyshev polynomials of the first kind and produces the minimum main-lobe width for a specified maximum side-lobe level. These two-parameter windows can control the main-lobe width and the ripple ratio but cannot control the pattern of the side lobes. The three-parameter ultraspherical window can control the main-lobe width and the ripple ratio as well as the side-lobe pattern.

In this chapter, a method is proposed for selecting the parameters of the ultraspherical window so as to achieve prescribed spectral characteristics such as ripple ratio, main-lobe width, null-to-null width, and side-lobe roll-off ratio. The chapter is structured as follows. In Section 3.2 a method for designing windows that satisfy prescribed spectral characteristics is proposed. The method entails a variety of short algorithms that can be used to determine two of the three independent parameters based on the prescribed spectral characteristics. In Section 3.3 an empirical formula that can be used to accurately predict the window length (the third parameter) required so as to achieve multiple prescribed spectral characteristics simultaneously is proposed. In Section 3.4 the ultraspherical window's effectiveness in achieving prescribed spectral characteristics is compared with respect to that in other windows. Section 3.5 presents examples and demonstrates the accuracy of the proposed method. Section 3.6 describes two applications of the proposed method in the areas of beamforming and image processing.

3.2 Prescribed Spectral Characteristics

With the appropriate selection of the parameters μ, x_μ, and N, the ultraspherical window can be designed so as to achieve prescribed specifications for the side-lobe roll-off ratio, the ripple ratio, and one of the two width characteristics simultaneously. Parameter μ alters the side-lobe roll-off ratio, x_μ provides a trade-off between the ripple ratio and a width characteristic, and N allows different ripple ratios to be obtained for a fixed width characteristic and vice versa. In some applications the window length N may be fixed. Such a scenario limits the designer's choice in achieving prescribed specifications for the side-lobe roll-off ratio and either the ripple ratio or a width characteristic but not both.

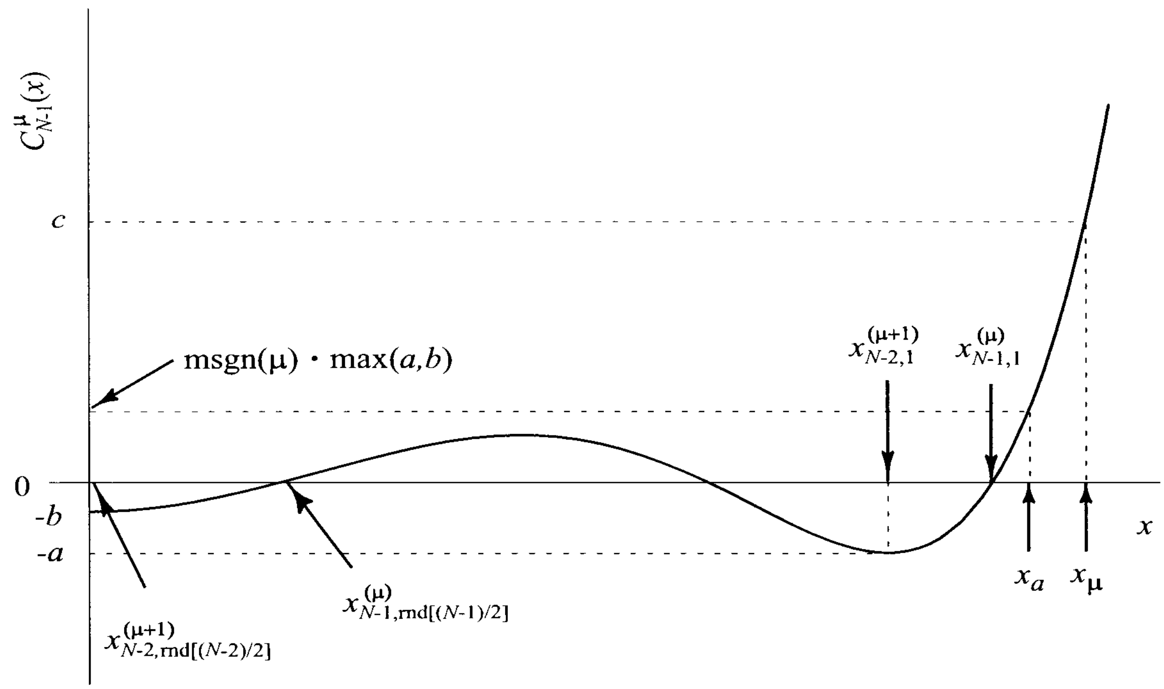


Figure 3.1. *Some important quantities of the ultraspherical polynomial $C^{\mu}_{N-1}(x)$ for the values $\mu = 2$ and $N = 7$.*

For the case where N is adjustable, a prediction of N is possible which allows one to achieve prescribed specifications for the side-lobe roll-off ratio, the ripple ratio and a width characteristic simultaneously.

In the subsections that follow, algorithms are proposed that enable one to achieve each prescribed specification to a high degree of precision. Some important quantities to be used are identified in Fig. 3.1 which depicts a plot of $C^{\mu}_{N-1}(x)$ for the values $\mu = 2$ and $N = 7$. The modified sign (msgn) and max functions are defined as

$$\mathrm{msgn}(x) = \begin{cases} -1 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

$$\max(x, y) = \begin{cases} x & \text{for } x \geq y \\ y & \text{for } y > x \end{cases}$$

3.2.1 Side-lobe roll-off ratio

To generate a window for a fixed N and a prescribed side-lobe roll-off ratio s, one can select the parameter μ appropriately. This can be accomplished by solving the one-dimensional minimization problem

$$\underset{\mu_L \leq \mu \leq \mu_H}{\text{minimize}} \ F = \left[s - \left| \frac{C_{N-1}^{\mu}\left(x_{N-2,1}^{(\mu+1)}\right)}{C_{N-1}^{\mu}\left(x_{N-2,\text{rnd}[(N-2)/2]}^{(\mu+1)}\right)} \right| \right]^2 \tag{3.1}$$

where the values of $C_n^{\mu}(x)$ are given by Eq. (2.14) and $x_{N-2,1}^{(\mu+1)}$ and $x_{N-2,\text{rnd}[(N-2)/2]}^{(\mu+1)}$, which are identified in Fig. 3.1, are the largest and smallest zeros, respectively, of the derivative of $C_{N-1}^{\mu}(x)$, namely, $2\mu C_{N-2}^{\mu+1}(x)$. The zero $x_{N-2,1}^{(\mu+1)}$ can be found using Algorithm 2.1 with $l = 1$, $\lambda = \mu + 1$, $n = N - 2$, and $\varepsilon = 10^{-6}$. The zero $x_{N-2,\text{rnd}[(N-2)/2]}^{(\mu+1)}$ can be found using Algorithm 2.1 with $l = \text{rnd}[(N-2)/2]$, $\lambda = \mu + 1$, $n = N - 2$, and $\varepsilon = 10^{-6}$.

Simple algorithms such as dichotomous, Fibonacci, or golden section line searches as outlined in [27] can be used to perform the minimization in Eq. (3.1). The lower and upper bounds on μ in Eq. (3.1) can be set to

$$\begin{aligned} &\mu_L = 0 \text{ and } \mu_H = 10 && \text{for } s > 1 \\ &\mu_L = -0.9999 \text{ and } \mu_H = 0 && \text{for } 0 < s < 1 \end{aligned} \tag{3.2}$$

If $s = 1$, then no minimization is necessary and $\mu = 0$ yields the Dolph-Chebyshev window. The bound $\mu_L = -0.9999$ was chosen because $C_{N-1}^{\mu}(x)$ has a singularity at $\mu = -1$. Also, for values of $\mu \leq -1.5$, the zero $x_{n,1}^{(\mu)}$ coincides with the zero $x_{n,2}^{(\mu)}$ rendering the resulting window useless for our purposes. The bound $\mu_H = 10$ was chosen because the improvements in the side-lobe roll-off ratio that can be achieved for values of $\mu > 10$ are negligible.

The ultraspherical window imposes limits on the side-lobe roll-off ratio that can be achieved for low values of N. For example, if $N = 7$, window designs with $S = 20 \log_{10} s$ outside the range $-10.19 < S < 12.78$ dB are not possible for any value of μ. For this reason, the side-lobe roll-off ratio's design range must be limited for a given N to that

Table 3.1. *Limiting Side-Lobe Roll-Off Ratios for Small Values of N*

N	min S (dB)	max S (dB)
5	−6.02	4.95
6	−7.65	7.88
7	−10.19	12.78
8	−11.43	16.25
9	−13.05	20.82
10	−14.02	24.32
11	−15.20	28.55
12	−16.00	31.93
13	−16.93	35.83
14	−17.61	39.05
15	−18.37	42.67
16	−18.96	45.72
17	−19.61	49.07
18	−20.13	51.96
19	−20.69	55.08
20	−21.15	57.81

produced using $\mu_L = -0.9999$ and $\mu_H = 10$. The limiting values are shown in Table 3.1 for window lengths in the range $5 \leq N \leq 20$ which spans the practical design range $-20 \leq S \leq 60$ dB.

3.2.2 Null-to-null width

To generate a window with μ and N fixed and a prescribed null-to-null half width of ω_n rad/s, one can select the parameter x_μ appropriately. This can be accomplished by calculating x_μ using the expression

$$x_\mu = \frac{x^{(\mu)}_{N-1,1}}{\cos(\omega_n/2)} \tag{3.3}$$

where the zero $x^{(\mu)}_{N-1,1}$ can be found using Algorithm 2.1 with $l = 1$, $\lambda = \mu$, $n = N - 1$, and $\varepsilon = 10^{-6}$.

3.2.3 Main-lobe width

To generate a window with μ and N fixed and a prescribed main-lobe half width of ω_r rad/s, one can select the parameter x_μ appropriately. This can be accomplished by calculating x_μ using the expression

$$x_\mu = \frac{x_a}{\cos(\omega_r/2)} \tag{3.4}$$

where x_a is defined by $C^\mu_{N-1}(x_a) = \text{msgn}(\mu)\cdot\max(a,b)$ as identified in Fig. 3.1. Parameter x_a is found through a three-step process. First, the zero $x^{(\mu+1)}_{N-2,1}$ is found using Algorithm 2.1 with $l = 1$, $\lambda = \mu + 1$, $n = N - 2$, and $\varepsilon = 10^{-6}$, and then the parameter $a = \left|C^\mu_{N-1}\left(x^{(\mu+1)}_{N-2,1}\right)\right|$ is calculated. Second, the zero $x^{(\mu+1)}_{N-2,\text{rnd}[(N-2)/2]}$ is found using Algorithm 2.1 with $l = \text{rnd}[(N-2)/2]$, $\lambda = \mu + 1$, $n = N - 2$, and $\varepsilon = 10^{-6}$ and then the parameter $b = \left|C^\mu_{N-1}\left(x^{(\mu+1)}_{N-2,\text{rnd}[(N-2)/2]}\right)\right|$ is calculated. Third, since $\text{msgn}(\mu)\cdot\max(a,b) = C^\mu_{N-1}(x_a)$ as seen in Fig. 3.1, parameter x_a is found using a modified version of Algorithm 2.1 where Eq. (2.25) is replaced by

$$y_{k+1} = y_k - \frac{C^\lambda_n(y_k) - \text{msgn}(\mu)\cdot\max(a,b)}{2\lambda C^{\lambda+1}_{n-1}(y_k)} \tag{3.5}$$

and the starting point given in Eq. (2.24) is replaced by $y_1 = 1$. Instead of finding the largest zero of $f(x) = C^\mu_n(x)$, the modified algorithm finds the largest zero of $f(x) = C^\mu_n(x) - \text{msgn}(\mu)\cdot\max(a,b)$, which is parameter x_a. In the modified algorithm, $l = 1$, $\lambda = \mu$, $n = N - 1$, and $\varepsilon = 10^{-6}$.

3.2.4 Ripple ratio

To generate a window with μ and N fixed and a prescribed ripple ratio r, one can select the parameter x_μ appropriately. The parameter x_μ is found through a three-step process. First, the zero $x_{N-2,1}^{(\mu+1)}$ is found using Algorithm 2.1 with $l = 1$, $\lambda = \mu + 1$, $n = N - 2$, and $\varepsilon = 10^{-6}$ and then the parameter $a = \left|C_{N-1}^{\mu}\left(x_{N-2,1}^{(\mu+1)}\right)\right|$ is calculated. Second, the zero $x_{N-2,\text{rnd}[(N-2)/2]}^{(\mu+1)}$ is found using Algorithm 2.1 with $l = \text{rnd}[(N-2)/2]$, $\lambda = \mu + 1$, $n = N - 2$, and $\varepsilon = 10^{-6}$ and then the parameter $b = \left|C_{N-1}^{\mu}\left(x_{N-2,\text{rnd}[(N-2)/2]}^{(\mu+1)}\right)\right|$ is calculated. Third, the parameter x_μ is found using a modified version of Algorithm 2.1 where Eq. (2.25) is replaced by

$$y_{k+1} = y_k - \frac{C_n^{\lambda}(y_k) - \text{msgn}(\mu) \cdot \max(a,b)/r}{2\lambda C_{n-1}^{\lambda+1}(y_k)} \tag{3.6}$$

and the starting point given in Eq. (2.24) is replaced by

$$y_1 = \cosh\left[\frac{1}{N-1}\cosh^{-1}\left(\frac{1}{r}\right)\right] \tag{3.7}$$

Instead of finding the largest zero of $f(x) = C_n^{\mu}(x)$, the modified algorithm finds the largest zero of $f(x) = C_n^{\mu}(x) - \text{msgn}(\mu) \cdot \max(a,b)/r$ which is the parameter x_μ. In the modified algorithm $l = 1$, $\lambda = \mu$, $n = N - 1$, and $\varepsilon = 10^{-6}$.

3.3 Prediction of N

In some applications designers may be able to choose the window length N. In such applications, the extra degree of freedom allows for more flexible window designs to be obtained. Specifically, solutions that are required to meet both a prescribed ripple ratio and width characteristic are possible. In this section, an empirical equation is proposed that predicts the window length N required to yield a prescribed side-lobe roll-off ratio, ripple ratio, and main-lobe width simultaneously.

To obtain an equation for N, we employ the performance factor [28]

$$D = 2\omega_r(N-1) \tag{3.8}$$

which is used to give a normalized width that is approximately independent of N. Rearranging Eq. (3.8), an expression for N is obtained as

$$N \geq D/2\omega_r + 1 \tag{3.9}$$

where N is rounded up to the nearest integer. From Eq. (3.9), it becomes clear that N can be predicted by obtaining an accurate approximation of D.

3.3.1 Measurements and tendencies of D

To obtain realistic data for the approximation of D, windows of length $N = 7$, 9, 13, 19, 51, 127, and 255 were designed to cover the range $20 \leq R \leq 100$ in dB for the parameter range $-0.9999 \leq \mu \leq 10$. Figure 3.2 shows plots of D vs. R in dB for the two cross sections $\mu = 1$ and 10. The plots tend to be quadratic and are representative for the range $-0.9999 \leq \mu \leq 10$ considered in this work. Note the approximately linear behavior for $N = 255$ indicating the independence of the performance factor D with respect to N for large N, which agrees with previous observations concerning the performance factor D [28].

3.3.2 Data-fitting procedure

Before approximating D, the allowable error in the data-fitting procedure must be determined. From Eq. (3.9), we note that for $N >> 1$ a per unit error in D gives approximately the same per unit error in N, i.e.,

$$\frac{\Delta D}{D} = \frac{\Delta(N-1)}{N-1} = \frac{\Delta N}{N-1} \approx \frac{\Delta N}{N}. \tag{3.10}$$

For example, if $N = 127$ and a relative error in D of 1.00 percent is assumed, that is, $\Delta D/D = 0.01$, then an equivalent error of 1.26 samples in N occurs. Errors of this magnitude have been considered acceptable in the past [28] as N may be in error by at most 1 or 2 and only for high window lengths. Thus, the relative error $\Delta D/D \leq 0.01$ is sought throughout the approximation procedure.

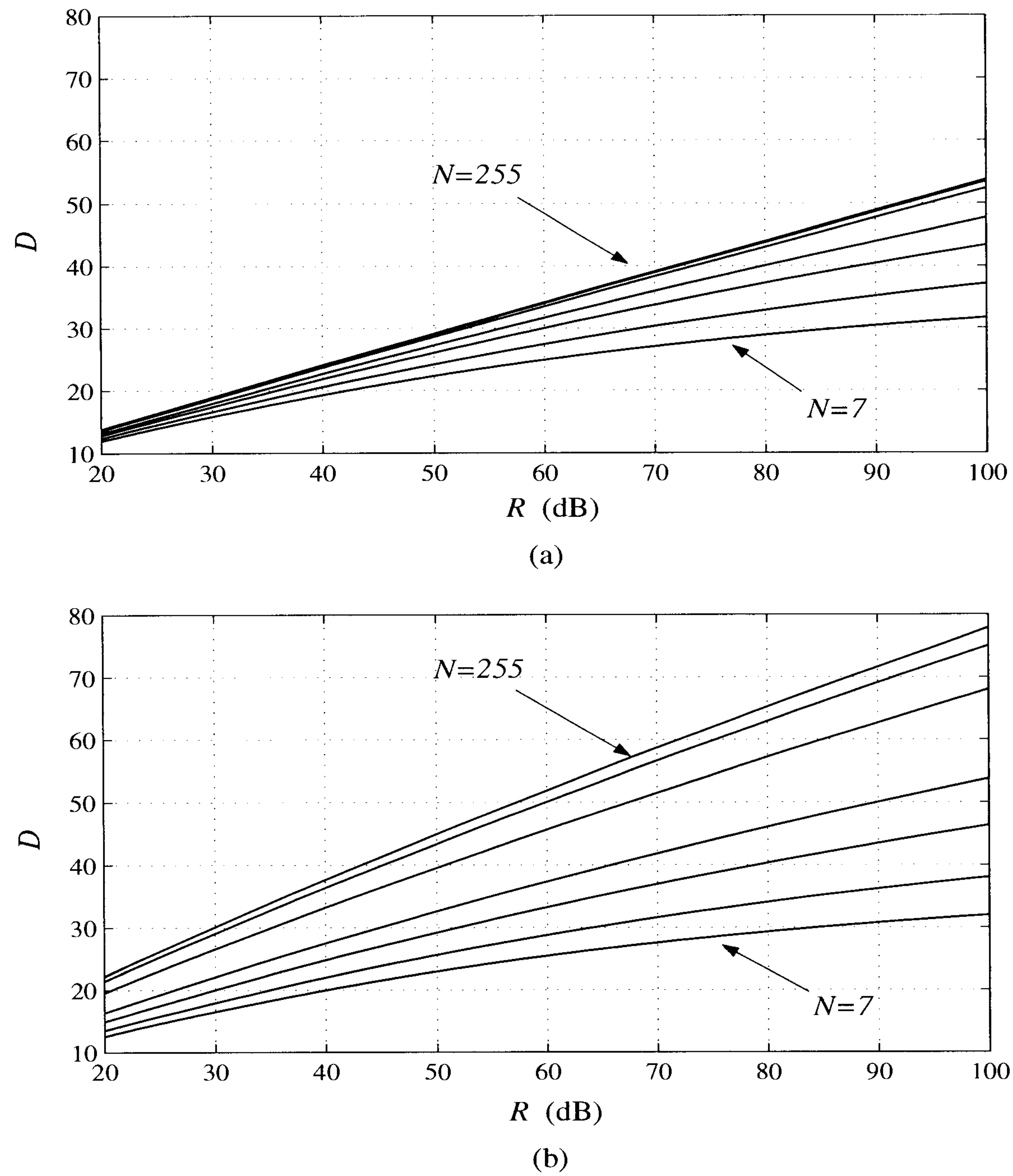


Figure 3.2. *Performance factor D vs. R in dB for windows of length $N = 7, 9, 13, 19, 51,$ $127,$ and 255 for values of (a) $\mu = 1$ and (b) $\mu = 10$.*

A general quadratic model was used for the approximation of D as a function of S in dB, R in dB, and the main-lobe half width ω_r. Such a model takes the form

$$D_{aprx}(S, R, \omega_r) = \sum_{i=0}^{2}\sum_{j=0}^{2}\sum_{k=0}^{2} a_{ijk}\phi(i,j,k) \tag{3.11}$$

where $\phi(i,j,k) = (S/20)^i R^j \omega_r^k$. The coefficients a_{ijk} were found through a linear least-squares solution of the overdetermined system of sampled data points $\{S, R, \omega_r, D\}$ where D is the dependent variable.

Two separate sets of 27 coefficients were found for the ranges $0 \leq S \leq 60$ and $-20 \leq S \leq 0$ given in dB and are provided in Tables 3.2 and 3.3, respectively. Two sets were required to produce accurate solutions due the nature of D and its relation to positive and negative S values. Figure 3.3 shows plots of the relative error of the predicted D vs. R for various window lengths over the cross sections $\mu = 1$ and -0.6. The mean of the absolute relative error for the approximations given by Tables 3.2 and 3.3 is 0.2874 and 0.2266 percent, respectively. Less error occurs for the coefficients in Table 3.3 because the approximation was performed over a smaller range of S than that used for Table 3.2. The absolute relative error exceeds 1.0 percent only for small values of R less than 20 and large values of R greater than 100.

In an attempt to reduce the number of approximation model coefficients, the quadratic model

$$D_{aprx}(S, R, \omega_r) = \sum_{i=0}\sum_{j=0}\sum_{k=0}^{l} a_{ijk}\phi(i,j,k) \tag{3.12}$$

where

$$l = i + j + k \leq 2 \tag{3.13}$$

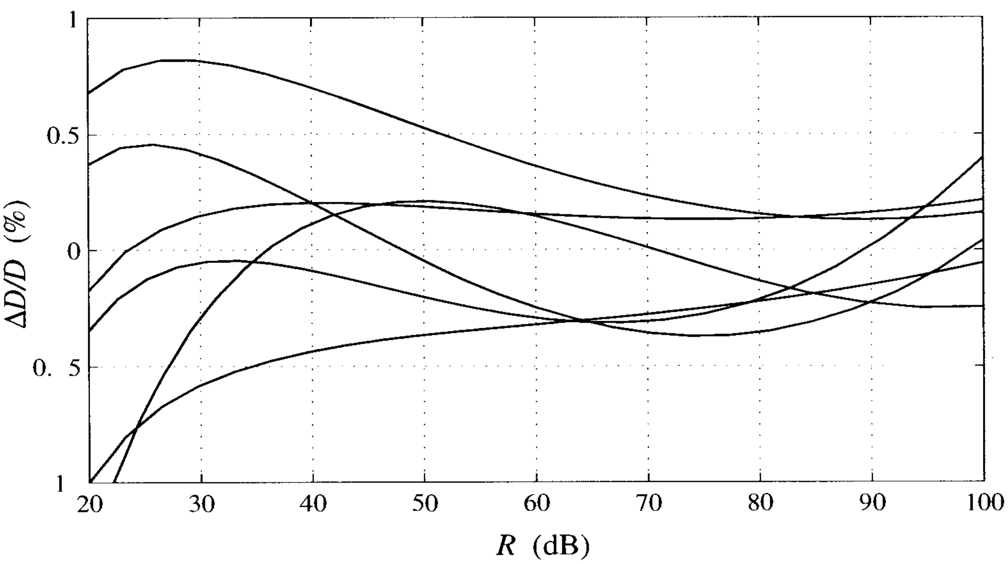
was investigated which yields 10 coefficients as opposed to 27. Using the same data fitting technique as before, the mean of the absolute relative error for the entire approximation was found to be 1.0911 percent. In 70 percent of the predictions, the absolute error was less than 1.0 percent.

Table 3.2. *Model Coefficients a_{ijk} in Eq. 3.11* $(S > 0)$

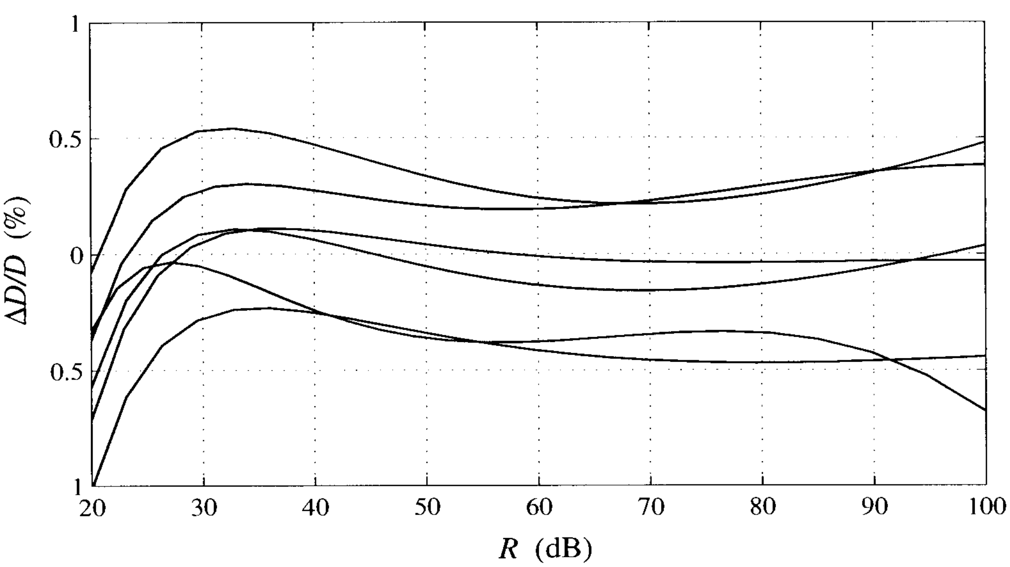
i	j	$k = 0$	$k = 1$	$k = 2$
0	0	2.699E+0	1.824E-1	−1.125E-1
	1	4.650E-1	−1.450E-2	−1.607E-2
	2	−6.273E-5	2.681E-4	−1.263E-4
1	0	2.657E-2	8.293E-2	−6.312E-2
	1	1.719E-3	1.846E-3	7.488E-5
	2	−4.610E-6	−1.801E-5	2.406E-6
2	0	−7.012E-5	3.882E-4	−1.703E-3
	1	−5.568E-6	7.549E-6	1.153E-5
	2	2.451E-8	−6.588E-8	1.139E-8

Table 3.3. *Model Coefficients a_{ijk} in Eq. 3.11* $(S < 0)$

i	j	$k = 0$	$k = 1$	$k = 2$
0	0	2.700E-0	1.699E-1	−1.126E-1
	1	4.648E-1	−1.321E-2	−1.646E-2
	2	−6.200E-5	2.593E-4	−1.230E-4
1	0	−2.214E-1	1.095E-1	−5.410E-2
	1	−2.066E-3	1.183E-3	5.045E-4
	2	1.723E-5	−1.617E-5	1.242E-6
2	0	−2.016E-3	−6.856E-3	5.755E-3
	1	−1.646E-5	1.248E-4	−9.390E-5
	2	3.492E-7	−1.409E-6	8.638E-7



(a)



(b)

Figure 3.3. *Relative error of predicted* D, $\Delta D/D$, *in percent vs.* R *in dB for window lengths* $N = 7, 9, 13, 19, 51, 127,$ *and* 255 *over the cross sections (a)* $\mu = 1$ *and (b)* $\mu = -0.6$.

On the basis of the above experiments, N can be accurately predicted using the formula

$$N = \text{int}\left[\frac{D_{aprx}(S, R, \omega_r)}{2\omega_r} + 1.5\right] \tag{3.14}$$

where D_{aprx} is given by the 27-term approximation model described in Eq. (3.11) using the coefficients provided in Tables 3.2 and 3.3.

The same process can be used to predict N for other width characteristics such as the null-to-null or 3-dB widths.

3.4 Comparison With Other Windows

For a fixed window length, two-parameter windows such as the Kaiser, Saramäki, and Dolph-Chebyshev windows can control the ripple ratio. The three-parameter ultraspherical window can control the ripple ratio as well as the side-lobe roll-off ratio. For comparison's sake, the ultraspherical window of the same length was designed to yield the side-lobe roll-off ratio and main-lobe width produced by the Kaiser window, for values of the Kaiser-window parameter α in the range $[1, 10]$, and the resulting ripple ratios for the two window families were measured and compared. The Dolph-Chebyshev and Saramäki windows were excluded from the comparison because these windows are special cases of the ultraspherical window that can be readily obtained by fixing parameter μ to 0 and 1, respectively. Figure 3.4a shows plots of the side-lobe roll-off ratio in dB obtained for the Kaiser window for varying length vs. $D = 2\omega_r(N-1)$ and Fig. 3.4b shows a plot of ΔR which is defined as

$$\Delta R = R_U - R_K \tag{3.15}$$

where R_U and R_K are the values of R for the ultraspherical and Kaiser windows, respectively, in dB for the same length, side roll-off ratio, and main-lobe width. As can be seen, the ultraspherical window offers a reduced ripple ratio for low values of D whereas the Kaiser window gives better results for large values of D. Thus for a given value of N, there is a corresponding main-lobe half width, say, ω_{rU}, for which the ultraspherical window

Table 3.4. *Model Coefficients for ω_{rU} in Eq. 3.16*

N_L	N_H	a	b	c	d
10	25	-1.149E-4	7.855E-3	-1.935E-1	2.238E+0
25	80	-1.495E-6	3.208E-4	-2.554E-2	9.692E-1
80	250	-2.520E-8	1.679E-5	-4.096E-3	4.451E-1

gives a better ripple ratio than the Kaiser window. For main-lobe half widths that are larger than ω_{rU}, the Kaiser window gives a smaller ripple ratio. A plot of ω_{rU} versus N is shown in Fig. 3.5. From this plot, a formula can be obtained for ω_{rU} as

$$\omega_{rU} = aN^3 + bN^2 + cN + d \quad \text{for } N_L \leq N \leq N_H \tag{3.16}$$

where the coefficients are presented in Table 3.4. In effect, if the point $[N, \omega_r]$ is located below the curve in Fig. 3.5, the ultraspherical window is preferred and if it is located above the curve, the Kaiser window is preferred.

3.5 Examples

Example 1: For $N = 51$, generate the windows that will yield $S = 20$ dB for (a) $\omega_r = 0.25$ rad/s and (b) $\omega_n = 0.25$ rad/s.

Figure 3.6 shows the amplitude spectrums of the windows obtained. Both designs meet the prescribed specifications and produced (a) $R = 42.97$ dB and (b) $R = 40.85$ dB. For both designs, the minimization of Eq. (3.1) resulted in $\mu = 0.9517$ and Eqs. (3.4) and (3.3) gave (a) $x_\mu = 1.0067$ and (b) $x_\mu = 1.0060$, respectively.

Example 2: For $N = 51$, generate the windows that will yield $R = 50$ dB for (a) $S = -10$ dB and (b) $S = 30$ dB.

Figure 3.7 shows the amplitude spectrums of the windows obtained. Both designs meet the prescribed specifications and produced main-lobe widths of (a) $\omega_r = 0.2783$ rad/s and (b) $\omega_r = 0.2975$ rad/s. Minimizing Eq. (3.1) resulted in (a) $\mu = -0.3914$ and (b)

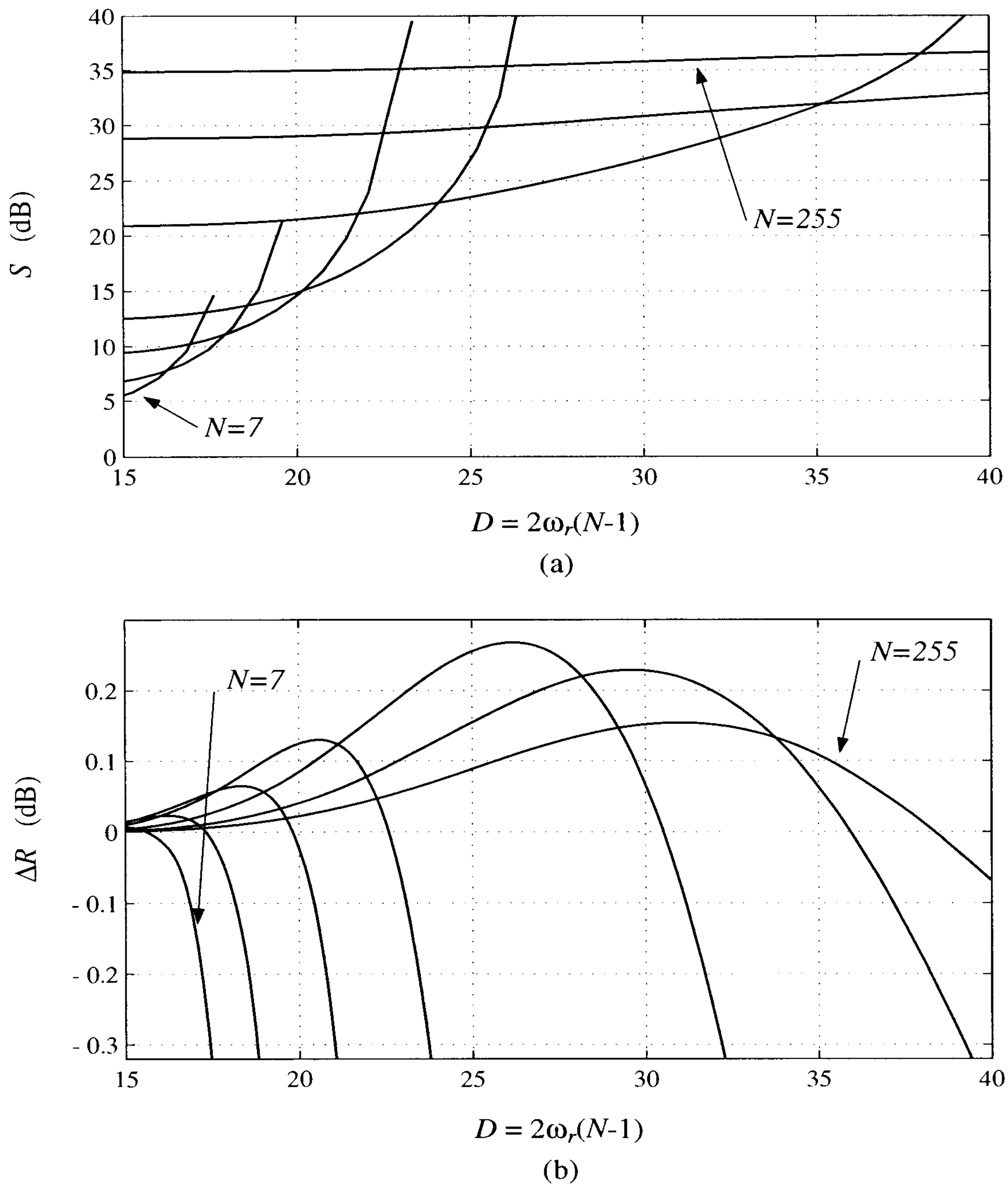


Figure 3.4. *(a) Side-lobe roll-off ratio in dB for Kaiser windows of length* $N = 7, 9, 13, 19, 51, 127,$ *and* 255. *(b) Change in the ripple ratio in dB provided by ultraspherical windows of the same length that were designed to match the Kaiser windows' side-lobe roll-off ratio and main-lobe width.*

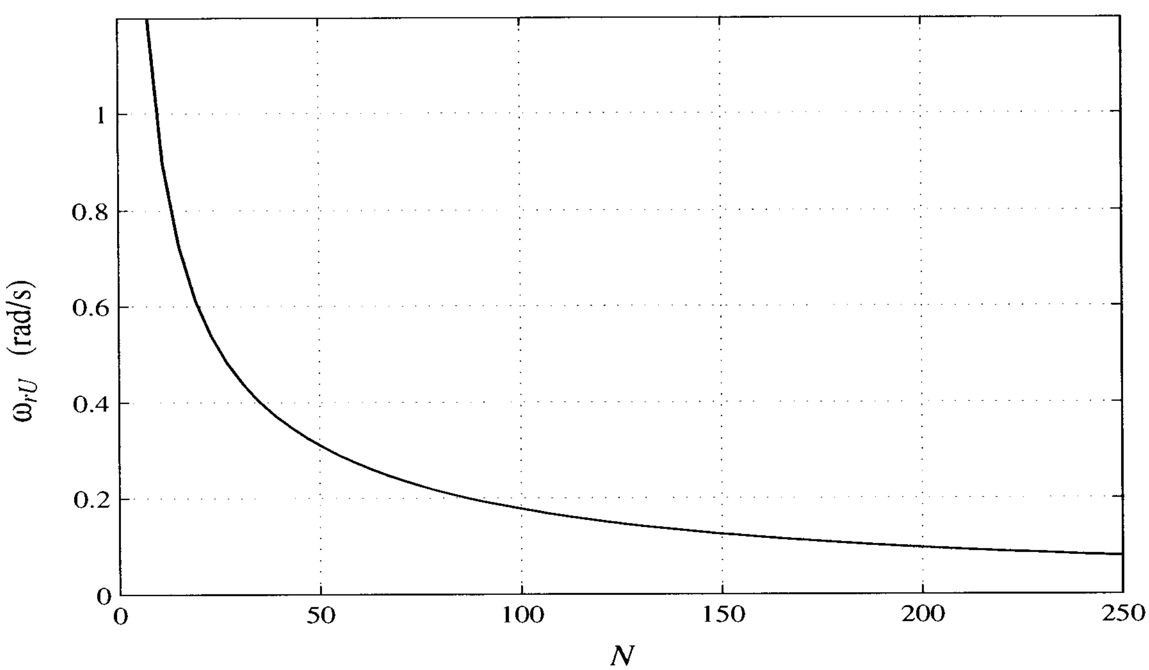


Figure 3.5. *Values of the main-lobe half width that achieve the same ripple ratio for both the Kaiser and ultraspherical windows.*

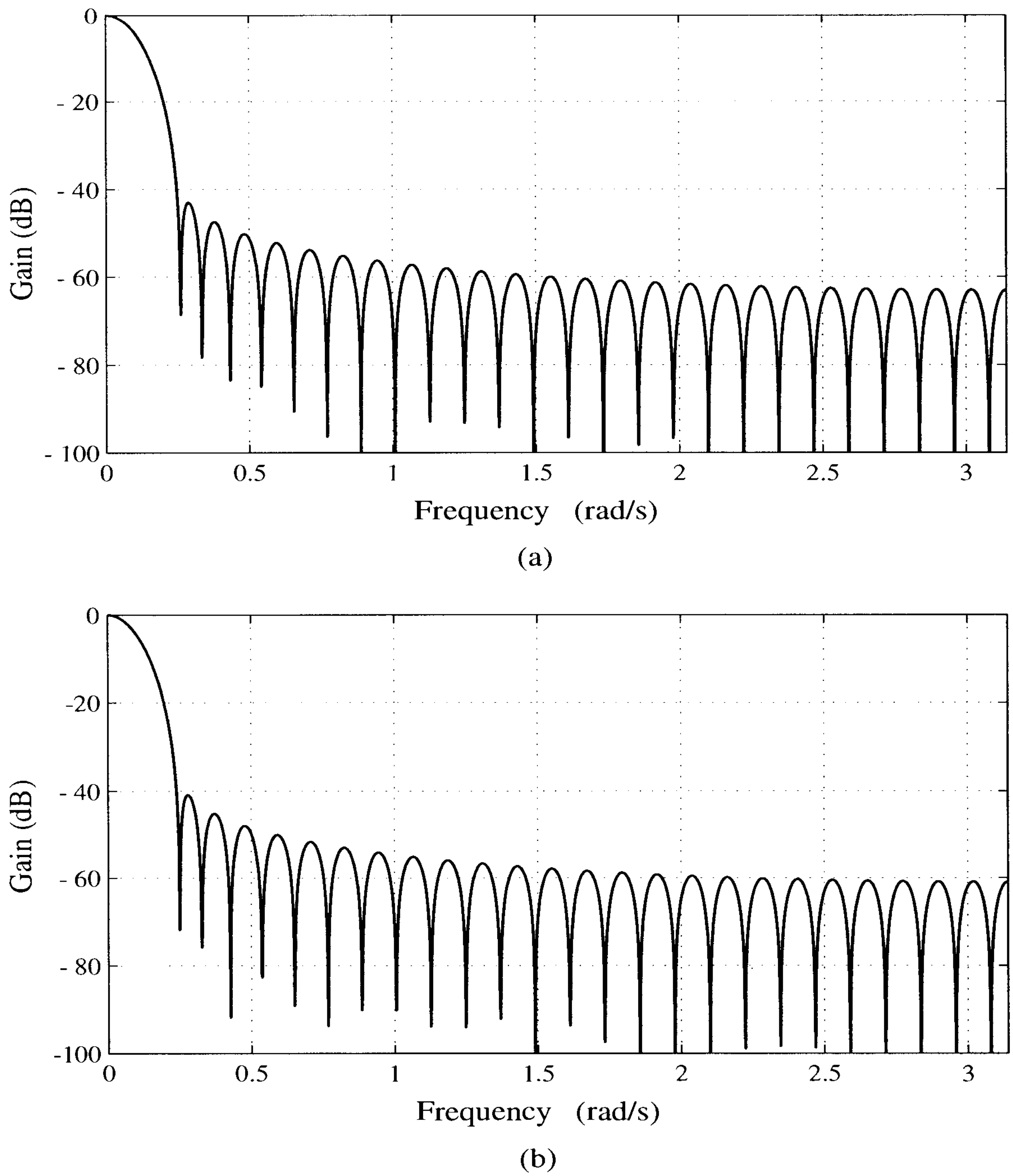


Figure 3.6. *Ultraspherical window amplitude spectrums for* $N = 51$ *yielding* $S = 20$ *dB for (a)* $\omega_r = 0.25$ *rad/s and (b)* $\omega_n = 0.25$ *rad/s (Example 1).*

$\mu = 1.5151$ and the procedure described in Section 3.2.4 gave (a) $x_\mu = 1.0107$ and (b) $x_\mu = 1.0091$.

Example 3: Predict the required window length N and generate the ultraspherical window that will yield $\omega_r = 0.2$ rad/s and $R \geq 60$ dB for (a) $S = 10$ dB and (b) $S = -10$ dB.

A consequence of rounding N up to the nearest integer is that one prescribed spectral characteristic is oversatisfied. For the method presented in this chapter, one will always achieve S and either ω_r or R to a high degree of precision by using either Eq. (3.4) or the procedure described in Section 3.2.4 as appropriate to calculate parameter x_μ. In this example, we oversatisfy R by using Eq. (3.4). Figure 3.8 shows the amplitude spectrums of the windows obtained. Both designs meet the prescribed characteristics and oversatisfied R by (a) 0.47 dB and (b) 0.41 dB. Using the prediction formula given in Eq. (3.14), the window lengths required to yield the prescribed characteristics were (a) $N = 81$ and (b) $N = 83$. Minimizing Eq. (3.1) resulted in (a) $\mu = 0.3756$ and (b) $\mu = -0.3378$ and Eq. (3.4) gave (a) $x_\mu = 1.0049$ and (b) $x_\mu = 1.0053$.

To examine the accuracy of the window length prediction formula, windows were designed to yield the same prescribed characteristics with window lengths taken to be one less than predicted by Eq. (3.14), i.e., for (a) $N - 1 = 80$ and (b) $N - 1 = 82$. Figure 3.9 shows the amplitude spectrums obtained for N and $N - 1$ in the critical area near the main-lobe edge. All windows were found to satisfy the S and ω_r specifications; however, both windows of the reduced length fell short of $R \geq 60$ dB by (a) 0.35 dB and (b) 0.51 dB. The results demonstrate the accuracy of Eq. (3.14) in predicting the lowest value of N needed to yield the set of prescribed spectral characteristics simultaneously.

Example 4: For $N = 101$, generate Kaiser and ultraspherical windows that will yield (a) $R = 50$ dB and (b) $R = 70$ dB and compare the results obtained.

The required Kaiser-window parameter α for (a) and (b) can be predicted using the

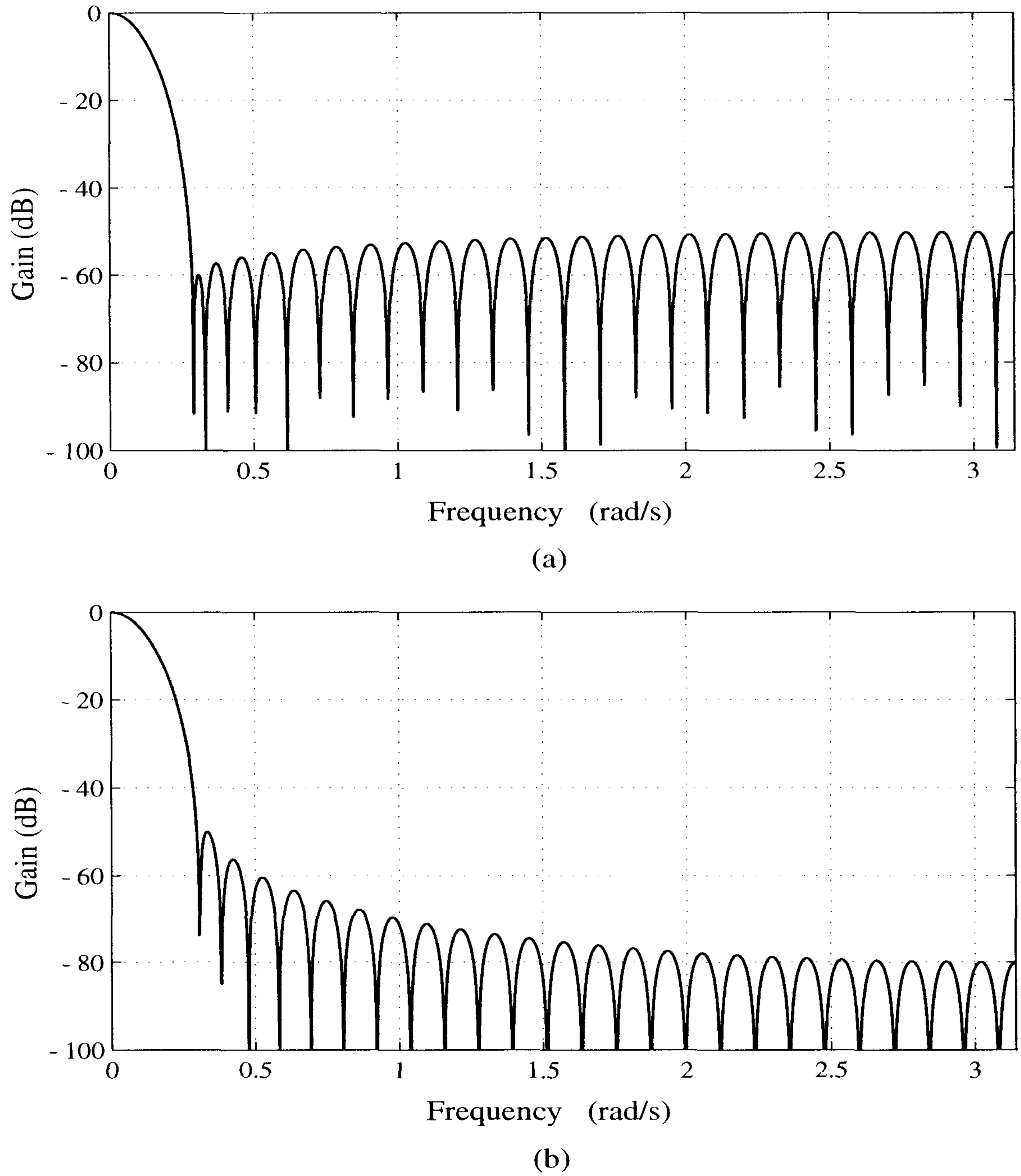
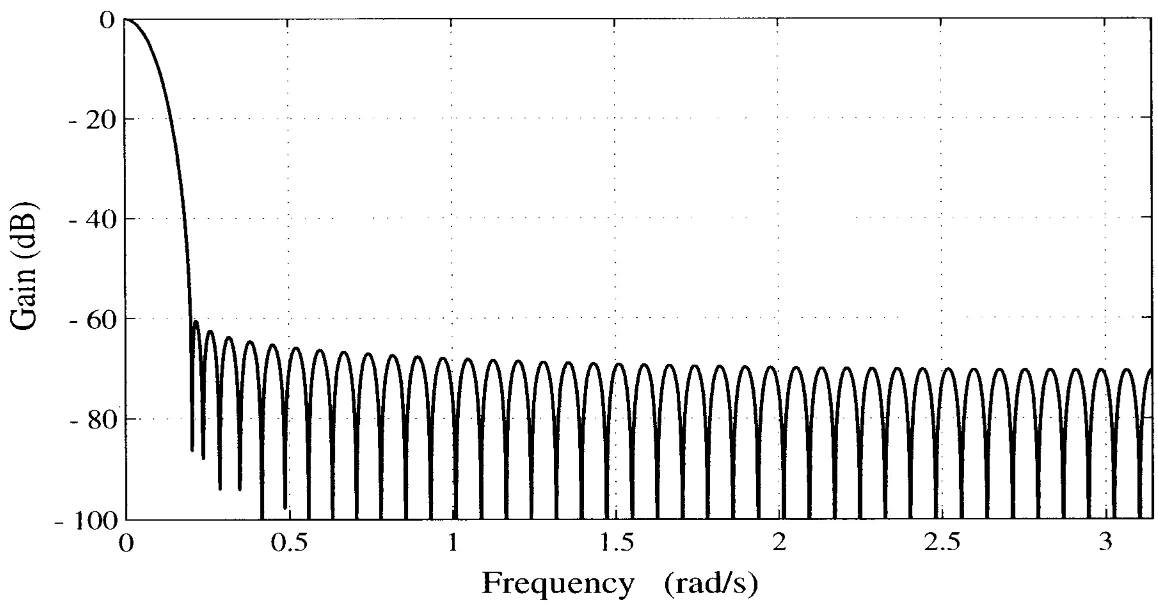
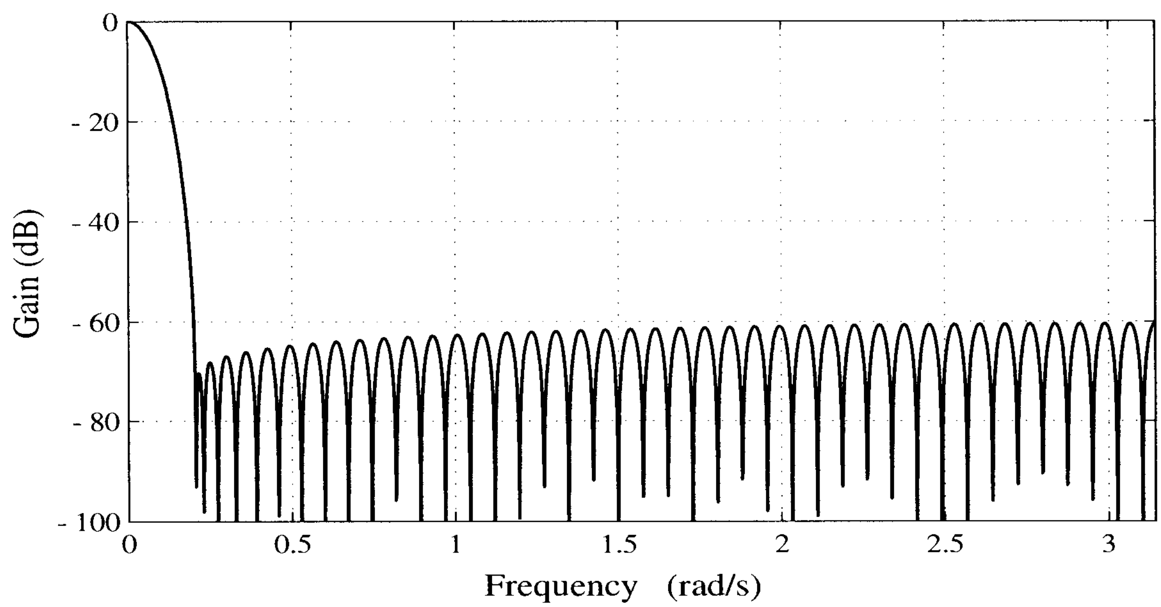


Figure 3.7. *Ultraspherical window amplitude spectrums for $N = 51$ yielding $R = 50$ dB for (a) $S = -10$ dB and (b) $S = 30$ dB (Example 2).*



(a)



(b)

Figure 3.8. *Ultraspherical window amplitude spectrums yielding* $\omega_R = 0.2$ *rad/s and* $R \geq 60$ *dB for (a)* $S = 10$ *dB and (b)* $S = -10$ *dB (Example 3a).*

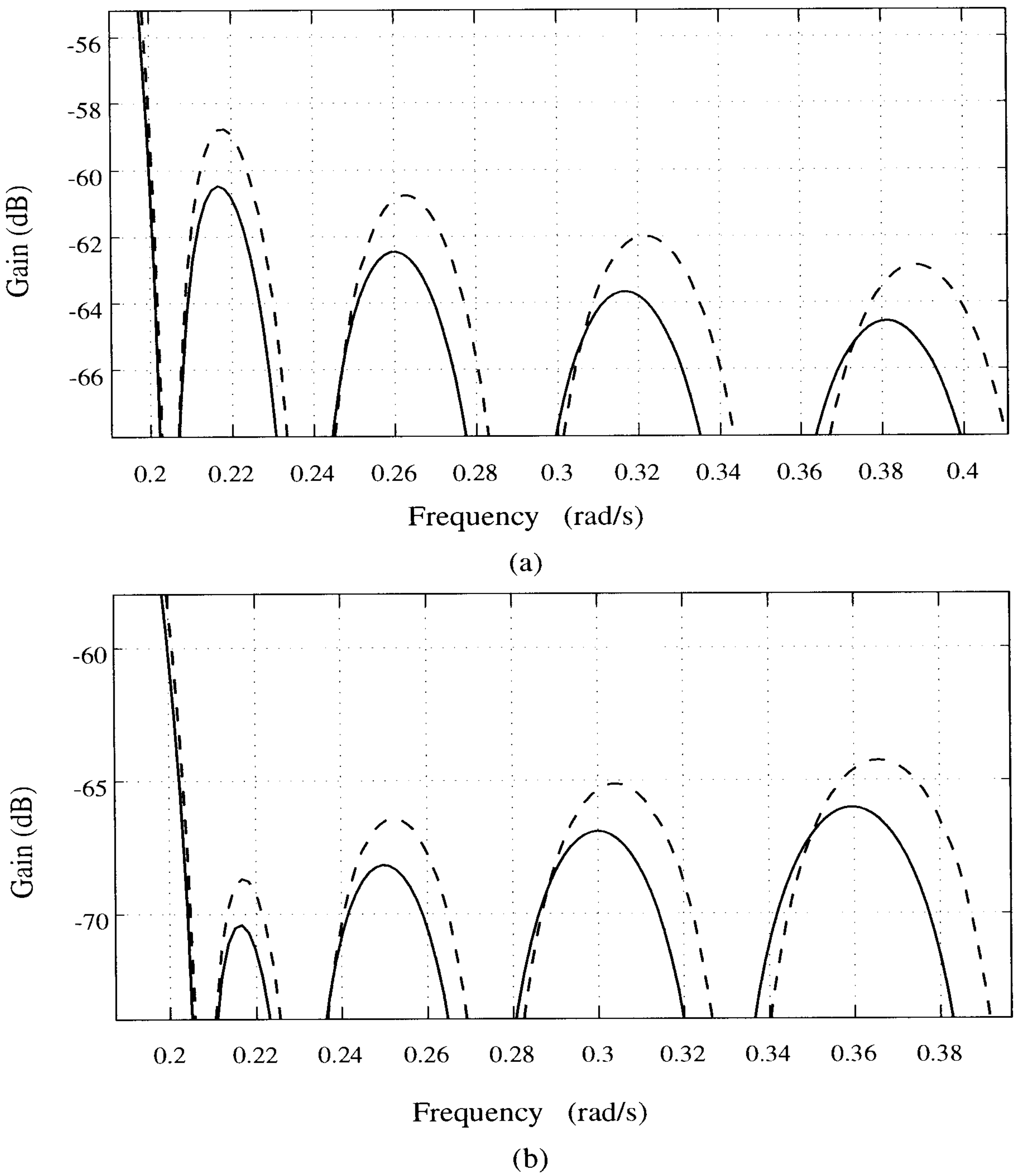


Figure 3.9. *Ultraspherical window amplitude spectrums for predicted N (solid line) and predicted $N-1$ (dashed line) yielding $\omega_R = 0.2$ rad/s and $R \geq 60$ dB for (a) $S = 10$ dB and (b) $S = -10$ dB (Example 3b).*

formula [29]

$$\alpha = \begin{cases} 0 & R \leq 13.26 \\ 0.76609(R-13.26)^{0.4} + 0.09834(R-13.26) & 13.26 < R \leq 60 \\ 0.12438(R+6.3) & 60 < R \leq 120 \end{cases} \tag{3.17}$$

as $\alpha = 6.8514$ and 9.4902 producing main-lobe half widths of $\omega_r = 0.1462$ and 0.1964 rad/s, respectively. The ultraspherical window was designed to yield the same side-lobe roll-off ratios and main-lobe widths as the Kaiser window measured as (a) $S = 29.19$ dB and (b) $S = 32.02$ dB. Minimizing Eq. (3.1) resulted in (a) $\mu = 1.0976$ and (b) $\mu = 1.2165$ and the procedure described in Section 3.2.4 gave (a) $x_\mu = 1.0023$ and (b) $x_\mu = 1.0044$. The difference in R was (a) $\Delta R = 0.2236$ and (b) $\Delta R = -0.4496$ dB. Thus, the ultraspherical window gives a better ripple ratio in (a) and the Kaiser window gives a better ripple ratio in (b) in agreement with Eq. (3.16).

3.6 Applications

The ultraspherical window has been presented in terms of its spectral characteristics to facilitate its use for a diverse range of applications. In this section, two window applications, beamforming and image processing, are presented to illustrate the benefits obtained by exercising the proposed method's flexibility.

3.6.1 Beamforming

In radar, ocean acoustics, and ultrasonics it is necessary to design antenna or transducer systems with specific directivity properties, i.e., for point-to-point communication systems a high gain in one direction with low gain in all other directions is considered desirable. Known as beamforming, this activity shapes the radiation pattern (or beam) of a transmitted signal so that most of its energy propagates towards the intended receiver or target. Similarly, when receiving signals, the receiver sensitivity (or beam) can be directed towards the transmitter or source to receive the maximum signal strength possible. Directing and focusing signal energy in this fashion leads to the rejection of interference from other sources and to reduced power requirements for transmitter and receiver power, which in turn provides cost savings.

One practical and common antenna/transducer configuration is the linear array, which is characterized by having all its radiating elements positioned in a straight line. Linear arrays can consist of one continuous radiating element or a number of individual discrete elements. Generally, discrete elements are favored because of their capability to dynamically change the directivity properties of the array. The array factor (AF) is used to describe an array's directivity properties. For a broadside array of length N with amplitude excitations for each isotropic element being symmetrical about the center of the array, the AF is given by [30]

$$\mathrm{AF}(\theta) = \begin{cases} \sum_{n=1}^{r} a'_n \cos[(2n-1)u] & \text{for odd } N \\ \sum_{n=1}^{r} a_n \cos[2(n-1)u] & \text{for even } N \end{cases} \tag{3.18}$$

where

$$\begin{aligned} u &= \text{the spatial frequency (degrees/m)} \\ &= (\pi d/\lambda)\cos\theta \\ \theta &= \text{the bearing angle (degrees)} \\ d &= \text{the spacing between elements (m)} \\ \lambda &= \text{the wavelength of the signal (m)} \\ a_n &= \text{the excitation coefficients or currents (A)} \\ a'_n &= \begin{cases} a_n & n \neq 1 \\ \frac{1}{2}a_n & n = 1 \end{cases} \\ r &= \begin{cases} (N+1)/2 & \text{for odd } N \\ N/2 & \text{for even } N \end{cases} \end{aligned}$$

The relationship between $\mathrm{AF}(\theta)$ and a_n is analogous to the relationship between $W(e^{j\omega T})$ and $w(nT)$. This similarity allows window design techniques to be applied directly to the design of antenna arrays. As in window designs, the trade-off between the main-lobe width and the side-lobe level of the AF is of primary importance. In the uniform array the excitation coefficients are all equal, as in the rectangular window, and hence the main-lobe width of the AF is narrow and side-lobe levels are large. At the other extreme, the binomial array's AF has no side lobes but has a large main-lobe width. Practical difficulties also arise with the implementation of the binomial array because the difference between excitation coefficients can be considerable leading to disparate current requirements. The Dolph-Chebyshev array, which offers an adjustable trade-off between the main-lobe width and side-lobe level, overcomes these implementation difficulties and is generally accepted as being a practical compromise between the uniform and binomial arrays. The AF suggests that the Dolph-Chebyshev array is best used when no prior knowledge of the interference sources is available, i.e., the likelihood of interference is equal in all directions. However, if the general direction of interference sources can be identified, no adjustments can be made to the side-lobe roll-off ratio when using the Dolph-Chebyshev array. If the directions of the interference sources are known precisely, the method of Shpak [31] generates optimal patterns with respect to some error criterion that achieves prescribed null locations by using a Remez-type exchange algorithm.

A solution that requires less computation than the method of Shpak yet achieves more flexibility than Dolph-Chebyshev designs could be to use the three-parameter ultraspherical window, in which case the excitation coefficients are given by

$$a_n = w[(r+n-1)T] \quad \text{for } n = 1, 2, ..., r \tag{3.19}$$

where $w(nT)$ are the coefficients provided by Eq. (2.1) resulting in

$$\text{AF}(\theta) = C_{N-1}^{\mu}(x_\mu \cos u). \tag{3.20}$$

This is equivalent to the amplitude function of the ultraspherical window given in Eq. (2.21) with the substitution $u = \omega T/2$. Similarly, all the techniques developed in this chapter are easily transferable to customizing the directivity properties of linear arrays. Fair comparisons between the two AFs can be made by designing ultraspherical and Dolph-Chebyshev arrays of the same length and the same null-to-null width, and then measuring the ripple ratios. To accomplish this, we make $\cos(\omega_n/2)$ in Eq. (3.3) equal for both the Dolph-Chebyshev and ultraspherical arrays which yields the relation

$$\frac{x^{(\mu)}_{N-1,1}}{x_\mu} = \frac{x^{(0)}_{N-1,1}}{x_0} = \frac{\cos\left[\frac{\pi}{2(N-1)}\right]}{x_0} \tag{3.21}$$

where x_0 is given by Eq. (2.29). Substituting and rearranging yields the closed-form expression for the ripple ratio

$$r = \frac{1}{\cosh\left\{(N-1)\cosh^{-1}\left[\frac{x_\mu}{x^{(\mu)}_{N-1,1}}\cos\left(\frac{\pi}{2(N-1)}\right)\right]\right\}} \tag{3.22}$$

that the Dolph-Chebyshev array of the same length and null-to-null width would produce compared to an ultraspherical array. This expression can be used to judge how much ripple ratio is sacrificed to attain a given side-lobe pattern.

Figure 3.10 shows enlarged plots around the first null of three ultraspherical arrays designed with $N = 31$, $\omega_n = 0.5$ rad/s ($\theta_n = 28.6479$ deg), and $S = -10$, 0, and 10 dB. The first side-lobe peak is 4.38 dB less for the case $S = -10$ dB and 3.84 dB more for the case $S = 10$ relative to the peak for the case $S = 0$ (i.e., the Dolph-Chebyshev array). On the other hand, the furthest side-lobe peak (not shown) is 5.62 dB more for $S = -10$ and 6.16 dB less for $S = 10$ dB relative to the peak for $S = 0$. The ripple ratio for the Dolph-Chebyshev array is given by Eq. (3.22) as -58.35 dB. An important observation is that the positioning of the second null weighs heavily on the amplitude of the first side lobe, which, in turn, is very important in determining the amplitude of the remaining side lobes. To this extent, an alteration in the amplitude of the first side lobe greatly influences the amplitude of the remaining side lobes in an inverse fashion, i.e., increasing the first side-lobe amplitude decreases most of the remaining side-lobe amplitudes. Experimental results indicate that the side lobe envelope of the ultraspherical array tends to cross that of the Dolph-Chebyshev array within the first three side lobes adjacent to the main lobe. In this respect, negative S values are preferred to the Dolph-Chebyshev array for narrowband interference sources that are confined to this region. Alternatively, positive S values are

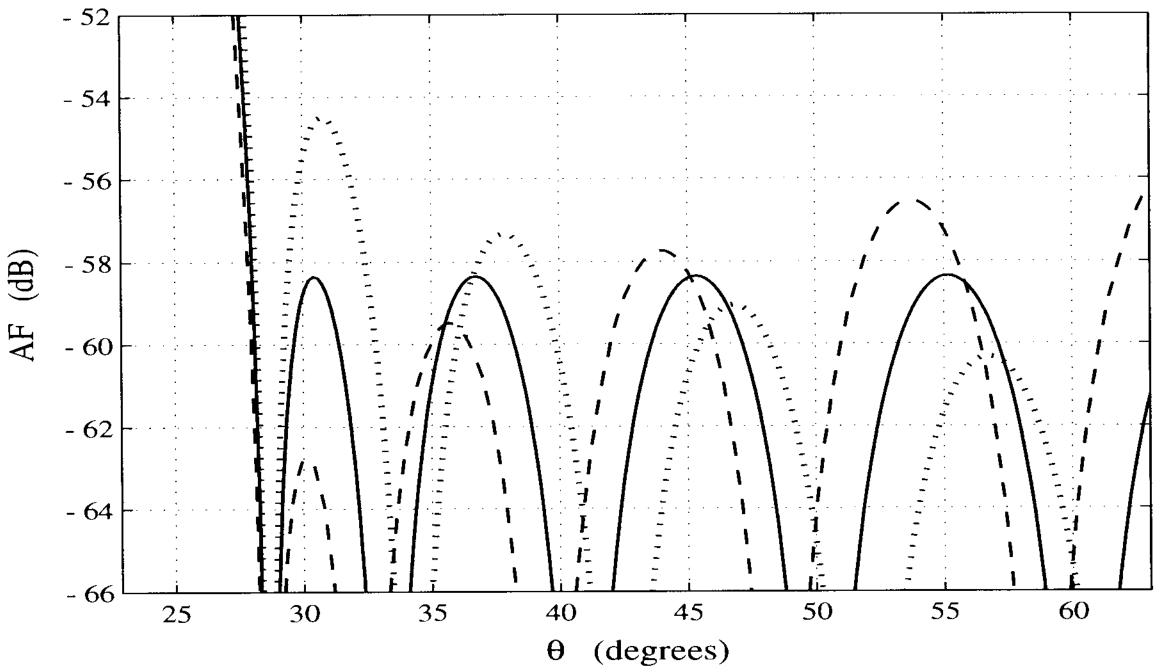


Figure 3.10. *AF for the ultrasperical array of length $N = 31$ and $\theta_n = 28.6479$ deg for the cases where $S = 0$ dB (solid line), $S = -10$ dB (dashed line), and $S = 10$ dB (dotted line).*

preferred for interference sources that fall past this region. Using the methods proposed in this chapter, antenna array designers are provided with an easy-to-use visual design approach for deciding what amount of trade-off between side-lobe pattern and ripple ratio is best for their particular situation.

3.6.2 Image Processing

With the ever-expanding gamut of computer monitors, hand-held devices such as digital cameras and video recorders, and high-end medical imaging systems, consumers can often base purchasing decisions on a few key image quality measures. One such measure is an image's contrast ratio (CR) which, simply put, defines the difference in light intensity between the darkest black and brightest white shades within an image. A high CR allows one to discern detailed differences between colors producing a crisp and sharp image. On the other hand, a low CR results in a blurring or smearing effect producing an image with little clarity. A direct consequence of the CR measure is its effect on an imaging system's capability to detect low-contrast objects residing near high-contrast objects, which can be of the utmost importance in some medical imaging applications, e.g., detecting cancerous tumors. Also, interpretation of an image's quality has been shown, through human trials, to be directly related to the CR measure [32].

A number of imaging systems such as synthetic aperture radar (SAR) [33], computerized tomography (CAT scans) [33], and charge-coupled device (CCD)-based X-rays [34] construct images by using two-dimensional windowed inverse DFTs on spatial frequency-domain data. For these systems CR tolerance is usually specified in terms of the worst-case spectral leakage of the window function used, which is directly related to the window's main-lobe to side-lobe energy ratio (MSR). Strictly speaking, the CR is defined as [35]

$$\mathrm{CR} = \frac{E_s + E_m}{E_s} = 1 + \mathrm{MSR} \tag{3.23}$$

where the side-lobe and main-lobe energies are given by

$$E_s = \int_{\omega_r}^{\pi} |W(e^{j\omega T})|^2 \, d\omega \tag{3.24}$$

and

$$E_m = \int_{0}^{\omega_r} |W(e^{j\omega T})|^2 \, d\omega \tag{3.25}$$

respectively, and MSR $= E_m/E_s$. By referring to the window's spectral representation as the inner product of the Fourier kernel

$$\mathbf{v} = [1 \; e^{-j\omega T} \; e^{-j2\omega T} \; \ldots \; e^{-j(N-1)\omega T}] \tag{3.26}$$

with the window coefficient vector $\mathbf{w}$, i.e., $W(e^{j\omega T}) = \mathbf{w}^T\mathbf{v}$, the side-lobe energy E_s can be expressed in the form

$$E_s = \mathbf{w}^T\mathbf{Q}\mathbf{w} \tag{3.27}$$

where

$$\mathbf{Q} = \mathbf{Q}(\omega_r) = 2\int_{\omega_r}^{\pi} \mathbf{V}\, d\omega \tag{3.28}$$

and $\mathbf{V} = \mathbf{v}\mathbf{v}^*$. The elements of $\mathbf{Q}$ are given by

$$q(n,m) = \begin{cases} -(\omega_r/\pi)\text{sinc}[\omega_r(m-n)] & \text{for } m \neq n \\ 1 - \omega_r/\pi & \text{for } m = n \end{cases} \tag{3.29}$$

where $\mathbf{Q}$ is a real, symmetric, positive-definite Toeplitz matrix. Using Parseval's theorem, the total energy is found as

$$E_t = E_m + E_s = \mathbf{w}^T\mathbf{w} \tag{3.30}$$

where a simple rearrangement yields the main-lobe energy E_m. Thus a window's CR can be calculated as

$$\text{CR} = \frac{\mathbf{w}^T\mathbf{w}}{\mathbf{w}^T\mathbf{Q}\mathbf{w}}. \tag{3.31}$$

Using the flexible three-parameter ultraspherical window for the windowing operation, the side-lobe patterns can be easily adjusted to alter the energy contained in the side lobes and, consequently, the value of the CR measure. Figure 3.11 shows plots of the normalized CR vs. the side-lobe roll-off ratio S in dB for various main-lobe half-width quantities. The curves are convex with easily-discernible global maximum values. As such, the ultraspherical window that possesses the maximum CR for a given window length N and main-lobe width ω_r can be found through the appropriate selection of S. This can be accomplished by solving the one-dimensional optimization problem

$$\underset{S_L \leq S \leq S_H}{\text{minimize}}\ F = -\text{CR} = -\frac{\mathbf{w}^T\mathbf{w}}{\mathbf{w}^T\mathbf{Q}\mathbf{w}} \tag{3.32}$$

where vector $\mathbf{w}$ is calculated using Eq. (2.1) and the techniques described in Sections 3.2.1 and 3.2.3, the $\mathbf{Q}$ matrix is calculated using Eq. (3.29), $S_L = 0$ dB, and $S_H = 30$ dB. For the example with $N = 31$ and $\omega_r = 0.4$ rad/s, the solution of Eq. (3.32) yields a maximum CR value of 41.01 dB occurring at $S = 17.75$ dB. The corresponding parameters for the ultraspherical window are $\mu = 1.0810$ and $x_\mu = 1.0166$.

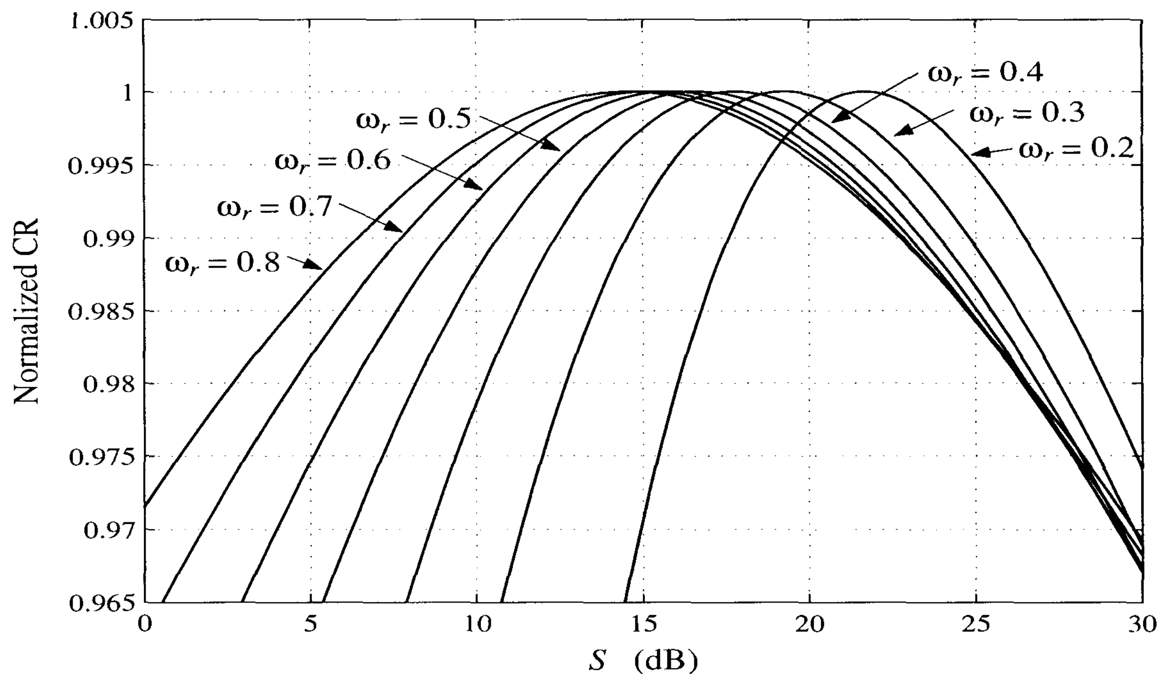


Figure 3.11. *The normalized CR vs. side-lobe roll-off ratio S with various main-lobe half width quantities for the ultraspherical window of length $N = 31$.*

3.7 Conclusions

A method for selecting the parameters of the ultraspherical window so as to achieve prescribed spectral characteristics has been proposed. The method comprises a collection of techniques that can be used to determine the independent parameters of the ultraspherical window such that a specified ripple ratio, main-lobe width or null-to-null width along with a specified side-lobe roll-off ratio can be achieved. The Kaiser, Saramäki, and Dolph-Chebyshev two-parameter windows can yield a specified ripple ratio and main-lobe width; however their side-lobe patterns cannot be controlled as in the proposed method. Experimental results have shown that the desired characteristics can be achieved with a high degree of precision. A difference in the performance of the ultraspherical and Kaiser windows has been identified, which depends critically on the required specifications. A rule for selecting either the ultraspherical or Kaiser window based on the performance difference was proposed. In addition, an expression is provided that can be used to judge how much ripple ratio is sacrificed to attain a given side-lobe pattern when compared to the Dolph-Chebyshev pattern. This is useful for antenna array designers who may need to trade-off between side-lobe pattern and ripple ratio for the application at hand. The proposed method can also be used to increase the contrast ratio in imaging systems that construct images by using two-dimensional windowed inverse DFTs on spatial frequency-domain data.

Chapter 4

Design of Nonrecursive Digital Filters Using the Ultraspherical Window

4.1 Introduction

Nonrecursive digital filters are usually designed to have a symmetrical impulse response thereby achieving linear phase (or constant group delay) and filter realizations with a reduced number of multiplications. Comparisons between different digital filter types (recursive and nonrecursive) and their attributes (linearity, time variance, and casuality) can be found in [1] and [36]. One popular method for nonrecursive digital-filter design is the window method which was discussed in Section 1.6 along with the weighted-Chebyshev method. As mentioned, the window method is based largely on closed-form solutions and as a result it is straightforward to apply and entails a relatively insignificant amount of computation; however, it usually yields suboptimal designs whereby the filter order required to satisfy a given set of specifications is not the lowest that can be achieved. The window method is useful when computational requirements for digital-filter design must be kept to a minimum such as in applications where the design has to be carried out on-line in real or quasi-real time. Simple signal processing algorithms and structures [37] can address these problems by trading between the accuracy of results and the utilization of implementation resources. In [38], [39] a window-based algorithmic approach to the design of low-power frequency-selective digital filters is presented whereby reduction of the average power consumption of the filter is achieved in speech processing and high-fidelity hardware by dynamically varying the filter length based on signal statistics. In these applications, flexible windows that allow one to achieve prescribed filter specifications with reduced filter length and whose coefficients can be generated quickly are highly desirable.

In this chapter, the window method for digital-filter design is used in conjunction with the ultraspherical window to design nonrecursive digital filters, digital differentiators, and Hilbert transformers so as to achieve prescribed specifications with minimal design computation and reduced filter length (as compared with designs obtained using other windows). The chapter is structured as follows. Section 4.2 introduces relevant information concerning the window method. Section 4.3 proposes a choice for the ultraspherical window's parameters that yield prescribed specifications. Section 4.4 provides a concise algorithm for the filter design. Section 4.5 provides comparisons with designs based on other windows as well as designs based on the Remez algorithm. Section 4.6 describes methods for highpass, bandpass, and bandstop filter design. Section 4.7 deals with the design of digital differentiators and Hilbert transformers so as to achieve prescribed specifications. Section 4.8 provides design examples. Section 4.9 provides concluding remarks.

4.2 Window Method

In the window method, an idealized frequency response is assumed and upon the application of the Fourier series, an infinite-duration impulse response is obtained. For a lowpass filter, we have

$$H_{id}(e^{j\omega T}) = \begin{cases} 1 & \text{for } |\omega| \leq \omega_c \\ 0 & \text{for } \omega_c < |\omega| \leq \omega_s/2 \end{cases} \tag{4.1}$$

where ω_c and ω_s are the cutoff and sampling frequencies, respectively. The infinite-duration impulse response is obtained as

$$h_{id}(nT) = \begin{cases} \omega_c/\pi & \text{for } n = 0 \\ \dfrac{1}{n\pi} \sin \omega_c nT & \text{for } n \neq 0 \end{cases} \tag{4.2}$$

where $-\infty \leq n \leq \infty$. The design of highpass, bandpass, and bandstop filters is discussed later. A realizable filter is obtained by multiplying the infinite-duration impulse response by the window function, i.e., by letting

$$h_0(nT) = w(nT)h_{id}(nT) \tag{4.3}$$

where $w(nT)$ is a window function of length $N = 2M+1$. The idealized impulse response $h_{id}(nT)$ requires an insignificant amount of computation and, therefore, the computation required to design a filter is practically the same as that required for the calculation of the coefficients of the window, which was addressed in Chapter 2. If N is odd, M is an integer and $|n| = \{0, 1, 2, ..., M\}$ is used for both the window and impulse response. If N is even, M is a fraction and $|n| = \{0.5, 1.5, 2.5, ..., M\}$ is used [1]. Odd-length nonrecursive filters are assumed throughout because the frequency response of an even-length symmetric nonrecursive filter is 0 at the Nyquist frequency, which is inappropriate for highpass and bandstop filters. However, this property of even-length nonrecursive filters can be used for the design of Hilbert transformers as discussed later. A causal filter can be obtained by delaying the impulse response by a period MT, i.e.,

$$h(nT) = h_0\left[(n-M)T\right] \quad \text{for } 0 \leq n \leq N-1 \tag{4.4}$$

The frequency response of the filter is given by the convolution of the idealized frequency response and the spectral representation of the window, i.e.,

$$H(e^{j\omega T}) = \frac{1}{\omega_s} \int_{-\omega_s/2}^{\omega_s/2} H_{id}(e^{j\theta T}) W(e^{j(\omega-\theta)T})\, d\theta \tag{4.5}$$

where $W(e^{j\omega T})$ is given by Eq. (1.11).

4.3 Choice of Window Parameters

A nonrecursive (noncausal) lowpass filter is typically required to satisfy the equations

$$\begin{aligned} 1-\delta_p \leq H(e^{j\omega T}) \leq 1+\delta_p \quad & \text{for } |\omega| \in [0, \omega_p] \\ -\delta_a \leq H(e^{j\omega T}) \leq \delta_a \quad & \text{for } |\omega| \in [\omega_a, \omega_s/2] \end{aligned} \tag{4.6}$$

where δ_p and δ_a are the passband and stopband ripples and ω_p and ω_a are the passband and stopband edge frequencies, respectively. In nonrecursive filters designed using the window method, the passband ripple turns out to be approximately equal to the stopband ripple, i.e., $\delta_p \approx \delta_a$. Therefore, one can design a filter that has a prescribed passband ripple or a prescribed stopband ripple. If the specifications call for a maximum passband ripple A_p and a minimum stopband attenuation A_a, both specified in dB, then it can easily be shown that [1]

$$\delta_p = \frac{10^{0.05A_p} - 1}{10^{0.05A_p} + 1} \quad \text{and} \quad \delta_a = 10^{-0.05A_a} \tag{4.7}$$

By designing a filter on the basis of

$$\delta = \min(\delta_p, \delta_a) \tag{4.8}$$

then if $\delta = \delta_p$ a filter will be obtained that has a passband ripple which is equal to A_p dB and a minimum stopband attenuation which is greater than A_a dB; and if $\delta = \delta_a$ a filter will be obtained that has a minimum stopband attenuation which is equal to A_a dB and a passband ripple which is less than A_p dB.

The ultraspherical window parameters μ, x_μ, and N, must be chosen such that the filter specifications are satisfied with the lowest possible filter length N. For a given set of prescribed specifications the optimum values of μ and x_μ could be determined through a trial-and-error approach but such an approach would be laborious and time consuming. Fortunately, a fairly general method that parallels Kaiser's method [7] can be used to design filters that would satisfy arbitrary prescribed filter specifications. Through extensive experimentation, we found out that parameters μ and x_μ control the passband and stopband ripples and, consequently, the actual stopband attenuation, namely,

$$A_a = -20\log_{10}(\delta) \tag{4.9}$$

Strictly speaking, parameter x_μ alters the window's ripple ratio at the expense of the null-to-null width, in effect, providing a trade-off between the two just like parameter α in the Kaiser window [7] and parameter x_0 in the Dolph-Chebyshev window [9]. Thus x_μ has a strong influence on the stopband attenuation. On the other hand, parameter μ controls the window's side-lobe pattern which affects the stopband attenuation but not to the extent that x_μ does. This property is observed in Fig. 2.2 where windows with $\mu = 0$ and 1 yield ripple ratios of -45.84 and -39.85 dB, respectively. On the other hand, the filter length N controls the transition bandwidth of the filter, namely,

$$B_t = \omega_a - \omega_p \tag{4.10}$$

but has little effect on the stopband attenuation. Consequently, the required value of N is dependent on parameter μ while being relatively independent of parameter x_μ.

The value of parameter μ that minimizes the filter length for a set of prescribed specifications can be determined by comparing the performance of filters designed using the ultraspherical window with varying values of the adjustable parameters for a fixed filter length and cutoff frequency as in [8]. The transition bandwidth is measured from the resulting filter and used to calculate the performance measure

$$D = B_t(N-1)/\omega_s \tag{4.11}$$

which is a normalized transition bandwidth that is approximately independent of the filter length [7], [8], [28]. Figure 4.1 shows plots of the stopband attenuation vs. D for filters designed using the ultraspherical window with $\mu = 0$, 0.4, 0.6, and 1. As can be seen, the filter performance depends critically on the choice of parameter μ. In addition, we note that there is no unique fixed value of μ that yields minimum stopband attenuation, i.e., the optimal value of μ changes with D. As such, it is possible to select an optimal value of μ that minimizes the filter length for a set of prescribed specifications. The value of μ that minimizes the filter length was found by calculating the value of μ that maximizes the stopband attenuation for a given normalized transition bandwidth D. Through curve fitting, an empirical formula for the optimal μ was derived as

$$\mu = -1.721 \times 10^{-5} A_a^2 + 6.721 \times 10^{-3} A_a + 1.897 \times 10^{-1} \tag{4.12}$$

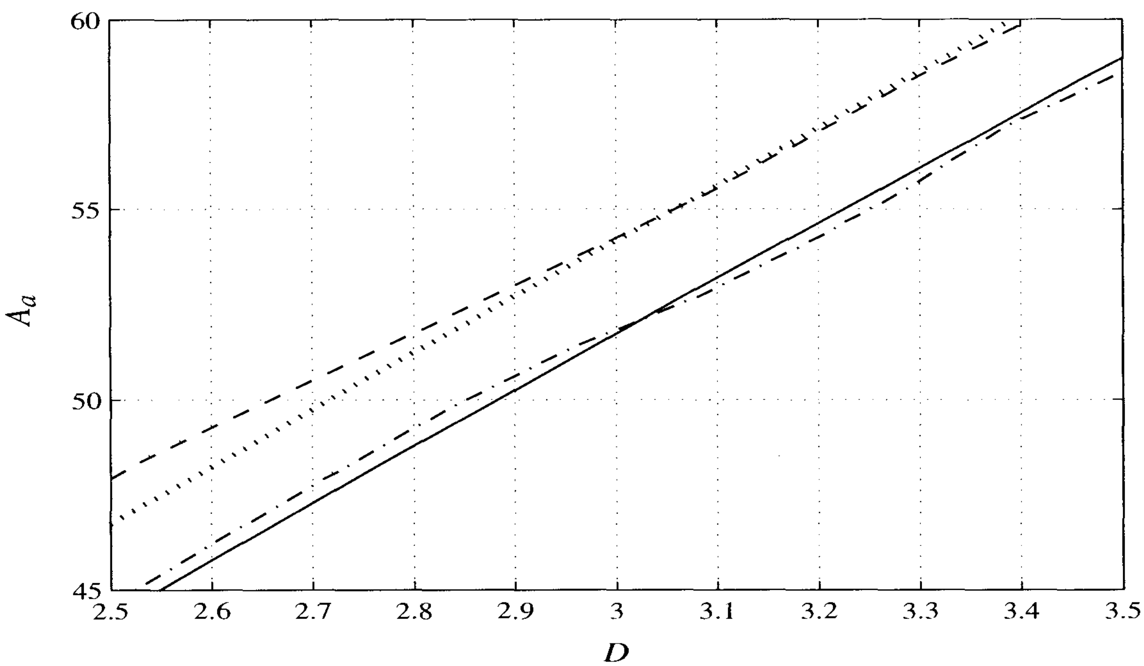


Figure 4.1. *Stopband attenuation vs. D for filters designed using the ultraspherical window with* $\mu = 0$ *(dash-dotted line),* 0.4 *(dashed line),* 0.6 *(dotted line), and* 1 *(solid line) for the filter design parameters* $N = 127$, $\omega_c = 0.4\pi$ *rad/s, and* $\omega_s = 2\pi$ *rad/s.*

This estimate provides relatively accurate predictions for μ for most practical purposes, i.e., it holds true for low as well as high values of N.

The minimum filter length required so as to achieve a desired stopband attenuation and transition bandwidth can be determined as the smallest odd integer satisfying the inequality [7]

$$N \geq \frac{D\omega_s}{B_t} + 1 \tag{4.13}$$

From Eq. (4.13) it becomes clear that N can be predicted by obtaining an accurate approximation for D. As can be observed in Fig. 4.1, D is influenced by both the stopband attenuation and the parameter μ. Through curve fitting, an empirical formula was deduced for D corresponding to the value of μ given by Eq. (4.12) as

$$D = \begin{cases} 4.645 \times 10^{-5} A_a^2 + 6.216 \times 10^{-2} A_a - 4.818 \times 10^{-1} & \text{for } A_a \leq 80 \\ 1.710 \times 10^{-5} A_a^2 + 7.089 \times 10^{-2} A_a - 8.937 \times 10^{-1} & \text{for } A_a > 80 \end{cases} \tag{4.14}$$

The final window parameter x_μ provides a trade-off between the stopband attenuation and the transition bandwidth of the filter and can be determined using Eq. (2.22). It is clear that parameter x_μ can be predicted by obtaining an approximation for parameter β. Figure 4.2 shows plots of parameter β vs. stopband attenuation for filters designed using the ultraspherical window with $\mu = 0$, 0.4, 0.6, and 1. As can be seen, β varies significantly depending on the choice of the stopband attenuation and parameter μ. Through curve fitting, an empirical formula was derived for parameter β, which corresponds to the value of μ given by Eq. (4.12), as

$$\beta = \begin{cases} 4.024 \times 10^{-5} A_a^2 + 2.423 \times 10^{-2} A_a + 3.574 \times 10^{-1} & \text{for } A_a \leq 60 \\ 7.303 \times 10^{-5} A_a^2 + 2.079 \times 10^{-2} A_a + 4.447 \times 10^{-1} & \text{for } 60 < A_a \leq 120 \\ 6.733 \times 10^{-6} A_a^2 + 3.337 \times 10^{-2} A_a - 1.192 \times 10^{-1} & \text{for } 120 < A_a \leq 180 \end{cases} \tag{4.15}$$

Equations (4.12), (4.13), (4.14), and (4.15) provide a closed-form Kaiser-like method for achieving prescribed specifications while minimizing the filter length N through the appropriate selection of the window parameters. However, for some applications one may be willing to increase N so as to achieve different frequency selectivity characteristics. For instance, increased stopband roll-off, i.e., increased suppression of stopband energy furthest from the transition bandwidth (see [40]), can be achieved by increasing parameter μ but this has the effect of decreasing the stopband attenuation. Thus to achieve the same stopband attenuation, N must be increased. To accommodate for these scenarios, estimates for D and β were obtained as

$$D = aA_a^2 + bA_a + c \tag{4.16}$$

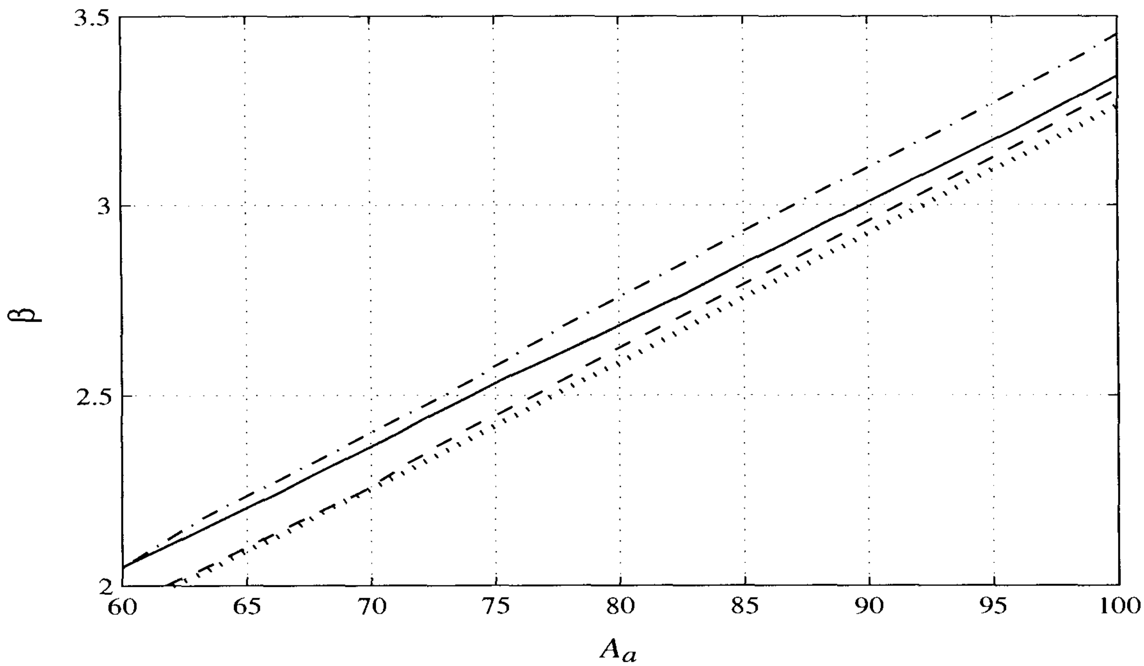


Figure 4.2. *Parameter β vs. stopband attenuation for filters designed using the ultraspherical window with $\mu = 0$ (dash-dotted line), 0.4 (dashed line), 0.6 (dotted line), and 1 (solid line) for the filter design parameters $N = 127$, $\omega_c = 0.4\pi$ rad/s, and $\omega_s = 2\pi$ rad/s.*

and

$$\beta = \begin{cases} a_1 A_a^2 + b_1 A_a + c_1 & \text{for } A_a \le 60 \\ a_2 A_a^2 + b_2 A_a + c_2 & \text{for } A_a > 60 \end{cases} \tag{4.17}$$

for the values $\mu = \{0, 0.1, 0.2, ..., 1.0\}$ where the model coefficients are given in Tables 4.1 and 4.2, respectively. The estimate for D should be used in conjunction with Eq. (4.13) to predict the required value of N for the particular selection of μ and a set of prescribed filter specifications. Estimates for D and β that correspond to values of μ in the range $[0, 1]$ that are not included in Tables 4.1 and 4.2 can be obtained using cubic spline interpolation where $(\mu_i)_1^n = \{0, 0.1, 0.2, ..., 1.0\}$ are the abscissa values and $(D_i)_1^n$ and $(\beta_i)_1^n$ are their corresponding ordinate values (see Chap. 7 of [41]). This window-parameter alteration technique can provide designers with a simple approach for tailoring a filter's frequency selectivity for a particular application while still achieving prescribed specifications.

Table 4.1. *Model Coefficients for Parameter D*

μ	a	b	c
0.0	−4.198E−5	7.784E−2	−7.778E−1
0.1	−2.961E−5	7.574E−2	−7.659E−1
0.2	−1.747E−5	7.348E−2	−7.369E−1
0.3	−5.808E−6	7.109E−2	−6.924E−1
0.4	6.462E−6	6.844E−2	−6.266E−1
0.5	3.221E−5	6.408E−2	−5.048E−1
0.6	6.111E−5	5.957E−2	−3.733E−1
0.7	7.789E−5	5.736E−2	−3.061E−1
0.8	6.328E−5	5.975E−2	−3.531E−1
0.9	3.620E−5	6.391E−2	−4.377E−1
1.0	1.532E−5	6.717E−2	−4.974E−1

Table 4.2. *Model Coefficients for Parameter β*

μ	a_1	b_1	c_1	a_2	b_2	c_2
0.0	7.337E−5	2.533E−2	3.404E−1	1.534E−5	3.183E−2	1.585E−1
0.1	7.895E−5	2.430E−2	3.401E−1	1.680E−5	3.142E−2	1.357E−1
0.2	8.930E−5	2.265E−2	3.645E−1	1.811E−5	3.094E−2	1.233E−1
0.3	1.126E−4	1.971E−2	4.261E−1	1.847E−5	3.055E−2	1.126E−1
0.4	1.240E−4	1.774E−2	4.779E−1	2.434E−5	2.912E−2	1.535E−1
0.5	1.265E−4	1.656E−2	5.203E−1	5.085E−5	2.439E−2	3.260E−1
0.6	1.134E−4	1.690E−2	5.359E−1	7.947E−5	1.956E−2	5.033E−1
0.7	8.981E−5	1.845E−2	5.281E−1	7.299E−5	2.120E−2	4.171E−1
0.8	6.355E−5	2.070E−2	5.033E−1	3.755E−5	2.748E−2	1.763E−1
0.9	8.045E−5	1.987E−2	5.308E−1	2.149E−5	2.983E−2	1.373E−1
1.0	9.410E−5	1.925E−2	5.550E−1	9.433E−6	3.158E−2	1.144E−1

4.4 Design Algorithm

Based on findings from the previous section, a lowpass nonrecursive filter that would satisfy the specifications

- Passband edge: ω_p
- Stopband edge: ω_a
- Passband ripple: A_p
- Stopband ripple: A_a
- Sampling frequency: ω_s

can be designed using the following algorithm:

Algorithm 4.1 Lowpass filter design using the ultraspherical window

Step 1: Input ω_p, ω_a, A_p, A_a, and ω_s. Find the 'design' δ using Eq. (4.8) and then update A_a using Eq. (4.9).

Step 2: Calculate the window parameter μ using Eq. (4.12).

Step 3: Calculate the filter length N using Eq. (4.13) in conjunction with Eqs. (4.10) and (4.14). Round N up to the nearest odd integer.

Step 4: Calculate the window parameter x_μ using Eq. (2.22) in conjunction with Eq. (4.15) and the method described in Algorithm 2.1 for calculating $x_{N-1,1}^{(\mu)}$.

Step 5: With μ, x_μ, and N known, the coefficients of the ultraspherical window can be generated from Eq. (2.47).

Step 6: Calculate the relevant terms of the infinite-duration impulse response using Eq. (4.2) with $\omega_c = (\omega_p + \omega_a)/2$.

Step 7: Obtain the noncausal finite-duration impulse response using Eq. (4.3).

Step 8: Obtain the causal design using Eq. (4.4).

Step 9: Check the design obtained to ensure that the filter satisfies the prescribed specifications. If it does not, increase N by 2 and go to Step 4.

4.5 Comparison with Other Windows

The performance of different windows was compared by designing filters for fixed values of N and ω_c [8]. The transition bandwidth for the resulting filters was measured and used to calculate D using Eq. (4.11). Figure 4.3 shows plots of the stopband attenuation vs. D for $N = 127$, $\omega_c = 0.4\pi$ rad/s, and $\omega_s = 2\pi$ rad/s for a variety of fixed and adjustable windows. Expressions for these windows can be found in [1], [5], while the Nuttall window is described in [42]. For the adjustable windows (Kaiser, Dolph-Chebyshev, Saramäki, ultraspherical, and Gaussian windows) a number of filters were designed by altering the independent window parameter. As can be seen, the ultraspherical window offers better performance than the Kaiser, Dolph-Chebyshev, Saramäki, and Gaussian windows achieving an average increase in the stopband attenuation of 2.48 dB relative to that in the Kaiser window, 4.29 dB relative to that in the Dolph-Chebyshev window, and 2.21 dB relative to that in the Saramäki window. The Gaussian window provides much poorer results than the other adjustable windows. For the sake of comparison, equiripple designs based on the weighted-Chebyshev method of Parks-McClellan [17] were also carried out assuming equal values for the passband and stopband ripples, i.e., $\delta_p = \delta_a$. The weighted-Chebyshev method increases the stopband attenuation by about 2.93 dB on the average but this is to be expected since the Remez algorithm yields designs that are L_∞ optimal.

The performance of different windows was also compared by finding the required filter length so as to achieve a set of prescribed specifications. Figure 4.4 shows plots of the actual stopband attenuations achieved for a fixed transition bandwidth of $B_t = 0.2$ rad/s and filter length N for lowpass filters designed using the Kaiser, Dolph-Chebyshev, and ultraspherical windows. Results for the Saramäki window have been omitted as they are very similar to those of the Kaiser window. The filters were designed so as to achieve the transition bandwidth $B_t = 0.2$ rad/s to a high degree of precision by fine tuning the independent window parameter using optimization techniques. As can be seen, for a given filter length, the ultraspherical window increases the stopband attenuation relative to the

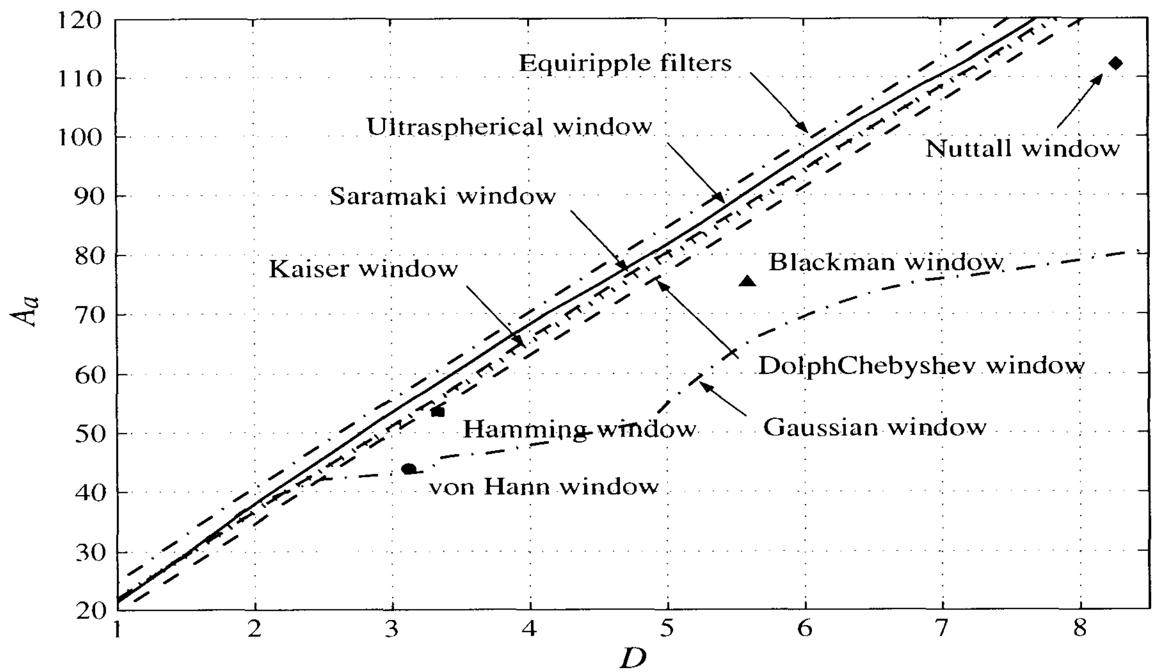


Figure 4.3. *Stopband attenuation vs. D for filters designed using various windows with $N = 127$ and $\omega_c = 0.4\pi$ rads/s. Results for equiripple filters of the same length with $\delta_p = \delta_a$ are included for comparison.*

attenuation in the Kaiser and Dolph-Chebyshev windows achieving on the average an increase of 2.61 dB relative to that in the Kaiser window and 4.49 dB relative to that in the Dolph-Chebyshev window. Alternatively, for prescribed specifications the ultraspherical window yields lower-order filters than the Kaiser or Dolph-Chebyshev windows. On the other hand, the weighted-Chebyshev method increases the stopband attenuation relative to that in the ultraspherical window by about 2.76 dB on the average. Filters designed using the weighted-Chebyshev method were designed with equal values for the passband and stopband ripples.

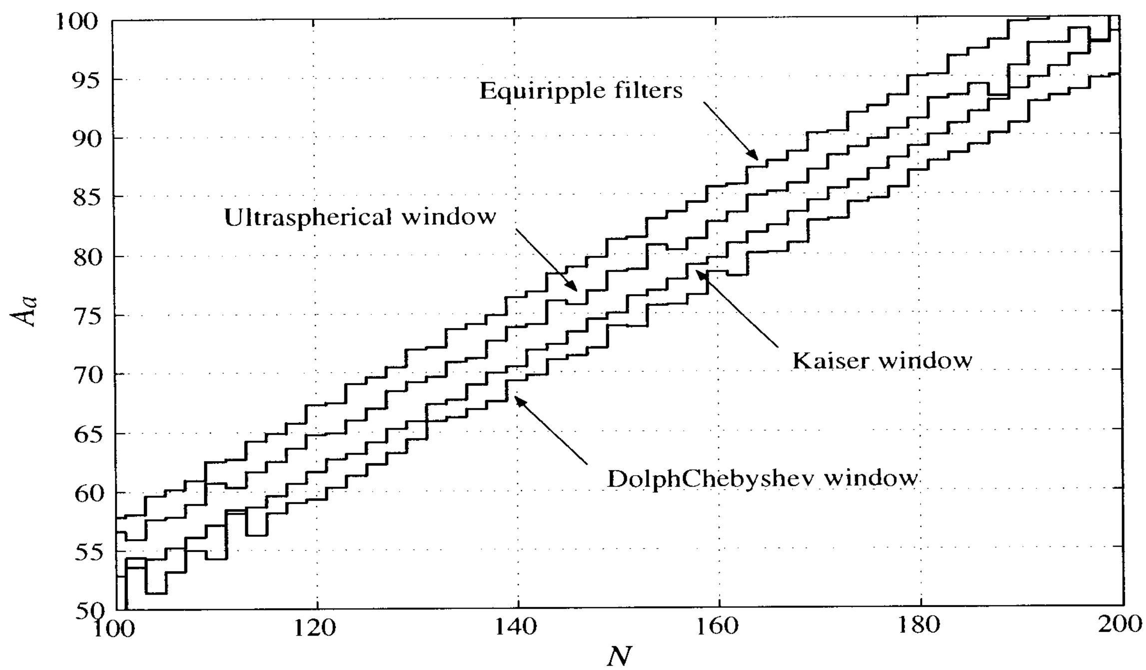


Figure 4.4. *Actual stopband attenuation A_a achieved by filters designed with length N and transition bandwidth $B_t = 0.2$ rad/s. The equiripple filters were designed with $\delta_p = \delta_a$.*

4.6 Highpass, Bandpass, and Bandstop Filters

The above design method can be readily extended to the design of highpass, bandpass, and bandstop filters by following the procedure in [1]. For instance, the specifications for highpass filters assume the form

$$\begin{aligned} -\delta_a \le H(e^{j\omega T}) \le \delta_a \quad & \text{for } |\omega| \in [0, \omega_a] \\ 1 - \delta_p \le H(e^{j\omega T}) \le 1 + \delta_p \quad & \text{for } |\omega| \in [\omega_p, \omega_s/2] \end{aligned} \tag{4.18}$$

The ideal frequency response is taken as

$$H_{id}(e^{j\omega T}) = \begin{cases} 0 & \text{for } |\omega| \le \omega_c \\ 1 & \text{for } \omega_c < |\omega| \le \omega_s/2 \end{cases} \tag{4.19}$$

with $\omega_c = (\omega_p + \omega_a)/2$. Straightforward analysis gives the infinite-duration impulse response as

$$h_{id}(nT) = \begin{cases} 1 - \omega_c/\pi & \text{for } n = 0 \\ -\dfrac{1}{n\pi} \sin \omega_c nT & \text{for } n \neq 0 \end{cases} \tag{4.20}$$

The transition bandwidth in Eq. (4.13) is

$$B_t = \omega_p - \omega_a \tag{4.21}$$

The specifications for bandpass filters assume the form

$$\begin{aligned} -\delta_a \leq H(e^{j\omega T}) &\leq \delta_a && \text{for } |\omega| \in [0, \omega_{a1}] \\ 1 - \delta_p \leq H(e^{j\omega T}) &\leq 1 + \delta_p && \text{for } |\omega| \in [\omega_{p1}, \omega_{p2}] \\ -\delta_a \leq H(e^{j\omega T}) &\leq \delta_a && \text{for } |\omega| \in [\omega_{a2}, \omega_s/2] \end{aligned} \tag{4.22}$$

The ideal frequency response is taken as

$$H_{id}(e^{j\omega T}) = \begin{cases} 0 & \text{for } 0 \leq |\omega| < \omega_{c1} \\ 1 & \text{for } \omega_{c1} \leq |\omega| \leq \omega_{c2} \\ 0 & \text{for } \omega_{c2} < |\omega| \leq \omega_s/2 \end{cases} \tag{4.23}$$

with

$$\omega_{c1} = \omega_{p1} - \frac{B_t}{2} \quad \text{and} \quad \omega_{c2} = \omega_{p2} + \frac{B_t}{2} \tag{4.24}$$

where the design is based on the narrower of the two transition bandwidths, i.e.,

$$B_t = \min[(\omega_{p1} - \omega_{a1}), (\omega_{a2} - \omega_{p2})] \tag{4.25}$$

Straightforward analysis gives the infinite-duration impulse response as

$$h_{id}(nT) = \frac{\sin(\omega_{c2} nT)}{\pi n} - \frac{\sin(\omega_{c1} nT)}{\pi n} \quad \text{for all } n \tag{4.26}$$

The specifications for bandstop filters assume the form

$$\begin{aligned} 1 - \delta_p \leq H(e^{j\omega T}) &\leq 1 + \delta_p && \text{for } |\omega| \in [0, \omega_{p1}] \\ -\delta_a \leq H(e^{j\omega T}) &\leq \delta_a && \text{for } |\omega| \in [\omega_{a1}, \omega_{a2}] \\ 1 - \delta_p \leq H(e^{j\omega T}) &\leq 1 + \delta_p && \text{for } |\omega| \in [\omega_{p2}, \omega_s/2] \end{aligned} \tag{4.27}$$

The ideal frequency response is taken as

$$H_{id}(e^{j\omega T}) = \begin{cases} 1 & \text{for } 0 \leq |\omega| \leq \omega_{c1} \\ 0 & \text{for } \omega_{c1} < |\omega| < \omega_{c2} \\ 1 & \text{for } \omega_{c2} \leq |\omega| \leq \omega_s/2 \end{cases} \tag{4.28}$$

with

$$\omega_{c1} = \omega_{p1} + \frac{B_t}{2} \quad \text{and} \quad \omega_{c2} = \omega_{p2} - \frac{B_t}{2} \tag{4.29}$$

where the design is based on the narrower of the two transition bandwidths, i.e.,

$$B_t = \min[(\omega_{a1} - \omega_{p1}), (\omega_{p2} - \omega_{a2})] \tag{4.30}$$

Straightforward analysis gives the infinite-duration impulse response as

$$h_{id}(nT) = \begin{cases} 1 - (\omega_{c2} - \omega_{c1})/\pi & \text{for } n = 0 \\ \dfrac{\sin \omega_{c1} nT - \sin \omega_{c2} nT}{n\pi} & \text{for } n \neq 0 \end{cases} \tag{4.31}$$

With the above modifications, Algorithm 4.1 can also be used to design highpass, bandpass, and bandstop filters as well as multiband filters [43].

4.7 Digital Differentiators and Hilbert Transformers

One advantage of the window method is the ease with which it can be applied to a wide range of filter design problems. In this section, we employ the window method for the design of digital differentiators and Hilbert transformers.

4.7.1 Digital Differentiators

In signal processing, the need often arises for the derivative of a signal at some time instant $t = t_1$. For example, if $y(nT)$ is required to be the first derivative of $x(t)$ at $t = nT$, we can write

$$y(nT) = f[x(t)] = \left.\frac{dx(t)}{dt}\right|_{t=nT} \tag{4.32}$$

Digital differentiators (DDs) have an ideal frequency response

$$H(e^{j\omega T}) = j\omega \quad \text{for } |\omega| \leq \omega_s/2 \tag{4.33}$$

Since differentiators amplify high frequency errors such as instrumentation measurement errors, band-limited differentiators are quite useful. Practical band-limited differentiator design can be accomplished in terms of a nonrecursive filter whose frequency response is required to satisfy the equations

$$\begin{aligned} j(\omega - \delta_p) \leq H(e^{j\omega T}) \leq j(\omega + \delta_p) &\quad \text{for } |\omega| \in [0, \omega_p] \\ -j(\delta_a) \leq H(e^{j\omega T}) \leq j(\delta_a) &\quad \text{for } |\omega| \in [\omega_a, \omega_s/2] \end{aligned} \tag{4.34}$$

For a band-limited differentiator, the ideal frequency response is taken as

$$H_{id}(e^{j\omega T}) = \begin{cases} j\omega & \text{for } |\omega| \leq \omega_c \\ 0 & \text{for } \omega_c < |\omega| \leq \omega_s/2 \end{cases} \tag{4.35}$$

with $\omega_c = (\omega_p + \omega_a)/2$. Straightforward analysis gives the infinite-duration impulse response as

$$h_{id}(nT) = \begin{cases} \dfrac{\omega_c \cos(n\omega_c)}{n\pi} - \dfrac{\sin(n\omega_c)}{n^2\pi} & \text{for } n \neq 0 \\ 0 & \text{for } n = 0 \end{cases} \tag{4.36}$$

The transition bandwidth in Eq. (4.13) is

$$B_t = \omega_a - \omega_p \tag{4.37}$$

In DDs the passband ripple and stopband attenuation are dependant on the cutoff frequency of the differentiator. To account for this, a correction factor for A_a of the form

$$A_a^{'} = A_a - A_{cor} \tag{4.38}$$

is required where $A_a^{'}$ is the corrected design attenuation whose value replaces A_a in Algorithm 4.1, A_a is the desired design attenuation, and A_{cor} is a correction factor given by

$$A_{cor} = a\omega_c^2 + b\omega_c + c \tag{4.39}$$

The values of the coefficients a, b, and c for the Kaiser, Dolph-Chebyshev, and ultraspherical windows are given in Table 4.3. Examples of differentiators designed using the window method and Remez algorithm can be found in [1].

Table 4.3. *Estimate Coefficients for Parameter A_{cor}*

Window Function	a	b	c
Kaiser	2.422	−13.73	10.86
Dolph-Chebyshev	2.700	−14.23	12.25
ultraspherical	1.506	−11.10	8.170

4.7.2 Hilbert Transformers

In signal processing, it is sometimes necessary to form an *analytic signal* [1], which can be generated by a complex filter with frequency response

$$H_A(e^{j\omega T}) = 1 + jH(e^{j\omega T}) \tag{4.40}$$

where $H(e^{j\omega T})$ is a Hilbert transformer which has an ideal frequency response given by

$$H(e^{j\omega T}) = \begin{cases} -j & \text{for } 0 < \omega < \omega_s/2 \\ j & \text{for } -\omega_s/2 < \omega < 0 \end{cases} \tag{4.41}$$

Practical Hilbert transformer design can be carried out by designing a nonrecursive filter whose frequency response is required to satisfy the equations

$$\begin{aligned} j(1-\delta_p) \le H(e^{j\omega T}) \le j(1+\delta_p) \quad & \text{for } \omega \in [\omega_{p1}, \omega_s/2] \\ j(-1-\delta_p) \le H(e^{j\omega T}) \le j(-1+\delta_p) \quad & \text{for } \omega \in [-\omega_s/2, -\omega_{p1}] \end{aligned} \tag{4.42}$$

Straightforward analysis gives the infinite-duration impulse response as

$$h_{id}(nT) = \begin{cases} \dfrac{2}{n\pi}\sin^2\dfrac{n\pi}{2} & \text{for } n \ne 0 \\ 0 & \text{for } n = 0 \end{cases} \tag{4.43}$$

The transition bandwidth in Eq. (4.13) is

$$B_t = 2\omega_{p1} \tag{4.44}$$

Like differentiators, it was found that Hilbert transformers required a correction factor for A_a of the form

$$A_a' = A_a + A_{cor} \tag{4.45}$$

where A_a' is the corrected design attenuation whose value replaces A_a throughout Algorithm 1, A_a is the desired design attenuation in dB, and A_{cor} is a correction factor given by $A_{cor} = 6.414$, 5.236, and 6.457 for the Kaiser, Dolph-Chebyshev, and ultraspherical windows, respectively.

4.8 Examples

Example 1: Design a lowpass filter with $\omega_p = 1$, $\omega_a = 1.2$ rad/s, and $A_a = 80$ dB using the Kaiser, Dolph-Chebyshev, and ultraspherical windows.

The adjustable parameters for the Kaiser, Dolph-Chebyshev, and ultraspherical windows were calculated as $\alpha = 7.857$, $\beta = 2.803$, and $\beta = 2.574$, respectively. The adjustable parameter α was calculated using the expression found in [1] while β was calculated using Eqs. 4.10 and 4.17 for the ultraspherical and Dolph-Chebyshev windows, respectively. The additional parameter calculated for the ultraspherical window was $\mu = 0.6173$. The stopband attenuations achieved were 79.38, 82.27, and 79.36 dB with transition bandwidths 0.1987, 0.1994, and 0.1965 rad/s, respectively.

To achieve the desired specifications more precisely a simple technique described in [6] can be employed. First, the actual stopband attenuation of the filter A_{ar} is measured for the estimated value of β. Then β is re-estimated using an adjusted design attenuation $A_a' = A_a - (A_{ar} - A_a)$ where A_a is the desired design stopband attenuation. With this modification, the re-estimated parameters assume the values $\alpha = 7.926$, $\beta = 2.725$, and $\beta = 2.596$ respectively. The stopband attenuations were 80.03, 79.96, and 79.83 dB with transition bandwidths 0.2005, 0.1923, and 0.1983 rad/s, respectively. The filter lengths required so as to achieve the specifications were $N = 159$ for the Kaiser window, $N = 165$ for the Dolph-Chebyshev window, and $N = 153$ for the ultraspherical window. Figure 4.5 shows the amplitude responses of the designed filters.

Example 2: Design a bandstop filter with $\omega_{p1} = 0.5$, $\omega_{a1} = 0.7$, $\omega_{a2} = 2.0$, $\omega_{p2} = 2.2$ rad/s, and $A_a = 40$ dB using the Kaiser, Dolph-Chebyshev, and ultraspherical windows.

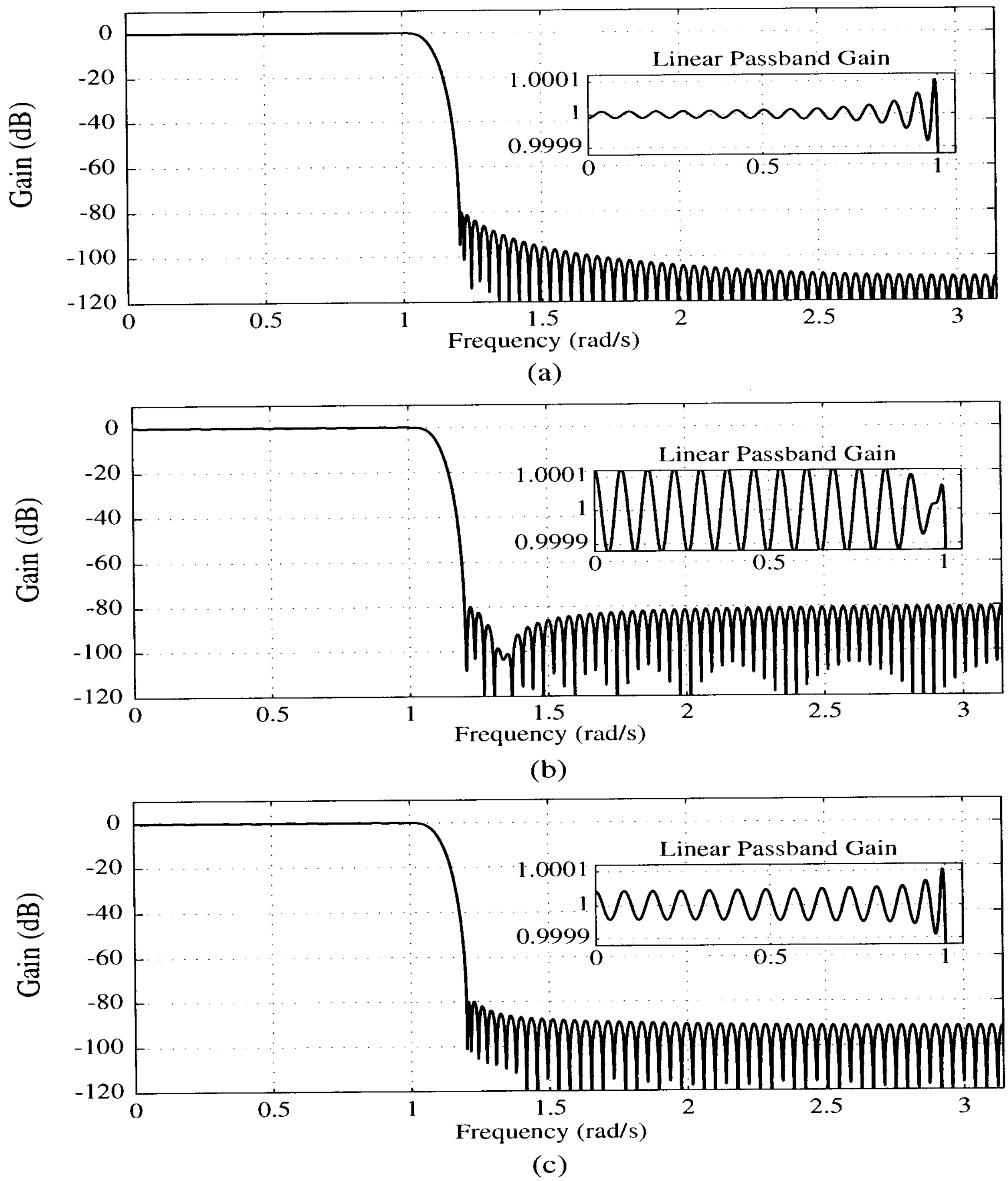


Figure 4.5. *Example 1: Amplitude responses of lowpass filters designed using various window functions. (a) Kaiser window. (b) Dolph-Chebyshev window. (c) Ultraspherical window.*

The adjustable parameters for the Kaiser, Dolph-Chebyshev, and ultraspherical windows were calculated as $\alpha = 3.395$, $\beta = 1.471$, and $\beta = 1.391$, respectively. The additional parameter calculated for the ultraspherical window was $\mu = 0.5960$. The stopband attenuations achieved were 41.34, 37.36, and 38.40 dB with transition bandwidths 0.1963, 0.2027, and 0.2033 rad/s, respectively. Using the modification discussed in Example 1 to improve stopband attenuation accuracy, the re-estimated parameters assume the values $\alpha = 3.235$, $\beta = 1.554$, and $\beta = 1.420$, respectively. The stopband attenuations were 39.92, 40.07, and 39.94 dB with transition bandwidths 0.1905, 0.2188, and 0.2125 rad/s, respectively. The filter lengths required so as to achieve the specifications were $N = 73$ for the Kaiser window, $N = 73$ for the Dolph-Chebyshev window, and $N = 67$ for the ultraspherical window. Figure 4.6 shows the amplitude responses of the designed filters.

Example 3: Design a band-limited differentiator with $\omega_p = 1.0$, $\omega_a = 1.5$ *rad/s and* $A_a = 50$ *dB using the Kaiser, Dolph-Chebyshev, and ultraspherical windows.*

The adjusted design attenuations from Eq. (4.38) for the Kaiser, Dolph-Chebyshev, and ultraspherical windows were calculated as $A_a^{'} = 52.52, 51.32$, and 53.35, respectively. The adjustable parameters were calculated as $\alpha = 4.829$, $\beta = 1.834$, and $\beta = 1.765$, respectively, while the additional parameter for the ultraspherical window was calculated as $\mu = 0.4994$. The design attenuations achieved were 51.95, 47.91, and 49.28 dB with transition bandwidths 0.4931, 0.4959, and 0.4759 rad/s, respectively. Using the modification discussed in Example 1 to improve design attenuation accuracy, the re-estimated parameters assume the values $\alpha = 4.613$, $\beta = 1.990$, and $\beta = 1.785$, respectively. The attenuations were 50.34, 52.86, and 49.93 dB with transition bandwidths 0.4734, 0.5446, 0.4833 rad/s, respectively. The filter lengths required so as to achieve the specifications were $N = 41$ for the Kaiser window, $N = 41$ for the Dolph-Chebyshev window, and $N = 39$ for the ultraspherical window. Figure 4.7 shows the amplitude responses of the designed differentiators.

Example 4: Design a Hilbert transformer with $\omega_{p1} = 0.2$ *rad/s and* $A_p = 80$ *dB using the Kaiser, Dolph-Chebyshev, and ultraspherical windows.*

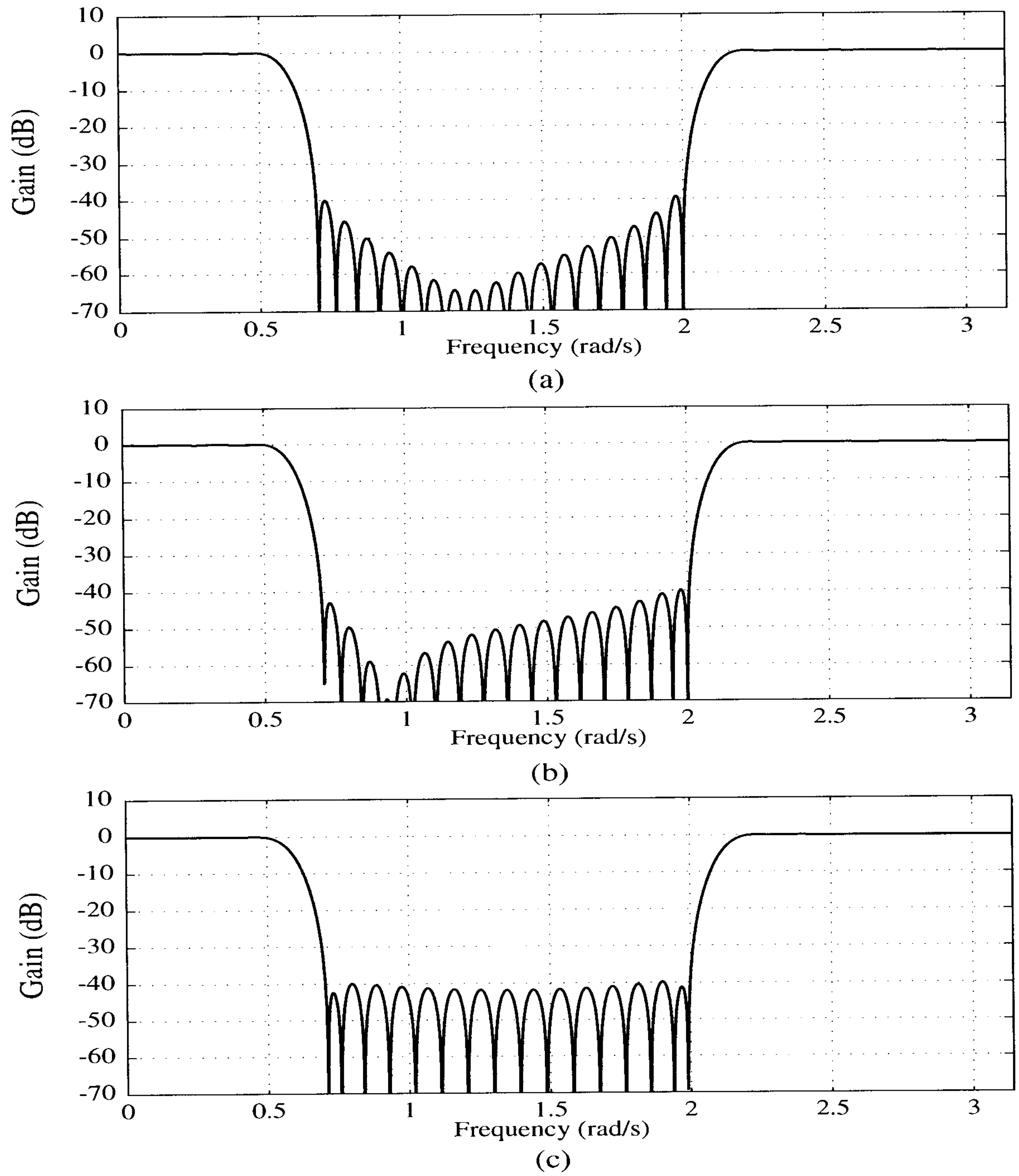


Figure 4.6. *Example 2: Amplitude responses of bandsop filters designed using various window functions. (a) Kaiser window. (b) Dolph-Chebyshev window. (c) Ultraspherical window.*

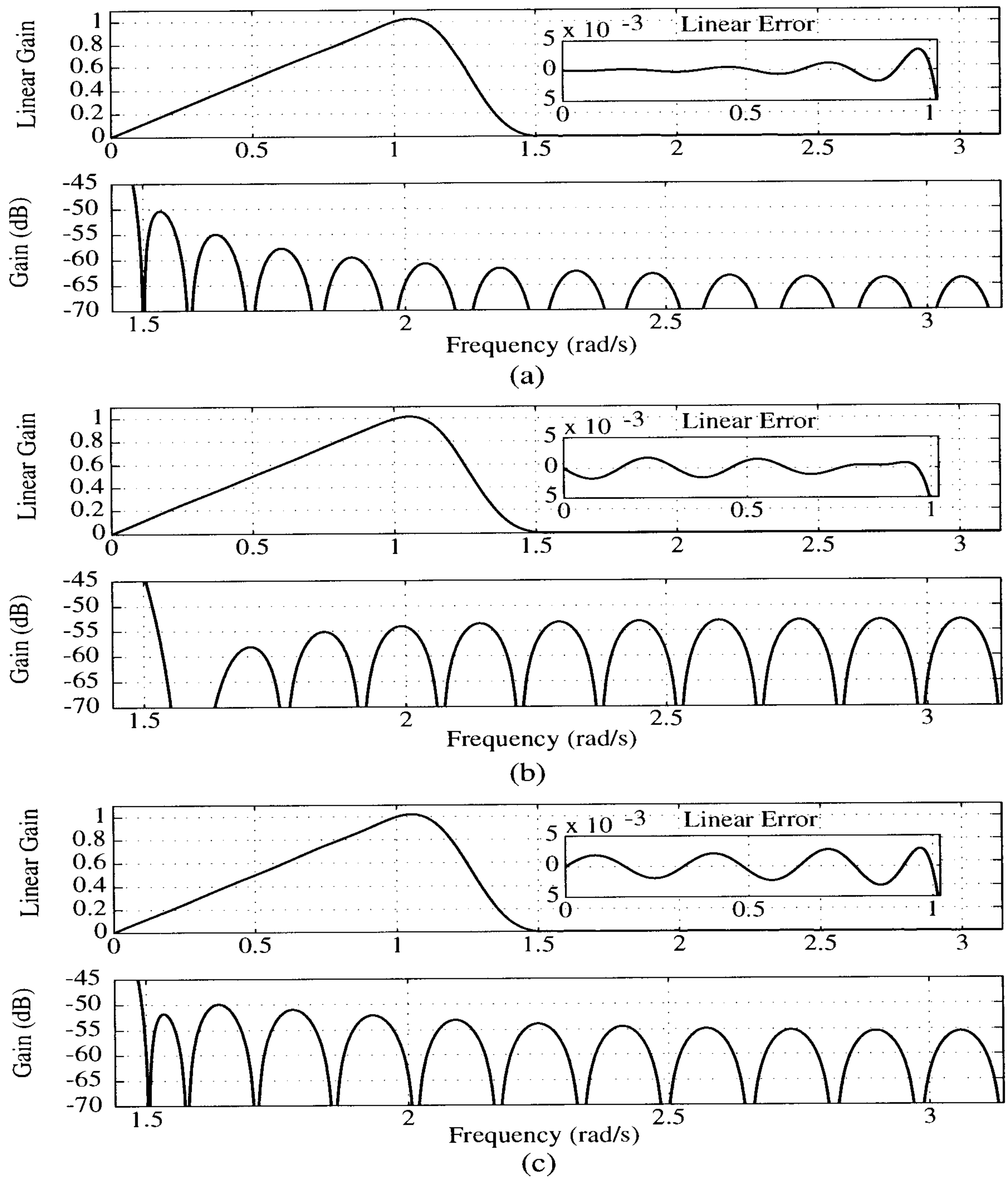


Figure 4.7. *Example 3: Amplitude responses of band-limted differentiators designed using various window functions. (a) Kaiser window. (b) Dolph-Chebyshev window. (c) Ultraspherical window.*

The adjusted design attenuations from Eq. (4.45) for the Kaiser, Dolph-Chebyshev, and ultraspherical windows were calculated as $A_a^{'} = 86.41$, 85.24, and 86.46, respectively. The adjustable parameters were calculated as $\alpha = 8.564$, $\beta = 2.983$, and $\beta = 2.789$, respectively, while the additional parameter for the ultraspherical window was calculated as $\mu = 0.6445$. The design attenuations achieved were 80.12, 79.59, and 79.39 dB with transition bandwidths 0.3941, 0.3889, and 0.3847 rad/s, respectively. Using the modification discussed in Example 1 to improve design attenuation accuracy, the re-estimated parameters assume the values $\alpha = 8.550$, $\beta = 2.997$, and $\beta = 2.809$, respectively. The attenuations were 79.96, 80.01, and 79.93 dB with transition bandwidths 0.3935, 0.3912, and 0.3879 rad/s, respectively. The filter lengths required so as to achieve the specifications were $N = 88$ for the Kaiser window, $N = 90$ for the Dolph-Chebyshev window, and $N = 86$ for the ultraspherical window. Figure 4.8 shows the amplitude responses of the designed Hilbert transformers.

4.9 Conclusions

An efficient method for designing nonrecursive digital filters based on the ultraspherical window has been proposed. Economies in computation are achieved in two ways. First, by using the efficient formulation of the window coefficients presented in Chapter 2, the amount of computation required is reduced to a small fraction of that required by standard methods. Second, the filter length and the independent window parameters that would be required so as to achieve prescribed specifications in lowpass, highpass, bandpass, and bandstop filters as well as in digital differentiators and Hilbert transformers are efficiently determined through empirical formulas. The ultraspherical window yields lower-order filters relative to designs obtained using other windows yielding on the average a reduction of 3.07% relative to the Kaiser window, 2.86% relative to the Saramäki window, and 5.30% relative to the Dolph-Chebyshev window. Alternatively, for a fixed filter length, the ultraspherical window increases the stopband attenuation relative to the other windows achieving

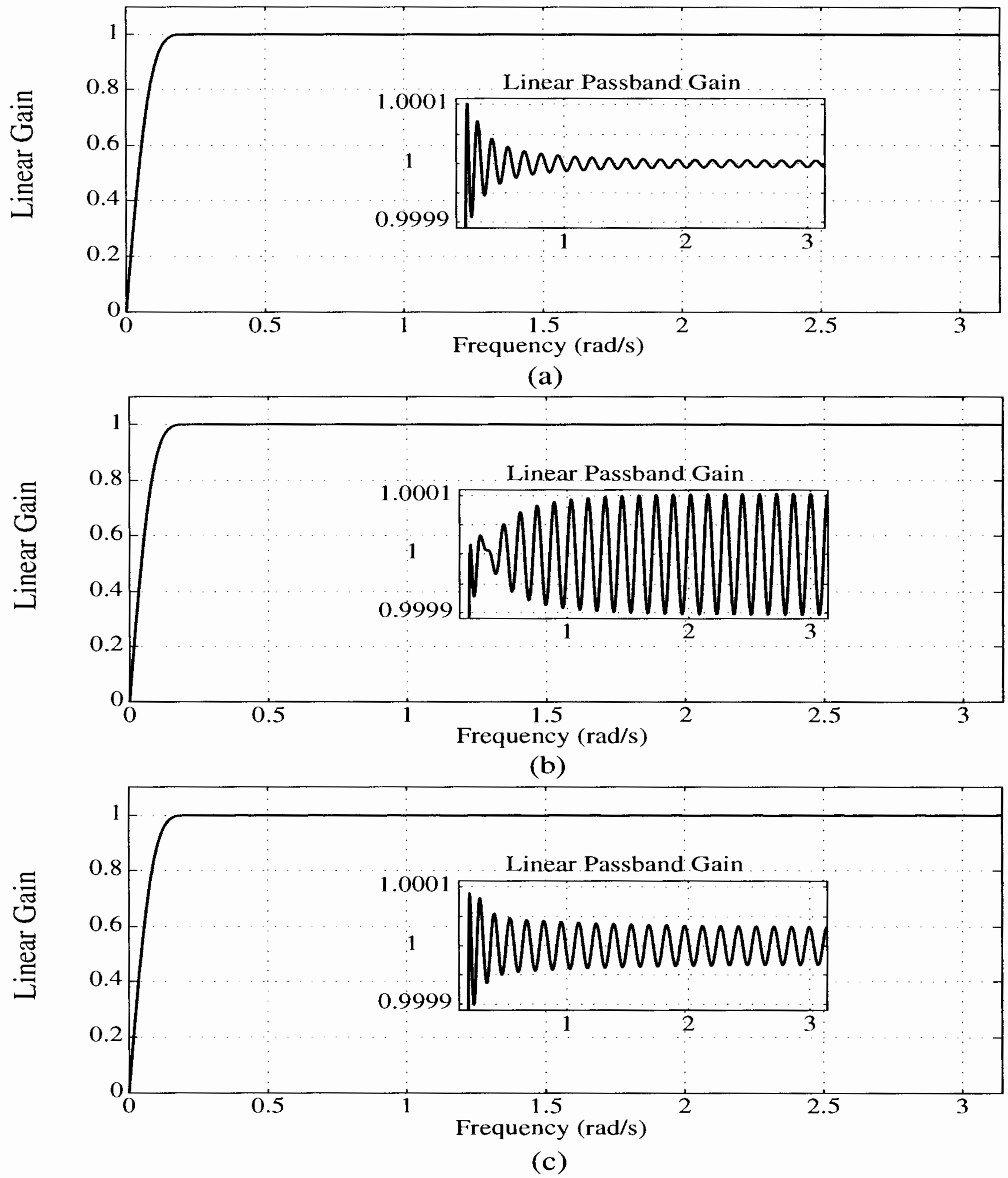


Figure 4.8. *Example 4: Amplitude responses of Hilbert transformers designed using various window functions. (a) Kaiser window. (b) Dolph-Chebyshev window. (c) Ultraspherical window.*

on the average an increase of 2.61 dB relative to the Kaiser window, 2.42 dB relative to the Saramäki window, and 4.49 dB relative to the Dolph-Chebyshev window. On the other hand, the weighted-Chebyshev method increases the stopband attenuation relative to that of the ultraspherical window by about 2.76 dB on the average; however, the computation required by the weighted-Chebyshev method is far greater than that required by the proposed method.

Chapter 5

An Efficient Closed-Form Design Method for Cosine-Modulated Filter Banks

5.1 Introduction

A fundamental system used in multirate applications is the M-channel maximally-decimated filter bank. For cosine-modulated filter banks (CMFBs) the analysis and synthesis filters are cosine-modulated versions of a lowpass prototype filter which is typically designed to minimize three error components (amplitude distortion, phase distortion, and aliasing) that are inherent in the system. An early design method for CMFBs was proposed by Creusere and Mitra [46] who used the weighted-Chebyshev method to design the prototype filter. Next, Lin and Vaidyanathan [47] used the Kaiser window approach to design the prototype filter while, recently, Cruz-Roldán et al. [48] modified this method to include other windows. These methods are iterative and, therefore, they are not suitable for applications where the design must be carried out in real or quasi-real time such as in the perceptual coding of digital audio [44] and heartbeat detection in ECG signals [45]. For such applications a closed-form window method is preferred.

In this chapter, a closed-form method for designing prototype filters for M-channel CMFBs so as to achieve a prescribed stopband attenuation and channel overlap is proposed.

The chapter is structured as follows. Section 5.2 reviews the design of CMFBs using the window method. Section 5.3 presents a closed-form method for designing prototype filters for CMFBs using the ultraspherical window. Section 5.4 presents a design example and compares the reconstruction error of CMFBs designed by the proposed method as well as the amount of computation required to design them with those of CMFBs designed by other known methods.

5.2 Design of CMFBs Using the Window Method

The input-output relations for an M-channel CMFB are given by [49]

$$Y(z) = T_0(z)X(z) + \sum_{l=1}^{M-1} T_l(z)X(zW_M^l) \tag{5.1}$$

where $W_M = e^{-j2\pi/M}$ and

$$T_0(z) = \frac{1}{M}\sum_{k=0}^{M-1} F_k(z)H_k(z) \tag{5.2}$$

$$T_l(z) = \frac{1}{M}\sum_{k=0}^{M-1} F_k(z)H_k(zW_M^l) \quad \text{for } l = 1, \dots, M-1 \tag{5.3}$$

In the above expressions, $T_0(z)$ is the transfer function of the filter bank, $T_l(z)$ for $1 \leq l \leq M-1$ are the aliasing transfer functions, and $H_k(z)$ and $F_k(z)$ are the individual transfer functions of the analysis and synthesis filters, respectively. The impulse responses for the analysis and synthesis filters are of length N and are given by [49]

$$h_k(n) = 2h_p(n)\cos\left[\omega_k\left(n - \frac{N-1}{2}\right) + \theta_k\right] \tag{5.4}$$

$$f_k(n) = h_k(N-1-n) \tag{5.5}$$

for $k = 0, 1, \dots, M-1$. $h_p(n)$ is the impulse response of a linear-phase nonrecursive prototype filter of length N and

$$\omega_k = \frac{(2k+1)\pi}{2M} \quad \text{and} \quad \theta_k = (-1)^k\frac{\pi}{4} \tag{5.6}$$

The reconstruction errors are measured by examining the frequency response of the filter bank. First, there is no phase distortion in the system because the filter bank has a linear phase characteristic. Second, the error in the amplitude response of the filter bank is given by

$$e_m(\omega) = 1 - |T_0(e^{j\omega})| \quad \text{for } \omega \in [0, \pi] \tag{5.7}$$

Third, the worst-case and total aliasing distortion of the filter bank are given by

$$e_a(\omega) = \max_{1 \le l \le M-1} |T_l(e^{j\omega})| \quad \text{for } \omega \in [0, \pi] \tag{5.8}$$

and

$$e_{ta}(\omega) = \left[\sum_{l=1}^{M-1} |T_l(e^{j\omega})|^2 \right]^{1/2} \quad \text{for } \omega \in [0, \pi] \tag{5.9}$$

respectively.

Small reconstruction errors can be achieved by designing the lowpass prototype filter using the window method as described in Chapter 4. Prescribed stopband attenuation and channel overlap can be achieved by requiring the nonrecursive (noncausal) prototype filter to satisfy the lowpass filter specification given by Eq. (4.6). Closed-form methods for constructing the ultraspherical, Kaiser, and Saramäki windows so as to achieve prescribed lowpass specifications can be found in Chapter 4 of [1] and in [8], respectively.

5.3 Efficient Design of Prototype Filter

The conditions for approximate reconstruction in a filter bank can be stated in terms of the frequency response of the prototype filter as [49]

$$|H_p(e^{j\omega})| \approx 0 \quad \text{for } |\omega| > \pi/M \tag{5.10}$$

$$|T_0(e^{j\omega})| = \frac{1}{M} \sum_{k=0}^{2M-1} \left| H_p\left(e^{j(\omega - k\pi/M)}\right) \right|^2 \approx 1 \tag{5.11}$$

The degree to which the first approximation is achieved influences the aliasing errors incurred in Eqs. (5.8) and (5.9) in a directly proportional fashion, i.e., if $|H_p(e^{j\omega})| = 0$ for

$|\omega| > \pi/M$ then no aliasing errors occur. Similarly, the degree to which the second approximation is achieved influences the amplitude-response error in Eq. (5.7) in a directly proportional fashion. One way to reduce the amplitude-response error is to design the prototype filter so that the square of its amplitude response is approximately the same as the square of the amplitude response of a Nyquist (or $2M$-band) filter. By doing this, the neighboring shifted versions of the prototype filter's amplitude response in Eq. (5.11) overlap so that $|T_0(e^{j\omega})|$ is approximately allpass. To achieve this, it is required that $|H_p(e^{j\omega})|^2 \approx 1/2$ at $\omega = \pi/2M$, or equivalently

$$\left|H_p\left(e^{j\omega}\right)\right| \approx 1/\sqrt{2} \tag{5.12}$$

The prototype filter is usually required to have a stopband attenuation of A_a dB and a stopband edge frequency of

$$\omega_a = \frac{(1+\rho)\pi}{2M} \tag{5.13}$$

where ρ is a system design parameter called the *roll-off factor*. Parameter ρ controls the channel overlap as illustrated in Fig. 5.1.

The required design can be obtained by selecting the passband edge frequency ω_p so that $|H_p(e^{j\pi/2M})| \approx 1/\sqrt{2}$. This can be accomplished by solving the optimization problem

$$\min_{\omega_p} F = \left[\left|H_p(e^{j\pi/2M})\right| - 1/\sqrt{2}\right]^2 \tag{5.14}$$

$$\text{subject to:} \quad 0 \leq \omega_p \leq \pi/2M$$

where M, A_a, and ρ are the system design parameters. For each iteration, the computation of the objective function requires the re-estimation of the filter length and window parameters as well as the calculation of the window coefficients and the infinite-duration impulse response of the filter. Simple one-dimensional optimization algorithms such as dichotomous, Fibonacci, or golden section line searches, as outlined in [27], can be used to solve the minimization in Eq. (5.14).

The above design method was used to deduce empirical formulas that can be used to estimate the passband edge frequency ω_p that minimizes the objective function in Eq. (5.14).

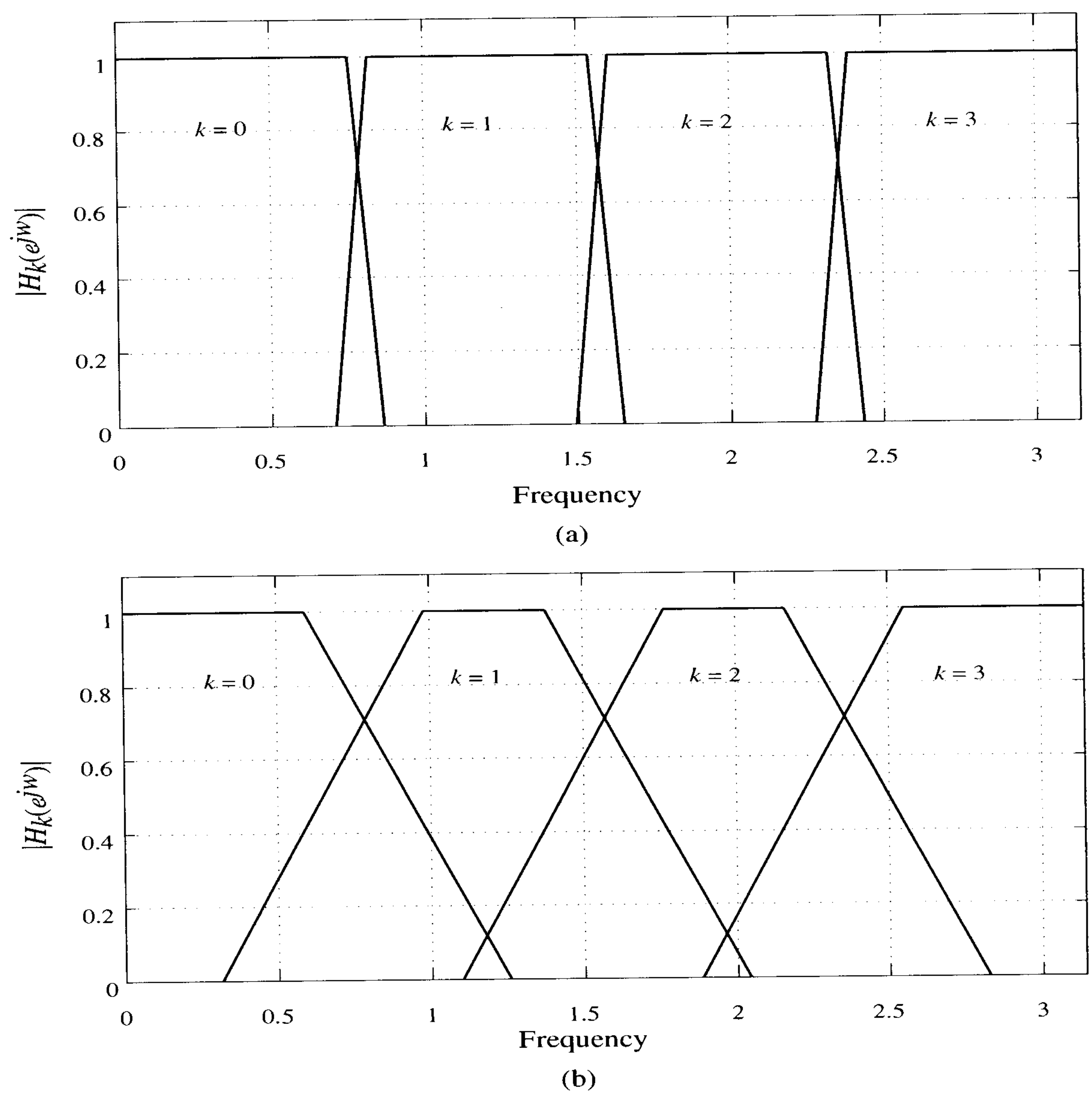


Figure 5.1. *Amplitude responses for the idealized analysis filters with $M = 4$ channels for (a) $\rho = 0.2$ and (b) $\rho = 1.2$.*

Figure 5.2 shows plots of the objective function vs. the performance measure

$$D = \omega_p \times \frac{2M}{\pi} \tag{5.15}$$

for various values of ρ where D is a normalized measure of the passband edge frequency that is approximately independent from the number of channels. By examining a large number of designs, it was observed that the optimal solution for a given ρ remains approximately the same (on the order of 0.01% difference in D) regardless of the number of channels, i.e., F vs. D does not change as M changes. Figure 5.3 shows plots of the resulting optimal values of D vs. ρ for various values of A_a. Through curve fitting, an empirical formula was derived for D in the form

$$D = \begin{cases} a_1\rho^2 + b_1\rho + c_1 & \text{for } 0.5 \leq \rho < 1.0 \\ a_2\rho^2 + b_2\rho + c_2 & \text{for } 1.0 \leq \rho \leq 1.5 \end{cases} \tag{5.16}$$

where the model coefficients for the values $A_a = \{-50, -60, -70, ..., -150\}$ are given in Tables 5.1 and 5.2 for the ultraspherical and Kaiser windows, respectively. Model coefficients for the Saramäki window have also been obtained but are not included. Estimates for D corresponding to values of A_a in the range $[-150, -50]$ that are not included in Tables 5.1 and 5.2 can be obtained using cubic spline interpolation.

The empirical formulas provide an accurate estimate of the optimal ω_p over a range of system design parameters, namely, $M \in \{1, 2, 3, ...\}$, $-150 \leq A_a \leq -50$, and $0.5 \leq \rho \leq 1.5$.

Based on the above findings, an M-channel CMFB with a prescribed stopband attenuation A_a and a prescribed roll-off ratio ρ can be designed by using the ultraspherical, Kaiser, or Saramäki window. A design algorithm that can be used for the case of the ultraspherical window is as follows:

Algorithm 1- Prototype filter design for CMFBs using the ultraspherical window

Step 1: Input M, A_a, and ρ.

Step 2: Calculate the stopband ripple δ_a using Eq. (4.7).

Table 5.1. *Model Coefficients for D for the Ultraspherical Window in Eq. 5.16* $(0.5 \leq \rho \leq 1.5)$

A_a	a_1	b_1	c_1	a_2	b_2	c_2
−50	4.247E−2	−6.463E−1	1.020E+0	1.091E−1	−8.717E−1	1.178E+0
−60	−1.146E−2	−6.098E−1	9.966E−1	−1.090E−2	−6.061E−1	9.942E−1
−70	−3.803E−3	−6.458E−1	9.993E−1	1.126E−2	−6.822E−1	1.021E+0
−80	−2.042E−4	−6.731E−1	1.001E+0	3.961E−3	−6.838E−1	1.008E+0
−90	−1.650E−3	−6.887E−1	1.000E+0	−6.863E−3	−6.757E−1	9.922E−1
−100	−2.006E−3	−7.028E−1	9.998E−1	2.577E−3	−7.131E−1	1.006E+0
−110	−4.775E−3	−7.110E−1	9.982E−1	−2.948E−4	−7.175E−1	1.000E+0
−120	1.256E−3	−7.301E−1	1.001E+0	−3.871E−3	−7.212E−1	9.971E−1
−130	−4.209E−4	−7.383E−1	1.000E+0	2.854E−4	−7.405E−1	1.002E+0
−140	6.519E−4	−7.488E−1	1.001E+0	7.473E−3	−7.655E−1	1.011E+0
−150	−8.206E−4	−7.548E−1	1.000E+0	−5.601E−4	−7.560E−1	1.001E+0

Step 3: Calculate the stopband edge frequency ω_a using Eq. (5.13).

Step 4: Calculate the parameter D in Eq. (5.16) using Table 5.1.

Step 5: Calculate the passband edge frequency ω_p using

$$\omega_p = D \times \frac{\pi}{2M} \tag{5.17}$$

Step 6: Using the parameters δ_a, ω_p, and ω_a, calculate the ultraspherical window's parameters μ, x_μ, and N and then its window coefficients.

Step 7: Calculate the relevant terms of the infinite-duration impulse response using Eq. (4.2) with $\omega_c = (\omega_p + \omega_a)/2$.

Step 8: Obtain the noncausal finite-duration impulse response for the prototype filter using Eq. (4.4).

Algorithms that can be used with the Kaiser and Saramäki windows can be obtained by modifying Algorithm 5.1 in two locations. In Step 4, D in Eq. (5.16) is calculated using

Table 5.2. *Model Coefficients for D for the Kaiser Window in Eq. 5.16* $(0.5 \le \rho \le 1.5)$

A_a	a_1	b_1	c_1	a_2	b_2	c_2
−50	3.096E−2	−6.345E−1	1.015E+0	8.307E−2	−8.016E−1	1.130E+0
−60	2.150E−3	−6.286E−1	1.002E+0	−1.543E−2	−5.931E−1	9.838E−1
−70	−1.173E−3	−6.502E−1	9.998E−1	−1.909E−2	−6.043E−1	9.711E−1
−80	3.107E−4	−6.735E−1	1.000E+0	−3.959E−4	−6.738E−1	1.002E+0
−90	−1.839E−3	−6.880E−1	9.994E−1	−9.398E−3	−6.680E−1	9.866E−1
−100	1.504E−3	−7.074E−1	1.001E+0	7.341E−3	−7.237E−1	1.011E+0
−110	−7.255E−4	−7.167E−1	9.996E−1	1.756E−2	−7.601E−1	1.025E+0
−120	2.536E−3	−7.329E−1	1.001E+0	−9.054E−3	−7.085E−1	9.882E−1
−130	4.979E−4	−7.398E−1	1.000E+0	6.255E−3	−7.547E−1	1.010E+0
−140	2.707E−4	−7.484E−1	1.000E+0	−4.075E−3	−7.403E−1	9.964E−1
−150	−4.951E−4	−7.553E−1	9.999E−1	−1.072E−2	−7.322E−1	9.870E−1

model coefficients that correspond to the Kaiser window (Table 5.2) and the Saramäki window (not included). Second, Step 6 requires equations that yield the window function's parameters and its coefficients. Closed-form equations for these tasks for the Kaiser and Saramäki windows can be found in [1] and [8], respectively.

5.4 Design Example and Comparisons

The proposed method was used to design 32-channel CMFBs with a stopband attenuation of −100 dB. A CMFB satisfying these specifications can be used in the MPEG audio coder, and has been used on a number of occasions to compare different filter-bank design methods [46], [47], [48]. Designs were obtained using the proposed method with the ultraspherical, Kaiser, and Saramäki windows for different values of the roll-off factor ρ. Figure 5.4 shows plots of the amplitude response of the prototype filter, the amplitude response of the filter bank, and the total aliasing error of the filter bank designed using the proposed

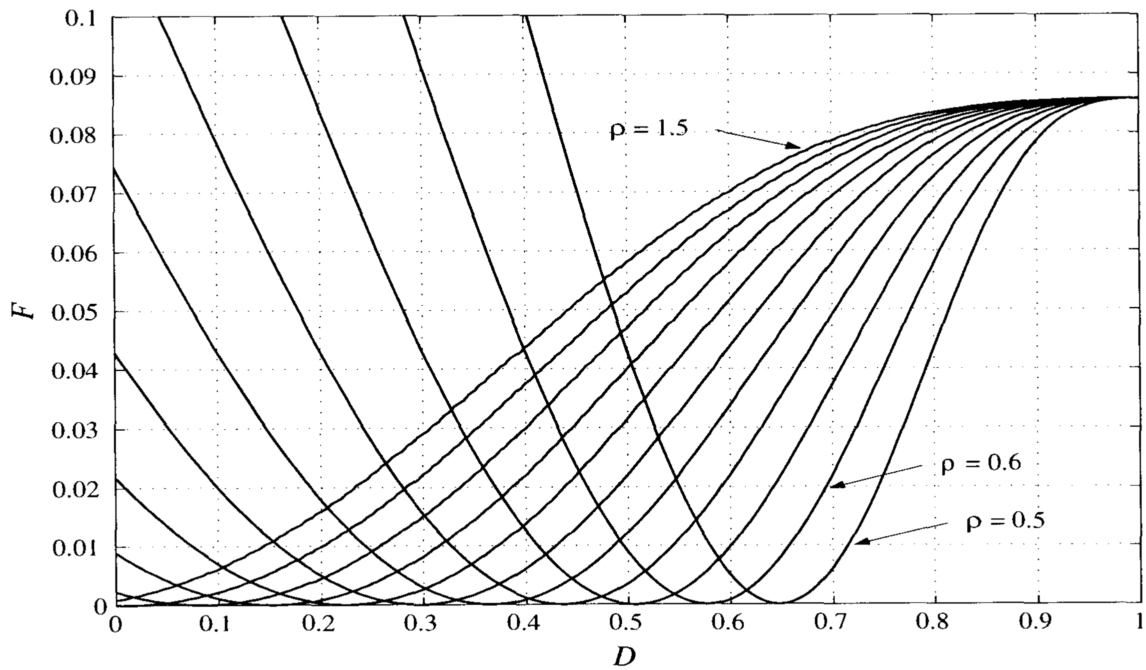


Figure 5.2. *Values of the objective function over the range* $0 \leq D \leq 1$ *for various values of the roll-off factor* ρ *for the ultraspherical window.*

method with an ultraspherical window and $\rho = 1.05$. Table 5.3 lists the maximum reconstruction errors and the prototype filter length associated with a small set of the designed filter banks. Figure 5.5 shows plots of the maximum reconstruction errors of the resulting MPEG-compliant filter banks that were designed over the range $0.5 \leq \rho \leq 1.5$. The plots show that when the channel overlap increases (larger values of ρ) the aliasing errors increase significantly while the amplitude distortion of the filter bank remains relatively constant over the entire range of ρ. The average percentage decrease in error provided by the Kaiser window over the Saramäki and ultraspherical windows was, respectively, 11.69% and 12.17% for $\max|e_m(\omega)|$, 1.34% and 26.51% for $\max|e_a(\omega)|$, and 2.11% and 34.65% for $\max|e_{ta}(\omega)|$. On the average, the Kaiser window provides the smallest reconstruction errors although cases exist where the use of the Saramäki or ultraspherical windows yields smaller reconstruction errors. For instance, when $\rho = 1.2$ Table 5.3 reveals that the use of

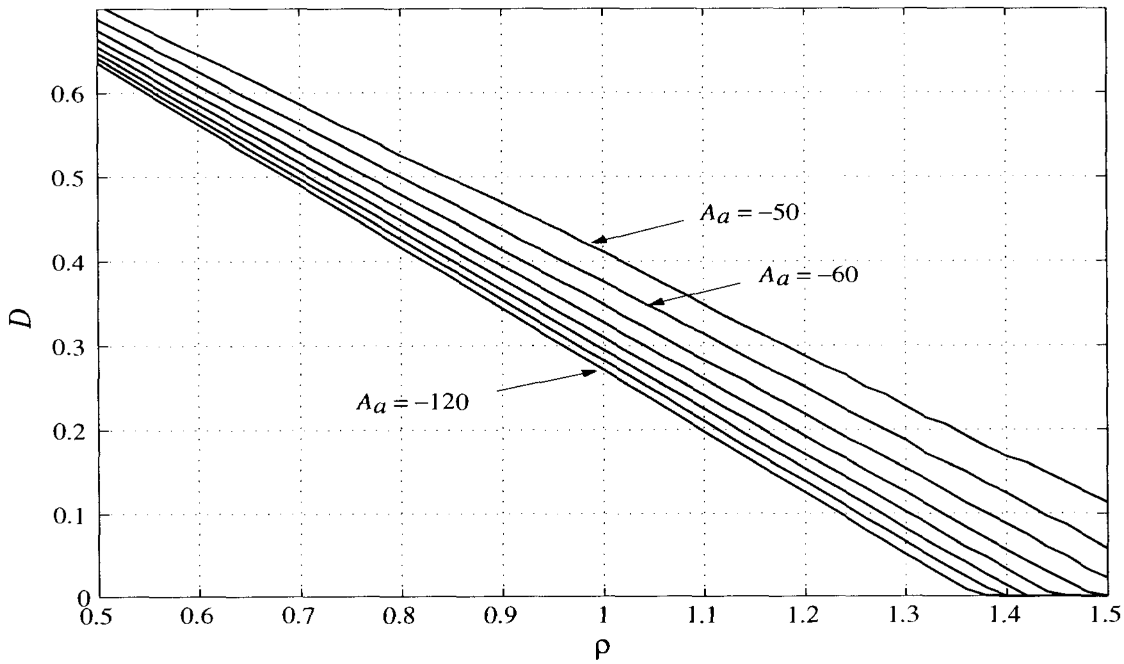


Figure 5.3. *Values of D that minimize the objective function over the range* $0.5 \leq \rho \leq 1.5$ *for various stopband attenuations for the ultraspherical window.*

the Saramäki window yields a reduction in all of the maximum errors relative to the designs obtained using the Kaiser window.

In terms of filter length, the filter banks designed using the ultraspherical window provided an average decrease in the prototype filter length of 3.07% and 2.86% relative to the Kaiser and Saramäki windows, respectively. This is a reflection of the results reported in Chapter 4 which indicate that the ultraspherical window can yield lower-order filters than those designed using the Kaiser or Saramäki windows over a wide range of lowpass filter specifications. The computation time required to design the prototype filter using the proposed method and the methods of [47] and [48] were 0.0470, 3.2868, and 7.6658s, respectively. In effect, the proposed method reduced the amount of computation time to 1.43% of that required by the method in [47] or to 0.61% of that required by the method in [48]. The computation time of the proposed method included costs associated with the

Table 5.3. *Reconstruction Error Comparison for the Design Example*

Design Method	N	$\max\lvert e_m(\omega)\rvert$	$\max\lvert e_a(\omega)\rvert$	$\max\lvert e_{ta}(\omega)\rvert$
Method in [46]	439	$1.80 \cdot 10^{-3}$	$6.13 \cdot 10^{-7}$	$2.54 \cdot 10^{-6}$
Method in [47]	467	$2.42 \cdot 10^{-3}$	$2.67 \cdot 10^{-7}$	$3.86 \cdot 10^{-7}$
Method in [48]	439	$3.06 \cdot 10^{-3}$	$1.85 \cdot 10^{-7}$	$2.61 \cdot 10^{-7}$
Kaiser $\rho = 1.00$	483	$3.13 \cdot 10^{-3}$	$3.73 \cdot 10^{-7}$	$5.52 \cdot 10^{-7}$
Kaiser $\rho = 1.05$	460	$3.23 \cdot 10^{-3}$	$9.78 \cdot 10^{-8}$	$1.48 \cdot 10^{-7}$
Kaiser $\rho = 1.10$	439	$3.40 \cdot 10^{-3}$	$1.84 \cdot 10^{-7}$	$2.60 \cdot 10^{-7}$
Kaiser $\rho = 1.20$	401	$3.85 \cdot 10^{-3}$	$2.38 \cdot 10^{-7}$	$3.80 \cdot 10^{-7}$
Saramäki $\rho = 1.00$	482	$3.90 \cdot 10^{-3}$	$3.86 \cdot 10^{-7}$	$5.63 \cdot 10^{-7}$
Saramäki $\rho = 1.05$	459	$3.53 \cdot 10^{-3}$	$9.13 \cdot 10^{-8}$	$1.39 \cdot 10^{-7}$
Saramäki $\rho = 1.10$	438	$3.90 \cdot 10^{-3}$	$1.85 \cdot 10^{-7}$	$2.62 \cdot 10^{-7}$
Saramäki $\rho = 1.20$	402	$3.68 \cdot 10^{-3}$	$2.29 \cdot 10^{-7}$	$3.73 \cdot 10^{-7}$
ultraspherical $\rho = 1.00$	468	$3.85 \cdot 10^{-3}$	$6.63 \cdot 10^{-7}$	$9.44 \cdot 10^{-7}$
ultraspherical $\rho = 1.05$	446	$3.80 \cdot 10^{-3}$	$1.58 \cdot 10^{-7}$	$2.26 \cdot 10^{-7}$
ultraspherical $\rho = 1.10$	425	$4.41 \cdot 10^{-3}$	$2.63 \cdot 10^{-7}$	$4.34 \cdot 10^{-7}$
ultraspherical $\rho = 1.20$	390	$4.20 \cdot 10^{-3}$	$4.56 \cdot 10^{-7}$	$1.17 \cdot 10^{-6}$

spline interpolation algorithm. The prototype filter designed using the method in [46] had reduced maximum amplitude error, increased maximum aliasing error, and increased maximum total aliasing error relative to that produced by the other methods (see Table 5.3). This method yields a prototype filter of reduced length, as may be expected, but a huge amount of computation is required for the design due to the repeated use of the Remez exchange algorithm, which makes this method impractical for applications where designs must be carried out in real or quasi-real time. In addition, it has been reported that significant human intervention in the optimization process is required because the algorithm frequently fails to converge to the global minimum [46].

The proposed method was used to design 16-channel CMFBs with $\rho = 1.00$ for stop-

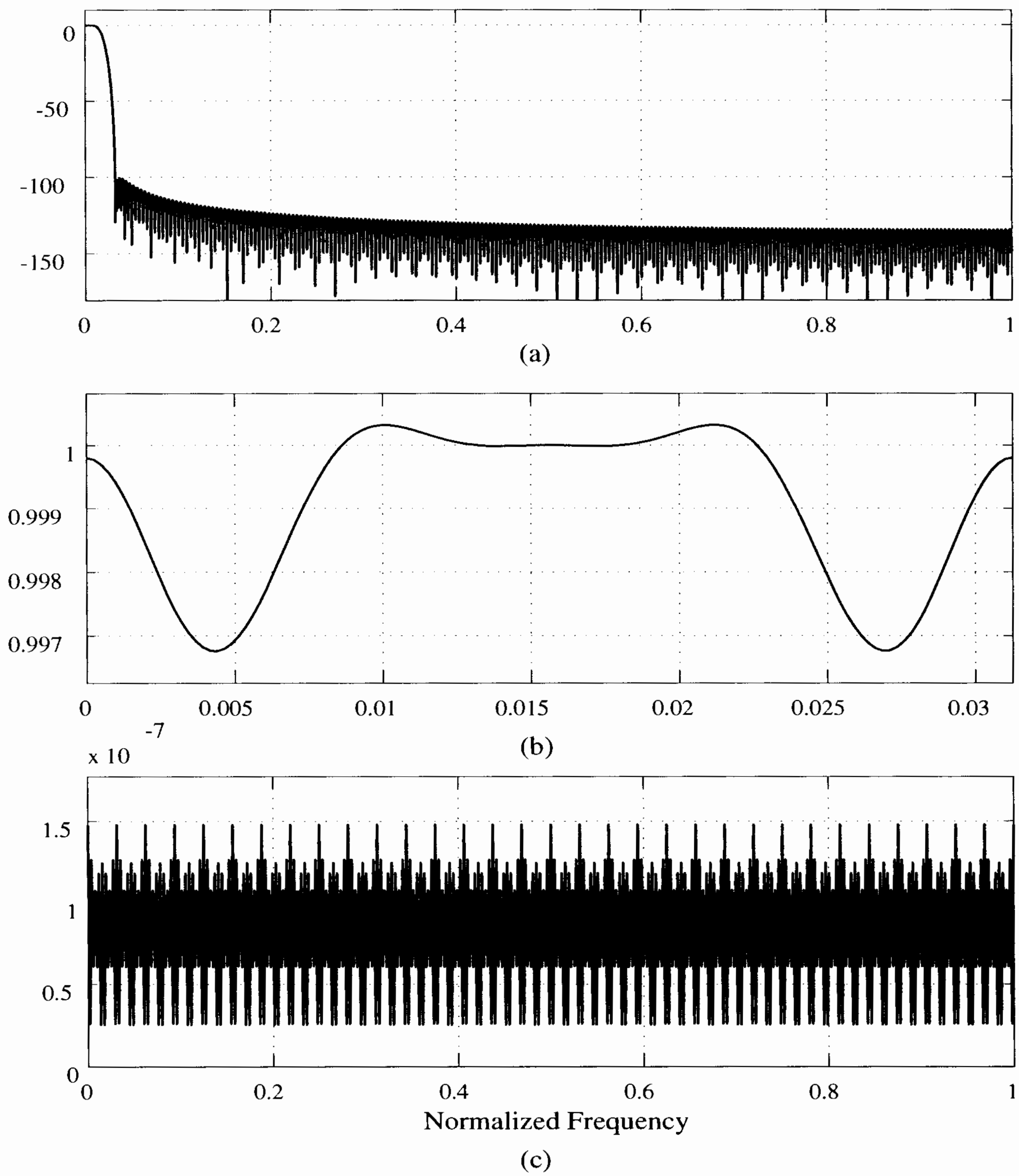


Figure 5.4. *Performance of the CMFB designed using the proposed method with the design parameters* $M = 32$, $A_a = -100$ *dB, and* $\rho = 1.05$ *for the ultraspherical window. (a) Amplitude response of the prototype filter in dB. (b)* $|T_0(e^{j\omega})|$ *over* $[0, \pi/M]$. *(c) Total aliasing error* $e_{ta}(\omega)$.

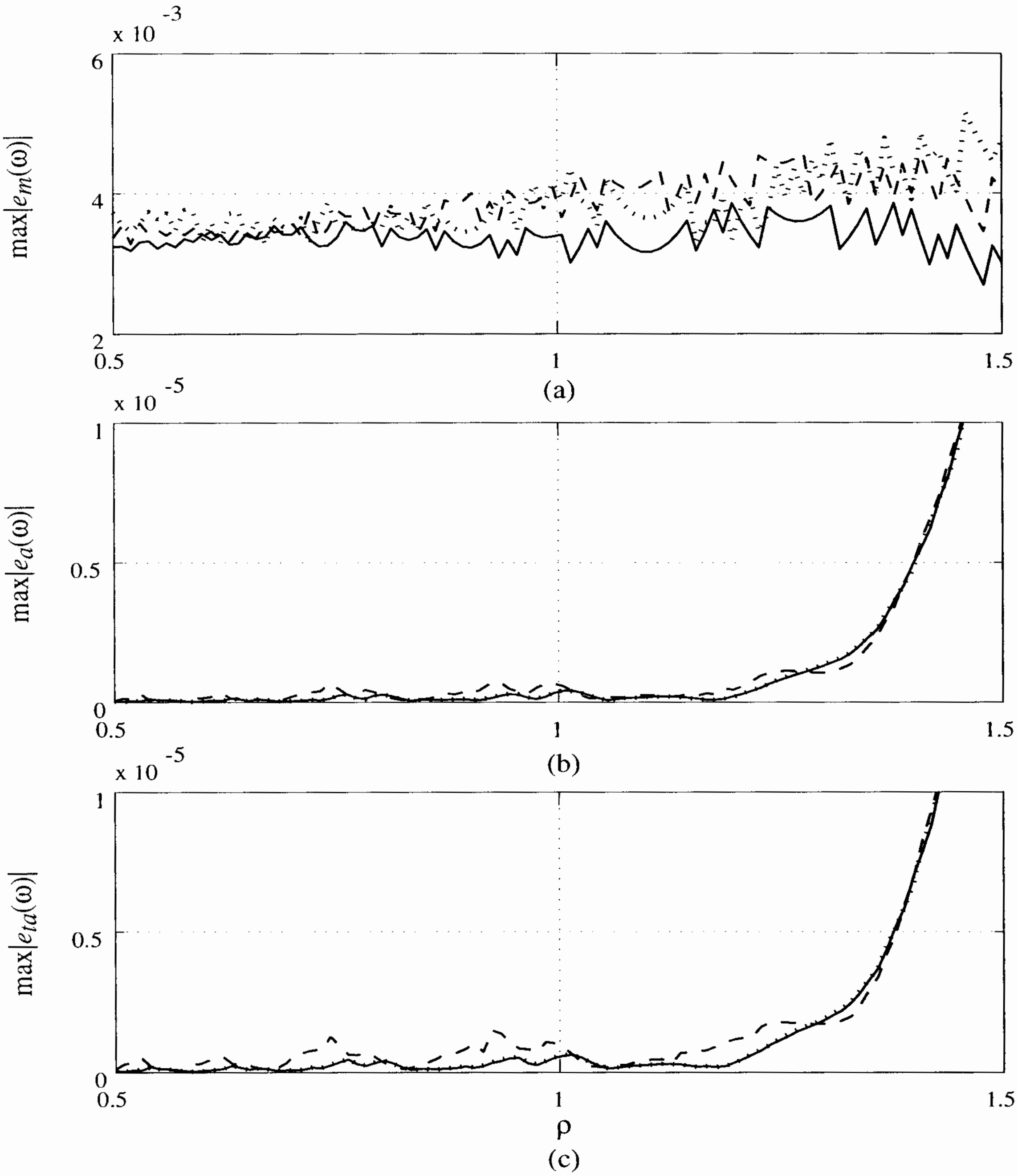


Figure 5.5. *CMFB designed using the proposed method with $A_a = 100$ dB and $M = 32$ for the ultraspherical (dashed line), Kaiser (solid line), and Saramäki (dotted line) windows. (a) Maximum amplitude distortion. (b) Maximum aliasing distortion. (c) Maximum total aliasing distortion.*

band attenuations varying over the range $[-150, -50]$. The roll-off factor was chosen to be 1.00 so that only channels which are immediately adjacent to one another overlap, i.e., for a given channel two channels (one on either side) will overlap (see Fig. 5.1). CMFBs satisfying the same specifications were then designed using the methods of [47] and [48]. Note that the prototype filter length required by the proposed method for a given set of design specifications was found to be the same as that required by the methods of [47] and [48]. This is due to the fact that the prototype filter length is equal to the length of the window function, which is calculated from an empirical equation that is based on the stopband attenuation and channel overlap of the system. Figure 5.6 shows plots of the percentage difference in the maximum reconstruction errors for the CMFBs designed with the proposed method relative to those of the designs obtained with the other two methods, which are given by

$$A = \frac{\max|e_m(\omega)|_{other} - \max|e_m(\omega)|_{prop.}}{\max|e_m(\omega)|_{prop.}} \times 100\% \tag{5.18}$$

$$B = \frac{\max|e_a(\omega)|_{other} - \max|e_a(\omega)|_{prop.}}{\max|e_a(\omega)|_{prop.}} \times 100\% \tag{5.19}$$

$$C = \frac{\max|e_{ta}(\omega)|_{other} - \max|e_{ta}(\omega)|_{prop.}}{\max|e_{ta}(\omega)|_{prop.}} \times 100\% \tag{5.20}$$

When compared with the methods of [47] and [48], the proposed method was found to provide an average increase of 9.53% and 1.52% in the maximum amplitude error, an average increase of 0.011% and decrease of 0.007% in the maximum aliasing error, and an average increase of 0.062% and 0.035% in the maximum total aliasing error, respectively. However, the proposed method was found to reduce computational costs associated with the design of the prototype filter to 1.36% of that required by the method in [47] or to 0.68% of that required by the method in [48] on the average.

5.5 Conclusions

An efficient closed-form method for the design of M-channel cosine-modulated filter banks using the window method so as to achieve prescribed stopband attenuation in the subbands

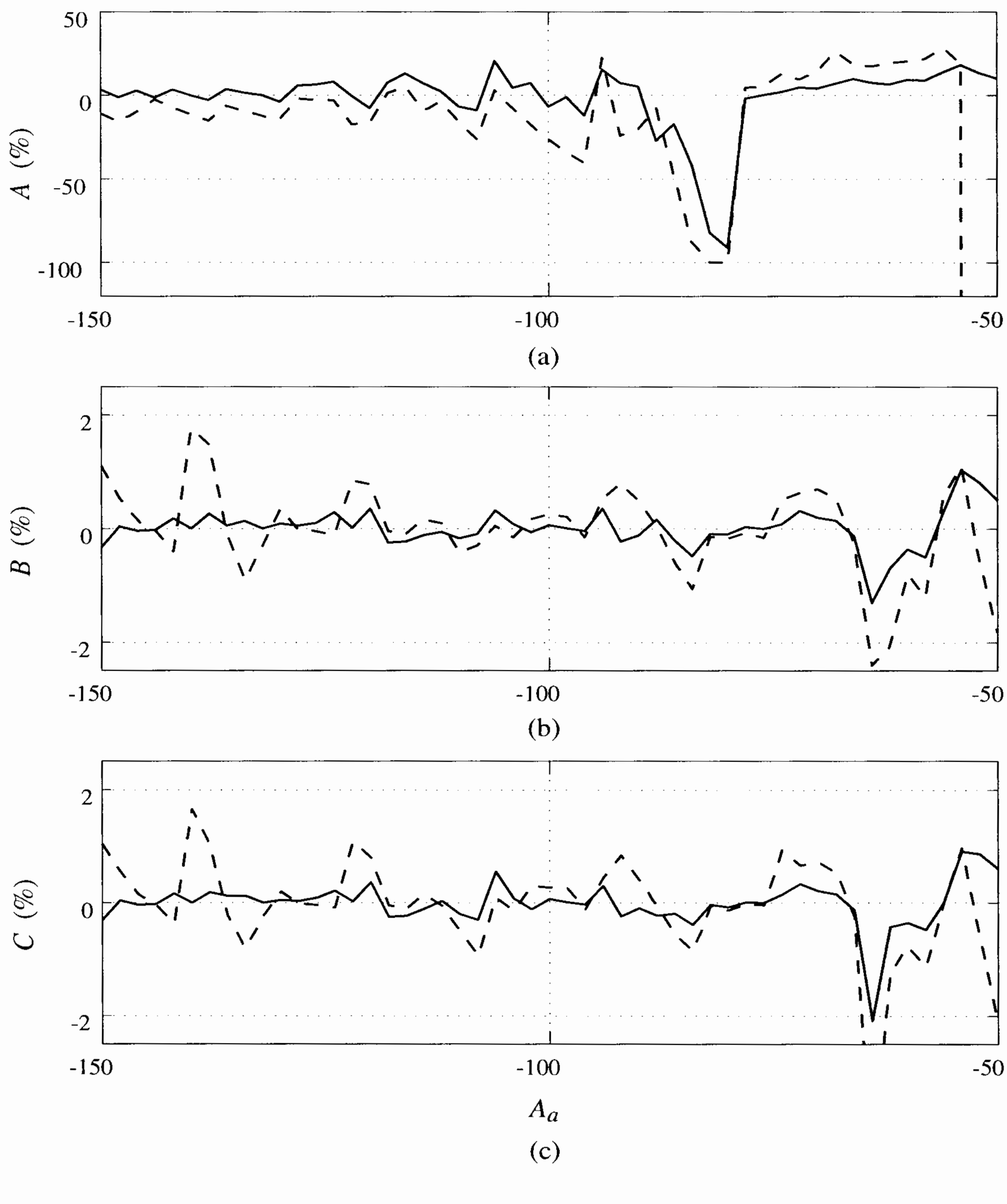


Figure 5.6. *Percentage difference in the maximum reconstruction errors for the proposed method relative to that produced by the methods of [47] (dashed line) and [48] (solid line).*

and channel overlap was proposed. The method is based on empirical formulas that give the required filter length and window parameters that would satisfy the prescribed specifications. Experimental results have shown that, on the average, use of the Kaiser window yields filter banks with the smallest reconstruction error achieving an average percentage decrease in error over the Saramäki and ultraspherical windows of, respectively, 11.69% and 12.17% for the maximum amplitude error in the filter bank, 1.34% and 26.51% for the maximum aliasing error in the filter bank, and 2.11% and 34.65% for the maximum total aliasing error in the filter bank. On the other hand, use of the ultraspherical window yields filter banks with the least amount of design computational complexity (due to the efficient formulation proposed in Chapter 2) and prototype filters with the shortest length (as described in Chapter 4). When compared with the window-based methods of [47] and [48], the proposed method, on the average, increased the average maximum amplitude error by 9.53% and 1.52%, respectively, provided almost no change in the average aliasing error and the average total aliasing error, and produced prototype filters of the same length. However, the computation required is usually a small fraction, less than 2% on the average, of that required by the other methods making it very suitable for applications where the design must be carried out in real or quasi-real time.

Chapter 6

Application of Windows to the STDFT Method for Gene Prediction

6.1 Introduction

The gene-prediction problem involves identifying and locating genes in an organism's genome. Over the past decade deoxyribonucleic acid (DNA) genomes for many organisms have been sequenced and effective methods for gene prediction are necessary to determine their functional capabilities. DNA sequences comprise long chains of nucleotides. There are four different kinds of nucleotides which can be represented using the four-character alphabet A, T, C, and G (adenine, thymine, cytosine, and guanine, respectively). DNA sequences can be separated into two regions known as genes and intergenic spaces. Genes are further separated into two regions known as exons (coding regions) and introns (noncoding regions).

The gene-prediction problem encompasses the problem of identifying coding regions, which direct the formation of proteins in eucaryotes (cells with a nucleus). A diverse range of gene-prediction methods exist that employ pattern recognition techniques, hidden Markov models, neural networks, and other techniques. A good survey of these methods which includes a discussion of their attributes can be found in [50]. A recent method for gene prediction uses DSP-based techniques to locate the well known *period-three (P-3) property* in DNA sequences [51]-[53]. The P-3 property originates from the codon's triplet structure and is mostly present in coding regions but is absent elsewhere.

Gene prediction on the basis of the P-3 property can be carried out by using the discrete Fourier transform (DFT) in conjunction with the window technique [1] in a method that is often referred to as the short-time DFT (STDFT) method. In recent work on gene prediction based on the P-3 property, the window used with the STDFT method has largely been the rectangular window [51], [52], [53], [54]. Unfortunately the amplitude spectrum of the rectangular window provides low side-lobe attenuation relative to the main lobe, which makes this window a poor choice for suppressing background noise that is common in DNA sequences. Use of the rectangular window can cause noncoding regions to be inadvertently identified as coding regions. This problem can be remedied to some degree by employing windows with increased side-lobe attenuation.

In this chapter, the application of the ultraspherical window along with the STDFT method for gene identification based on the well known period-three property is explored. The chapter is structured as follows. Section 6.2 describes the STDFT method as applied to gene prediction. Section 6.3 explores the application of windows and provides examples and comparisons. Section 6.4 provides concluding remarks.

6.2 Application of STDFT Method for Gene Prediction

The STDFT method for gene prediction comprises three steps. First, the DNA character sequence is converted to a numeric sequence. Second, a measure for the P-3 property is calculated for a portion of the DNA sequence, i.e., for a window of nucleotides. Third, the window is successively shifted along the DNA sequence and the measure is re-calculated for each window of nucleotides.

The DNA character sequence is converted to a numeric sequence by creating a binary indicator sequence $u_\alpha(n)$ for each of the four nucleotides $\alpha = \{A, T, C, G\}$ [51]. These sequences are constructed by using a 1 or 0 to indicate the presence or absence, respectively, of a given nucleotide (see Table 6.1). This mapping preserves correlations between coding

Table 6.1. *Binary Indicator Sequences*

$x(n)$	G	G	A	T	A	T	C	A	C
$u_A(n)$	0	0	1	0	1	0	0	1	0
$u_T(n)$	0	0	0	1	0	1	0	0	0
$u_C(n)$	0	0	0	0	0	0	1	0	1
$u_G(n)$	1	1	0	0	0	0	0	0	0

regions and does not introduce false correlations between character symbols.

One measure for the P-3 property is the total energy in the Fourier spectrum of a DNA sequence, which is the sum of the individual power spectra of each binary indicator sequence [51], i.e.,

$$F_{1w}(k) = |U_A(k)|^2 + |U_T(k)|^2 + |U_C(k)|^2 + |U_G(k)|^2 \tag{6.1}$$

where $U_\alpha(k)$ is the windowed-DFT of $u_\alpha(n)$ over N samples given by

$$U_\alpha(k) = \sum_{n=0}^{N-1} w(n)u_\alpha(n)e^{-j2\pi kn/N} \quad \text{for } 0 \leq k \leq N-1 \tag{6.2}$$

where $w(n)$ is a right-sided window of length N. The P-3 property can be identified by plotting the windowed-DFT. If the P-3 property is present, the DFT coefficient $U_\alpha(N/3)$ is significantly larger than the surrounding DFT coefficients. Consequently, $F_{1w}(N/3)$ is large in a coding region. Spectral estimates that avoid 'picket fence' effects are obtained when N is a multiple of three.

A second measure for the P-3 property is given by [52]

$$F_{2w}(k) = \left| \frac{1}{N} \left[aU_A(k) + tU_T(k) + cU_C(k) + gU_G(k) \right] \right|^2 \tag{6.3}$$

where

$$a = 0.10 + 0.12j \quad t = -0.30 - 0.20j$$
$$c = 0 \qquad g = 0.45 - 0.19j$$

Parameters a, t, c, and g were calculated to maximize the discrimination between coding regions in the organism *S.cerevisiae* and random DNA sequences. Like $F_{1w}(N/3)$, $F_{2w}(N/3)$ is large in a coding region.

To achieve base-domain resolution, the STDFT representation of $F_{1w}(N/3)$ and $F_{2w}(N/3)$ was used, which is obtained by calculating one of the two measures for a data window of N samples, sliding the window by one or more samples, and recalculating the measure.

Any one of several windows can be used in the STDFT method [1] but recent work on the application of the method for gene prediction has focused on the use of the rectangular window. Window lengths can range from a few hundred to a few thousand; however longer windows can compromise the base-domain resolution while shorter windows tend to increase the level of noise. A window length of 351 has been found to provide good experimental results when using the rectangular window [51].

Below we explore the application of some of the more powerful windows in the above method in order to achieve increased accuracy in gene prediction. The windows considered are the ultraspherical, Kaiser, Dolph-Chebyshev, and Saramäki windows.

6.3 Examples and Comparisons

The STDFT method was used to analyze coding regions in gene F56F11.4 in the *C.elegans* chromosome III over the bases 7021 to 15120. This 8100-length DNA sequence, which was obtained using the Nucleotide Accession Number 'AF099922' at the Genbank website, contains five known coding regions at the base locations listed in Table 6.2 relative to 7021. Figures 6.1a and 6.1b show plots of $F_{1w}(N/3)$ for the rectangular window and Kaiser window with $\alpha = 3.0$, designated as $F_{1R}(N/3)$ and $F_{1K}(N/3)$; and Figures 6.1c and 6.1d show plots of $F_{2w}(N/3)$ for the rectangular window and Kaiser window with $\alpha = 3.0$, designated as $F_{2R}(N/3)$ and $F_{2K}(N/3)$, respectively. All windows were of length 351 and the DNA sequence was zero padded on either end to ensure that the STDFT representation corresponds to the center of the window. Figure 6.2 illustrates the normalized amplitude

Table 6.2. *The Five Coding Regions in Gene F56F11.4*

Exon	Relative Location	Length
1	929-1135	207
2	2528-2857	330
3	4114-4377	264
4	5465-5644	180
5	7255-7605	351

spectrum for the rectangular and Kaiser windows. The Kaiser window provides 23.81 dB of relative side-lobe attenuation with a main-lobe width of 0.0496 rad/s. On the other hand, the rectangular window provides 13.26 dB of relative side-lobe attenuation with a main-lobe width of 0.0355 rad/s.

$F_{1R}(N/3)$ shows four discernible peaks for coding regions 2, 3, 4, and 5 but the first coding region is unrecognizable. Measure $F_{1K}(N/3)$ shows five discernible peaks for all coding regions but noise located around the first coding region is still large. Conversely, measures $F_{2R}(N/3)$ and $F_{2K}(N/3)$ show five discernible peaks which identify all of the coding regions. To evaluate the accuracy of the measures in predicting the coding regions, we have employed the signal-to-noise (SNR) performance metric

$$\text{SNR} = \frac{\sum_{n_0 \in [\text{coding regions}]} Y(n_0)}{\sum_{n_1 \in [\text{noncoding regions}]} Y(n_1)} \tag{6.4}$$

where n is the nucleotide location at the center of the data window and $Y(n)$ is the value of a P-3 property measure, e.g., $F_{1R}(N/3)$, located at nucleotide n on the DNA sequence. This metric takes into account base-domain resolution and has been shown to be an important factor in the detectability of coding regions [54]. Table 6.3 lists the SNR values achieved for each measure. For both gene-prediction measures, the Kaiser-windowed versions increased the SNR relative to their rectangular-windowed counterparts.

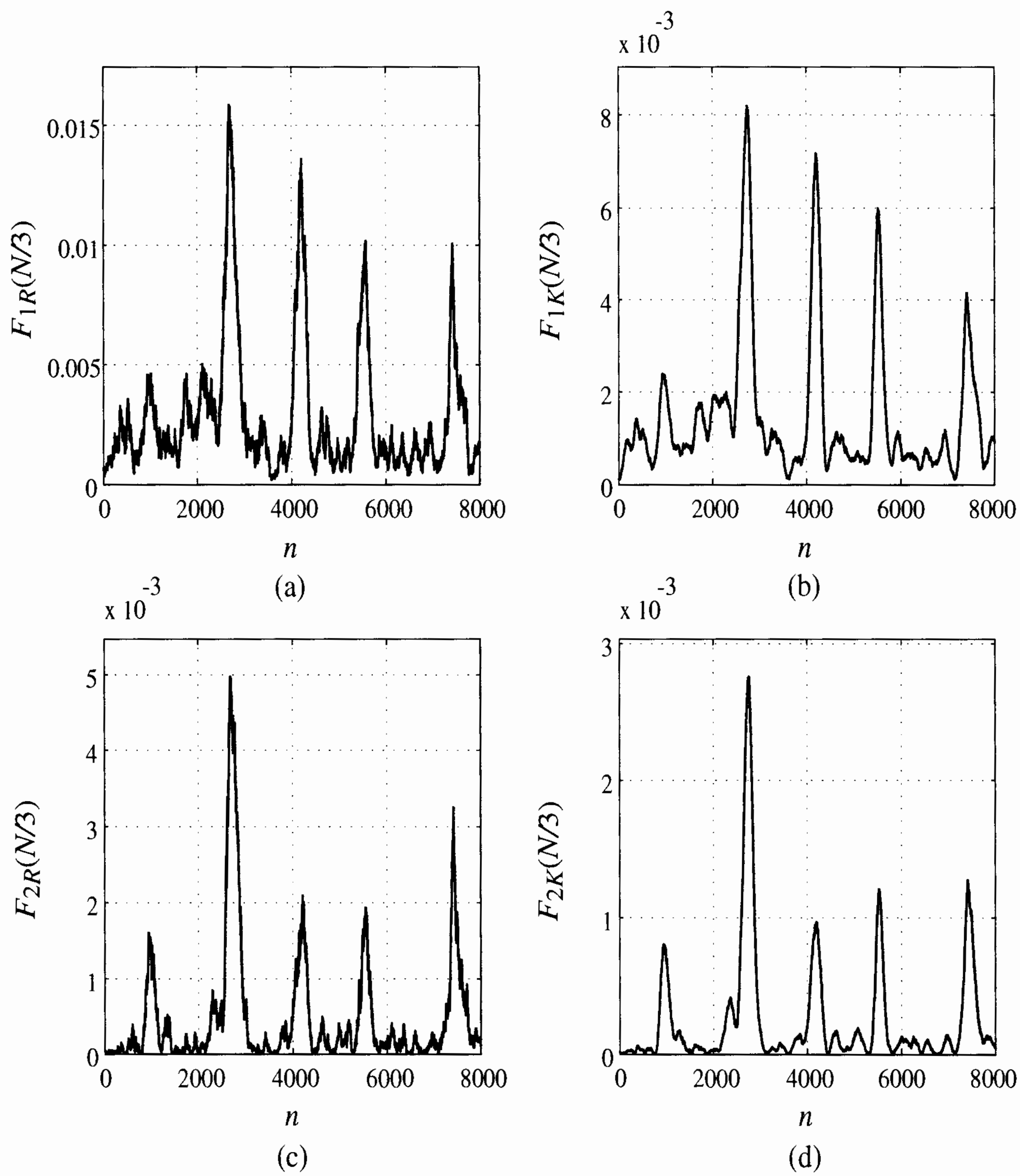


Figure 6.1. *STDFT representations for various gene-prediction measures vs. their base location n for the gene F56F11.4 in the C.elegans chromosome III.*

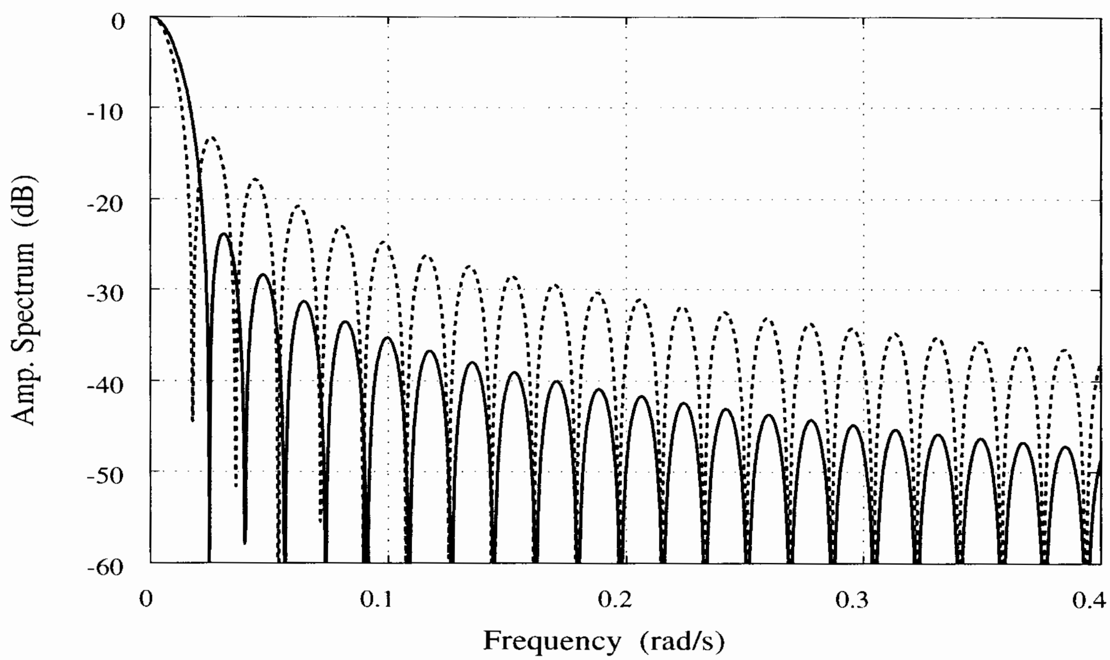


Figure 6.2. *Normalized amplitude spectrum for a Kaiser window with $a = 3.0$ (solid line) and a rectangular window (dashed line) of length $N = 351$.*

Table 6.3. *SNR Achieved for Gene-Prediction Measures*

Measures	SNR
$F_{1R}(N/3)$	0.7783
$F_{1K}(N/3)$	0.8707
$F_{2R}(N/3)$	1.6672
$F_{2K}(N/3)$	1.8656

Table 6.4. *Optimization Results for Various Windows*

Window	SNR	N	Θ
Kaiser	1.8957	405	$\alpha = 3.6226$
Dolph-Chebyshev	1.8906	513	$r = 56.4721$
Saramäki	1.8957	405	$\beta = 1.5304$
ultraspherical	1.8959	435	$\mu = 1.5733$
			$\beta = 1.6460$

6.3.1 Optimizing the Window Parameters for the Gene F56F11.4

The selection of a particular window and its adjustable parameters influence the method's capability in identifying coding regions. Figure 6.3 shows plots of the SNR obtained for the $F_{2K}(N/3)$ STDFT representation vs. the Kaiser window's adjustable parameter α for various window lengths. To facilitate the selection of window parameters, solutions were obtained for the optimization problem

$$\max_{N,\Theta} \text{SNR} \tag{6.5}$$

where N is the window length (restricted to be a multiple of 3) and Θ represents the set of other available window parameters. The optimization algorithm has been implemented using the MATLAB function `fmincon`. Table 6.4 shows optimization results for the Kaiser, Dolph-Chebyshev, Saramäki, and ultraspherical windows. The ultraspherical window achieved an SNR value slightly higher than the Kaiser and Saramäki windows while the Dolph-Chebyshev window provided the lowest SNR. Figure 6.4 illustrates the STDFT representation obtained when using the ultraspherical window with its optimal values. As can be seen, the noise level between coding regions is significantly reduced relative to those in Fig. 6.1a and b.

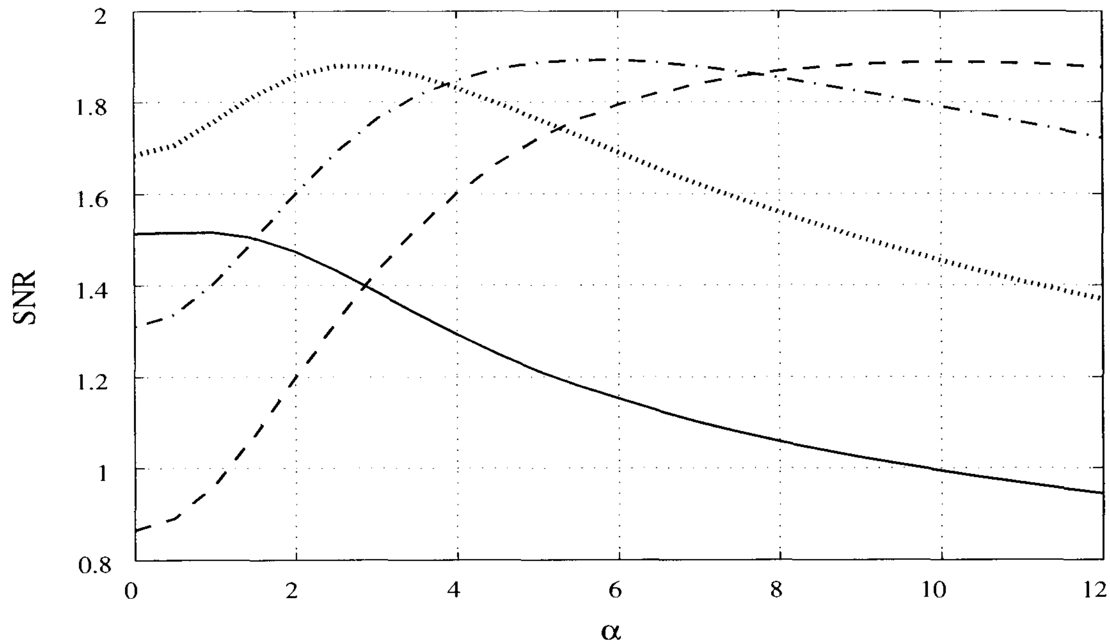


Figure 6.3. *SNR achieved for the* $F_{2K}(N/3)$ *STDFT representation vs. the adjustable parameter* α *for window lengths* $N = 201$ *(solid line),* 351 *(dotted line),* 501 *(dashed-dotted line), and* 651 *(dashed line).*

6.4 Conclusions

The application of the ultraspherical window as well as other windows along with the STDFT method for gene identification based on the well known period-three property was explored. A window was employed to suppress spectral noise originating from noncoding regions in the DNA sequence. A method for tailoring the independent parameters of the ultraspherical window for the identification of a particular gene was proposed. When the method was applied to gene F56F11.4 of the *C.elegans* organism, the SNR-based measure for gene identification was increased by 13.72% relative to that achieved when using the rectangular window. Comparisons show that the ultraspherical, Kaiser, and Saramäki windows yield approximately the same SNR values when their parameters are optimized. The Dolph-Chebyshev window yields an SNR value that is 0.28% smaller than that of the other

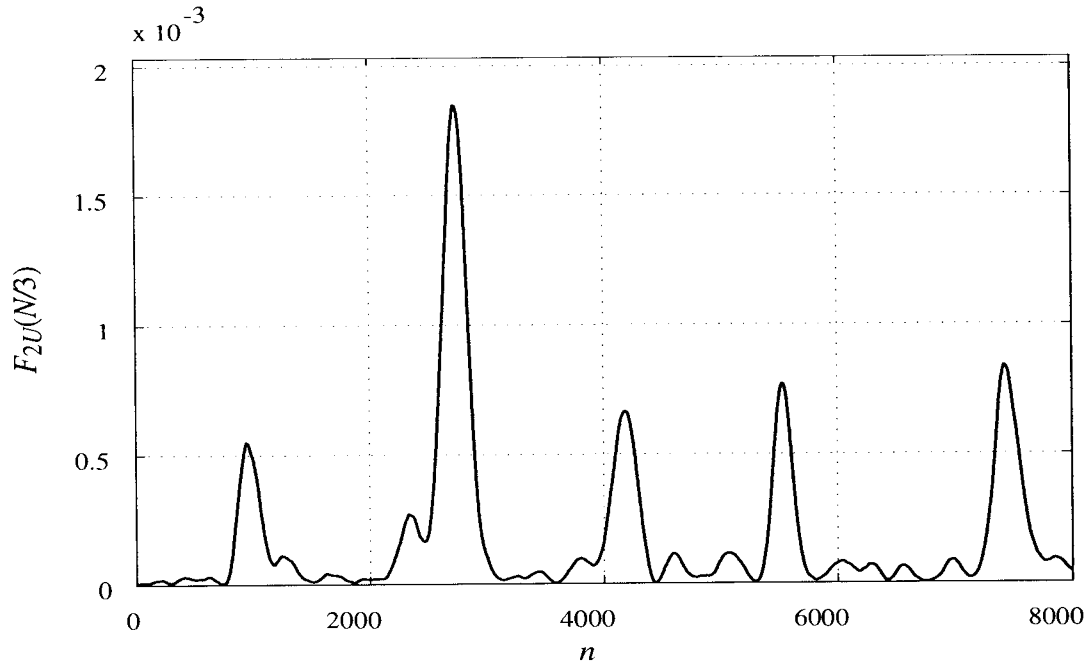


Figure 6.4. $F_{2U}(N/3)$ *STDFT representation with* $N = 435$, $\mu = 1.5733$, *and* $\beta = 1.6460$.

windows.

Chapter 7

Conclusions

7.1 Introduction

The major objective of this thesis has been to utilize the flexible nature of the ultraspherical window function to improve window-based DSP applications. Initial efforts in achieving this goal focused on furthering aspects of the window function itself, i.e., determining its spectral properties and providing fast algorithms for calculating its coefficients. Once the fundamentals of the window function had been investigated, a number of window-based DSP applications were explored which included beamforming, image processing, nonrecursive filter design, filter bank design, and genomic signal processing.

In this chapter, the contributions of the thesis are summarized and suggestions for further research are presented.

7.2 Thesis Results

In Chapter 2, two methods for evaluating the coefficients of the ultraspherical window were presented. The two methods yield the same window values for the same independent parameters μ, x_μ, and N. Economies in computation are achieved through an efficient formulation for the window coefficients which entails a computational complexity of $O(N)$ as compared with $O(N^2)$ for Streit's formulation. The amount of computation required by the new formulation is on the average 4.49% that required by Streit's formulation and

9.27% that required for the evaluation of the Kaiser window coefficients. In addition, a method for setting the null-to-null width of the ultraspherical window to $4\beta\pi/N$, i.e., β times that of the rectangular window, was introduced. The chapter has also shown that the Dolph-Chebyshev and Saramäki windows are special cases of the ultraspherical window and can be obtained by setting μ to 0 and 1, respectively.

In Chapter 3, a method for selecting the three independent parameters of the ultraspherical window so as to achieve prescribed spectral characteristics was proposed. The method comprises a collection of techniques that can be used to achieve a specified ripple ratio and either a main-lobe width or null-to-null width along with a user-defined side-lobe pattern. The side-lobe pattern in other known two-parameter windows such as the Kaiser, Saramäki, and Dolph-Chebyshev windows cannot be controlled. Experimental results have shown that the desired characteristics can be achieved with a high degree of precision. A difference in the performance of the ultraspherical and Kaiser windows has been identified, which depends critically on the required specifications. A rule for selecting either the ultraspherical or Kaiser window based on the performance difference was proposed. In addition, an expression was provided that can be used to judge how much ripple ratio is sacrificed to attain a given side-lobe pattern when compared to the Dolph-Chebyshev pattern. This is useful for antenna array designers who may need to trade-off between side-lobe pattern and ripple ratio for the application at hand. The proposed method can also be used to increase the contrast ratio in imaging systems that construct images by using two-dimensional windowed inverse DFTs on spatial frequency-domain data.

In Chapter 4, an efficient closed-form method for the design of nonrecursive digital filters using the ultraspherical window was proposed. Economies in computation are achieved in two ways. First, by using the efficient formulation of the window coefficients presented in Chapter 2, the amount of computation required is reduced to a small fraction of that required by standard methods. Second, the filter length and the independent window parameters that would be required so as to achieve prescribed specifications in lowpass, highpass, bandpass, and bandstop filters as well as in digital differentiators and Hilbert transformers are efficiently determined through empirical formulas. The ultraspherical window yields lower-order filters relative to designs obtained using other windows yielding on the average a reduction in the filter order of 3.07% relative to that in the Kaiser window, 2.86% relative to that in the Saramäki window, and 5.30% relative to that in the Dolph-Chebyshev window. Alternatively, for a fixed filter length, the ultraspherical window increases the stopband attenuation relative to that in the other windows achieving on the average an increase of 2.61 dB relative to the Kaiser window, 2.42 dB relative to the Saramäki window, and 4.49 dB relative to the Dolph-Chebyshev window. On the other hand, the weighted-Chebyshev method increases the stopband attenuation relative to that of the ultraspherical window by about 2.76 dB on the average but the amount of computation required to design filters is far greater than that required by the proposed method.

In Chapter 5, an efficient closed-form method for the design of M-channel CMFBs using the ultraspherical window so as to achieve prescribed stopband attenuation in the subbands and channel overlap was described. The design of the prototype filter is based on the proposed method for the design of lowpass filters described in Chapter 4. Experimental results have shown that, on the average, use of the Kaiser window yields filter banks with the smallest reconstruction error achieving an average percentage decrease in error over the Saramäki and ultraspherical windows of, respectively, 11.69% and 12.17% for the maximum amplitude error in the filter bank, 1.34% and 26.51% for the maximum aliasing error in the filter bank, and 2.11% and 34.65% for the maximum total aliasing error in the filter bank. On the other hand, CMFBs designed using the ultraspherical window require the least amount of computation and yield prototype filters with the shortest length when compared to designs obtained using other windows. When compared to the window-based methods of [47] and [48], the proposed method increased the average maximum amplitude error by 9.53% and 1.52%, respectively, provided almost no change in the average aliasing error and the average total aliasing error, and produced prototype filters of the same length. However, the computational effort required by the proposed design method is a small fraction, less than 2%, of that required by the other two methods making it very suitable for applications where the design must be carried out in real or quasi-real time. When compared to a filter-bank design method that employs the weighted-Chebyshev method for the prototype filter design [46], the proposed method requires significantly less computation; the method in [46] requires a huge amount of computation due to the repeated use of the Remez exchange algorithm.

In Chapter 6, the application of the ultraspherical window as well as other known windows along with the short-time discrete Fourier transform method for gene identification based on the well known period-three property was explored. The ultraspherical window is employed to suppress spectral noise originating from noncoding regions in the DNA sequence. A method for tailoring the independent parameters of the ultraspherical window for the identification of a particular gene was proposed. When the method was applied to gene F56F11.4 of the *C.elegans* organism, a signal-to-noise (SNR)-based measure for gene identification was increased by 13.72% relative to that achieved when using the rectangular window. Comparisons show that the ultraspherical, Kaiser, and Saramäki windows yield approximately the same SNR values when their parameters are optimized. The Dolph-Chebyshev window yields an SNR value that is 0.28% smaller than that of the other windows.

7.3 Future Research

The temporal and spectral characteristics of a window largely determine what applications the window is best suited for. For example, through the alteration of the side-lobe pattern of the ultraspherical window the contrast ratio of inverse-DFT-based imaging systems was improved (Chapter 3) and nonrecursive filter designs were obtained with increased stopband attenuation (Chapter 4). One possibility for future window designs is to use multiobjective optimization algorithms and the concept of Pareto optimality [27] to yield designs so as to simultaneously achieve a variety of temporal and spectral characteristics in the best possible fashion for a particular application, e.g., figures of merit for windows when used in conjunction with the DFT include the equivalent noise bandwidth, processing gain, overlap correlation, and scalloping loss [5]. Pareto-optimal multiobjective optimization methods come in many flavors including the ε-constraint method [55], goal attainment method [56], and a variety of evolutionary algorithms [57] just to name a few. Use of these methods for window designs for particular applications would be very interesting and most likely quite fruitful.

Recently a pattern-synthesis method for introducing multiple steerable nulls in an otherwise omnidirectional pattern has been proposed for circular dipole antenna arrays [58], [59] and for cylindrical patch antenna arrays [60]. Synthesis patterns of this type can suppress noise caused by stationary or mobile jammers and still maintain an otherwise omnidirectional coverage. The width of the nulls and the gain ripple in the omnidirectional pattern are controlled by the use of window functions. The Hamming window has been considered for this design. By employing flexible windows (like the ultraspherical window) it is expected that improved results can be obtained. Furthermore, patterns that yield prescribed null width and gain ripple are possible by using methods similar to those of Chapter 4 for the design of nonrecursive filters.

Transmultiplexers (TMUXs) are used for interfacing between time-division and frequency-division multiplex systems [61]. Recently TMUXs have been designed using filter banks which can be used in several communication applications including code-division multiple access, discrete multi-tone, and orthogonal frequency-division multiplexing [62], [63], [64]. CMFBs are considered good candidates for coding because of their capability to achieve high discrimination (high stopband attenuation) and due to the availability of fast algorithms for efficiently implementing the sub-carrier modulators in a parallel processing structure. An efficient design method for CMFB-based transmultiplexers would be to use the method described in Chapter 5 for the prototype filter design. This method could be used to quickly adapt TMUXs based on changing requirements for channel characteristics such as frequency-time spreading, inter symbol interference, and inter channel interference.

Finally, over the past decade DNA genomes for many organisms have been sequenced and effective methods for analyzing their characteristics are required. In Chapter 6 we investigated tailoring the ultraspherical window to maximize the detectability of the P-3 property to identify coding regions. Another well-known property present in DNA sequences is the period-ten-eleven (P-10/11) property, which signifies folding in the DNA molecule (DNA supercoiling) and can be used to identify the location of α-helix structures [65]. As in the methods described in Chapter 6, one could tailor windows to detect the P-10/11 property thereby providing a means for identifying α-helix structures. The detection of repeats of arbitrary length and their positional relation to landmarks in DNA sequences would also be interesting.

Bibliography

[1] A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters*, McGraw-Hill, New York, NY, 2005.

[2] J. W. Gibbs, "Fourier series," *Nature*, vol. 59, pp. 200 and 606, 1899.

[3] L. Fejér, "Sur les fonctions bornees et integrables," *Comptes Rendus Hebdomadaries, Seances de l'Academie de Sciences*, Paris, vol. 131, pp. 984-987, 1900.

[4] C. Lanczos, *Applied Analysis*, Van Nostrand, Princeton, NJ, 1956.

[5] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, pp. 51-83, Jan. 1978.

[6] T. Saramäki, "Finite impulse response filter design," *Handbook for Digital Signal Processing*, S. K. Mitra and J. F. Kaiser, eds., Wiley, New York, 1993.

[7] J. F. Kaiser, "Nonrecursive digital filter design using I_0-sinh window function," *IEEE Int. Symp. on Circuits and Systems*, San Francisco, California, vol. 1, pp. 20-23, Apr. 1974.

[8] T. Saramäki, "A class of window functions with nearly minimum sidelobe energy for designing FIR filters," *IEEE Int. Symp. on Circuits and Systems*, Portland, Oregon, vol. 1, pp. 359-362, May 1989.

[9] C. L. Dolph, "A current distribution for broadside arrays which optimizes the relationship between beamwidth and side-lobe level," *Proc. IRE*, vol. 34, pp. 335-348, Jun. 1946.

[10] S. R. Seydnejad and R. L. Kitney, "Real-time heart rate variability extraction using the Kaiser window," *IEEE Trans. on Biomedical Engineering*, vol. 44, no. 10, pp. 990-1005, Oct. 1997.

[11] R. M. Rangayyan, *Biomedical Signal Analysis: A Case-Study Approach*, Wiley-Interscience, New York, 2002.

[12] S. He and J.-Y. Lu, "Sidelobe reduction of limited diffraction beams with Chebyshev aperture apodization," *Journal of Acoustical Society of America*, vol. 107, no. 6, pp. 3556-3559, Jun. 2000.

[13] E. Torbet et al., "A measurement of the angular power spectrum of the microwave

background made from the high Chilean Andes," *The Astrophysical Journal*, vol. 521, pp. L79-L82, Aug. 1999.

[14] B. Picard, E. Anterrieu, G. Caudal, and P. Waldteufel, "Improved windowing functions for Y-shaped synthetic aperture imaging radiometers," *IEEE Int. Symp. on Geoscience and Remote Sensing*, vol. 5, pp. 2756-2758, Jun. 1999.

[15] P. Lynch, "The Dolph-Chebyshev window: A simple optimal filter," *Monthly Weather Review*, vol. 125, pp. 655-660, Apr. 1997.

[16] R. L. Streit, "A two-parameter family of weights for nonrecursive digital filters and antennas," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 32, no. 1, pp. 108-118, Feb. 1984.

[17] T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. on Circuit Theory*, vol. CT-19, pp. 189-194, Mar. 1972.

[18] J. H. McClellan and T. W. Parks, "A unified approach to the design of optimum FIR linear-phase digital filters," *IEEE Trans. on Circuit Theory*, vol. CT-20, pp. 697-701, Nov. 1973.

[19] D. J. Shpak and A. Antoniou, "A generalized Remez method for the design of FIR digital filters," *IEEE Trans. on Circuits and Systems*, vol. 37, no. 2, pp. 161-174, Feb. 1990.

[20] U. W. Hochstrasser, "Orthogonal polynomials," in *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, M. Abramowitz and I. A. Stegun, eds., Nation Bureau of Standards Applied Mathematics Series, US Government Printing Office, Washington DC, 1965.

[21] J. J. Soltis, "New Gegenbauer-like and Jacobi-like polynomials with applications," *Journal of the Franklin Institute*, vol. 330, no. 3, pp. 635-639, 1993.

[22] J. J. Soltis, "Analytic wavelets based upon deorthogonalized Gegenbauer polynomials," *Proc. of the IEEE-SP Int. Symp. on Time-Frequency and Time-Scale Analysis*, vol. 1, pp. 389-391, Philadelphia, Pennsylvania, Oct. 1994.

[23] A. Saèd, J. J. Soltis, and M. Ahmadi, "Gegenbauer (ultraspherical) polynomials for Gabor-type wavelet approximation and FIR filter function generation in wavelet analysis," *Canadian Conf. on Elec. and Comp. Eng.*, vol. 2, pp. 878-881, Montreal, Quebec, Sept. 1995.

[24] A. G. Deczky, "Unispherical windows," *IEEE Int. Symp. on Circuits and Systems*, vol. 2, pp. 85-88, Sydney, Australia, May 2001.

[25] R. J. Stegen, "Excitation coefficients and beamwidths of Tschebysceff arrays," *Proc.*

IRE, pp. 1671-1674, Nov. 1953.

[26] Á. Elbert, "Some recent results on the zeros of the Bessel functions and orthogonal polynomials," *Journal of Computational and Applied Mathematics*, vol. 133, no. 1-2, pp. 65-83, Aug. 2001.

[27] R. Fletcher, *Practical Methods of Optimization*, Wiley, New York, NY, 1987.

[28] O. Herrmann, L. R. Rabiner, and D. S. K. Chan, "Practical design rules for optimum finite impulse response low-pass digital filters," *Bell System Technical Journal*, vol. 52, no. 6, pp. 769-799, Jul.-Aug. 1973.

[29] J. F. Kaiser and R. W. Schafer, "On the use of the I_0-sinh window for spectrum analysis," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 28, no. 1, pp. 105-107, Feb. 1980.

[30] C. A. Balanis, *Antenna Theory: Analysis and Design*, Wiley, New York, NY, 1982.

[31] D. J. Shpak, "A method for the optimal pattern synthesis of linear arrays with prescribed nulls," *IEEE Trans. on Antennas and Propogation*, vol. 44, no. 3, pp. 286-294, Mar. 1996.

[32] G. G. Kuperman and T. D. Penrod, "Evaluation of compressed synthetic aperture radar imagery," *IEEE National Aerospace and Electronics Conference*, vol. 1, pp. 319-326, May 1994.

[33] D. C. Munson, J. D. O'Brien, and W. K. Jenkins, "A tomographic formulation of spotlight-mode synthetic aperture radar," *Proc. IEEE*, vol. 71, no. 8, pp. 917-925, Aug. 1983.

[34] H. Jiang, W. R. Chen, and H. Liu, "Techniques to improve the accuracy and to reduce the variance in noise power spectrum measurements," *IEEE Trans. on Biomedical Engineering*, vol. 49, no. 11, pp. 1270-1278, Nov. 2002.

[35] J. W. Adams, "A new optimal window", *IEEE Trans. on Signal Processing*, vol. 39, no. 8, pp. 1753-1769, Aug. 1991.

[36] L. R. Rabiner, J. F. Kaiser, O. Herrmann, and M. T. Dolan, "Some comparisons between FIR and IIR digital filters," *Bell System Technical Journal*, vol. 53, pp. 305-331, Feb. 1974.

[37] S. H. Nawab, A. V. Oppenheim, A. P. Chandrakasan, J. M. Winograd, and J. T. Ludwig, "Approximate signal processing," *Journal of VLSI Signal Processing Systems*, vol. 15, no. 1-2, pp. 177-200, Jan. 1997.

[38] J. T. Ludwig, S. H. Nawab, and A. P. Chandrakasan, "Low-power digital filtering using approximate processing," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 3, pp. 395-400, Mar. 1996.

[39] C. J. Pan, "A stereo audio chip using approximate processing for decimation and interpolation filters," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 1, pp. 45-55, Jan. 2000.

[40] J. W. Adams, "FIR digital filters with least-squares stopbands subject to peak-gain constraints," *IEEE Trans. on Circuits and Systems*, vol. 38, no. 4, pp. 376-388, Apr. 1991.

[41] W. Cheney and D. Kincad, *Numerical Mathematics and Computing*, Brooks/Cole, Pacific Grove, California, 1995.

[42] A. H. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 29, no. 1, pp. 84-91, Feb. 1981.

[43] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill, New York, NY, 1998.

[44] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451-513, Apr. 2000.

[45] V. X. Afonso, W. J. Tompkins, T. Q. Nguyen, and S. Luo, "ECG beat detection using filter banks," *IEEE Trans. on Biomedical Engineering*, vol. 46, no. 2, pp. 192-202, Feb. 1999.

[46] C. D. Creusere and S. K. Mitra, "A simple method for designing high-quality prototype filters for M-band pseudo QMF banks," *IEEE Trans. on Signal Processing*, vol. 34, no. 4, pp. 1005-1007, Apr. 1995.

[47] Y.-P. Lin and P.P. Vaidyanathan, "A Kaiser window approach for the design of prototype filters of cosine modulated filterbanks," *IEEE Signal Processing Letters*, vol. 5, no. 6, pp. 132-134, Jun. 1998.

[48] F. Cruz-Roldán, P. Amo-Lopez, S. Maldonado-Bascon, and S. S. Lawson, "An efficient and simple method for designing prototype filters for cosine-modulated pseudo-QMF banks," *IEEE Signal Processing Letters*, vol. 9, no. 1, pp. 132-134, Jan. 2002.

[49] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, 1993.

[50] M. Q. Zhang, "Computational prediction of eukaryotic protein-coding genes," *Nature Reviews, Genetics*, vol. 3, pp. 698-709, Sept. 2002.

[51] S. Tiwari, S. Ramachandran, A. Bhattacharya, S. Bhattacharya, and R. Ramaswamy, "Prediction of probable genes by Fourier analysis of genomic sequences," *Computer Applications in the Biosciences*, vol. 13, no. 3, pp. 263-270, 1997.

[52] D. Anastassiou, "Genomic signal processing," *IEEE Signal Processing Magazine*, vol. 18, no. 4, pp. 8-20, 2001.

[53] P. P. Vaidyanathan and B.-J. Yoon, "The role of signal-processing concepts in ge-

nomics and proteomics," *Journal of the Franklin Institute*, vol. 341, no. 1-2, pp. 111-135, 2004.

[54] T. W. Fox and A. Carreira, "A digital signal processing method for gene prediction with improved noise suppression," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 1, pp. 108-114, 2004.

[55] S. H. Hollingdale, *Methods of operational analysis in newer uses of mathematics* (J. Lighthill, ed.), Penguin Books, 1978.

[56] F. W. Gembicki, "Vector optimization for control with performance and parameter sensitivity indices," Ph.D. Dissertation, Case Western Reserve Univ., Clevland, Ohio, 1974.

[57] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.

[58] J. Abele, J. Joubert, and J. W. Odendaal, "Introduction of multiple nulls in otherwise omnidirectional pattern of circular dipole array," *Electronics Letters*, vol. 34, no. 14, pp. 1371-1373, 1998.

[59] J. Abele, J. Joubert, and J. W. Odendaal, "Introduction of multiple nulls with controllable depths in an omnidirectional pattern of a circular array," *Signal Processing*, vol. 80, no. 1, pp. 141-149, 2000.

[60] P. Niemand, J. Joubert, and J. W. Odendaal, "Null-forming synthesis for cylindrical patch arrays," *Electronics Letters*, vol. 34, no. 14, pp. 1371-1373, 1998.

[61] M.G. Bellanger and J.L. Daguet, "TDM-FDM transmultiplexer: digital polyphase and FFT," *IEEE Trans. on Communications,* vol. 19, pp. 628-634, 1971.

[62] A. N. Akansu, M. V. Tazebay, and R. A. Haddad, "A new look at digital orthogonal transmulitplexers for CDMA communications," *IEEE Trans. on Signal Processing*, vol. 45, no. 1, pp. 263-267, 1997.

[63] A. N. Akansu, P. Duhamel, X. Lin, and M. Courville, "Orthogonal transmulitplexers in communications: A review," *IEEE Trans. on Signal Processing*, vol. 46, no. 4, pp. 263-267, 1998.

[64] L. Chen, K. P. Chan, T. Q. Nguyen, and X. H. Dai, "Algorithm for the design of transmultiplexers with time-frequency spread criteria," *Electronics Letters*, vol. 36, no. 17, pp. 1499-1500, 2000.

[65] H. Herzel, O. Weiss and E. N. Trifonov, "10-11 bp periodicities in complete genomes reflect protein structure and DNA folding," *Bioinformatics,* vol. 15, pp. 187-193, 1999.