On the Use of Discrete Prolate Spheroidal Windows for Frequency Selective Filter Design

Lav Varshney

Applications of Signal Processing
School of Electrical and Computer Engineering
Cornell University
February 23, 2004

Abstract—The FIR filter design problem is considered with respect to a combined integrated squared error-Chebyshev error criterion. Truncation of the DTFT of the desired frequency response with tapered windows is introduced and the discrete prolate spheroidal window is shown to be a good window for this purpose, due to its optimal mainlobe width-sidelobe energy tradeoff. The discrete prolate spheroidal window is shown to be a scalar multiple of the zero-order discrete prolate spheroidal sequence, and a commutable matrix reformulation for numerically well-conditioned computation is given. Discrete prolate spheroidal window parameters are related to frequency selective filter parameters through empirical study. Integrated squared error-Chebyshev error tradeoffs for filters designed with discrete prolate spheroidal windows are compared with other window methods, with the optimal constrained least squares method, and with the optimal Chebyshev method. It is found that the discrete prolate spheroidal window method is suboptimal, but is best among window methods. Finally, a suggestion for the use of discrete prolate spheroidal windows for spectral analysis is provided. MATLAB code is included in the appendices.
I. Introduction

Filter design is fundamentally a problem of finding a practical, realizable filter that approximates an ideal, desired filter. Frequency domain methods of filter design attempt to minimize the difference between the desired frequency response and the actual frequency response, with respect to an error criterion. If one considers the causal, linear phase, finite impulse response, frequency selective filter design problem, the desired passband frequency response will be $e^{j\omega \tau}$, for some constant $\tau$, and the desired stopband frequency response will be 0, with discontinuities at the boundaries between bands.

The optimal least squares filter can be found by simply truncating the discrete-time Fourier transform (DTFT) of the ideal frequency response. Maintenance of linear phase requires truncation with preservation of symmetry in the unit pulse response. Optimality of the filter with regard to the integrated squared error criterion is guaranteed by the orthogonality principle. Due to the nonuniform convergence of the truncated response, the optimal least squares filter suffers from the Gibbs phenomenon. This phenomenon manifests itself in the frequency domain as a convolution between the ideal rectangular frequency response and the Dirichlet (periodic sinc) function, the DTFT of the rectangular truncation.

For many applications of frequency-selective filters, the large amplitude ripples introduced by Gibbs phenomenon are objectionable, since they may introduce amplitude distortion in the passband and may allow strong single frequency interferers in the stopband to pass without sufficient attenuation. Thus it is seen that the integrated squared error criterion is often not the natural criterion for optimization. Determination of a suitable error criterion for filter design has been a subject of considerable debate. In a humorous\(^1\) correspondence article from the 1950s, Ernst A. Guillemin commented that everyone seemed to take for granted that the integrated squared error criterion is also nature’s error criterion, but that other error criterion such as the Chebyshev maximum error criterion may be more suitable [1].

The optimal Chebyshev filter, the filter that minimizes maximum error over all frequency bands, is an equiripple filter and can be found with the Parks-McClellan algorithm. The Parks-McClellan algorithm formulates the filter design problem as a problem in polynomial approximation and uses Remez exchange to find the optimal filter. The design algorithm disallows the use of adjacent frequency bands in the desired frequency response so as to use the alternation theorem to recognize the optimal solution. Therefore a “transition region” must be introduced in the desired frequency response, altering the approximation problem, and often adversely affecting the result for applications in which there is no natural transition region. Furthermore, the optimal Chebyshev filters have large integrated squared error, and so are also considered suboptimal for many applications of frequency-selective filters. Thus Chebyshev

\(^1\) Guillemin describes [1] as “light and fluffy with a bit of humor and fun thrown in” [2].
error is also not the natural criterion for optimization for many applications. In fact, many applications of frequency-selective filters require a tradeoff between the two error criteria.

II. The Compromise Between Integrated Squared Error and Chebyshev Error

Many techniques have been proposed for designing filters that have good integrated squared error and Chebyshev error performance, that is to say, filter design techniques that consider an error criterion that is a function of the two standard error metrics.

Traditional ways to make the tradeoff between integrated squared error and Chebyshev error are based on reducing Gibbs phenomenon in the optimal least squares filter design. Remember that Gibbs phenomenon is the cause of large Chebyshev errors in optimal least squares filters. One method to reduce Gibbs phenomenon in optimum least squares filters is to modify the desired response by replacing the sharp cutoff with a gradual transition. Gradually decreasing trigonometric and polynomial transition functions have been suggested to replace the discontinuity. Another method is to modify the desired response by introducing “don’t care” transition bands. The error function is given zero weight in these frequency bands. Equivalently, one can consider the support of the error function not to include the transition bands. Thus when the filter is optimized for the integrated squared error criterion, only the passband and stopband are considered, and the result is generally a reduction in maximum error. In general, the sharp cutoffs are smoothed as a result of transition bands [3]. Both of these methods change the original approximation problem to one, which is believed to result in better combined integrated squared-Chebyshev performance.

Constrained least squares filter design is another method for achieving a tradeoff between the squared error and the maximum error. This method designs the optimal least squares filter under the constraint of fixed maximum Chebyshev error. It can produce a wide range of filters that reflect choices made in the least squares-Chebyshev error tradeoff. The design technique employs an exchange algorithm that uses Lagrange multipliers and Kuhn-Tucker conditions on each iteration [4].

An additional method to make a tradeoff between integrated squared error and Chebyshev error has been to use nonrectangular, smoothly tapering windows to truncate the DTFT of the desired frequency response. The tapered windows serve to reduce Gibbs phenomenon, the cause of large Chebyshev errors in optimal least squares filters. Many types of window functions have been proposed for the design of frequency selective filters. The use of parameterized families of window functions allow a continuum of filters to be designed. Examples include the cosine-on-pedestal family, which includes the Hamming and Hanning windows; the Gaussian or Weierstrass family; the Dolph-Chebyshev family; and the Kaiser family [5],[6].
III. Selection of a Window Family for Filter Design

A general window used for filter design is a finite-extent, symmetric, discrete-time sequence that generally displays tapering at the edges. The DTFT of such a discrete-time sequence displays a mainlobe and smaller sidelobes, which arise due to the finite truncation. Note that the DTFT of a rectangular window is the Dirichlet function (periodic sinc), with a mainlobe and sidelobes, and that the DTFT of a general window is the convolution of some function with this Dirichlet function. Figure 1 shows a time-domain rectangle sequence, the corresponding frequency domain response (the Dirichlet function), and the Dirichlet function on a relative decibel scale. Figure 2 shows the Hamming window, one of the most simple window functions, and a member of the cosine-on-pedestal family. As can be seen, the mainlobe of the Hamming window, like all general window functions is wider than the rectangular window, but that the sidelobe levels are lower.

Figure 1. The Rectangular Window. (a) time-domain response (b) frequency domain response magnitude (c) frequency domain response magnitude on relative decibel scale.
Windows with smaller sidelobes yield better approximations, in terms of Chebyshev error, of the ideal response at a discontinuity. This can be seen, since the application of a window involves convolving the ideal frequency response with the frequency of the window, the smaller the sidelobes, the smaller the resulting ripples in the designed filter. The mainlobe width of a window is related to the transition width of the designed filter. Figure 3 shows the definitions for ripple and transition width for a filter.

Desirable windows for filter design can be determined by optimizing the tradeoff between mainlobe width and sidelobe area. The window function that achieves this optimal tradeoff is maximally concentrated around zero frequency in the frequency domain [5]. The Kaiser family of window functions has received particular attention due to the near-optimal tradeoff between mainlobe width and sidelobe area [5],[7], however, discrete prolate spheroidal windows (DPSW) have the optimal energy concentration at low frequencies, and thus may be considered the most desirable for use as filter design windows.
IV. Prolate Spheroidal Wave Functions and Discrete Prolate Spheroidal Sequences

The continuous time problem of maximally concentrating a time-limited function to a limited bandwidth was considered by Slepian, Pollak, and Landau and they found that the prolate spheroidal wave functions (PSWF) are maximally concentrated in this regard [8]-[11].2 Due to the difficulty in computing PSWFs, Kaiser introduced an approximation with I₀-sinh functions that use modified Bessel functions of the first kind and zeroth order [7]. Kaiser windows are in fact generated by sampling this I₀-sinh approximation.

The discrete-time equivalent to the continuous time problem is determining the extent to which a finite-extent discrete-time sequence can have its spectrum concentrated on an interval in the fundamental period of the periodic Fourier spectrum. It was shown that the sequences that maximize the concentration for a given bandwidth $W$ are the discrete prolate spheroidal sequences (DPSS) [13]-[15]. Slepian presented and characterized DPSS completely, related them to discrete prolate spheroidal wave functions (DPSWF), and also formulated them in an easier to calculate form [16]. A brief review of Slepian’s formulation and results is presented in Section V and VI. The reader is referred to Slepian’s work for a complete exposition.

2 An easier to read summary and commentary on the Prolate Spheroidal Wave Function series of papers is given by Slepian in [12].
V. Discrete Prolate Spheroidal Sequences

DPSS are defined in terms of their length, $N$, and the frequency interval $(-W,W)$ in which they are maximally concentrated. The DPSS of $k$th order for a given $N$ and $W$, $v[n]^{(k)}(N,W)$, is defined as the real solution to the system of equations

$$
\sum_{m=0}^{N-1} \sin(2\pi W(n-m)) v[m]^{(k)}(N,W) = \lambda^{(k)}(N,W)v[n]^{(k)}(N,W), \quad n = 0, \pm 1, \pm 2, \ldots
$$

for each $k = 0, 1, 2, \ldots, N-1$, and with a specific normalization to ensure uniqueness. It can be shown that the system has $N$ distinct eigenvalues and eigenvectors. The $\lambda^{(k)}(N,W)$ are the eigenvalues of the system and are related to the amount of concentration that is achieved. The system may be expressed in matrix-vector form, using the $N \times N$ matrix $H(N,W)$ with elements

$$
\{H(N,W)\}_{m,n} = \frac{\sin(2\pi W(m-n))}{\pi(m-n)}, \quad m, n = 0, 1, \ldots, N-1
$$

in the form

$$
H(N,W)v[n](N,W) = \lambda(N,W)v[n](N,W)
$$

It can further be shown that the eigenvalues $0 < \lambda_0^{(k)} < \lambda_1^{(k)} \ldots < \lambda_{N-1}^{(k)} < 1$ are clustered near zero and one. This means that the individual eigenvectors are very ill conditioned numerically, which will be important for the computation of the DPSS, as will be shown in Section VII.

VI. Discrete Prolate Spheroidal Windows

Consider an index-limited sequence $h[n]$, such that $h[n]$ is zero for indices $n < N_0$ and for $n > N_0 + N - 1$, with DTFT $H(f)$. The natural measure of the extent to which $H(f)$ is concentrated in the frequency interval $(-W,W)$ is given by

$$
\mu = \frac{\int_{-W}^{W} |H(f)|^2 df}{\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} |H(f)|^2 df}
$$

Through the use of the definition of the DTFT, Parseval’s theorem and Euler’s relationship, this concentration factor may be expressed as

$$
\mu = \frac{\sum_{n=N_0}^{N_0+N-1} \sum_{m=N_0}^{N_0+N-1} \frac{\sin(2\pi W(n-m))}{\pi(n-m)} h[n]h^*[n]}{\sum_{n=N_0}^{N_0+N-1} |h[n]|^2} = \frac{N_0+N-1}{N_0} \frac{\sum_{n=0}^{N-1} \sum_{m=0}^{N_0} \frac{\sin(2\pi W(n-m))}{\pi(n-m)} h[n+N_0]h^*[n+N_0]}{\sum_{n=0}^{N-1} |h[n+N_0]|^2}
$$

The index-limited sequence of length $N$ that maximizes the concentration, $\mu$, for a given bandwidth $W$ is found to be the sequence $h[n+N_0] = cv[n]^{(0)}(N,W)$ for $n = 0, 1, \ldots, N-1$, a scalar
multiple of the DPSS of zero-order. Thus the index-limited sequence with most concentrated spectrum in $-W \leq f \leq W$ is

$$h[n] = \begin{cases} v^{(0)}_{n-N_0}(N,W), & N_0 \leq n \leq N_0 + N - 1 \\ 0, & \text{otherwise} \end{cases}$$  \hspace{1cm} (6)$$

The concentration of the spectrum $\mu$ is the eigenvalue, $\lambda_0(N,W)$, associated with the DPSS and the spectrum is given by

$$H(f) = dU_0(N,W;f)e^{j\pi(2N_0+N-1)f}, \forall f$$

where $d$ is independent of $f$, and $U_0(N,W;f)$ is the DPSWF of zero-order that corresponds to the DPSS.

It has been shown that the sequence that maximizes the frequency concentration for a given bandwidth $W$ is a scalar multiple of the zero-order DPSS, so this is in fact is the discrete prolate spheroidal window.

VII. Reformulation of the Discrete Prolate Spheroidal Window for Easier Computation

In Section V, it was given that the computation of DPSS using the system of equations given in (1) or equivalently in matrix-vector form in (2) and (3) was numerically ill-conditioned. Slepian showed that the DPSSs index-limited to $(0,N-1)$ also satisfy the difference equation

$$\frac{1}{2}n(N-n)v[n-1]^{(k)} + \left(\frac{N-1}{2} - n\right)^2 \cos(2\pi W) - \theta_k v[n]^{(k)} + \frac{1}{2}(n+1)(N-1-n)v[n+1]^{(k)} = 0$$

where dependence of the DPSSs on $N$ and $W$ is omitted for conciseness. This difference equation reformulation may be expressed as a problem of finding eigenvectors of a symmetric tridiagonal matrix, which has well-separated eigenvalues, and consequently numerically well-conditioned eigenvectors. The entries of the tridiagonal matrix are given by

$$\{T(N,W)\}_{i,j} = \begin{cases} \frac{1}{2}(N-i), & j = i-1 \\ \left(\frac{N-1}{2} - i\right)^2 \cos(2\pi W), & j = i \\ \frac{1}{2}(i+1)(N-1-i), & j = i+1 \\ 0, & \text{otherwise} \end{cases}$$  \hspace{1cm} (9)$$

This reformulation can also be understood in the framework of the common eigenvectors of commutable matrices [17]. It can be shown that the matrix $T(N,W)$ commutes with the original linear system matrix $H(N,W)$ from (2), that is to say


An explanation of the reason why commutable matrices have common eigenvectors is given in Box A. The disadvantage of this reformulation is that the eigenvalues no longer correspond to the frequency concentration values.
VIII. Computation of the Discrete Prolate Spheroidal Window

The DPSW is a scalar multiple of the zero-order DPSS for a given $N$ and $W$, and is given by an eigenvector associated with the dominant eigenvector of the matrix $T(N,W)$. Since $T$ is a symmetric, tridiagonal matrix, one might consider the use of the QL algorithm or other advanced eigendecomposition algorithms as in [18], however since we are only concerned with the dominant eigenvalue and associated eigenvector, the power method is used.

An eigenvector of an $N \times N$ matrix $A$ is a nonzero vector $x$ such that $Ax = \lambda x$ for some scalar $\lambda$. A scalar $\lambda$ is called an eigenvalue of $A$ if there is a nontrivial solution $x$ of $Ax = \lambda x$; such an $x$ is called an eigenvector corresponding to $\lambda$.

There are many numerical methods for determining eigenvalues and eigenvectors. The power method applies to an $N \times N$ matrix $A$ with a strictly dominant eigenvalue $\lambda_1$, i.e. that the absolute value of $\lambda_1$ is strictly larger than all other eigenvalues. Note that by Lemma 7.7.1 in [18], the eigenvalues of $T$ will be distinct, so there will be a dominant eigenvalue. In this case, the power method iteratively produces a scalar sequence that approaches $\lambda_1$ and a vector sequence that approaches a corresponding eigenvector. The algorithm is given in Figure 4.

1. Select an initial vector $x_0$ whose largest entry is 1.
2. For $k = 0, 1, \ldots$
   a. Compute $Ax_k$.
   b. Let $\lambda_k$ be an entry in $Ax_k$ whose absolute value is as large as possible.
   c. Compute $x_{k+1} = (1/\lambda_k)Ax_k$.
3. For almost all choices of $x_0$, the sequence $\{\lambda_k\}$ approaches the dominant eigenvalue, and the sequence $\{x_k\}$ approaches a corresponding eigenvector.

Figure 4. The Power Method for Estimating a Strictly Dominant Eigenvalue.

Note that the eigenvector corresponding to a particular eigenvalue is not unique and that if $x$ is an eigenvector, then so is $kx$, where $k$ is any nonzero scalar. Thus eigenvectors computed by different algorithms may appear to disagree completely, though they only differ by a scalar factor. The eigenvector produced by the power method has a maximum element equal to 1.
Since it is desired that the center element of an odd-length window be 1, this is the scalar multiple of the zero-order DPSS that is used as the DPSW.

Another point worth noting is that the power method can fail to converge to the largest eigenvalue \( \lambda_1 \), for certain initial vectors \( x_0 \). This failure would occur if \( x_0 \) is orthogonal to the eigenvector corresponding to the largest eigenvalue. In that case, the method would converge to the largest eigenvalue whose eigenvector is contained in the spectral decomposition of \( x_0 \).

Appendix 1 gives a listing of MATLAB code that may be used to compute the DPSW. A utility function that implements the power method is also included.

**IX. Examples of Discrete Prolate Spheroidal Windows**

A few examples of discrete prolate spheroidal windows are presented in this section. The window length, \( N \), as well as the bandwidth of concentration, \( W \), parameterize the family of DPSWs. The length can be any positive integer, whereas \( W \) is a real number between 0 and 0.5. Discrete time-domain representations along with frequency domain response magnitudes on relative decibel scales are given. As can be noted from Figures 5, 6, and 7, the mainlobe width is directly related to the bandwidth of concentration parameter and minimally dependent on window length, as should be expected. The sidelobe level is a function of both the window length and the bandwidth of concentration parameter. A more in-depth analysis of these relationships, in the context of filter parameters will be given in Section X.

![Figure 5. Discrete Prolate Spheroidal Window with N = 31 and W = 0.1. (a) Time-domain response (b) frequency domain response magnitude on relative decibel scale.](image-url)
X. Design Considerations for DPSW Filters: Determination of Parameters

In this section, the relationships between the two DPSW parameters and the three filter parameters defined in Section III, particularly Figure 3, are characterized. The three filter parameters were the passband ripple ($\delta_P$), stopband ripple ($\delta_S$), and transition width ($\Delta\omega$). This empirical characterization was performed for a lowpass filter with desired cutoff frequency of 0.22. The results, however should generalize to all frequency selective filters. If desired, one could convert these results into empirical design formulas, as was done by Kaiser for the Kaiser family of windows (see [7] or Eq. 7.62 and 7.63 in [5]). Figures 8, 9, and 10 show the maximum ripples and transition width as functions of the window length (or equivalently the filter length) for a few discrete values of the concentration parameter, $W$. Figures 11, 12, and 13 show the
maximum ripples and transition width as functions of the concentration parameter for a few
discrete values of window length. Following Kaiser, the ripple values are presented in decibels,
using the definition

\[ A = -20 \log_{10} (\delta) \]  \hspace{1cm} (11)

Some points to note in these figures include the fact that passband ripple is always
slightly greater than stopband ripple, the fact that ripple decreases as filter length increases and
as the concentration parameter \( W \) increases. Furthermore, the transition width is almost exactly a
linear function of \( W \), in fact given by \( 2W \), and is nearly independent of the filter length (Figure
10 demonstrates the slight dependence).

Appendix 2 gives a listing of MATLAB code that may be used to calculate the three filter
parameters of a given filter.

![Figure 8. Maximum Passband Ripple as a function of Filter Length for Low Pass Filters Designed using the Discrete Prolate Spheroidal Window with fixed frequency concentration parameter W.](image-url)
Figure 9. Maximum Stopband Ripple as a function of Filter Length for Low Pass Filters Designed using the Discrete Prolate Spheroidal Window with fixed frequency concentration parameters $W$.

Figure 10. Transition Width as a function of Filter Length for Low Pass Filters Designed using the Discrete Prolate Spheroidal Window with fixed frequency concentration parameters $W$. 
Figure 11. Maximum Passband Ripple as a function of frequency concentration parameter $W$ for Low Pass Filters Designed using the Discrete Prolate Spheroidal Window with fixed Filter Length.

Figure 12. Maximum Stopband Ripple as a function of frequency concentration parameter $W$ for Low Pass Filters Designed using the Discrete Prolate Spheroidal Window with fixed Filter Length.
Figure 13. Transition Width as a function of frequency concentration parameter $W$ for Low Pass Filters Designed using the Discrete Prolate Spheroidal Window with fixed Filter Length. The function appears noisy due to numerical problems.

XI. Comparison of DPSW-Filters and other Filters

In this penultimate section, a comparison of filters designed using the discrete prolate spheroidal window and filters designed using other methods is made. The two error criterion that were first considered in Sections I and II will be used to make this comparison, namely integrated squared error and Chebyshev error. Design methods compared to DPSW include the constrained least squares method of Selesnick, Lang, and Burrus, the Parks-McClellan optimal Chebyshev method, and the use of other window functions. The Kaiser family, the Hanning window, the Hamming window, and the rectangular window are considered.

Note that the integration performed in calculating the integrated squared error is performed over the entire frequency axis from 0 to 0.5, but the maximization performed in calculating the Chebyshev error is only performed over the passband as delineated from 0 to the left edge of the transition width, marked in Figure 3. Numerical integration with a composite 4-point Newton-Cotes quadrature rule was used [19]. Box B gives an overview of numerical integration with the Newton-Cotes rule. Appendix 3 lists MATLAB code that may be used to calculate the integrated squared error and the Chebyshev error (a simplification of code from Appendix 2), and a utility function.
As in Section X, for all methods, a lowpass filter with desired cutoff of 0.22 is designed, however generalization to all frequency selective filters should be possible. In all cases, a filter of length 51 is considered. Figure 14 shows the integrated squared error as a function of Chebyshev error for the constrained least squares filter design method, which is optimal. Figure 15 shows the same function for filters designed with DPSW, and similarly Figure 16 for Kaiser, and Figure 17 for Parks-McClellan. Since the Parks-McClellan algorithm requires a finite transition region, it was specified symmetrically around the desired cutoff frequency and varied. A comparison of these four methods, along with Hamming, Hanning, and Rectangular windowing is given in Figure 18.

**Box B: Numerical Integration [19]**

An m-point quadrature rule $Q$ for the definite integral

$$I = \int_a^b f(\omega) d\omega \quad (B.1)$$

is an approximation of the form

$$Q = (b - a)\sum_{k=1}^{m} w_k f(\omega_k) \quad (B.2)$$

The abscissas, $\omega_k$, and the weights, $w_k$, define the quadrature rule and are chosen so $Q \approx I$. A composite quadrature rule may be formed by applying the quadrature rule to portioned subintervals of the interval between the limits of integration. The Newton-Cotes family of quadrature rules are derived by integrating uniformly spaced polynomial interpolants of the integrand $f(\omega)$, and include the trapezoidal rule and Simpson rule as low order cases. The weights in Equation (B.2) for the lowest ordered Newton-Cotes rules are given in Table B-1.

<table>
<thead>
<tr>
<th>Order (m)</th>
<th>Weights ($w_k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 (Trapezoidal)</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>3 (Simpson)</td>
<td>$\frac{1}{6}$</td>
</tr>
<tr>
<td>4</td>
<td>$\frac{1}{8}$</td>
</tr>
<tr>
<td>5</td>
<td>$\frac{7}{40}$</td>
</tr>
<tr>
<td>6</td>
<td>$\frac{19}{288}$</td>
</tr>
</tbody>
</table>

Table B-1. Weights for Newton-Cotes Quadrature Rules

As in Section X, for all methods, a lowpass filter with desired cutoff of 0.22 is designed, however generalization to all frequency selective filters should be possible. In all cases, a filter of length 51 is considered. Figure 14 shows the integrated squared error as a function of Chebyshev error for the constrained least squares filter design method, which is optimal. Figure 15 shows the same function for filters designed with DPSW, and similarly Figure 16 for Kaiser, and Figure 17 for Parks-McClellan. Since the Parks-McClellan algorithm requires a finite transition region, it was specified symmetrically around the desired cutoff frequency and varied. A comparison of these four methods, along with Hamming, Hanning, and Rectangular windowing is given in Figure 18.
Figure 14. Tradeoff Curve between Chebyshev Error and Integrated Squared Error for Lowpass Filters designed using the Constrained Least Squares Method.

Figure 15. Tradeoff Curve between Chebyshev Error and Integrated Squared Error for Lowpass Filters designed using the Discrete Prolate Spheroidal Window Method.
Figure 16. Tradeoff Curve between Chebyshev Error and Integrated Squared Error for Lowpass Filters designed using the Kaiser Window Method.

Figure 17. Tradeoff Curve between Chebyshev Error and Integrated Squared Error for Lowpass Filters designed using the Parks-McClellan Method.
It is evident from Figure 17 that the optimal constrained least squares method produces the best results, that the Parks-McClellan algorithm is able to match constrained least squares for some portions of the tradeoff curve, and that the discrete prolate spheroidal window is the best among the window functions. It is interesting to note that although the Kaiser window is designed as an approximation to the DPSW, it has slightly worse performance. Furthermore, the Hamming and Hanning windows are clearly suboptimal.

A listing of MATLAB code for generating various filters with different methods is given in Appendix 4.

XII. On the Use of Discrete Prolate Spheroidal Windows for Spectral Analysis

Although the discrete prolate spheroidal window was found to be suboptimal for filter design in the integrated squared error-Chebyshev error tradeoff, it was found to be the best window for this application among those compared. This result suggests that the discrete prolate spheroidal window may find use as a good window for spectral analysis. Windowing is in fact,
more central to spectral analysis than to filter design, which allows greater use of optimization theory. The slightly better performance of the DPSW over the Kaiser window suggests that this benefit would carry over into spectral analysis and since the Kaiser window is widely used for spectral analysis, the DPSW may provide performance gains in that field.

In order to facilitate the use of discrete prolate spheroidal windows for spectral analysis, Verma, Bilbao, and Meng have determined empirical formulas relating the window length and the bandwidth concentration factor $W$ to relative peak sidelobe height [20], similar to the empirical formulas determined for Kaiser windows [21].
References


Appendix 1 – MATLAB Code for Computing Discrete Prolate Spheroidal Windows

function X = prolate(N,W)
%PROLATE Digital Prolate Spheroidal Window.
% PROLATE(N,W) returns the W-valued N-point Digital Prolate Spheroidal Window.
% See also KAISER, BARTLETT, BLACKMAN, BOXCAR, CHEBWIN, HAMMING, HANN,
% and TRIANG.

% Lav Varshney, Cornell University, ECE 426
% February 6, 2004

even = 0;
if(mod(N,2) == 0)
    N = 2*N + 1;
    even = 1;
end

%set up the tridiagonal T matrix
T = diag(((N-1)/2-(1:N)+1).^2)*cos(2*pi*W) + ...
    diag(0.5.*((2:N)-1).*(N-(2:N)+1),1) + diag(0.5.*((2:N)-1).*(N-(2:N)+1),-1);

%compute the eigenvector associated with the dominant eigenvalue
X = eigpower(T);
if(even)
    X = X(2:2:end);
end

function [x,lambda] = eigpower(A,tol,x)
%EIGPOWER Calculate dominant eigenvalue, corresponding eigenvector using power method
% [V,L] = EIGPOWER(A,T,X) calculates the dominant eigenvalue, L, and corresponding
% eigenvector, V, of the matrix A. T is an argument that specifies the tolerance
% for convergence of the eigenvalue computed. X is an argument that represents an
% initial guess for the eigenvector. The maximum element of X must be 1.
% [V,L] = EIGPOWER(A,T) uses a default starting guess of [1 zeros(1,length(A)-1)]'
% [V,L] = EIGPOWER(A) uses same default starting guess, and a tolerance of 10^-16
% See also EIG, EIGS.

% Lav Varshney, Cornell University, ECE 426
% February 6, 2004

%make sure matrix is square
[m,n] = size(A);
if (m ~= n)
    error('Matrix must be square.');</n
if (nargin == 3)
    if(length(x) ~= m)
        error('Initial guess must be same size as matrix.');
    end
    if(max(abs(x)) ~= 1)
        error('Initial guess must have largest entry 1 (in magnitude).');
    end
else
    x = [1 zeros(1,m-1)]
end
%set default tolerance
if (nargin == 1)
    tol = 1e-16;
end

%maximum number of iterations
maxiter = 100000;

%stopping criterion, change in eigenvalue between iterations
dif = Inf;
lambda_old = Inf;

%make the matrix sparse, to quicken arithmetic operations
A = sparse(A);

ii = 0;
while((ii < maxiter) & (dif > tol))
    y = A*x;
    [t r] = max(abs(y));

    %estimate for eigenvalue
    lambda = y(r);

    %estimate for eigenvector
    x = y/y(r);

    dif = abs(lambda_old - lambda);
    lambda_old = lambda;
    ii = ii+1;
end
Appendix 2 – MATLAB Code for Calculating Error Parameters for Filters

function [delP,delS,wid] = err(h)
%ERR Determines the error criteria values for the filter.
% [DP,DS,W] = ERR(h) calculates the maximum passband ripple, DP,
% the maximum stopband ripple, DS, and the transition width, W. The ripple
% values are in dB. The input parameter is the FIR unit pulse response
% of the filter in question, h.
% Lav Varshney, Cornell University, ECE 426
% February 21, 2004

% calculate the frequency response of the filter
[H,W] = freqz(h,1,10000,1);

%%%%%
% calculate the maximum value of frequency magnitude to determine passband ripple
[delp,delP_ii] = max(abs(H));

% subtract ideal to get ripple value (assume ideal has magnitude 1)
delp = delp - 1;

% convert to dB
delP = 20*log10(delp);

%%%%%
% calculate the frequency at which the first zero crossing takes place
% this is accomplished by taking advantage of linear phase with pi phase
% shifts at zero crossings, so the minimum of the unwrapped angle is the
% first zero crossing
[m,min_ph_ii] = min(unwrap(angle(H)));

% only consider portion of frequency response after first zero crossing
Hs = H(min_ph_ii:end);

% stopband ripple is the maximum value in this region (assume ideal has 0 magnitude).
[delS,delS_ii] = max(20*log10(abs(Hs)));
dels = 10^(delS/20);

%%%%%
% to calculate transition width, only consider frequencies between
% maximum passband ripple and first zero crossing
Ht = H(delP_ii:min_ph_ii);
Wt = W(delP_ii:min_ph_ii);

% find the sample that is closest to (1 - delp)
[edge_p, edge_p_ii] = min(abs(abs(Ht) - (1-delp)));
edge_p_w = Wt(edge_p_ii);

% find the sample that is closest to dels
[edge_s, edge_s_ii] = min(abs(abs(Ht) - dels));
edge_s_w = Wt(edge_s_ii);

% difference is the transition width
wid = edge_s_w - edge_p_w;

% convert to positive dB.
delS = -delS;
delP = -delP;
function Q = ise(H, H_ideal, M)
%ISE Integrated Squared Error.
% Q = ISE(H,HI,M) calculates the integrated squared error between the
% filter frequency response H, and the ideal response HI. An M-point
% composite Newton-Cotes quadrature rule is used to approximate this value.
% M = 2 corresponds to the trapezoid rule, and M = 3 corresponds to
% Simpson's Rule. Only 2 <= M <= 11 are allowed.

% absolute squared error
E2 = (abs(abs(H_ideal)-abs(H))).^2;

%starting frequency point
a = 0;
%ending frequency point
b = 0.5;

%weights to perform Newton-Cotes rule as inner product
w = NCWeights(M);

%estimate of integral
Q = 0;

%first subinterval for composite rule
first = 1;
last = M;

%iterate through all subintervals
for ii = 1:(length(E2)-1)/(M-1)
    %apply Newton-Cotes rule on subinterval, and increase estimate
    Q = Q + (w'*E2(first:last));
    %update subinterval
    first = last;
    last = last+m-1;
end

%normalize for number of subintervals and subinterval length (frequency)
Q = (b-a)/((length(E2)-1)/(m-1))*Q;

function delp = err2(H)
%ERR2 Determines the maximum passband ripple for the filter.
% DP = ERR2(H) calculates the maximum passband ripple, DP.
% The input parameter is the frequency response
% of the filter in question, H.

% absolute Varshney, Cornell University, ECE 426
% February 22, 2004

%determine the maximum value of frequency magnitude to determine passband ripple
[delp,delP_ii] = max(abs(H));

%subtract ideal to get ripple value (assume ideal has magnitude 1)
delp = delp - 1;
function w = NCWeights(m)
%NCWEIGHTS Newton-Cotes Numerical Integration Weight Vector.
% NCWEIGHTS(M) Returns the weights for the M-point Newton-Cotes rule.
% M is an integer that is between 2 and 11, inclusive

% This function is modified from C.F. Van Loan, Introduction to Scientific Computing.
% Lav Varshney, Cornell University, ECE 426
% February 17, 2004

if m == 1
    w = 1;
elseif m == 2
    w = [1 1]'/2;
elseif m == 3
    w = [1 4 1]'/6;
elseif m == 4
    w = [1 3 3 1]'/8;
elseif m == 5
    w = [7 32 12 32 7]'/90;
elseif m == 6
    w = [19 75 50 75 19]'/288;
elseif m == 7
    w = [41 216 27 216 41]'/840;
elseif m == 8
    w = [751 3577 1323 2989 2989 1323 3577 751]'/17280;
elseif m == 9
    w = [989 5888 -928 10496 -4540 10496 -928 5888 989]'/28350;
elseif m == 10
    w = [2857 15741 1080 19344 5778 5778 19344 1080 15741 2857]'/89600;
elseif m == 11
    w = [16067 106300 -48525 272400 -260550 427368 -260550 272400 -48525 106300 16067]'/598752;
else
    w = [];
end
Appendix 4 – MATLAB Code for Comparison of Various Filters

%ideal filter
W = 0:5e-5:0.49995;
Hideal = ones(10000,1);
Hideal(find(W>.22)) = 0;

N = 51;

%constrained least squares
del = logspace(-1,-5,200);
jj = 1;
for dd = del
    h_fircls = fircls(N-1,[0 .44 1],[1 0],[1+dd dd],[1-dd -dd]);
    H_fircls = abs(freqz(h_fircls,1,10000,1));
    msse(ii,jj) = ise(H_fircls,Hideal,4);
    jj = jj+1;
end

%discrete prolate spheroidal window
h = firls(N-1,2*[0 .22 .22 .5],[1 1 0 0]);
B = linspace(.001,.25,200);
jj = 1;
for dd = B
    win = prolate(N,dd).';
    hwin = h.*win;
    Hwin = freqz(hwin,1,10000,1);
    del(jj) = err2(Hwin);
    msse(jj) = ise(Hwin,Hideal,4);
    jj = jj+1;
end

%Kaiser Window
h = firls(N-1,2*[0 .22 .22 .5],[1 1 0 0]);
beta = linspace(0,6,200);
jj = 1;
for dd = beta
    win = kaiser(51,dd).';
    hwin = h.*win;
    Hwin = freqz(hwin,1,10000,1);
    del(jj) = err2(Hwin);
    msse(jj) = ise(Hwin,Hideal,4);
    jj = jj+1;
end

%Parks-McClellan
t_wid = linspace(eps,.1,200);
jj = 1;
for dd = t_wid
    hpm = remez(50,2*[0 .22-dd .22+dd .5],[1 1 0 0]);
    Hpm = freqz(hpm,1,10000,1);
    del(jj) = err2(Hpm);
    msse(jj) = ise(Hpm,Hideal,4);
    jj = jj+1;
end
%Rectangular
h = fir1(N-1,2*[0 .22 .22 .5],[1 1 0 0]);
[H,W] = freqz(h,1,10000,1);
del = err2(H);
msse = ise(H,Hideal,4);

%Hamming
h = fir1(N-1,2*[0 .22 .22 .5],[1 1 0 0]);
win = hamming(51).';
hhamm = h.*win;
Hhamm = freqz(hhamm,1,10000,1);
del = err2(Hhamm);
msse = ise(Hhamm,Hideal,4);

%Hanning
h = fir1(N-1,2*[0 .22 .22 .5],[1 1 0 0]);
win = hanning(51).';
hhann = h.*win;
Hhann = freqz(hhann,1,10000,1);
del = err2(Hhann);
msse = ise(Hhann,Hideal,4);