



Time Windows for Linear Prediction of Speech

Peter Kabal

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

Version 1 October 2003
Version 2 December 2005
Version 2a November 2009

© 2009 Peter Kabal

Abstract

This report examines the time windows used for linear prediction (LP) analysis of speech. The goal of windowing is to create frames of data each of which will be used to calculate an autocorrelation sequence. Several factors enter into the choice of window. The time and spectral properties of Hamming and Hann windows are examined. We also consider windows based on Discrete Prolate Spherical Sequences including multiwindow analysis. Multiwindow analysis biases the estimation of the correlation more than single window analysis. Windows with frequency responses based on the ultraspherical polynomials are discussed. This family of windows includes Dolph-Chebyshev and Saramäki windows. This report also considers asymmetrical windows as used in modern speech coders. The frequency response of these windows is poor relative to conventional windows. Finally, the presence of a “pedestal” in the time window (as in the case of a Hamming window) is shown to be deleterious to the time evolution of the LP parameters.

Time Windows for Linear Prediction of Speech

1 Introduction

This report examines time windows used in linear prediction (LP) analysis of speech. The goal of the windowing is to create frames of data each of which will be used to calculate an autocorrelation sequence. The low-order correlation values are used to generate a LP fit to the speech spectrum. Several factors enter into the choice of window. Both the time and frequency properties are important. The properties of Hamming and Hann windows are examined. A modified version of these windows is suggested.

We also consider windows based on Discrete Prolate Spherical Sequences (DPSS). It is demonstrated that with a proper choice of the time-bandwidth parameter, a DPSS window is close to a Hamming window. Multiwindow analysis can be implemented using an orthonormal family of DPSS windows. The DPSS windows have the property of maximally concentrating the energy in the main lobe of the frequency response. Multiwindow analysis biases the estimation of the correlation more than single window analysis. This report also considers asymmetrical windows as used in modern speech coders. The frequency response of these windows is poor relative to conventional windows. The presence of a "pedestal" in the time window (as in the case of a Hamming window) is shown to be deleterious in the time evolution of the LP parameters. This report also discusses windows based on the ultraspherical polynomials.

An earlier version of this report including material on windows for transform processing. This material is now in a separate report [1].

2 Linear Predictive Analysis

Linear predictive analysis fits an all-pole model to the local spectrum of a (speech) signal. The model is derived from the autocorrelation sequence of a segment of the speech. The LP spectral fit is determined by solving a set of linear equations based on the correlation values.

Let the input signal be $x[n]$. This signal is windowed,

$$x_w[n] = w[n] x[n]. \quad (1)$$

The linear prediction formulation minimizes the difference between the windowed signal and a linear combination of past values of the windowed signal,

$$e[n] = x_w[n] - \sum_{k=1}^{N_p} p_k x_w[n-k]. \quad (2)$$

The goal is to minimize the total squared error,

$$\varepsilon = \sum_{n=-\infty}^{\infty} |e[n]|^2. \quad (3)$$

For the case that the window is finite in length, the terms in the sum for the squared error will be non-zero only over a finite interval.

The predictor coefficients (p_k) which minimize ε can be found from the following set of equations

$$\begin{bmatrix} r[0] & r[1] & \cdots & r[N_p - 1] \\ r[1] & r[0] & \cdots & r[N_p - 2] \\ \vdots & \vdots & \ddots & \vdots \\ r[N_p - 1] & r[N_p - 2] & \cdots & r[0] \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{N_p} \end{bmatrix} = \begin{bmatrix} r[1] \\ r[2] \\ \vdots \\ r[N_p] \end{bmatrix}. \quad (4)$$

The autocorrelation values are given by

$$r[k] = \sum_{n=-\infty}^{\infty} x_w[n] x_w[n-k]. \quad (5)$$

Again for finite length windows, the sum needs be evaluated only over a finite interval — the rest of the correlation coefficients will be zero. In vector-matrix notation,

$$\mathbf{R}\mathbf{c} = \mathbf{r}. \quad (6)$$

Let the prediction error filter be denoted by $A(z)$,

$$A(z) = 1 - \sum_{k=1}^{N_p} p_k z^{-k}. \quad (7)$$

The autocorrelation formulation for the optimal prediction coefficients gives a matrix \mathbf{R} which is

Toeplitz. The Levinson-Durbin algorithm can be used to efficiently solve for the predictor coefficients. The prediction error filter ($A(z)$) will be minimum phase and the corresponding synthesis filter $1/A(z)$ will be stable. The frequency response (power spectrum) of the synthesis filter serves as a model of the signal spectrum.

3 Time Windows in Linear Prediction Analysis of Speech

The focus is on discrete-time windows, but lessons from continuous-time still apply. A continuous-time window which has a finite discontinuity (for example, a rectangular window) has a frequency response that falls off as $1/f$ asymptotically. A continuous-time window which is continuous but with a finite discontinuity in the first derivative (for example, a triangular window) has a frequency response that falls off as $1/f^2$. A continuous-time window which has a finite discontinuity in the second derivative (for example, a Hann window) has a frequency response that falls off asymptotically as $1/f^3$. Smoothness implies a more rapid fall off.¹ However, smoothness influences the effective length of the window. Windows with significant tapers can have an effective length which is substantially less than the full length.

The effect of a time window can be described in the frequency domain as a convolution of the frequency response of the window with the frequency response of the signal. The convolution smears frequency features, with the amount of smearing depending on the width of the main lobe of the window frequency response. In addition, spectral leakage from distant frequency components will occur if the sidelobe level of the window response is large.

Additional bandwidth expansion prior to LP analysis can be implemented by lag windowing the autocorrelation sequence, often with a Gaussian window. Lag windowing will not be explored in this report. See [3] for more on bandwidth expansion.

One important property of windows is the window length. In speech coding, a window length of 30 ms (240 samples at a sampling rate of 8 kHz) has been found to be a reasonable compromise in terms of the dynamics of speech production. The window has to be long enough that correlation values can be estimated by averaging lagged values, but not too long such that the local statistical properties of the signal change significantly within the window span. In fact, the update interval is often smaller than the window lengths (20 ms or 160 samples, for instance). This means that the windows used for LP analysis of adjacent frames overlap. This window length will be a constant for our window comparisons. It is to be noted, that speech coding systems offer some robustness to poor LP analysis. For most speech coders, the LP residual signal is coded for transmission and can compensate somewhat for inadequate spectral (LP) modelling.

¹A proof of the statement of the relationship between discontinuities in the derivatives and frequency roll-off is given in [2].

3.1 Sampling Continuous-Time Windows

Many of the standard windows are sampled versions of continuous-time windows. There is a question of where to sample the continuous-time window. Consider a continuous-time window $w(t)$ of length W which is symmetrical about zero. A discrete-time window can be created by sampling this window at N uniformly spaced points from t_0 to t_1 , where the end points of the discrete-time window are assumed to lie in the interval $[-W/2, W/2]$,

$$w[n] = w\left(t_0 + n \frac{t_1 - t_0}{N - 1}\right). \quad (8)$$

The discrete-time window is a one-sided window. It will be symmetric about its middle if the continuous-time window is symmetric and $t_1 = -t_0$.

The sampling interval is $\Delta t = (t_1 - t_0)/(N - 1)$. In an early paper by Harris [4], the suggestion was to set $t_0 = -W/2$ and $\Delta t = W/N$ which gives $t_1 = W - \Delta t$. This is an unsymmetrical sampling arrangement. For windows which are zero at the ends (for instance the triangular window or the Hann window), we can then omit the first point and get an $N - 1$ point window, or we can add an additional zero point at the end and get an $N + 1$ point window. Many recent textbooks choose $t_1 = -t_0 = W/2$ for the conventional windows. With these end points, $\Delta t = W/(N - 1)$. We argue that this is not usually the best choice.

Our preference is to use the same sampling interval as suggested by Harris but shift the sampling points by half of a sample. This gives symmetrical windows with desirable periodic extension properties. The modified sampling pattern is $\Delta t = W/N$, giving $t_1 = -t_0 = W/2 - \Delta t/2$ and

$$w[n] = w\left(\frac{W}{2} \frac{2n - N + 1}{N}\right). \quad (9)$$

This choice gives the smallest value of Δt while still covering the continuous-time window. A small value of sampling interval will generally reduce the aliasing due to sampling of the continuous-time window. If we view the Discrete-Time Fourier Transform as an approximation to the continuous-time Fourier Transform, the approximation to integration is improved with this choice of Δt . Also some continuous-time windows are composed of the sum of sinusoidal terms which evolve through a multiple of a half period across the window. This choice of Δt ensures that the discrete-time window also has sinusoidal terms which evolve in step. We will refer to discrete-time windows derived from continuous-time windows with this sampling interval as *modified* versions of standard windows.²

²The modified sampling pattern generally results in a slightly narrower main lobe for the same sidelobe attenuation.

3.2 Hann and Hamming Windows

Hann and Hamming windows fall in the class of raised-cosine windows. They are both commonly used in speech and audio processing. The modified Hann and Hamming windows are given by³

$$w[n] = \begin{cases} \frac{1+\alpha}{2} - \frac{1-\alpha}{2} \cos\left(\frac{\pi(2n+1)}{N}\right), & 0 \leq n \leq N-1, \\ 0, & \text{elsewhere.} \end{cases} \quad (10)$$

Altering α allows the characteristics to change from a rectangular window ($\alpha = 1$), to a Hamming window ($\alpha = 0.08$), to the Hann window ($\alpha = 0$). Appendix A analyzes the modified Hann windows both from the point of view of a product of time sequences (modulation) and as a convolution of time sequences.

The overall normalized frequency response for the Hann window is plotted on a dB scale in Fig. 1. As expected, the smoothness of the window (continuous-time version smooth up to the second derivative), leads to a steep fall-off of the sidelobes, although the first sidelobe is uncomfortably large at only 31.5 dB down from the peak at $\omega = 0$.

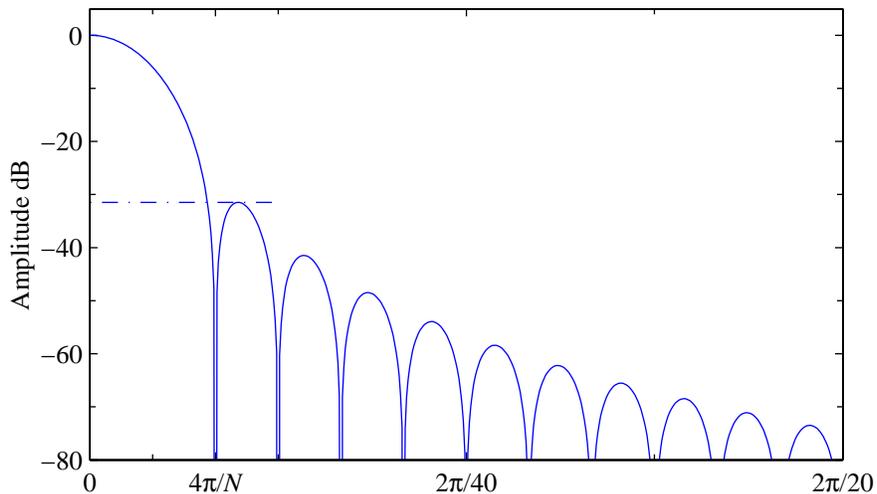


Figure 1 Normalized frequency response of a modified Hann window ($N = 240$). The broken horizontal line is at -31.5 dB.

For the Hamming window $\alpha = 0.08$. In effect the window sits on a rectangular pedestal. The pedestal increases the attenuation of the near-in sidelobes in the frequency response (see Appendix A). However, the addition of the pedestal makes the overall function discontinuous. Thus

³The modified versions of the Hann and Hamming windows appear in Mitra [5], although there they are symmetrical about zero and have an odd number of coefficients. The modified versions are common in the array processing literature, see for instance [6].

the Hamming window has better near-in sidelobe suppression at the expense of poorer far-out suppression. This is illustrated in Fig. 2 which can be compared to the corresponding figure for the Hann window (Fig. 1). The minimum sidelobe attenuation is 42.7 dB.

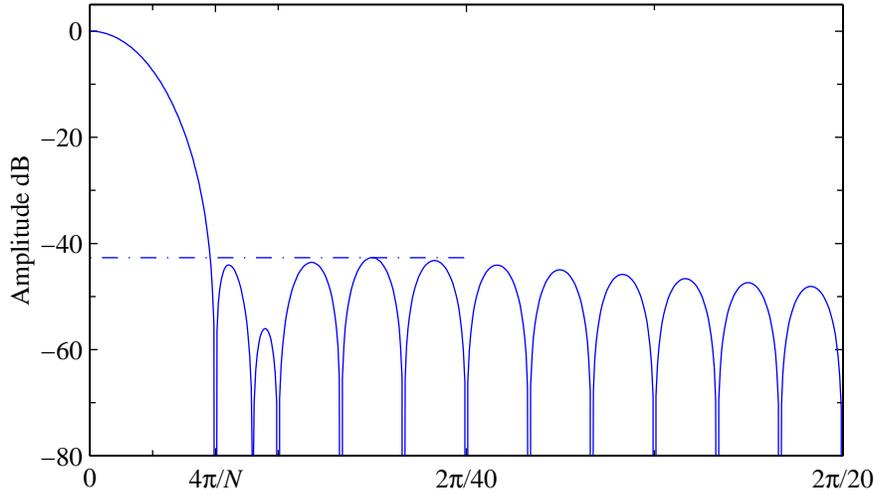


Figure 2 Normalized frequency response of a modified Hamming window ($N = 240$). The broken horizontal line is at -42.7 dB.

For a Hann or Hamming window with 8 kHz sampling, the main lobe width of the frequency response is 133 Hz between zero crossings.

4 Discrete Prolate Spheroidal Sequences

The discrete prolate spheroidal sequences (DPSS's) are those finite length sequences which concentrate the maximum amount of energy in a given bandwidth [7]. The DPSS's are parameterized by the time bandwidth product NW , where W is the normalized one-sided bandwidth in Hz. There is no closed form for these windows.⁴ However, windows are most often pre-computed and stored, so the complexity of the functional description of the window shape need not be a concern.⁵

Like the Hamming window, the DPSS's also sit on pedestals. The DPSS window which has the most energy within $|\omega| \leq 3.5\pi/N$ ($NW = 1.75$) closely matches the central part of a Hamming window. These two windows are plotted in Fig. 3. The solid line is the modified Hamming window ($N = 240$). The dashed line is the discrete prolate spheroidal sequence of the same length. The DPSS sequence has been normalized such that the interpolated value mid-way between the

⁴The discrete prolate spheroidal sequences can be calculated using the `dpss` function in MATLAB.

⁵The Kaiser-Bessel window [8] is an approximation to the zeroth order DPSS. See Section 5.1.

largest samples is unity.

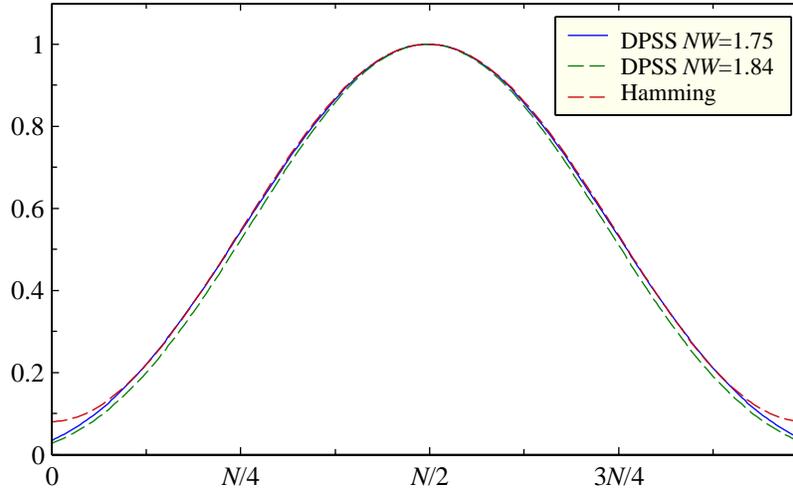


Figure 3 Discrete prolate spheroidal sequence window, time-bandwidth product $NW = 1.75$ (solid line) and DPSS window, time-bandwidth product $NW = 1.84$ (lower dashed line). Also shown is the Hamming window (upper dashed line). The DPSS windows have been normalized to have a maximum height of unity. All windows have length $N = 240$.

The normalized frequency response of the DPSS window is plotted in Fig. 4. The main lobe width is slightly smaller than that for either the Hann or Hamming window. The first sidelobe attenuation (38.8 dB) is between that for a Hann window (31.5 dB) and that for a Hamming window (42.7 dB). Likewise, the rate of fall-off of the side-bands is also between that of the Hann and Hamming windows.

For the sample DPSS above, the bandwidth was chosen so that the central part of the time window closely matches the Hamming window. Choosing the time-bandwidth product to be $NW = 1.84$ gives a main lobe width in the frequency domain equal to that of the Hamming window. For this choice of bandwidth, the first sidelobe attenuation is now 40.5 dB. The pedestal on which the window sits is reduced (see Fig. 3).

4.1 Multiwindow Analysis with DPSS Windows

The DPSS window above is the first in a family of orthogonal windows for multiwindow (multi-taper) spectral analysis [7]. The number of windows (M) with significant concentration of energy in the given bandwidth is determined by the time-bandwidth product

$$M \leq 2NW - 1, \quad (11)$$

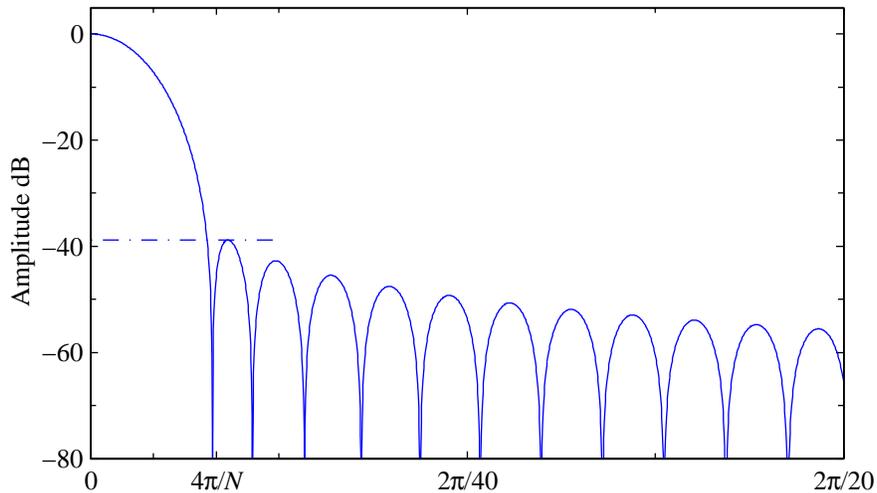


Figure 4 Normalized frequency response of a discrete prolate spheroidal sequence window (time-bandwidth product $NW = 1.75$, $N = 240$). The broken horizontal line is at -38.8 dB.

where W is the one-sided bandwidth in Hz. Multiwindow analysis estimates the power spectrum by averaging the spectral estimates using each window. The resulting averaged estimate can have a reduced variance. However, in the speech processing application that is the focus of this work, it is the low-order correlation terms that need to be estimated. A lower variance in the power spectral estimate does not necessarily translate to a better correlation estimate.

For the case of $NW = 1.75$ and $N = 240$, the first three windows are shown in Fig. 5. The first window has 99.98 % of the energy within the design bandwidth. The corresponding figures for the second and third windows are 99.09 % and 88.64 %. If a fourth window had been used, it would have less than 50 % of its energy within the design bandwidth.

The autocorrelation can be calculated for each of the windows and the final correlation obtained as the average of these correlations. For this analysis, it is assumed that each of the windows, $w_m[n]$ is normalized to unit energy. The correlation estimate using window m is (c.f. Eq. (1) and Eq. (5)),

$$r_m[k] = \sum_{n=-\infty}^{\infty} w_m[n]x[n]w_m[n-k]x[n-k]. \quad (12)$$

The final correlation estimate is

$$\hat{r}[k] = \sum_{m=0}^{M-1} \beta_m r_m[k]. \quad (13)$$

There are different choices for the weighting factors, β_m . Here we will consider only simple averaging, $\beta_m = 1/M$.

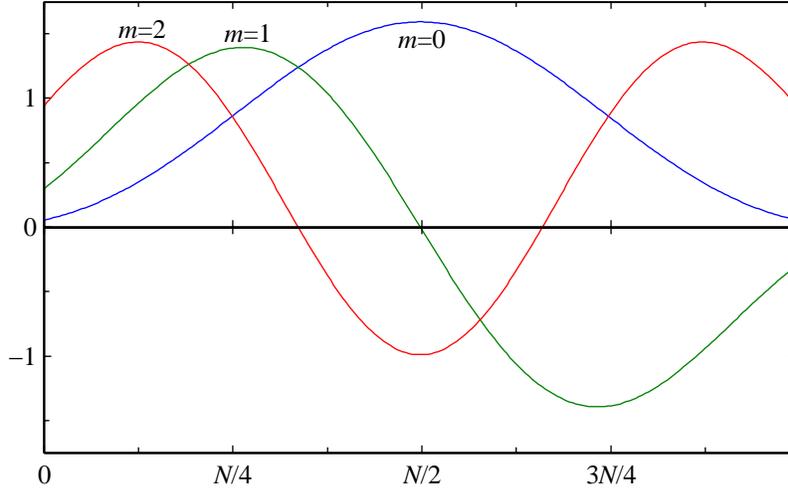


Figure 5 Discrete prolate spheroidal sequence windows for a time-bandwidth product of $NW = 1.75$ and $N = 240$. The amplitudes have been multiplied by \sqrt{N} .

Following the development in [9], the expected value of $\hat{r}[k]$ is given as

$$E[\hat{r}[k]] = Q[k]r[k], \quad (14)$$

where $r[k]$ is the true correlation and $Q[k]$ is the average correlation of the window sequences,

$$Q[k] = \frac{1}{M} \sum_{m=0}^{M-1} \sum_{n=-\infty}^{\infty} w_m[n]w_m[n-k]. \quad (15)$$

The formulation in Eq. (14) has the form of a lag window, $Q[k]$, acting on the correlation lags. Note, however, that the true correlation values appear on the righthand side of this equation. For lag windowing to be applied, the correlation that it acts on would be an estimated correlation, calculated after applying a data window to isolate a segment of speech. For the typical frame lengths used in speech coding, the effect of the data window cannot be neglected. As such, the effect of the multiple windows acting on the data signal cannot be replaced by a lag window. A further discussion of the relationship between lag windows and multiwindow analysis can be found in [10].

Let the time-bandwidth product be $NW = 1.75$ and the window be of length $N = 240$ as before. The function $Q[k]$ is plotted in Fig. 6 for different values of M . The function $Q[k]$ can be used to measure the correlation estimation bias. The bias is the difference between the average estimated correlation value and the true correlation value. Deviations of $Q[k]$ from unity result in a larger bias. Since the first DPSS window for this time-bandwidth product is close to a Hamming

window, the curve for a Hamming window would fall nearly on top of the curve for $M = 1$. The function is always positive. For $M = 2$ and $M = 3$, the function predicts a severe bias for large lags. Since the function goes negative for certain lag values, the average correlation estimate will be of the wrong sign in those regions. Note however, that for LP analysis we only need lags 0 to 10 and the penalty in bias for the difference configurations for that lag range is small.

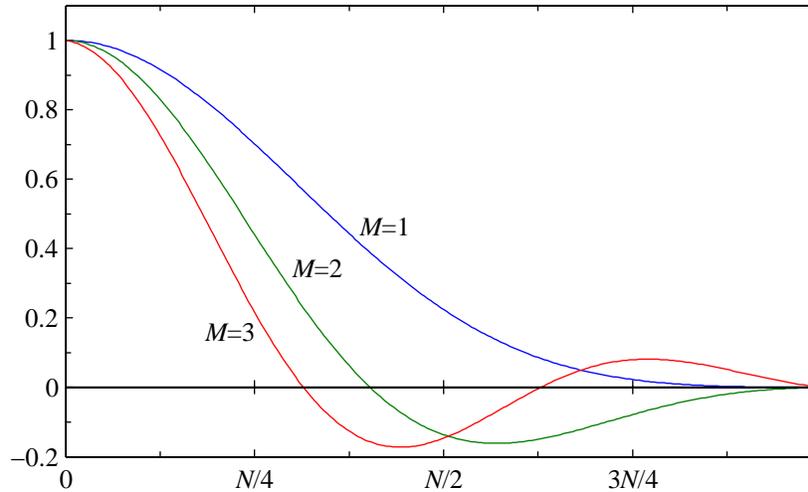


Figure 6 The quantity $Q[k]$ for DPSS windows different values of M (time-bandwidth product $NW = 1.75$, $N = 240$).

The equivalent frequency response of the multiwindow analysis can be obtained from the Fourier transform of the sum of the correlation values ($Q[k]$). This is plotted in Fig. 7. For $M = 3$, the main lobe is noticeably flatter than for a single window ($M = 1$). For this time-bandwidth product, the first sidelobe has an attenuation of only -16.6 dB, leading to a potential for severe spectral leakage. The sidelobes for the constituent windows add to give poor off-peak rejection. The curves for $M = 2$ and $M = 3$ show that there is a tradeoff with the number of windows. Fewer windows give lower sidelobes, but more windows reduce the variance of the spectral estimates.

Multiwindow analysis was applied to LP analysis. For the purposes of illustration, a sample frame was created as follows. White Gaussian noise was passed through a twelfth order all-pole filter to create the signal to be analyzed. For a signal sampled at 8 kHz, frames of length $N = 240$ were formed. For a given frame, first, a single window DPSS with time-bandwidth product $NW = 1.75$ was used to estimate the correlations. Multiwindow analysis was also performed ($M = 3$) using DPSS's with the same time-bandwidth product. The LP analysis was tenth order ($N_p = 10$). The results differ for different noise sequences, but Fig. 8 shows the LP spectral fits to the data for a typical frame. The ragged light line is the power spectrum computed from the data in the frame. The dotted line is the reference spectrum, i.e. the spectrum of the twelfth order all-pole

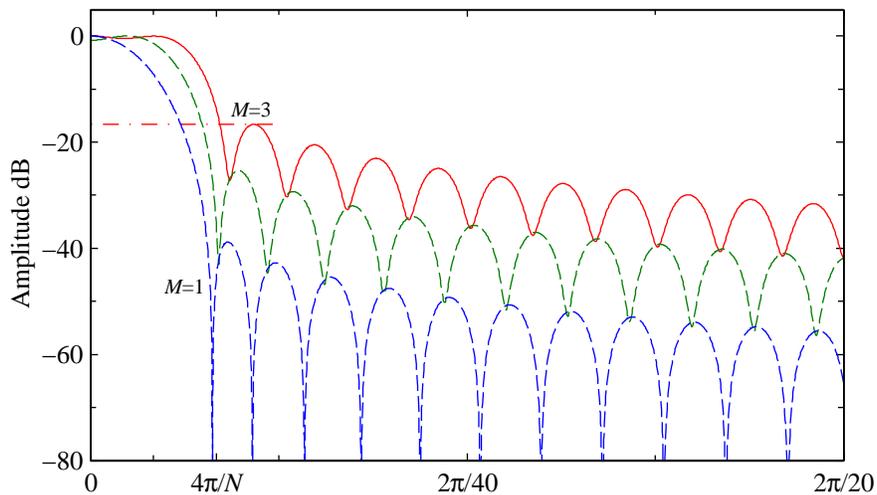


Figure 7 Normalized frequency response for multiwindow analysis using DPSS windows (time-bandwidth product $NW = 1.75$, $N = 240$). The broken horizontal line is at -16.6 dB.

filter. The smooth solid line is the LP fit using a single window. LP analysis with a Hamming window gives very similar results. The dashed line is the LP fit using multiwindow analysis. One can notice that for this data, the LP spectral fits differ considerably, with the multiwindow analysis giving a somewhat less peaky spectral fit. This was generally true, though the results change with different noise sequences. The broadening of the responses is consistent with the fact that the frequency response of the ensemble of windows gets flatter as more windows are used. This broadening smears the power spectrum.

4.2 Minimum Bias Windows

The sinusoidal windows introduced in [11] can also be used for multiwindow analysis. These windows minimize the “local bias” of the power spectral estimate. The local bias is the just the leading term of an expansion of the spectral bias, corresponding to a minimization of the second central moment of the power spectrum of the window response. For discrete-time windows, the window which minimizes this second moment is determined from an eigenvalue formulation. In fact, this formulation gives the orthogonal windows needed for a multiwindow analysis. As shown in [11], the minimum bias windows can be closely approximated by sinusoidal windows. The m th window in a multiwindow analysis is given in [11] by

$$w_m[n] = \sqrt{\frac{2}{N+1}} \sin\left(\frac{\pi(n+1)(m+1)}{N+1}\right) \quad 0 \leq n < N, 0 \leq m < N. \quad (16)$$

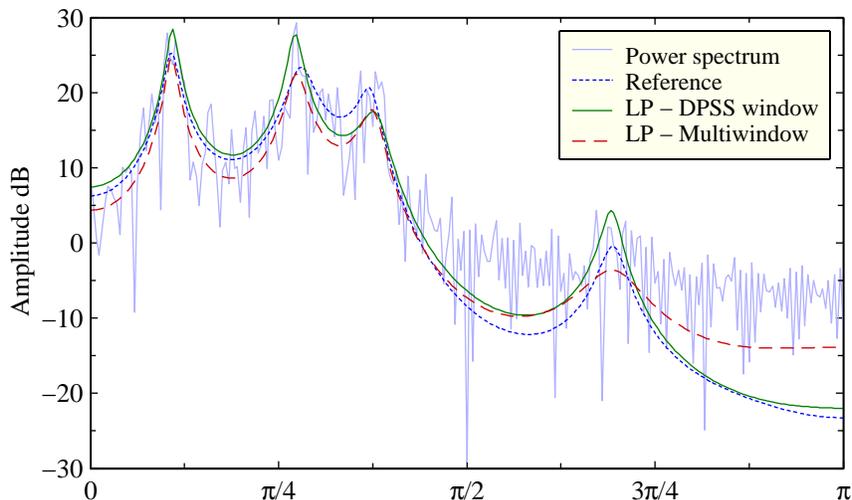


Figure 8 LP analysis for a frame of data ($N = 240$). The power spectrum is shown by the dotted line. The solid line is the tenth order LP spectral fit using a single DPSS window (time-bandwidth product $NW = 1.75$). The dashed is the tenth order LP spectral fit using multiwindow DPSS analysis ($M = 3$, time-bandwidth product $NW = 1.75$).

This window is normalized to unit energy. For $m = 0$, this window is (to within a scale factor) the square root of a standard Hann window.

If we sample the sinusoid to correspond to the sampling points of the modified Hann window, we get

$$w_m[n] = \sqrt{\frac{2}{N}} \sin\left(\frac{\pi(n + 1/2)(m + 1)}{N + 1}\right) \quad 0 \leq n < N, 0 \leq m < N. \quad (17)$$

This is the square root of a modified Hann window. It is also the window most-often used for modified discrete transform (MDCT) processing.

Unlike the DPSS's, the effective bandwidth increases with increasing numbers of these sinusoidal windows in a multiwindow analysis.

5 Adjustable Windows

The “fixed” windows such as the Hann and Hamming have one parameter, the window length N . The window length affects the main lobe width without substantially changing the sidelobe heights. We have met one “adjustable” window already, the prolate spheroidal window. For this type of window, the time-bandwidth product is adjustable. For a fixed N , the time-bandwidth product trades off main lobe width against sidelobe suppression. In this section, we discuss other adjustable windows and then show that they are part of, or closely related to, a family of adjustable

windows whose frequency responses are expressed in terms of orthogonal polynomials.

5.1 Kaiser-Bessel Windows

The Kaiser-Bessel window was formulated to be a simple-to-compute approximation to the zero'th order discrete prolate spheroidal sequences in the main lobe [8]. As such, it tends to maximize the energy in the main lobe for a given sidelobe level. The discrete-time Kaiser-Bessel window is a sampled version of a continuous function with a parameter β . This parameter is analogous to the time-bandwidth product and trades off main lobe width against side lobe attenuation. The standard form of the Kaiser-Bessel window is

$$w[n] = \frac{I_0\left(\beta\sqrt{1 - \left(\frac{2n}{N-1} - 1\right)^2}\right)}{I_0(\beta)} \quad 0 \leq n < N, \quad (18)$$

where $I_0(\cdot)$ is the zero'th-order modified Bessel function of the first kind. As β increases, the main lobe width widens and the side lobe attenuation increases. For $\beta = 0$, the Kaiser-Bessel window is a rectangular window. For $\beta = 5.3$, the Kaiser-Bessel window is close to a Hamming window.

Kaiser-Bessel windows are used for designing filters. In that application, a Kaiser-Bessel window is applied to the impulse response of an ideal filter. Empirical formulas for estimating N and β based on filter requirements (main lobe width and sidelobe attenuation) are available (see for instance [5]). The present author has used Kaiser-Bessel designs for on-the-fly design of lowpass filters for sample rate conversion [12].

We can apply the modified sampling pattern to the continuous-time Kaiser-Bessel window to obtain the modified Kaiser-Bessel window,

$$w[n] = \frac{I_0\left(\beta\sqrt{1 - \left(\frac{2n+1}{N} - 1\right)^2}\right)}{I_0(\beta)} \quad 0 \leq n < N. \quad (19)$$

The modified version gives a narrower main lobe for the same sidelobe suppression. The difference is most obvious for small N . At large N , with an appropriate adjustment to β , the standard and modified Kaiser-Bessel windows give very similar results.

The Kaiser-Bessel window is obtained by sampling a continuous-time window. As discussed in Appendix B, this continuous-time window is the limiting case of the time function corresponding to a frequency response determined by a Chebyshev polynomial of the second kind.

We can assess the match between the Kaiser-Bessel window and the discrete-time window found directly from the Chebyshev frequency response. These discrete-time windows are also known as the Saramäki windows [13]. For this evaluation, we generate Saramäki windows with

a 40 dB sidelobe suppression for different window lengths N . We then generate a discrete-time Kaiser-Bessel window of the same length. The error is measured as

$$\varepsilon = \frac{1}{N}(\mathbf{w}_s - a\mathbf{w}_k)^T(\mathbf{w}_s - a\mathbf{w}_k), \quad (20)$$

where \mathbf{w}_s is the vector of Saramäki window coefficients and \mathbf{w}_k is the vector of Kaiser-Bessel window coefficients. The scale factor a is chosen to minimize the mean-square error,

$$a = \frac{\mathbf{w}_s^T \mathbf{w}_k}{\mathbf{w}_k^T \mathbf{w}_k}. \quad (21)$$

The Kaiser-Bessel window is parameterized by β . For a given N , β is optimized to give the best match to the Saramäki window. The results are shown in Fig. 9 and show that the modified sampling pattern applied to the Kaiser-Bessel function is a much better fit to the Saramäki window. It should be noted that the value of the parameter β at a given N is different for the standard and modified Kaiser-Bessel windows.

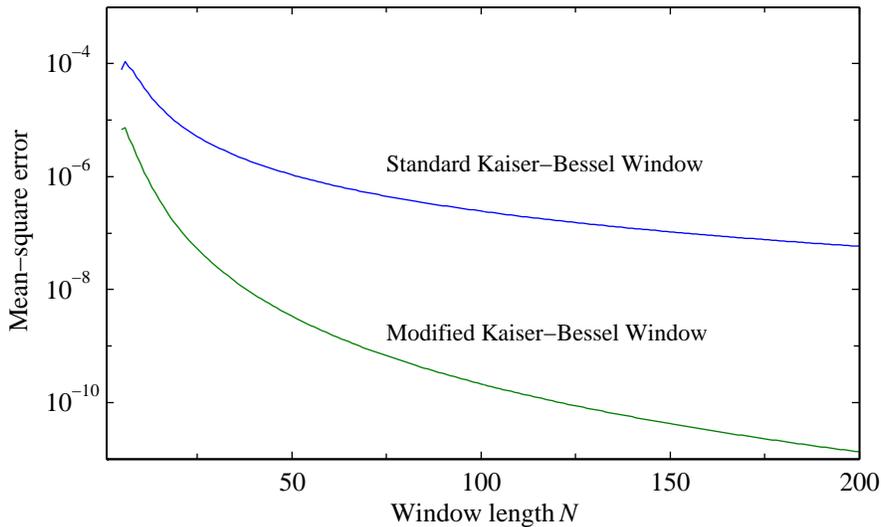


Figure 9 Mean-square error of a Kaiser-Bessel window relative to a Saramäki window (designed for 40 dB stopband attenuation). For each filter length, the parameter β of the Kaiser-Bessel window has been optimized to give the best fit to the Saramäki window.

5.2 Dolph-Chebyshev Windows

The Dolph-Chebyshev window [14] has the narrowest main lobe (measured between zero crossings) for a given side lobe attenuation. The Dolph-Chebyshev window has equiripple side lobes. This window allows one to trade off main lobe width against side lobe attenuation. The frequency response of the Dolph-Chebyshev window is given by a Chebyshev polynomial. The discrete-time window can be obtained by sampling the frequency response and using an inverse Discrete Fourier Transform as detailed in Appendix B. An explicit formula can also be developed for the window coefficients (see Appendix B).

The frequency response of a Dolph-Chebyshev window ($N = 240$) is shown in Fig. 10. For this plot, the sidelobe attenuation was set at 42.7 dB, the same as that for a Hamming window. For this sidelobe attenuation, the main lobe (between zeros) is 7 % narrower than that for a modified Hamming window with the same minimum sidelobe attenuation.

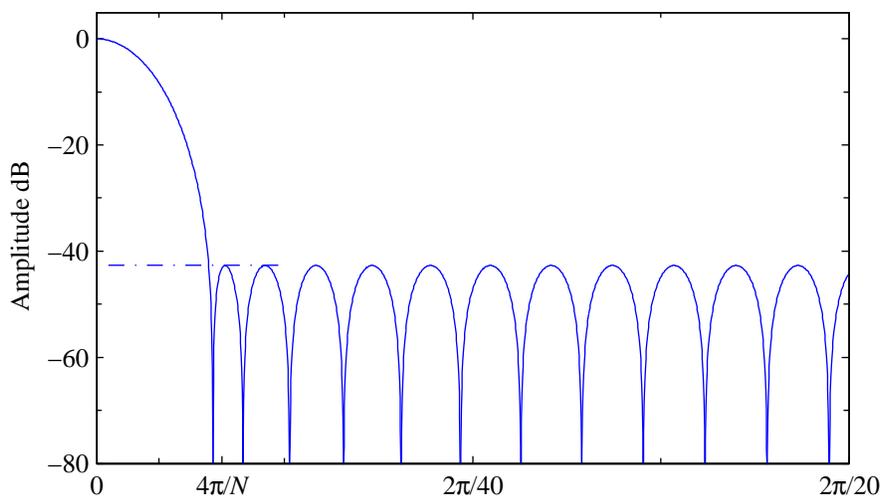


Figure 10 Normalized frequency response for a Dolph-Chebyshev window ($N = 240$). The sidelobe attenuation for this example is set at 42.7 dB (same as for a Hamming window).

The Dolph-Chebyshev window has a peculiarity. The outer samples of the window rise above their neighbours. This is more noticeable for large N — in the limit, the end points become impulses. The plot of the Dolph-Chebyshev window in Fig. 11 shows this effect. This window has been normalized so that the largest value is unity.⁶

The Dolph-Chebyshev window in the example above sits on a pedestal of about 0.07. We

⁶For N even, the two middle points of the window are unity. This is not the same scaling applied to a Hann or Hamming window. For those windows, the underlying continuous-time window is normalized to unit height and the two middle points of the discrete-time window would have values slightly less than unity.

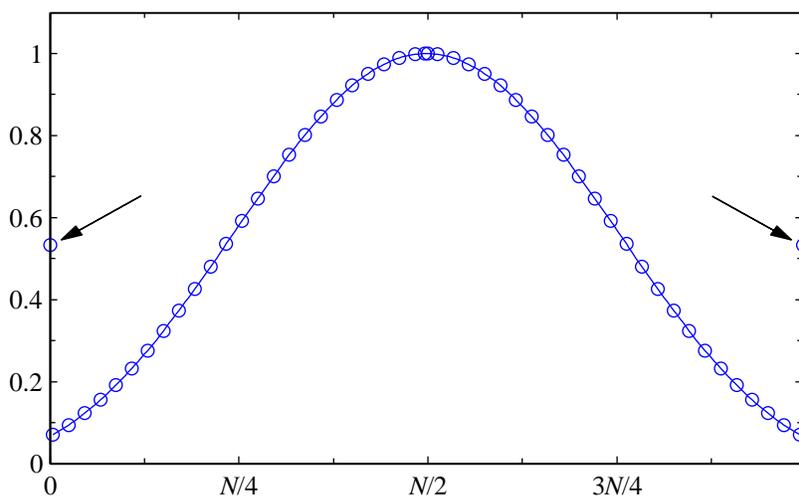


Figure 11 Dolph-Chebyshev window ($N = 240$) giving the frequency response shown in Fig. 10. The circles are drawn on only a subset of the samples of the window to reduce clutter. The arrows point to the end points of the window.

will see later in Section 7.1 that a pedestal on a window causes anomalous results in LP analysis. The large outer values would exacerbate these effects. The Dolph-Chebyshev window is not an appropriate candidate for LP analysis.

5.3 Ultraspherical Windows

The ultraspherical polynomials are a family of polynomials that are a generalization of Chebyshev and Legendre polynomials. A more detailed discussion is in Appendix B. The ultraspherical polynomial type is characterized by a parameter α . The parameter α sets the type of adjustable window. For one setting ($\alpha = 0$), we get Dolph-Chebyshev windows (based on Chebyshev polynomials of the first kind); for another setting ($\alpha = 1$), we get Saramäki windows (based on Chebyshev polynomials of the second kind) [13]. The parameter α controls how fast the sidelobe amplitudes fall off. In fact, for α negative, the sidelobe amplitudes increase with frequency. For such a setting, the window coefficients can become negative — normally not a desirable property.

Appendix B also gives expressions for continuous-time windows obtained as the limit of a large number of window coefficients. In the case of $\alpha = 1$, the continuous-time window is exactly the continuous-time Kaiser-Bessel window. This indicates that there is a close relationship between the sampled Kaiser-Bessel function and the discrete-time Saramäki window, as explored earlier in Section 5.1.

6 Asymmetrical Windows

In speech processing, one can distinguish between the LP parameter extraction process and the actual processing of a frame of speech samples. Most often the length of the analysis window used for parameter estimation is larger than the frame of speech to be processed. For symmetric windows, the centre of the analysis window is typically centred on the frame of speech, reaching both into samples before and after the speech frame. This of course means that an additional delay must be imposed for the look-ahead samples. An early paper considered the use of such asymmetrical windows [15]. Some low delay coders, for example the ITU-T G.729 coder [16], use an asymmetric analysis window. The peak of the window occurs such that the window emphasizes the more recent samples. In this way, the amount of look-ahead can be reduced while keeping the peak of the window centred on the speech frame.

We will consider the window of the form used in the ITU-T G.729 speech coder,

$$w[n] = \begin{cases} \frac{1+\alpha}{2} - \frac{1-\alpha}{2} \cos\left(\frac{2\pi n}{2N_L - 1}\right), & 0 \leq n \leq N_L - 1, \\ \cos\left(\frac{2\pi(n - N_L)}{4N_R - 1}\right), & N_L \leq n \leq N_L + N_R - 1, \\ 0, & \text{elsewhere.} \end{cases} \quad (22)$$

For the G.729 window, $\alpha = 0.08$, $N_L = 200$, and $N_R = 40$. This asymmetrical window consists of the first half of a traditional Hamming window ($\alpha = 0.08$, length 400) taking up 200 samples, followed by a cosine taper of length 40. The window is plotted in Fig. 12.

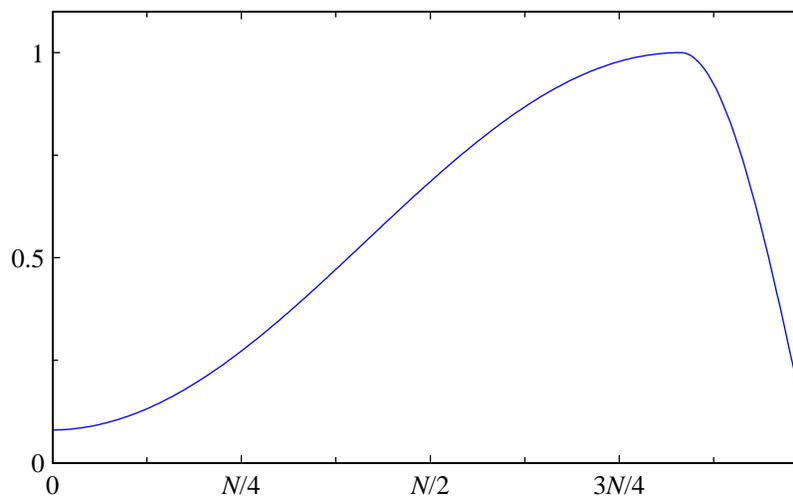


Figure 12 Asymmetrical G.729 window ($N = 240$).

The frequency response of this window is plotted in Fig. 13. The response does not show a clearly defined main lobe — there are no zero crossings in the vicinity of the main lobe. The attenuation at $\omega = 4\pi/N$ is only 18.1 dB. For comparison, the plot also shows the frequency response of a Hamming window of the same length. It can be seen that compared to traditional

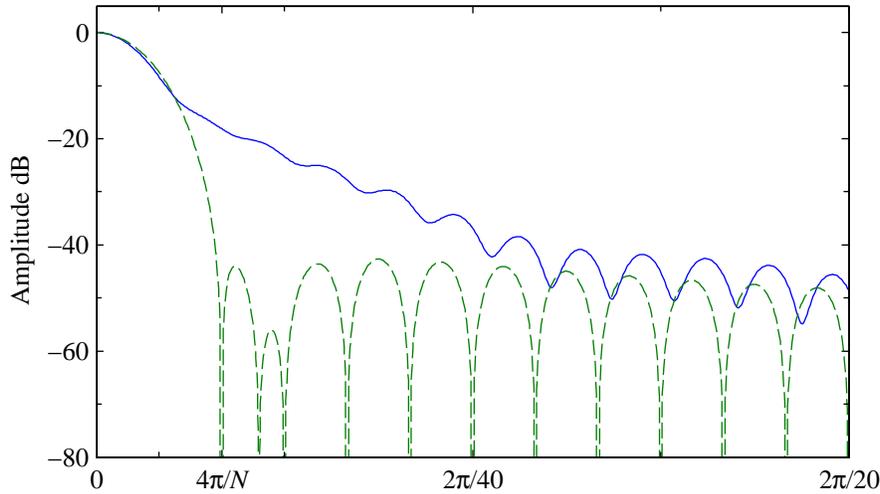


Figure 13 Frequency response (solid line) of the asymmetrical G.729 window ($N = 240$). The broken line is the frequency response of a modified Hamming window.

symmetric windows, this asymmetric window has a very poor frequency response.

It is interesting to note that the SMV coder standardized by 3GPP2 [17] cycles through three windows, each of length 240. The first is a Hamming window; the second marries the first half of a 300 sample Hamming window with the second half of a 180 sample Hamming window; and the third is very close to the asymmetrical window shown above. One can see the frequency response for two of these windows in Fig. 13. While the main lobes match to some degree, the amount of leakage would vary significantly when switching windows. One can speculate that even in steady sounds, the LP parameters will have spurious changes brought about by the window switching. Whether these changes affect the overall performance is an open question.

6.1 Modified Asymmetrical Window

As an attempt to improve the response of the G.729 asymmetric window, the definition of the asymmetrical window was modified in the same way as the conventional Hann and Hamming windows were modified. First, let us look at the problems with the conventional definition of the G.729 window.

Consider the discrete variable n to be integer-spaced samples of a continuous variable t . If we

look at the underlying continuous functions, we see a Hamming window starting at $t = 0$. The centre of the continuous Hamming window is at $t = N_L - 1/2$. The cosine rolloff begins at $t = N_L$ and goes to zero at $t = N_L + N_R - 1/4$. We note the following.

1. There is a half-sample “gap” between the end of the Hamming window and the start of the cosine rolloff.
2. The period of the cosine element of the Hamming window part ($2N_L - 1$) is mismatched to the length of the pedestal (N_L).
3. The period of the cosine rolloff is $4N_R - 1$, which is incommensurate with the period of the Hamming window part for the values used in the G.729 window.
4. The Hamming window part has a pedestal, while the cosine rolloff part does not.

A modified version of the asymmetrical window can be defined.

$$w[n] = \begin{cases} \frac{1+\alpha}{2} - \frac{1-\alpha}{2} \cos\left(\frac{\pi(2n+1)}{2N_L}\right), & 0 \leq n \leq N_L - 1, \\ \beta + (1-\beta) \cos\left(\frac{\pi(2(n-N_L)+1)}{4N_R}\right), & N_L \leq n \leq N_L + N_R - 1, \\ 0, & \text{elsewhere.} \end{cases} \quad (23)$$

This modified window has a half-Hamming window, which in continuous time starts at $t = -1/2$ and ends at $t = N_L - 1/2$. The cosine rolloff starts at $t = N_L - 1/2$ and ends at $t = N_L + N_R - 1/2$. The cosine rolloff is on a pedestal of height β .

Unfortunately, these modifications do not significantly improve the frequency response.

6.2 Optimization of the Asymmetrical Window

We attempted an iterative improvement scheme. Starting from the definition of the asymmetrical window, the coefficients were modified to improve the fraction of the energy in the frequency domain within $|\omega| \leq 4\pi/N$. The peak of the window was constrained to be at N_L , with a monotonic lefthand side and a monotonic righthand side. The optimization scheme took a step in one coefficient at a time within the monotonicity constraint. The order of adjustment was randomized. With optimization, the fraction of out-of-band energy reduced from 2.1% to 0.71%. The result is a window with a flattened top (see Fig. 14). The optimized window has its centre of mass more concentrated in the middle of the window than was the case for the original G.729 window. This shift of the centre of mass works against the goal of the emphasizing samples nearer to the end of the frame. In addition, the optimized window has disconcerting discontinuities.

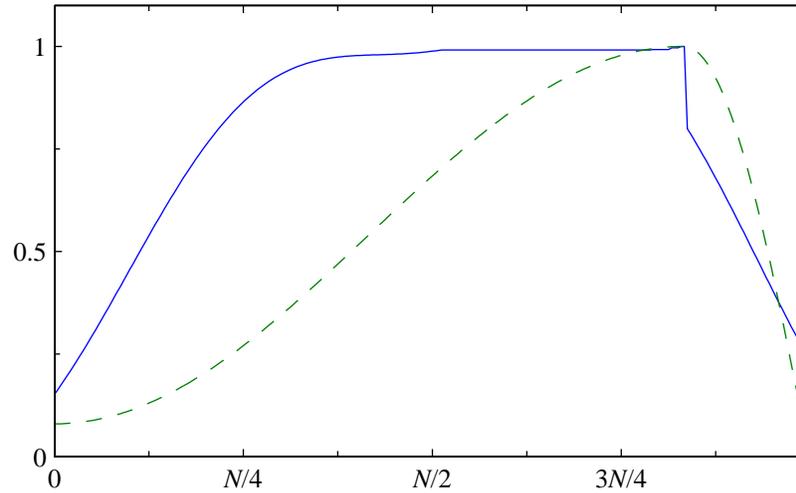


Figure 14 Asymmetrical window ($N = 240$), optimized to concentrate energy in $|\omega| < 4\pi/N$ (solid line). The dashed line is the G.729 asymmetrical window.

Recently, Chu [18] has optimized windows to maximize the average prediction error energy after LP analysis. The resulting windows take on an asymmetric form, but with a much narrower “main lobe” than the conventional asymmetric windows in, for instance, G.729.

7 Comparison of Window Properties

Table 1 shows the properties of the various windows considered above. The values given are those for a window length of $N = 240$. Values which scale with the window length are given in term of N . The table shows two DPSS windows. The first was designed to match the central portion of a Hamming window. The second was designed to have a spectral null at $|\omega| = 4\pi/N$, giving a main lobe width equal to that for a Hann or Hamming window. The properties given in the table are as follows.

1. The 6 dB bandwidth is the double-sided bandwidth measured at the half-amplitude point. For the modified Hann window, the half-amplitude point occurs exactly at $2\pi/N$, giving a 6 dB bandwidth of $4\pi/N$. The values in the table are normalized to the sampling rate. To get the value on the ω scale, multiply by 2π . To get the value in Hz, multiply by the sampling frequency in Hz. For 8 kHz sampling and $N = 240$, the 6 dB bandwidth of $2/N$ corresponds to 67 Hz.
2. The main lobe width (double-sided) is the distance between zero crossings surrounding the main lobe of the frequency response of the window. This is measured as a fraction of the

sampling rate. This main lobe width measure does not apply to the asymmetric window which does not exhibit zeros in the frequency response. For 8 kHz sampling and $N = 240$, a main lobe width of $4/N$ corresponds to 133 Hz.

3. The table gives the minimum attenuation for $|\omega| > 4\pi/N$. For the asymmetrical window, since there is no null near $4\pi/N$, the minimum attenuation occurs at $4\pi/N$. For the other windows, the minimum attenuation occurs for one of the sidelobes beyond $4\pi/N$.
4. The table column labelled "sidelobe energy" gives the fraction of energy for $|\omega| > 4\pi/N$.
5. The window energy is the sum of the squared values of the window (time-domain). All of the windows have been normalized such that the interpolated window value at the middle of the window is unity. The window energy can be used to normalize the window so that it results in an unbiased estimator for white noise inputs.
6. The pedestal height for each window is given in the table. For most of the windows, this was determined from the definition of the window. For the DPSS windows, there is no analytic expression for the window. For these windows, the pedestal height was estimated by extrapolating the window function out half a sample at either end. For the asymmetric window, the pedestal heights for the left portion of the window and for the right portion of the window are given separately.

Table 1 Properties of time windows ($N = 240$). The sidelobe attenuation corresponds to the largest sidelobe for $|\omega| > 4\pi/N$. The sidelobe energy is the fraction of the energy for $|\omega| > 4\pi/N$. Frequency values in the table are normalized by the sampling rate.

Window	6 dB Bandwidth	Main Lobe Width	Sidelobe Attenuation	Sidelobe Energy	Window Energy	Pedestal
Rectangular	$1.21/N$	$2/N$	17.8 dB	5.0 %	N	100 %
Hann	$2.01/N$	$4.01/N$	31.5 dB	0.051 %	$0.373N$	0 %
Mod. Hann	$2/N$	$4/N$	31.5 dB	0.051 %	$0.375N$	0 %
Hamming	$1.82/N$	$4.03/N$	42.7 dB	0.036 %	$0.396N$	8 %
Mod. Hamming	$1.82/N$	$4/N$	42.7 dB	0.037 %	$0.397N$	8 %
DPSS ($NW = 1.75$)	$1.84/N$	$3.89/N$	38.8 dB	0.017 %	$0.394N$	0.33 %
DPSS ($NW = 1.84$)	$1.87/N$	$4.01/N$	40.5 dB	0.012 %	$0.386N$	0.26 %
Asymmetric G.729	$1.70/N$	–	18.1 dB	2.1 %	$0.415N$	8, 0.99 %

7.1 Effect of the Window Pedestal

The usual argument for a tapered window is that as it slides across the signal, new samples are brought into play gradually. The Hamming window is a sinusoidal window sitting on a pedestal of relative height 0.08. The pedestal can cause substantial changes in the estimated LP parameters even when the window moves ahead by a single sample. For comparison, we will use a Hann window, i.e. a window without a pedestal.

For this experiment, the window is advanced in time one sample at a time. Figure 15 shows the effect on a segment of male speech. The plot shows the line spectral frequencies (LSF's) derived from a tenth order LP analysis [19]. The top plot shows the speech segment (the word "the" from the sentence "Kick the ball straight and follow through"). The second plot shows the energy under a Hann window as the window is advanced one sample at a time. The next plot shows the evolution of the line spectral frequencies using a Hann window (advanced one sample at a time). The bottom plot shows the LSF's when a Hamming window is used. The inset on the top waveform plot shows a Hamming window centred on 440 ms. This position corresponds to the roughness of the LSF tracks near 440 ms. If one superimposes the LSF tracks for the two types of windows, one observes that the "glitches" (Hamming window) in each cluster tend to occur on just one side of the smooth curve (Hann window).

In practice, the windows are advanced less often than every sample. The effect is to sub-sample the LSF tracks. It can be seen that if the sample point lies on one of the "outliers", the results for the two types of windows will be substantially different. Furthermore, the less smooth trajectories of the Hamming windowed speech parameters will be less amenable to differential coding. The effects demonstrated by this example are by no means rare — the spurious variations occur in many speech segments.

This experiment was also run with the other time windows considered earlier. For the asymmetric G.729 window, the spurious variations were as large or larger than for the Hamming window. For the DPSS window, the spurious variations were present but attenuated. This is consistent with the fact that the pedestal for the DPSS window is smaller than for the Hamming window or for the asymmetric window.

Al-Naimi *et al* [20] recognized the problems with the LSF tracks. They propose generating LP parameters at a high rate and then use an anti-aliasing filter to smooth out the variations before resampling at the frame desired rate. They show improved performance for a vector quantizer (using a moving average predictor) which is trained and operated with the smoothed LSF's. An alternative might be to use pitch-synchronous analysis with the aim of ensuring that the window location is more favourable aligned to large amplitude pitch pulses.

The results shown here indicate that the filtering proposed in [20] may be largely unnecessary.

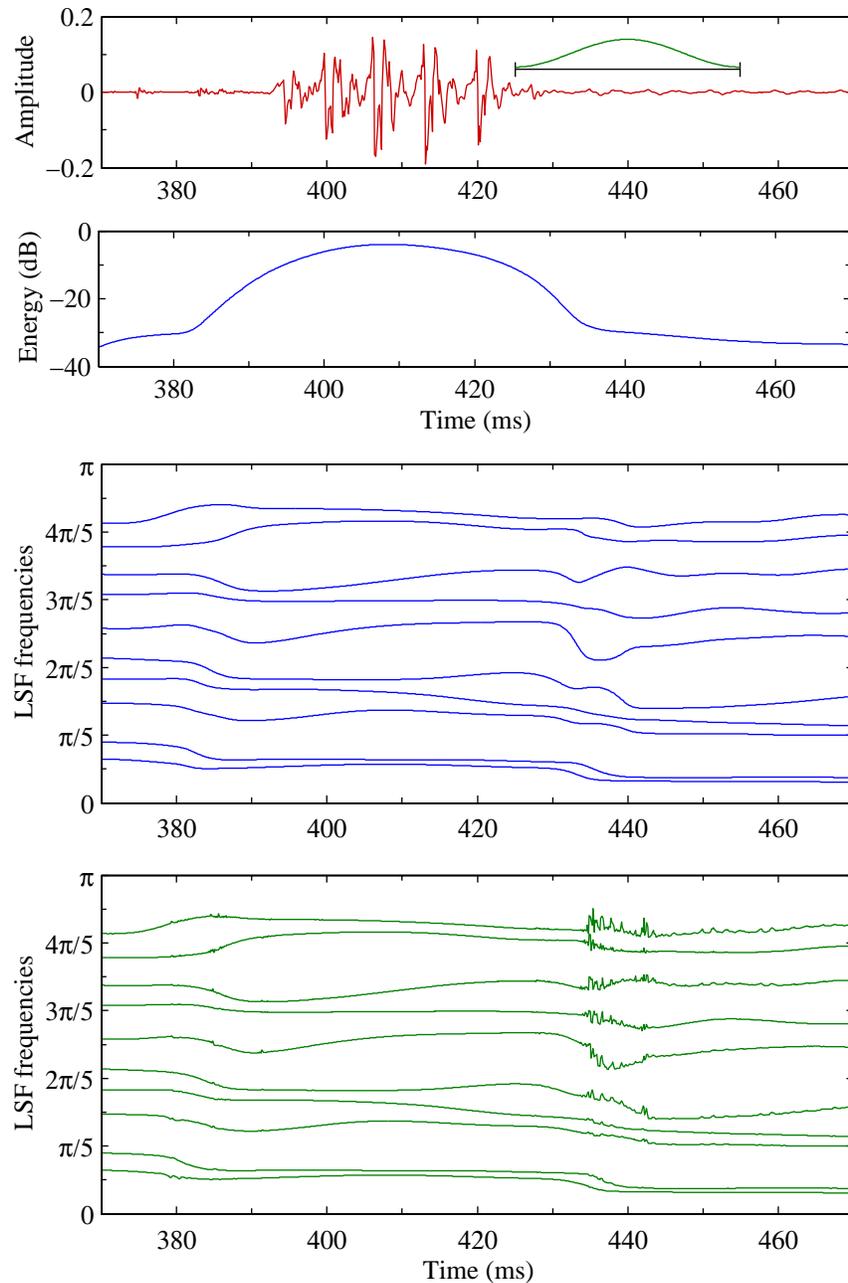


Figure 15 Effect of the window pedestal on LP parameters. The top plot shows the speech segment being analyzed. The inset shows a Hamming window centred on 440 ms. The second plot shows the energy under a Hann window. The next plot shows the evolution of the LSF parameters when a Hann window is used (window advanced one sample at a time). The bottom plot shows the evolution of the LSF parameters when a Hamming window is used (window advanced one sample at a time).

The simple expedient of using a window with no pedestal removes the spurious variations in the LSF tracks.

8 Summary

This report has examined the properties of windows used for the linear predictive analysis of speech. The suitability of a window for this purpose is a compromise between time-domain properties (length, shape, symmetry, and presence of discontinuities) and frequency-domain properties (main lobe width and shape, and sidelobe suppression). In the following, we give some suggestions. Instead of the conventional Hann and Hamming windows, use the modified versions for slightly improved properties. The DPSS window (bandwidth $3.5/N$) seems to be slightly preferable to a Hamming window. The main lobe is more compact and there is a better drop-off for the far out sidelobes, at the expense of a slightly poorer attenuation of the first sidelobe. The pedestal height is also smaller. Use the asymmetrical G.729 window only if the reduced look-ahead is of prime importance. The frequency resolution and sidelobe suppression for the asymmetrical window is very poor. For most speech applications, avoid windows with significant pedestals. This points to the use of the Hann window or as a compromise the DPSS window, which has a much smaller pedestal than the Hamming window. Other choices of the bandwidth for the DPSS window may be useful for speech processing. Generally, larger bandwidths will tradeoff a larger main lobe width against better sidelobe suppression and smaller pedestal height.

Appendix A Raised-Cosine Windows

Consider a raised-cosine window defined in continuous time,

$$w(t) = \begin{cases} \frac{1+\alpha}{2} + \frac{1-\alpha}{2} \cos\left(\frac{2\pi t}{W}\right), & |t| \leq \frac{W}{2}, \\ 0, & \text{elsewhere.} \end{cases} \quad (24)$$

This window is of length W and centred at the origin. Altering α allows the characteristics to change from a rectangular window ($\alpha = 1$), to a Hamming window ($\alpha = 0.08$), to the Hann window ($\alpha = 0$).

A discrete-time window can be created by sampling the continuous-time window. Different discrete-time windows result from different sampling patterns. The N uniformly spaced samples will be taken starting at $t = t_0$ and ending at $t = t_1$, where the end points are assumed to lie in the interval $[-W/2, W/2]$. The window is then

$$w[n] = \begin{cases} \frac{1+\alpha}{2} + \frac{1-\alpha}{2} \cos\left(\frac{2\pi n t_1 + (N-1-n)t_0}{W(N-1)}\right), & 0 \leq n \leq N-1, \\ 0, & \text{elsewhere.} \end{cases} \quad (25)$$

The discrete-time Fourier transform of the raised-cosine window is

$$W(\omega) = e^{-j\omega \frac{N-1}{2}} \left(\frac{1+\alpha}{2} \text{Dsinc}(\omega, N) + e^{j2\pi \frac{t_1+t_0}{2}} \frac{1-\alpha}{4} [\text{Dsinc}(\omega-\omega_o, N) + \text{Dsinc}(\omega+\omega_o, N)] \right), \quad (26)$$

where the function Dsinc is defined as⁷

$$\text{Dsinc}(\omega, N) = \frac{\sin(\omega N/2)}{\sin(\omega/2)} \quad (27)$$

and ω_o is $2\pi(t_1 - t_0)/(W(N-1))$. This frequency response of the window is the sum of three Dsinc functions. The function $\text{Dsinc}(\omega, N)$ is periodic in ω (period 2π for N odd, and period 4π for N even⁸), takes on the value N at $\omega = 0$ and has zeros at $\omega = 2\pi k/N$ for $k = 1, \dots, N-1$. The other Dsinc functions in the frequency response have the same zero crossing spacing, but are displaced in frequency. The zero crossings of the three Dsinc terms will coincide only if ω_o is an integer multiple of $2\pi/N$. We will create symmetric windows (with a linear phase frequency

⁷For N odd, the Dsinc function can be expressed in terms of the Dirichlet kernel, $\text{Dsinc}(\omega, N) = D_{(N-1)/2}(\omega)$, where $D_n(\omega)$ is the Dirichlet kernel.

⁸After adding a linear phase term, the overall function $e^{j\omega(N-1)/2} \text{Dsinc}(\omega, N)$ is periodic with period 2π for any integer N .

response) by setting $t_0 = -t_1$.⁹

A.1 Standard Raised-Cosine Windows

A standard formulation for a discrete-time raised-cosine window is obtained by choosing $t_1 = W/2$ and $t_0 = -t_1$, giving $\omega_o = 2\pi/(N-1)$,

$$w[n] = \begin{cases} \frac{1+\alpha}{2} - \frac{1-\alpha}{2} \cos\left(\frac{2\pi n}{N-1}\right), & 0 \leq n \leq N-1, \\ 0, & \text{elsewhere.} \end{cases} \quad (28)$$

For $\alpha = 0$, this is the window returned by, for instance, the `hann(N)` function call in MATLAB. The end points of the discrete-time window are zero. The `hann(N-1, 'periodic')` function call in MATLAB returns the same window without the last zero-valued sample. The `hanning(N-2)` function call in MATLAB returns the same window without either of the two zero-valued end-points.

For the standard Hann window, the zero-crossings of the 3 terms in the frequency response do not coincide. The overall frequency response has a first zero crossing which occurs between $2\pi/N$ and $2\pi/(N-1)$. The exact value must be found numerically for a given N .

The conventional Hamming window is given by Eq. (28) with $\alpha = 0.08$. This is the window returned by the `hamming(N)` function call in MATLAB. The Hamming window is a Hann window of height 0.92 sitting on a pedestal of height 0.08. The frequency response of the conventional Hamming window has the problems of mismatched zero crossing terms noted above.

A.2 Modified Raised-Cosine Windows

A modified raised-cosine window can be defined with sampling pattern given by $t_1 = W/2 - W/(2N)$ and $t_0 = -t_1$, resulting in $\omega_o = 2\pi/N$. The modified discrete-time raised-cosine windows can be written as

$$w[n] = \begin{cases} \frac{1+\alpha}{2} - \frac{1-\alpha}{2} \cos\left(\frac{\pi(2n+1)}{N}\right), & 0 \leq n \leq N-1, \\ 0, & \text{elsewhere.} \end{cases} \quad (29)$$

The modified Hann window can be calculated in MATLAB as `h=hann(2*N+1); h=h(2:2:end)`. The modified Hamming window is returned with `h=hamming(2*N+1); h=h(2:2:end)`.

⁹It is shown in [21] that one can create a symmetric window ($w[n] = w[N-1-n]$) from an unsymmetric one (by taking the symmetric part of the unsymmetric time response) and the result will have a uniformly better frequency response than the unsymmetric one.

The frequency response of the modified raised-cosine window is

$$W(\omega) = e^{-j\omega \frac{N-1}{2}} \left(\frac{1+\alpha}{2} \text{Dsinc}(\omega, N) + \frac{1-\alpha}{4} \left[\text{Dsinc}\left(\omega - \frac{2\pi}{N}, N\right) + \text{Dsinc}\left(\omega + \frac{2\pi}{N}, N\right) \right] \right), \quad (30)$$

For the modified windows, the zero crossings of all the Dsinc terms match. In fact, a Discrete Fourier Transform (DFT) of length N will have only three nonzero coefficients.

A.2.1 Frequency response of the modified Hann window

The frequency response (without the linear phase term) of the modified Hann window ($\alpha = 0$) is plotted in Fig. 16. The broken lines show the three terms that contribute to the sum (shown with a solid line). Notice that the term for the Dsinc function centred at zero (dashed line) and the terms for the other Dsinc functions (dash-dot lines) are out of phase and tend to cancel in the tails. This results in diminished sidelobes relative to a rectangular window (central Dsinc function alone).

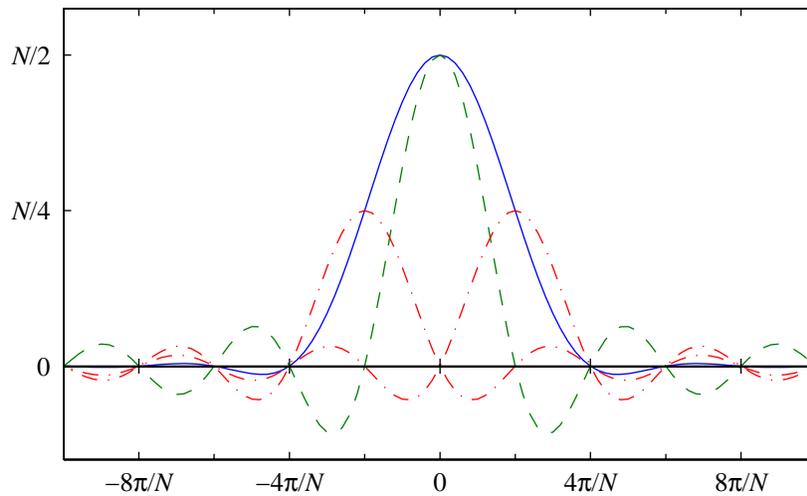


Figure 16 Frequency response of a modified Hann window ($N = 240$). The solid line is for the overall window. The broken lines show the three additive components.

The frequency response of the modified raised-cosine window can also be written in a product form (for N even),

$$W(\omega) = e^{-j\omega \frac{N-1}{2}} \text{Dsinc}(\omega, N/2) \times \frac{\cos\left(\frac{\omega N}{4}\right)}{\cos(\omega) - \cos\left(\frac{2\pi}{N}\right)} \left[\frac{1+\alpha}{2} \left(\cos(\omega) - \cos\left(\frac{2\pi}{N}\right) \right) + (1-\alpha) \sin^2\left(\frac{\omega}{2}\right) \cos\left(\frac{\pi}{N}\right) \right], \quad (31)$$

The frequency response (without the linear phase term) of the modified Hann window ($\alpha = 0$) is shown in Fig. 17. The two terms in the product are shown as broken lines, with the product being shown with a solid line. The first factor (dashed line) is the frequency response of a half-length rectangular window.¹⁰ This response has zero-crossings every $2\pi/(N/2)$. The second factor (dash-dot line) not only inserts extra zero crossings, but since it decreases in amplitude away from the main lobe, it reduces the sizes of the sidelobes in the overall frequency response.

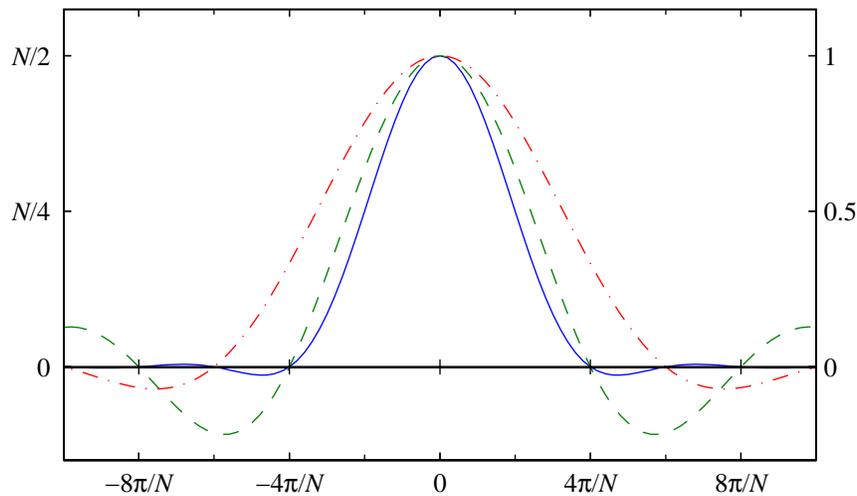


Figure 17 Frequency response of a modified Hann window ($N = 240$). The solid line is for the overall window. The broken lines show the two product factors. The right-hand scale applies to the second product factor (dash-dot line).

The modified Hann window is the convolution of the time responses corresponding to the two factors in the frequency response. The time responses of these factors are shown in Fig. 18. The first factor (dashed line) is a half-length rectangular window (assuming N is even). The second factor (dash-dot line) is a sine-like lobe.

A.2.2 Frequency response of the modified Hamming window

The Hamming window is formed by adding a pedestal to a Hann window. This results in better cancellation of the first few sidelobes. Comparing Fig. 19 with the corresponding figure for a Hann window (Fig. 16), it can be seen that increasing the relative amplitude of the central term (dash-dot line) reduces the first sidelobe of the frequency response. However further out in frequency, the cancellation will not be as good as for the Hann window.

¹⁰This construction does not work for N odd, since for that case the half-length windows length is not an integer.

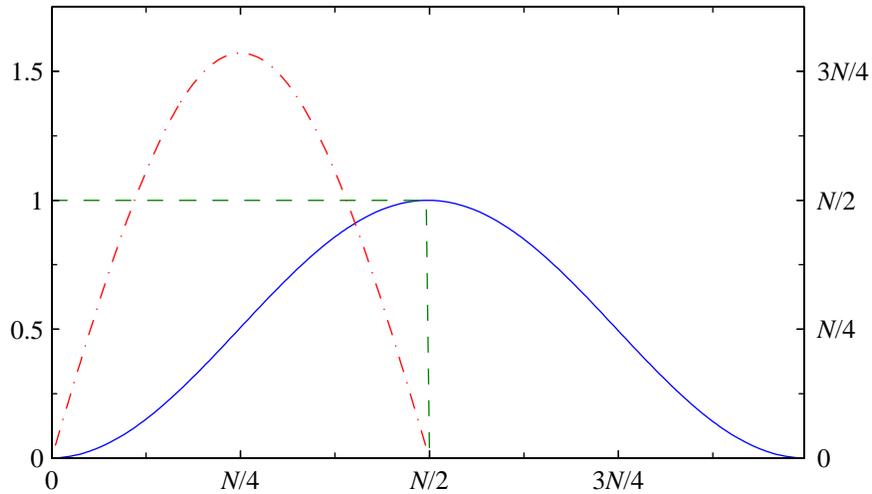


Figure 18 Modified Hann window ($N = 240$) as the convolution of two terms. The first term is a rectangular window of length $N/2$ (dashed line). The second term is a sine lobe (dash-dot line, using the right-hand scale).

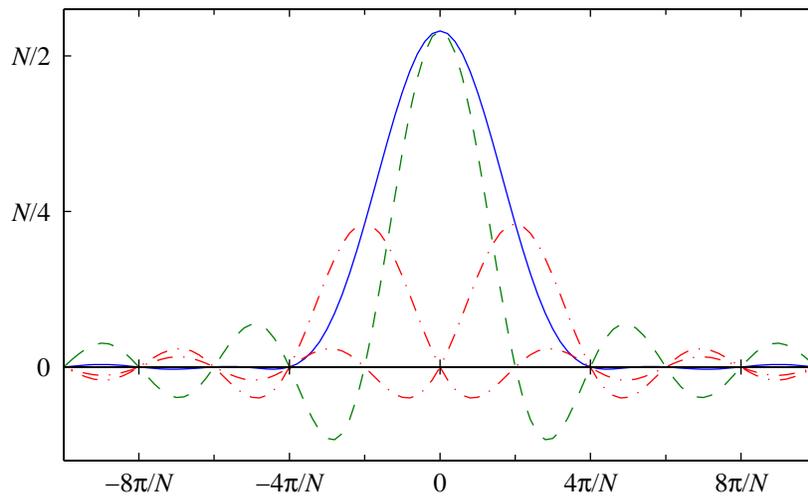


Figure 19 Frequency response of a modified Hamming window ($N = 240$). The solid line is for the overall window. The broken lines show the three additive components.

A.2.3 Comparison of modified and standard windows

At moderately large values of N , say around 240, the envelope of the frequency response of the standard and modified Hamming windows are largely indistinguishable. The details, of course, differ since the positions of the zero crossings differ.

For the Hann window, the situation is different. For the modified Hann window, the zeros have spacing $2\pi/N$, up to and including $\omega = \pi$. For the standard Hann window, the zeros have a slightly larger spacing, so that the last zero before π is closer to π than is the case for the modified Hann window. This close spacing of the zeros near π results in additional attenuation near $\omega = \pi$. The difference between the windows is illustrated in Fig. 20.

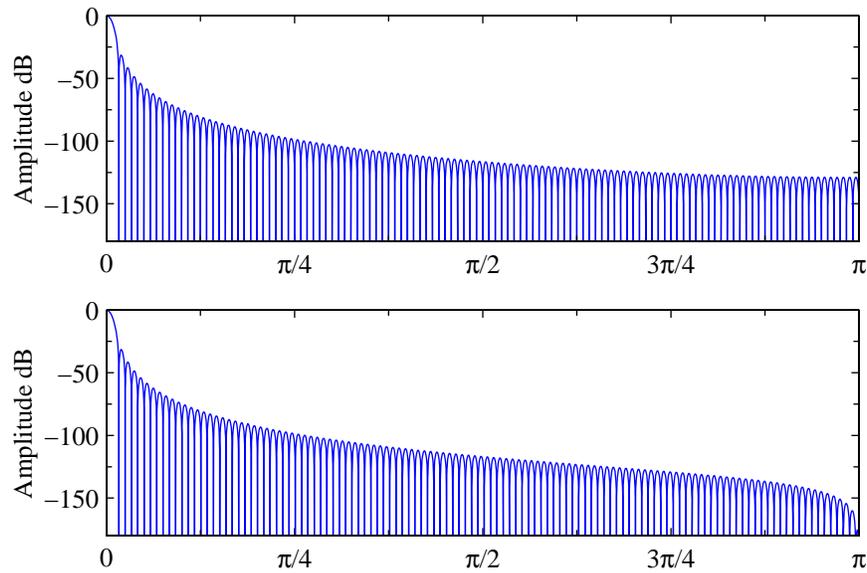


Figure 20 Normalized frequency response of the modified Hann window (top) and standard Hann window (bottom), $N = 240$.

Appendix B Windows Based on Orthogonal Polynomials

Streit [22] introduced the use of the ultraspherical family of polynomials (also known as the Gegenbauer polynomials) for window design. The idea was picked up more recently by Deczky [23] and Bergen and Antoniou [24] who provide design rules.

The ultraspherical polynomials, $C_m^{(\alpha)}(\cdot)$, satisfy the following recursion,

$$C_m^{(\alpha)}(x) = \frac{1}{m} \left[2(m-1+\alpha)x C_{m-1}^{(\alpha)}(x) - (m-2+2\alpha)C_{m-2}^{(\alpha)}(x) \right], \quad (32)$$

with initial conditions $C_0^{(\alpha)}(x) = 1$ and $C_1^{(\alpha)}(x) = 2\alpha x$. These polynomials are orthogonal over the interval $[-1, +1]$ with respect to the weighting function $(1-x^2)^{\alpha-\frac{1}{2}}$. $C_m^{(\alpha)}(x)$ is a polynomial of order m in x . From the recursion, it can be seen that $C_m^{(\alpha)}(x)$ contains only odd (resp. even) powers of x when m is odd (resp. even). As a consequence $C_m^{(\alpha)}(x)$ is an odd or even function of x depending on the value m .

B.1 Special Cases

For specific values of α , the ultraspherical polynomials become well known polynomials. For $\alpha = 0$, with an appropriate limiting operation, they become Chebyshev polynomials of the first kind [22],

$$T_m(x) = \frac{m}{2} \lim_{\alpha \rightarrow 0} C_m^{(\alpha)}(x) \quad (33)$$

$$T_m(x) = 2xT_{m-1}(x) - T_{m-2}(x), \quad T_0(x) = 1; T_1(x) = x. \quad (34)$$

Windows designed with $\alpha = 0$ are the Dolph-Chebyshev windows.

For $\alpha = \frac{1}{2}$, we get the Legendre polynomials,

$$P_m(x) = C_m^{(1/2)}(x) \quad (35)$$

$$P_m(x) = \frac{1}{m}(x(2m-1)P_{m-1}(x) - (m-1)P_{m-2}(x)), \quad P_0(x) = 1; P_1(x) = x. \quad (36)$$

For $\alpha = 1$, we get Chebyshev polynomials of the second kind.

$$U_m(x) = C_m^{(1)}(x) \quad (37)$$

$$U_m(x) = 2xU_{m-1}(x) - U_{m-2}(x), \quad U_0(x) = 1; U_1(x) = 2x. \quad (38)$$

Windows designed with $\alpha = 1$ are the same as the Saramäki windows [13].¹¹

B.2 Frequency Response

We can make a substitution, $x = x_0 \cos(\omega/2)$.¹² This maps the ω -interval $[0, 2\pi]$ to the x -interval $[-x_0, +x_0]$ (in reverse). This transformation has the effect of warping the x axis. The values of the maxima and minima of the polynomial within the interval are unchanged by this transformation.

Define

$$B_m^{(\alpha)}(\omega) = C_m^{(\alpha)}(x_0 \cos(\omega/2)). \quad (39)$$

We now have a periodic function in ω . For m even, $C_m^{(\alpha)}(x)$ is an even function and for m odd, $C_m^{(\alpha)}(x)$ is an odd function,

$$C_m^{(\alpha)}(x) = \begin{cases} C_m^{(\alpha)}(-x) & m \text{ even,} \\ -C_m^{(\alpha)}(-x) & m \text{ odd.} \end{cases} \quad (40)$$

In terms of $B(\omega)$,

$$B_m^{(\alpha)}(\omega) = \begin{cases} B_m^{(\alpha)}(\omega + 2\pi) & m \text{ even,} \\ -B_m^{(\alpha)}(\omega + 2\pi) & m \text{ odd.} \end{cases} \quad (41)$$

The function $B_m^{(\alpha)}(\omega)$ has period 2π if m is even and period 4π if m is odd.

For a polynomial of order $N - 1$ (N coefficients), we can form

$$W(\omega) = e^{-j\omega(N-1)/2} B_{N-1}^{(\alpha)}(\omega). \quad (42)$$

This function is now periodic with period 2π . This is the Discrete-Time Fourier Transform (DTFT) of a discrete-time window sequence. It is also a linear phase response. But more importantly, the frequency response is a finite length polynomial in $\exp(-j\omega)$,

$$W(\omega) = \sum_{n=0}^{N-1} w[n] e^{-j\omega n}. \quad (43)$$

The window coefficients can be found as the inverse Discrete Fourier Transform of samples of the frequency response.

¹¹For N odd, Saramäki calculates his windows as the sum of Chebyshev polynomials of the first kind. This sum is the Chebyshev polynomial of the second kind, viz. $U_M(x) = 1 + 2 \sum_{m=1}^{M/2} T_{2m}(x)$ [25, Eq. 22.12.2].

¹²For Chebyshev polynomials of the first kind, a different form involving polynomials of half the order is sometimes used. For M even, $T_M(x) = T_{M/2}(2x^2 - 1)$ [26, Eq. (21), p. 52], which for $x = x_0 \cos(\omega/2)$ becomes $T_M(x_0 \cos(\omega/2)) = T_{M/2}(x_0^2 \cos(\omega) + x_0^2 - 1)$.

B.2.1 Derivation for N odd

For N odd, the statement given above can be justified as follows. We will use the following facts.

- (a) For N odd, $C_{N-1}^{(\alpha)}(x)$ is a polynomial in even powers of x .
- (b) A polynomial in $\cos(\omega)$ can be expressed as an expansion in terms of $\cos(\omega k)$.

$$\begin{aligned}
W(\omega) &= e^{-j\omega(N-1)/2} B_{N-1}^{(\alpha)}(\omega) \\
&= e^{-j\omega(N-1)/2} C_{N-1}^{(\alpha)}(x_0 \cos(\omega/2)) \\
&= e^{-j\omega(N-1)/2} \sum_{k=0}^{(N-1)/2} a_{2k} (x_0 \cos(\omega/2))^{2k} \\
&= e^{-j\omega(N-1)/2} \sum_{k=0}^{(N-1)/2} a_{2k} x_0^{2k} \left(\frac{1 + \cos(\omega)}{2}\right)^k \\
&= e^{-j\omega(N-1)/2} \sum_{k=0}^{(N-1)/2} b_k (\cos(\omega))^k \\
&= e^{-j\omega(N-1)/2} \sum_{k=0}^{(N-1)/2} c_k \cos(\omega k) \\
&= e^{-j\omega(N-1)/2} \sum_{k=0}^{(N-1)/2} c_k \frac{e^{j\omega k} + e^{-j\omega k}}{2} \\
&= \sum_{n=0}^{N-1} w[n] e^{-j\omega n}.
\end{aligned} \tag{44}$$

The window has a Type 1 linear phase response (symmetric with an odd number of coefficients) [5],

B.2.2 Derivation for N even

For N even, $C_{N-1}^{(\alpha)}(x)$ is a polynomial in odd powers of x .

$$\begin{aligned}
W(\omega) &= e^{-j\omega(N-1)/2} B_{N-1}^{(\alpha)}(\omega) \\
&= e^{-j\omega(N-1)/2} C_{N-1}^{(\alpha)}(x_0 \cos(\omega/2)) \\
&= e^{-j\omega(N-1)/2} \sum_{k=0}^{N/2-1} a_{2k+1} (x_0 \cos(\omega/2))^{2k+1} \\
&= e^{-j\omega(N-1)/2} \cos(\omega/2) \sum_{k=0}^{N/2-1} a_{2k+1} x_0^{2k+1} \left(\frac{1 + \cos(\omega)}{2}\right)^k
\end{aligned} \tag{45}$$

$$\begin{aligned}
&= e^{-j\omega(N-1)/2} \cos(\omega/2) \sum_{k=0}^{N/2-1} b_k (\cos(\omega))^k \\
&= e^{-j\omega(N-1)/2} \cos(\omega/2) \sum_{k=0}^{N/2-1} c_k \cos(\omega k) \\
&= e^{-j\omega(N-1)/2} \frac{e^{j\omega/2} + e^{-j\omega/2}}{2} \sum_{k=0}^{N/2-1} c_k \frac{e^{j\omega k} + e^{-j\omega k}}{2} \\
&= \sum_{n=0}^{N-1} w[n] e^{-j\omega n}.
\end{aligned}$$

The window in this case has a Type 2 linear phase response (symmetric with an even number of coefficients) [5]. The response has a null at $\omega = \pi$ as indicated by the $\cos(\omega/2)$ term that comes out of the summation in the intermediate results above.

B.3 Design of Ultraspherical Windows

The design of an ultraspherical window with N coefficients to give a first sidelobe attenuation of R proceeds as follows. The polynomial order is $M = N - 1$. The ultraspherical polynomial can be implemented directly from the recursive relation of Eq. (32).

We will need to estimate the height of the first sidelobe. This is found by examining the appropriate interval between zero crossings. The zero crossings can be numbered with k , where $k = 1$ gives the largest largest zero crossing (just below $x = 1$), $k = 2$, gives the the second largest zero crossing, and so on. An estimate of the position of the k th zero crossing is given by

$$\tilde{x}_k = \frac{x_k^{(0)} x_k^{(1)}}{\alpha x_k^{(1)} + (1 - \alpha) x_k^{(0)}}, \quad (46)$$

where $x_k^{(0)}$ is the k th zero crossing of the Chebyshev polynomial of the first kind, and $x_k^{(1)}$ is the k th zero crossing of the Chebyshev polynomial of the second kind. The zero crossings of the Chebyshev polynomials are given by

$$x_k^{(0)} = \cos\left(\pi \frac{2k-1}{2M}\right), \quad x_k^{(1)} = \cos\left(\pi \frac{k}{M+1}\right), \quad 1 \leq k \leq M. \quad (47)$$

The estimate above is based on a linear combination of the inverse of the known roots. The estimate in Eq. (46) can be used both for α inside of the interval $[0, 1]$ and for reasonable values of α outside of that interval.

To find the sidelobe amplitude, first estimate the position of the second to last zero crossing; call it \tilde{x}_2 . Search for the largest sidelobe amplitude in the interval $[\tilde{x}_2, 1]$. Let the largest sidelobe

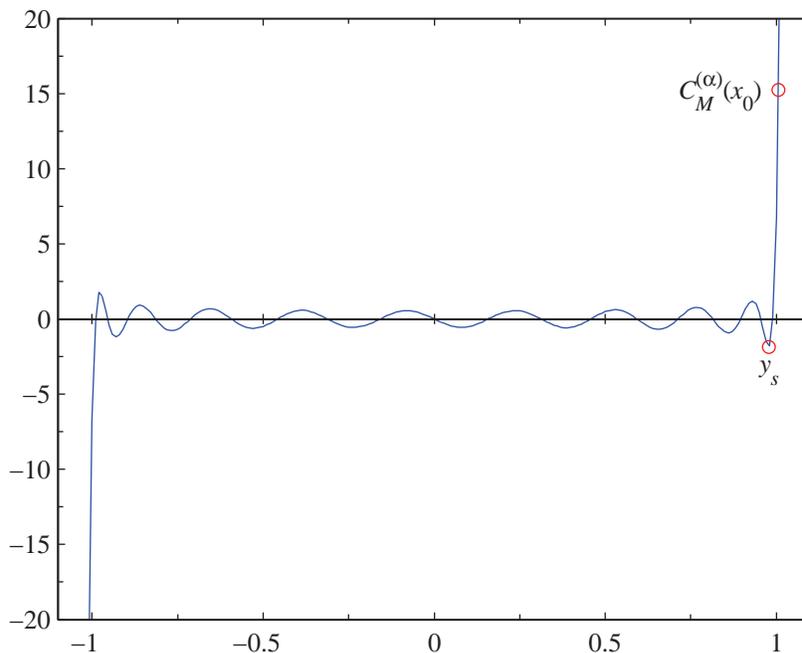


Figure 21 A plot of $C_M^{(\alpha)}(x)$ for $M = 19$ and $\alpha = 0.8$ against x . The point x_0 gives a main lobe to sidelobe ratio of 10.

amplitude be denoted as y_s . Given a desired sidelobe attenuation R , we then search for the x_0 which sets $C_M^{(\alpha)}(x_0) = R |y_s|$. The search interval is $x > 1$. The plot in Fig. 21 shows the ultraspherical polynomial for $M = 19$ and $\alpha = 0.8$. The positions of the peak of the first sidelobe y_s , and $C_M^{(\alpha)}(x_0)$ are marked with circles on this plot for $R = 10$.

The frequency response $B_M^{(\alpha)}(\omega)$ is obtained from $C_M^{(\alpha)}(x)$ with the substitution $x = x_0 \cos(\omega/2)$, see Eq. (39). This frequency response for our example is plotted in Fig. 22. The final step in the determination of the discrete-time window frequency response is to apply a time shift to the response to get $W(\omega)$ as given in Eq. (42). The inverse DFT of $W(2\pi k/N)$ gives the N sample time window. Finally, scaling is applied so that the maximum value of the samples is unity.

The MATLAB routines to calculate ultraspherical windows appear in Appendix C.

B.4 Parameters of the Ultraspherical Windows

The parameter α changes the type of window and within each type of window; the parameter x_0 trades off main lobe width against sidelobe attenuation. The parameter α can be used to choose the rate of decrease (or increase) in sidelobe attenuation from sidelobe to sidelobe. For $\alpha = 0$, all sidelobes have the same height. For $\alpha > 0$, the sidelobe heights decrease with frequency. For $\alpha < 0$, the sidelobe heights increase with frequency. Note that for $\alpha < 0$, not all of the window

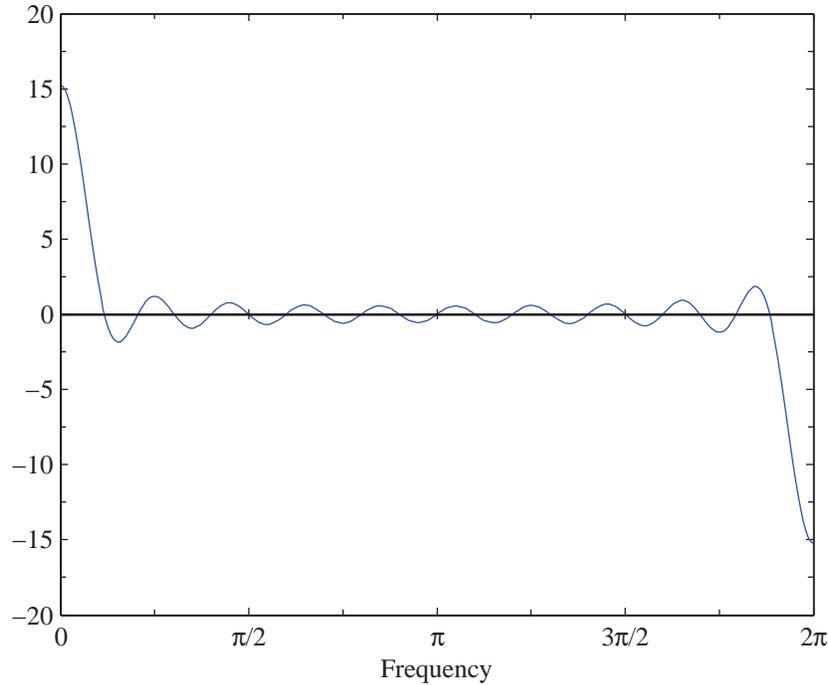


Figure 22 A plot of $B_M^{(\alpha)}(\omega)$ for $M = 19$, $\alpha = 0.8$ and a main lobe to sidelobe ratio of 10.

coefficients are necessarily of the same sign. Examples of windows designed for different values of α appear in [22, 23, 24].

In the above, we have shown how to calculate the spectrum of the window and suggested that the discrete-time window can be obtained using the inverse DFT. A direct formula for the discrete-time window is also available [22, 24].

B.5 Continuous-Time Windows

In [22], Streit provides the continuous-time versions of the ultraspherical windows. These are obtained by letting N go to infinity. The resulting continuous-time windows (normalized to unit height) are

$$w(t) = \left(\sqrt{1-t^2}\right)^{\alpha-1} \frac{I_{\alpha-1}(\beta\sqrt{1-t^2})}{I_{\alpha-1}(\beta)}, \quad -1 \leq t \leq 1. \quad (48)$$

For $\alpha = 1$, this is the continuous-time version of the Kaiser-Bessel window. The Kaiser-Bessel windows are the continuous-time versions of the discrete-time Saramäki windows based on the Chebyshev polynomials of the second kind.

In the case of $\alpha = 0$, corresponding to the continuous-time version of the Dolph-Chebyshev

window, a special form ensues,¹³

$$w(t) = \begin{cases} \frac{I_1(\beta\sqrt{1-t^2})}{I_1(\beta)\sqrt{1-t^2}} + \frac{1}{\beta I_1(\beta)}\delta(1-|t|), & -1 \leq t \leq 1, \\ 0, & \text{otherwise.} \end{cases} \quad (49)$$

The parameter β can be determined from the sidelobe attenuation R ,

$$\beta = \cosh^{-1}(R). \quad (50)$$

Note the mass functions at the ends of the continuous time window. The values for these masses are given by van der Maas [27] and Taylor [28].

B.6 Additional Notes

The Saramäki window becomes a rectangular window for $R = 4.56$ (13.2 dB). As R is reduced below this value, the window starts to thin in the middle to eventually become two pulses at the end points of the window. For $R < 4.56$, the main lobe is narrower than for the rectangular window. The continuous-time Kaiser-Bessel window becomes rectangular for $\beta = 0$. There is no setting of the Kaiser-Bessel window corresponding to the Saramäki windows with $R < 4.56$.

The continuous-time version of the Dolph-Chebyshev window has a smooth envelope plus impulses of area $1/(\beta I_1(\beta))$ at the ends. The area of the impulses is a decreasing function of R for $R \geq 1$. We can estimate the contribution of the impulses to the end points of a discrete-time window as

$$w_{ex} = \frac{N}{2} \frac{1}{\beta I_1(\beta)}. \quad (51)$$

From this estimate, if R is 20 dB, the end-points are huge; for R equal to 40 dB, the end points are large except for very short windows. For larger attenuations (60 or 80 dB), the end points are more reasonable even for relatively large values of N .

¹³In Streit [22], the continuous-time window is expressed as the inverse Fourier transform of the expression in Eq. (48). For $\alpha = 0$, the inverse Fourier transform gives probability masses in the result. For $\alpha < 0$, the inverse transform diverges.

Appendix C MATLAB Routines

This appendix gives MATLAB routines for producing some of the windows discussed in this report.

C.1 Modified Versions of Standard Windows

MATLAB provides standard windows which are sampled versions of continuous-time windows. The modified sampling pattern can be obtained by increasing the number of samples and then subsampling to get samples at the desired points. Some of these are shown below.

```
function h = ModHann(N)
% Modified sampling pattern applied to a Hann window

h = hann(2*N+1);
h = h(2:2:end);

return

function h = ModHamming(N)
% Modified sampling pattern applied to a Hamming window

h = hamming(2*N+1);
h = h(2:2:end);

return

function h = ModKaiserBessel(N, beta)
% Modified sampling pattern applied to a Kaiser-Bessel window

h = kaiser(2*N+1, beta);
h = h(2:2:end);

return
```

C.2 Ultraspherical Windows

The following routines calculate the ultraspherical windows. The routines are in three files, some of which contain internal functions.

C.2.1 UltraSphericalWin.m

This is the main routine to calculate ultraspherical windows.

```
function h = UltraSphericalWin (N, RdB, alpha)
% Window design based on an Ultraspherical polynomial
% - First sidelobe RdB down from the peak of the main lobe
% - Window parameter alpha: alpha=0 gives the Dolph-Chebyshev
%   (based on a Chebyshev polynomial of the first kind) window;
%   alpha=1 gives the Saramaki (based on a Chebyshev polynomial
%   of the second kind) window

% $Id: UltraSphericalWin.m,v 1.1 2006/09/27 18:17:12 pkabal Exp $

% Sample the polynomial at  $x = x_0 \cos(2 \pi k/N)$ , where  $x_0$  is
% the location of the first sidelobe. Then  $x$  is in the interval
%  $[-x_0, x_0]$ 

x0 = UltraSphericalx0(RdB, N-1, alpha);

f = (0:N-1) / N;
w = 2 * pi * f;

x = x0 * cos(w / 2);
HR = UltraSphericalPoly(x, N-1, alpha);
H = exp(-j * w * (N-1)/2) .* HR;

h = real (ifft(H)); % Knock out any residual imaginary component

% Flip the sign for alpha < 0
if (alpha < 0)
    h = -h;
end

h = h' / max(h);

return
```

C.2.2 UltraSphericalx0.m

This routine calculates the parameter x_0 for a given desired sidelobe attenuation.

```
function x0 = UltraSphericalx0 (RdB, M, alpha)
% Find the point  $x_0$  on the Ultraspherical polynomial that is RdB above
% the level of the first sidelobe.

% $Id: UltraSphericalx0.m,v 1.1 2006/09/27 18:19:30 pkabal Exp $
```

```

% Look for the first sidelobe in the interval just below x = 1
% The search range is delimited by [xL, 1], where xL is an estimate
% of the zero location based on interpolating the zero location of
% Chebyshev polynomials of the first and second kind.
xU = 1;
xL = EstZero(2, M, alpha);

% Minimum in the interval [xL, xU]
[xMin, SL] = fminbnd(@USP, xL, xU, optimset('TolX',1e-14), M, alpha);
if (xMin == xL || xMin == xU || SL >= 0)
    error('UltraSphericalx0: Failed to find a minimum');
end

% Find the amplitude of the point x0
% Looking for C_M(x0) = R * |SL|.
R = 10^( RdB / 20);
RM = R * abs(SL);

% Look for the point on the polynomial equal to RM
x0 = fzero(@USP_R, 1, optimset('TolX', 1e-14), M, alpha, RM);

return

%-----
function xr = EstZero (k, M, alpha)

% Estimate of the k'th zero of Pm(x)
% k = 1 for largest zero, k = 2 for the next largest zero, ...

% Interpolate/extrapolate the exact (inverse) roots of the
% Chebyshev polynomials (first and second kind)
xL0 = cos(pi * (k - 1/2) / M);    % Exact roots for alpha = 0
xL1 = cos(pi * k / (M+1));       % Exact roots for alpha = 1
xr = (xL0 * xL1) / (alpha * xL0 + (1 - alpha) * xL1);

return

%-----
function v = USP (x, M, alpha)
% Ultraspherical polynomial, flipped in sign for alpha < 0

if (alpha < 0)
    v = -UltraSphericalPoly(x, M, alpha);
else
    v = UltraSphericalPoly(x, M, alpha);
end

%-----
function y = USP_R (x, M, alpha, RM)

```

```
y = USP(x, M, alpha) - RM;

return
```

C.2.3 UltraSphericalPoly.m

This routine calculates the ultraspherical polynomial for a given argument.

```
function y = UltraSphericalPoly (x, M, alpha)
% Evaluate an ultraspherical polynomial at x (can be a vector of values)
% Special form for alpha = 0: Chebyshev of the first kind scaled by 2/M.

% $Id: UltraSphericalPoly.m,v 1.1 2006/09/27 18:18:29 pkabal Exp $

if (alpha < 0 && fix(alpha) == alpha)
    error('UltraSphericalPoly: alpha cannot be a negative integer');
end

yp = 1;
if (alpha == 0)
    y = x;
else
    y = 2 * alpha * x;
end

if (M == 0)
    y = yp;
end

for (m = 2:M)
    ypp = yp;
    yp = y;
    if (alpha == 0)
        y = 2 * x .* yp - ypp;
    else
        y = (1 / m) * (2 * (m+alpha-1) * x .* yp - (m+2*alpha-2) * ypp);
    end
end

if (alpha == 0 && M > 0)
    y = (2 / M) * y;
end

return
```

References

- [1] P. Kabal, *Windows for Transform Processing*, Technical Report, Electrical & Computer Engineering, McGill University, Dec. 2005 (on-line at www-mmsp.ECE.McGill.CA/MMSP/Documents/).
- [2] N. C. Beaulieu and M. O. Damen, "Parametric Construction of Nyquist-I Pulses", *IEEE Trans. Commun.*, vol. 52, pp. 2134–2142, Dec. 2004.
- [3] P. Kabal, "Ill-conditioning and Bandwidth Expansion in Linear Prediction of Speech", *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (Hong Kong), pp. I-824–I-827, April 2003.
- [4] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform", *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, Jan. 1978.
- [5] S. K. Mitra, *Digital Signal Processing: A Computer-based Approach*, third ed., McGraw-Hill, 2005.
- [6] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, Wiley Interscience, 2002.
- [7] D. B. Percival and A. T. Walden, *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*, Cambridge University Press, 1993.
- [8] J. F. Kaiser and R. W. Schafer, "On the Use of the I_0 -sinh Window for Spectrum Analysis", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, no. 1, pp. 105–107, Feb. 1980.
- [9] A. Hanssen, "On Multiwindow Estimators for Correlation", *Proc. IEEE Workshop Statistical Signal and Array Processing* (Pocono Manor, PA), pp. 640–644, Aug. 2000.
- [10] M. L. McCloud, L. L. Scharf and C. T. Mullis, "Lag-Windowing and Multiple-Data-Windowing are Roughly Equivalent for Smooth Spectrum Estimation", *IEEE Trans. Signal Processing*, vol. 47, pp. 839–843, March 1999.
- [11] K. S. Riedel and A. Sidorenko, "Minimum Bias Multiple Taper Spectral Estimation", *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 188–195, Jan. 1995.
- [12] P. Kabal, *Audio File Programs and Routines*, AFsp software package (on-line at www-mmsp.ECE.McGill.CA/Documents/Software/).
- [13] T. Saramäki, "A Class of Window Functions with Nearly Minimum Sidelobe Energy for Designing FIR Filters", *Proc. Int. Symp. Circuits, Systems* (Portland, OR), pp. 359–362, May 1989.
- [14] C. L. Dolph, "A Current Distribution for Broadside Arrays Which Optimizes the Relationship Between Beam Width and Side-Lobe Level", *Proc. Inst. Radio Engineers*, vol. 34, no. 6, pp. 335–348, June 1946.
- [15] D. A. F. Florêncio, "Investigating the Use of Asymmetric Windows in CELP Coders", *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (Minneapolis, MN), pp. 427–430, April 1993.

- [16] ITU-T, Geneva, *Recommendation G.729, Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, Mar. 1996.
- [17] 3GPP2 Document C.S0030-0, *Selectable Mode Vocoder Service Option of Wideband Spread Spectrum Communication Systems*, Version 2.0, Dec. 2001.
- [18] W. C. Chu, "Gradient-Descent Based Window Optimization for Linear Prediction Analysis", *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (Hong Kong)*, pp. I-460–I-463, April 2003.
- [19] P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 34, no. 6, pp. 1419–1426, Dec. 1986.
- [20] K. Al-Naimi, S. Villette and A. Kondoz, "Improved LSF Estimation Through Anti-Aliasing Filtering," *Proc. IEEE Workshop on Speech Coding (Tsukuba City, Japan)*, pp. 2–4, Sept. 2002.
- [21] H. Rohling and J. Schuermann, "Discrete Time Window Functions with Arbitrarily Low Side-lobe Level", *Signal Processing*, vol. 5, no. 2, pp. 127–138, March 1983.
- [22] R. L. Streit, "A Two-Parameter Family of Weights for Nonrecursive Digital Filters and Antennas", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 32, no. 1, pp. 108–118, Feb. 1984.
- [23] A. G. Deczky, "Unispherical Windows", *IEEE Int. Symp. Circuits, Systems (Sydney, Australia)*, pp. II-85–II-88, May 2001.
- [24] S. W. A. Bergen and A. Antoniou, "Design of Ultraspherical Window Functions with Prescribed Spectral Characteristics", *Proc. J. Applied Signal Processing*, vol. 13, no. 13, pp. 2053–2065, Oct. 2004.
- [25] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions*, Knovel, 1972.
- [26] L. Fox and I. B. Parker, *Chebyshev Polynomials in Numerical Analysis*, Oxford University Press, 1968.
- [27] G. J. van der Maas, "A Simplified Calculation for Dolph-Tchebycheff Arrays", *J. Applied Physics*, vol. 25, no. 1, pp. 121–124, Jan. 1954.
- [28] T. T. Taylor, *Dolph Arrays of Many Elements*, Technical Memorandum, Hughes Aircraft R&D Lab, Aug. 1953 (available on-line at www.DTIC.mil, document AD0018054).