Window function

From Wikipedia, the free encyclopedia

In signal processing, a **window function** (also known as an **apodization function** or **tapering function**[1]) is a mathematical function that is zero-valued outside of some chosen interval. For instance, a function that is constant inside the interval and zero elsewhere is called a *rectangular window*, which describes the shape of its graphical representation. When another function or a signal (data) is multiplied by a window function, the product is also zero-valued outside the interval: all that is left is the part where they overlap; the "view through the window". Applications of window functions include spectral analysis, filter design, and beamforming.

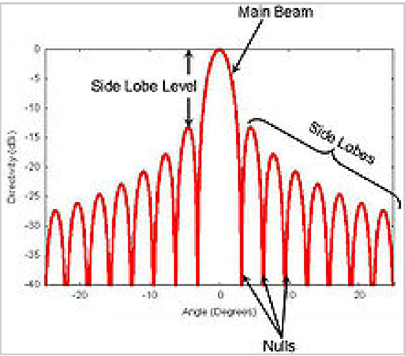
A more general definition of window functions does not require them to be identically zero outside an interval, as long as the product of the window multiplied by its argument is square integrable, that is, that the function goes sufficiently rapidly toward zero.[2]

In typical applications, the window functions used are non-negative smooth "bell-shaped" curves,[3] though rectangle and triangle functions and other functions are sometimes used.

Contents

Applications

Applications of window functions include spectral analysis and the design of finite impulse response filters.



Typical window function frequency response, showing main lobe, side lobe level, and side lobe fall-off.

Spectral analysis

The Fourier transform of the function $\cos(\omega t)$ is zero, except at frequency $\pm\omega$. However, many other functions and data (that is, *waveforms*) do not have convenient closed form transforms. Alternatively, one might be interested in their spectral content only during a certain time period.

In either case, the Fourier transform (or something similar) can be applied on one or more finite intervals of the waveform. In general, the transform is applied to the product of the waveform and a window function. Any window (including rectangular) affects the spectral estimate computed by this method.

Windowing

Windowing of a simple waveform, like $\cos(\omega t)$ causes its Fourier transform to have non-zero values (commonly called spectral leakage) at frequencies other than ω. It tends to be worst (highest) near ω and least at frequencies farthest from ω.

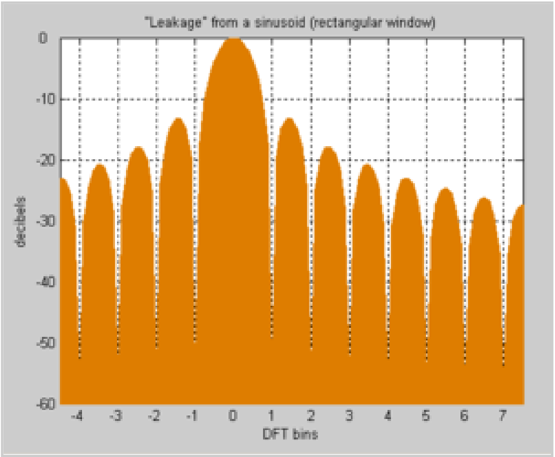


Figure 1: Zoom view of spectral leakage

If there are two sinusoids, with different frequencies, leakage can interfere with the ability to distinguish them spectrally. If their frequencies are dissimilar, then the leakage interferes when one sinusoid is much smaller in amplitude than the other. That is, its spectral component can be hidden by the leakage from the larger component. But when the frequencies are near each other, the leakage can be sufficient to interfere even when the sinusoids are equal strength; that is, they become *unresolvable*.

The rectangular window has excellent resolution characteristics for signals of comparable strength, but it is a poor choice for signals of disparate amplitudes. This characteristic is sometimes described as *low-dynamic-range*.

At the other extreme of dynamic range are the windows with the poorest resolution. These high-dynamic-range low-resolution windows are also poorest in terms of ***sensitivity***; this is, if the input waveform contains random noise close to the signal frequency, the response to noise, compared to the sinusoid, will be higher than with a higher-resolution window. In other words, the ability to find weak sinusoids amidst the noise is diminished by a high-dynamic-range window. High-dynamic-range windows are probably most often justified in *wideband applications*, where the spectrum being analyzed is expected to contain many different signals of various strengths.

In between the extremes are moderate windows, such as Hamming and Hann. They are commonly used in *narrowband applications*, such as the spectrum of a telephone channel. In summary, spectral analysis involves a tradeoff between resolving comparable strength signals with similar frequencies and resolving disparate strength signals with dissimilar frequencies. That tradeoff occurs when the window function is chosen.

Discrete-time signals

When the input waveform is time-sampled, instead of continuous, the analysis is usually done by applying a window function and then a discrete Fourier transform (DFT). But the DFT provides only a coarse sampling of the actual DTFT spectrum. **Figure 1** shows a portion of the DTFT for a rectangularly-windowed sinusoid. The actual frequency of the sinusoid is indicated as "0" on the horizontal axis. Everything else is leakage. The unit of frequency is "DFT bins"; that is, the integer values on the frequency axis correspond to the frequencies sampled by the DFT. So the figure depicts a case where the actual frequency of the sinusoid happens to coincide with a DFT sample, and the maximum value of the spectrum is accurately measured by that sample. When it misses the maximum value by some amount [up to 1/2 bin], the measurement error is referred to as ***scalloping loss*** (inspired by the shape of the peak). But the most interesting thing about this case is that all the other samples coincide with **nulls** in the true spectrum. (The nulls are actually zero-crossings, which cannot be shown on a logarithmic scale such as this.) So in this case, the DFT creates the **illusion** of no leakage. Despite the unlikely conditions of this example, it is a popular misconception that visible leakage is some sort of artifact of the DFT. But since any window function causes leakage, its apparent *absence* (in this contrived example) is actually the DFT artifact.

Noise bandwidth

The concepts of resolution and dynamic range tend to be somewhat subjective, depending on what the user is actually trying to do. But they also tend to be highly correlated with the total leakage, which is quantifiable. It is usually expressed as an equivalent bandwidth, B. Think of it as redistributing the DTFT into a rectangular shape with height equal to the spectral maximum and width B. The more leakage, the greater the bandwidth. It is sometimes called *noise equivalent bandwidth* or *equivalent noise bandwidth*, because it is proportional to the average power that will be registered by each DFT bin when the input signal contains a random noise component (or **is** just random noise). A graph of the power spectrum, averaged over time, typically reveals a flat *noise floor*, caused by this effect. The height of the noise floor is proportional to B. So two different window functions can produce different noise floors.

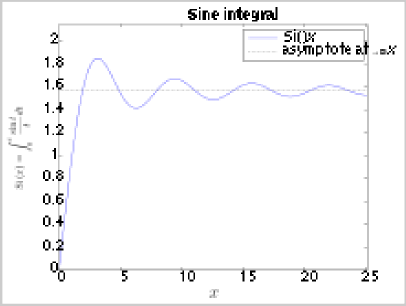
Processing gain

In signal processing, operations are chosen to improve some aspect of quality of a signal by exploiting the differences between the signal and the corrupting influences. When the signal is a sinusoid corrupted by additive random noise, spectral analysis distributes the signal and noise components differently, often making it easier to detect the signal's presence or measure certain characteristics, such as amplitude and frequency. Effectively, the signal to noise ratio (SNR) is improved by distributing the noise uniformly, while concentrating most of the sinusoid's energy around one frequency. *Processing gain* is a term often used to describe an SNR improvement. The processing gain of spectral analysis depends on the window function, both its noise bandwidth (B) and its potential scalloping loss. These effects partially offset, because windows with the least scalloping naturally have the most leakage.

For example, the worst possible scalloping loss from a Blackman–Harris window (below) is 0.83 dB, compared to 1.42 dB for a Hann window. But the noise bandwidth is larger by a factor of 2.01/1.5, which can be expressed in decibels as**:** $10\ \log_{10}(2.01\ /\ 1.5) = 1.27$. Therefore, even at maximum scalloping, the net processing gain of a Hann window exceeds that of a Blackman–Harris window by**:** 1.27 +0.83 -1.42 = 0.68 dB. And when we happen to incur no scalloping (due to a fortuitous signal frequency), the Hann window is 1.27 dB more *sensitive* than Blackman–Harris. In general (as mentioned earlier), this is a deterrent to using high-dynamic-range windows in low-dynamic-range applications.

Filter design

Main article: Filter design



Sine integral, showing part of frequency response for a truncated low-pass filter.

Windows are sometimes used in the design of digital filters, for example to convert an "ideal" impulse response of infinite duration, such as a sinc function, to a finite impulse response (FIR) filter design. Window choice considerations are related to those described above for spectral analysis, or can alternatively be viewed as a tradeoff between "ringing" and frequency-domain sharpness.[4]

Window examples

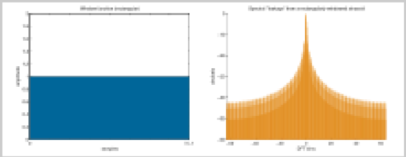
Terminology**:**

- N represents the width, in samples, of a discrete-time window function. Typically it is an integer power-of-2, such as $2^{10} = 1024$.
- n is an integer, with values **0 ≤ n ≤ N-1**. So these are the time-shifted forms of the windows: $w(n) = w_0\left(n - \frac{N-1}{2}\right)$, where $w_0(n)$ is maximum at **n=0**.
  - Some of these forms have an overall width of $N-1$, which makes them zero-valued at n=0 and n=$N-1$. That sacrifices two data samples for no apparent gain, if the DFT size is N. When that happens, an alternative approach is to replace $N-1$ with N in the formula.
- Each figure label includes the corresponding noise equivalent bandwidth metric (B), in units of *DFT bins*. As a guideline, windows are divided into two groups on the basis of B. One group comprises $1 \le B \le 1.8$, and the other group comprises $B \ge 1.98$. The Gauss and Kaiser windows are families that span both groups, though only one or two examples of each are shown.

High- and moderate-resolution windows

Rectangular window

$$w(n) = 1$$



Rectangular window; B=1.00

The rectangular window is sometimes known as a Dirichlet window. It is the simplest window, taking a chunk of the signal without any other modification at all, which leads to discontinuities at the endpoints (unless the signal happens to be an exact fit for the window length, as used in multitone testing, for instance). The first side-lobe is only 13 dB lower than the main lobe, with the rest falling off at about 6 dB per octave.[5][6]

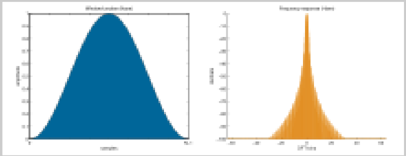
Hann window

Main article: Hann function

$$w(n) = 0.5\;\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right)$$ [note 1]

- Note that:

$$w_0(n) = 0.5\;\left(1 + \cos\left(\frac{2\pi n}{N-1}\right)\right)$$



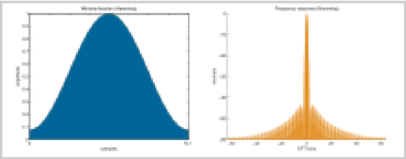
Hann window; B = 1.50

The ends of the cosine just touch zero, so the side-lobes roll off at about 18 dB per octave.[7]

The Hann and Hamming windows, both of which are in the family known as "raised cosine" windows, are respectively named after Julius von Hann and Richard Hamming. The term "Hanning window" is sometimes used to refer to the Hann window.

Hamming window

The "raised cosine" with these particular coefficients was proposed by Richard W. Hamming. The window is optimized to minimize the maximum (nearest) side lobe, giving it a height of about one-fifth that of the Hann window, a raised cosine with simpler coefficients.[8][9]



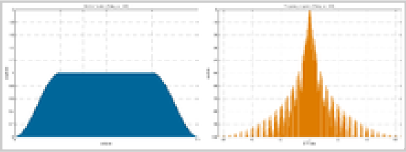
Hamming window; B=1.37

$$w(n) = 0.54 - 0.46\;\cos\left(\frac{2\pi n}{N-1}\right)$$ [note 1]

- Note that:

$$\begin{aligned} w_0(n) &\overset{\text{def}}{=} w(n+\tfrac{N-1}{2}) \\ &= 0.54 + 0.46\;\cos\left(\frac{2\pi n}{N-1}\right) \end{aligned}$$

Tukey window



Tukey window, α=0.5; B=1.22

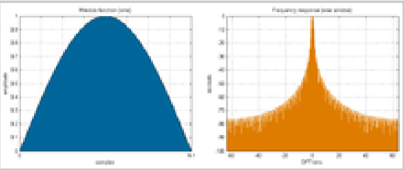
$$w_0(n) = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\pi\frac{|n| - \alpha\frac{N}{2}}{(1-\alpha)\frac{N}{2}}\right)\right] & \text{when } \frac{\alpha N}{2} \leqslant |n| \leqslant \frac{N}{2} \\ 1 & \text{when } 0 \leqslant |n| \leqslant \frac{\alpha N}{2} \end{cases}$$

The Tukey window[10], also known as the *tapered cosine window*[1] (http://www.mathworks.com/access/helpdesk/help/toolbox/signal/tukeywin.html) , can be regarded as a cosine lobe of width $\frac{\alpha N}{2}$ that is convolved with a rectangle window of width $\left(1-\frac{\alpha}{2}\right)N$. At α=1 it becomes rectangular, and at α=0 it becomes a Hann window.

Cosine window

$$w(n) = \cos\left(\frac{\pi n}{N-1} - \frac{\pi}{2}\right) = \sin\left(\frac{\pi n}{N-1}\right)$$ [note 1]

- also known as *sine window*
- *cosine window* describes the shape of $w_0(n)$



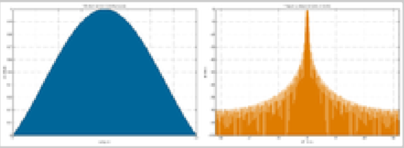
Cosine window; B=1.24

Lanczos window

$$w(n) = \operatorname{sinc}\left(\frac{2n}{N-1} - 1\right)$$

- used in Lanczos resampling
- for the Lanczos window, sinc(x) is defined as sin(πx)/(πx)
- also known as a *sinc window*, because**:**

$$w_0(n) = \operatorname{sinc}\left(\frac{2n}{N-1}\right)$$ is the main lobe of a normalized sinc function

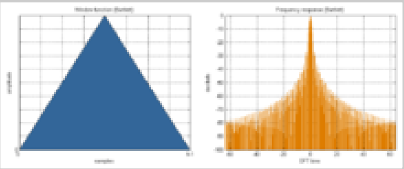


Sinc or Lanczos window; B=1.31

Triangular windows

Bartlett window with zero-valued end-points:

$$w(n) = \frac{2}{N-1} \cdot \left(\frac{N-1}{2} - \left|n - \frac{N-1}{2}\right|\right)$$

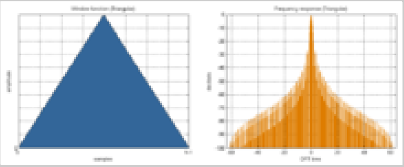


Bartlett window; B=1.33

With non-zero end-points:

$$w(n) = \frac{2}{N} \cdot \left(\frac{N}{2} - \left|n - \frac{N-1}{2}\right|\right)$$

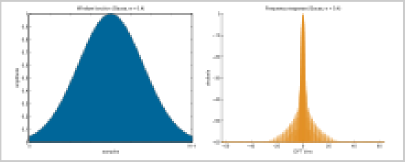
Can be seen as the convolution of two half-sized rectangular windows, giving it a main lobe width of twice the width of a regular rectangular window. The nearest lobe is -26 dB down from the main lobe.[11]



Triangular window; B=1.33

Gaussian windows

The frequency response of a Gaussian is also a Gaussian (it is an eigenfunction of the Fourier Transform). Since the Gaussian function extends to infinity, it must either be truncated at the ends of the window, or itself windowed with another zero-ended window.[12]



Gauss window, σ=0.4; B=1.45

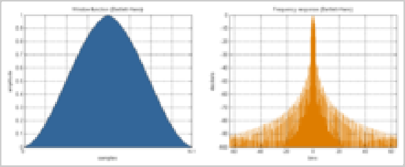
Since the log of a Gaussian produces a parabola, this can be used for *exact* quadratic interpolation in frequency estimation.[13][14][15]

$$w(n) = e^{-\frac{1}{2}\left(\frac{n-(N-1)/2}{\sigma(N-1)/2}\right)^2}$$
$$\sigma \leq\ 0.5$$

Bartlett–Hann window

$$w(n) = a_0 - a_1 \left|\frac{n}{N-1} - \frac{1}{2}\right| - a_2 \cos\left(\frac{2\pi n}{N-1}\right)$$

$$a_0 = 0.62; \quad a_1 = 0.48; \quad a_2 = 0.38$$



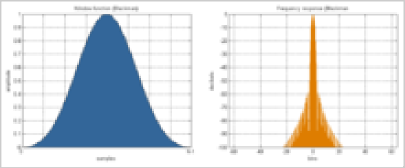
Bartlett-Hann window; B=1.46

Blackman windows

Blackman windows are defined as:[note 1]

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right)$$

$$a_0 = \frac{1-\alpha}{2}; \quad a_1 = \frac{1}{2}; \quad a_2 = \frac{\alpha}{2}$$

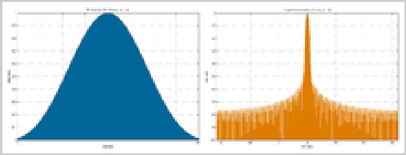


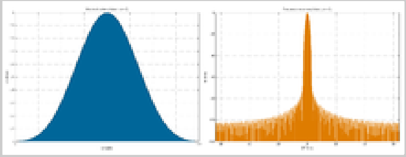
Blackman window; α = 0.16; B=1.73

By common convention, the unqualified term *Blackman window* refers to α=0.16.

Kaiser windows

Main article: Kaiser window


Kaiser window, α =2; B=1.5


Kaiser window, α =3; B=1.8

A simple approximation of the DPSS window using Bessel functions, discovered by Jim Kaiser.[16][17]

$$w(n)=\frac{I_0\left(\pi\alpha\sqrt{1-(\frac{2n}{N-1}-1)^2}\right)}{I_0(\pi\alpha)}$$

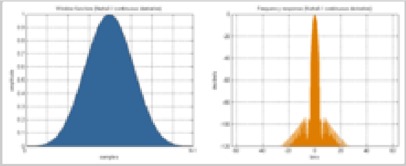
where I_0 is the zero-th order modified Bessel function of the first kind, and usually $\alpha = 3$.

- Note that:

$$w_0(n)=\frac{I_0\left(\pi\alpha\sqrt{1-(\frac{2n}{N-1})^2}\right)}{I_0(\pi\alpha)}$$

Low-resolution (high-dynamic-range) windows

Nuttall window, continuous first derivative

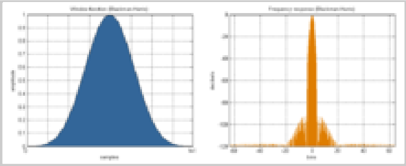

Nuttall window, continuous first derivative; B=2.02

$$w(n)=a_0-a_1\cos\left(\frac{2\pi n}{N-1}\right)+a_2\cos\left(\frac{4\pi n}{N-1}\right)-a_3\cos\left(\frac{6\pi n}{N-1}\right)$$ [note 1]

$$a_0=0.355768;\quad a_1=0.487396;\quad a_2=0.144232;\quad a_3=0.012604$$

Blackman–Harris window

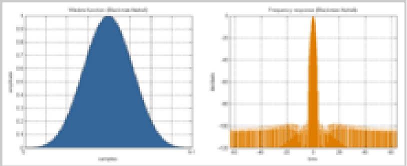
A generalization of the Hamming family, produced by adding more shifted sinc functions, meant to minimize side-lobe levels[18][19]


Blackman–Harris window; B=2.01

$$w(n)=a_0-a_1\cos\left(\frac{2\pi n}{N-1}\right)+a_2\cos\left(\frac{4\pi n}{N-1}\right)-a_3\cos\left(\frac{6\pi n}{N-1}\right)$$ [note 1]

$$a_0 = 0.35875; \quad a_1 = 0.48829; \quad a_2 = 0.14128; \quad a_3 = 0.01168$$

Blackman–Nuttall window

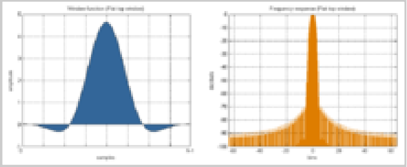


Blackman–Nuttall window; B=1.98

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right) - a_3 \cos\left(\frac{6\pi n}{N-1}\right)$$ [note 1]

$$a_0 = 0.3635819; \quad a_1 = 0.4891775; \quad a_2 = 0.1365995; \quad a_3 = 0.0106411$$

Flat top window



Flat top window; B=3.77

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right) - a_3 \cos\left(\frac{6\pi n}{N-1}\right) + a_4 \cos\left(\frac{8\pi n}{N-1}\right)$$

[note 1]

$$a_0 = 1; \quad a_1 = 1.93; \quad a_2 = 1.29; \quad a_3 = 0.388; \quad a_4 = 0.032$$

Other windows

Bessel window

Dolph-Chebyshev window

Minimizes the Chebyshev norm of the side-lobes for a given main lobe width.[20]

Hann-Poisson window

A Hann window multiplied by a Poisson window, which has no side-lobes, in the sense that the frequency response drops off forever away from the main lobe. It can thus be used in hill climbing algorithms like Newton's method.[21]

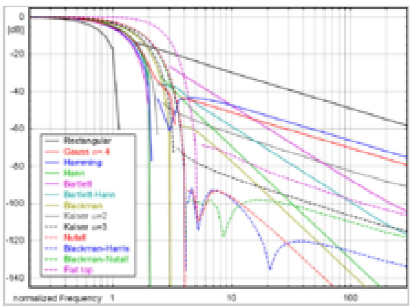
Exponential window

Rife-Vincent window

DPSS or Slepian window

The DPSS (digital prolate spheroidal sequence) or **Sepian** window is used to maximize the energy concentration in the main lobe.[22]

Comparison of windows



Stopband attenuation of different windows

When selecting an appropriate window function for an application, this comparison graph may be useful. The graph shows only the main lobe of the window's frequency response in detail. Beyond that only the envelope of the sidelobes is shown to reduce clutter. The frequency axis has units of FFT "bins" when the window of length N is applied to data and a transform of length N is computed. For instance, the value at frequency ½ "bin" is the response that would be measured in bins k and k+1 to a sinusoidal signal at frequency k+½. It is relative to the maximum possible response, which occurs when the signal frequency is an integer number of bins. The value at frequency ½ is referred to as the maximum *scalloping loss* of the window, which is one metric used to compare windows. The rectangular window is noticeably worse than the others in terms of that metric.

Other metrics that can be seen are the width of the main lobe and the peak level of the sidelobes, which respectively determine the ability to resolve comparable strength signals and disparate strength signals. The rectangular window (for instance) is the best choice for the former and the worst choice for the latter. What cannot be seen from the graphs is that the rectangular window has the best noise bandwidth, and despite its 3 dB potential scalloping loss, it is the best choice for detecting a sinusoid at low signal-to-noise ratios.

Overlapping windows

When the length of a data set to be transformed is larger than necessary to provide the desired frequency resolution, a common practice is to subdivide it into smaller sets and window them individually. To mitigate the "loss" at the edges of the window, the individual sets may overlap in time. See Welch method of power spectral analysis and the Modified discrete cosine transform.

See also

- Multitaper
- Apodization
- Welch method
- Short-time Fourier transform
- Window design method

See also: Window function (SQL)

Notes

1. ^ a b c d e f g h Windows of the form:

$$w(n) = \sum_{k=0}^{K} a_k \cos\left(\frac{2\pi kn}{N-1}\right)$$

have only 2K+1 non-zero DFT coefficients, which makes them good choices for applications that require windowing by convolution in the frequency-domain. In those applications, the DFT of the unwindowed data vector is needed for a different purpose than spectral analysis. (see Overlap-save method)

References

1. ^ Eric W. Weisstein (2003). *CRC Concise Encyclopedia of Mathematics* (http://books.google.com/books?id=aFDWuZZslUUC&pg=PA97&dq=apodization+function&lr=&as_brr=0&ei=27ycSPCdJZDwsgOdxIieBQ&sig=ACfU3U1EhHuuq88rRYF2W01Jj1o2Ab-6wA#PPA95,M1) . CRC Press. ISBN 1584883472. http://books.google.com/books?id=aFDWuZZslUUC&pg=PA97&dq=apodization+function&lr=&as_brr=0&ei=27ycSPCdJZDwsgOdxIieBQ&sig=ACfU3U1EhHuuq88rRYF2W01Jj1o2Ab-6wA#PPA95,M1.
2. ^ Carlo Cattani and Jeremiah Rushchitsky (2007). *Wavelet and Wave Analysis As Applied to Materials With Micro Or Nanostructure* (http://books.google.com/books?id=JuJKu_0KDycC&pg=PA53&dq=define+%22window+function%22+nonzero+interval&lr=&as_brr=3&ei=iyGbSKX_OJCKtAPXwaD2CQ&sig=ACfU3U1zZzq20w9GO1fbmNcvD3MlQFevsg#PPA53,M1) . World Scientific. ISBN 9812707840. http://books.google.com/books?id=JuJKu_0KDycC&pg=PA53&dq=define+%22window+function%22+nonzero+interval&lr=&as_brr=3&ei=iyGbSKX_OJCKtAPXwaD2CQ&sig=ACfU3U1zZzq20w9GO1fbmNcvD3MlQFevsg#PPA53,M1.
3. ^ Curtis Roads (2002). *Microsound*. MIT Press. ISBN 0262182157.
4. ^ Mastering Windows: Improving Reconstruction (http://www.cg.tuwien.ac.at/research/vis/vismed/Windows/MasteringWindows.pdf)
5. ^ https://ccrma.stanford.edu/~jos/sasp/Properties.html
6. ^ https://ccrma.stanford.edu/~jos/sasp/Rectangular_window_properties.html
7. ^ https://ccrma.stanford.edu/~jos/sasp/Hann_or_Hanning_or.html
8. ^ Loren D. Enochson and Robert K. Otnes (1968). *Programming and Analysis for Digital Time Series Data* (http://books.google.com/books?id=duBQAAAAMAAJ&q=%22hamming+window%22+date:0-1970&dq=%22hamming+window%22+date:0-1970&lr=&as_brr=0&ei=4LEASfHtJoXWsgOcz7WdDA&pgis=1) . U.S. Dept. of Defense, Shock and Vibration Info. Center. pp. 142. http://books.google.com/books?id=duBQAAAAMAAJ&q=%22hamming+window%22+date:0-1970&dq=%22hamming+window%22+date:0-1970&lr=&as_brr=0&ei=4LEASfHtJoXWsgOcz7WdDA&pgis=1.
9. ^ https://ccrma.stanford.edu/~jos/sasp/Hamming_Window.html
10. ^ Tukey, J.W. (1967), "An introduction to the calculations of numerical spectrum analysis", *Spectral Analysis of Time Series*: 25--46
11. ^ https://ccrma.stanford.edu/~jos/sasp/Properties_I_I_I.html
12. ^ https://ccrma.stanford.edu/~jos/sasp/Gaussian_Window_Transform.html
13. ^ https://ccrma.stanford.edu/~jos/sasp/Matlab_Gaussian_Window.html
14. ^ https://ccrma.stanford.edu/~jos/sasp/Quadratic_Interpolation_Spectral_Peaks.html
15. ^ https://ccrma.stanford.edu/~jos/sasp/Gaussian_Window_Transform_I.html
16. ^ https://ccrma.stanford.edu/~jos/sasp/Kaiser_Window.html
17. ^ https://ccrma.stanford.edu/~jos/sasp/Kaiser_DPSS_Windows_Compared.html
18. ^ https://ccrma.stanford.edu/~jos/sasp/Blackman_Harris_Window_Family.html
19. ^ https://ccrma.stanford.edu/~jos/sasp/Three_Term_Blackman_Harris_Window.html

20. ^ https://ccrma.stanford.edu/~jos/sasp/Dolph_Chebyshev_Window.html
21. ^ https://ccrma.stanford.edu/~jos/sasp/Definition_I_I_I_I.html
22. ^ https://ccrma.stanford.edu/~jos/sasp/Slepian_or_DPSS_Window.html

Other references

- Harris, Fredric j. (January 1978). ""On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform" (http://web.mit.edu/xiphmont/Public/windows.pdf) ". *Proceedings of the IEEE* **66** (1): 51–83. Article on FFT windows which introduced many of the key metrics used to compare windows.

- Nuttall, Albert H. (February 1981). "Some Windows with Very Good Sidelobe Behavior". *IEEE Transactions on Acoustics, Speech, and Signal Processing* **29** (1): 84–91. Extends Harris' paper, covering all the window functions known at the time, along with key metric comparisons.

- Oppenheim, Alan V.; Schafer, Ronald W.; Buck, John A. (1999). *Discrete-time signal processing*. Upper Saddle River, N.J.: Prentice Hall. pp. 468–471. ISBN 0-13-754920-2.

- Bergen, S.W.A.; A. Antoniou (2004). "Design of Ultraspherical Window Functions with Prescribed Spectral Characteristics". *EURASIP Journal on Applied Signal Processing* **2004** (13): 2053–2065. doi:10.1155/S1110865704403114 (http://dx.doi.org/10.1155%2FS1110865704403114) .

- Bergen, S.W.A.; A. Antoniou (2005). "Design of Nonrecursive Digital Filters Using the Ultraspherical Window Function". *EURASIP Journal on Applied Signal Processing* **2005** (12): 1910–1922. doi:10.1155/ASP.2005.1910 (http://dx.doi.org/10.1155%2FASP.2005.1910) .

- Park, Young-Seo, "System and method for generating a root raised cosine orthogonal frequency division multiplexing (RRC OFDM) modulation", US patent 7065150 (http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=US7065150) , published 2003, issued 2006

- LabView Help, Characteristics of Smoothing Filters, http://zone.ni.com/reference/en-XX/help/371361B-01/lvanlsconcepts/char_smoothing_windows/

- Evaluation of Various Window Function using Multi-Instrument, http://www.multi-instrument.com/doc/D1003/Evaluation_of_Various_Window_Functions_using_Multi-Instrument_D1003.pdf

Retrieved from "http://en.wikipedia.org/wiki/Window_function"
Categories: Statistical theory | Data analysis | Time series analysis | Fourier analysis | Signal processing | Digital signal processing

- Disclaimers