

## Contents

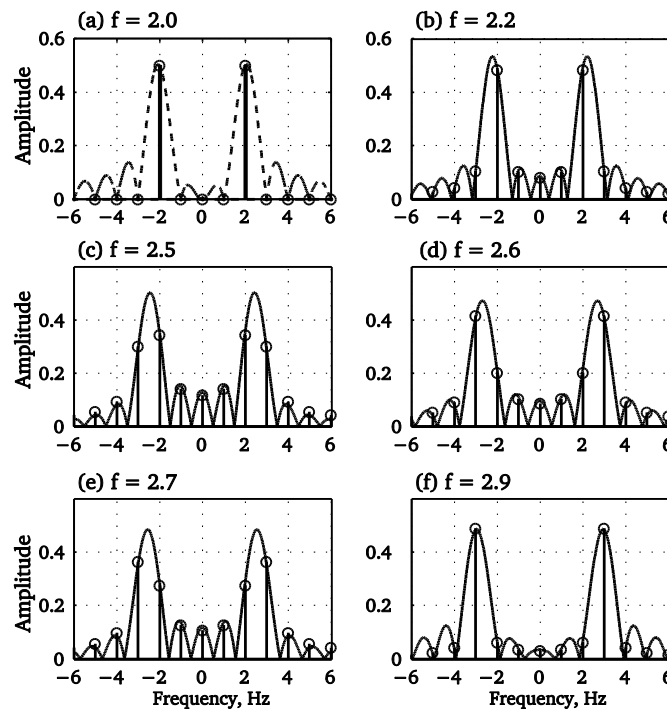
<b>Signal Windowing .....</b>	<b>2</b>
<b>Smearing and leakage due to truncation .....</b>	<b>3</b>
<b>The rectangular window is everywhere .....</b>	<b>5</b>
<b>Window Quality metrics .....</b>	<b>8</b>
<b>Filtering with Windows .....</b>	<b>9</b>
<b>Spectral estimation .....</b>	<b>10</b>
<b>Comparing windows .....</b>	<b>11</b>
<b>The flexible Kaiser window .....</b>	<b>13</b>

# 7

## Signal

### Windowing

When a signal frequency falls exactly on a DFT bin, it will present itself as a single well-defined impulse in the frequency domain. But when it falls in between bins, we do not see the peak of the signal but get a smeared version of the signal across several bins. Since we rarely have a signal that consists of just one single frequency aligned perfectly with the sampling frequency, smearing is nearly always present whenever we do a DFT of a practical signal with many frequency components. Since a real signal has very many frequencies, it is near-impossible to select a DFT size (especially when FFT limits it to powers of 2!) such that the period in samples of every single frequency in the signal divide evenly into it. Both of these effects that of the signal frequencies not coinciding exactly with DFT harmonics, and the discontinuities at the ends of the signals are caused by the same thing, which is that we do not have an integer number of periods of each component frequency in the DFT.



*Figure 7.1 Comparing DTFT with DFT, we see that if DFT bins do not align with the signal frequency, we get a smeared spectrums.*

### Smearing and leakage due to truncation

In Fig. 7.1, we see the DFT of sinusoids of frequencies: 2.0, 2.2, 2.5, 2.6, 2.7 and 2.9 Hz. The sampling frequency for all these cases is  $F_s = 16$  Hz, which is well above the Nyquist rate. We obtain 16 samples for each sinusoid and then do a 16 point DFT on the collected samples. In Fig. 7.1(a) we see that when  $f_c = 2$ , the signal falls exactly on a bin, (Bin resolution =  $F_s/N = 1$  Hz) hence the DFT matches the ideal expected. In (b), we see DFT for  $F_c = 2.2$ . The resolution is still 1 Hz, and now the peak no longer falls on a bin (it is at 2.2 Hz) and hence the nearest bin does not measure the full amplitude. The spectrum looks smeared, particularly towards the bottom. At  $f_c = 2.5$ , exactly half-way to the next bin, the smearing is the worst. Now the energy is split between bins that straddle the actual frequency. The spectrum then begins again to improve as the signal frequency again approaches a bin. This is leakage that is occurring because DFT bins do not align with the maximum signal energy.

The cause of this smearing is that the 16 samples do not constitute an integer number of periods for the cases where the frequency is not equal to 2.0. A frequency of 2 Hz when sampled at 16 samples per second, results in 8 samples per period. So 16 samples is exactly 2 cycles. But a frequency of 2.5 Hz sampled at 16 samples per second means 6.4 samples in one cycle or 2.5 cycles in 16 samples. Now we do not have an integer number of cycles in  $N=16$  samples hence we get smearing. Same happens for all the other frequencies until we get to 4 Hz, when once again we get an integer ratio and the maximum energy aligns with the bins. So even if some of the components in a signal fall on DFT bins, but since not all do, we will get leakage. The leakage can be large enough to cause us to make erroneous conclusions from these smeared spectrums.

We can mitigate but never completely remove the leakage by multiplying the signal with a windowing function prior to doing the DFT. This was noted in Chapter 6, Fig. 6.24. These windowing functions attenuate the ends of the signal so that the spurious frequencies that result in the DFT from discontinuities are reduced. Windowing can improve the smearing effects. But windowing seems to offer this advantage at a cost. The improvement in the spectrum is not uniform. Windowing a signal forces us to choose between competing goals of improved resolution in near and far parts of a signal. We don't get both.

The resolution of a DFT is defined as the minimum frequency that can be discriminated by it. In a general sense, this is equal to the sampling frequency divided by the size of DFT. Since sampling frequency is in Hz, this number is in Hertz as well.

$$\text{Resolution} = \frac{F_s}{N} \text{ Hz} \quad (7.1)$$

$F_s$  is the sampling frequency (the inverse of sample time,  $T_s$ ) and  $N$  is the length of the DFT. A small resolution is better than a large resolution. Improving the resolution means decreasing this number. If you increase the sampling frequency, the numerator increases the range of frequencies that can be resolved but this makes the resolution worse. So increasing the sampling

frequency by itself is not helpful for improving the resolution. Increasing N, the DFT size improves the frequency resolution for a given sampling frequency. Note that we cannot increase both at the same time by the same rate, as that will not change the resolution, only the range.

Generally we select  $F_s$  to be 4 to 8 times the highest signal frequency and DFT size N is selected based on the desired frequency resolution as well as hardware and software limitations.

The DFT is performed on a finite number of samples, usually a number that is an integer power of 2. The size of DFT always implies a truncation of the signal in time. This truncation can be described mathematically as the application of an observation period, called a **window**, to the signal. In terms of samples, we can describe the truncation window as an all-ones function of a finite length.

$$w_N(k) = \begin{cases} 1 & 0 \leq k < N \\ 0 & \text{otherwise} \end{cases} \quad (7.2)$$

When we select N samples for a DFT, it is implied that the signal is multiplied by this window up to N samples and then boom, rest are zeroed out. So N forms the length of the truncation window as well as the length of the signal. This is our default window *always* present whenever we talk about a finite number of samples of any signal. All finite signals are hence windowed signals as in Eq. (7.3). They are really an infinitely long signal multiplied by a window function. The default window in case of straight forward truncation is the *rectangular* window.

$$x_N[n] = w_N[n] x[n] \quad (7.3)$$

The effect of this multiplication in time domain is that the DFT of such a truncated (windowed) signal is the convolution of the true DFT of the signal with the DFT of the window function.

$$X_N[k] = W_N[k] * X[k] \quad (7.4)$$

The rectangular window is called an **All-ones window** for its length. The N point DFT of this all-ones signal is the Diric function shown here. Note this is the DFT of just the rectangular window function.

$$W(\omega) = e^{-j\omega\left(\frac{N-1}{2}\right)} \frac{\sin N \frac{\omega}{2}}{\sin \frac{\omega}{2}} \quad (7.5)$$

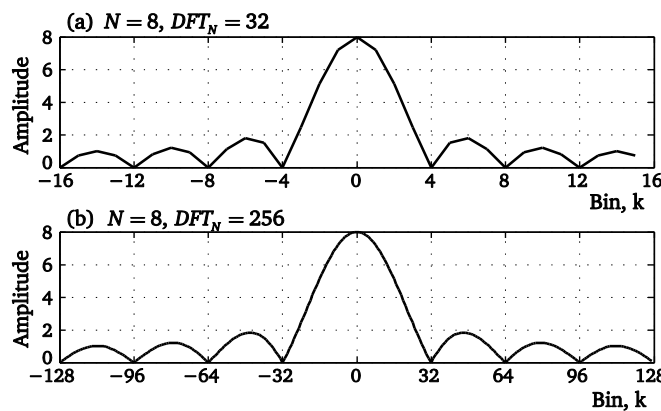
The DFT of the truncated signal is from eq. (7.4) given by the convolution of the signal DFT with the Eq. (7.5), the Diric function. The DFT of a finite signal that is truncated (and all signals are) always has this Diric term in it.

$$X_N[k] = e^{-j\omega\left(\frac{N-1}{2}\right)} \frac{\sin N \frac{\omega}{2}}{\sin \frac{\omega}{2}} * X[k] \quad (7.6)$$

The finite-time DFT hence is a “convoluted” DFT and not the DFT of the real signal. When we do a DFT where the bin resolution is an integer multiple of the signal frequency, we do not see the effect of the Diric function. Doing an idealized, perfect DFT is near impossible for real signals. A real signal has many frequencies and they are all sampled at different points, leading to the effect of the Diric function in Eq. (7.6) being observable in most all DFTs.

The windows come in many shapes, with rectangular window the most elementary. The DFTs of the windows have a fancy mathematical name, **Kernel**. The kernel of a rectangular window is called the Dirichlet kernel. Now although this is a sophisticated way of referring to the DFT of a window function, in this chapter, we will not use the term kernel and just call it “the DFT” so as not to make the topic jargon heavy.

### The rectangular window is everywhere



**Figure 7.2 – The spectrum of a rectangular window plotted using the sinc function.**

In Fig. 7.2(a) we see a 32-point DFT of an 8-point rectangular window. This is actually the *sinc* function plotted for  $N = 8$ , the presumed period of the signal. The reason for using the Sinc function for this plot is that Diric function for one period can be represented by a *sinc* function. For a DFT we need calculate only one period, so both sinc and Diric would result in the same thing.

In Fig. 7.2(a), we note that the height of the main lobe of the 8 point window DFT is equal to 8, the length of the window in samples. When normalized by  $N$ , that is equivalent to the gain of this window. A rectangular window function does not affect the area of the signal so it is gain-neutral. As we know from Fourier series, the 0<sup>th</sup> coefficient of the FFT is the area under the signal or its DC value. In Fig. 7.2 (b), we see the DFT of the same rectangular window but the 8 samples have been zero-padded for a 256-point DFT. Note that the height of the main lobe has not changed and neither has the number of side lobes. This because the height of the main lobe is a function of the *window length in sample points* and not the size of the DFT. But since there are more bins (larger  $N$ ), the frequency resolution is better in the second plot.

In Fig. 7.3, we increase the number of samples from 8 to 12 and then do the same thing. We do two zero-padded DFTs for  $N = 32$  and  $N=256$  points.

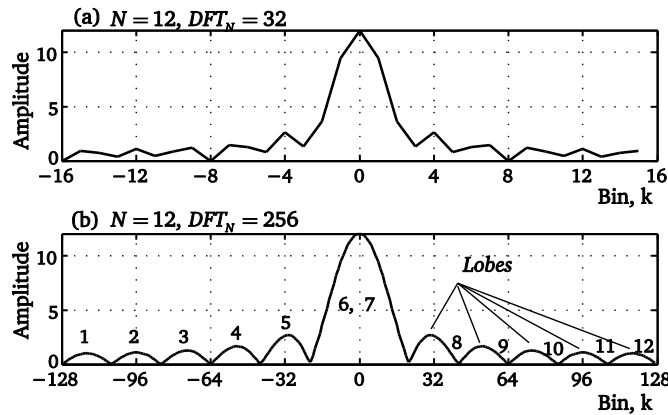


Figure 7.3 - (a) The DFT of a 12-point signal with  $N = 32$ , (b) Same 12 point window with a 256-point DFT showing a smoother plot.

Notice that the number of side lobes in Fig 7.2 and 7.3 are different. Why did the number of side lobes change from 3 to 5 when we changed the number of samples from 8 to 12? We see three side lobes in Fig. 7.2 and 5 in the Fig 7.3, the second case. In the second case the window is assumed to be 12 samples long. The number of lobes now equals 5 on each side (actually six, if we count two for the main lobe) or a total of 12, which is equal to the number of points or the length of the window. So the number of side lobes is not a function of the DFT size but how many sample points you have used for the window itself. This is a property of this window. As window length is increased, the number of side lobes increase along with the length.

The number of side lobes hence is always equal to  $N$ . Here we do a 64 point DFT of a 16 sample window and we get 16 lobes, despite a smaller DFT size. (Counting the center one as 2 lobes.)

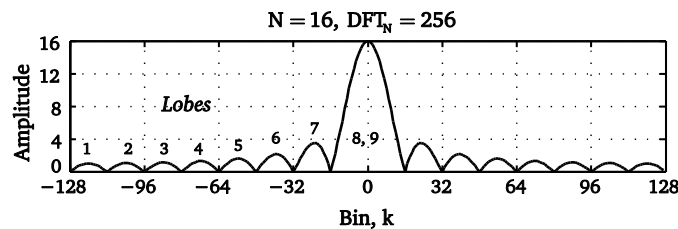


Figure 7.4, Window length 16 with DFT size = 256 gives us 16 lobes.

However, no matter how many side lobes we get, the difference in the height between the main lobe and the first side lobe (a ratio of powers, not magnitude) always stays at about 13.4 dB for the rectangular window. The first side lobe height is not a function of  $N$ . It is a nature of this window function. In fact the relative heights of the side lobes do not change with the length of the signal. There are more sidelobes but their relative heights do not change. The rate at which the sidelobes fall-off is an important parameter of this and all other window functions. The desired fall-off is *quick*.

Where is this 13.4 dB coming from? Let's compare the height of the main-lobe to the one next to it. We take the ratio of the two lobes, one at 0 frequency and the other at  $w = \frac{3\pi}{N}$ , which is the location of the next lobe.

$$\frac{W\left(\frac{3\pi}{N}\right)}{W(0)} = \frac{\sin\left(\frac{N\omega}{2}\right)}{N \sin\left(\frac{\omega}{2}\right)} = \frac{\sin\left(\frac{3\pi}{2}\right)}{N \sin\left(\frac{3\pi}{N}\right)} = \frac{1}{N \frac{3\pi}{2N}} = \frac{2}{3\pi} = 4.71$$

In dB this is equal to

$$R_{w=3\pi/N} = 20 \log\left(\frac{2}{3\pi}\right) = -13.4 \text{ dB}$$

Note the 20 instead of 10 as the multiplier because we are computing energy. You probably already know this 13.4 dB number. This is how far down the first lobe is from the main lobe of a rectangular function, be it for a pulse or a signal or a window function as in this discussion. In Figure 7.4 we plot the DFT of a rectangular window. The y-axis is in dBs and represents the energy or the power of the signal.

Let's compare the height of the second lobe to the third.

$$\frac{W\left(\frac{6\pi}{N}\right)}{W\left(\frac{3\pi}{N}\right)} = \frac{\sin\left(\frac{N\omega}{2}\right)}{N \sin\left(\frac{\omega}{2}\right)} = \frac{\sin\left(\frac{3\pi}{2}\right)}{N \sin\left(\frac{3\pi}{N}\right)} = \frac{1}{N \frac{3\pi}{2N}} = \frac{2}{3\pi} = 4.71$$

Check this.

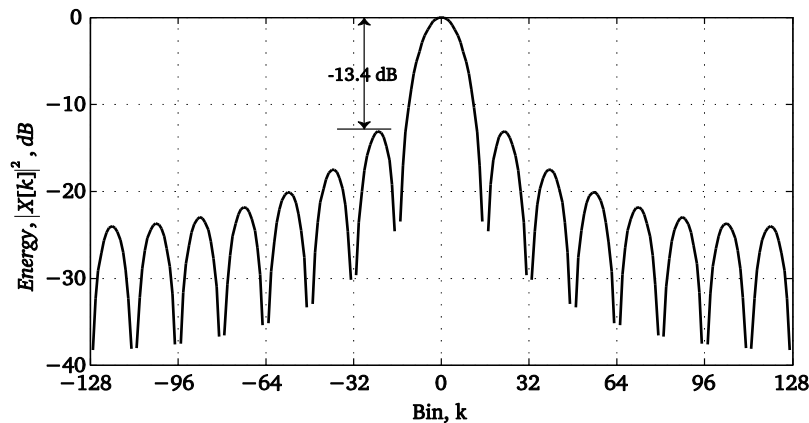
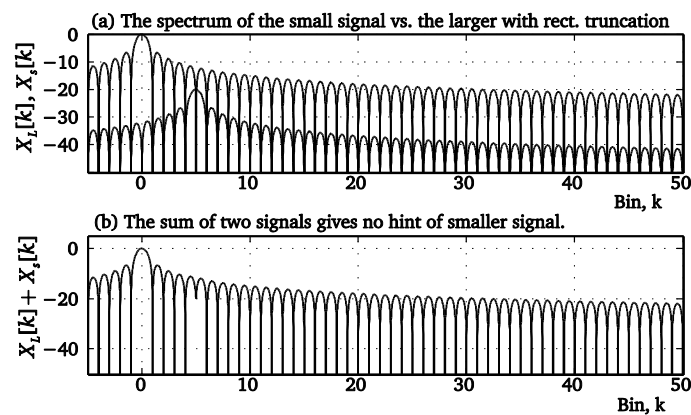


Figure 7.5 - The  $20 \log |X[k]|$  of a rectangular function in dBs.

What effect does this issue have on a signal and its DFT? At best the effect of the rectangular window is benign when the length of the window is equal to samples per period times an integer number of periods of the signal to which it is applied. In such a case, you will not see any effect in the DFT. Let's say that the signal is a pure sinusoids of period  $N$ , then the DFT of a signal of length  $N$  or  $kN$  where  $k$  is an integer, will be exactly the same. But when  $k$  is not an integer, we

will see the effect of the rectangular window in form of leakage. The rectangular window is called the default window because it is essentially always there for a finite length signal.

A big problem with our default window is that it can obscure small amplitude signals. In Fig. 7.6 we show two signals offset by a small frequency. The smaller signal is 20 dB below in level to the larger signal. We plot the DFTs and overlay them. We see in (a) that the smaller signal is below the side lobes of the larger one even at its peak. When we add the two signals in (b), we see nary a sign of the smaller signal. This is a problem. The side lobes of a rectangular signal are considered too high (we say, they don't roll-off fast enough.) for most applications such that they obscure smaller components of the signal. If there are spurious signals present in a signal, they are likely to cause distortion which can be heard but if not seen on the scope we may not realize they are there.



**Figure 7.6 – The small signal is obscured by the larger signal when the signal is truncated using a rectangular window.**

So can we fix this problem? There have been many different windows devised to improve on the behavior of the rectangular window. The most commonly used are the Von Hann (called Hanning window for some reason), the triangular window and the Hamming window. These are called **general cosine windows**.

### Window Quality metrics

Here are the parameters that can be used to compare windows depending on the application.

**Sidelobe level**- This is the ratio of the first sidelobe to the main lobe. For a rectangular window this number is 13.4 dB and for a triangular window it is approximately 27 dB. This number is important when we want to resolve closely spaced narrowband signals.

**Mainlobe width** – This is how wide the main lobe is. This parameter is a measure of how much a window spreads the signal. The rectangular window is always the *narrowest*.

**Fall-off** – This parameter tells us how quickly the sidelobes are falling off. For a rectangular window each side lobe drops off from the first one by -13.4 dB and then by an additional 6 dB



per octave. (An octave is the log of the ratio of the frequencies, so 3 octave would mean double the frequency.) From this we see that the third sidelobe is at app. -19 dB. For a triangular window the first lobe is 27 dB down from the main lobe, the second sidelobe is further 12 dB down. Hanning window has the fastest sidelobe fall-off of all commonly used windows at 18 dB per octave. This is easily seen by examining the DFT of the window.

Scalloping Loss – This is a measure of the total power in the side lobes vs. the main lobe. We want this number to be as low as possible. This loss is highest for the rectangular window since it is obvious that it has a lot of high lobes.

3 dB bandwidth – This is the bandwidth at which the DFT of the window is down by 3 dB. The rectangular window is the narrowest in this regard.

Coherent Power Gain - We said that all windows have a DC gain. The largest gain comes from the rectangular window normalized to 1.0, the triangular window hence would have a gain of 0.5. Most other windows have a gain of 0.5 or less. This is not particularly important parameter, but helps us understand the relative structure of each window.

Windowing has wide applications. It can be used in the field of filter design, beamforming as well as in spectral analysis. We will briefly discuss the use of windows in filter design and then go to the topic that relates to Fourier analysis.

### Filtering with Windows

A window is in fact a filter. Any function that distorts a signal in any way can end up changing its frequency response and by that definition, we can see that windowing is a form of filtering. The windows are used to create low pass filters from ideal impulse response. The response of an ideal filter (a rectangular pulse) is a sinc function. This is an ideal brick-wall filter with a sharp cutoff at a given frequency, as we see in Fig. 7.7 (b). The time-domain signal that gives rise to this frequency response is an infinitely long sinc function. But we *do* have to truncate it by necessity, and when it is truncated, it no longer offers the wonderful brick-wall quality. It has a roll-off and has ripples in-band.

The truncated signal response is shown in Fig. 7.7 (f). It is has a ripple in the passband which comes from the finite terms in the summation. This is also the same thing as the Gibbs phenomena mentioned in Chapter 1. The longer the truncation, the better this “filter”. If we multiply the sinc function with any well-behaved function in time-domain, the result is a filtered signal in the frequency domain.

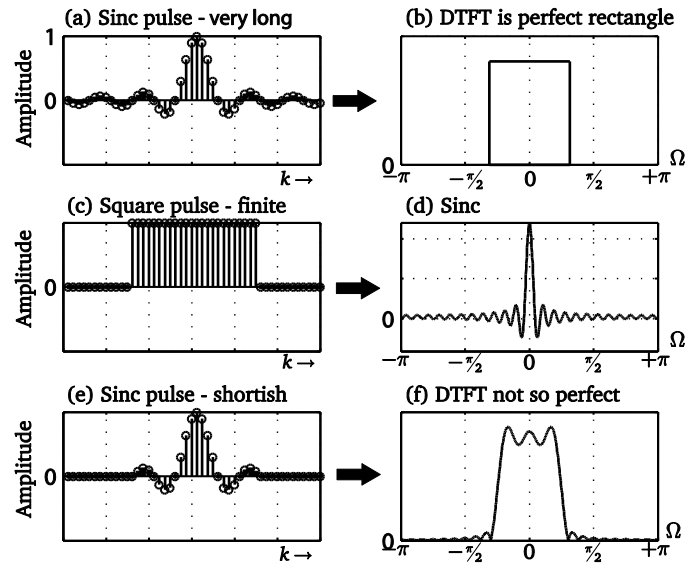


Figure 7.7 - In (a) we see the impulse response of the ideal low pass (brickwall) filter. If we could build an infinitely long signal, then clearly we would have this ideal low pass filter. However, we can't, so we truncate the ideal impulse response to a finite number of samples. The DTFT of such a function is well known by now as shown in (d).

Window functions of which there are many to choose from, can be used to create these filters. Here we have two filters created by windowing with two particular windows, the Hanning and the Blackman. See that the width of the main lobe is different based for each, as well as the roll-off or the taper at the edges. These are properties of the window functions and can be used to create many different filters depending on the application.

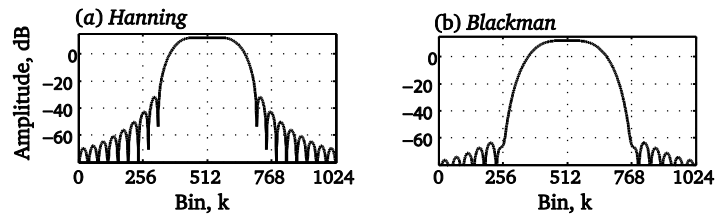


Figure 7.8 – Filter response created from windowing the impulse response of an ideal filter, (a) using the Hanning window with a narrower passband and (b) with Blackman window with a wider passband but better roll-off.

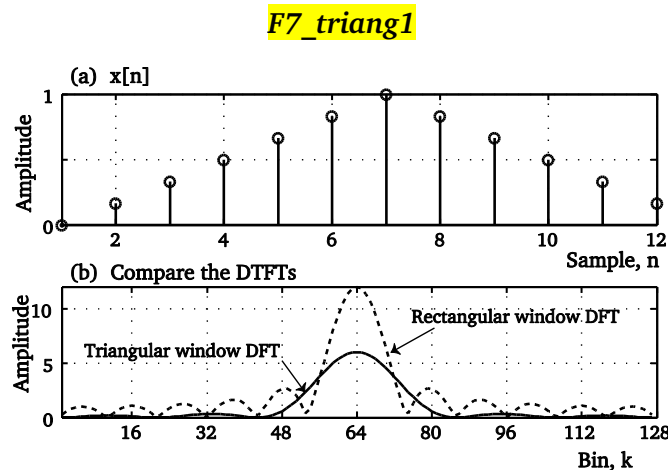
### Spectral estimation

We now discuss the use of windows in spectral analysis. Fourier transform is a wonderful tool but it has many limitations. Windows are used in Fourier analysis to control the effects of smearing of components that do not fall on the bins. We cannot make this effect go away so we mitigate it as best as we can. One of the ideas is that if the end points of the signal were zero, then the total signal presented to the DFT is made periodic. There are no discontinuities at the junction between one period and the next. So we look at a window function that forcibly pushes the end points to zero. However, recognize that windows are ultimately a type of filter and will

distort the signal. One hopes that it is done in a way that helps us to better see the behavior of the signal.

### Comparing windows

The simplest window we can think of after the rectangular window is the *triangular window*. It is just a linear attenuation of all signal amplitudes from the center of the signal to the ends on each side as shown in Figure 7.9 (a). The triangular window is also called *Bartlett window*, named after the man who described its effect on spectral estimation. Its DFT is plotted along with the DFT of the no-window or the rectangular window in (b).



**Figure 7.9 - The triangular window in a linear attenuation of all samples down to zero at the ends. Its DFT in (b) has a wider main lobe than the rectangular window but very low side**

Comparing the triangular window DFT plotted in Fig. 7.9 with the rectangular, we see that the peak of the triangular window DFT is about half of the rectangular window DFT. This is obvious when you note that the triangular window is actually attenuating the signal. The loss is exactly 3 dB. The other thing we note is that its main lobe is wider than the rectangular window but its sidelobes roll-off faster. So here we have one advantage that is faster roll-off but a disadvantage in that the main lobe is wider than the rectangular window. This sort of thing happens with nearly all the windows. If they offer good roll-off then we end up with a more spreading due to the wider main lobe.

A linear triangular window is given by the expression

$$w_N[n] = 1 - \left| \frac{n - (N-1)/2}{(N-1)/2} \right| \quad 0 \leq n \leq N \quad (7.7)$$

One interesting thing about the triangular window function is that, a triangular function is generated by the convolution of two rectangular signals. The DFT of the convolution of the two rectangular functions then is equal to the DFT of the rectangular window squared. The DFT of the rectangular window is a sinc function, hence the DFT of the triangular function is a sinc-squared function.

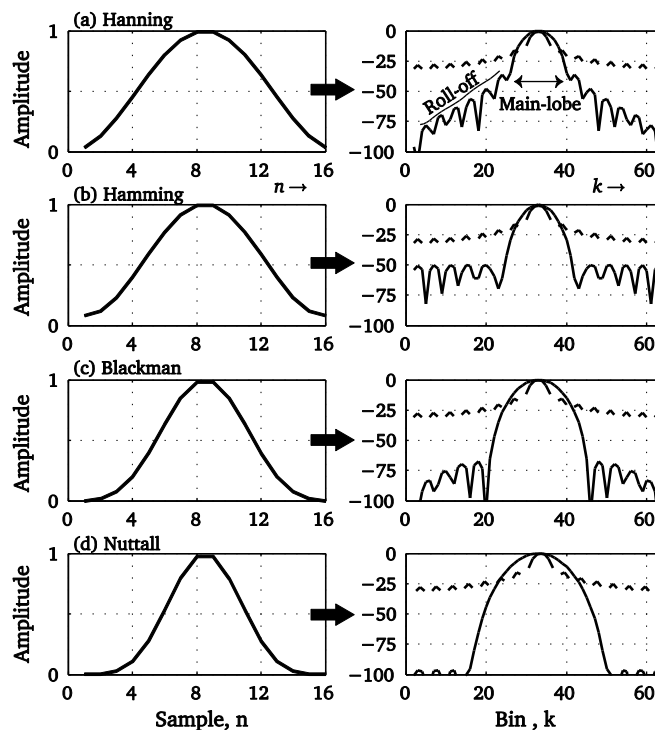
$$W_N[k] = \text{sinc}[k]^2 \quad (7.8)$$

In engineering most things are derivative. One person gets a bright idea and then many work on it to improve it incrementally. Same here. All these window functions are actually related to each other and not independent, unrelated inventions. Once the idea of window was described, a big class of windows called the generalized cosine windows were devised. These are given by this basic equation with variable parameters,  $a_k$  with  $N$  being the length of the window, and  $K =$  order of the window.

$$w[n] = \sum_{k=0}^K a_k \cos\left(\frac{2\pi kn}{N}\right) \quad \text{for } 0 \leq n \leq N-1 \quad (7.9)$$

By varying these parameters the following windows are created.

Window	$K$	$a$	$b$	$c$
Rectangular	1	1	0	0
Hanning	2	0.5	-.5	0
Hamming	2	0.54	-.46	0
Blackman	3	.42	-.5	.08
Nuttall	4	.36	.49	.14



## F7\_4windows

**Figure 7.10 - In each case, as the side lobes decrease, the main lobe bandwidth increases. Hence there is less smearing but less ability to resolve close-in components.**

With each window, there is a tradeoff between the increasing width of the main lobe and the decay rate of the side lobes. In Figure 7.10, we plot several of these on dB scale so the differences are more obvious. The rectangular window is plotted as reference and it is clear how poor its roll-off is compared to these other windows. The conclusion we can draw from this plot is that all windows have wider main lobe than the rectangular window but better roll-off of energy in side lobes.

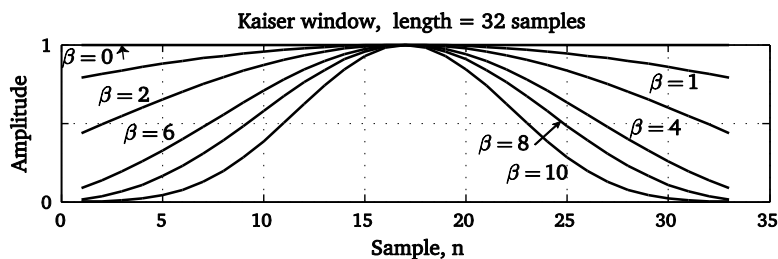
### The flexible Kaiser window

Most windows are of a fixed response as we see in Fig. 7.10. There is another class windows that have variable parameters which allow one to change the response by changing *some* parameters. Dolph-Chebyshev and Kaiser are two such types. Of these Kaiser window is better known and here we examine its properties. The window is defined by the equation

$$w_{\text{kaiser}}[n] = \frac{I_0\left[\beta\sqrt{1-\left[\frac{n-m}{m}\right]^2}\right]}{I_0[\beta]} \quad \text{for } 0 \leq n \leq N-1 \quad (7.10)$$

where  $m = (N-1)/2$

Here  $I_0$  is the 0-order Bessel function. By varying the parameter  $\beta$ , the time domain shape of the function can be changed, which results in a different frequency response. In Fig. 7.11, we see some possible shapes of the Kaiser window for  $\beta$  from 0 to 10. As  $\beta$  is increased, the window begins to force the end points to zero.

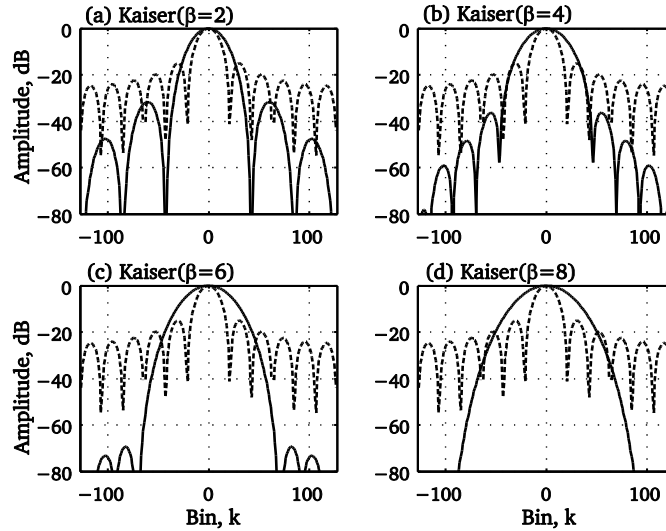


## F7\_kaiser

**Figure 7.11 - The shape of the window obtained by varying the Kaiser window parameter,  $\beta$ .**

When  $\beta$  is 0, the Kaiser window is identical to the rectangular window. By changing this parameter, in fact most all of the other windows can be approximated by the Kaiser window. For example if  $\beta = 4.86$ , we get pretty nearly the Hamming window. When  $\beta = 5.3$ , we get very nearly the Hanning window. So perhaps that is a good thing, however, in reality, it is hard to remember these parameters and most people end up using a window they know. Despite that the Kaiser window is very powerful. We can change the parameters and immediately get a custom

response. The Kaiser window allows us to fine tune the side lobes vs. the main lobe width. By changing  $\beta$ , the Kaiser window parameter, the window can be made to resemble all other windows.



**F7\_kaiser4**

**Figure 7.12 - Kaiser window parameter allowing modifying the properties of the effect on the fly.**

In Fig. 7.12, the Kaiser window DFT is plotted along with the rectangular window DFT. As  $\beta$  increases, the main lobe gets wider but the side lobes drop. A wider main lobe implies that a signal maybe misidentified. If there is smaller signal nearby, a wider window will likely mask it and will not allow it to be seen in the DFT, as we saw in Fig. 7.6. Hence when we want to discriminate between nearby signal components, we want to pick a window that has a narrower main lobe. When we want to find far away components faithfully, we use a window that has lower side lobes.

**Example 7-1**

Let's examine this signal. It contains several sinusoids of non-harmonic frequencies.

$$x = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) + \cos(2\pi f_3 t) + 0.7 \cos(2\pi f_4 t) + 0.2 \cos(2\pi f_5 t) + 0.1 \cos(2\pi f_6 t)$$

$$f_1 = .375, f_2 = .375 \times 1.3, f_3 = 2.375, f_4 = 6.33, f_5 = 7.45, f_6 = 10.45$$

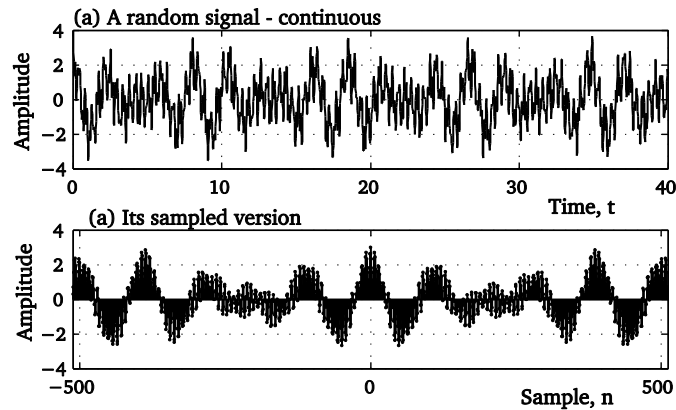
The sampling frequency is 48 samples per second, and the FFT size is 1024. From this we

calculate the frequency resolution  $R_s = \frac{48}{1024} = .0469 \text{ hz}$

The discrete version of the signal is given by

$$x[n] = \cos(2\pi f_1 \frac{n}{F_s}) + \cos(2\pi f_2 \frac{n}{F_s}) + \cos(2\pi f_3 \frac{n}{F_s}) + 0.7 \cos(2\pi f_4 \frac{n}{F_s}) + 0.2 \cos(2\pi f_5 \frac{n}{F_s}) + 0.1 \cos(2\pi f_6 \frac{n}{F_s})$$

Here we replaced variable  $t$ , with  $\frac{n}{F_s}$ . In Fig. 7.12, we see the signals plotted both as a continuous signal and as the sampled discrete signal. In the continuous form, it is really hard to see if the signal is periodic. This is how real signals look, we just never quite know what the period is. We have to guess. What we want is to over-estimate, so we often end up picking more than one period, and often several. But then the problem is what we have selected visually is not liable to be exactly an integer number of periods and hence we will have discontinuities at the ends.

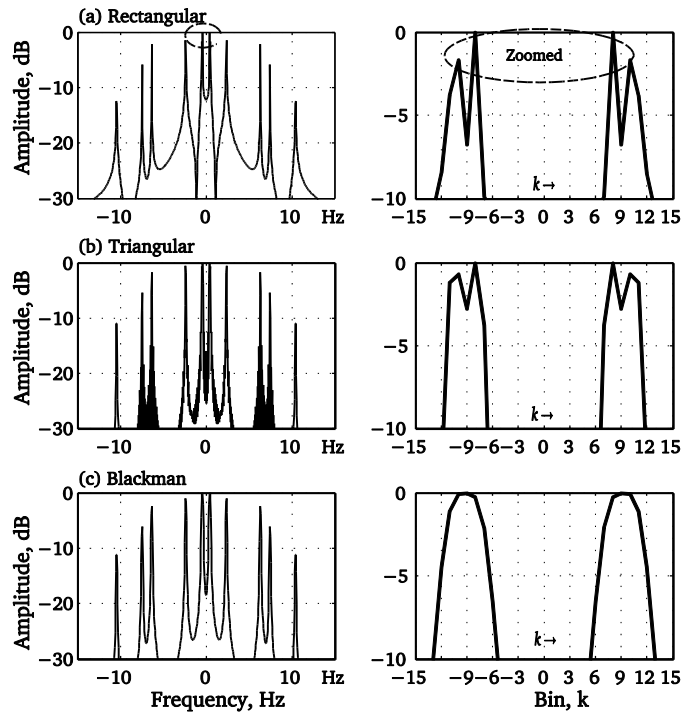


F7\_random

**Figure 7.13 – The continuous and the discrete version of the signal. Which part do we use to do the DFT?**

So we select the piece shown in (b), 1024 samples long DFT as shown in Figure 7.13. For a DFT of 1024 points, the component at  $f_1$  is located at the 8<sup>th</sup> bin and component  $f_2$  is located between bin 10 and 11.  $f_3$  is also located exactly on a bin but all the others are not so lucky and fail to fall on bins. In the plot in Fig. 7.14, the x-axis on the right side showing the whole spectrum is given in frequency. This is computed by multiplying the harmonic index  $k$  with  $N/F_s$ . On the right the detailed view of the x-axis is given in bin numbers instead.

Now we look at the DFT of this signal using a “no-window”, which is the rectangular window, a triangular window in row (b) and the Blackman window in row (c)). What we see is that when it comes to discriminating between the two very close signals,  $f_1$  and  $f_2$ , the rectangular window does the best job. Note the detailed view of the two signals on the right. The triangular window then is somewhat better and the Blackman window rather poor. On the other hand, take a look at the last component that is 10 dB below the ones in the center. The rectangular window measures it as -12.41 dB, whereas the Blackman comes much closer to -10 dB.



F7\_compare2

Figure 7.14 – DFT of a signal using different windows. All windows offer improvements in different areas.

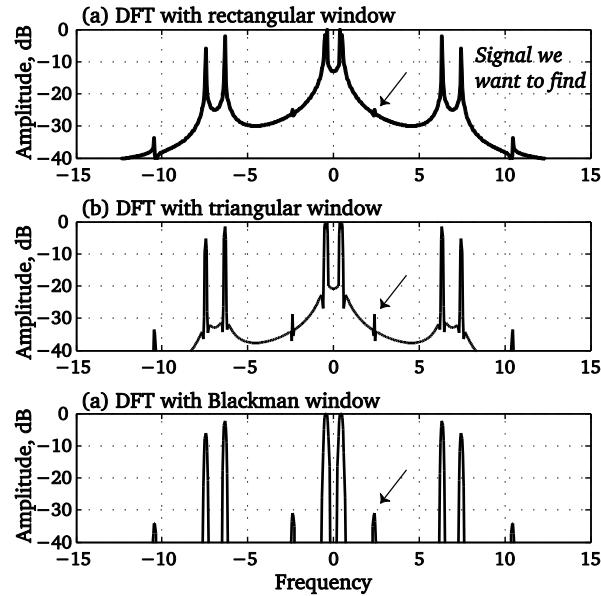
#### Example 7-2

Now we have attenuated the two of the components of the same signal,  $f_3$  and  $f_6$  by 30 and 33 dB per the following equation.

$$x[n] = \cos(2\pi f_1 \frac{n}{F_s}) + \cos(2\pi f_2 \frac{n}{F_s}) + .001 \cos(2\pi f_3 \frac{n}{F_s}) + .7 \cos(2\pi f_4 \frac{n}{F_s}) + 0.2 \cos(2\pi f_5 \frac{n}{F_s}) + 0.0005 \cos(2\pi f_6 \frac{n}{F_s})$$

When we examine the spectrums in Figure 7.15 for this case, we see that the rectangular window loses these signals. We barely see the first one. The triangular window does a bit better job and Blackman the best.





F7\_715

*Figure 7.15 - DFT of example signal which contains small amplitude signals, show how some windows are better than others in allowing these small signals to be seen.*

So in general when using windows to improve the spectrum of a signal keep in mind that when we are interested in signal discrimination near the center, we should leave the signal unwindowed (a rectangular window) but when looking at faraway small signals, a different window is better. But even that is a rough guideline. It seems that there are no hard lessons one can use. The use of windows in spectral estimation is more of an art than a science.

### Example 7.3

Now we will examine the application to some real data collected. In Fig. 7.16 we see 50,000 samples of a signal collected. The signal is a 16APSK modulated data after it has gone through a non-linear device with noise. We need to examine the spectrum to determine how much noise the signal is causing in an adjacent band.

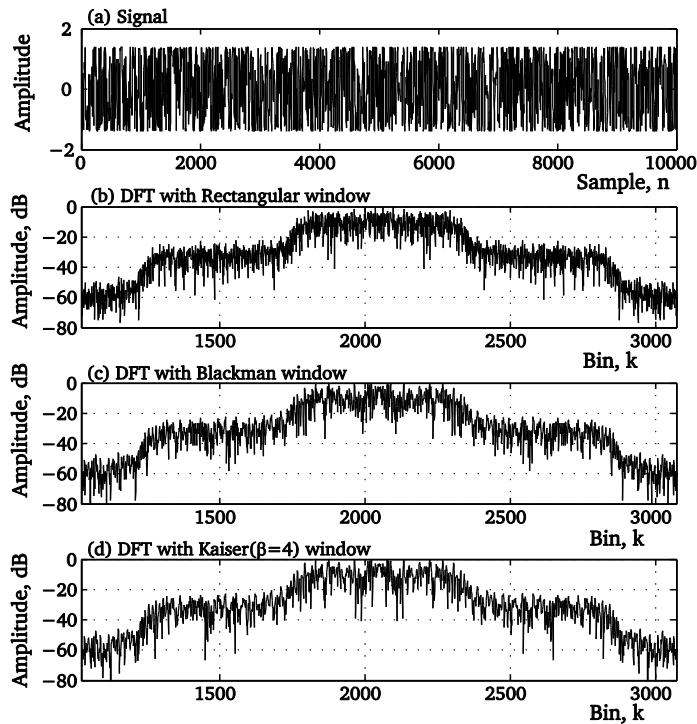


Figure 7.16 – Spectral analysis of a random signal using windows.

### F7\_randomSigs

In this case, we do three different DFTs of 4096 points long, each with a different window. We have problems here. First, the signal has a great deal of variation. Both Blackman and the Kaiser window do manage to reduce some of this variation but the spectrum is still far too variable to be of much practical use.

The obvious thought that comes to mind is that we may do a bunch of DFTs in different places of the signal and then average them. Remarkably this does not reduce the variance. This brings us to the subject of random signals and the type of issues that the randomness introduces in spectral analysis. We will proceed with this subject in the next chapter. The application of windows to random data for spectral estimation is an important one, and often the main reason why we do Fourier analysis.

### Summary

1. If the DFT size is not exactly an integer multiple of the period of the every single frequency in the signal, the resulting spectrum will contain leakage.
2. The leakage occurs when a signal component frequency does not fall precisely on a DFT bin.
3. The bin frequency resolution is given by sampling frequency divided by the size of the DFT,  $N$ .
4. The frequency of each bin is bin times the resolution.

5. For most signal we do not know the period. However by looking at the ends of the signal, we can tell if we will get leakage. If the ends do not match, then leakage will be present.
6. The use of windows improves the DFT by artificially distorting the signal such that it appears periodic to the DFT. It is done by forcing the end points to be zero.
7. There are two types of windows, those that have fixed parameters and those that have parameters that can be changed to change the response.
8. Bartlett, Hanning, Hamming, Nuttall, Cosine, Blackman are fixed types. Kaiser window has variable parameters.
9. Windows are used for filter design and in spectral estimation.
10. The use of windows depends on the type of signal and is very application specific.

Copyright 2015 Charan Langton, All Rights reserved.