

# A Comprehensive Windows Tutorial

Howard A. Gaberson, Oxnard, California

This article is a tutorial attempt to provide an easier analysis of how windows work. I begin by looking at individual spectrum bins as affected by off-bin-centered tones with six different windows. I define and use the convolution theorem to explain why the windows do what they do. I review the sophisticated misunderstood fat-center-peaked, picket-fence-looking FFTs of various windows that come from adding zeros to increase resolution. I then use these drawings in a step-by-step graphical convolution to show how this or that window will affect the FFT of your data. This is a procedure to demonstrate how any oddly shaped, even graphically defined, window affects data. These techniques are tools to allow you to invent a graphical window to affect data in new ways. To demonstrate how to use graphically defined windows, I use digitized drawings of a Hawaiian mountain and the popular cartoon beagle as windows. To use these windows, I demonstrate FFT-IFFT interpolation and resampling, which in itself is interesting. Finally, I compare how eight windows affect the spectrum of typical vibration data.

We need a window for vibration spectrum analysis because the beginning does not match the end of the data segment we are analyzing and because we virtually never have an integer number of periods of any cyclic information in the signal segment/chunk. The discrete Fourier transform (DFT) analysis is really a Fourier series expansion of whatever segment we input. It is considered periodic. Mismatched ends and noninteger numbers of periods garbles the result. Windows somewhat alleviate that garbling but in a complicated way.

Windows are shapes – like a hill, a hump in the center, and they go to zero at the ends. When you multiply these window shapes term by term with a segment of data, you force the ends of the data segment to zero, so at least the ends match the beginning. You cannot really fix the noninteger number of periods. Figure 1 shows a window application. In Figure 1a we have a 1,024 value plotted list of acceleration data; in Figure 1b, a 1024 value Hanning window plotted for the same time values. Figure 1c shows the product of the two. The beginning and ends certainly match.

I am sure there are more than 100 windows described in the literature. Figure 2, shows six of them that I happen to use and will illustrate here: boxcar (or uniform), Hanning, Hamming, Gaussian, Kaiser Bessel, and flat-top (this is very different from the boxcar or uniform window). Signal analyzers frequently offer five or six to choose from, with variably helpful discussions of when to use each. Hanning is the most widely recommended. Reference 1 gives 44 and Reference 2 gives eight more; these might be the most widely referenced windows papers, but these authors hadn't yet heard of the flat-top. Ron Potter<sup>3</sup> gives about five windows and is famous for an unpublished HP publication. Potter did have a flat-top in there. Finally, the venerable *Brüel & Kjær Technical Review* published the first fully disclosed flat-top window and a nice version of the Kaiser Bessel window.<sup>4,5</sup> These two publications are downloadable from the B&K web site. I read a paper<sup>6</sup> from the 2004 International Modal Analysis Conference where the investigator was working on keeping the side lobes extremely low because of all the new high-resolution, 24-bit, data-collecting hardware. If you really want to dig into things, Harris<sup>7</sup> wrote a difficult 64-page article with excellent drawings that I recommend.

## Defining the Bin-Centered Concept

Non-bin centered or not, an integer number of periods or cycles in the analyzed data segment is the topic to start with. A sine wave component of our signal is bin centered if it has an integer number of periods in the data segment being ana-

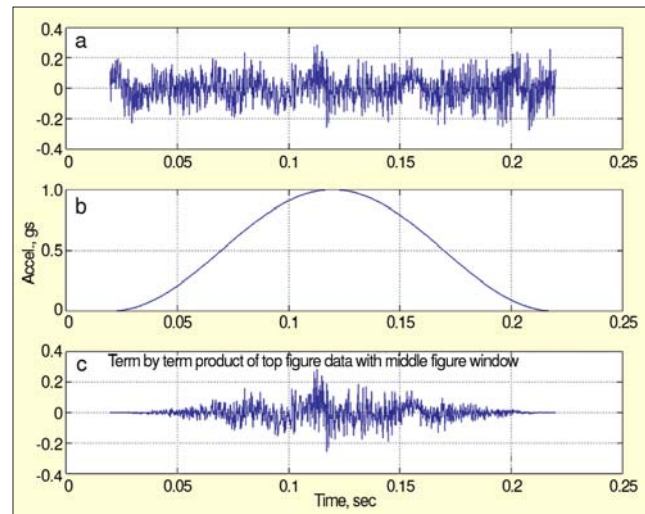


Figure 1. This shows a 1,024-point Hanning window applied to 1,024-point chunk of air handler acceleration.

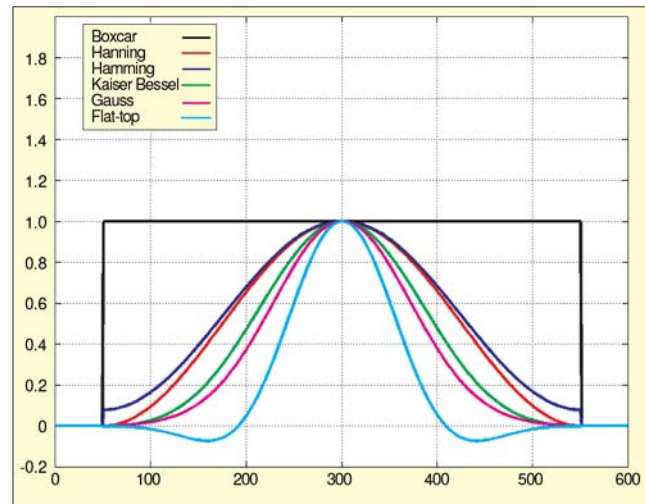


Figure 2. Six common window shapes.

lyzed, which almost never happens without planning. We can develop a formula for testing if a sine wave signal component is bin centered as follows. A sine wave is given by Equation 1:

$$y = \sin 2\pi ft \quad (1)$$

where  $f$  is the frequency in cycles/second (Hz). The period of a sine wave  $P$  (seconds per cycle) is the reciprocal of the frequency (Equation 2):

$$P = \frac{1}{f} \quad (2)$$

Since we have to work with digitized data, define:

$N$  = number of samples in data segment being analyzed

$f_s$  = sample rate in samples/second

The time interval between samples  $h$  is the reciprocal of the sampling rate  $f_s$  (Equation 3):

$$h = \frac{1}{f_s} \quad (3)$$

The signal duration  $T$  is given by Equation 4:

$$T = Nh = \frac{N}{f_s} \quad (4)$$

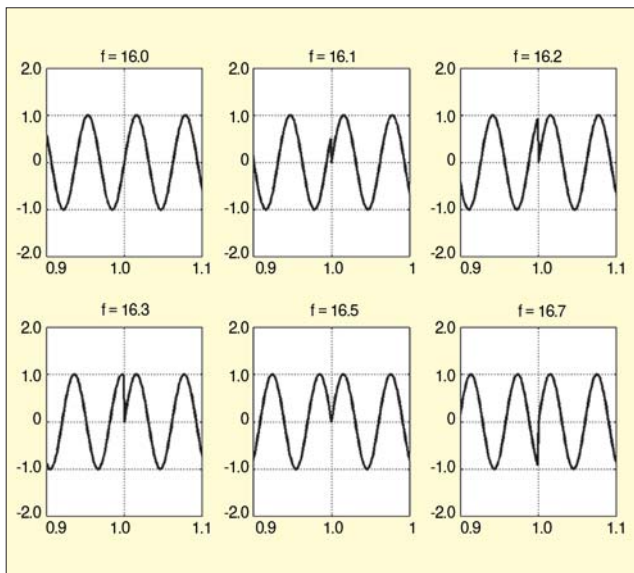


Figure 3. Non-bin-centered tone time histories; except for the case of exactly 16 Hz, the beginnings and ends do not match.

The number of periods  $N_p$  has to be given by the duration over the period (Equation 5):

$$N_p = \frac{T}{P} \quad (5)$$

By substituting Equations 2 and 4 into Equation 5, we find:

$$N_p = \frac{Nf}{f_s} \quad \text{from} \quad \frac{T}{P} = \frac{N/f_s}{1/f} \quad (6)$$

Now, as long as  $N_p$  is a whole number, the sine wave in the data segment is bin centered or has an integer number of periods. Such test segments of a sine wave signal are needed for calibrating a window coefficient in our spectrum calculation programs, so this is a handy formula to keep in mind. In the case of a sampling rate of 1,024/second and if our number of samples is also 1,024, the number of periods will be equal to the frequency. I will set things up that way here. To illustrate how the end and beginning not matching affects things, I will draw an expanded view of the end of a 16-Hz, 1,024-point sine wave chunk connected to the beginning of the identical 16-Hz, 1,024-point sine wave chunk. This is shown for 16 Hz and several frequencies close to 16 Hz in Figure 3. I have expanded the region where the end of the first sine meets the beginning of the second sine to make it clear that a discontinuity occurs when the frequency is not a whole number. Notice that the two curves match perfectly (end to beginning) only for the case of 16 Hz, which makes the signal bin centered.

### Non-Bin-Centered Effects

Now we are in an excellent position to consider the effects that those six windows I mentioned help us alleviate (but cannot really cure) the picket fence and leakage effects of a non-bin-centered signal on the spectra we compute. In case you are rusty, I want to review some spectrum fundamentals so you will know what to expect. I use the acronyms DFT and FFT synonymously. DFT means the discrete Fourier transform, and FFT means the fast Fourier transform, the calculation procedure used to calculate the DFT. All signal analyzers and data collectors use an FFT to initially compute and then develop the spectrum from that result.

The DFT of an  $N$ -point signal, sampled at  $f_s$  gives us a spectrum of  $(N/2 + 1)$  amplitudes at frequencies from 0 to  $f_s/2$ . We frequently call these  $(N/2 + 1)$  amplitudes the bin values. The  $\Delta f$  or bin frequency spacing, is  $1/T = f_s/N$ . We get a frequency value starting at  $f = 0$ , and values spaced at  $\Delta f$  all the way up to  $f_s/2$ . There will be  $N/2$  intervals between 0 and  $f_s/2$ ; therefore, each interval or space is  $\Delta f = (f_s/2)/(N/2) = f_s/N$ . This is  $1/T$ . The time between samples is  $1/f_s$ , and  $(N)(1/f_s) = T$ . These

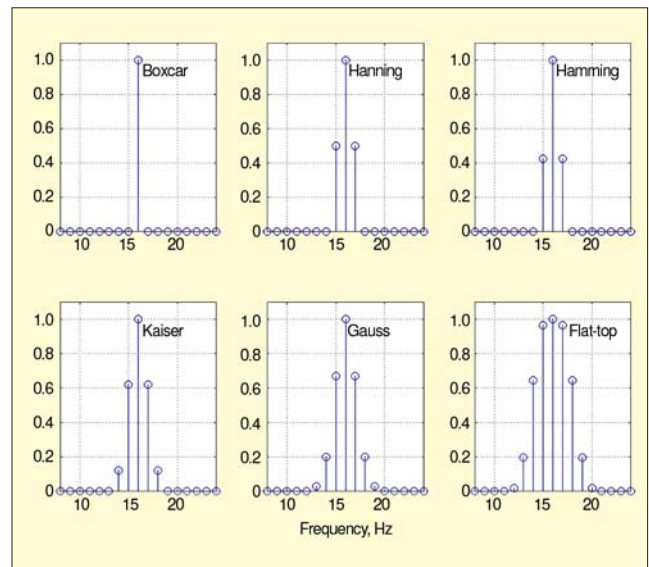


Figure 4. Expanded DFTs for 16 Hz; this is the bin-centered case, six different windows.

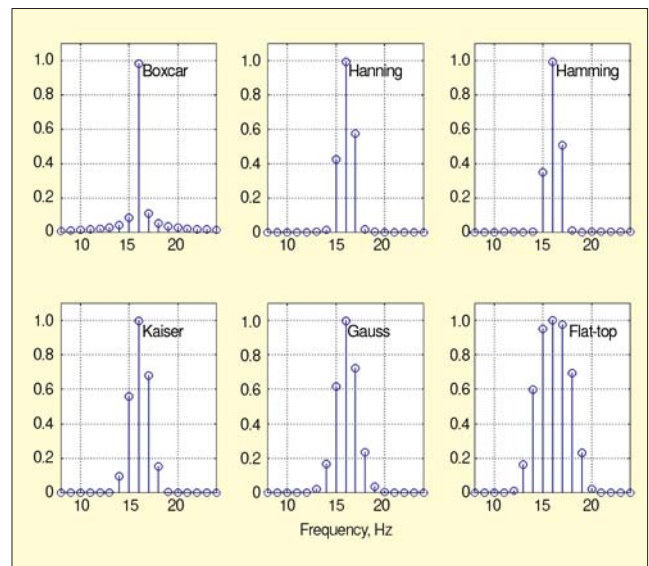


Figure 5. Expanded DFTs, 16.1 Hz 10% off bin center, six different windows; all windows seem reasonable in this situation, but the boxcar window shows side lobe leakage.

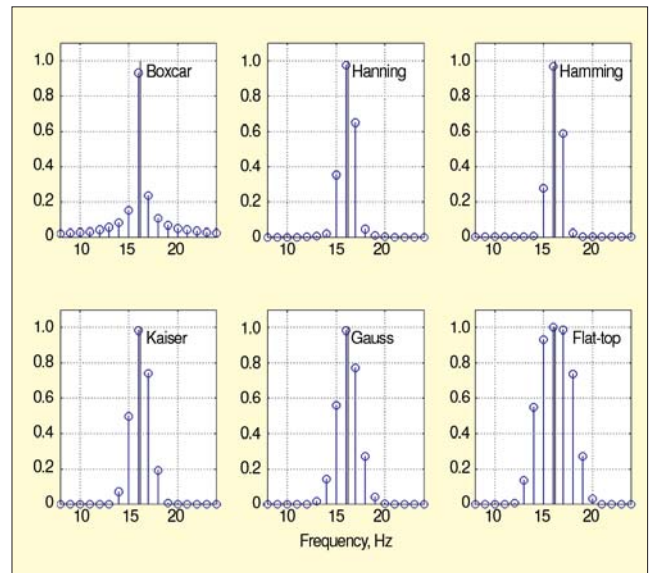


Figure 6. Expanded DFTs, 16.2 Hz, 20% off bin center, six different windows; thin line shows actual tone positioned at 16.2 Hz.



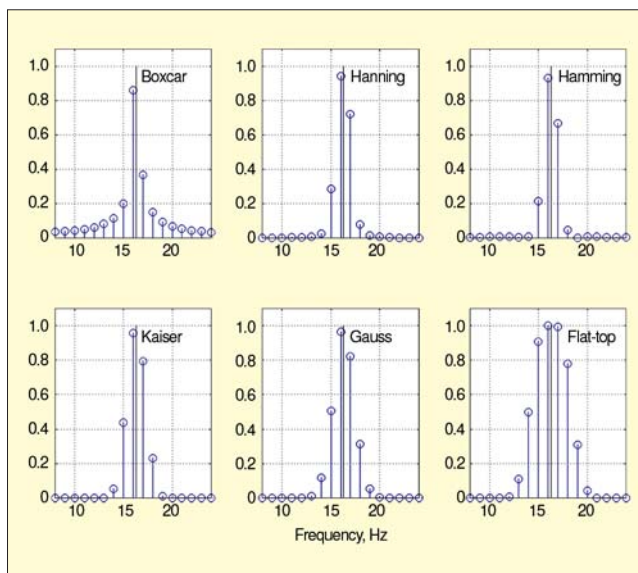


Figure 7. Expanded DFTs, 16.3 Hz, 30% off bin center, six different windows; boxcar is close to 20% in error now.

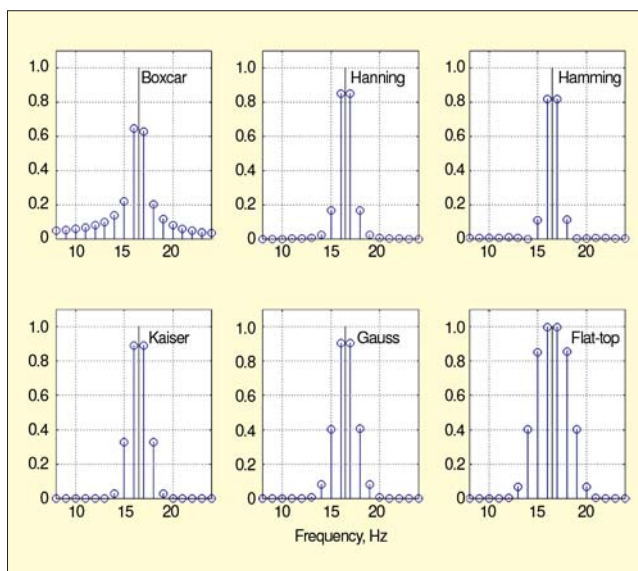


Figure 8. Expanded DFTs, 16.5 Hz, 50% off bin center, six different windows; this is the worst case, but note the flat-top window still gets the amplitude correct.

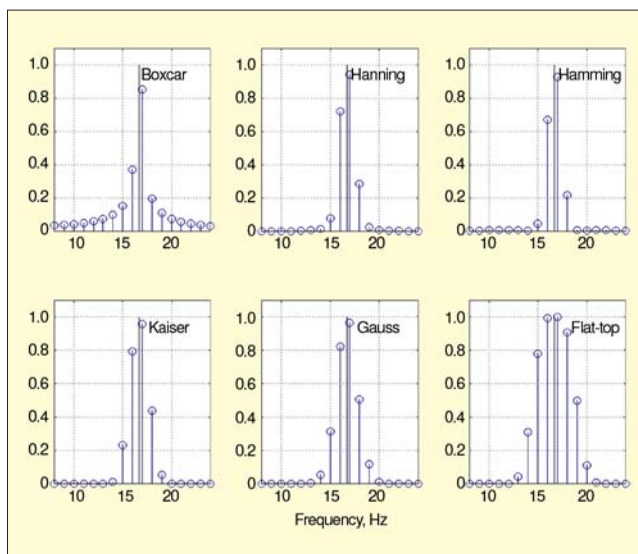


Figure 9. Expanded DFTs, 16.7 Hz, 70% off bin center, six different windows; this is really 30% off 17 Hz.

facts are important; you have to remember them. It is trouble to prove, but if you want to see it proved, ask me for a copy of Reference 8.

Now, let us check out all six windows on bin-centered and the other Figure 3 cases with varying degrees of not being bin centered. We start with a 1,024-sample signal of a 16-Hz sine sampled at 1,024/sec. From Equation 6, this will have exactly 16 periods. With 1,024 samples sampled at 1,024/sec, the signal duration  $T$  equals 1 sec. On the spectrum,  $\Delta f = 1/T = 1/(1 \text{ sec}) = 1 \text{ Hz}$ . If we stay with 1,024 samples and the same 1,024 sampling rate, the  $\Delta f = 1 \text{ Hz}$  stays the same no matter what we do to the frequency to change the number of periods and examine non-bin-centered effects.

Figures 4 through 9 show an expanded view of the spectra (around 16 Hz), with the six windows applied to the degrees of non-bin centeredness of Figure 3. In these spectra, I have used MATLAB's "stem-plot" feature to emphasize the individual bin values. Figure 4, shows the six windows applied to the spectrum calculation when the frequency is exactly 16 Hz, the bin-centered case. The boxcar or "no window" shows that we have a single value of one in the spectrum in the 16-Hz bin. All of the other five windows show content in the bins next to 16 Hz. In this case, they are indicating content where we happen to know there actually is none. There is some stuff to dislike about windows, but as we move on, you will come to agree that we have to use them. Both the Hanning and the Hamming windows indicate content at 15 and 17 Hz as well. The Kaiser Bessel window shows content in five bins, from 14 to 18 Hz. The Gauss window shows content from 13 to 19 Hz, and the flat-top window shows content from 12 to 20 Hz. Recall that the signal only contains an amplitude of 1 at 16 Hz.

Figure 5 shows what happens if we raise the frequency to 16.1 Hz and thus have 16.1 periods, 10% off being bin centered. The bin content shifts; the boxcar windowed spectrum now shows content from 13 to 17 Hz and shows the amplitude at 16 Hz down a little. All the other windows seem to say we still have an amplitude of 1 and a sine wave at 16 Hz. In Figure 6, I have shown the spectra for a 16.2 Hz sine wave with 16.2 periods, 20% off bin centered. I have shown the 16.2-Hz amplitude as a black line. The boxcar spectrum is indicating 0.92 amplitude content at 16 Hz, with additional content spread from probably 10 to 23 Hz. This indication of content where there is none is called leakage. Notice that all the other windows except the flat-top are indicating a content less than 1 still at 16 Hz. The flat-top is still hitting 1 right on the money at 16 Hz, and even the 17 Hz bin is pretty close to 1.

Figure 7, shows the situation with all six windows for 16.3 Hz, 30% off bin centered, 16.3 periods. Now the poor boxcar spectrum thinks we have a tone of 0.85 at 16 Hz, with content in all the adjacent bins shown from 8 to 24 Hz. The fact that the boxcar spectrum shows a 15% error is called the picket-fence effect. Notice that all the other windows do much better than the boxcar, with the flat-top showing we have content between 16 and 17 Hz with an amplitude of 1. The Hanning seems to beat the Hamming, and the Kaiser and the Gauss are considerably more accurate.

The worst case is Figure 8, the spectra for 16.5 Hz or 16.5 periods. The Hanning spectrum indicates equal content at 16 and 17 Hz of about 0.84, a 16% error. That is how this picket-fence error goes; the boxcar spectrum indicates roughly equal content at 16 and 17 Hz of about 0.63, or a 37% picket-fence error. Except for the flat-top, which indicates an amplitude of 1 content at 16 and 17 Hz, the Gauss window shows maybe 0.96, and the Kaiser Bessel is close behind at about 0.95. Notice also that the boxcar spectrum shows content in every bin, and the other windows restrict this spreading of content. This effect is called reducing leakage; this is a leakage of content into adjacent bins where we know there is no content in the true signal. Picket fence and leakage reduction: that is why we need windows. We cannot eliminate the effects, but windows reduce them. Figure 9 shows the same information about 16.7 Hz, or 30% off 17 Hz.

## More DFT Background:

We have to look a little closer at the DFT and how it is used to compute our spectrum. We have to look at the DFT of a window, and I have to explain this shifted idea. The DFT equations are Equations 7a and 7b:

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \quad (7a)$$

$$x_n = \sum_{k=0}^{N-1} X_k e^{i2\pi kn/N} \quad (7b)$$

Equation 7a is the DFT analysis equation; it analyzes  $N$  samples of digitized data, the  $x_n$  into  $N$  DFT values, the  $X_k$ . The  $NX_k$  values are complex sine wave amplitude values; we say they are complex because they contain an amplitude and a phase or the same thing is real and imaginary parts. Equation 7b is the synthesis equation; it exactly transforms the complex  $X$ 's back to the original  $x$ 's. However, I have the  $1/N$  where I think it belongs, not where it is usually placed.

To use Equations 7a and 7b for vibration analysis, there are some ground rules. The data list is sampled at sampling rate  $f_s$ . The  $x_n$  of Equation 1 are samples from a signal that was accurately sampled and band limited to  $f_s/2$ ; this means its Fourier transform is zero for all frequencies greater than half the sampling rate. There is an even number of  $N$  samples in the list. In Equations 7a and 7b, the complex exponential is also exactly Equation 8:

$$e^{-i2\pi kn/N} = \cos \frac{2\pi kn}{N} - i \sin \frac{2\pi kn}{N} \quad (8)$$

We can see the frequency in that  $2\pi kn/N$ , if we multiply it by  $hf_s$  which equals 1, by Equation 3, and rearrange as in Expression 8a:

$$\cos \left( 2\pi \frac{kf_s}{N} nh \right) \text{ is analogous to } \cos 2\pi ft \quad (8a)$$

Thus, in digitized terms,  $nh$  is the discrete time, and  $kf_s/N$ , is the frequency. The time index is  $n$  and the frequency index is  $k$ .

Let us repeat: the DFT as an exact transformation of a digitized vibration signal. It transforms the data into discrete (or sampled) complex exponentials (or equivalent sinusoids). The transform is the list of their amplitudes as a function of frequency. Equation 7a transforms the signal into  $N$  sine waves; each  $X_k$  is the amplitude and phase of a complex sine wave. Figure 10 attempts to show one of the  $k$  discrete complex sinusoids in 3-D.  $X_k$  is its complex amplitude or a vector from the origin to the beginning point of the discrete spiral. The little circles on the spiral represent the values of the discrete sinusoid for the sequence of  $n$  values. The smooth spiral on which the data lie is the underlying curve, the curve with time taken to be continuous or with  $nh$  replaced by  $t$  in Equation 8a. Each  $X_k$  is the complex amplitude of one of the discrete spirals. Equation 7b says to add them all up and you have your original signal. The transform is the list of complex  $X$ 's.

Since it is an exact transformation, we are able to exactly inverse transform the DFT back to the original data. The transform is a set of amplitudes and phases of complex sine waves that form a continuous curve when added together. When the curve is evaluated at the signal sampling instants, it exactly reproduces the signal. The continuous curve, which is the sum of the  $N$  sine waves, is the continuous periodic band-limited curve the original  $x$ 's were sampled from. The inverse transform evaluates all these sine waves at the signal sample instants and adds them up. I want to also use the word reconstruction for the inverse transform operation, the IFFT.

The DFT is most economically computed using the FFT. I believe MATLAB's algorithm is able to deliver efficient results for all  $N$  values, not just powers of 2. The FFT will compute  $N$  Fourier coefficients from a sequence of  $N$  numbers. The  $X$ 's are numbered from 0 to  $N-1$ . If  $N$  is even,  $X_0$  is the DC or average value; it is real and contains one value.  $X_{N/2}$  also contains one

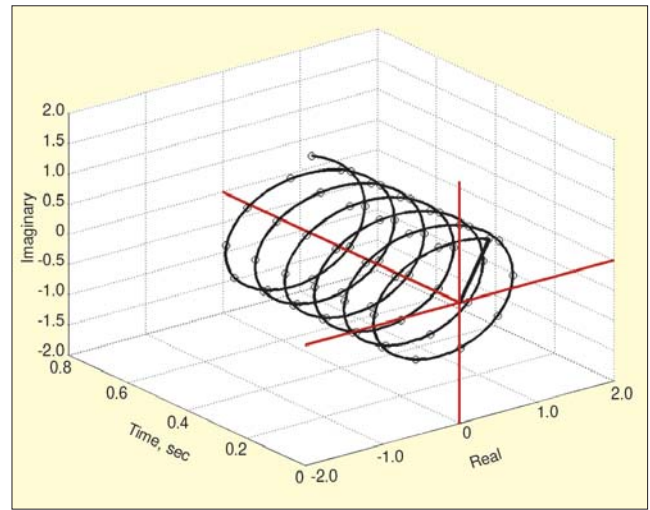


Figure 10. The DFT concept:  $X$  (thick) is a transform value; it has a length and is at an angle, its phase. It is the starting point of the complex sine wave spiral. This represents one of the discrete complex sinusoids in 3-D. The straight-line vector from the origin represents,  $X_k$ . The little circles on the spiral beginning at the tip of the straight line are the values for the sequence of  $n$  values.

real value; it is special. It is the content at the Nyquist frequency or half the sampling rate. It turns out to be the sum of the sequence with alternate signs reversed. In between these two are  $(N/2 - 1)$  unique complex values, each containing two values. The  $X$  values from  $X_{N/2+1}$  to  $X_{N-1}$  are not unique but are complex conjugates of the values from  $X_{N/2-1}$  down to  $X_1$ . The values symmetrical about  $X_{N/2}$  are complex conjugate pairs. Thus for the  $N$  values of the sequence, we get  $N$  unique values from the transform.

## Relation of DFT $X$ 's to Harmonic Content

The spectrum of a machinery vibration signal is a plot of content vs frequency. If you find time to look up Fourier series theory from any old book (25 years or so) you will find that they use  $a$ 's and  $b$ 's to do the analysis of a segment of a signal, assumed periodic. (However, the analysis will work almost no matter what you assume; it will make a periodic function of that segment.) Then they reconstruct the signal in terms of these  $a$ 's and  $b$ 's as follows. They say if  $x(t)$  is periodic with period  $T$ , it can be exactly synthesized from its Fourier series coefficients,  $(a_k, b_k)$ , as Equation 9:

$$x(t) = \frac{a_0}{2} \sum_{k=1}^{\infty} \left( a_k \cos \frac{2\pi kt}{T} + b_k \sin \frac{2\pi kt}{T} \right) \quad (9)$$

for  $k = 0, 1, 2, \dots$ ,

The Fourier series coefficients are given by Equations 9a and 9b:

$$\frac{a_k}{2} = \frac{1}{T} \int_0^T x(t) \cos \frac{2\pi kt}{T} dt \quad (9a)$$

$$\frac{b_k}{2} = \frac{1}{T} \int_0^T x(t) \sin \frac{2\pi kt}{T} dt \quad (9b)$$

The  $a_k$  and  $b_k$  are the harmonic content. The  $k$ th harmonic is given by Equation 10:

$$x_k(t) = a_k \cos \frac{2\pi kt}{T} + b_k \sin \frac{2\pi kt}{T} \quad (10)$$

Its frequency is  $k/T$ . We can also write the  $k$ th harmonic in terms of amplitude and phase as follows:

$$x_k(t) = A_k \cos \left( \frac{2\pi kt}{T} - \phi_k \right) \quad (10a)$$

Here  $A_k$  is the amplitude and  $\phi_k$  is the phase. The phase is the angle in radians to the first positive peak. The amplitude and phase are given by Equations 10b and 10c:

$$A_k = \sqrt{a_k^2 + b_k^2} \quad (10b)$$

$$\phi_k = \tan^{-1} \left( \frac{b_k}{a_k} \right) \quad (10c)$$

$A_k$  is the content or the amplitude of the  $k$ th harmonic or tone;  $\phi_k$  is its phase. This is the quantity shown on signal analyzers and data collectors. More manipulating leads us to the following. The  $X_k$ 's from  $k = 1 \dots (N-1)/2$  that we get from the DFT are related to the content as Equations 11, 11a, and 11b:

$$X_k = \frac{a_k}{2} - i \frac{b_k}{2} \quad (11)$$

$$A_k = 2|X_k| \quad (11a)$$

$$\phi_k = \tan^{-1} \left( \frac{\text{Re}(X_k)}{-\text{Im}(X_k)} \right) \quad (11b)$$

The amplitude and phase of the  $k$ th harmonic defined in Equation 10a are summarized in Equations 12 and 12a:

$$A_k = 2|X_k|, A_0 = X_0, A_{N/2} = X_{N/2}, \quad (12)$$

$$\phi_k = \tan^{-1} \left( \frac{\text{Re}(X_k)}{-\text{Im}(X_k)} \right) \quad (12a)$$

The DFT, which is most economically computed using the FFT, will compute  $N$  Fourier coefficients from a sequence of  $N$  numbers.  $X_0$  is the DC or average value; it is real and contains one value.  $X_{N/2}$  also contains one real value. In between these two are  $(N/2 - 1)$  unique complex values, each containing two values. The  $X$  values from  $X_{N/2+1}$  to  $X_{N-1}$  are not unique but are complex conjugates of the values from  $X_{N/2-1}$  down to  $X_1$ . The values symmetrical about  $X_{N/2}$  are complex conjugate pairs. Thus for the  $N$  values of the sequence, we get  $N$  unique values from the transform.

That is dry and tough. One more DFT concept and then we proceed, and it is 'shifted.' The values past  $X_{N/2}$  can be picked up as a group and lifted to the other side of zero so that now instead of having values on either side of the Nyquist value,  $X_{N/2}$ , be complex conjugate pairs, the values on either side of zero frequency or  $X_0$  are complex conjugate pairs. In this method of plotting a DFT we will have positive and negative frequencies. This is a common method of plotting a DFT. We are used to plotting a spectrum, which is content versus frequency from 0 to the Nyquist frequency, or  $f_s/2$ . Many people not in this business plot the  $X_k$ 's and use positive and negative frequencies. That is what I mean by shifted.

### DFT of a Window

Now let us look at the DFT magnitude of some windows; the DFT or the  $X_k$ 's, not the spectrum. I am going to make all these windows 1,024 points long, and I am going to sample them at 1,024 samples per second. Therefore, the time duration of each window will be one second; the  $\Delta f$ , which is one over the duration, will be 1 Hz. We will start with a boxcar window. The DFT is going to be digitized Fourier series of a boxcar window considered periodic and going on forever. It is a straight line with an amplitude of 1. It only has a DC component; it should have one line in its DFT with amplitude 1 at frequency equal to 0. Figure 11 shows this. In Figure 11a, we see the 1,024 ones as a horizontal line. In Figure 11b, we have the DFT magnitude and we see a 1 at 0 frequency. In Figure 11c, I show the shifted DFT magnitude with positive and negative frequencies. We still have a 1 at 0 frequency. In Figure 11d, I expand the plot to only slow values from -10 to 10 Hz, which shows a triangle with a base from -1 to 1 Hz and an apex of 1 at 0 frequency; we get this triangle because the plotter draws straight lines between the values. In Figure 11e, I use the stem plot feature that makes it clear we only have one value at zero frequency with an amplitude of 1.

Now let us look at the DFT magnitude of a Hanning window. The equation of the Hanning window is Equation 13:

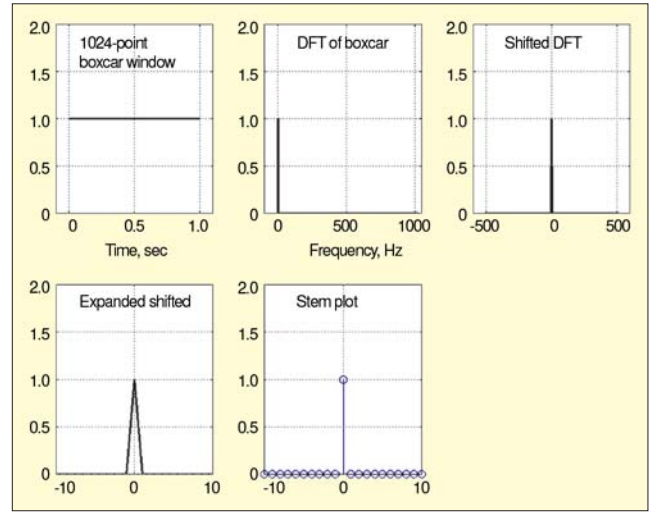


Figure 11. DFT of a 1,024-point boxcar window; only one line, DC.  $F_s = 1024$ , so  $T = 1$  sec;  $\Delta f = 1$  Hz.

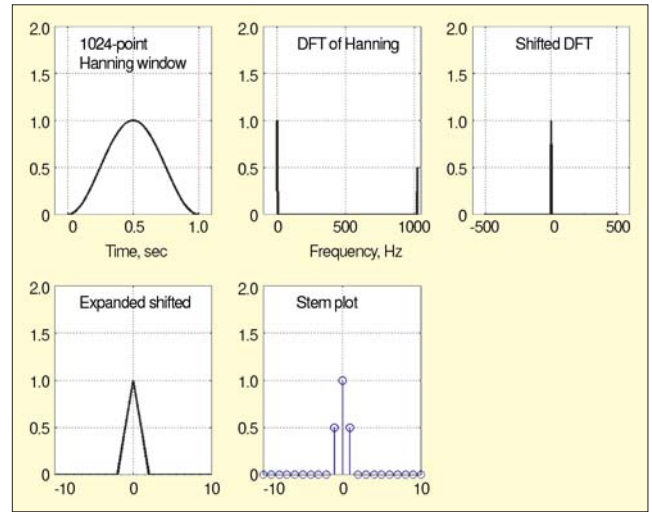


Figure 12. DFT 1,024 point Hanning window; only three lines;  $F_s = 1024$ , So  $T = 1$  sec;  $\Delta f = 1$  Hz.

$$w = \frac{1}{2} \left( 1 - \cos \frac{2\pi n}{N} \right) \quad (13)$$

This is a raised cosine with an average or DC value of 1/2. It has a period of our window length, 1 second, thus its  $\Delta f$  is 1 Hz. The DFT is going to be a digitized Fourier series of this 1 Hz raised cosine with an average value of 1/2 considered periodic and going on forever. Figure 12a shows its time history. In Figure 12b, we see its 1,024 value DFT. We see what appears to be a 1 at 0 frequency and a 1/2 at the 1024th value. In Figure 12c, we see the results of shifting the 513th to the 1,024th value over on the other side of 0. In Figure 12d we expand the plot to only slow values from -10 to 10 Hz, which shows a triangle with a base from -2 to +2 Hz and an apex of 1 at 0 frequency. We get this triangle, because the plotter draws straight lines between the values. In Figure 12e, I use the stem plot feature that makes it clear we only have three values; one at 0 frequency with an amplitude of 1, and two values of 1/2 at -1 and +1 Hz. By Equation 12, these two values of 1/2 would add to 1 when we convert this to a spectrum so the spectrum will have a 0 frequency or DC value of 1 and a 1 Hz value of 1. They both should be 1/2, but I normalized the DFT plot to make the largest value 1. The Hanning window has a conveniently simple DFT; three numbers. We will soon discuss that the convolution can be used to apply the window, and a convolution with just these three numbers made significant economic sense in the late 1960s and early '70s when the new signal analyzers became digital and started using the FFT. Most of the popular windows have simple DFTs for this reason.



## Effect of Adding Lots of Zeros

This may seem silly for a minute, but just for a minute. The DFT of a digitized time history returns  $N + 1$  frequency values when it receives  $N$  time history values equally spaced between zero and  $f_s$ . I will append nine times 1,024 zeros to the end of a 1,024 point window time history. Now this forces the DFT to analyze a periodic time history with the window, a nine times as long string of zeros, and the window again, and so on; the DFT will now calculate 10,241 frequency values. For the boxcar or uniform window, this is very different from our periodic boxcar that was just a horizontal line. Figure 13a, shows a time history for a 10,240-point boxcar window in which the first 1,024 values are ones. (You have to append a lot of zeros to see the window shape.) The time history is 10 seconds long. The sample rate is 1024 sample/second, so the first 1,024 points have a duration of 1 second. Figure 13b is a view of just the first 1,050 points of the time history to show that we have the same boxcar window. Figure 13c shows the magnitude of the DFT of the zero-padded boxcar 10-second time history. Now the plot appears to show the value 1 at 0 frequency and a 1 at the highest frequency. The shifted DFT magnitude in Figure 13d shows content near 0 frequency. To see what is going on near 0, I expanded the plot to just show the region from -5 to +5 Hz in Figure 13e. This is the amazing result of adding the zeros. Instead of getting the single line of Figure 11e, we now have a continuous curve with a main lobe 2 Hz wide between an array of side lobes 1 Hz wide. In Figure 11f, I have plotted a semi-log plot showing the log of the DFT values. Quite a complicated result. This is the kind of DFT I will use to show what the windows do.

I will pursue adding zeros in examples with the Hanning window shown in Figure 14. Our Hanning window in Figure 12 was a cosine with an amplitude of 1/2 and a mean value of 1/2 oscillating forever. In Figure 14a, we see the Hanning window with nine times as many zero values appended. Figure 14c shows the DFT magnitude and Figure 14d the shifted DFT magnitude, again with all of the content apparently clustered near 0 frequency. But reducing the frequency range from -5 to +5 Hz shows a main lobe 4 Hz wide with a slight ripple apparent. This ripple becomes apparent on the semi-log plot of Figure 14f. The main lobe is fatter, 4 Hz wide, and the ripple magnitude is considerably reduced; the highest lobe is about a 0.02. These side lobes are again 1 Hz wide.

Finally, in Figure 15, I did the same thing for the flat-top window to show the extremely wide main lobe, which we shall see enables the flat-top window to provide extreme amplitude accuracy. The reason it is called the flat-top is because the main lobe has a flat top. The side lobes are so small they do not show in this plot. Now we move on to the convolution theorem where we will use these ideas.

## Convolution Theorem

The only way I can analyze or explain window effects (that is, why the window affects leakage, the content in the adjacent bin, and reduces picket fence errors) is via the convolution theorem,<sup>9</sup> which is difficult. The convolution theorem for us is: the DFT of the term-by-term product of two digitized time signals (a digitized time signal is a long list of digitized acceleration or velocity values) is equal to the convolution of the DFTs of each of the signals. If we want the DFT of the product of two time signals, we can get it by a convolution of the two individual DFTs. And that is our situation exactly. We want to understand the DFT of a windowed signal, and we will do it by looking at the convolution of the DFT of unwindowed signal and the DFT of the window. So we have to use the convolution.

What is a convolution? It is horribly confusing, important, and widely used. It drives most students nuts and still confuses me every now and then. The equation is:

$$X(k) = \sum_{q=0}^{q=N-1} Y(q)W(k-q) \quad (14)$$

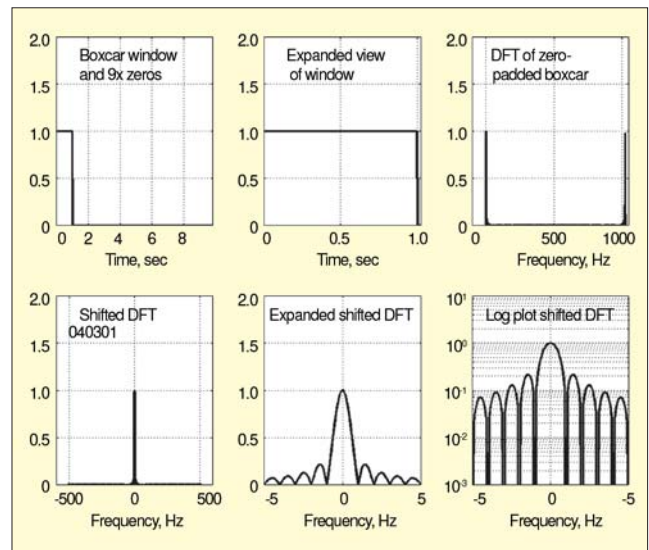


Figure 13. 1,024 point boxcar window, 9x zero-padded, expanded, DFT, shifted, expanded, semi-log plot.

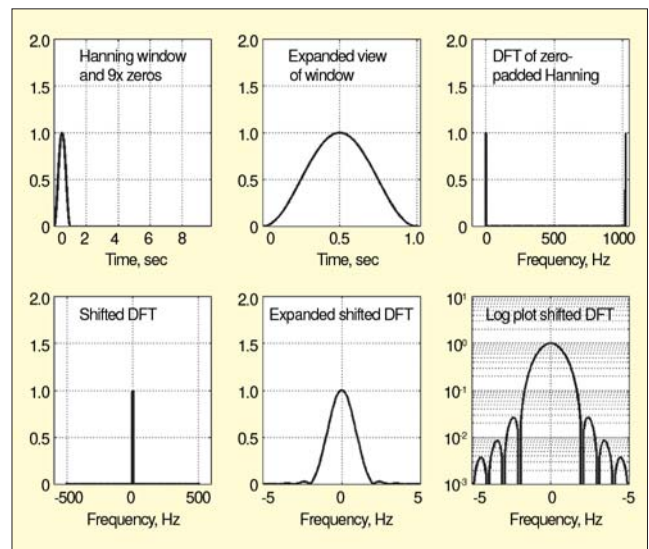


Figure 14. 1,024 point Hanning window, 9x zero-padded DFT, shifted, expanded, and semi-log plot.

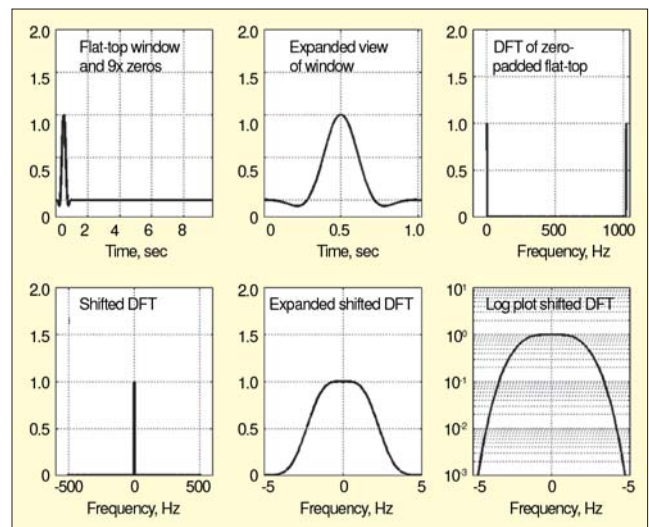


Figure 15. 1,024 point flat-top window, DFT, shifted, expanded, semi-log plot.

For our situation,  $Y(q)$  represents the DFT of our unwindowed data as collected, and  $W$  is the DFT of the window.  $X$  is the DFT of the windowed data that we can convert to the

spectrum we see in the signal analyzer. The equation says (and realize that  $X$ ,  $Y$  and  $W$  are lists of  $N$  numbers, like 1,024 numbers) that the value of the windowed DFT at frequency  $k$  is the sum of the products of the flipped, shifted, window DFT values centered at frequency  $k$  and the DFT of the data. We are going to do this in the explanation and I will draw it shortly, so I think this will become clear. But remember: sum of products of flipped, shifted, window DFT, and data DFT.

### Applying Convolution Theorem to Windows Analysis

How does the convolution theorem help us understand windows? We will view the window as a long digitized time signal, a string of mostly zeros with a non-zero digitized window shape in the center. Thus, the process of term-by-term multiplying the long window with the long signal conceptually forms our windowed segment that our analyzer/data collector FFTs. I need this concept to explain what we saw in Figures 4 through 9. Figures 16 and 17 illustrate this for the boxcar and Hanning windows.

In Figure 16, by speaking of a windowed segment, I mean considering the term-by-term product of the upper plot (pretend it is a long vibration signal, say 50,000 points) and the middle plot is an equally long list of zeros with a set of ones (perhaps 1,024) in the region where we want to select a segment of the signal for analysis. By multiplying these two graphs or data lists term by term, we end up with the third plot, a segment of the long signal. The third plot represents a chunk of data our signal analyzer would analyze. By speaking of a windowed segment, for now, I mean the product of the upper plot (the signal), and the middle plot of the long list of zeros with a set of ones in the middle. Similarly, Figure 17 shows the same concept, except this time we have a centered Hanning window flanked by a huge number of zeros. Now you can see why I needed the DFT of the window with the huge amount of zeros.

We know that the DFT of a signal returns spectrum values at a set of frequencies  $\Delta f$  apart. The  $\Delta f$  is  $1/T$ , where  $T$  is the time duration of the signal segment. If we have a signal that is stable and infinitely long (let  $f_e$  mean forever)  $T_{f_e}$  is infinitely long,  $N_{f_e}$  is infinite,  $\Delta f_{f_e}$  is infinitesimally small; its DFT has infinitely many lines and is exact for every tone or harmonic content the signal contains. This is an idea of the true spectrum concept: no picket fence or frequency error and no leakage. It has lines everywhere and is exact for every tone in that stable signal. That is the true spectrum we want, but we cannot have it because we cannot collect or analyze such a long signal.

We use the infinitely long zeros-window-zeros of Figures 16b and 17b to select a chunk  $T$  long of that stable, very long time history. The DFT of this windowed chunk will still have lines everywhere, because it had infinitely many time values. So it will also have lines at the frequencies that our signal analyzer can compute. To explain what the window will do to various kinds of signals analyzed on any real signal analyzer, we will convolve the highly detailed zero-padded window DFT with the DFT of a sine wave signal. We know the DFT of our  $T$  length signal analyzer spectrum is going to be a discrete vector with values for frequencies starting at 0, and at every  $1/T$ , on up to  $N/2$ .

And what does this 'convolved' mean? This means the value of the convolution at frequency  $k$ . Or: flip the window DFT left right, center it on one  $k$  value, term by term multiply, and add up the products. Summarizing: take the DFT of the window, which is symmetric if plotted with positive and negative frequencies – center it at some frequency  $f_1$  on the DFT of the signal – term-by-term multiply the two DFTs together; add up the products; and that is value of the convolution for  $f_1$ . Continue this for every frequency of interest.

Let us illustrate this with the examples of Figure 7, the case of a 30% of bin-centered signal. Figure 18 illustrates this for the boxcar window. The true signal is a tone at 16.3 Hz shown as a black line. In Figure 18a, the window DFT (blue) is centered at 13 Hz. The product of the DFT of the window with the DFT of the 16.3 Hz tone is short red line at 13 Hz. In Figure

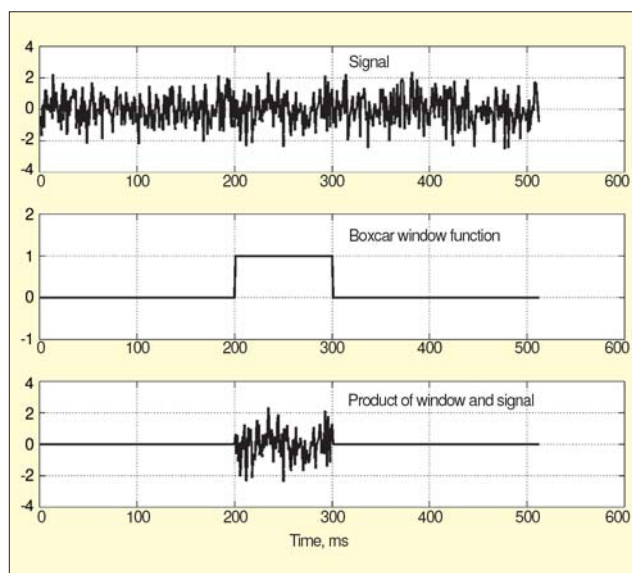


Figure 16. Boxcar window to select a segment of data from long time history.

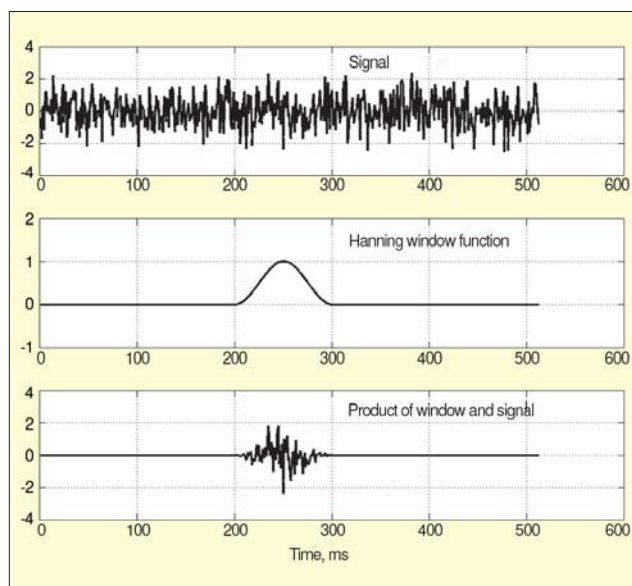


Figure 17. Hanning window to select a segment of data from long time history.

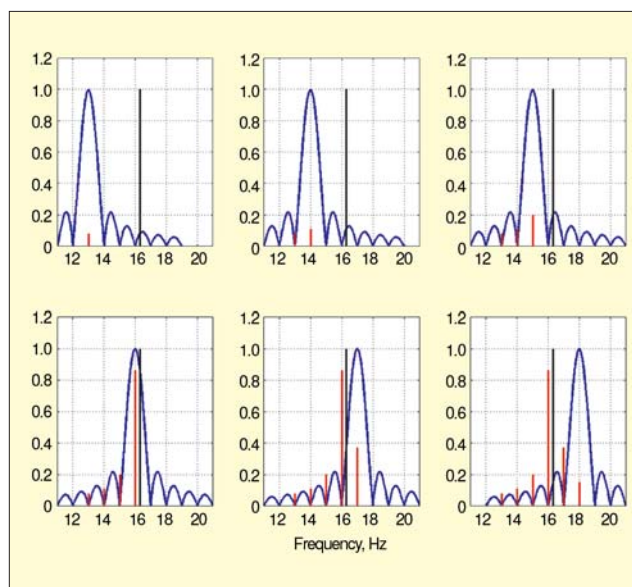


Figure 18. Convolution steps of boxcar window with 16.3 Hz tone.



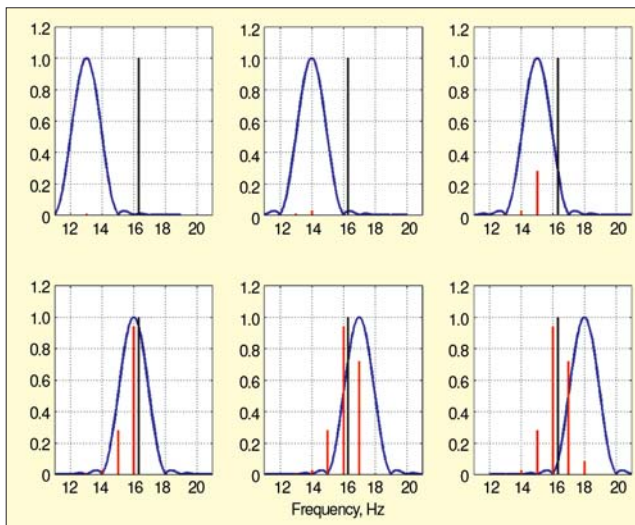


Figure 19. Convolution steps of Hanning window with 16.3-Hz tone; this shows why the several lines appear.

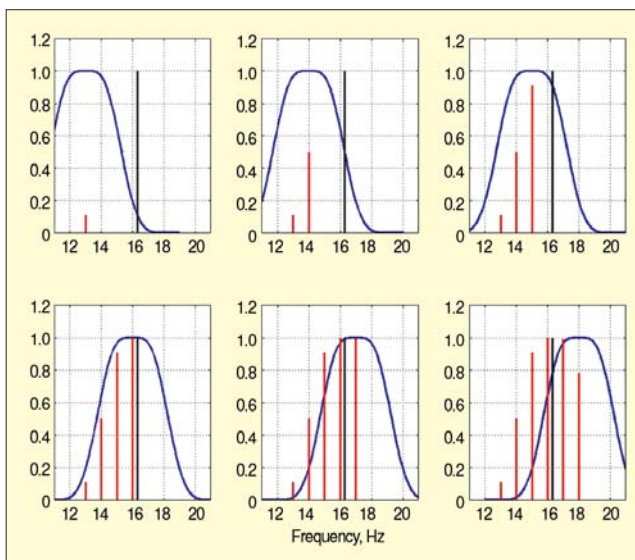


Figure 20. Convolution steps of flat-top window with 16.3-Hz tone; shows line build-up and accuracy due to the wide flat-top of the window DFT.

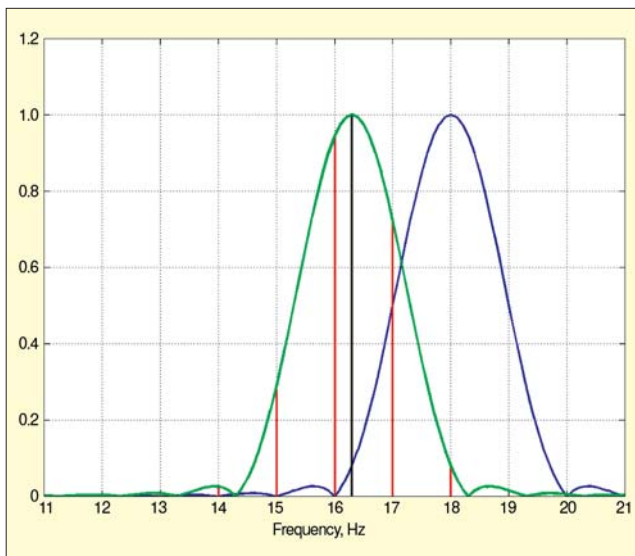


Figure 21. The green curve is the Hanning window zero-padded DFT centered on the 16.3-Hz tone. Where it crosses the bin center frequencies (11, 12, 13, 14, 15, 16, 17, 18, 19, 20, etc.) are the height of the lines it will induce in the DFT. These red lines are the same as indicated in Figure 19.

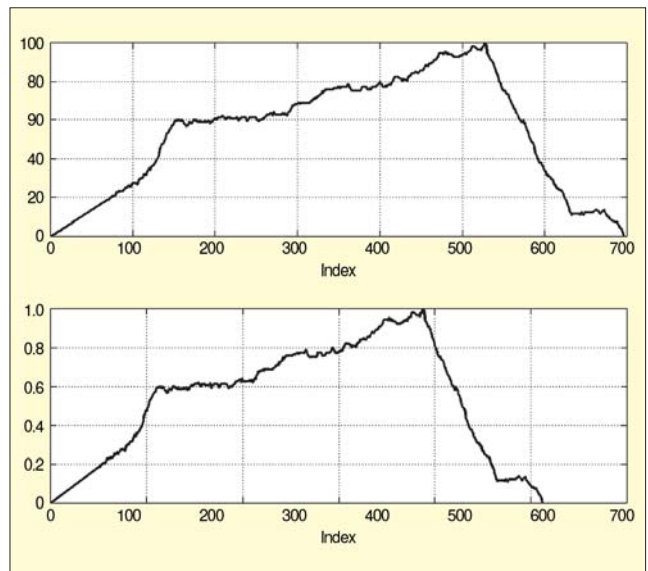


Figure 22. After editing, Diamond Head was 696 samples; it was resampled to 1,024.

18b, the window is centered at 14 Hz. The product of the DFT of the window with the DFT of the 16.3 Hz tone is the red line at 14 Hz. This red line height is equal to the height of the intersection of the blue window DFT with the black tone. In Figure 18c, the window is centered at 15 Hz. The six positions of the window DFT result in the six red lines shown on Figure 18f. This is the same result I obtained by taking the boxcar windowed FFT and shown in the stem plot of Figure 7a. The same example is illustrated for the Hanning and the flat-top windows in Figures 19 and 20. That is how the window works and why we saw the results of Figures 4 through 9.

### Main Lobe Significance

My next observation is also difficult. The convolution of the window DFT with the data DFT is the same as the convolution of the data DFT with the window DFT. By looking at Figures 19 and 20, notice that if we had a set of vertical grid lines at 12, 13, 14, 15, 16, 17, 18, 19, 20 Hz (which are the bin-centered frequencies,  $k/T$ ) and we centered the window DFT on the 16.3 Hz tone, the intersection of the window DFT with the grid lines would give us the same result that we came up with in the above paragraph by shifting the window to do the convolution. This is illustrated in Figure 21. The shape of the main lobe of the zero-padded window DFT sets the lines or bin values caused by any tone. The equation of the main lobe is well known for the standard windows. If you believe that a group of lines is caused by a tone, you can use the formulas from the appendix in Reference 12 to estimate the true frequency and amplitude. Especially if there is not too much going on in the little side lobes.

### Graphically Defined Windows

Since we have discussed how to estimate window effects, I want to present a procedure for inventing a window defined only graphically. I chose two shapes to illustrate the idea. One is Diamond Head, a well-known mountain crater on the Hawaiian island of Oahu, and the other is well known beagle comic character sleeping on his doghouse. In both cases, I scanned the images into the computer and then digitized them with digitizing software.<sup>11</sup> The digitizing software does not allow you to select the sampling rate or the number of samples, so these windows must be resampled. The Diamond Head window was originally sampled to 696 samples; this was resampled to 1024 samples by adding zeros to the DFT and inverse (IFFT) transforming it back to the time domain.<sup>10</sup> Figure 22 gives a plot of the original digitizing and the resampled version. Figure 23 shows the zero-padded version, its DFT, and both linear and semi-log plots of its expanded central region. We are adding



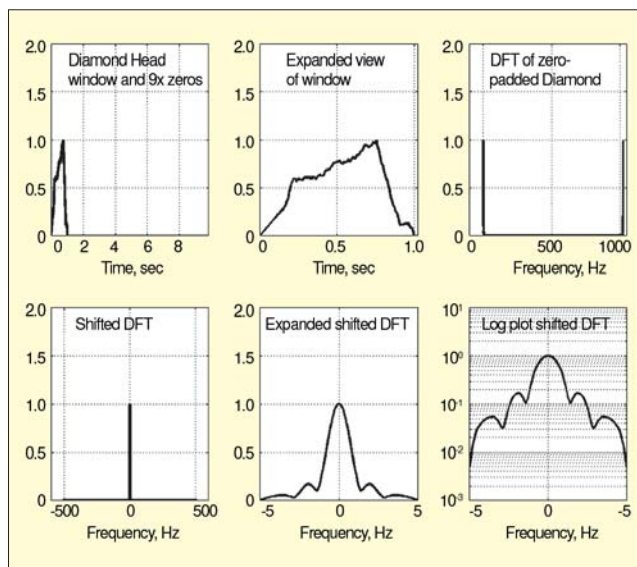


Figure 23. Diamond Head window, 9x zero-padded, DFT, shifted, linear, semi-log.

only high-frequency zeros and an even number for economical computing. The current Nyquist or center value is the highest frequency value and is real. We make the new Nyquist 0. Half of the old Nyquist is placed on each side as highest frequency non-zero values. (Add a complex zero.) Then place an odd number of complex zeros in the transform center for the desired values in reconstruction. Because of the  $1/N$  in MATLAB, the reconstructed window must be renormalized to make its maximum value 1. Adding zeros is a sensible practical procedure and certainly worked well here.

In the case of the beagle window, I digitized the drawing with 170 samples, and this was resampled by adding zeros to the DFT, and inverse transforming to bring the time plot to 1,024 samples. Figure 24 shows a plot of both the original and resampled versions of the window, and Figure 25 shows the zero-padded version, its DFT, and both linear and semi log plots of its expanded central region.

### Side Lobe Height

From our discussion of Figures 18, 19, and 20, we can see the problems of side lobe height. The windowed spectrum level at any of the frequencies will be the sum of the products of the window DFT and the data DFT. I illustrated the easy problem of the convolution with a single tone. Windows with high DFT side lobes or skirts allow noise and other tones or content to contaminate or increase the apparent level for the line at which the window is centered. Figures 26 and 27 show composite linear and semi-log plots of the side lobes of the six windows.

### Effects on Machinery Vibration Data

Finally, I made two busy plots, Figures 28, and 29, showing the eight windows applied to some pump vibration data. They provide an opportunity to examine spectral changes expected from different windows. The running speed is close to 20 Hz. There is a strong  $2\times$  component near 40 Hz, probably due to misalignment. The content at  $7\times$  is because the pump has seven vanes. The high skirt characteristic of the boxcar window shows well on Figure 29, the semi-log plot. The wide flat-top window smears the peaks and obscures detail. It is interesting that the beagle and the Diamond Head windows do as well as they do.

### Conclusion

I hope some of this adds some insight to your understanding of windows. Maybe it will give you enough background to tackle some of the difficult references. It certainly seems to me from References 1, 2, and 9, that many bright people have devoted years to developing windows. All of the figures and cal-

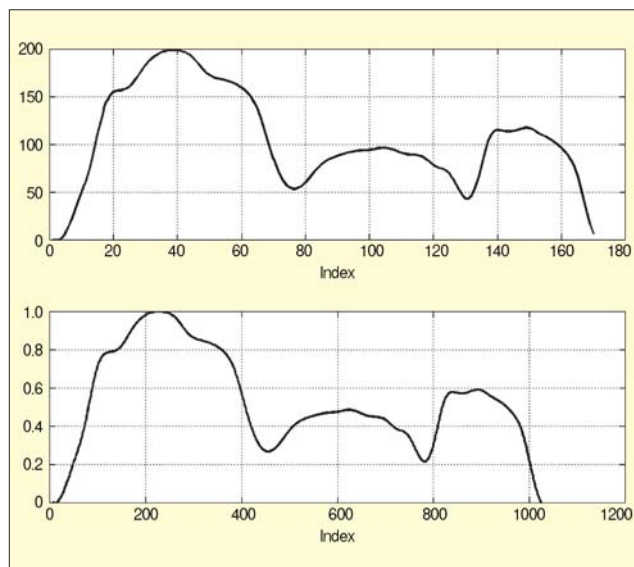


Figure 24. The digitized and edited, as well as resampled beagle window.

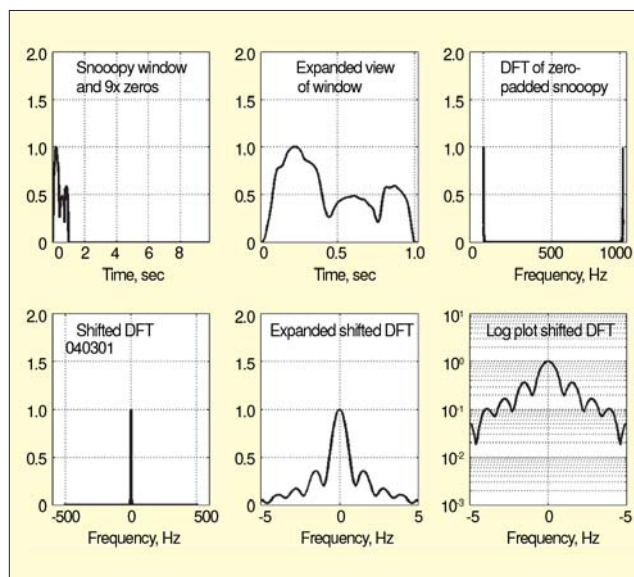


Figure 25. Beagle window, 9x zero-padded, DFT, shifted DFT, linear, semi-log plots.

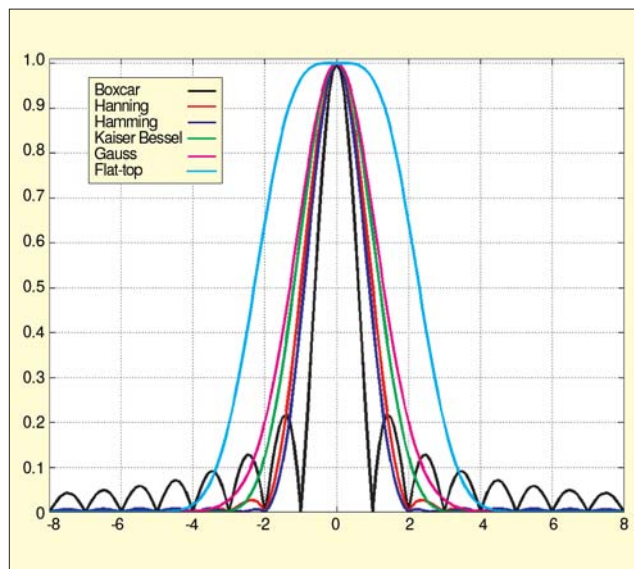


Figure 26. DFTs of common windows, shifted, expanded, zero-padded, linear plot.

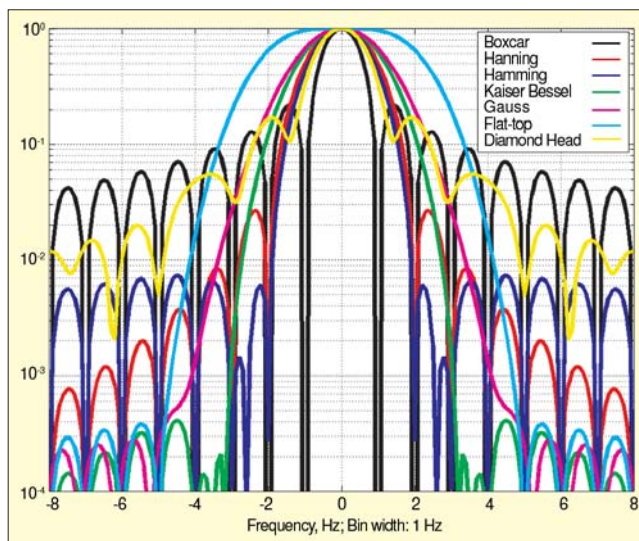


Figure 27. Semi-log plot of shifted, zero-padded, expanded windows showing skirts.

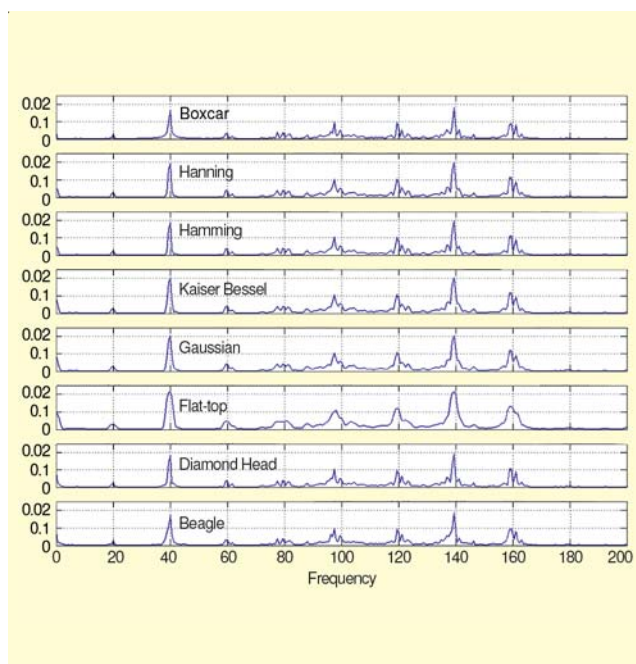


Figure 28. Eight linear spectra from the bearing of a large pump, each analyzed with the indicated window. Note the errors in peak amplitudes; the flat-top should be most accurate.

culations were done with MATLAB.<sup>12</sup> I kept copies of all the little programs used to draw the figures. If you would like copies of the programs to see how I have done them, let me know.

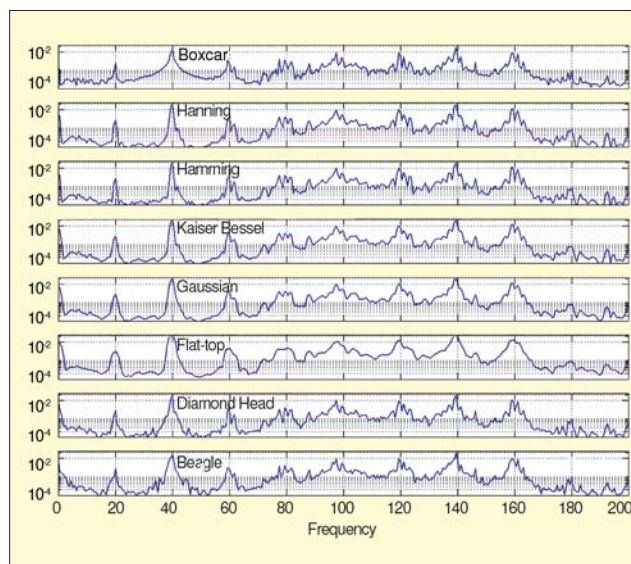


Figure 29. Eight semi-log spectra from the bearing of a large pump, each analyzed with the indicated window. High skirts of the boxcar window show well around 40 Hz. The windows have an effect to be sure.

## References

1. Harris, F. J., "On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proc. IEEE*, Vol. 66, No. 1, pp. 51-83, January 1978.
2. Nuttall, A. H., "Some Windows with Very Good Sidelobe Behavior," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, No. 1, pp. 84-91, February 1981.
3. Potter, R. W., "Compilation of Time Windows and Line Shapes for Fourier Analysis," Handout notes from an HP seminar circa 1978.
4. Gade, S., and Herlufsen, H., "Use of Weighting Functions in DFT/FFT Analysis (Part I)," *Brüel Kjør Technical Review*, No. 3, pp. 1-28, 1987.
5. Gade, S., and Herlufsen, H., "Use of Weighting Functions in DFT/FFT Analysis (Part II)," *Brüel & Kjør Technical Review*, No. 4, pp. 1-35, 1987.
6. Tran, T., Dahl, M., Claesson, I., and Lago, T., "High Accuracy Windows for Today's 24 Bit ADCs," IMAC 22, Session 29, Signal Processing, Soc. for Experimental Mechs., [www.sem.org](http://www.sem.org).
7. Harris, F. J., *Trigonometric Transforms, a Unique Introduction to the FFT*, Spectral Dynamics, Inc., San Marcos, CA., October 1977.
8. Gaberson, H. A., "The DFT and the FFT as a Discrete Fourier Series of Sampled Data," MFPT Advanced Signal Analysis Course Notes, Section 2, January 2003.
9. McConnell, K. G., *Vibration Testing: Theory and Practice*, John Wiley & Sons, Inc., pp. 266-278, 1995. (Most digital signal processing books cover the convolution theorem, but this one is more directed to our area.)
10. Gaberson, H. A., "Using the FFT for Filtering, Transient Details, and Resampling," *Proc. National Technical Training Symposium and 27th Annual Meeting*, Vibration Institute, Willowbrook, IL, pp. 127-136, July 2003.
11. UN-SCAN-IT, Silk Scientific Inc., Orem, UT.
12. MATLAB® for Windows, Version 5.3, *High-Performance Numeric Computation and Visualization Software*, The MathWorks, Inc., Natick, MA, 1999.

SV

The author can be contacted at: [hgaberson@att.net](mailto:hgaberson@att.net).