**SIA Software
Company, Inc.**

The "Mystery" of Data Windows

Data Windows, also called Window Functions, are among the most confusing aspects of Fast Fourier Transform (FFT) analysis. This is due in part to the fact that there are many uses for these functions other than FFT analysis. Text books tend to be very through on the subject — providing *much* more information than is needed to understand the basics — and so a simple, application-specific explanation (i.e., a straight answer) can be a difficult thing to find.

If you are not interested in the mathematics or theory of data windowing and are just looking for a "rule of thumb" for selecting a data window (or not), we can cut to the chase: *Do* use a data window. As for which type, the *Hanning* and *Hamming* windows may well be the most common data window types used for audio applications. In our own experience, we have found that either of these generally work quite well for most measurement applications using SIA-Smaart Pro.

If you *are* interested in learning a little more about the "nuts and bolts" of data window functions, lets start at the beginning. Suppose that you have some type of audio signal such as music, speech an impulse response, or a simple sin wave. We will assume that we are talking about a well-behaved, regular-old audio signal (and let mathematicians worry about esoteric cases). Let's say that we have "digitized" the signal into a time-domain audio data stream and that our data stream is stored in a Windows audio "wave" (*.wav) file.

To convert the wave file to the frequency domain you intend to use FFT mathematics. You select an FFT size based on the desired *frequency response* and the rate at which you're data changes. As an example, we'll use a 256-point FFT with no overlap. This means that you are going to divide your wav file into some number (depending on the length of your wave file) of 256-sample "slices," with each section beginning where the previous one ended as in Figure #1 below.
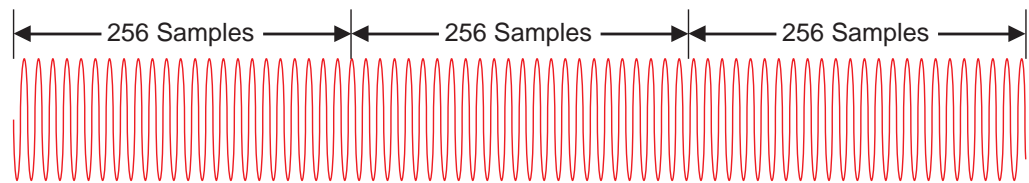

Figure #1: A time-domain data series divided into (256-point) FFT-size "slices."

You can now take the FFT of each slice. But notice that at the ends of each slice, the wave file has been abruptly chopped off. The point at which each slice begins and ends is arbitrarily determined by the FFT frame size so there is no guarantee that the beginning and end values will be zero — in fact, it's not very likely.

This gives rise to an oscillatory behavior in the Fourier Transform results called the *Gibbs Phenomenon*. The mechanics of how and why this happens lie in the realm of of what most of us might call "Big Math," but the effect is easy to see. As a result of having abrupt truncations at the ends of the time domain data series, the spectrum of the FFT will include unwanted "side lobes" — which brings us to data windows. ***Data windows help to reduce "truncation errors," arising from the arbitrary segmentation a data stream into FFT frames, by reducing the amplitude of the time-domain data nearest to the beginning and end of each frame.***

Let's look at an example: If we take the first 256-point section of the sin wave in Figure #1, we can see how the ends are just left hanging. This section happens to begin with an amplitude value of zero (the next one won't) but it ends at a point in the cycle of the waveform that is determined arbitrarily by the FFT size.
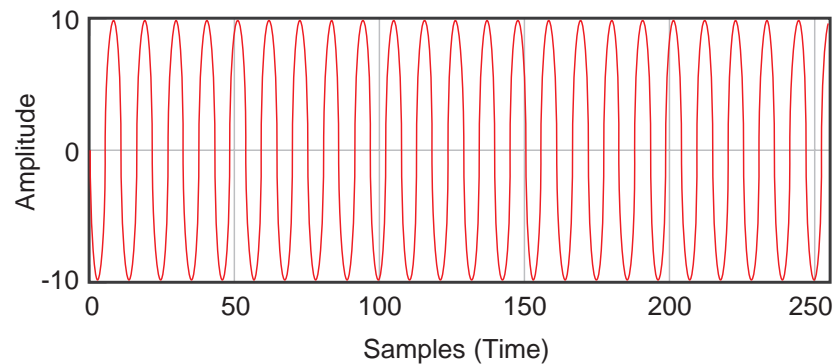

Figure #2: The first 256-point section of the sin wave shown in Figure #1 in a "rectangular" window (i.e., no data window function has been applied.

Simply taking the FFT of this data (without using a data window) yields the frequency-domain representation shown below.
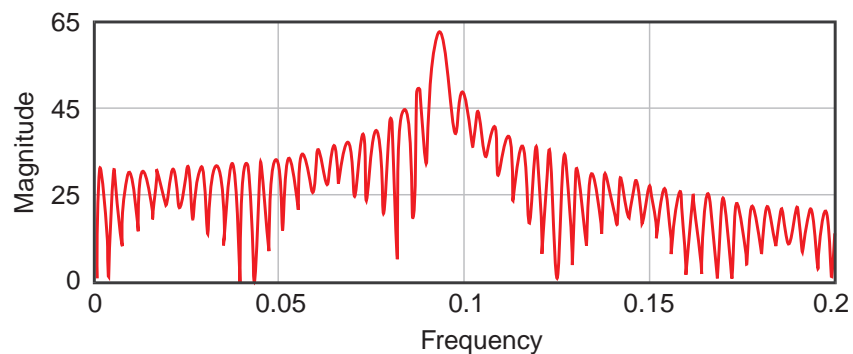

Figure #3: The FFT of the time-domain data shown in Figure #2.

Since the sin wave consists of only one frequency, what we would like to see here is a single "spike" at the sin wave's frequency. However, as a result of the Gibbs Phenomenon, the FFT data displays substantial "side lobes" on each side of the center peak. We can use one of a number of "data windows" to help correct for this. A data window is just a mathematical function used to reduce the amplitude of data at each edge of the time domain "slice" prior to calculating an FFT.

Figure #4 below shows the time-domain data from Figure #2 with a Hamming data window applied. The data window function has multiplied the amplitude value of each sample in the frame by a number between zero and one — generating multipliers near zero for the earliest samples in the frame, progressively increasing with each sample until it reached one at the center, then tapering off (symetrically) toward zero again at the end. Note that the window function is scaled to unity (i.e., 1) so the original audio data in the middle of our slice is not affected.
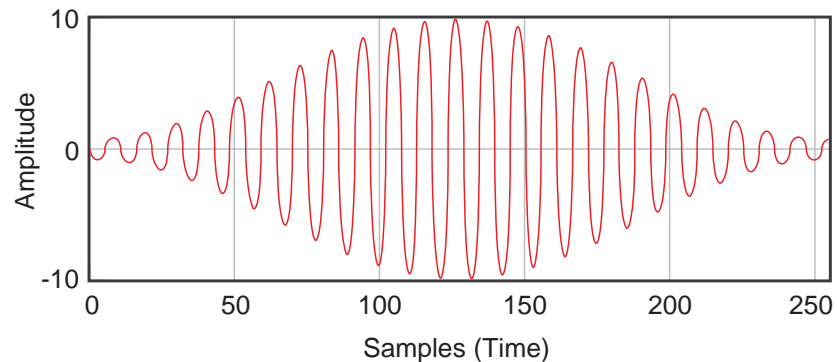

Figure #4: *A Hamming data window applied to the time-domain data from Figure #2.*

Once the data window has been applied, we can recalculate the FFT. Figure #5 below shows the result. Notice that the side lobes that were so obvious in Figure #3 have been substantially reduced.
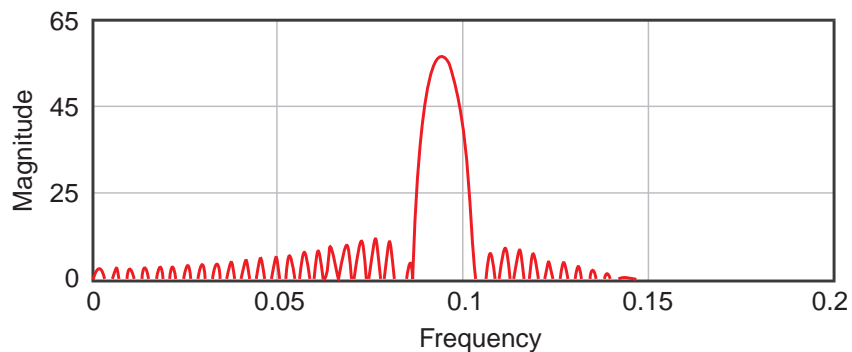

Figure #5: *The FFT of the "windowed" time-domain data shown in Figure #4.*

It should be noted however, that there is some "cost" involved. Generally speaking, the more you drive down the side lobes, the wider the center lobe becomes. You can see in Figure #5 that the center peak has, in fact, become "wider" (compared to the un-windowed FFT in Figure #3) as a result of the data windowing.

That brings us to the question of "What exactly is the difference between the various different window types?" The short answer is the "shape" of the window. The actual window functions are mathematical formulas but by plotting these formulas on a graph, we can see some of the similarities and differences. Figure #6 on the next page shows a plot of all of the window types that are included in SIA-Smaart Pro.
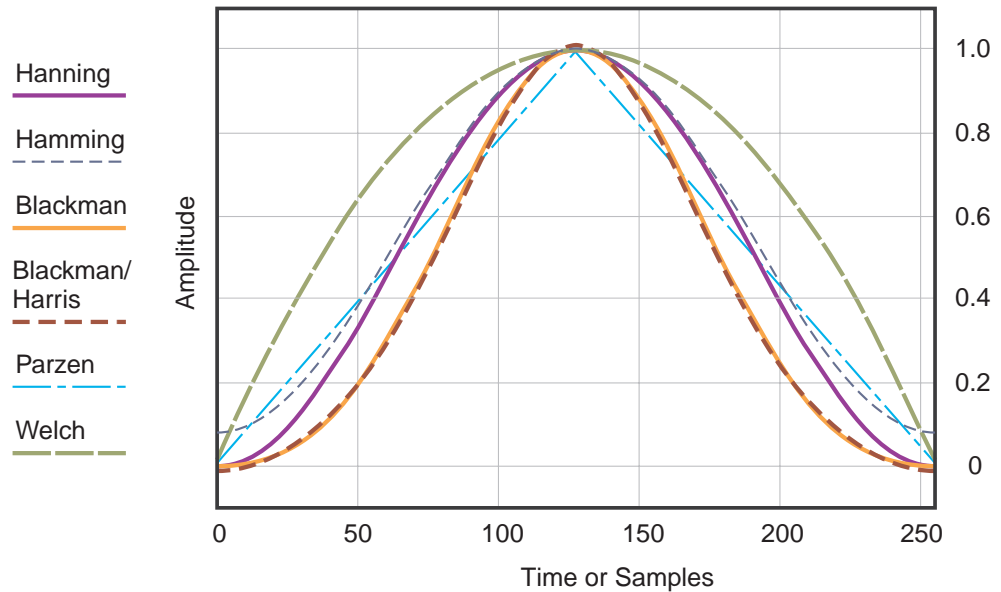
Figure #6: *The Data Window types built into SIA-Smaart Pro.*
Please note that data windows act equally on positive and negative
amplitudes values as shown in Figure #4 on the previous page

Changing the shape of the window affects that relationship between the reduction in the magnitude of the side lobes and the increase in the width of the center lobe. The next question often asked is, "OK, so which data window should I actually use?" The answer is that for most audio data, the "gentlest" windows, Hanning, Hamming and Blackman will generally yield the best end result.

A Publication of SIA Software Company, Inc.
One Main Street
Whitinsville, MA 01588
USA

telephone: 508-234-6158
e-mail: support@siasoft.com
www.siasoft.com