Making bigger waves!

I've been using my Iyonix (and before that my RiscPC) for many years to work with audio. Until relatively recently this was mainly for editing and generating 'CD Audio' files. Hence I tended to use the default 'data' format which is understood by CDBurn / DVDBurn. However there have been some developments over the last year. The most sigificant was that I have started to use Linux/ROX for some audio purposes. In addition some audio sources – like 'Freeview' and DVD Video – use a sampling rates which differ from the 44.1ksamples/sec that is the CD Audio standard. For those reasons I decided it would be convenient to change over and base my work with audio to files using the 'Wave' format.

During January I also acquired a new toy – a Tascam HD-P2 audio recorder. This is a superb recorder that makes recordings onto Compact Flash (CF) memory cards. It can record at sampling rates of 44.1/48/88.2/96/176.4/192 ksamples/sec, and can also record 24 bits per sample as well as a mere 16!

The CF cards can be accessed using a card reader and either my Iyonix or my Linux/ROX boxes. But the point I'd like to focus on here is that the recordings are made into files that are in the 'Broadcast Wave Format' (BWF) which is a form of Wave file with some extensions in its header designed to provide more metadata useful for professionals in the recording and broadcast industries. Hence in addition to moving to WAV file formats I also needed to ensure that I had a set of RO applications that could cope with sampling rates up to 192 ksample/sec. I may say more about the BWF, 24 bit, and how to convert these into the more common WAV in an article in the future. Here I want to present and explain a set of new versions of applications which can be used on an Iyonix to work on the common types of 16 bit WAV files.

I have provided five applications for Jim to distribute with this article. One is a new application I have called !WAV_Cleaner. This can be used to 'clean out' various 'extras' that may be in the header of a Wave file. Most of the Wave files I encounter use a standard 44-byte header that just specifies the basic properties like the sample rate. But some Wave headers contain additional information 'chunks'. !WAV_Cleaner can be used to make a fresh copy of the Wave file that preserved the data payload but removes these extras. This makes it easier to read and process the files with other applications.

The other four applications are 'upgraded' versions of !WAV_FFTScan, !WAV_Gen, !WAV_Reader, and !WAV_Stats. They essentially all work in much the same way as previous apps in the Track/WAV series. But there are a few useful differences. The change they share is that they all now work with LPCM 16 bit *and* 24 bit WAV files with sample rates up to 192ksample/sec. In addition I have added one new feature to !WAV_FFTScan...

A while ago Roger (aka Perdix) asked on the emailing list why my FFT programs did not use one of the 'better' methods for what is called 'windowing' or 'apodisation'? He had spotted that I had been using a fairly simple form of windowing based on a triangular window. (I'll explain this a bit more later.) However there are various alternatives which can give more useful spectra. Now at the time I was quite happy with the simple approach as it gave results that were good enough for what I was doing, and no-one had said they wanted anything better. But to check, I asked on the email list if anyone else was interested in improving this aspect of my FFT applications. No-one responded, so I let the matter settle.

However when I got the Tascam I started to do a series of calibration tests on it to see how well it worked, and promptly ran into the limitations of my simple windowing approach. This can be seen by looking at Figure 1

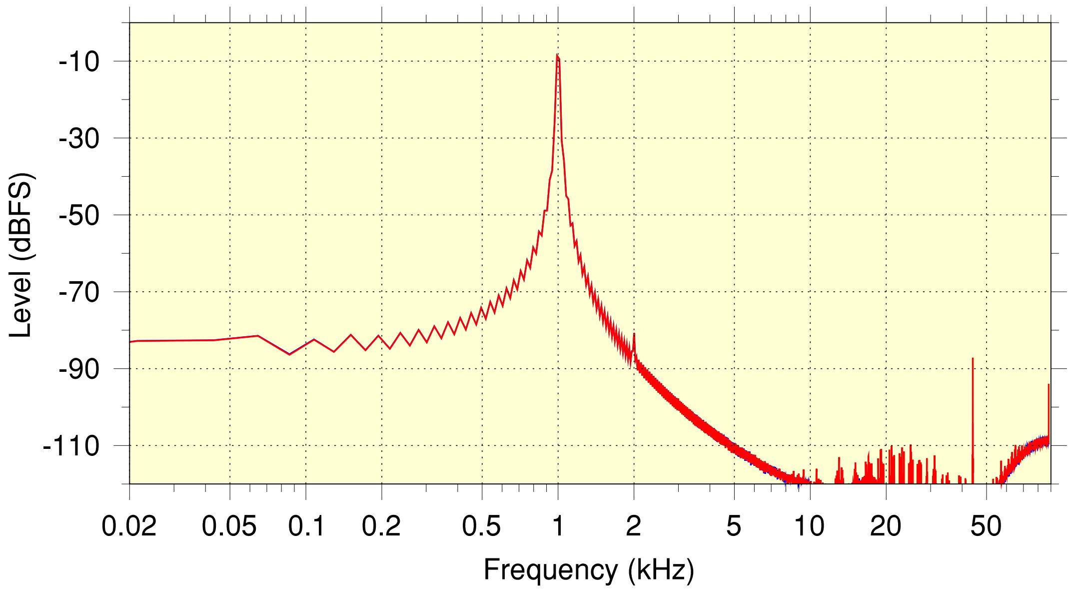

Figure 1 Triangular Window results

This shows the spectum I got for a recording made at 176.4ksample/sec for an input 1kHz sinusoid. Nominally the input was an almost pure sinewave. However the input was of a higher amplitude than the Tascam was designed to accept, and was therefore into its 'headroom' levels. Looking catefully at Figure 1 you may just be able to see a slight spike at 2 kHz, sitting on thr slope down from the 1kHz peak. Now, in principle, the distortion level should have been of the order of 0·001% which corresponds to -100dB relative to the peak. Yet here there seems to be a slight hint of an unwanted component that was only around 70dB below the 1kHz signal. The problem was that it was quite hard to decide if this was real, and what its actual level was if the slope from the 1kHz were removed. To know more I had to reduce the sloping sides produced by the 1 kHz tone.

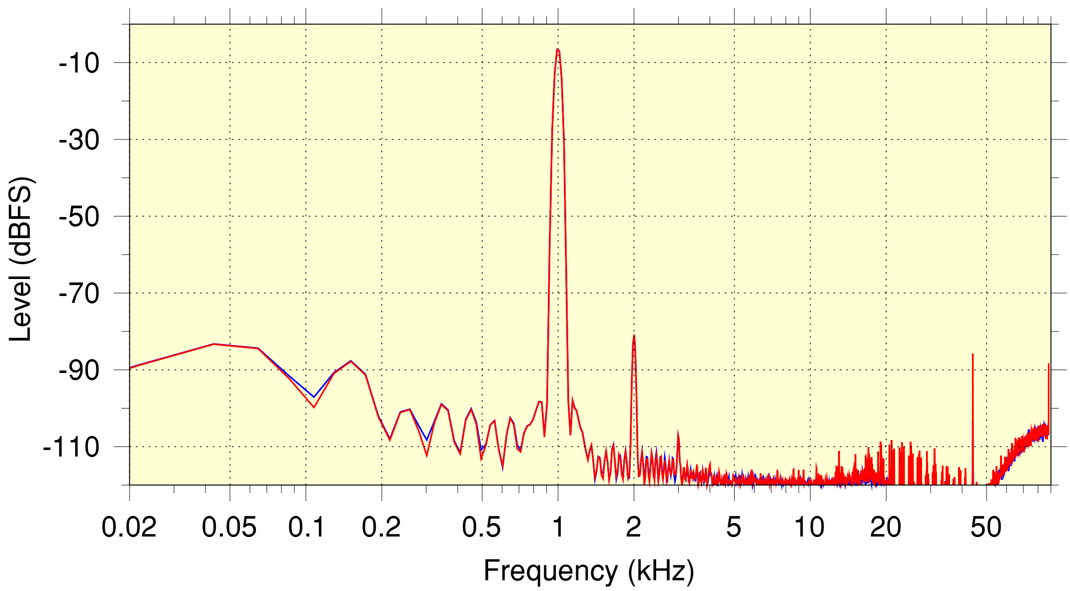

Figure 2 BH Window results for the same input

Figure 2 shows the spectrum I got if I changed the windowing to use the 'Blackman-Harris' function which Roger suggested. You can see that the change is quite dramatic! Now the 1kHz peak has essentially been stripped of its gradually sloping sides and the unwanted distortion componet at 1 kHz is now clearly revealed. Now that I could see what was happening in the spectra I was able to continue to experiment and test, and this confirmed that the distortion in the Tascam was, indeed, as low as around 0·001% provided I ensured the input level it was given wasn't greater than around 0·775Volts. The recorder would record OK for higher input levels, but the distortion tended to rise. The reason seems to be that there is a buffer amplifier and protection circuit in the Tascam before its input gain control, so winding down the gain control doesn't prevent over-large input signals from stressing this buffer and producing excess distortion. The solution in practice is to RTFM and note you are told the max normal input is 0.775Volts!

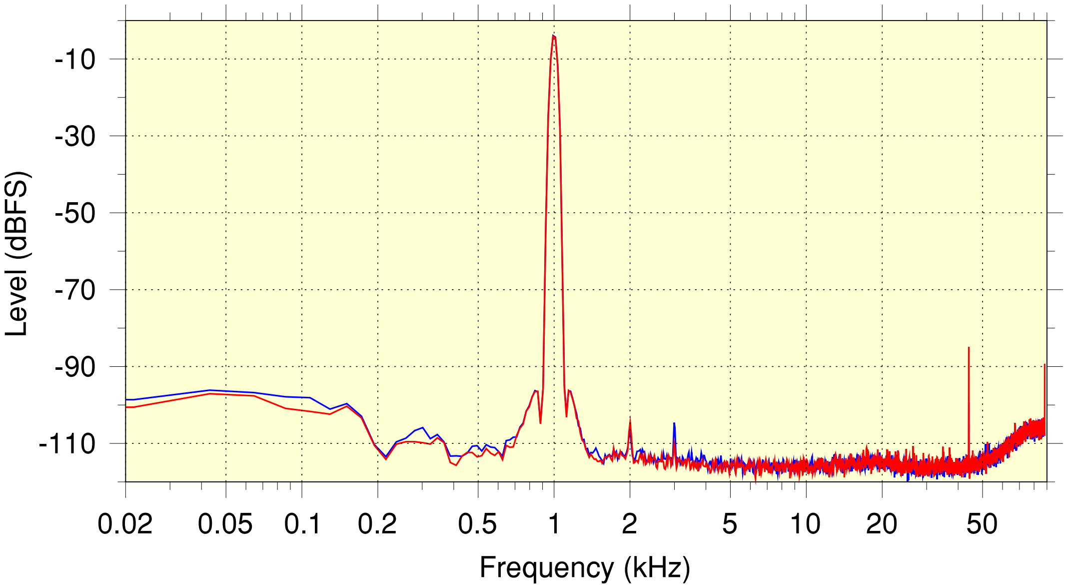

Figure 3 BH window again, but input at a sensible level!

Just to confirm this, Figure 3 shows what you get when recording with the input to the tascam recorder at a more sensible level. You can see that now the distortion is minimal.

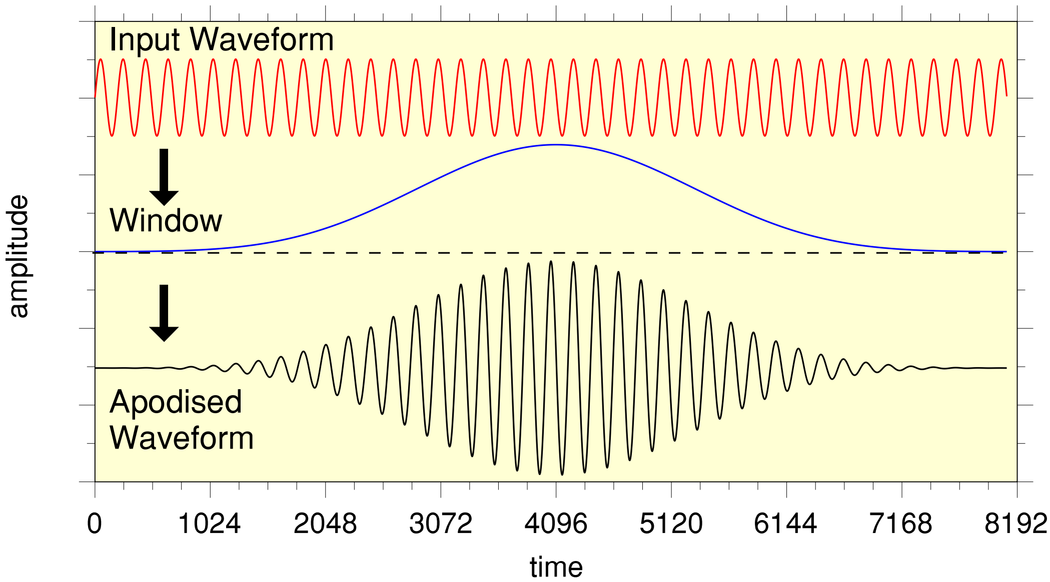

Figure 4 Applying the Blackman-Harris Window

Having seen the general effect of Windowing Figure 4 illustrates how it is applied. This shows three waveforms. The upper waveform is a simple example of an input – in this case a sinusoid – which we wish to give to the FFT process. The middle waveform is the shape of the Blackman-Harris windowing function. We then multiply the two shapes together to obtain the windowed waveform shown at the bottom of Figure 4. This has the effect of redistributing the signal energy around the spectrum.

You might expect that an input which consisted of a single sinusoid should always give an output spectrum that showed power at only one frequency, and zero at all other frequencies. However in practice the FFT is only given a waveform that is limited in two significant ways. Firstly it if of finite duration. Secondly, it consists of a series on discrete data points (samples). Hence it isn't actually a pure sinusoid which would – by the definition of a sinusoid – be continuous and extend over an infinite time period! So in practice the FFT tends to output a spectrum where the power is spread out in some way. Using a window function alters this, and can give more useful results. However the 'best' choice of window shape depends both on the properties of the input waveform and what we are trying to learn about it.

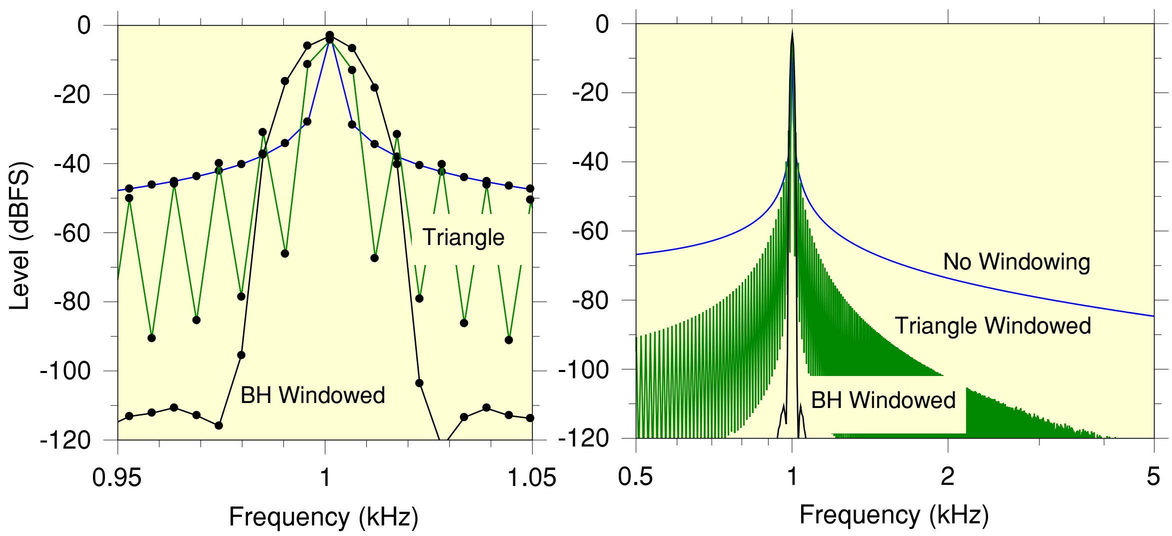

Figure 5 Effects of Window function.

To illustrate the effects of choosing various Window functions for the apodisation I used WAV_FFTScan to process an input that consisted of a 1001 Hz sinusoid, and chose each of the apodisation options in turn. Each of the graphs in Figure 5 therefore has three lines, one for 'no apodisation', one for 'triangular apodisation', and one for 'BH apodsation'. Looking at these you can see that – in the graph covering the 0.5 - 5 kHz range – BH gives the greatest reduction in the 'wings' or 'slopes', so is the best choice if you wanted to see any other components produced by distortion that might appear a few kHz away from the 1001Hz input.

However the 'zoomed in' graph shows the price this has to pay. Here the plot with no windowing comes to a sharp point at about 1001Hz. But the BH windowed spectrum shows a broadly rounded top near 1001 Hz, although it swiftly falls away to well below -100dB when we move further away from 1001 Hz. Although not obvious on these plots the BH (and to a lesser extent triangular) version also slightly alters the peak level found due to way signal power is spread around the graph.

Overall this means the 'best' choice depends on the nature of the signal being FFT processed, and what you wish to learn from the resulting specrum. For an essentially pure sinusoid that happens to have an integer number of cycles in the data processed then no apodisation tends to be best. For tasks like assessing distortion or seeing the spectrum when there are a few frequency compoents, BH tends to be optimum. And triangular works well if you have a signal with a range of frequency components against a general 'noise like' background. So "yer pays yer money and yer takes yer choice". When in doubt, choose BH and see if the results look OK. But for something like accurately spotting the frequency and size of a near-sinusoid then triangular works well.

Jim Lesurf
1600 Words
3rd Feb 2010