BY IAN BELL

Fast Fourier Transform (FFT) and LTspice

FOR the past couple of issues we have covered the topic of Total Harmonic Distortion (THD) in response to a question from frequent *EPE Chat Zone* contributor **741**. We have looked at the definition of THD and related characteristics, and the theory behind these. We also looked at ways to measure THD, which was part of 741's original question.

As readers of the previous THD articles will have seen, the frequency spectrum of a signal is important to the understanding of THD, and can be used directly to calculate THD. Spectra are an important tool for studying signals and have many uses, not just the calculation of THD.

The spectra which we drew to illustrate our THD discussion were straightforward and clear-cut. However, when using real spectrum analysers and simulators to observe spectra, the situation is rarely so simple. This leads us to the second part of 741's question about measuring and observing a signal's spectrum using a Fast Fourier Transform (FFT).

On LTspice I placed an ideal sine generator, then chose View/FFT. I noticed the wide 'skirts' leading up to the peak at the sine frequency. I wondered what determines the sharpness of the peak.

Spicing it up

Before getting to the specifics of 741's question we will take a brief look at the SPICE simulation, for the benefit of readers who may not be very familiar with it. We will then look at the general concepts behind calculating a spectrum, because some understanding of this will make the settings in the LTspice dialog (which controls the display of an FFT) much more meaningful and easier to use.

This will also help with understanding the errors and approximations which can occur in a spectrum, which are at the heart of 741's question. Next month, we will look in more detail at the specifics of using LTspice to display spectra.

Spice is an acronym for Simulation Program with Integrated Circuit Emphasis. It was originally developed in the early 1970s at the University of California, Berkeley (see **http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE/ and http://embedded.eecs.berkeley.edu/pubs/downloads/spice/index.htm**) and is still available from there. Spice is now a *de facto* industrial standard for computer-aided electronic circuit analysis with many commercial versions based on the original work from Berkeley.

Although Spice was initially developed for analysing ICs, it can be applied to any electrical network (of resistors, capacitors, transistors etc). It was originally an analogue circuit simulator, but modern versions allow logic gates and more complex digital functions to be included, allowing digital and mixed-signal (analogue and digital) circuits to be simulated. However, Spice would not normally be used for large fully digital circuits.

Spice simulation

To simulate a circuit with Spice you have to first draw the schematic using the simulator's schematic capture software. Alternatively, a text description of the circuit (a netlist) could be used, as was the case in early versions of Spice.

However, your circuit alone is not enough for a simulation – you have to define what happens at its inputs (stimulus signals) and outputs (eg, appropriate loads). You also need to specify what kind of simulation/analysis to perform and provide the necessary parameters (eg, how much time to spend simulating). We will be looking at the use of transient simulation and the FFT analysis mentioned by 741.

Professional Electronics Computer Aided Design (ECAD) software can be very expensive, but fortunately for the amateur on a tight budget there are a number of free Spice simulators available. The LTspice simulator to which 741 refers is one of these. It can be downloaded in its full form from the Linear Technology website: **www.linear.com/designtools/software/**.

LTspice is optimised for simulating switched-mode power supplies, which form an important part of Linear Technology's product line. However, it also does a fine job with 'ordinary' Spice simulation tasks. Linear Technology produced the simulator because many other Spice simulators struggled with

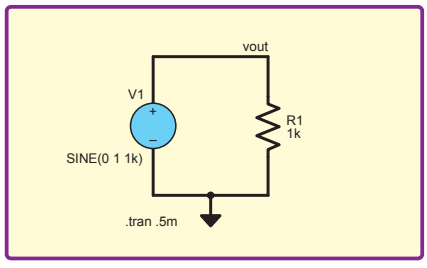

Fig.1. A very simple circuit can be used to get a spectrum (FFT) analysis in Spice to investigate the effect of various simulation and analysis parameters

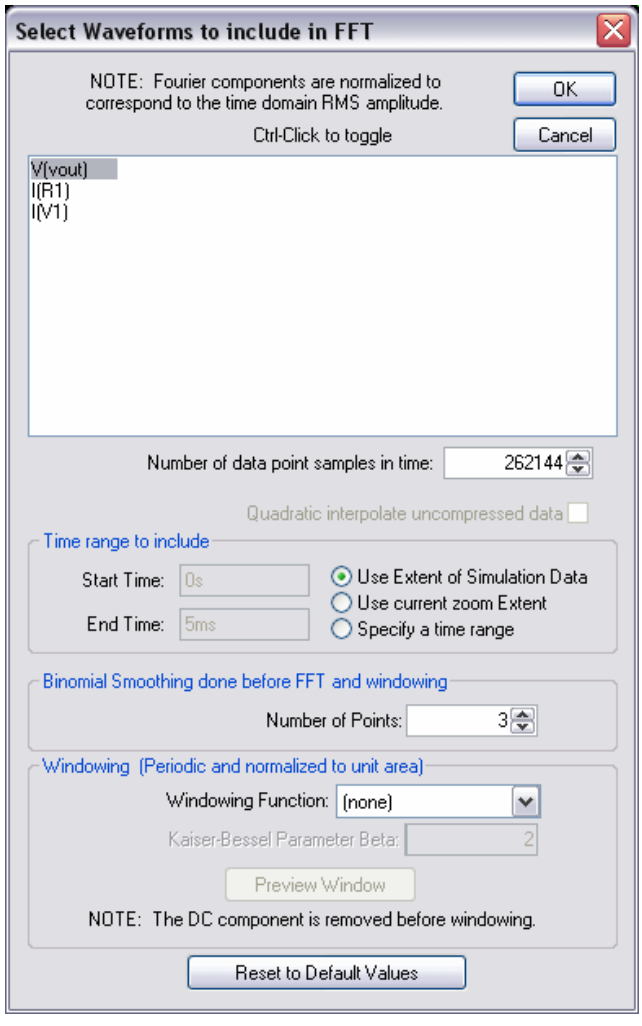

Fig.2. The FFT dialog in LTspice; understanding the parameters set in this dialogue is necessary to get the best out of spectral analysis

switch-mode circuits, due to the complexity of the waveforms involved. It comes with a set of models of Linear Technology devices, including op amps (over 300 of them) and regulators, but other 'standard' Spice models will work with it if you have them available.

Easy to use

LTspice has an easy-to-use user interface and it is very straightforward to change something in a circuit and run the simulation again to see what difference it makes. Linear Technology's philosophy in designing the simulator was to allow a lot of flexibility, while providing sufficient warning messages if circuits are badly flawed. However, analogue circuit simulation is not a mindless point and click activity. Spice simulations can sometimes produce apparently wrong or misleading results without producing any error messages or warnings.

This is a problem if the user does not understand its limitations, how to interpret the results, or how to use the right options and settings to get the best results. 741's question is an example of this – FFT analysis may produce apparently strange results if the data or settings are insufficient or inappropriate.

Minimal circuit

A very minimal circuit can be used to experiment with the FFT function in LTspice. All you need is a voltage source and a resistive load, as shown in Fig.1. The spectrum of the voltage across the load can be displayed and the various parameters of the simulation and the way the FFT is calculated can be varied to see what effect they have on the spectrum displayed. The voltage source can be switched between a sinewave and a square wave to create a simple, or more complex, true spectrum, as required.

To get a spectrum like the one 741 is asking about, do the following. Enter the schematic shown in Fig.1, right click on the voltage source, click on 'Advanced' and select a sine function in the voltage source dialog. Set the amplitude to 1 and the frequency to 1kHz and click OK to close the dialog.

Use the 'Edit Simulation Cmd' function from the Simulation menu and select transient simulation. Enter a stop time of 5m and click OK. Run the simulation and add vout as a trace on the waveform window. You should see five cycles of a 1kHz sinewave. Right click on the waveform and select View -> FFT. This will open the FFT dialog (see Fig.2). Check vout is selected and click OK.

The spectrum produced will be like that shown in Fig.3 (this shows only part of the default spectrum displayed). We would expect a very narrow peak at 1kHz and nothing at other frequencies. There is a peak at 1kHz, but, as indicated by 741, it is much wider than expected, ranging from 800Hz to 1.2kHz. This is due to spectral leakage, which we will discuss later.

There is also a lot of 'noise' at other frequencies, much of which is due to the fact that LTspice compresses the waveform data from the simulation. A better spectrum is obtained if the compression is disabled, we will discuss this next month.

In order to understand how to get the most out of a spectrum (FFT) analysis in Spice, we need to be aware of a few things about how spectra are produced from measured or simulated data, and the errors and approximations which are inherent in this process.

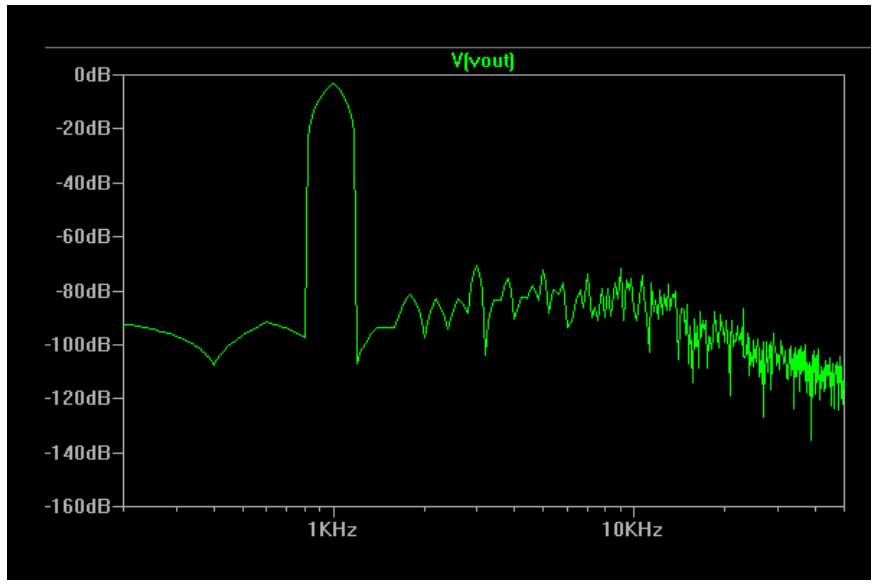

*Fig.3. The influence of things such as data compression and spectral leakage result in a spectrum which is **not** very close to that expected for a pure sinewave*

Fourier series

As we mentioned in the previous article on THD, one of the mathematical concepts underlying the spectrum is the Fourier series. Any *periodic* waveform can be formed from a sum of sinewaves at different amplitudes, and a plot of the amplitude of these frequencies (against frequency) is called a spectrum. More specifically, a Fourier series is a set of frequencies which are multiples of a fundamental frequency. A frequency which is n times the fundamental is called the nth harmonic. The only periodic waveform which has only one frequency in its spectrum is the sinewave itself.

The Fourier series is a powerful mathematical tool for understanding and analysing signals, but it has the limitation that it only applies to periodic signals, of which sinewaves and square waves are simple well known examples. More generally, a periodic waveform is one for which we can draw a portion of the waveform of length, T, in time and the whole waveform consists of this shape repeated continuously. T is called the period of the waveform. Many real-world signals are not periodic (eg, voice signals and one-off pulses).

We will not go into the details of the mathematics here, but it is possible to extend the idea of the Fourier series to cover non-periodic waveforms. Briefly, what happens is that instead of the spectrum being a set of individual discrete frequencies (the harmonics we mentioned above), we get a spectrum which is a continuous function of frequency.

Most readers will be familiar with the idea of an analogue signal or waveform – one that varies continuously with time, and which can take any value (within a specific range). At each instant in time an analogue signal has a defined value, which may be different from the value at any other instant. Similarly, the spectrum of a non-periodic signal is 'analogue' – the amplitude varies continuously with frequency and may have a different value at each and every frequency.

Fourier transform

The mathematics which defines the continuous spectrum of a signal is called the Fourier transform. Thus, if we have a signal we are interested in, in theory we can apply the Fourier transform to the signal to find its spectrum. There is, however, another practical difficulty here. We do not know (in general) the mathematical function which represents our signal; but this is the 'input' required by the Fourier transform. Looked at another way, we would need an infinite number of measurements, even for an arbitrarily small section of signal, (or simulation data points) in order to find the spectrum.

In practice, we have to measure (sample) the signal (or perform the simulation) at finitely spaced time points, and use just this data. As the Fourier transform is defined for continuous signals, we have to again adapt the mathematics to account for this limitation. For sampled signal data we can use the Discrete Fourier Transform (DFT) to find the spectrum.

Using the DFT we are able to build hardware, or write software, which will find the spectrum of an arbitrary signal. However, there are still a couple of limitations to consider. First, we can only consider the signal for a limited time, whereas the DFT is defined in terms of an infinite number of sample points (over infinite time). This is not the same problem as with the Fourier transform, because a finite section of waveform is now represented by a finite number of sample points.

By using a limited number of samples we do not break the technique, but we do reduce the accuracy of the spectrum we produce; we will look at this in more detail in a moment. The number of sample points is one of the parameters which can be set by the user when calculating a spectrum using LTspice.

The second limitation is that we can only perform the DFT calculation a finite number of times. So practically, we can only find a discrete spectrum, rather than the true continuous spectrum.

Each DFT calculation provides us with a single frequency point in our spectrum (amplitude at that particular frequency) – plotted directly, the spectrum will

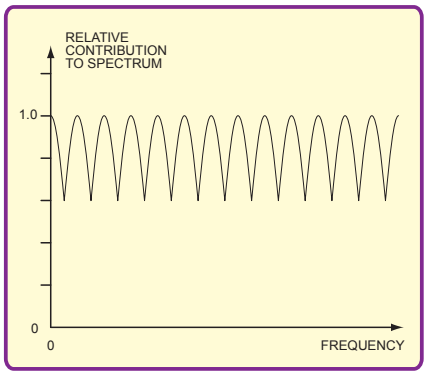

Fig.4. Variation in weighting of contribution to spectrum with frequency (the picket fence effect)

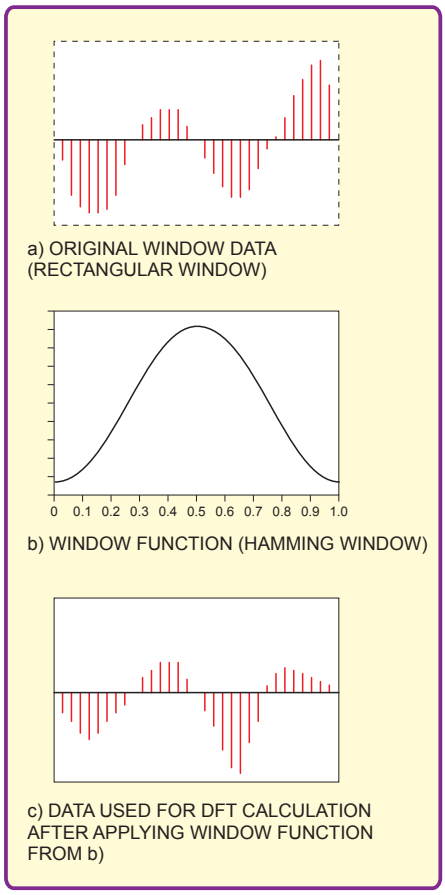

Fig.6. Applying a window function to the set of data used to produce the spectrum can improve the quality of the spectrum by reducing discontinuities introduced by imposing the window

look like a histogram. Of course, if we calculate a large number of frequency points we can plot them on a graph with a smooth curve through them and get what looks like a continuous spectrum (like Fig.3).

Picket fence

It is useful to think of the spectrum as being like a histogram, with the spectrum divided into frequency bins, but unfortunately the DFT spectrum is not a perfect histogram of the frequency content of the signal. In comparison, if we were looking at the statistics of people's height, everyone between 5ft 4½in and 5ft 5½in could be classed as being 5ft 5in (put in the

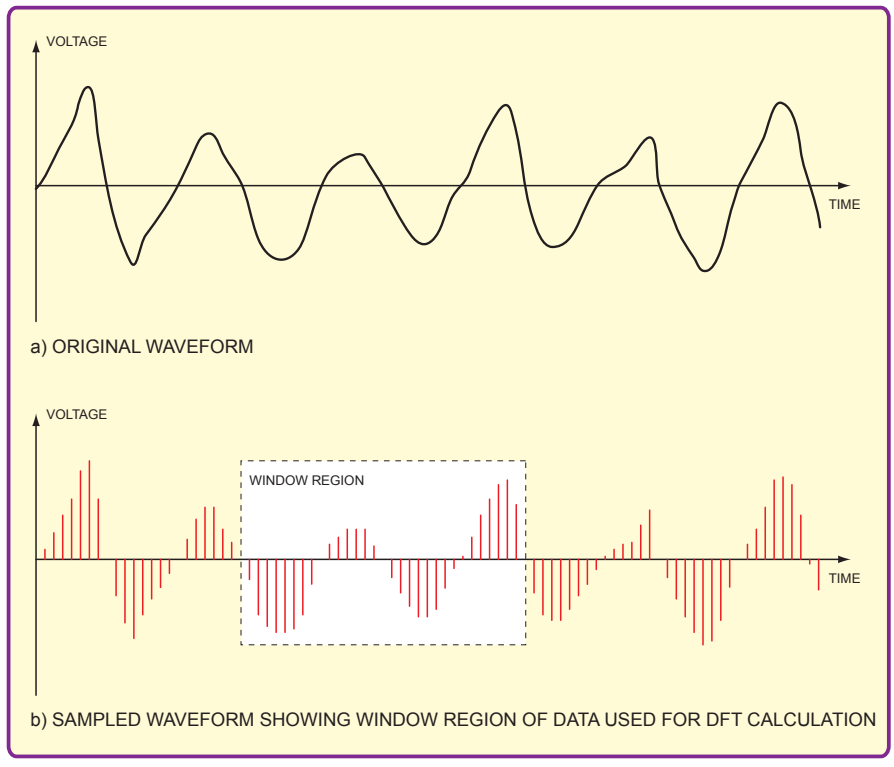

Fig.5. The data used to display a spectrum is a window of samples of the original analogue waveform

'5ft 5in bin' in the histogram). Everyone in this range would have an equal influence on the total height of the relevant bar on the histogram.

However, for a DFT-based spectrum frequencies close to the centre of the bin (the exact frequency at which the DFT calculation is performed) have a stronger influence on the spectrum than those in-between (see Fig.4). This would be like giving more importance in height statistics to people who were close to an exact number of inches tall, and less importance to those in-between.

The nature of the DFT means that the frequencies at which the spectrum is calculated do not necessarily coincide with the most important frequencies in the original signal. This is exasperated by the uneven weighting described earlier. The result is referred to as the picket fence effect, because the DFT spectrum is like looking at a scene through a picket fence, you can only see through the gaps, and may miss important details.

Specifically, the DFT spectrum may show peaks in the spectrum as frequencies different from the actual peaks, and the actual peaks are likely to be lower than those in the true spectrum. The shape of the weighting graph in Fig.5, may also remind you of a picket fence.

Limitation

Returning to the limitation of a finite number of samples, this is illustrated in Fig.6, which shows that the set of samples we actually use is like a window looking at part of the signal. The fact that we look at just this window, rather than the whole signal, distorts the spectrum that we observe; we can never see the true spectrum. Mathematical analysis of this distortion shows that the observed spectrum is spread out compared with the true spectrum. This is known as a *spectral leakage*. This accounts for the 'skirts', or lack of peak sharpness, which 741 observed.

The abrupt start and finish of the window shown in Fig.5 results in a large amount of spectral leakage. This can be used by decreasing the amplitude of the samples towards the end of the window. To do this, the data in the window is scaled by a windowing function, as illustrated in Fig.6.

Windowing functions can be selected in the FFT dialog in LTspice (Fig.2), and there are many to choose from. For simple repetitive waveforms like sinewaves, window functions may not help much, but the rectangular window must be aligned correctly with the waveform to get a good spectrum.

So far in this article we have referred to the Discrete Fourier Transform (DFT) many times, whereas 741's question mentioned the Fast Fourier Transform (FFT). They are, in fact, the same thing. The FFT is a very clever way of *implementing* the DFT mathematics in hardware or software.

It is very efficient and fast (hence the name of course), but it normally requires that the number of waveform data points used is a power of 2; this is not usually an onerous restriction. You will see that the 'number of data point samples in time' parameter in the LTspice FFT dialog (Fig.2) can be set in powers of two from 256 (2^8) to 16777216 (2^{24}).

In this article we have looked at some aspects of the DFT (and hence FFT), which is used by LTspice to display spectra for simulated waveforms. These ideas help us understand some of the parameters which can be set in the FFT dialog. Next month, we will look in more detail at how to use LTspice to display useful spectra.