

Fundamentals of Signal Integrity

ARTICLE #8

By Dr. Howard Johnson
Signal Consulting, Inc.
June 15, 2009

Summary

LeCroy Corporation, in association with Signal Consulting, Inc., has prepared an eight-part series on the fundamentals of signal integrity. Authored by the world's foremost authority on signal integrity, Dr. Howard Johnson, the series is a "must read" for engineers who need a clear understanding of issues essential to high-speed performance.

Other papers in the series include *Confirm The Diagnosis, Adequate Bandwidth, and Step Response Test*. To read other parts in the series, please visit: <http://www.lecroy.com>

Jitter Capture

Einstein points out that everything must be measured relative to something else. In the context of jitter, my question is, "Relative to what?" To investigate that question I shall first discuss a general problem of statistical representation.

A general problem of statistical representation

Suppose I ask you to measure the standard deviation of some electronic signal. You probably remember that standard deviation is the root-mean-square (rms) measure of deviations from the mean (average value). The first thing you must do, then, is capture some samples from the signal and find its mean. Then you can begin looking at differences from that mean and commence calculating the rms deviation.

What if, at the outset of your calculations, you begin with the wrong mean value? Obviously your calculations, and everything based on them, would be incorrect.

Let me illustrate the difficulty of finding the correct mean with an example. Figure 1 depicts a 100-MHz sine wave. The horizontal scale is 2 ns/div.

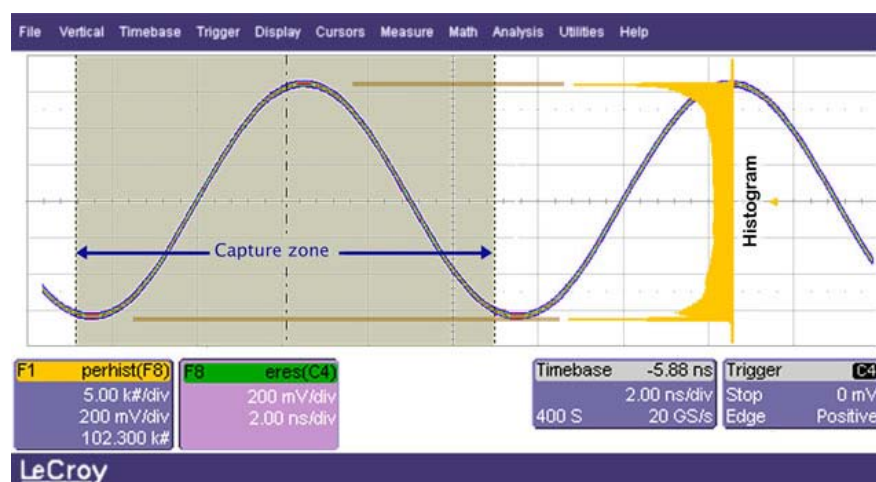


Figure 1 - One full cycle of a repetitive waveform contains enough information to reconstruct the entire signal histogram.

In this simple example the signal repeats every 10 ns. Figure 1 exploits that fact, sampling one exact period of the waveform as shown in the yellow-shaded region. Every cycle is the same, so there is no point in taking additional data. The histogram at right (rotated 90 degrees) represents the various vertical values sampled. From the histogram you may extract the mean and other parameters you seek.

Repetition makes statistical measurement easy. Every individual cycle of the signal contains all the information you need.

The informational content of a non-repetitive signal is more widely dispersed. Accurate statistical measurement of a continuous, non-repetitive signal requires data spanning multiple cycles of the lowest-frequency components present in the signal. Going back to the signal in Figure 1, imagine what might happen if you don't capture enough data? What if you come from an advanced galaxy where 10 ns seems like a R-E-A-L-L-Y L-O-N-G T-I-M-E? If you do not know that the signal is going to repeat, then, after capturing data samples for only one or two ns, you might become bored, stop recording, and call it a day.

Depending on where in the signal cycle your little burst of captured samples occurs; the signal might appear deceptively quiescent. For example, Figure 2 samples a region only 1.5 ns wide, taken at a location near the trough. The sampled values are taken from the yellow-shaded region at the left side of the screen.

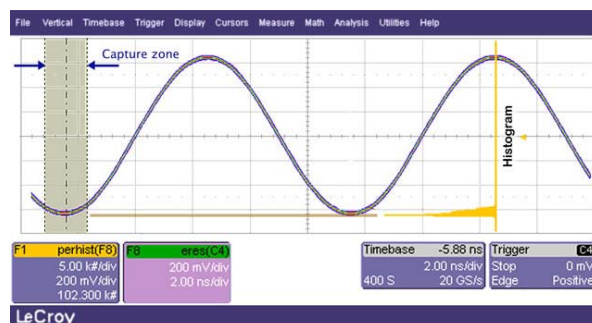


Figure 2 - A short burst of samples taken near the trough produces a non-representative histogram.

The mean value averaged over that short burst of samples (yellow region) represents only the lower extreme of signal excursion. That's not right. Even worse, for the samples inside the yellow region, *the deviation around that mean is too compact*. Given only this limited sampling of data you may (erroneously) conclude that this signal exhibits few if any deviations from the mean.

In a second example, a short burst of samples captured near the zero crossing presents an entirely different histogram (Figure 3). This histogram correctly predicts the mean, but exhibits a misleading trend: all the data points within the central yellow region ascend in a continuous straight line. A financial analyst faced with such data might predict that the signal goes up forever. As you know, probably all too well, that never happens.

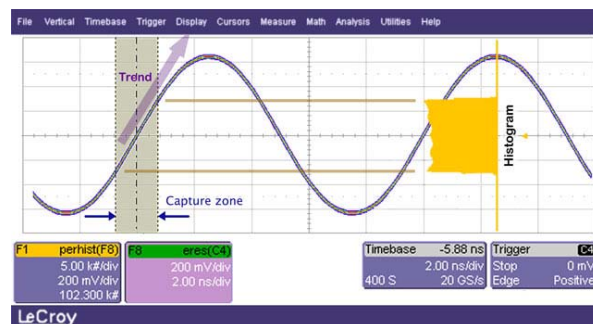


Figure 3 - Near the zero crossing, this signal trends up linearly.

Regardless what kind of systems you study, from serial links to stock markets, you must always collect enough data to see the full extent of a signal's true variations before you can accurately estimate its parameters.

Required data capture time

Let's return to the problem of measuring jitter.

Mentally re-scale Figure 3 to a frequency of 100 KHz. Suppose it represents a clock phase wandering up and down slowly in sync with a switching power supply operating at 100 KHz (the clock phase in many systems does that). If you capture data in short bursts (perhaps 20,000 samples at 20 GS/s = 1 us), then your short burst of samples might make a pattern just like that shown in Figure 2 or 3.

No matter where you place the short burst, it never sees the full measure of the jitter deviation. Furthermore, in contrast to the pictures shown in Figures 2 and 3, the TIE measurement process, by its nature, optimizes the phase positioning of the reference clock *within each short burst*. In effect, the TIE process "re-centers" the histogram of jitter taken from each short burst of samples.

Applying that idea to the short burst of samples captured in Figure 2 near the bottom of a sine wave, the re-centering effect shifts the histogram up vertically, centering it at zero. Even if you aggregate the information from millions of histograms, the re-centering of each short burst destroys all information about the phase of one burst relative to the next. You'll never obtain the correct statistics that way.

The proper representation of low-frequency jitter requires coherently-sampled data records that span long periods of time.

So, how much data must you capture?

Mathematically, if the most slowly-moving features of the signal undulate at frequency f_0 , then the required data capture time T_{capture} is given by:

$$T_{\text{capture}} = (\text{several times}) \times (1/f_0) \quad [1]$$

Taken to an extreme, as f_0 approaches DC, the required capture time soars to infinity. Therefore, if your signal includes significant components that go all the way down to 0 Hz, you must capture data for all time. Yep. For all time. Talk about inconvenient!

It would be better for you if jitter had a limited bandwidth that did not extend to zero frequency.

Is there such a thing as DC jitter?

Set up a wideband wireless transmitter and receiver at either end of a large warehouse. To make the numbers easy, let's suppose they operate at a baud rate of 1 GHz. Lock the receiver onto the transmitted bit stream.

If the warehouse is 100 feet long there are at any one time about 100 bits of information stored in the space between the transmitter and receiver (radio waves in air travel at a speed of approximately 1 foot/ns). Now pick up the transmitter and move it closer to the receiver, cutting the distance in half (Figure 4). From the perspective of the receiver information now arrives 50 bit times *earlier* than previously. That's 50 bit times worth of peak-to-peak jitter. If I move back and forth once a day, the same shift in timing recurs at a daily rate (11.57 micro-Hz).



Figure 4 - The space between transmitter and receiver stores one hundred bits of information. (Ed. Note: that's my wife, Liz, holding a cardboard box with wooden dowels stuck in the top for antennas. Cool use of light, no?).

To measure that kind of long-term variation in timing you would have to sample a coherent data record that spans the entire daily movement. That could require hundreds of billions of samples.

Astrophysical problems involve even more extreme amounts of jitter. An Earthbound receiver listening to transmissions from Mars experiences deviations in timing proportional to the distance between the two planets. As the distance varies from 36 to 250 million miles the total variation in delay (total jitter) varies by 1129 sec, or about 20 minutes. That's a lot of jitter, but it develops slowly. The distance between the planets undulates back and forth at a rate commensurate with the differences in their orbital frequencies. One complete undulation occurs every 778 days (14.9 nano-Hz).

Systems afflicted by $1/f$ noise, thermal drift, and other noise sources can also exhibit large, but very slow-moving, jitter.

I could do this all day long, but you probably get the idea: many systems exhibit large amounts of low-frequency jitter.

In an environment with near-zero-frequency jitter, the relation previously described suggests that you must capture near-infinite amounts of data to completely represent jitter. That's true, but a *complete representation of all jitter is not what you want to measure*.

What you really want to measure, whether you realize it or not, is a very different concept. What you want to do is this:

Measure only jitter that matters to your receiver.

Excluding wander

The receiver in my wireless example probably incorporates a PLL (phase-locked loop) that tracks slow-moving changes in the received data timing. The PLL forms an estimate of the localized frequency and phase of the incoming data stream. Based on that estimate it constructs an internal reference clock (the recovered data clock) and it uses that clock to decode the incoming data.

When you physically move the transmitter the PLL adjusts its local clock to track the incoming data. It does so with a response time characteristic of the PLL. As long as the incoming data timing adjusts itself slowly, on a scale of time long compared to the PLL response time, the PLL tracks almost perfectly.

In the context of jitter performance testing, long-period timing undulations matter very little to a tracking PLL. To differentiate those long-period undulations from the shorter, more urgent variations in timing that matter a LOT, engineers have over the years refined the concept of jitter into two separate terms: wander and jitter.

Long-period undulations, on a scale of time long compared to the tracking response time of your PLL, are called "wander". Short-period undulations, on a scale of time short compared to the tracking response time of your PLL, are called "jitter". The boundary between wander and jitter depends totally on the characteristics of your PLL. There is no other factor distinguishing the two. One man's wander could easily be another man's jitter, and vice-versa. To speak intelligently about jitter testing, the discussion must include consideration of the tracking bandwidth of the device receiving the jittery signal.

As long as your PLL remains locked, wander has little practical effect on system performance. Because it has little practical effect, even if your data signal incorporates massive amounts of wander you need not measure it, and the captured data record need not represent it. That is important. The wander in your signal is the only part that includes components extending down to zero frequency. Once you throw out the requirement to measure wander, the bandwidth of the remaining portion of the jitter signal no longer extends down to zero frequency. Any technique that measures only jitter, and excludes wander, eliminates the need for infinite data records, leading to this simplification:

Your captured data record need be no longer than several PLL response times.

You may choose to aggregate many independent data records to build a deeply detailed histogram of performance, but each individual record need be no longer than several PLL response times.

Practical measurement of jitter

In the context of a serial data communication system, assuming the receiver's PLL maintains lock, all that matters in the receiver circuit, in terms of jitter, are deviations between the instantaneous incoming data phase and the receiver's PLL-generated data recovery clock.

Given an identical input waveform, the TIE@LEVEL function measures almost the same deviations. The only discrepancy is that the TIE function measures deviations between the instantaneous incoming data phase and the TIE internal reference clock, not the receiver's PLL-generated data recovery clock.

To the extent that the TIE@LEVEL function and the receiver's PLL generate different internal reference clocks, because they use different tracking algorithms, their perceptions of jitter will differ. If you want to measure jitter the same way the receiver sees it, then program the TIE function to mimic the PLL algorithm for its internal reference clock generation:

Use for TIE jitter measurement the same PLL tracking algorithm as your receiver.

That is not merely possible; it is the *required* method for jitter measurement. It excludes wander in precisely the same way as your PLL circuit.

Provided that you take sufficiently long data records, this method sidesteps all issues about the existence of low-frequency wander. Whatever your receiver sees, your measurement sees also. I'll talk more about that in my next article.

Tying it all together

There exists no generalized, self-consistent way to measure the complete range of all jitter, because there is no way to capture jitter components all the way down to zero frequency. If you attempt to measure jitter at all frequencies you will discover your readings just getting higher and higher as you make your data records longer and longer. Even if your input signal is perfect, your scope isn't. In the limit, as you attempt to measure jitter over very long periods of time, you just end up measuring noise in the scope's reference oscillator that soars to infinity as the data record approaches infinite length.

Instead of trying to measure "all jitter", focus your attention on measuring the jitter of interest to your receiver. Capture coherently-sampled data records several times longer than the tracking response time of your receiver's PLL and measure jitter against a reference clock generated using the same PLL algorithm as your receiver.

A PLL-based reference clock is the relative signal against which Einstein would want you to measure jitter.

Best Regards,
Dr. Howard Johnson