Measuring small signals accurately: A practical guide

Steve Hageman - August 23, 2012

Introduction
Measurement of signals close to the your measuring instrument's noise floor is always difficult.

For many years the best Op Amps have had input noise floors of right around 1 nV/√Hz. This makes any measurement significantly below this value very difficult to measure. Likewise, in RF systems the best amplifiers have Noise Figures below 1 dB, this is also difficult to measure as there are not many amplifiers available that have much better noise figure than this.

Averaging our measurements only reduces the variance of what we are measuring, but does not reduce the effective noise or increase real measurement resolution.

There are two methods we can use to overcome this natural limitation however as will be shown below.

First a note: These methods are usually applied to instruments that make some kind of a measurement that involves a frequency response (the Cross Correlation method only works with signals that have had a DFT or FFT applied). While these methods could be applied to a strictly DC measurement, there are probably better methods, such as adding dither and oversampling.

Method #1: Noise De-Embedding
Noise De-Embedding is a method to measure the noise floor of the instrument in question then by using math to get a better accuracy of the actual signal even when the signal is below the instruments noise floor.

In spectrum analyzer circles it is well known that if one measures a signal that is 3dB above the noise floor, then that signal level is exactly at the noise floor. The two equal valued signals add in power to give a 3dB increase above the noise. But if you just used the measured value you would be 3 dB off in your measurement accuracy. This is true for CW signals or if the signals are noise (or digitally modulated, which looks much like random noise).

So by knowing our instrument's noise floor (by measuring it first) and knowing that our measured signal is 3 dB above the noise floor, we know that the signal is actually at the noise floor. This method only works when the instrument noise is uncorrelated with the signals noise - which is usually the case.

This correction method is commonly called "Noise Floor Correction" [1] or "Spectrum Analyzer Noise De-Embedding" [2] and is used in many single channel receiver and Spectrum Analyzers. The limit of the extension is usually about 7 dB. At that point the measured signal is only 1dB above the
instrument’s natural noise floor and the total measurement gets so low above the actual noise floor that it is difficult to detect it with any certainty.

The math is not difficult however, and because this is a scalar process it can be used with any measuring instrument.

In power terms: The internal instrument noise (Ni) adds with the measured noise (Nm) to give a displayed total noise (Nt), where \( N_t = N_i + N_m \).

For example if the total measured noise \( N_t \), is 2dB (which means that the difference between \( N_m \) and \( N_i \) is 2 db) then the linear power ratio of 2 dB is \( 10^{2/10} \).

If we set the normalized internal instrument noise to a value of 1.0, then by subtracting this 1.0 value from the linear ratio leaves us with: \( 1.58 - 1.0 = 0.58 \).

\textit{Note: We are showing only 2 digit precision, but the calculations are being done on a calculator and between steps we are not losing digits. To follow the results exactly do the math step by step, using the multi-digit results from the previous step as the input to the next step.}

We can then calculate the correction factor as: \( 10 \text{Log}10(0.58) = -2.33 \text{ dB} \).

So our total correction on this 2 dB signal is: \( -2.33 - 2.0 = -4.33 \text{ dB} \).

What we were actually measuring when we saw a signal that was apparently 2 dB above the noise floor is a signal that was actually -2.33 dB below the noise floor.

A typical correction factor curve that can be generated for this method is shown in Figure 1.

A downside of this method is that if you mis-measure the instrument's noise floor and it is actually higher than you think it is, then the math "Blows up" and you get a very bad result (Figure 1).
Figure 1 - A plot of the Noise De-Embedding correction factor. The Middle trace (Blue) is the Nominal correction factor. In our example: If we measure a signal 2 dB above the noise floor we can read that the total correction factor is -4.33 dB, which means that the signal measured was really: 2 - 4.33 = -2.33 dB below the noise floor. The Red and Green trace show what happens to the algorithm if the noise floor is higher (Red Trace, bottom trace in the figure) or lower (Green Trace, top trace in the figure) than we think it is. These traces show the effect of a 0.5 dB error in our noise floor estimation. As can be seen, even a 0.5 dB error can add a severe measurement uncertainty to the result. In practice we tend to limit the correction factor to 7 dB as about the maximum reasonable correction possible for this method.

So as with all things that involve a calibration, you need need to know how good your calibration is and how well it holds with time and temperature, etc., or measure the noise floor often.

Many of us that work around RF synthesizer circuits use our Spectrum Analyzers to measure Phase Noise as this is quick and easy and doesn't require the big specialized Phase Noise measurement system. The downside of this is that most of the time our spectrum analyzer's noise floor is about the same as even our low cost synthesizers. By using the Noise De-Embedding method we can effectively extend our measurement accuracy better than 6 dB lower than just using the spectrum analyzer measurement result alone.

Another common use for Noise De-Embedding is in RF Measurement receivers where we measure the signal strength of various signals over the air. If we are measuring a signal very close to the sensitivity limit of our receiver we can apply Noise De-Embedding to improve the accuracy of the measurement result by a significant amount all without increasing the cost of our receiver at all.

As can be seen the method can be used in any system that must measure signals that are very close to or below the measurement noise floor for an immediate increase in measurement accuracy.

Method #2: Cross Correlation

A DSP based technique for measuring very small signals that is gaining in popularity is called "Cross Correlation" (See Sidebar). This is basically a technique where the input signal is split between two input channels, where it is simultaneously measured and vector averaging is preformed to cancel the uncorrelated noise in the input channels as shown in Figure 2.

Figure 2 - In a Cross-Correlation system the input signal is split to two identical Vector Measurement Channels. Each of these input channels adds it's own noise to the input
signal, but since both of the Vector Measurement Channels input noise is uncorrelated with each other - with enough vector averaging we can reduce the effective uncorrelated measurement channels noise floor by upwards of 20 dB (10:1 in linear terms). With measurement hardware becoming lower cost all the time this concept is getting used more and more, even in low end instruments.

Vector averaging as opposed to straight scalar averaging adds digital processing gain to the measurement so that the uncorrelated measurement channels noise cancels and actually can be reduced with multiple vector averages. This is in contrast to straight scalar averaging of noise where only the variance of the noise is reduced not its absolute level.

The tradeoff is of course that you need to vector average the signal 10,000 times to get a 20 dB reduction in effective measurement system noise and the signal of interest must be consistent over all these averages.

Sidebar - Many forms of Cross Correlation

On searching the literature it is pretty easy to find numerous uses and examples of the term "Cross-Correlation". There are also many different measurements that use the same term "Cross-Correlation" but are really doing different things. Cross-Correlation in it's most common DSP implementation is used to find time shifted correlation between two signals - for instance in Radar systems cross correlation is used to find the time delayed reflection off a target. Here the process is a one shot calculation and is not usually buried in the measurement system's noise. This is not how we will be using the term Cross-Correlation, we are more interested in the common electronic measurement usage of the term specifically relating to a multiple channel vector averaging technique that can measure signals below a single instrument's natural noise floor.

But think of the possible result: Using 1 nV / √Hz input Op Amps in the Input Measurement Channels and with 10,000 Cross-Correlations, the effective instruments input noise is reduced 10:1 or to 0.1 nV / √Hz - Wow! Just try to build a single Analog Front End that good!

The Hardware Setup

The minimum system configuration is a dual-channel system (or two single-channel instruments) that sample simultaneously and can produce a vector result. This is pretty easily done with measuring instruments that digitize and FFT the result, as long as the complex result of the FFT can be read from the instrument then Cross-Correlation can be preformed.

You will have to search your instrument's manual to see if you can either get the raw time record (So that you can perform your own DFT/FFT) or you can get the complex FFT result directly. Most RF spectrum analyzers do not include this capability as they are "Magnitude Response" devices, the FFT analyzers on the market vary as to the availability of the proper data output. If you are building your
own measurement system then you won't have any issues with getting the proper data as you will have direct access to the ADC data output.

**Cross-Correlation Math**

Many papers that deal with Cross-Correlation gloss over the math part or in some cases are outright incorrect. One excellent paper that is correct was written by Messrs Rubiola and Vernotte titled: "The cross-spectrum experimental method" [3].

Cross-Correlation is also where we get to use the Phase Information that is inherent in our DFT results. As I discussed before in a previous article [4], the results of a DFT (or FFT) on a scalar series of samples (like from an ADC) is a vector or complex result. Normally we convert this vector result at each frequency bin to absolute magnitude for spectrum analysis applications. Here however we can put that complex DFT result to good use.

The basic procedure for Cross Correlation is to:

- Capture ADC data simultaneously from the two measurement channels
- Apply an appropriate window on the data if needed
- Perform a DFT (or FFT) on both channels of the ADC data
- Take the complex conjugate of one of the DFT's
- Vector multiply the DFT and the complex conjugate DFT
- Vector Accumulate the sum of the Vector Multiply step above
- Repeat above for as many averages as are needed

At the end of the vector averaging,

- Divide the Sum of the vector averages by the number of Cross-Correlations: n
- Find the Square Root of Magnitude Squared (see Footnote) of each vector point.

And that gives you the result in a Scalar Magnitude Spectrum Format which can than be scaled to an absolute measurement value or a suitable noise density plot, etc.

As was done previously with the DFT Article [4], I have built a small C# application that demonstrates Cross-Correlation using GNU licensed software that you are totally free to play around with and modify any way you wish [5].

Some detailed information on the math is in order and might save you hours of research and blowing dust off those college math books.

Taking the Complex Conjugate of a rectangular formatted complex number is very easy - all you have to do is to multiply the imaginary part of the number by -1.0.

Vector Multiplication is defined as follows,

\[(a+jb) \times (c+jd) = (ac-bd) + j(bc+ad)\]

Where \(a+jb\) and \(c+jd\) are vector or complex numbers that are naturally the result of our DFT or FFT operation [4].

The Vector Summation is accomplished as,
\[(a+ jb) + (c+ jd) = (a+c) +j(b+d)\]

To get the average of the vector sum, simply divide the real and imaginary parts by the number of summations or Cross-Correlations that were preformed.

All that is left to do is to get the scalar magnitude of the complex number to for display purposes as was described in an earlier article[4]. The scalar magnitude for the averaged vector summations needs to be modified somewhat because of the math manipulation that has been done on the intervening complex numbers.

If we just take the scalar magnitude of the resulting complex averaged value and convert to dB values as we would for a standard DFT [4],

\[\text{d BV} = 20 \times \log_{10}(\sqrt{a^2 + b^2})\] - Standard DFT Magnitude calculation

Where, the complex vector is represented as: \(a+jb\)

We will find that we over-calculate the magnitude by 3 dB because our final averaged complex result is the multiplication of two DFT's to begin with.

So our actual method of converting to scalar magnitude for a averaged Cross-Correlation measurement is,

\[\text{d BV} = (20 \times \log_{10}(\sqrt{a^2 + b^2})) - 3.0 \text{ dB}\]

This 3 dB loss is also why the processing gain for the cross correlation method is only one half of what we would expect. Normally when adding powers (or voltage squared values), we would expect that the processing gain would be,

\[\text{Power Processing Gain} = 10 \times \log_{10}(\text{Number of power averages})\] dB

But because we have to take this 3 dB correction into account we now end up with one half of the processing gain or,

\[\text{Cross Correlation Processing Gain} = 5 \times \log_{10}(\text{Number of power averages})\] dB

Hence we can see that to get a 20 dB of processing gain we have to take 10,000 averages.

**An Example**

An Example

Using the example C# application provided [5] I have setup a example that has the Gaussian noise exactly at the level [6] of a single tone (CW) input signal (all set to 0 dBV). Figure 3 shows the result of 1, 100 and 10,000 Cross-Correlations. As you can see the actual signal is more and more prominent with increased Cross-Correlations. The signal variance also reduces as the noise level is pushed lower and lower.
Figure 3 - The results of a Cross Correlation simulation. The Gaussian noise level was set to the amplitude of the single tone signal at 0 dBV. Even at 100 Cross-Correlations the signal is plainly visible out of the noise and at 10,000 Cross-Correlations the noise has been reduced by some 20 dB.

If you are measuring the spectral density of noise instead of a CW tone, what you see with increased cross correlations is a reduction in variance in the signal that you are trying to measure. This is because the effective noise floor is pushed lower and lower by the processing gain and the spectral density that you are measuring (the actual signal) is being averaged, so it shows less variance (Figure 4).
Figure 4 - If Cross-Correlations are performed on a noise like signal, increasing the Cross-Correlations has the effect of really averaging (and reducing the variance) of the actual noise signal as can be seen here for this simulation of a 0 dBV Gaussian noise signal. At 10,000 Cross-Correlations the variance is reduced from over 10 dBV peak to peak to less than 0.5 dBV peak to peak (Note the scale change between the graphs).

Practical Limitations
Cross-Correlation is an amazing technique, but it does have it’s limitations. When measuring noise we usually have long measurement times anyway and the thought of waiting around for 10,000 measurement cycles or more is daunting, unless you have all day for a measurement. But even with 100 Cross-Correlations you can get real results.

There is another practical limit even if you have infinite time to make measurements. This limit is: How uncorrelated are the measurement channels from each other? In practice it is difficult with measurement systems like this to really get much more than 20 to 40 dB un-correlation between channels. At these levels the power supply noise, channel to channel isolation, clock jitters and even the measured signal itself starts to be correlated and will limit the ultimate noise floor achievable.

Conclusion
Measuring very small signals close to or even below our instrument’s noise floor can be made more accurate by using Noise De-Embedding or if a multichannel Cross-Correlation technique is applied. The downsides of measurements near or below the noise floor is that it takes more time to make measurements than with high signal-to-noise ratios simply because we need to use lower bandwidth filters or other averaging techniques to reduce the noise variance. Neither of the techniques presented here help much with that problem.
But perhaps you have a new technique or two to use to make very low noise measurements more accurate and now have new insight about how to go about actually measuring signals that are in fact below your present instrument’s noise floors.

**About the author**

Steve Hageman is a confirmed "Analog-A-Holic" since about the fifth grade when he built his first short-wave receiver. After acquiring his first his first Apple ][ computer in 1982, Steve has always enjoyed marrying Software to Analog Hardware to build useful measurement systems. Since then Steve has had the pleasure of designing such diverse products as: Modular Data Acquisition Systems, Switching Power Supply Test Systems, Radio Receivers, RFIC Test Systems and most recently Software Defined Radios for Wireless Testing and Spectrum Analysis. Steve may be reached via his website at [www.AnalogHome.com](http://www.AnalogHome.com) or his blog at [AnalogHome.BlogSpot.com](http://AnalogHome.BlogSpot.com) or [The Practicing Instrumentation Engineer](http://ThePracticingInstrumentationEngineer).

**Footnote**

To find the scalar magnitude of a complex number (or vector) we take the square root of the real and imaginary parts squared like this,

\[
\text{Magnitude} = \sqrt{a^2 + b^2}
\]

We will call the operation "Scalar Magnitude" for brevity in the rest of the article.

**References**


[6] "[Gaussian Random Numbers for DSP Applications](http://gaussianrandomnumbers.org)," Hageman, S.

**Related links:**

[The practicing instrumentation engineer's guide to the DFT - Part 1: Understand DFT and FFT implementations](http://dftguide1.practicing-engineers-guide-to-the-dft.com)

[The practicing instrumentation engineer's guide to the DFT - Part 2: Spectral leakage and windowing](http://dftguide2.practicing-engineers-guide-to-the-dft.com)

[The practicing instrumentation engineer's guide to the DFT - Part 3: Other window types, averaging DFTs & more](http://dftguide3.practicing-engineers-guide-to-the-dft.com)