

# **DesignCon 2011**

## Wavelet Denoising For TDR Dynamic Range Improvement

Peter J. Pupalaikis, LeCroy Corporation

[PeterP@LeCroy.com](mailto:PeterP@LeCroy.com)



## **Abstract**

A technique is presented for removing large amounts of noise present in time-domain-reflectometry (TDR) waveforms to increase the dynamic range of TDR waveforms and TDR based s-parameter measurements.

## **Patent Disclosure**

Portions of the information provided in this paper are the subject of patents applied for.

## **Author(s) Biography**

Pete Pupalaikis was born in Boston, Massachusetts in 1964 and received the B.S. degree in electrical engineering from Rutgers University, New Brunswick, New Jersey in 1988.

He has worked at LeCroy Corporation for over 15 years and is currently Vice President and Principal Technologist. For the last two years he worked in the development of TDR based network analyzers. Prior to LeCroy, he served in the United States Army and as a consultant in embedded systems design.

Mr. Pupalaikis has dozens of patents in the area of measurement instrument design and is a member of Tau Beta Pi, Eta Kappa Nu and the IEEE signal processing, instrumentation, and microwave societies.

## Introduction

In order to present the concept of wavelet denoising as applied to time-domain reflectometry (TDR), this paper attempts to approach the topic in the following manner:

1. The need for denoising for TDR is shown from equations that describe the dynamic range of TDR and the main characteristics that impact dynamic range.
2. Denoising is presented from a Fourier domain standpoint. While Fourier methods are not applicable to TDR, it is helpful for analogies that are drawn later.
3. The reason Fourier methods are inapplicable to TDR are explored
4. The concept of wavelets are introduced
5. Denoising is presented from a wavelet domain standpoint.
6. Some results are presented for wavelet denoising.
7. The interpretation of wavelet denoised results is summarized.

## Time-Domain Reflectometry Dynamic Range

[1] provides a formula that shows the dynamic range provided by TDR<sup>1</sup>. A complete derivation of [1] is provided in *Appendix A - Derivation of Dynamic Range*.

$$SNR(f) = 10 \cdot \text{Log} \left( \frac{2 \cdot A^2 \cdot f_{bw} \cdot Fs_{act} \cdot T}{Ta^2 \cdot frac \cdot f^2 \cdot Fs_{eq}} \right) - Noise_{dBm} + P(f) + 2 \cdot (C(f) + F(f)) - 6$$

[1]

In [1], the following definitions are made which are dependent entirely on the conditions for measurement:

- $f$  is frequency (Hz)
- $T$  is the amount of time (s) to average for in the acquisition of the TDR step waveforms.

Furthermore, the following definitions are made which are entirely dependent on the TDR instrument used:

- $A = 0.2$  is the step amplitude (V)
- $f_{bw}$  is the band limit set on the sampler noise (Hz)
- $Fs_{act}$  is the effective actual sample rate of the sampler (S/s)
- $Ta$  is the equivalent time acquisition length (s)
- $frac$  is the fraction of the equivalent time acquisition containing reflections
- $Fs_{eq}$  is the equivalent time sample rate of the sampler (S/s)
- $Noise_{dBm}$  is the baseline sampler noise – with no band limit applied (dBm)
- $P(f)$  is the response of the pulser as a function of frequency calculated by taking the Discrete Fourier transform of the first difference of the step response (dB)
- $C(f)$  is the response of the cables used to connect the pulser/sampler to the device-under-test (DUT).
- $F(f)$  is the response of any fixtures used to connect the pulser/sampler to the DUT.

All of these characteristics are important for dynamic range and will depend on characteristics of the instrument used and the characteristics of the measurement. This paper will not discuss all of these characteristics, but will concentrate on a characteristic that depends more on the measurement and the characteristics of the device-under-test (DUT) as it relates to TDR in general.

In TDR the *signal* applied is in the form of impulsive energy contained in the wave front incident on the DUT. This energy is contained in the rising edge of the TDR waveform and is injected over a very short time interval. The resulting reflections that are measured by TDR come later as this incident wave front moves through the DUT. In TDR, one must wait for these reflections to return in order to measure them, which determines the length of the acquisition needed contained in the value  $T_a$ . During the TDR acquisition, both reflections and *noise* are acquired by the TDR instrument. It is the acquisition of noise during the acquisition that is especially troublesome and if we examine [1], we find that the time of the acquisition is one of the most important detractors from dynamic range. Again, this is because the longer the acquisition becomes, the more noise is acquired. We find that for every ten-fold increase in acquisition duration, the dynamic range drops by 20 dB. This acquisition length is uncontrollable to a large extent because it depends on the DUT itself.

In [1], we find the value  $frac$ , which has been provided to introduce the concept of fractional portion of the waveform acquisition containing real reflections. Wavelet denoising, in a simple sense can be viewed as a method whereby only the portion of the waveform containing the incident wave front and the reflections from the DUT are retained, while the portion of the acquisition containing no reflections, and therefore just noise, are discarded. Therefore, the benefit of wavelet denoising can be thought of as modifying the portion of  $T_a$  to contain only actual reflections. In other words, the dynamic range is affected by the portion of the waveform containing reflections instead of the time needed to acquire the reflections. Again said differently, the waveform is compressed into a shorter duration where that duration only contains actual reflections, thereby increasing dynamic range.

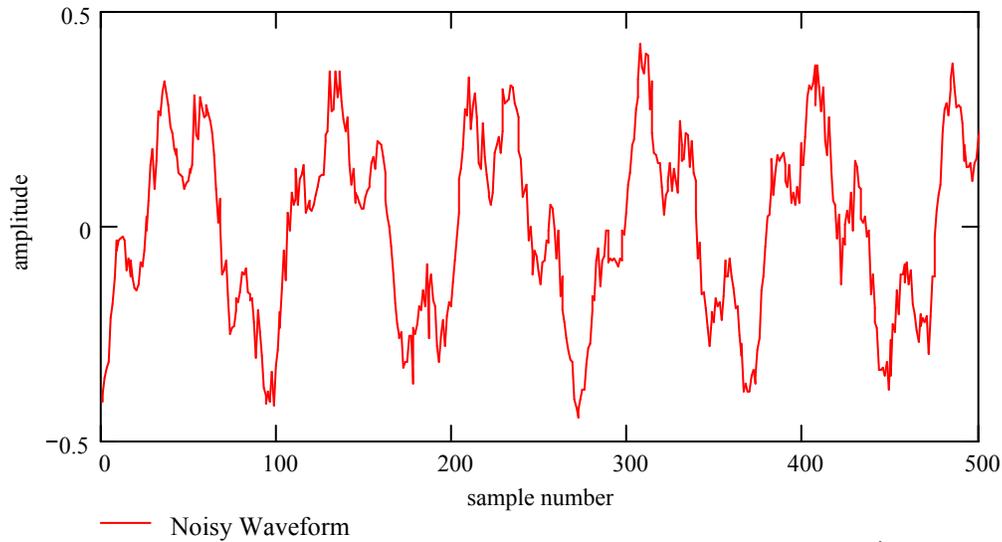
## Fourier-Domain Denoising

In order to understand the concept of wavelet denoising, it is helpful to first view denoising from a Fourier-domain standpoint. This is because Fourier techniques are generally well understood, while wavelets are not. While Fourier techniques are not applicable to TDR, we shall see that the denoising concept is exactly the same with only a change in the *basis functions* used for the denoising.

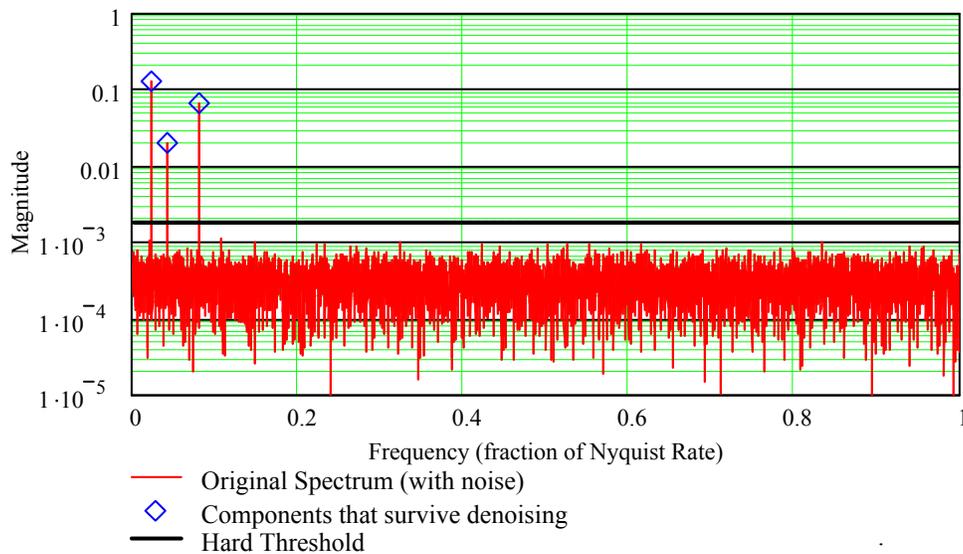
Consider Figure 1 which is shown to contain a noisy signal. It is desirable to have a signal with less noise in it, so we resort to Fourier-domain techniques. This means that we compute the discrete Fourier transform (DFT) and look at this signal from a frequency domain perspective. This is shown in Figure 2. Figure 2 shows some interesting characteristics of the noisy signal in Figure 1 that we'd like to rely on:

1. The signal consists of only a few large frequency components
2. The noise is white
3. The signal components are low in frequency

If we can truly rely on these characteristics, certain well-known techniques are possible for reducing noise. Regarding item 3, an obvious technique would be to filter using a low-pass filter if one can guarantee that the entire signal was below a certain frequency.



**Figure 1 – Example Noisy Waveform**



**Figure 2 – Frequency Content of Example Noisy Waveform**

Regarding item 1, a further, even more effective filtering operation is possible. If we knew the frequencies and number of the components that are considered components of the signal and these frequencies did not change, a more complex filter that retains only the components of interest can make huge changes in noise content. [13] shows how this works. [13] shows the relationship between the noise  $\sigma$  and the expected value  $Ea$  in each bin of the DFT of the

acquired signal. In [13], the value  $N_{bw}$  is intended to show bandwidth (i.e. is the last frequency of bin of interest which is the value calculated as  $f_n$  when  $N_{bw}$  is substituted for  $n$  in [11], but it can just as well correspond to the number of frequency components actually retained in the signal after a particular type of denoising operation. It is easy to show that if there is a value  $frac = N_{bw}/N$  which represents the fraction of the frequency bins in the DFT of the waveform containing actual signal, then the improvement in signal to noise (i.e. dynamic range) in a system becomes:

$$\Delta SNR = -10 \cdot \text{Log}(frac) \quad [2]$$

Note that the dynamic range improvement in [2] looks remarkably similar to the effect of a value  $frac$  on dynamic range in [1] – but in fact they are not the same  $frac$ . Here we are talking about a *fraction of the frequency spectrum* as opposed to a *fraction of the acquisition length in time*. This will become clear later.

Regarding the implementation of a Fourier based dynamic range improvement method, there are two predominant considerations. One is the knowledge of the frequency content of the signal. If the frequencies are known absolutely, then all that is required is to *filter* the waveform. If all of the frequencies of interest are at low frequencies, then a low-pass filter can be applied. If the frequencies are known and simple low-pass filtering is not applicable, one can simply compute the DFT and zero all of the bins where signal is known not to be present and compute the inverse DFT. For optimum denoising, these possibilities do not generally exist, so the next most important consideration is whether the noise is known and more specifically the *noise shape* is known. By noise shape we mean the expected value of the noise in the frequency bins of the DFT as a function of frequency. If the noise shape is not known, there are methods by which it can be calculated either with or without signal present by examining multiple waveform acquisitions. If the shape is known, but the absolute level of noise is not, estimates of noise are easily made using locations in the spectrum where no signal is ever present. For the purpose of this paper, let's assume the latter condition. In this situation, the noise is estimated on each waveform acquisition and this estimate is applied to a known noise shape to form an estimate of the noise in each bin of the DFT as a function of frequency. Assuming the noise is Gaussian, multiplying this frequency dependent noise estimate by a factor, like 5, we form a threshold that, to a high degree of probability, is always above the noise. Using such a threshold, the DFT spectrum is analyzed and components above the threshold are retained and components below the threshold are discarded. This is a so called *hard thresholding* operation in that the transfer function of the threshold is a signum function. In other words, if  $X_n$  represents the value of the DFT of the sequence  $x_k$  according to [8] and [10], and  $E_n$  is the frequency dependent expected value of the noise according to [10], then the denoised version of  $X_n$  is simply:

$$XD_n = \text{sgn}(|X_n| - 5 \cdot E_n) \cdot X_n \quad [3]$$

There are other possibilities than the use of the signum function. These are so called *soft thresholding*<sup>2</sup> operations, sometimes also referred to as *coefficient shrinkage*. My experience is that hard thresholding is simplest and works best for denoising in TDR applications, as well as

many others<sup>3</sup>. The result of applying this denoising algorithm to the waveform in Figure 1 is shown in Figure 3.

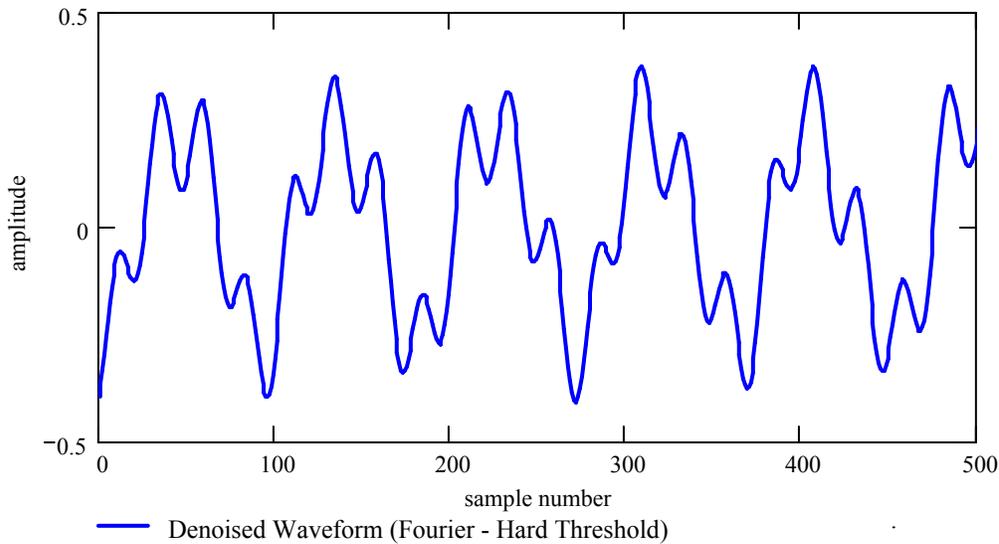


Figure 3 – Fourier Domain Denoised Waveform

## Hard Threshold Denoising Summary

In the last section, we looked at Fourier denoising techniques. While Fourier techniques are not suitable for TDR applications, there are many similarities between these and other techniques. In order to highlight these similarities, we prefer to generalize the concept of computing the DFT by describing this as *decomposing the waveform into a set of coefficients where each coefficient represents a value of a basis function in the set of functions described by the decomposition*. In the case of Fourier methods, the DFT is such a decomposition. Each coefficient of the DFT describes the complex size of a phasor or said differently, the amplitude and phase of a sinusoid. The coefficient index refers to the frequency of the sinusoid according to [11].

Therefore, the steps of denoising can be described as follows:

1. Acquire a time-domain waveform.
2. Decompose the waveform into the new domain for denoising.
3. Estimate the value of the noise in each coefficient of this decomposition.
4. Determine a threshold for each coefficient by multiplying the expected value of the noise in each coefficient by some value (usually 4-5).
5. Apply this threshold in a hard thresholding operation by keeping all components above the threshold and zeroing the coefficients below this threshold.
6. Compute the inverse decomposition to generate the denoised time-domain waveform.

## The Inapplicability of Fourier Denoising to TDR

As we have seen up to this point, Fourier denoising is simply a subset of hard threshold denoising in only one manner: the choice of the basis functions for the new domain formed by the decomposition. In Fourier denoising, the basis functions are sinusoids and the decomposition function determines amplitude and phase of these basis functions through the use of the discrete Fourier transform.

The choice of decomposition method chosen such as the DFT insofar as denoising is concerned is based on one thing alone – the ability of the decomposition to separate signal from noise. In the example we provided, the time-domain waveform contained no such capability for separation – each sample point of the waveform contains both signal and noise and each are indistinguishable. In the example shown, the DFT was able to separate *most* of the noise from the signal. We say most because in the frequency bins of interest where signal was located, each bin also contained noise, but most of the noise was spread in a manner where it was easy to eliminate very large amounts of noise because the actual components of interest were very small compared with all of the components present. This, by the way is why denoising is always related to the concept of compression. By keeping only the components of interest in the DFT, we compressed a very large time-domain waveform into a much smaller frequency-domain waveform.

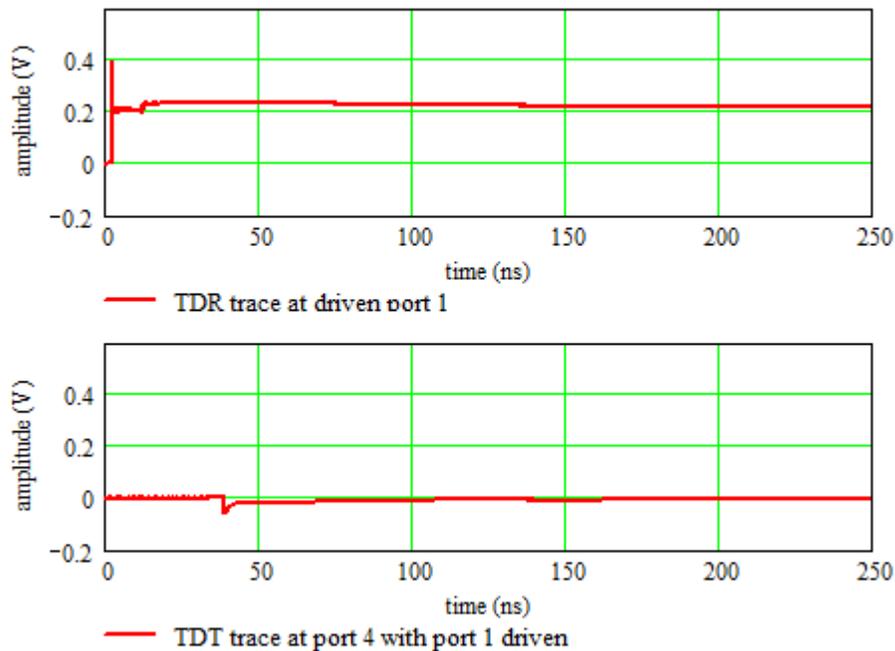
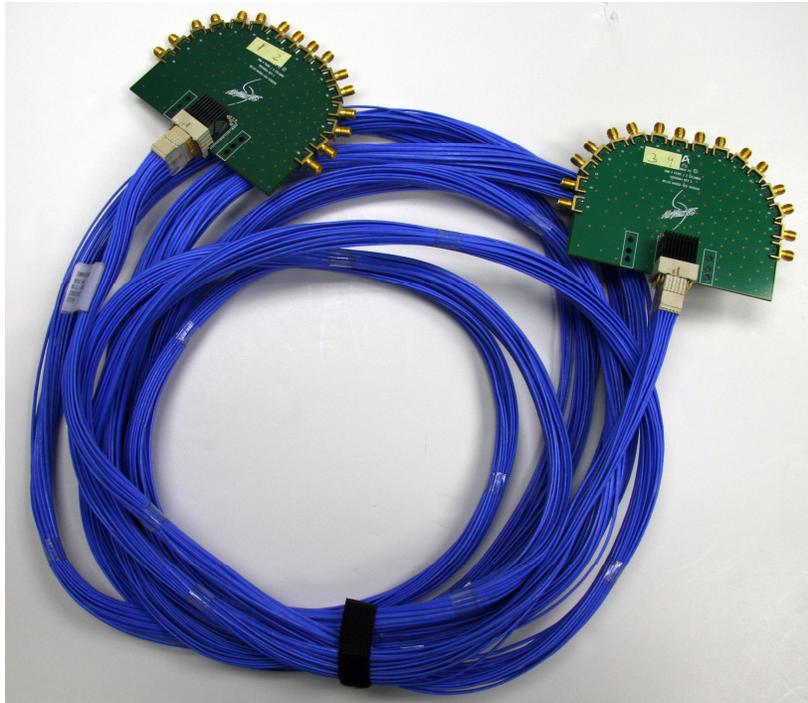


Figure 4 – TDR Waveforms Acquired for Cable Crosstalk Measurements

Therefore, the applicability of Fourier based denoising methods depends on most of the noise moving to bins where there are no frequencies of interest and the number of frequencies of interest being small relative to the total number (i.e.  $frac$  in [2] being small). In TDR, this is simply not the case.

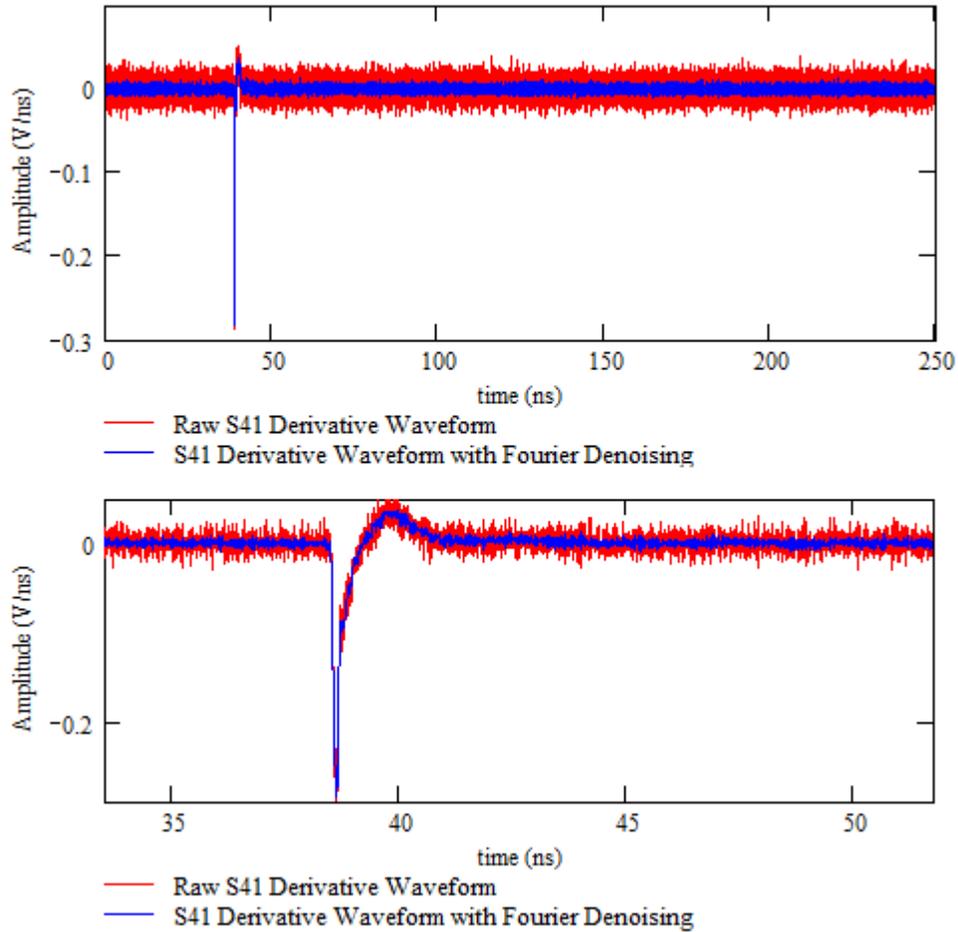
For the remainder of this paper, we will illustrate all points using an example cable far-end crosstalk (FEXT) measurement. Figure 4 shows TDR and TDT traces associated with such a measurement. The DUT is long cable, as shown in Figure 5 whose purpose is to transmit a differential signal whose positive and negative single-ended ports are on the near end, ports 1 and 2, and on the far end, ports 3 and 4 respectively. Therefore, the TDT trace at port 4 from a driven port 1 represents the AC coupled far-end crosstalk signal due to the incident wave front shown as the TDR trace on the driven port 1. With such a long cable being used, we'd expect this crosstalk signal to be very small and because of the length we expect to acquire a long waveform which presents the maximum challenge in dynamic range.



**Figure 5 – DUT used for example**

When taking TDR based s-parameter measurements, one separates the very short incident portion of the TDR waveform and converts to the frequency domain representation of the incident using the DFT. The remainder of the TDR waveform is converted similarly to form the reflect frequency domain representation. The entire TDT waveform is converted similarly to form the transmitted frequency domain representation. To form, what we call *raw measured s-parameters*, the incident frequency response is divided into both the reflected and transmitted frequency response. We will consider only these raw s-parameters in this paper with the understanding that various calibration algorithms are utilized to form true s-parameter measurements because the dynamic range aspect does not require this further complication.

While many methods in TDR are utilized to avoid computing a derivative of the TDR waveform<sup>4,5</sup>, it turns out that while noise is certainly amplified by the derivative, the signal content is amplified similarly with no net noise increase. In other words, since the energy in the incident waveform is really contained in the rising edge (i.e. the impulse formed by taking the derivative of the edge) and is dropping at 20 dB/decade due to the step-like nature of the waveform, the derivative normalizes the 20 dB/decade drop, and amplifies the noise similarly.

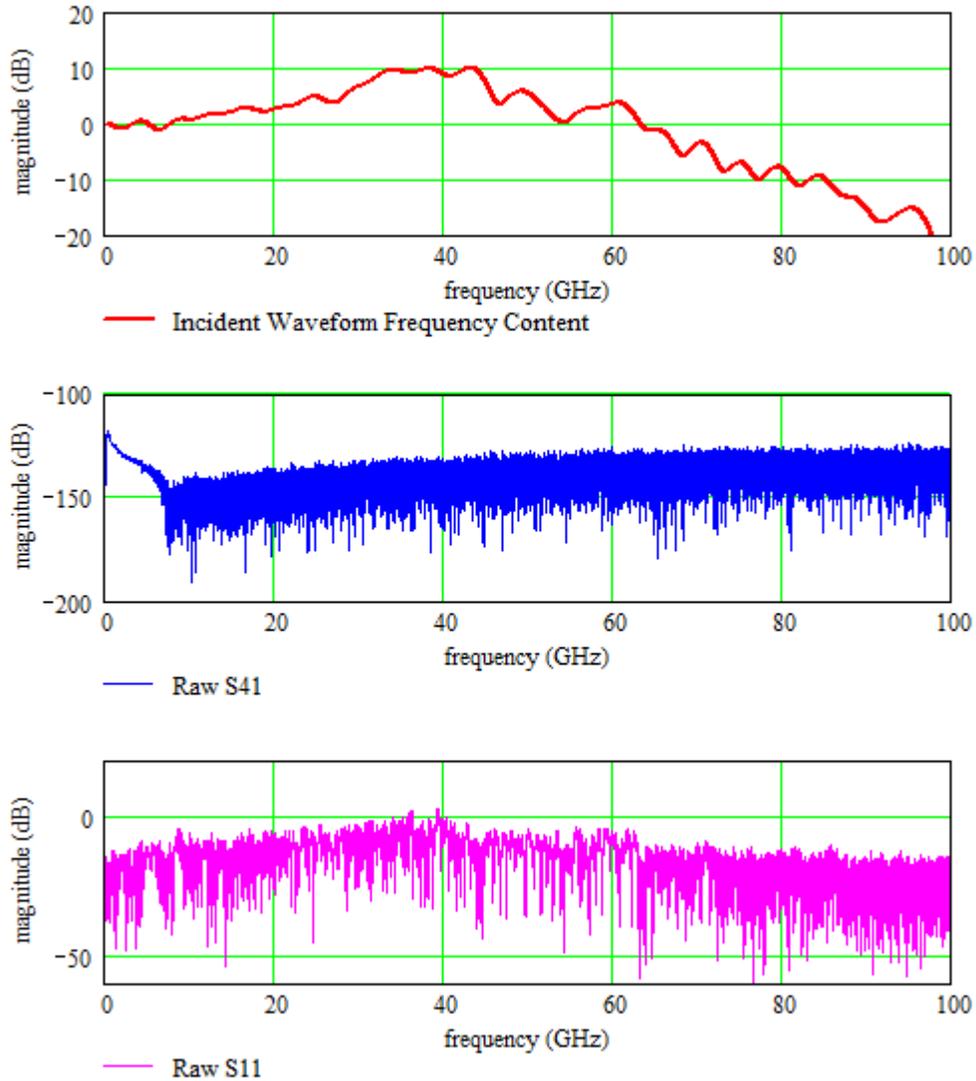


**Figure 6 – Crosstalk Derivative Waveforms used in TDR**

The derivative waveform of the step-like waveform at port 4 with port 1 driven (as shown in Figure 4) is shown in Figure 6 where we see a localized feature with large amounts of noise in the remainder of the waveform. Frequency responses formed by computing the DFT of the incident and reflect waveform at port 1 and the thru waveform at port 4 while driven at port 1 are shown in Figure 7. Here we see particularly in the raw measured S31 waveform (the crosstalk waveform) the large noise problem in the frequency domain.

For this particular application we require frequency content only out to 40 GHz (we are computing 40 GHz s-parameters) and we realize that there might be some opportunity to apply Fourier domain techniques by simply filtering the data out above 40 GHz. In this particular instrument, we are sampling at approximately 200 GS/s with a Nyquist rate of approximately 100 GHz. The waveform filtered in this manner is shown in blue in Figure 6. Here is where we might fool ourselves into thinking we did something worthwhile and we might develop some false ideas. In fact, although the blue waveform in Figure 6 is significantly cleaner from a time-domain perspective, if our goal was to compute s-parameters, the noise above 40 GHz would have no effect anyway, and the filtering operation did nothing to solve this problem. Another false idea would be to consider the shape of the noise as white and try to sample at higher sample

rates to spread more of the noise above 40 GHz. In other words, if we doubled the sample rate, we'd move more noise into unused regions between 40 GHz and the Nyquist rate. This would certainly be true if the noise were white, but remember, this would double the number of points in the record for a given acquisition window which would halve the acquisition speed and therefore halve the number of averages that could be performed on the waveform, thus nullifying the noise spreading effect.



**Figure 7 – Frequency Content of Crosstalk Derivative Waveforms**

In fact, what we see here is that Fourier domain techniques are not applicable to TDR because of the spread-spectrum nature of the energy in the stimulus waveform as well as the reflected and thru waveform responses.

This is where wavelets come in.

## What are Wavelets?

There are many types of wavelets. Here, we will constrain all discussion of wavelets to a particular, popular type called Daubechies wavelets<sup>6</sup>. This type of wavelet is an extension of the Haar wavelet<sup>7</sup>. Daubechies wavelets are a set of wavelets that are all formed in a similar manner which will be shown and differ only in the number of coefficients which will also be explained. Thus, a particular Daubechies wavelet is described by a coefficient count (where the Haar wavelet is a two coefficient wavelet). Here, we will consider wavelets whose number of coefficients is a power of two, wavelet transforms of time domain sequences that are a power of two, and like the Fourier transform, are assumed to repeat in time.

Wavelets allow decomposition of a time domain signal into a new set of basis functions. To follow this, consider that a discrete time domain representation of a signal is a set of coefficients, one per sample, where the coefficient value represents the size of a unit sample (or an impulse in continuous time). In a wavelet domain representation, each real valued coefficient represents the size of a wavelet function.

A wavelet decomposition contains the same number of coefficients as the time-domain waveform, but is broken into frequency bands called *scales* so named because each scale represents an octave greater frequency content. The scales of a wavelet transform are arranged such that the last scale contains for the most part the upper half of the frequency content of the signal, the next to last scale contains for the most part the preceding quarter of the frequency content, the scale before that contains for the most part the preceding eighth of the frequency content, etc. The last scale contains half the total coefficients; the next to last scale contains a quarter, the scale before that contains an eighth, etc. Each coefficient in each scale represents the size of a time-delayed wavelet function which is different for each scale, mostly in time length.

The wavelets represented by the coefficients in the last scale are very short in time duration, while the wavelets represented in the first scale span the entire length of the waveform. Thus, the first scale has the highest frequency resolution and the lowest time resolution, while the last scale has the lowest frequency resolutions and the highest time resolution.<sup>8</sup>

This can be seen through an example represented in Figure 8. Here, we show the inverse discrete wavelet transform (IDWT) of various unity wavelet coefficients in a 256 point wavelet domain waveform. We use a four-coefficient Daubechies wavelet which is perhaps the most popular wavelet aside from the Haar wavelet which is most used for educational purposes. Because the waveform is 256 points, we know from the previous discussion that there are 128 values possible in the last scale, 64 in the one before that, 32 in the one before that, etc. until we reach 8 values which are split into 4 and 4. Thus, each scale has 4, 4, 8, 16, 32, 64, and 128 values for the total of 256. The scales in Figure 8 are numbered 1 through 7 for each scale. The picture of scale 1 is the wavelet formed by a single unity wavelet coefficient. For the subsequent scales, we show three wavelets, each formed by an adjacent unity wavelet value to the left (red) and to the right (blue) to a unity wavelet value centered in the middle of the scale (magenta). Note that the wavelets in each scale look somewhat similar, but vary greatly in time length. For example, in scale 1, the wavelet spans the entire signal while in scale 7, a wavelet spans only four samples.

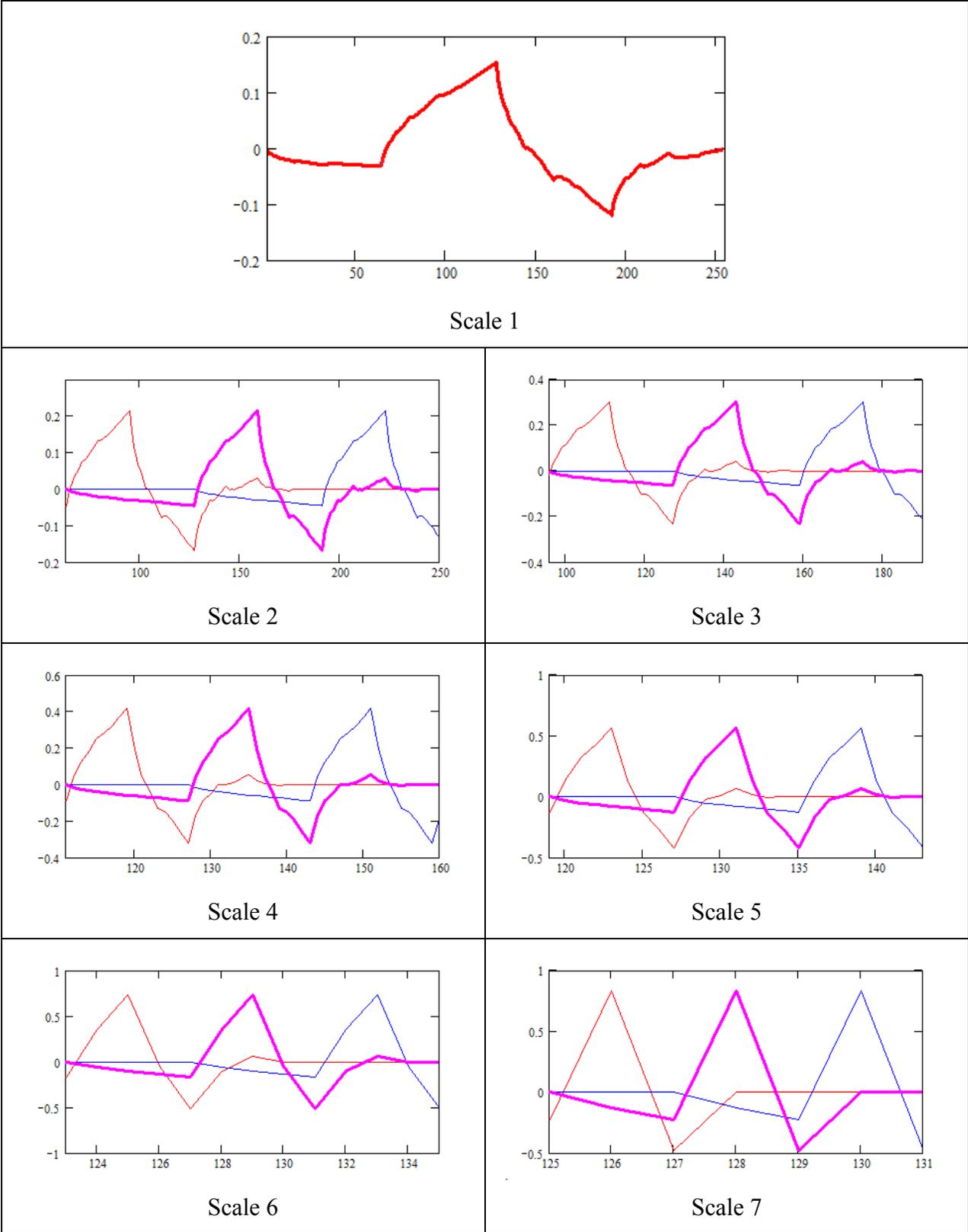


Figure 8 – Daubechies 4-tap wavelets

Decomposition of a time-domain waveform into a wavelet domain waveform is the act of determining the coefficient values of these wavelet functions such that when the wavelet functions are scaled by the coefficients and summed, the original time-domain waveform is formed.

The question that must be on the top of the reader’s mind is how this is performed, which must start with the description of the wavelet functions themselves. In fact, for Daubechies wavelets, there is no closed-form wavelet function (i.e. no non-iterative description of the wavelet functions) except for the wavelet in the last scale. Instead, the wavelet functions are a side-effect of the mechanics used in performing the transform.

To describe these mechanics, let’s stay with Daubechies four-coefficient wavelet. It is described by a *scaling* filter with four coefficients  $\mathbf{h} = (0.483, 0.837, 0.224, -0.129)$ . The scaling filter is simply a low-pass filter described by these coefficients. For example, for a Daubechies wavelet with coefficients  $\mathbf{h}$ , the low-pass filter function has frequency response of [4] where  $f$  is the fraction of the sample-rate and  $k$  for each coefficient value:

$$H(\mathbf{h}, f) = \frac{1}{\sqrt{2}} \sum_k h_k \cdot e^{-j2\pi kf}$$

[4]

Thus, the scaling filter response for Daubechies four-tap wavelet is shown in Figure 9. The quadrature-mirror filter (QMF) that corresponds to the scaling filter is called the *detail* filter which is also shown in Figure 9. The taps of the QMF high-pass filter are generally represented by the vector  $\mathbf{g}$ .

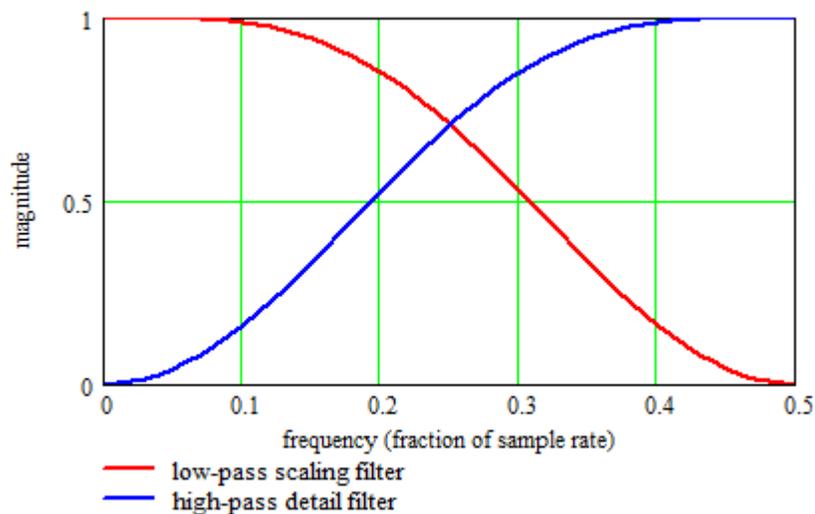


Figure 9 – Daubechies four-tap wavelet filter responses

For reasons you’ll understand soon, the wavelet function shown in scale 7 of Figure 8 (and always the last scale of the wavelet transform) corresponds to the coefficients in  $\mathbf{h}$  that are reversed in time and whose sign is alternately changed. In other words, if we have a  $K$ -tap

wavelet described by the coefficients of the scaling filter  $\mathbf{h}$ , then the wavelet function in the last scale of the wavelet transform is described as  $\mathbf{w}$  where  $k \in 0 \dots K - 1$ :

$$w_k = h_{K-1-k} \cdot (-1)^k$$

[5]

With the concept of the scaling filter  $\mathbf{h}$  and detail filter  $\mathbf{g}$  in mind, the wavelet transform is computed according to the algorithm schematically shown in Figure 10, where here  $K$  refers to the number of samples in the time-domain waveform,  $L$  refers to the length of the scaling filter, and  $B$  represents the number of scales in the result.

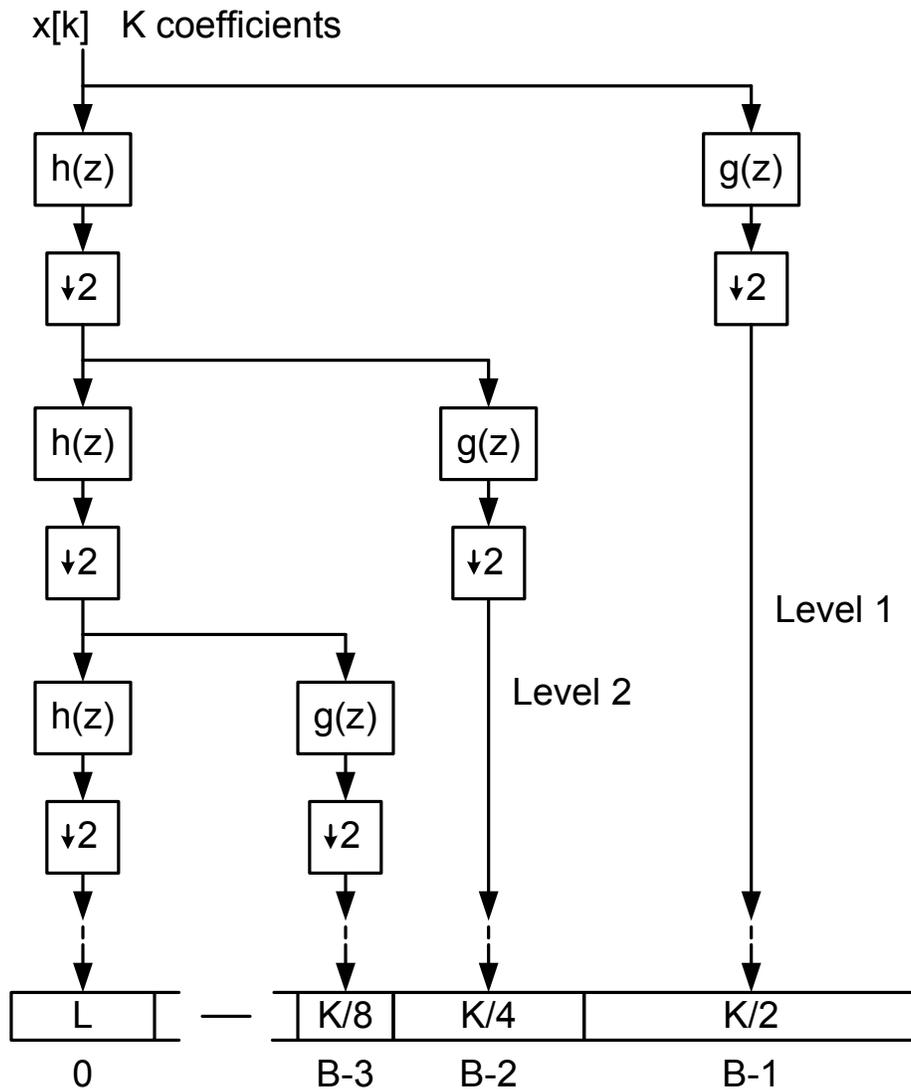


Figure 10 – Schematic Representation of Discrete Wavelet Transform

To compute the DWT, first, the input waveform  $\mathbf{x}$ , a vector of  $K$  points is passed through the detail filter with coefficients  $\mathbf{g}$ . The result is decimated, or down sampled by 2 which results in  $K/2$  coefficients. These are called *level 1* coefficients and represent the coefficients in the last scale. The input waveform is passed through the scaling filter with coefficients  $\mathbf{h}$  and also decimated by 2, which halves the number of points in the time-domain waveform presented to level 2. The process is continued until the time-domain waveform is the same length as  $\mathbf{h}$  and this result is placed into the first scale. It is because of these mechanics in the computation of the DWT that there is no closed-form wavelet function and that the wavelet function in the last scale is represented by [5].

## Properties of Wavelets

As previously mentioned, wavelets can be thought of loosely as basis functions that describe a waveform. The wavelet transform determines the coefficients applied to these basis functions. Wavelet transforms utilizing Daubechies wavelets, like the Fourier transform, provide coefficients of *orthogonal* functions. Orthogonality means that the coefficients in the transform represent functions that are as different from each other as possible. It also means that noise is not amplified moving in and out of the transform. The DWT implemented in Figure 10 has an inverse IDWT process that utilizes the same analysis filters as the DWT, also similar to the Fourier transform. Finally, the process is a lossless process.

Similar to the Fourier transform, the wavelet transform offers possibilities for compression and therefore denoising. This is possible because of the following items:

1. The effect of the wavelet transform on noise and noise shapes can be understood.
2. Thresholds can be formed that distinguish desirable wavelet coefficients from undesirable coefficients

Unlike the Fourier transform, however, wavelets offer possibilities for decomposition of the waveform not only by frequency, but also spatially in time. We discussed already the inapplicability of Fourier denoising techniques because the Fourier transform considers only frequency, and the TDR waveforms do not have unused frequency locations. This being said, we can imagine by considering [1] and more importantly the derivation of dynamic range in *Appendix A - Derivation of Dynamic Range* that there are unused, or unimportant time locations in a waveform. Since the wavelet transform separates waveforms in both time and frequency, while also having predictable effects on noise, it is the ideal transformation for separating TDR signal and noise.

## Wavelet Denoising Strategy

The reason why we showed Fourier domain denoising strategies earlier in this paper is because wavelet denoising strategies are exactly analogous. In the case of wavelet denoising of TDR, the DWT of the derivative waveform is computed and a threshold is determined to filter out statistically insignificant wavelet coefficients. Fortunately, the statistics of noise in the wavelet domain are the same as in the time domain. In other words, the standard deviation of a DWT of a waveform containing Gaussian noise is the same as that of the time domain waveform. Unfortunately, the spectral shape of the noise undergoes a complicated transformation that must be accounted for if the noise is not white; the derivative operation ensures that this is not the case

for TDR and in fact has a spectral shape shown in [18] which is approximately linear as shown in [19], which becomes approximately log-linear when considering the effect on wavelet scales. The application of the spectral noise shape to the hard threshold determination<sup>9</sup> is shown in Figure 11. Here, we see that the majority of coefficients in the DWT fall under the threshold placed at 5 standard deviations of the noise estimated from the last 30 percent of the last wavelet transform scale. In fact in this example, only approximately 140 out of 65,000 coefficients are retained for a compression amount of 99.8%. The computation of the IDWT of these remaining coefficients provides the result of the denoising operation in Figure 12 where we see that the noise is removed preserving the structure of the features of interest for TDR.

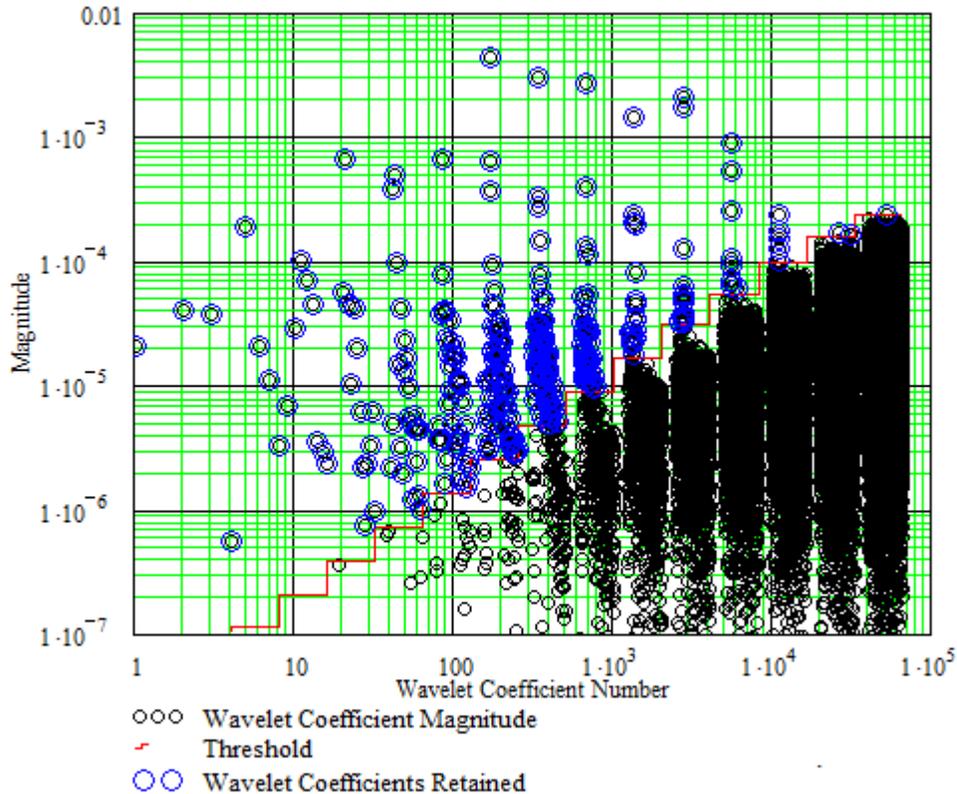


Figure 11 – Wavelet Denoising Hard Threshold Strategy

To see the magnitude of the time domain effect, consider Figure 13 which shows the absolute magnitude of the derivative waveform. Here we see the original noise at levels of around 50 mV/ns. By considering only the frequency content from 0 to 40 GHz, we see that this noise was reduced significantly, but the result of the wavelet denoising reduced this to levels that are about 1000 times lower.

The effect of the wavelet denoising is seen clearer in the final goal of denoising, the raw S31 crosstalk measurement in the frequency domain. This is shown in Figure 14 where we see the crosstalk measurement significantly cleaned up in the region from 0 to 5 GHz, and the measurement dropping as low as -100 dB.

The amazing result of the frequency content of the noise removed is also shown in Figure 14, where we see the noise shape approximately that of the derivative effect. Most importantly, this noise is removed over the entire frequency band, even underneath the low frequency area where there is significant signal content – something a Fourier denoising technique is simply incapable of.

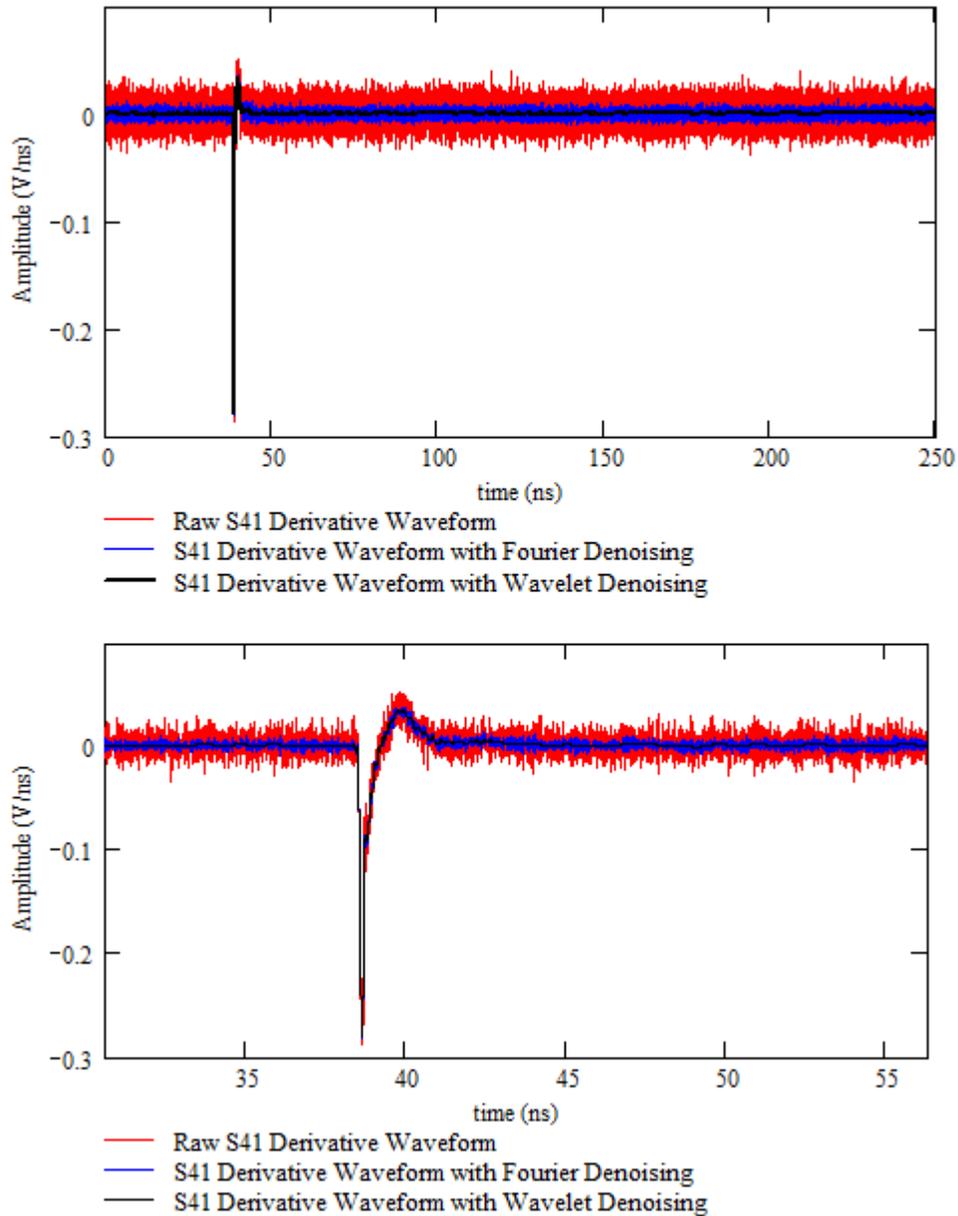


Figure 12 – Comparison of Denoised Crosstalk Derivative Waveforms

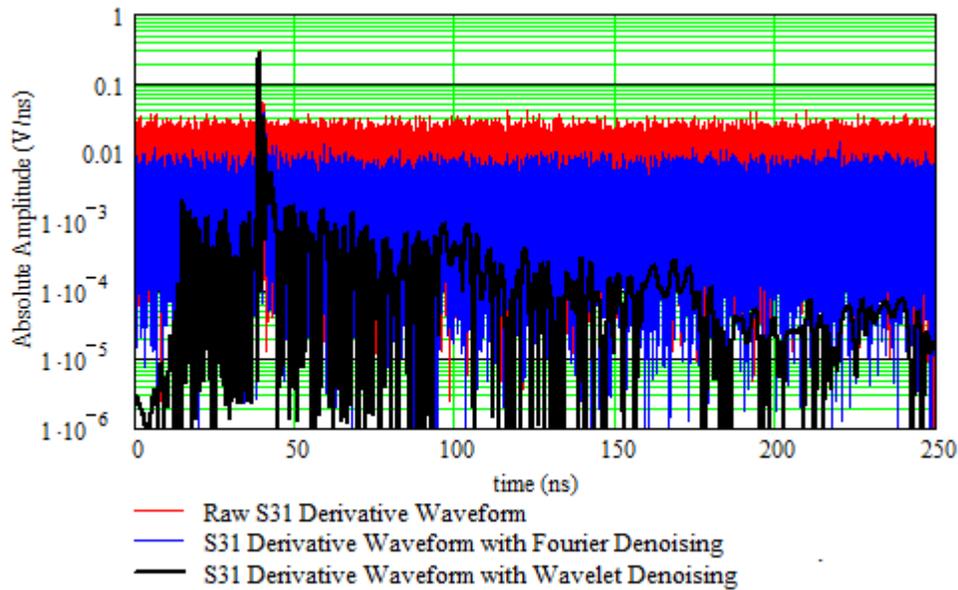


Figure 13 – Absolute Magnitude Comparison of Denoised Crosstalk Waveforms

## Considerations of Results

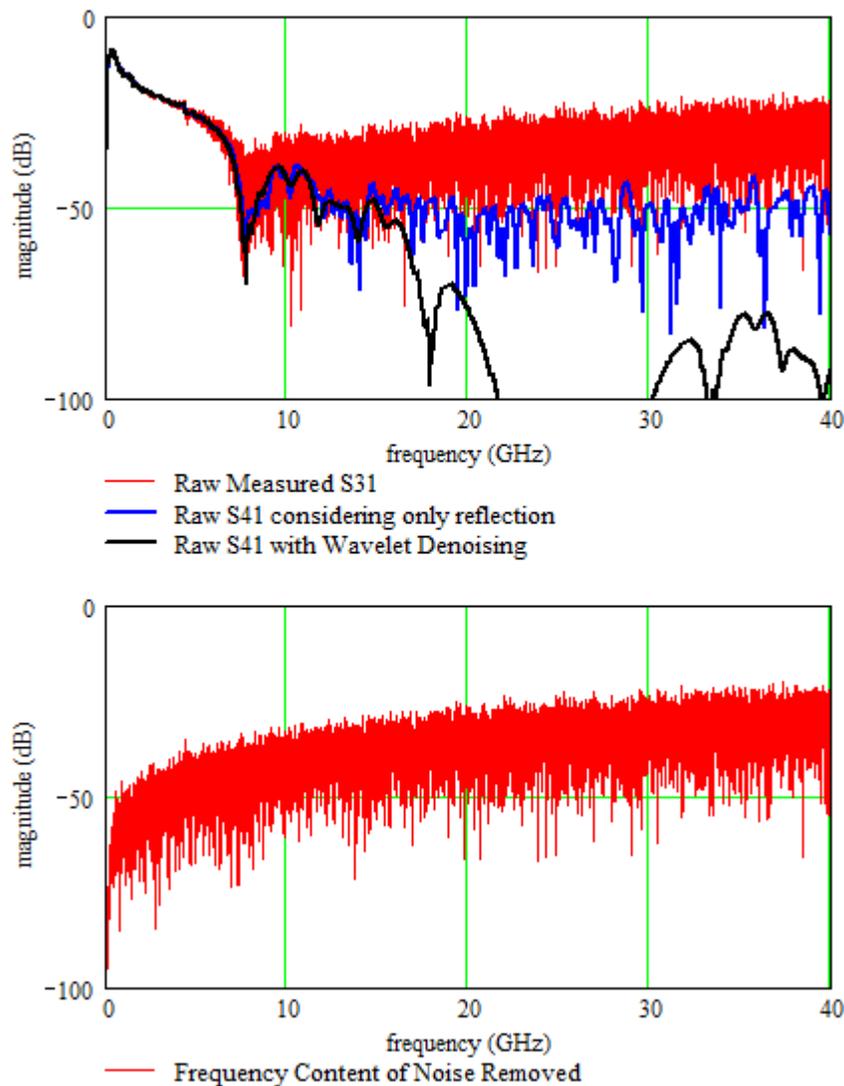
There are many considerations to make regarding wavelet denoising techniques. Most of the considerations surround the believability of the results. Figure 15 shows a comparison of results to the vector network analyzer (VNA) which is known to have higher dynamic range when compared to TDR instruments. Here we had a problem in the time-domain comparison because the VNA data looks unbelievable for S41. Note that we used some fixturing in the VNA measurement that resulted in about a 7 dB dynamic range impairment, but used a narrow 100 Hz IF bandwidth which caused the sweeps to take a long, but tolerable amount of time.

Other than VNA compares, there are ways to develop some expectations on dynamic range improvement of wavelet denoised results. The most important is to return to [2] where we consider the fractional portion of the TDR waveform containing actual reflections. Here we have been considering a 250 ns TDR acquisition which is a very long acquisition. The acquisition is long because the cable itself is very long. Unless physics is being violated, we know that nothing can arrive out of the other end of the cable until around 38 ns. Furthermore, the duration of the actual waveform portion that comes out is about 2-3 ns. This is 2.5 out of 250 or about 1% which leads to at least a 20 dB improvement in dynamic range if we only considered this 2 ns interval. This means that regardless of believe or disbelief in the concept of wavelets, it's reasonable to at least expect a 20 dB dynamic range improvement for this measurement. This is shown in Figure 14 where we consider this reflection “gating”.

Another consideration regards resonant structures. The DFT is designed to find sinusoids in everything as that is its decomposition basis function. The DWT is designed to find wavelets and wavelet like shapes. Since TDR is designed to find reflections and the derivative of the step-like reflections are very wavelet-like, wavelet denoising seems like a logical method. But TDR does not perform well with very narrow resonant structures, and wavelet decomposition probably does more to discard resonance information than to preserve it. After all, in order to preserve it,

it needs to build a sequence of wavelets that would all sum to a sinusoid at the resonant frequency while simultaneously keeping all of these wavelets above the noise floor in the DWT. While this has not been studied here, it seems like an unlikely scenario. Perhaps in future papers these ideas will be studied and perhaps there are hybrid techniques between wavelets and Fourier decomposition that are appropriate.

Regardless of the interpretation of these remarkable results, it is clear that simulations and other signal-integrity related activities that surround the time-domain behavior of the measured frequency-domain behavior in terms of s-parameters will benefit greatly from noise removed – noise that is clearly present in the measurement example used. This is clear in the S31 time domain measurement in Figure 15 where the wavelet denoised time-domain waveform is cleaner even than the VNA.



**Figure 14 – Frequency Content of Denoised Crosstalk Measurement with Content of Removed Noise**

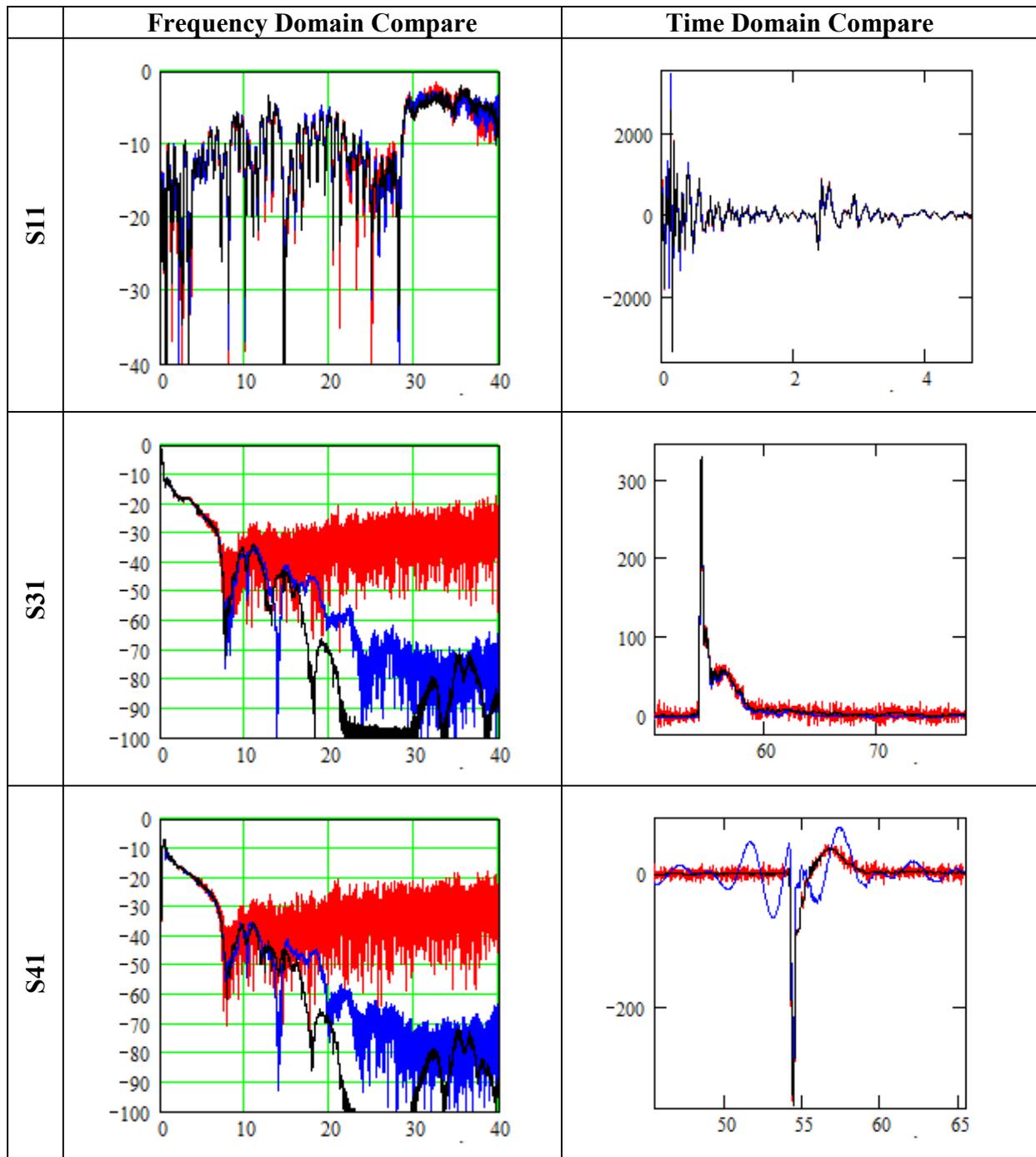


Figure 15 – Comparison of Not Denoised (red), VNA (blue), and Wavelet Denoised TDR (black)

## Summary

A technique was shown for denoising waveforms utilized in TDR measurements. This technique was found to remove large amounts of noise in measurements and was shown to be superior to other denoising possibilities. Some considerations were provided for interpreting these results. Further study is needed to understand the power of the results and possibly extend the methods presented.

## Appendix A - Derivation of Dynamic Range

In time-domain reflectometry, we acquire step waveforms, therefore we start with an acquired signal defined as follows:

$$w_k = s_k + \varepsilon_k \quad [6]$$

where  $w$  is the step waveform actually acquired,  $s$  is the step portion contain the signal of interest, and  $\varepsilon$  is the noise signal which we assume to be white, normally distributed, uncorrelated noise.

The signal content in the step is in the form of the frequency content of the derivative, so the derivation must consider this. Since during calculation we don't know the difference between the noise and the step, we must take the derivative of both. We will be approximating:

$$\frac{d}{dt} w(t) = \frac{d}{dt} (s(t) + \varepsilon(t)) = \frac{d}{dt} s(t) + \frac{d}{dt} \varepsilon(t) = x(t) + \frac{d}{dt} \varepsilon(t) \quad [7]$$

When we convert the two signals we are interested in to the frequency domain:

$$\begin{aligned} X &= DFT(x(t)) \\ DN &= DFT\left(\frac{d}{dt} \varepsilon(t)\right) \end{aligned} \quad [8]$$

We will calculate the dynamic range as a signal-to-noise ratio (SNR) and define this for each frequency as:

$$SNR_n = \frac{X_n}{DN_n} \quad [9]$$

In order to calculate the SNR, we calculate the frequency content of each of these components separately and take the ratio. We start with the noise component.

Given a noise signal  $\varepsilon$  which contains only uncorrelated, normally distributed, white noise, it has a mean of 0 and a standard deviation of  $\sigma$ , which is the same as saying it has an root-mean-square (rms) value of  $\sigma$ . We have  $K$  points of this signal  $\varepsilon_k$ ,  $k \in 0 \dots K-1$ .

If we calculate the discrete-Fourier-transform (DFT) of this noise signal, we obtain

$N+1$  frequency points  $N = \frac{K}{2}$ ,  $n \in 0 \dots N$ :

$$E_n = \frac{1}{K} \sum_k \varepsilon_k e^{-j2\pi \frac{nk}{K}} \quad [10]$$

where the frequencies are defined as:

$$f_n = \frac{n}{N} \frac{Fs}{2} \quad [11]$$

Where  $Fs$  is the sample rate.

By the definition of the rms value and by the equivalence of noise power in the time domain and frequency domain, we know the following:

$$\sqrt{\frac{1}{K} \sum_k \varepsilon_k^2} = \sigma = \sqrt{\sum_{n=0}^{N_{bw}} \left( E_n \cdot \frac{2}{\sqrt{2}} \right)^2} \quad [12]$$

$N_{bw}$  is the last frequency bin containing noise due to any band limiting of noise effects.

We define an average value of  $Ea$  that satisfies this relationship:

$$\sqrt{\sum_n \left( Ea \cdot \frac{2}{\sqrt{2}} \right)^2} = \sigma = \sqrt{N_{bw} \cdot \left( \frac{Ea \cdot 2}{\sqrt{2}} \right)^2} \quad [13]$$

And therefore:

$$Ea = \frac{1}{\sqrt{\frac{f_{bw}}{Fs/2}}} \cdot \frac{\sigma}{\sqrt{K}} \quad [14]$$

$f_{bw}$  is the frequency limit for the noise calculating by substituting  $N_{bw}$  for  $n$  in [11].

We, however, are taking the derivative of the signal. The derivative in discrete terms is defined as:

$$\frac{d}{dt} \varepsilon(t) \approx d\varepsilon_k = \frac{\varepsilon_k - \varepsilon_{k-1}}{Ts} \quad [15]$$

Where  $Ts = \frac{1}{Fs}$  is the sample period. Using the same equivalence in [12] and defining

$DN = DFT(d\varepsilon)$ , we have:

$$\sqrt{\frac{1}{K} \sum_k d\varepsilon_k^2} = \sigma = \sqrt{\sum_n \left( DN_n \cdot \frac{2}{\sqrt{2}} \right)^2} \quad [16]$$

Using the Z-transform equivalent of the derivative in the frequency domain, and an average value for the noise in  $DN$  it can be shown that:

$$\sqrt{\frac{1}{K} \sum_k d\varepsilon_k^2} = \sigma = \sqrt{\sum_n \left( \frac{\left| 1 - e^{-j2\pi \frac{f_n}{F_s}} \right|}{T_s} \cdot Ea \cdot \frac{2}{\sqrt{2}} \right)^2}$$

[17]

Therefore, the average noise component at each frequency is given by:

$$\overline{DN}_n = \frac{\left| 1 - e^{-j2\pi \frac{f_n}{F_s}} \right|}{T_s} \cdot Ea \cdot \frac{2}{\sqrt{2}}$$

[18]

We can make an approximation that allows one to gain further insight by expanding the numerator term in a series expansion:

$$\left| 1 - e^{-j2\pi \frac{f}{F_s}} \right| = \frac{2\pi f}{F_s} + O\left(\left(\frac{f}{F_s}\right)^3\right)$$

[19]

Which allows us to approximate the noise component as:

$$\frac{2\pi f_n}{F_s} \cdot Ea \cdot \frac{2}{\sqrt{2}} = \overline{DN}_n = \frac{2\pi f_n \sigma \sqrt{2}}{\sqrt{K} \sqrt{\frac{f_{bw}}{F_s/2}}}$$

[20]

Now that we have the noise component of dynamic range, we move to the signal component.

Without regard to the rise time or the frequency response of the step, which we will consider later, we define the signal such that, in the discrete domain, the integral of the signal forms a step:

$$s_k = s_{k-1} + x_k \cdot Ts$$

[21]

$x$  is an impulse such that  $x_0 = \frac{A}{Ts} = A \cdot Fs$  and is zero elsewhere such that  $s$  forms a step that rises to amplitude  $A$  at time zero and stays there.  $X = DFT(x)$  and therefore the signal components at each frequency is defined as:

$$X_n = \frac{A}{T_s} = A \cdot F_s \quad [22]$$

Again, to gain further insight, we define:

$$K \cdot T_s = \frac{K}{F_s} = T_a \quad [23]$$

Where  $T_a$  is the acquisition duration. Therefore:

$$X_n = \frac{A}{T_a} \quad [24]$$

Using [9], the ratio can therefore be expressed as:

$$SNR_n = \frac{X_n}{DN_n} = \frac{A \cdot \sqrt{K} \cdot \sqrt{f_{bw}}}{T_a \cdot 2 \cdot \pi \cdot f \cdot \sigma \cdot \sqrt{F_s}} \quad [25]$$

Since these are voltage relationships, we can express the SNR in dB as:

$$SNR_n = 20 \cdot \text{Log} \left( \frac{A \cdot \sqrt{K} \cdot \sqrt{f_{bw}}}{T_a \cdot 2 \pi \cdot f \cdot \sigma \cdot \sqrt{F_s}} \right) = 10 \cdot \text{Log} \left( \frac{A^2 \cdot K \cdot f_{bw}}{T_a^2 \cdot 4 \cdot \pi^2 \cdot f^2 \cdot \sigma^2 \cdot F_s} \right) \quad [26]$$

And using [23], finally:

$$SNR_n = 10 \cdot \text{Log} \left( \frac{A^2 \cdot f_{bw}}{T_a \cdot 4 \cdot \pi^2 \cdot f^2 \cdot \sigma^2} \right) \quad [27]$$

We would like to express the noise in dBm, so we have:

$$Noise_{dBm} = 20 \cdot \text{Log}(\sigma) + 13.010 = 10 \cdot \text{Log}(\sigma^2 \cdot 20) \quad [28]$$

And therefore:

$$\sigma^2 = \frac{10^{\frac{Noise_{dBm}}{10}}}{20} \quad [29]$$

Substituting [29] in [27]:

$$SNR_n = 10 \cdot \text{Log} \left( \frac{A^2 \cdot f_{bw} \cdot 20}{Ta \cdot 4 \cdot \pi^2 \cdot f^2 \cdot 10^{\frac{Noise_{dBm}}{10}}} \right) = 10 \cdot \text{Log} \left( \frac{A^2 \cdot f_{bw} \cdot 20}{Ta \cdot 4 \cdot \pi^2 \cdot f^2} \right) - Noise_{dBm}$$

[30]

Then, to clean things up, we extract some constants:

$$10 \cdot \text{Log} \left( \frac{20}{8 \cdot \pi^2} \right) = -6$$

[31]

And therefore:

$$SNR_n = 10 \cdot \text{Log} \left( \frac{2 \cdot A^2 \cdot f_{bw}}{Ta \cdot f^2} \right) - Noise_{dBm} - 6$$

[32]

Now let's consider some other factors. First, that there is a frequency response of the pulse, and a frequency response of the sampler. These responses can be aggregated into a single response. Since, in decibels, it is simply the frequency response of the step calculated by taking the DFT of the derivative of the step (isolating only the sampled incident waveform) and calculating in dB, this value can simply be added to the dynamic range:

$$SNR_n = 10 \cdot \text{Log} \left( \frac{2 \cdot A^2 \cdot f_{bw}}{Ta \cdot f^2} \right) - Noise_{dBm} + P(f) - 6$$

[33]

Next, we consider the effects of averaging. Averaging the waveform achieves a 3 dB reduction in noise with every doubling of the number of averages. This leads to an improvement of dynamic range by:

$$20 \cdot \text{Log}(\sqrt{avg}) = 10 \cdot \text{Log}(avg)$$

[34]

Which allows us to insert this directly into the numerator:

$$SNR_n = 10 \cdot \text{Log} \left( \frac{2 \cdot A^2 \cdot f_{bw} \cdot avg}{Ta \cdot f^2} \right) - Noise_{dBm} + P(f) - 6$$

[35]

We really don't want to consider dynamic range in terms of number of averages and instead to prefer to consider the amount of time we are willing to wait. The amount of averages taken in a given amount of time is given by:

$$avg = \frac{Fs_{act}}{Ta \cdot Fs_{eq}} \cdot T$$

[36]

In [36], we now need to distinguish what is meant by sample rate.  $Fs_{eq}$  becomes the equivalent time sample rate and replaces what we previously called  $Fs$ .  $Fs_{act}$  is the actual sample rate of the system and  $T$  is the amount of time over which acquisitions are taken. Substituting [36] in [35], we obtain:

$$SNR_n = 10 \cdot \text{Log} \left( \frac{2 \cdot A^2 \cdot f_{bw} \cdot Fs_{act} \cdot T}{Ta^2 \cdot f^2 \cdot Fs_{eq}} \right) - Noise_{dBm} + P(f) - 6 \quad [37]$$

Next, we consider the losses in the cabling and fixturing between the pulser/sampler and the device-under-test (DUT). Where  $F(f)$  and  $C(f)$  are the loss in the fixturing and cabling as a function of frequency respectively, the equation for dynamic range becomes:

$$SNR(f) = 10 \cdot \text{Log} \left( \frac{2 \cdot A^2 \cdot f_{bw} \cdot Fs_{act} \cdot T}{Ta^2 \cdot f^2 \cdot Fs_{eq}} \right) - Noise_{dBm} + P(f) + 2 \cdot (C(f) + F(f)) - 6 \quad [38]$$

considering the fact that the signal must pass through the cabling and fixturing twice.

There is one final consideration. That is the effect of denoising algorithms. Denoising algorithms have the effect of removing broadband noise from the acquisition primarily through means of detecting where uncorrelated noise is present in the signal in time. It is difficult to quantify these effects, but a conservative method considers the fact that the primary noise reduction occurs where there are no reflections. In other words, if we look at a denoised waveform, the primary effect is to remove the noise in the locations in the waveform devoid of reflections. The effect on noise, again conservatively speaking, is to retain only the portion of the waveform that contains reflections. Here, we will assume that the noise remains in these portions. Thinking this way, we can define a variable *frac* that contains the fractional portion of the acquisition that actually contains reflections relative to the portion that does not. This value is DUT dependent and modifies the acquisition duration  $Ta$ . Of course,  $frac = 1$  is used when denoising is not employed:

$$SNR(f) = 10 \cdot \text{Log} \left( \frac{2 \cdot A^2 \cdot f_{bw} \cdot Fs_{act} \cdot T}{Ta^2 \cdot frac \cdot f^2 \cdot Fs_{eq}} \right) - Noise_{dBm} + P(f) + 2 \cdot (C(f) + F(f)) - 6$$

[39] – Dynamic Range Equation

\*\*\*\*\*

## References

---

- <sup>1</sup> P. Pupalaiakis, "SPARQ Dynamic Range", LeCroy technical brief, 2010, [http://www.lecroy.com/files/whitepapers/SPARQ\\_Dynamic\\_Range.pdf](http://www.lecroy.com/files/whitepapers/SPARQ_Dynamic_Range.pdf)
- <sup>2</sup> S. G. Chang, B. Yu and M. Vetterli, "Adaptive Wavelet Thresholding for Image Denoising and Compression", IEEE Transactions on Image Processing, Vol. 9, No. 9, Sep. 2000
- <sup>3</sup> S. Tsai, "Power Transformer Partial Discharge (PD) Acoustic Signal Detection using Fiber Sensors and Wavelet Analysis, Modelling, Simulation", Master's Thesis, Chap. 4, Virginia Tech, Dec. 2002
- <sup>4</sup> W. L. Gans & N. S. Nahman, "Continuous and discrete Fourier transform of step-like waveforms", IEEE Trans. Inst. & Meas., vol. IM-31, pp. 97-101, June 1982
- <sup>5</sup> A. M. Nicolson, "Forming the fast Fourier transform of a step response in time-domain metrology", Electron. Lett., vol.9, pp. 317-318, July 1973
- <sup>6</sup> I. Daubechies, "Ten Lectures on Wavelets", Society for Industrial and Applied Mathematics, 1992, ISBN 0-89871-274-2
- <sup>7</sup> A. Haar, "Zur Theorie der orthogonalen Funktionensysteme", Mathematische Annalen, 69, pp 331-371, 1910
- <sup>8</sup> R. Polikar, "The Wavelet Tutorial", <http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>
- <sup>9</sup> P. Pupalaiakis, K. Doshi and A. Sureka, "Wavelet Denoising for Time Domain Network Analysis", U.S. Patent application 12/890,832