DesignCon 2009

Analysis of Random Noise and the Effect of Band-Limited Noise on Stressed-Eye Receiver Tolerance Tests

Ransom Stephens, Ransom's Notes
[ransom@ransomsnotes.com]

Marcus Müller, Agilent Technologies
[marcus_mueller2@agilent.com]

Abstract

High rate serial-data technologies require that receivers be tested under the stress of calibrated levels of Gaussian Random Jitter (RJ) and/or voltage noise. Every standard assumes that noise follows a Gaussian distribution with a white frequency spectrum. Several emerging standards require that stress signals used to test receivers be filtered, PCI Express Gen 2 requires two separately filtered Gaussian noise signals. In this paper we determine the effect of frequency band-limiting on the distribution of ideal and commercially available, noise sources. We analyze and compare simulated and real noise sources under different conditions.

Author(s) Biography

Ransom Stephens' company, Ransom's Notes, produces and presents content at every level of technical sophistication to help engineers advance to technology's cutting edge. He is the author of more than 200 articles in the electronics industry, science journals, and magazines. Dr. Stephens is an expert in signal integrity analysis of electrical and optical systems. He has introduced new measurement techniques, invented methods for extracting signals from noise, led an engineering commando team, and served on high data-rate standards committees.

Marcus Müller is an R&D Lead Engineer with Agilent Technologies' High-Speed Digital Test segment and works on the development of new test and analysis methods based on bit error ratio testers. He joined Agilent in 2000, right after receiving his master's degree from Stuttgart University, Germany

Introduction

Receiver Tolerance Testing

The successful operation of a high speed link at multi GBit/s rates requires either a very well designed and clean channel, or a receiver architecture that is able to tolerate signal impairments such as crosstalk, jitter, and amplitude noise. Every successful communications and computer standard that has emerged in the last 5-10 years followed the latter approach: PCI Express, Serial ATA and 10 GbE are just a few examples of standards that require receivers to include devices such as clock data recovery circuits and equalizers. This allows them to work with signals that may in some cases be so distorted that they are not even recognizable as a digital signal.

The ability of a receiver to work with a degraded input signal is tested with a well defined worst case signal; during the 'receiver tolerance test', the device must operate at a low Bit Error Ratio (often less than 10^{-12}) in order to be compliant to the standard. Traditionally, going back to SONET, the test signal included just a sinusoidal and a random jitter term: the frequency and amplitude of the sinusoidal were varied to test the clock data recovery circuit, while the random part was kept constant and emulated the random jitter of a real system. However, as receivers have become more elaborate, so have the signals to test them: receiver tolerance specifications now usually call for amplitude noise (both sinusoidal and random) in addition to jitter, along with worst case band limited channels to apply Inter-Symbol Interference.

Additionally, the specifications for random jitter have become more complicated than just specifying its amplitude. The CEI-6G standard, for example, requires random jitter subject to a 10 MHz high pass filter which prevents it from being tracked by the clock data recovery circuit. Recently, PCI Express 2.0 went one step farther, asking for two different RJ sources of different amplitudes, one with a bandwidth from 10 kHz to 1.5 MHz and the other 1.5-100 MHz.

Gaussian White Noise

A Gaussian distribution is determined by two parameters, the mean μ and standard deviation σ. Its probability density function or, more or less equivalently, its time-domain distribution, is given by

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

The Gaussian distribution appears in virtually every area of science and engineering. It is simple and well behaved, and the central limit theorem – that the combination of a large number of small amplitude, independent random variables follows a Gaussian distribution – provides compelling reason to believe that it is the foundation of many distributions.

An important property of a Gaussian distribution is that it is unbounded: the probability that a random variable is larger than x is always greater than zero, regardless of how large x is. And because of symmetry, the same is true for the negative side. As a consequence, the peak-to-peak value of a Gaussian random process is infinity. However, once we observe a random process it becomes finite, and therefore has an effective peak-to-peak that is finite too; its magnitude is a function of probability. For example, random variables that follow a Gaussian distribution of $\mu = 0.0$ and $\sigma = 1.0$ occur with values larger than 7 and smaller than -7 with a probability of less than 10^{-12}.

A more useful measure than the peak-to-peak is the c*rest factor,* which describes the effective peak-to-peak span of a signal in terms of the standard deviation of the distribution. The crest factor is usually defined as the ratio of the observed peak value to the rms value of the distribution:

$$\text{crest factor} = \frac{V_{Peak}}{V_{rms}}$$

For distributions that can exhibit asymmetries, like jitter and noise, it is more appropriate to use the ratio of the observed peak-to-peak to the rms value:

$$\text{noise crest factor} = \frac{V_{\max} - V_{\min}}{V_{rms}}$$

The second important property of White Gaussian Noise is its frequency spectrum: the power spectral density is flat over the bandwidth of interest, and there are no dominating frequencies. For finite length signals however, this is true only for the average.

Scope of this Paper

Receiver tolerance specifications make two assumptions when they specify random jitter or level noise components: one, that electronic noise is in fact Gaussian White Noise, and second that band limiting does not significantly change its time domain distribution. We were interested primarily in verifying the second assumption: if it were false, then the standard approach to random jitter tolerance testing with a white noise source and a set of filters were false.

Numerical Experiments

Most of the results presented in this paper are based on numerical experiments performed with random numbers generated by software (specifically the *randn* function in Matlab, which uses the Ziggurat algorithm [6]). This approach removes a great number of uncertainties – we didn't have to deal with imperfections that are present in real noise sources, filters, and measurement equipment – and provides random variables whose properties are well documented and are true to the theory to within the numerical accuracy of floating point numbers.

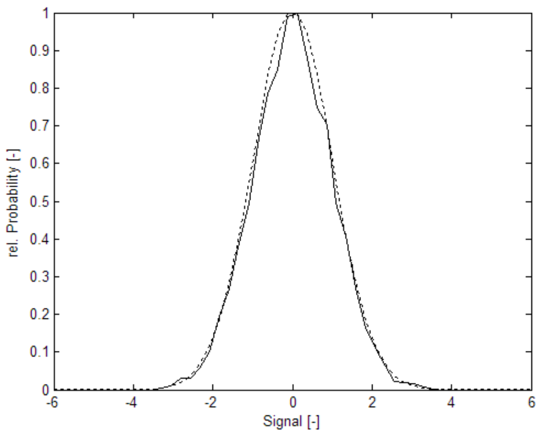
The approach also allowed us to analyze random data in a way that would not have been possible otherwise: for example, we are able to directly compare original and filtered data, look at the difference between small scale and large scale properties of samples, use a wide range of filters, and so on. Additionally, we were able to repeat experiments many times with minimal overhead and under the exact same boundary conditions

Effects of Sample Depth on Distribution

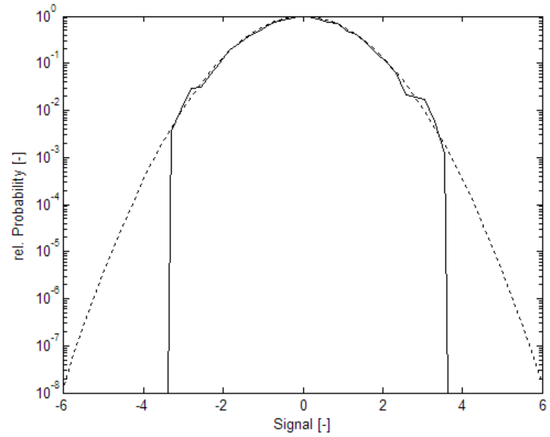
Our first numerical experiment shows the effect of the length of a sample on the observed distribution. A sample with a finite length must have a finite peak-to-peak value, and there is a systematic relationship between the sample length and observed peak-to-peak value and its crest factor.

In Figure 1, we show histograms for three different random samples consisting of 4096 (4K), 32K and 1M values respectively. Note how the linear plots (left side of the figure) are extremely close to the analytical probability density function with its characteristic bell shape. In logarithmic scale however, things look different: the histogram data closely follows the theory up to a certain point, where the probability becomes zero. Where exactly that happens depends on the length of the sample: the more values we look at, the larger is the observed peak-to-peak and therefore the crest factor.

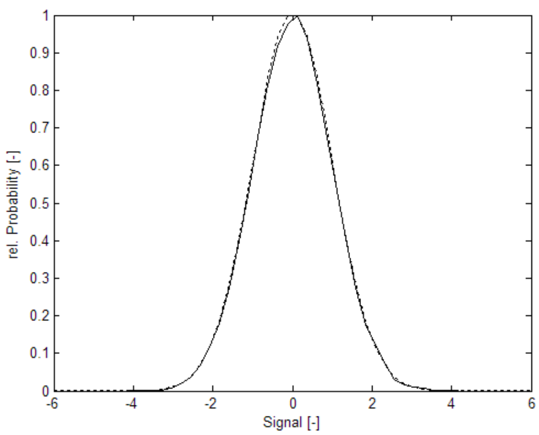
The results for a similar experiment are shown in Figure 2. This time, we created only a single sample, consisting of 10^6 random numbers. We then calculated the minimum and maximum, standard deviation, peak-to-peak, and crest factor for subsections of increasing size: starting with the first 10 values (#1-#10), then 20 values (#1-#20), 30 values, and so on, up to the length of the original sample. Plotted against the number of samples, the data shows that the standard deviation stabilizes after a surprisingly small number of samples (about 100) while the minimum and maximum and therefore also the peak-to-peak continues to grow. Once the standard deviation stabilizes, the crest factor is directly proportional to the peak-to-peak; in fact, the crest factor is equal to the peak-to-peak since we've used a standard normal distribution with a standard deviation of 1.0.



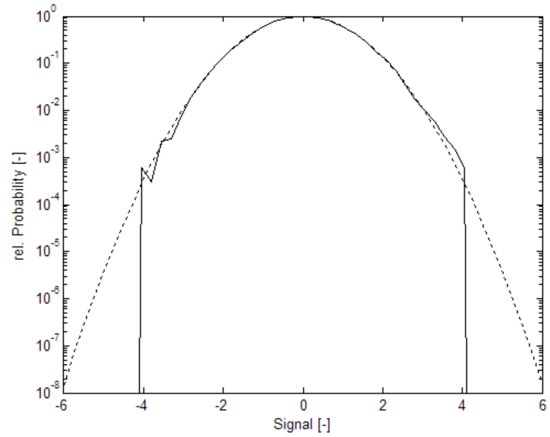
4K samples (linear) CF 6.8455



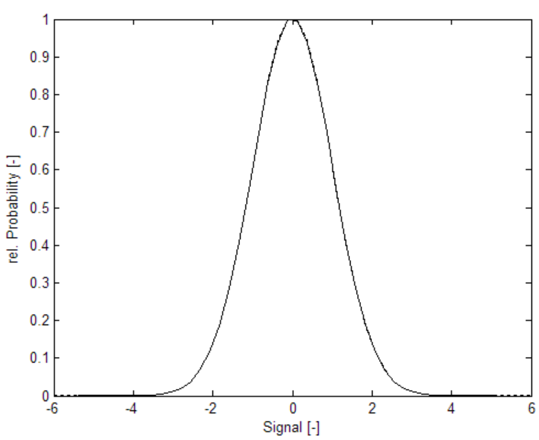
4K samples (logarithmic)



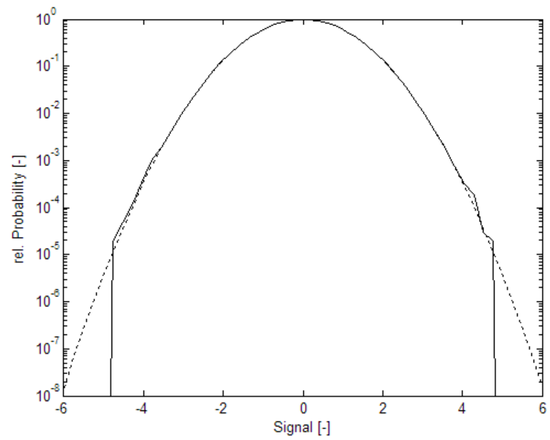
32K samples (linear) CF 8.1673



32K samples (logarithmic)

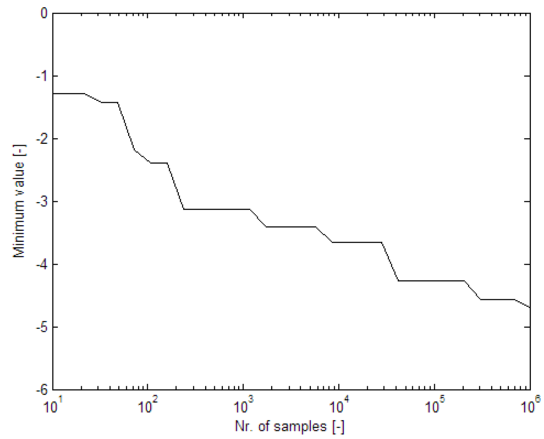


1M samples (linear) CF 10.2094

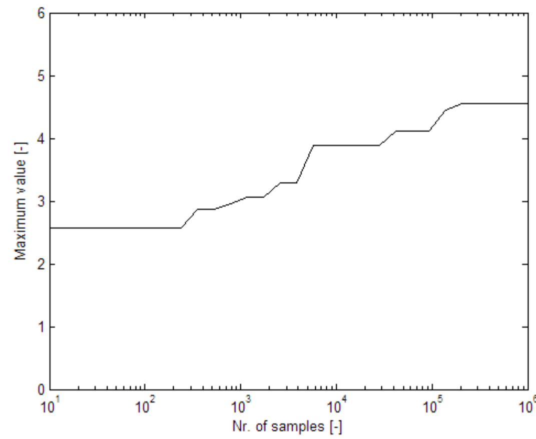


1M samples (logarithmic)

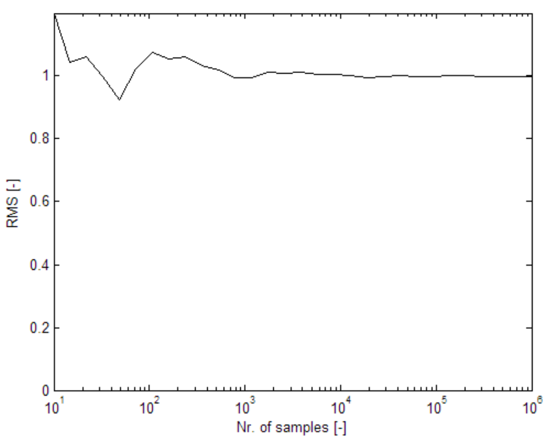
Figure 1 Histograms for random samples with different lengths versus probability density functions for an equivalent Gaussian distribution (dashed), in linear (left) and logarithmic (right) scale



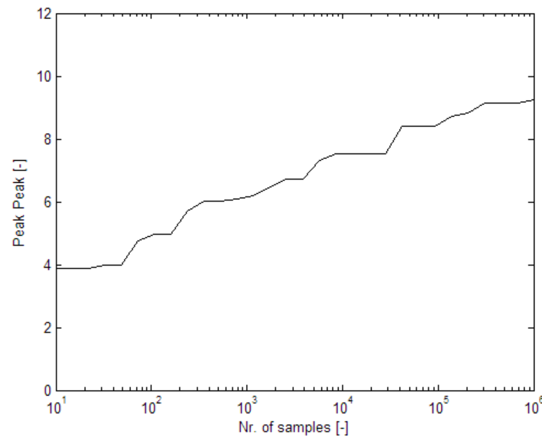
Minimum



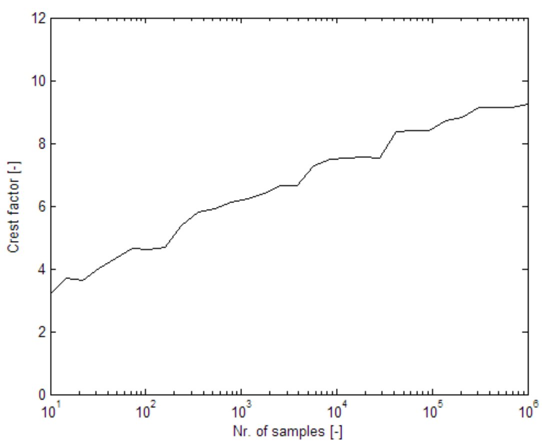
Maximum



Standard deviation (RMS)



Peak-to-peak



Crest Factor

Figure 2 Minimum, Maximum, Standard Deviation, Peak-to-peak and Crest Factor for subsections of a larger sample

Correlation between local frequency content and extremes

Gaussian white noise has a flat frequency spectrum. However that is true only for very large sample sizes, while the frequency content of smaller samples (or sections of a larger sample) is randomly distributed around the average power; if it were not true, the signal would not be random. Our next experiment was designed to show correlation between local high frequency content and the extreme values in a long sample of random noise.

The base data for this test was a vector of 1024x1024 (about 1×10^6) normally distributed random numbers. We chose a mean of 0.0 and a standard deviation of 1.0, the values for a standard normal distribution. The vector was then split into 1024 batches of 1024 samples each and each batch was analyzed independently.

We determined the frequency content of each batch with a Fast Fourier Transform (FFT). We assumed a sample rate of 2×10^9 samples per second, so the highest frequency that can be resolved is 1 GHz, a common spec for real noise sources. The local frequency content of the signal is shown in Figure 3, using a spectrogram graph: the color shows the power spectral density as a function of time on the horizontal and frequency on the vertical axis.

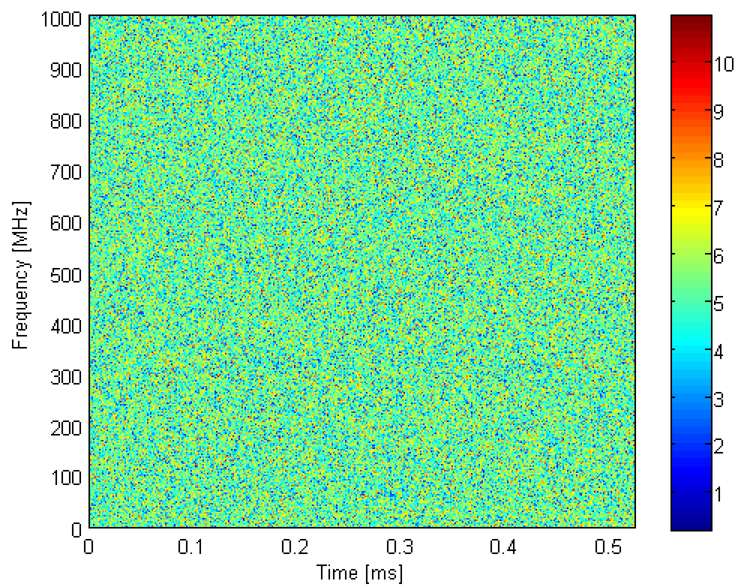


Figure 3 Sonogram style frequency spectrum of a random noise signal

The spectrogram does not show any obvious patterns in the data: the dominant color is a light green, equivalent to average power, and while there are excursions to both extremes they are randomly distributed. In grey scale, the plot looks like static noise on an old black and white TV which is exactly what we expect of random data.

Based on the power spectral density, we calculated a measure for the high frequency content of each batch: we arbitrarily defined a high frequency cutoff of 90% of the signal bandwidth, and then used the ratio of energy above 900 MHz to the total spectral energy to describe the high frequency content of each batch. A perfect noise signal has a flat frequency spectrum, and the expected average value for this measure is therefore 10%. Our results show a surprisingly large variability, with batches as low as 9.25% and as high as 11.75%, supporting our assumption that local high frequency content is not constant.

For every batch, we also calculated the average, minimum and maximum value, the standard deviation (rms) and peak-to-peak, the crest factor, and additionally the absolute largest value. Any correlation between the local high frequency content and local extreme values would appear in a scatter plot of these local descriptive values against the high frequency measure. However, none of the plots in Figure 4 show even a hint of correlation, so much that we didn't even bother to calculate correlation coefficients. [1]

The local properties of a long signal are apparently independent of the local high frequency content, which is somewhat counterintuitive. On the other hand, it is straightforward to imagine two signals with radically different frequency content, yet identical standard deviation and crest factor, for example, a square wave with variable frequency. In any case, this experiment didn't provide any evidence that a low pass filter would affect the extremes in a random signal more than the average values.

---

[1] In order to rule out an influence of inappropriate values for our experiment, we performed a crude manual sensitivity analysis: we varied several key parameters (batch size, number of batches, high frequency threshold) across a range of values, and didn't see significant influence on the correlation results. The number of batches and values per batch that we finally chose is a sweet spot in terms of frequency resolution (0.975 MHz) and ability to display (1024 batches can still be shown on a standard resolution monitor). We chose a sample length that is a power of two because that significantly speeds up the Matlab FFT.

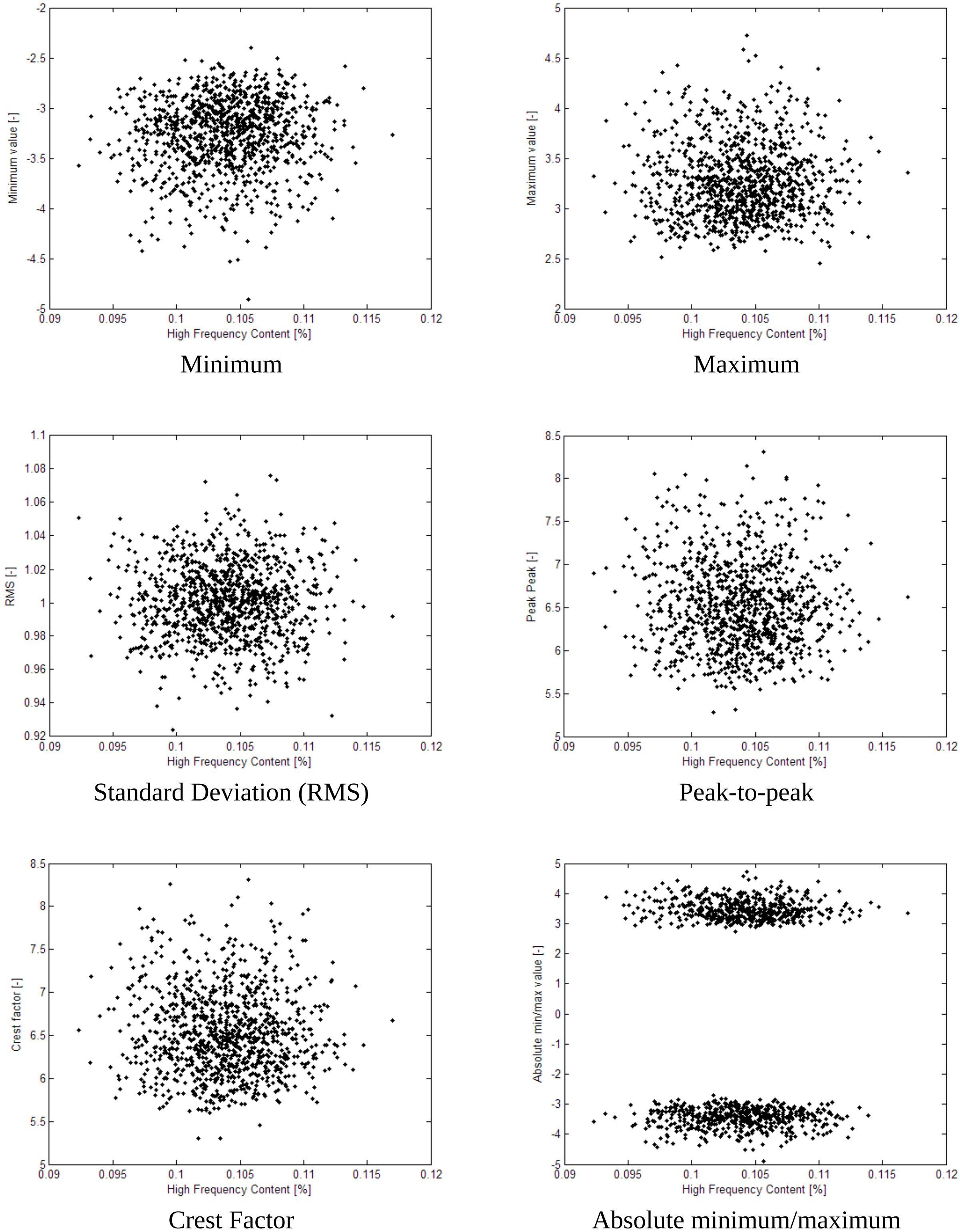


Figure 4 Correlation Analysis between high frequency content and local extreme values

Effect of low pass filter on extreme values

In our third experiment, we determined whether or not a low pass filter preferentially affects larger values in a random signal. If true, then the crest factor of low pass filtered data would be smaller than that of unfiltered data.

This experiment was based on a vector of 1024x1024 (about $1x10^6$) normally distributed random numbers with mean of 0.0 and standard deviation of 1.0, assuming a sample rate of 2 billion samples per second. The data was then filtered with a simple first order low pass filter. Figure 5 shows the time record of two runs using different 3 dB bandwidth settings for low pass filters of 900 MHz and 100 MHz, along with the ratio of filtered values to unfiltered values; this is essentially a time record of the value reduction due to the filtering.

With no filter in place, the ratio would be one. With an extremely aggressive low pass filter of very low bandwidth, the filtered signal would be nearly zero and the ratio would be correspondingly small. We therefore expect a ratio between 0 and 1, depending on the bandwidth setting. However, Figure 5 clearly shows that this is not the case. Note that the vertical axis for the ratio is in units of 10^4 and shows outliers as large as 5×10^4. There are also negative values, indicating that filtering actually changed the sign.

The reason for the extremely large ratios that we observed is that our original data contains values that are close to zero, and dividing any value by something very small tends towards infinity. We've therefore excluded value pairs where the absolute of the unfiltered value is below 0.1 from our further analysis [2].

Figure 6, in the top row, shows scatter plots of filtered vs. unfiltered data[3]. Both test cases show correlation, the 900 MHz more so than the 100 MHz filter. The slope of both point clouds represents the ratio of total energy in the unfiltered signal to filtered signal: on average, the filtered values are smaller than the unfiltered values, and the lower the bandwidth of the low pass filter, the smaller is the average ratio (note that we've used different vertical scales, shown in equal scale the difference is even more pronounced). This was already apparent in the time series plots in Figure 5, and is not surprising: the more energy a filter removes, the smaller the average output signal.

The final plots for this experiment, shown in the bottom row of Figure 6, show the ratio of filtered/unfiltered values in a scatter plot against the unfiltered values. Note how the filtered values for original values close to zero have a tendency to blow up, and also to change the sign: small negative values have equal likelihood of becoming relatively large negative or positive values! How can this happen?

---

[2] The cutoff is an arbitrary limit that turned out to work fine for our test cases

[3] The white bands in all plots in Figure 6 are caused by the removal of data with original values below the cutoff of 0.1

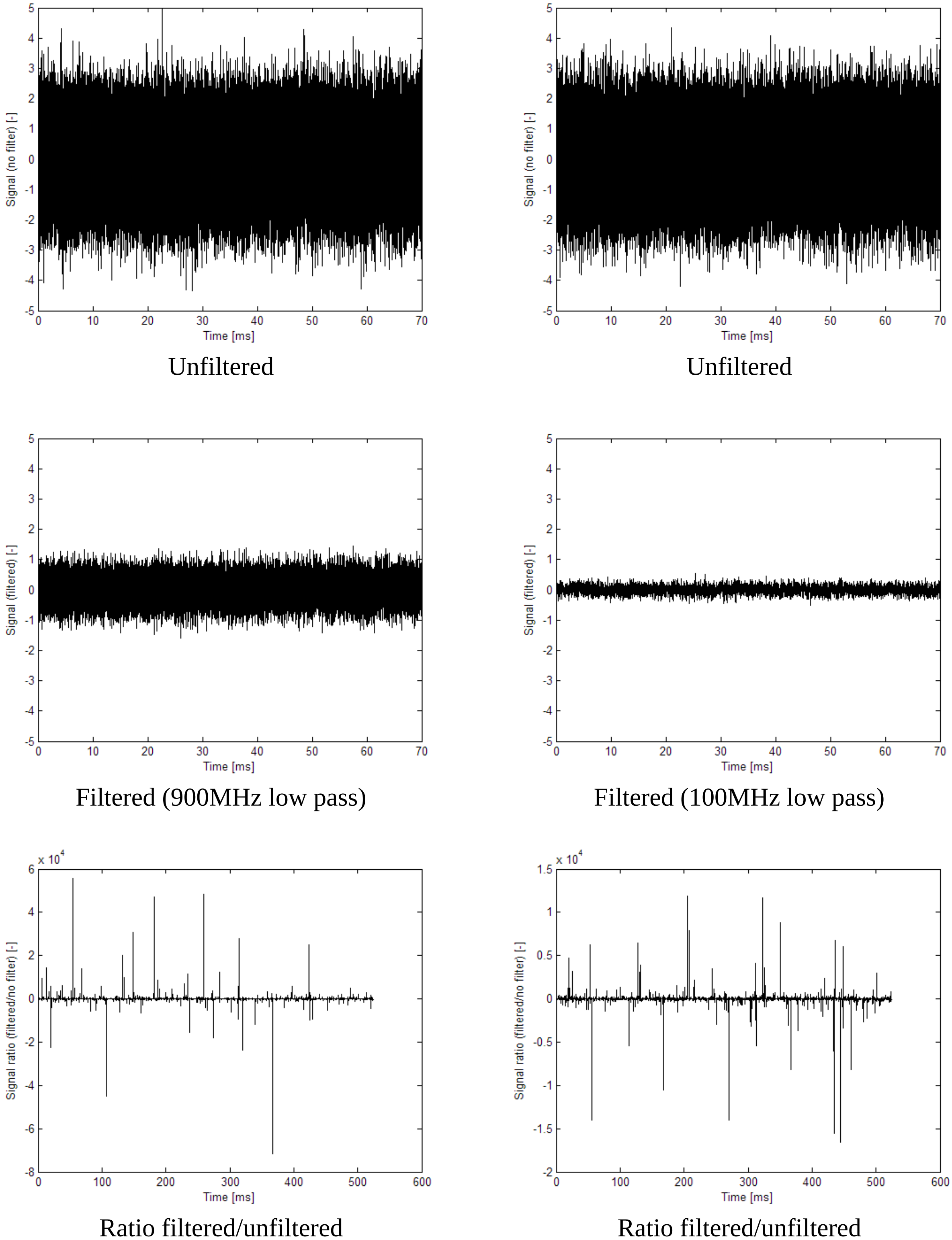


Figure 5 Effect of low pass filtering on random data, using 3dB bandwidth of 900MHz (left) and 100MHz (right). Time records of filtered (top) and unfiltered data (middle) and their ratio (bottom).

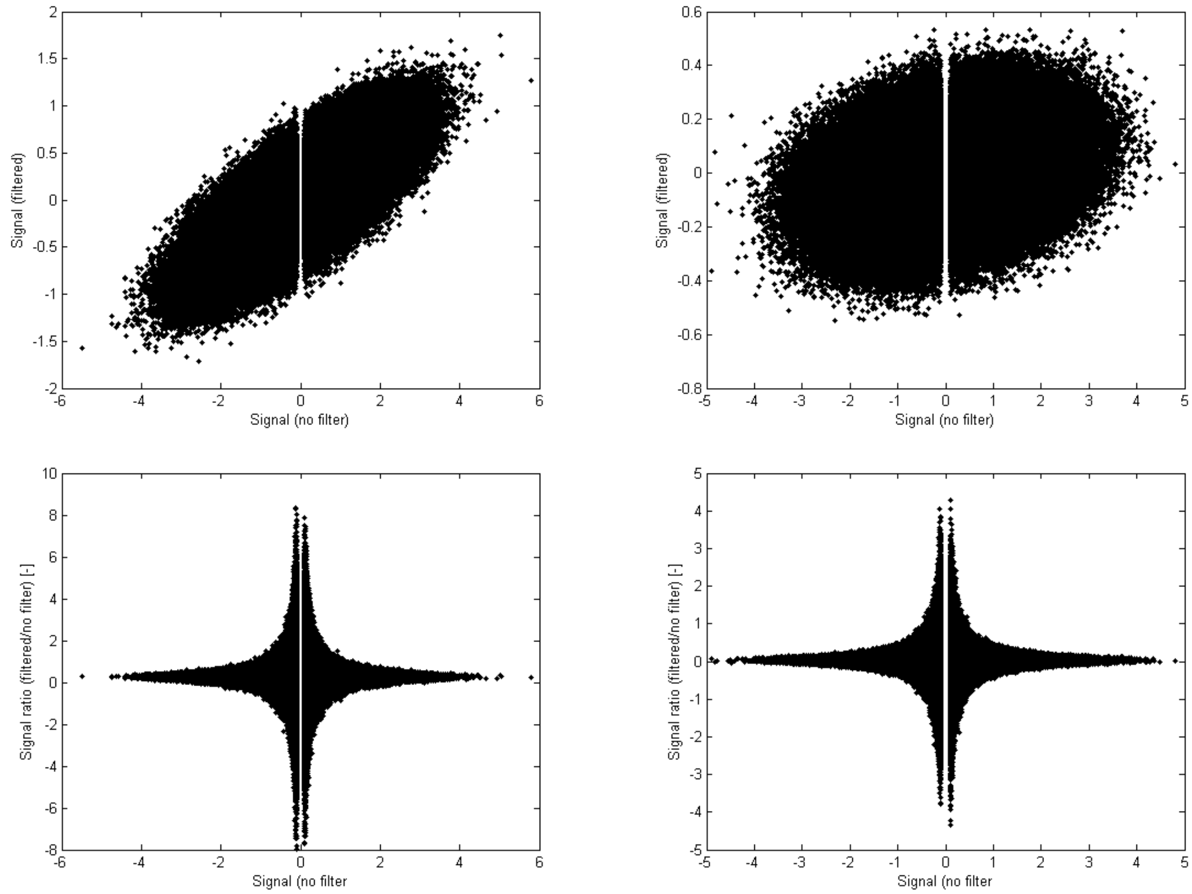


Figure 6 Effect of low pass filtering on random data, using 3dB bandwidth of 900MHz (left) and 100MHz (right). Scatter plots of filtered vs. unfiltered data (top) and their ratio vs. unfiltered data (bottom).

A low pass filter essentially puts an upper limit to the rate of change of a signal. It also introduces correlation between a value and the values surrounding it: the new value depends to a large degree not on the original value itself, but its neighbors. Imagine for example a moving average filter, which has a low pass characteristic; even a single outlier with a relatively large value can cause the average to change disproportionately if the other values are close to each other.

Because values with a relatively small absolute value have a higher probability of occurrence in a random signal, they are more likely to change more drastically than larger values; and that’s exactly what the bottom column in Figure 6 shows. More important however is that we don’t see any evidence that larger values are changed more than smaller values. Therefore, this experiment didn’t provide any reason to believe that a low pass filter would change the crest factor significantly.

Effect of low pass filter on distribution shape

The non linear effect of low pass filtering on a random signal that we observed in the previous experiment prompted another question: does the filtered noise signal still follow a Gaussian distribution? According to the theory, the shape of the distribution is independent of its frequency content. However, imagine a filter that removes most of the original signal's energy; the result will be a slowly varying DC signal, and it's hard to imagine that it would still be Gaussian.

Again, we generated vectors of 10^6 random numbers (following a standard normal distribution with $\mu = 0.0$ and $\sigma = 1.0$), and filtered them using different 3dB bandwidth settings for the low pass filters: 900 MHz and 100 MHz (same as in the previous experiment), and 10MHz, 1MHz, 100 kHz, and 10 kHz. We then calculated histogram data for the filtered values and plotted them in logarithmic scale (Figure 7). The dotted lines depict the probability density function of a Gaussian distribution with mean and standard deviation set to the respective values calculated from the filtered data.

The first two plots, for 900 MHz and 100 MHz, show distributions that are still in excellent agreement with the theoretical distributions. Both data sets show a slight asymmetry and don't follow the tails of the distribution exactly, but this is not necessarily an effect of the filtering. Refer back to Figure 1, where we plotted similar histograms for unfiltered data, and you will see distributions that are very similar.

At filter bandwidths of 10 MHz and 1 MHz, we start to see more significant deviations from the standard normal distribution, in particular in the tails of the distribution; also the asymmetry is more pronounced. However, a casual observer would still identify the distributions as Gaussian, especially if they were plotted in linear scale.

The last two plots show that there's a limit to how much low pass filtering we can apply before the result becomes non-Gaussian. For a filter bandwidth of 100 kHz, we can still somewhat recognize the characteristic bell shape, even though there are significant deviations from the normal distribution even in the center of the data. Finally, the data for a 10 kHz bandwidth is not Gaussian at all: the data shows four local maxima, is asymmetric, and doesn't exhibit the sort of tails that are characteristic for normally distributed data.

This experiment confirmed our intuition: in the limit, as the filter bandwidth becomes very small, the filtered data does no longer follow a Gaussian distribution. What is encouraging however is that the point where the Gaussian assumption breaks is fairly low: after all, we've filtered a noise signal with a 1 GHz bandwidth down to 1 MHz, by 3 orders of magnitude, and the resulting data was still in pretty good agreement with the theory.

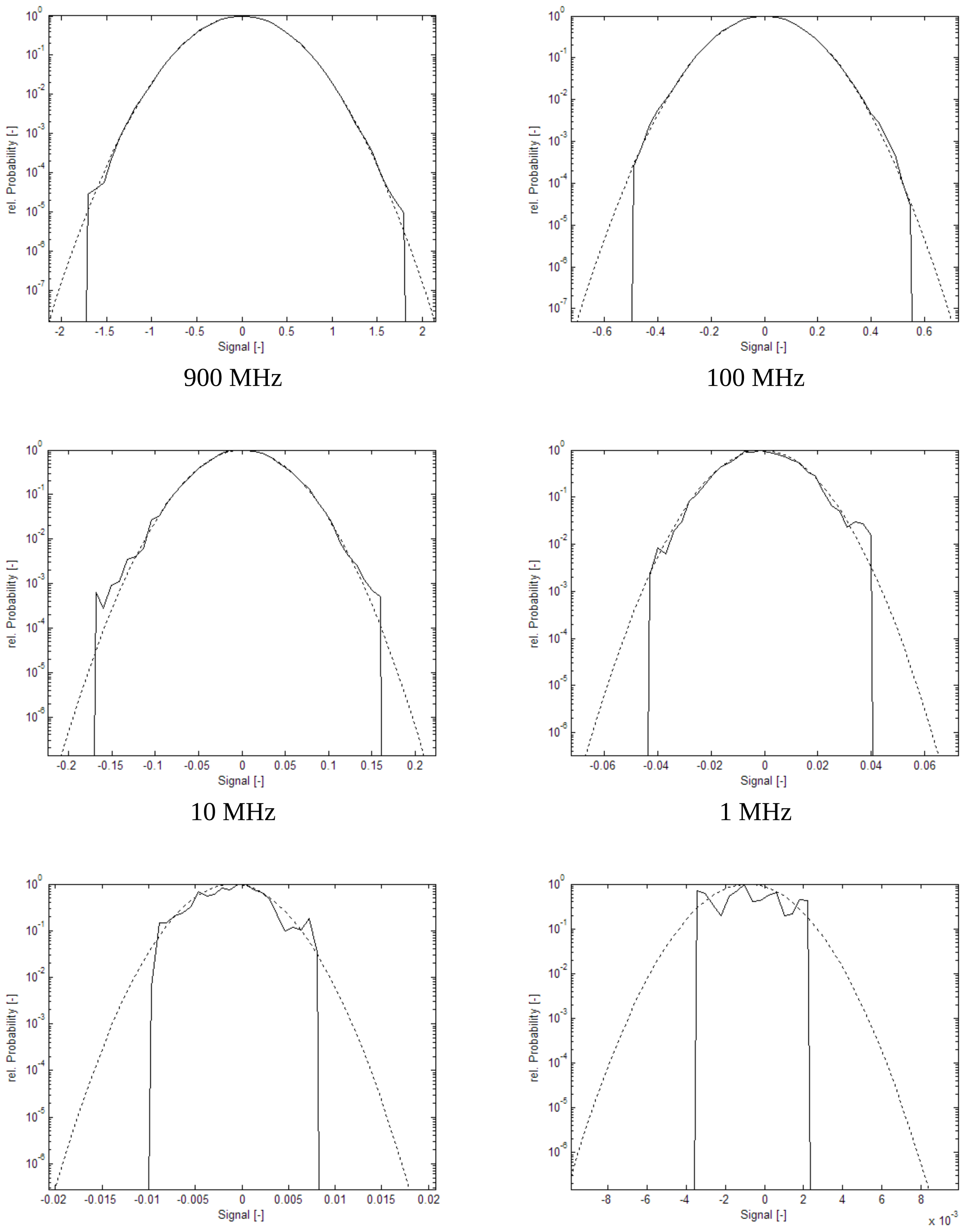


Figure 7 Effect of low pass filtering on the distribution of random data, using different low pass filter bandwidth settings

Effect of low pass filter bandwidth on crest factor

Our final numerical experiment is a variation of the previous one: again, we looked at the effect of a low pass filter on the distribution of a random signal. This time however, we used the parameters standard deviation, peak-to-peak, and crest factor to describe the resulting distributions numerically rather than plotting them against theory. This enabled us to run much more experiments: we used 30 different filter bandwidths, logarithmically spaced from 1 kHz to 1 GHz, and accumulated results from 50 runs at each corner frequency. The other parameters were unchanged, we used 10^6 samples and assumed a sample rate of 2×10^9 samples per second.

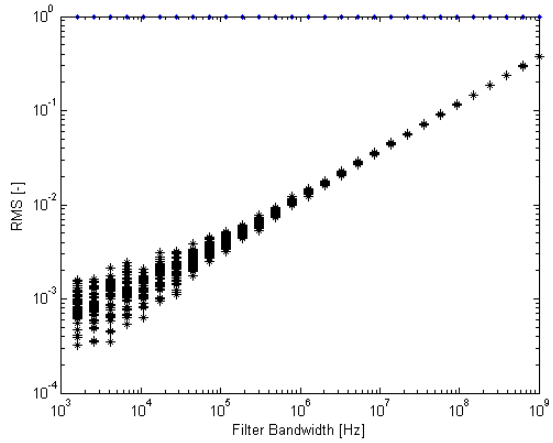
The results of a total of 1500 simulation runs are shown in Figure 8. The left column plots show the original (blue dots) and filtered values (black stars) for the standard deviation, peak-to-peak, and crest factor as a function of the filter bandwidth. The right column shows the ratio of filtered to original value for the same parameters. Notice that we've used a logarithmic scale for the filter bandwidth, and also for the parameters, with one notable exception: the ratio of filtered to original crest factor.

Low pass filtering affects the standard deviation and the peak-to-peak in a very similar fashion; they are decreasing almost linearly with decreasing filter bandwidth, down to about 100 kHz where we start to see much more variability from run to run (for the same corner frequency) and also see evidence that we may reach a kind of plateau. Recall from the previous experiment that 100 kHz is the frequency where the filtered distribution started to become markedly non-Gaussian! Therefore, standard deviation and peak-to-peak are no longer useful parameters to describe the distributions.

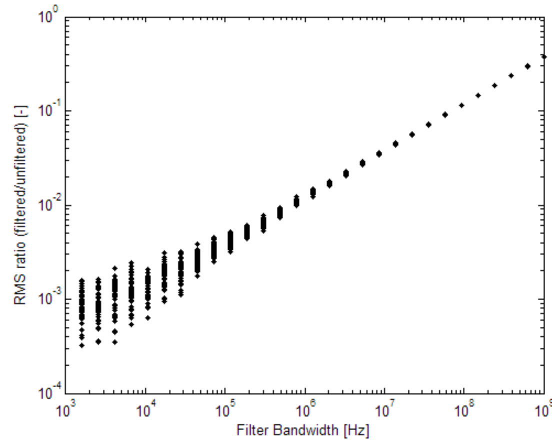
The value plot for the crest factor is different from the standard deviation and peak-to-peak plots in that both the original and the filtered data show much more variability from run to run. Also, the range of the filtered data is less than one order of magnitude, much less than the range for the other parameters, and is non linear in shape.

The ratio plots for standard deviation and peak-to-peak are very similar to the respective value plots. This is because the original values are independent of the filter bandwidth, with standard deviations very close to 1.0 and peak-to-peak values of about 10.0. At the lowest filter setting with a 1 kHz bandwidth, the total signal energy is reduced by about 99.9%: less than a thousandth of the original signal remains.

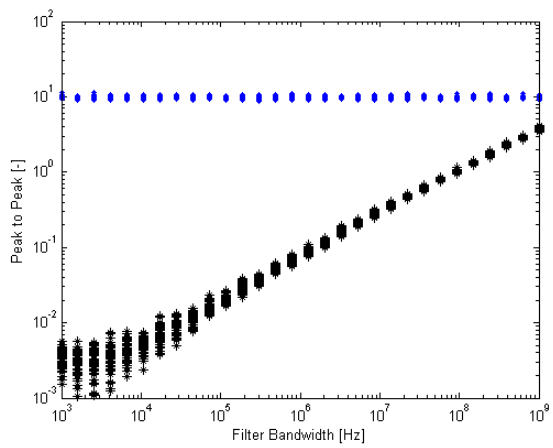
Finally, the ratio plot for the crest factor shows the effect of low pass filtering on the relative width of the distribution. According to theory, the crest factor should not be affected by a low pass filter, however the data doesn't agree: the crest factor for filtered signal is decreased significantly, down to as much as 25% of the original. What's even more interesting is that in some cases, the filtering even caused an increase in crest factor, by as much as 10%!



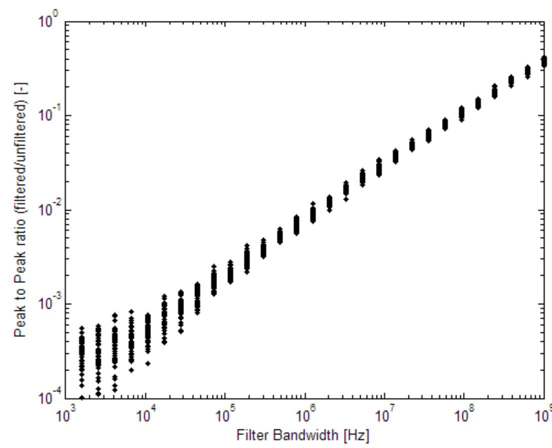
Standard deviation (RMS) values



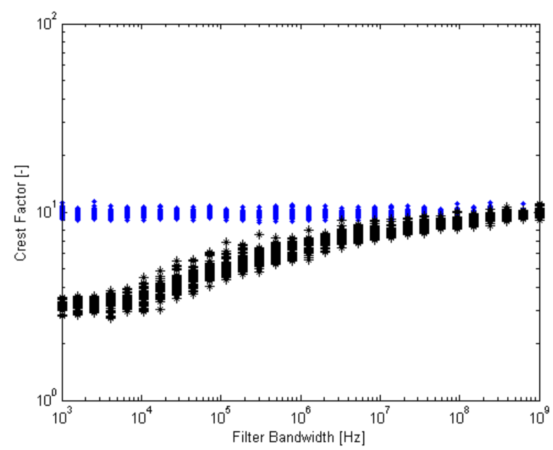
Standard deviation (RMS) ratio



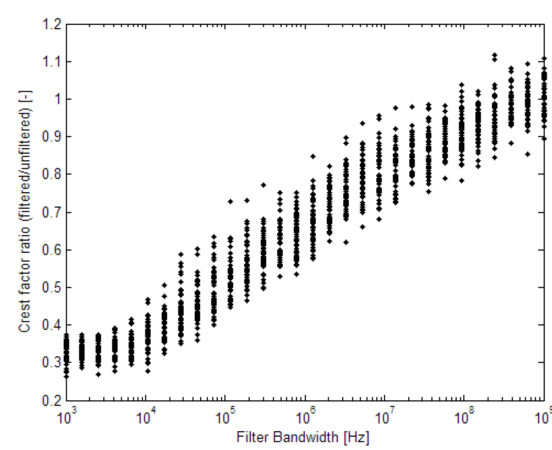
Peak-to-peak values



Peak-to-peak ratio



Crest factor values



Crest factor ratio

Figure 8 Effect of low pass filtering on the standard deviation, peak-to-peak, and crest factor of random data

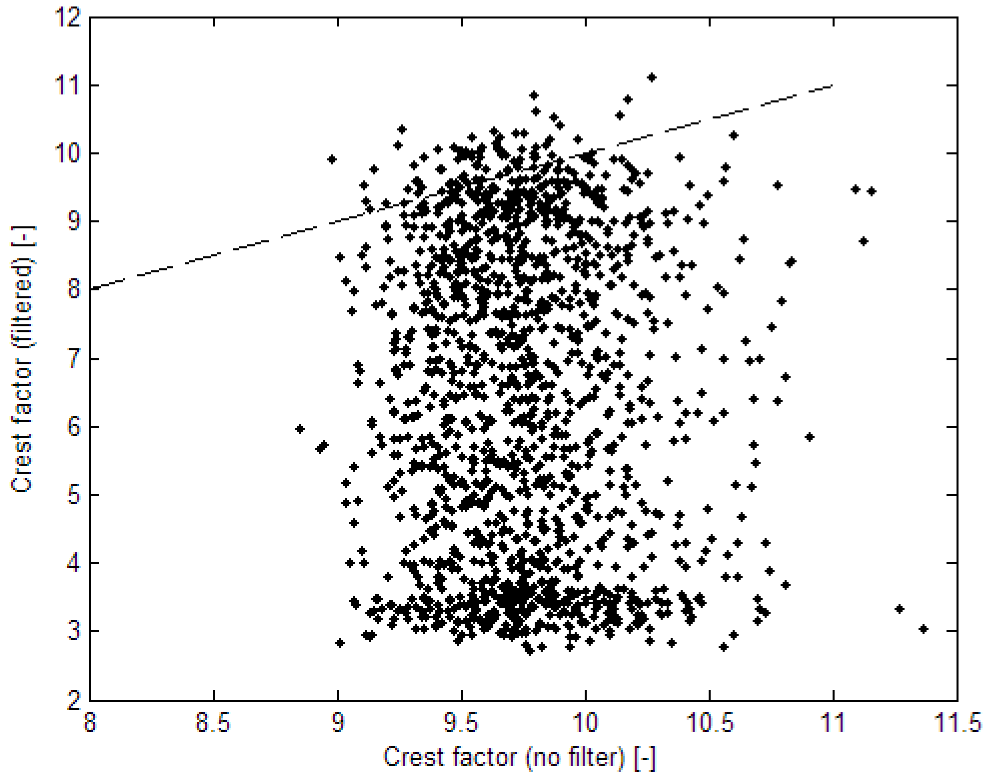


Figure 9 Effect of low pass filtering on the crest factor of random data

In Figure 9, we present the crest factor data again, this time in a scatter plot of filtered vs. original values; the dashed line depicts the 100% meridian where filtered and original values are equal. The plot shows the results from all 1500 simulation runs, independent of filter bandwidth.

Most remarkable about this data is not the effect of the filtering, but the variability in the unfiltered data. The average crest factor of the original data was 9.75, with a standard deviation of 0.0553 and a distinct skew towards the positive side. The minimum observed crest factor was 8.8479, the maximum 11.3658, a span of 2.5 units or about 25%. This alone is a concern if we think about the repeatability of a test.

This plot also shows more clearly the cases where the filtering caused the crest factor to increase. In about 60 out of 1500 simulation runs, according to Figure 8 all of them with filter bandwidth above 100 MHz, either hit a resonance in the filter or had a history leading to the peak values that caused the peak-to-peak to decrease less than expected based on the filter bandwidth.

Measurement Results

In order to verify the results from the numerical experiments, we have performed a series of measurements with a commercially available noise source, a NoiseCom PNG 7110 with 1 GHz of bandwidth. Histogram data was acquired with a digital sampling oscilloscope with a sampling rate of 8 GS/s and an analog bandwidth of 1.5 GHz.

We used three different filters in this experiment: two standard low pass filters with 3 dB bandwidth of 750 MHz and 190 MHz, and a special filter designed for PCI Express 2.0: its characteristic is a low pass with 1.5 MHz, followed by a 3 dB up to 100 MHz. Results are shown in the table, acquisition was stopped after 10^6 hits, 10^7 hits, etc to see the dependency of the values on the sample count.

		10^6 Hits	**$1x10^7$ Hits**	**$2x10^7$ Hits**	**$3x10^7$ Hits**	**$4x10^7$ Hits**	**$5x10^7$ Hits**	**10^8 Hits**
No Filter	Standard Deviation	95.26	95.69	95.90	95.37	95.27	95.19	95.08
	Peak-to-peak	957.00	1000.00	1029.00	1143.00	1143.00	1143.00	1157.00
	Crest Factor	10.05	10.45	10.73	11.98	12.00	12.01	12.17
750 MHz	Standard Deviation	60.33	60.29	60.21	60.23	60.24	60.25	60.21
Low pass	Peak-to-peak	643.00	714.00	714.00	729.00	729.00	757.00	757.00
	Crest Factor	10.66	11.84	11.86	12.10	12.10	12.56	12.57
190 MHz	Standard Deviation	23.01	22.96	22.95	22.95	22.94	22.94	22.95
Low pass	Peak-to-peak	371.00	371.00	371.00	371.00	386.00	386.00	386.00
(#1)	Crest Factor	16.12	16.16	16.17	16.17	16.83	16.83	16.82
190 MHz	Standard Deviation	26.63	26.63	26.64	26.64	26.63	26.62	26.62
Lowpass	Peak-to-peak	265.70	302.90	302.90	302.90	308.60	314.30	320.00
(#2)	Crest Factor	9.98	11.37	11.37	11.37	11.59	11.81	12.02
PCIe Filter	Standard Deviation	3.14	3.15	3.15	3.15	3.15	3.15	3.15
	Peak-to-peak	34.00	34.00	34.57	36.57	36.57	36.57	36.86
	Crest Factor	10.83	10.79	10.97	11.61	11.61	11.61	11.70

Note that we have repeated the experiment with the 190 MHz filter: the first run used the same vertical scaling as the unfiltered reference measurement and the 750 MHz low pass filter: 500 mV/div. Because the crest factors were so far off from the other experiments, we've repeated the measurement with a different scaling of 100 mV/div – notice how this influenced the results: measurements with real random data are incredibly difficult!

Overall, the data that we've captured was in good agreement with our numerical experiments: the crest factor values for 10^6 hits were in the same range as the numerical values for 10^6 random numbers, and the trend as the number of hits increases fits with our expectation that the crest factor increases continually. Even at 10^8 hits, we were still far from the specified crest factor limit of the noise source.

Summary and Conclusion

In receiver tolerance testing, we make use of two basic assumptions. First, that the peak-to-peak of a random signal at some probability level equals the rms value times a constant factor (for example 14 at 10^{-12}). And second, that the rms of two random signals adds up in a root of the sum of squares fashion. Both assumptions hold only if the noise distributions are truly Gaussian. The results from our numerical experiments show that low pass filtering does indeed affect the distribution of random signals, and as a result the total jitter or eye opening at the device, defined in terms of the bit error ratio, would not match the intended specification and the results of the test become difficult to interpret.

Additionally, our observations indicate that the crest factor may in fact not be a good measure for the distribution of a random signal: because the peak-to-peak of a random signal, and hence the crest factor, is dominated by only two values in a large sample, namely the minimum and maximum, it is bound to be randomly distributed. There have been concerns in the past about the repeatability of tests that use true random signals as a stimulus. However, to our knowledge the magnitude of the variability was not studied before. Additionally, there are no good alternatives: deterministic methods to create random signals that are sufficiently random yet repeatable have been described [2], but are not available yet with the necessary bandwidth.

References

[1] Ransom Stephens and Bob Muro, *Characterization of Gaussian Noise Sources*, DesignCon 2008

[2] Martin Mücke, Joachim Moll, Marcus Müller, *Precision Digital Noise Source*, DesignCon 2007

[3] Marcus Müller, Ransom Stephens, Russ McHugh, *Total Jitter Measurement at Low Probability Levels, using Optimized BERT Scan Method*, DesignCon 2005

[4] Dennis Derickson and Marcus Müller (editors), *Digital Communications Test and Measurement*, Prentice Hall, 2007

[5] W. Marshall Leach, Jr., *Dr. Leach's Noise Potpourri*, Georgia Institute of Technology School of Electrical and Computer Engineering, 1999

[6] George Marsaglia, Wai Wan Tsang,*The Ziggurat Method for Generating Random Variables*, Journal of Statistical Software, Vol. 5, Issue 8, Oct 2000