

A New Method for Jitter Decomposition Through Its Distribution Tail Fitting

Mike P. Li*, Jan Wilstrup+, Ross Jessen+, Dennis Petrich*

* Wavecrest Corporation
1735 Technology Dr., Suite 400
San Jose, CA 95110
mpeng@wavecrestcorp.com

+ Wavecrest Corporation
7275 Bush lake Road
Edina, MN 55439

Abstract

We present a new time-domain jitter separation method. Such a method automatically searches and fits the tail parts of the jitter histogram with nonlinear jitter models and estimates deterministic and random jitter components. Bit error rate (BER) calculation based on the deterministic and random jitter components is also discussed and demonstrated.

1. Introduction

In high-speed signal generating, transmission, and receiving, jitter is generally defined as any timing displacement (or error) referenced to the timing of an ideal signal. In other words, jitter is a signal associated with conventional waveform signal. Jitter can have many components associated with it. Historically, there are many ways to classify various jitters, and the most commonly used are either statistical based or signal property based (e.g., pattern dependent). In the statistical based classification, jitters are splitted into two big categories: one is the deterministic jitter (DJ), and the other is the random jitter (RJ). DJ is bounded and RJ is unbounded based on their definitions. In the signal property based classification, jitters are splitted into noise jitter (random motion of electrons) (NJ), and pattern dependent jitter (PDJ). These classification schemes are the first step and have a broad sense. For a general review of jitter and related issues, we refer the readers to ^{[1][2][3]} and references therein. In this paper, we will adopt a hybrid approach to jitter classification since it covers most of the jitter sources.

Jitter can be further separated within each broad category. Within deterministic jitter, jitter can be further classified into periodic, duty cycle distortion (DCD), intersymbol interference (ISI), periodic, and bounded uncorrected (BU) components. Within the random jitter, jitter can be a single Gaussian (SG) or multiple Gaussian (MG). Each jitter component has some specific corresponded root causes and characteristics. The general jitter sources are lossy

media, reflections, cross-talk, electromagnetic interference (EMI), systematical modulations, and pattern dependency, for DJ; and thermal noise, shot noise, flick noise, random modulation, non-stationary interference, for RJ. In the real world applications, what is measured, in general, is the total jitter, namely both DJ and RJ are mixed together. To understand the root cause of the jitter, separating and identifying each jitter component is essential. As the speed for microprocessor, memory, data bus, and transmission media increases steadily, if not exponentially, failures caused by jitter will become more and more severe. Therefore, how to model, simulate, measure, and characterize jitter will become even more important and challenging for high-speed signals in the forth coming years.

Recently, we have developed a method for DJ and RJ separation based on Blackman-Tukey algorithm^[4]. In such a method, the DCD+ISI is obtained by calculating the mean of the time error (measured edges versus that expected from a pattern). periodic jitter (PJ) and RJ are calculated through FFT spectrum estimation of the variance via auto-correlation function of the time jitter. Since spectrum estimation is required in this method, a time series of jitter measurements are needed. In the case when no jitter time record (i.e., jitter is measured as a function of time) is available, such a method may not apply.

In the time domain measurements, jitter can be measured for a specific edge transition or over a time span of many edge transitions. In each case, many jitter samples are collected for each edge transition so that statistical information can be gathered and analyzed. A typical example in real world practice is the jitter histogram measured for a specific edge transition, or jitter histograms measured over a time span of many edge transitions. Such a jitter histogram reflects the mixture of DJ and RJ processes associated with the edge transitions. For many years, this information is available, but, to our knowledge, there is *no* theory or method established to decompose the total jitter histogram into DJ and RJ components. What was and has been used to quantify jitter are statistical

peak-to-peak value and 1σ standard deviation, based on the entire histogram distribution that has both DJ and RJ components. The correct way to quantify jitter is to use peak-to-peak value for DJ since it is bounded, and 1σ standard deviation for RJ since it is unbounded and random. It can be seen that the use of peak-to-peak value and 1σ standard deviation based on a total histogram that has both DJ and RJ components is not only misleading, but also statistically wrong.

In this paper, we present a method/algorithm to decompose a total jitter histogram into DJ and RJ components through tail fitting. In section 2, we will discuss the theory of the total jitter histogram and its relationship with DJ and RJ processes. In section 3, we will describe an algorithm that automatically identifies the tail parts of the distribution and fits them with the Gaussian distributions, to estimate the DJ and RJ components. In section 4, we present our Monte Carlo simulation results that verify the validity of our algorithm. In section 5, we show some practical application examples using our jitter software implemented with the algorithm and to verify the correlation, as well as to demonstrate the advantage of having DJ and RJ components. In section 6, we will summarize our results.

Note: Theoretical analysis and computer simulations have been performed. The tail search and fitting algorithms have been implemented in our current jitter analysis software product. A patent application has been filed with the U. S. patent office.

2. Jitter Histogram Distribution and Its Relationship with DJ and RJ

Theoretically, the tail part of the histogram distribution reflects the random jitter process. Physically, random jitter is due to the random motion of particles within a device or transmission media. The random velocity of these particles in an equilibrium state is proven to be best described by a Gaussian distribution. Therefore random jitter is naturally modeled by a Gaussian function. Since multi-temperature particle distribution is possible, a multi-Gaussian distribution function may be needed to model certain random jitter processes. A single Gaussian jitter histogram distribution is defined as :

$$h(t) = N_{\max} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \quad (1)$$

where N_{\max} is maximum event count, t is the jitter, μ and σ are the Gaussian mean and standard deviation respectively.

The measured total jitter histogram represents the scaled-up total jitter probability distribution function (PDF). On the other hand, the *convolution* of RJ PDF with DJ PDF gives the total PDF, if DJ and RJ processes are independent. In most cases, such an assumption is valid. Therefore the tail part of the distribution should be mostly determined by the random jitter, which, in general, has a Gaussian type of distribution. The random noise can be quantified by the 1σ standard deviation of Gaussian distribution, while DJ can be quantified by the peak-to-peak value. Depending on the error probability level, the total RJ can be a multiple of the σ , deduced from the Gaussian distribution.

In the absence of DJ, the histogram of the jitter should be roughly a Gaussian distribution. Under this condition, there is only one peak in the distribution which corresponds to zero DJ. The rms RJ is the σ value. When both DJ and RJ come to play together, the measured jitter distribution will be broadened and no longer a Gaussian as a whole. On the other hand, both ends of the distribution should still keep the Gaussian type of tails since DJ PDF is bounded. These tail part distributions can be used to deduce the RJ number. Because of the DJ, the mean of each tail is no longer the same and multi-peaks can be present in the histogram. The jitter difference between far left peak value and far right peak value will give rise to the DJ. Figure 1 is a schematic drawing of such a broadened total histogram in the presence of both DJ and RJ.

If there is no bias and statistical sampling noise in the measurement, the two tails, which represent the random process, should be symmetrical. Since it is not possible to completely randomize measurements and reduce the sampling noise to zero, the σ values for the far left and far right Gaussian tails may not be the same. The total RJ value should be the average of these two, and DJ is the distance between two peaks of far left and far right Gaussian tails, namely

$$RJ = (\sigma_l + \sigma_r) / 2 \quad (2)$$

and

$$DJ = \mu_r - \mu_l \quad (3)$$

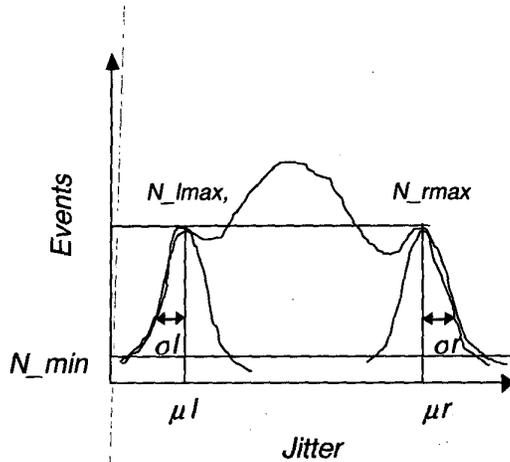


Figure 1 Schematic drawing of total jitter histogram in the presence of DJ and RJ.

3. Tail Search and Fitting

3.1) The requirements and specifications

Identifying the tail part of the histogram distribution, and then to fit them with the Gaussian function are the key to DJ and RJ separation with a given total jitter histogram. It is not possible to tell where the tail part of the histogram is without studying each individual data and its relationship with the neighboring data. The easiest way to identify a tail part is through the graphical displaying of the histogram and picking up the tail part via visual inspection. The disadvantage of such an approach is that it lacks repeatability, and it cannot be adopted for production test. Therefore the requirements for a search algorithm should be: i.) it is capable of finding the true tail part quickly, accurately, and repeatedly; ii.) it has to be automatic (i.e., no user intervention or visual inspection are required).

The fitting procedure should be able to deal with the statistical fluctuation and factor this into the fitting routines. The tail part has the lowest event counts and statistical uncertainty can be high. Simple straight forward least-square fit algorithm will not work since the statistical error will propagate into the fitting parameters. This in turn gives rise large to errors in DJ and RJ estimation. A more advanced non-linear fitting algorithm is needed to meet these requirements.

3.2) Algorithm for tail identification

One of the key characteristics of Gaussian tail is its monotonicity. That means: for the left side of the tail, it monotonically increases; for the right part of the tail, it monotonically decreases. Due to the presence of DJ, monotonicity will break and local maximums near the left and right part of tails. Without DJ, there is only one maximum that corresponds to the mean of the distribution.

A difficult issue that a tail search algorithm faces is the statistical fluctuation. In the presence of statistical fluctuations, the monotonicity of a real Gaussian distribution is no longer true, and using the raw fluctuated data to find the local maximum points for both left and right tails will be extremely difficult, if not impossible. The solution ought to be first filter out the noise and then use the smoothed histogram to locate the maximum points. There are two ways to achieve that, in general. One is through direct time domain averaging; another is through FFT to get the spectrum, then apply a low-pass filter, and do the IFFT. In time domain averaging one needs to determine how many data points to use since this determines the smooth level of the curve. In the FFT/IFFT approach, one has to determine the bandwidth of the filter. The number of averaging points and filter bandwidth may need to be adjusted, depending on the fluctuation noise frequency and amplitude. In other words, a rule-based artificial intelligent algorithm must be used to enable the smoothing algorithm to deal with a wide range of fluctuation amplitudes and frequencies. This is an important requirement to guarantee that smoothing only washes away the unwanted fluctuation noise, not the true feature of the jitter histogram.

Once the smoothed histogram $hs(t)$ is obtained either through the time domain averaging or time-frequency domain FFT-filtering-IFFT, the maximum locations can be found by calculating the first and second order derivatives of the jitter histogram. The only maximum points of interest are the first maximum from the far left and the first maximum from the far right.

3.3) Algorithm for the tail fitting

One should use a fitting algorithm which weights the data record based on the quality of each data. The bigger the error, the less role it should play in minimizing the difference between model expected value and the measured value. Thus, we need to use χ^2 as a gauge to determine how good the fit is. The fitting function is Gaussian and the fitting algorithm is nonlinear so it can handle both linear and non-linear fitting functions. For details of χ^2 theory, we refer the readers to [5] and references therein.

χ^2 fitting is an iterating process, in contrast to linear equation solving in the case of linear least-squared fitting. The final answer is obtained when the iteration converges. For this reason, initial values of the fitting parameters are needed. A primitive way to do this is to try different initial values and to see whether they converge to the same final values. If the initial guessed values are far from the final actual values, it may either take longer time to converge,

or get stuck at a local χ^2 minimum and never converge to the final global χ^2 minimum point. Calculations should be carried out to estimate the initial fitting parameters by using the tail parts of histogram so that the initial fitting parameters are close to the final converging values. This will also make the iteration to converge rapidly and to avoid stuck-in local minimum (pivot). We would like to emphasize that the χ^2 method has proven to be robust.

4. Monte Carlo Simulations

4.1.) Histogram with statistical noise

To test how well the search and the fitting algorithms worked, we need simulations. We started with a known bimodal histogram, which is represented by two added Gaussian distributions superimposed with a random noise. This makes the overall histogram be close to that of actual measurements. The overall histogram is represented by the following equation of

$$h(x) = N_l e^{-\frac{(t-\mu_l)^2}{2\sigma_l^2}} + N_r e^{-\frac{(t-\mu_r)^2}{2\sigma_r^2}} + N_n \text{ran}(t) \quad (4)$$

where N_l , N_r are the peak values, μ_l , μ_r are means, σ_l , σ_r are standard deviations, respectively, for two Gaussian distributions. $\text{ran}(t)$ is a random number generating function based on Monte Carlo method. It has a mean of zero and a standard deviation of unity. N_n is the amplitude for the random number envelopes. For a Monte Carlo based random number generation, we refer the readers to [6].

A good search and fitting algorithm should return the fitted parameters that are consistent with these pre-defined in the simulation. A critical test would be: can an accurate fitting parameter be obtained in the presence of significant statistical fluctuations, i.e., N_n is a significant portion of N_l or N_r . Otherwise, no accurate parameters can be obtained since all the real world measurements are subject to statistical fluctuation.

4.2) Fitting results

There are two scenarios we need to treat differently. The first is that when two Gaussian distributions are well separated, i.e., when $\mu_r - \mu_l > \sigma_l + \sigma_r$. Under such condition, the two distributions are not well mixed, and the tail parts up to the point of the first maximum are essentially uncontaminated. Therefore, we could use both left and right tail data from its lower value to the first

maximums. This will enhance the tail data usage and the Gaussian model will be better constrained. This can correspond to the case when $DJ > 2 RJ$ in the jitter analysis applications. Such a histogram is very common in spread-spectrum clock devices.

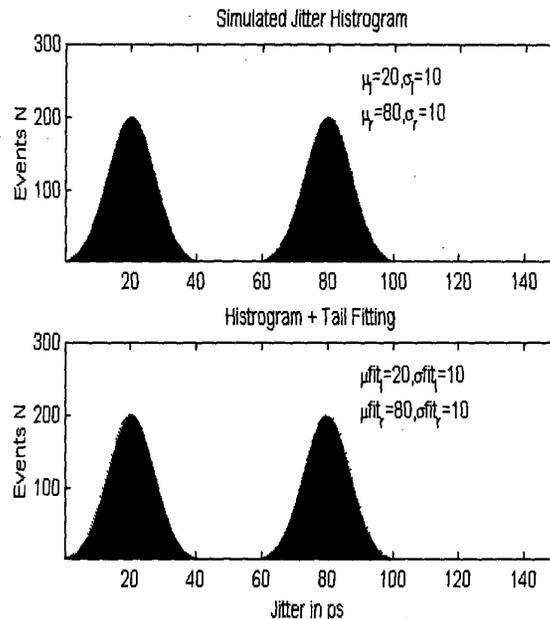


Figure 2.1 $N_n = 0$, no statistical fluctuation.

The second is when two Gaussian distributions are not well separated, namely $\mu_r - \mu_l < \sigma_l + \sigma_r$. Under such condition, the contamination of two distributions could extend to the tail parts. As a result, one should only use the lower parts of the tails for the fitting to minimizing contamination. A conservative way is to use the tail part from the lowest event count to half of the N_l or N_r . This can correspond to the case when $DJ < 2 RJ$ in the jitter analysis applications.

Figures 2.1-2.2 show the results corresponding to two well-separated Gaussian histograms (i.e., $\mu_r - \mu_l > \sigma_l + \sigma_r$). Figure 3.1-3.2 show the results corresponding to two

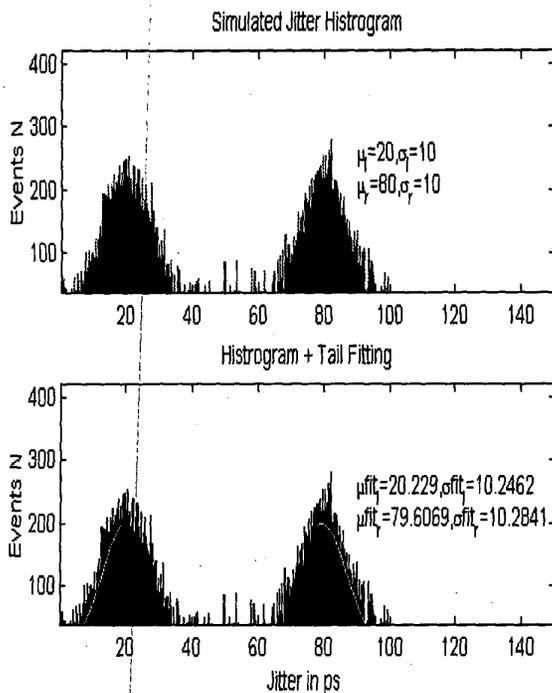


Figure 2.2 $N_n = 30$, with significant statistical fluctuation

mixed Gaussian histograms (i.e., $\mu_r - \mu_l < \sigma_l + \sigma_r$). In each case, non-fluctuated ($N_n = 0$) and fluctuated ($N_n \neq 0$) scenarios are considered. In both cases, the fitted parameters are consistent with the simulated parameters to within 4% even when the statistical fluctuation reaches 15% of the total histogram peak.

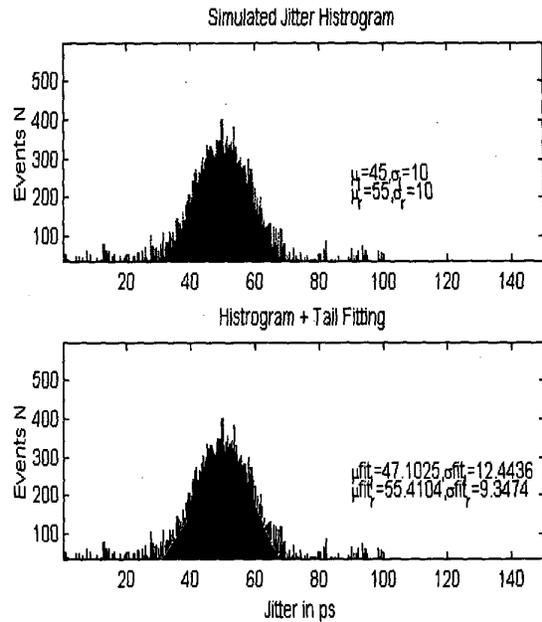


Figure 3.2. $N_n=30$, with significant statistical fluctuation.

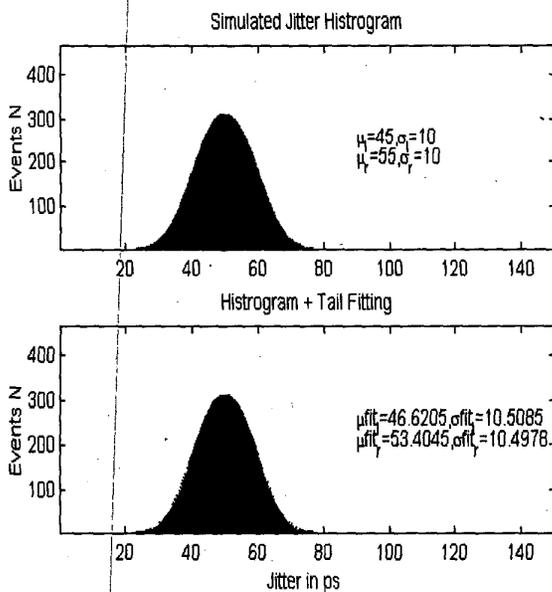


Figure 3.1 $N_n = 0$, no statistical fluctuation.

5. A Practical Case Study

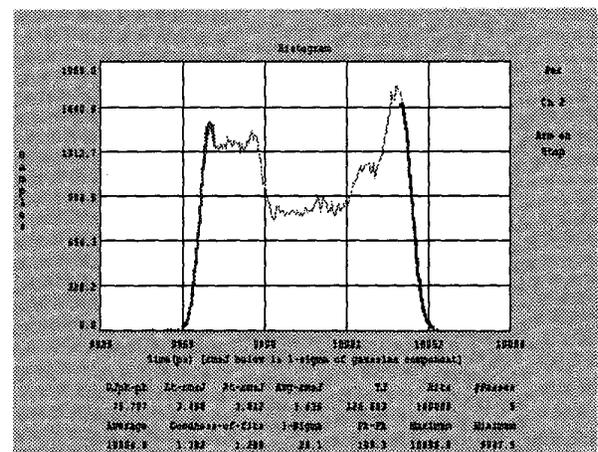


Figure 4. Histogram tail search and fitting algorithm application for a clock jitter, using the Wavecrest DTS system as the measurement instrument.

The search and fitting algorithms discussed in sections 3 and 4 are implemented in the Wavecrest Virtual Instrument software. Using the Wavecrest DTS 2075 system, and a clock input to the instrument, we measured the jitter histogram and use the software to decompose DJ and RJ. Figure 4 shows an example for clock jitter histogram and DJ and RJ values deduced from the tail fitting algorithm. In this case, the histogram has twin-peak DJ process that is caused by a periodic modulation. This is a 100 MHz clock signal with a 5 MHz periodic modulation. Thick lines indicate the Gaussian model fit to the tail part of the distribution overlaying to the measured histogram thin lines. DJ value gives the modulation jitter amplitude.

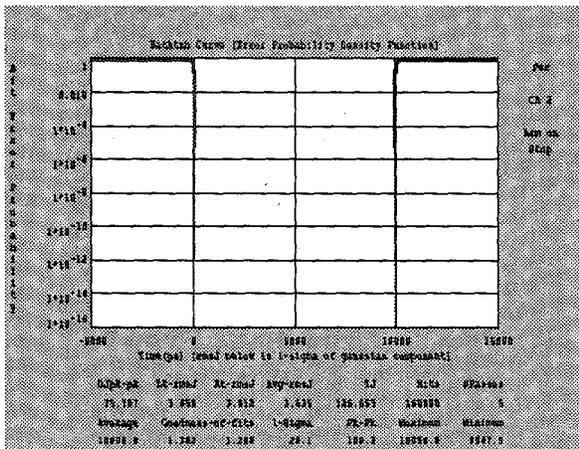


Figure 5 BER curves for the same clock signal.

With the DJ and RJ values, clock performance can be predicted via bit error rate error (BER) curve that is calculated through the measured total histogram, as well RJ number. Details on the how to calculate BER curve can be found in [3] and reference therein. Figure 5 shows the BER curve (sometime it is also called bathtub curve). Thick lines indicate the actual measured BER, and thin lines indicate the extrapolated BER based on a RJ Gaussian PDF. BER curve is an important overall performance indicator for time critical ICs and systems. In serial data communication, total jitter is normally specified at an error probability level of 10^{-12} . In Figure 5 example, the operational margin is 9873.4 ps, and the total jitter is 126.6 ps, at 10^{-12} probability level.

Correlation study is carried out by comparing the RJ and DJ values with those obtained by using the Blackman-Tukey method. The agreement is good. In most cases, the difference was less than 5%. However, as we mentioned before, tail fitting method does not require the time span of the jitter record. Consequently, tail-fitting algorithms have a wide application in the field of jitter analysis.

6. Summary and Conclusions

We have developed a general-purpose automated search and nonlinear fitting algorithm, with special emphasis on deterministic jitter and random jitter separation. We have shown that under significant statistical fluctuation, our algorithm can still separate DJ and RJ accurately and repeatedly. This algorithm does not require any user intervention and applies in both laboratory and production applications. Our algorithm can apply to either a single histogram, or a series jitter histogram (for deterministic and random jitter spectrum analysis).

These algorithms can also be useful in other general-purpose signal analysis applications. For example, one can use such method to analyze the phase noise spectrum, and determine what kind of noise processes are involved in a specific device, such as clock PLL or clock recovery PLL. Other examples include: DJ and RJ separation for eye-histogram, and bounded uncorrelated jitter (BUJ) separation that can be caused by crosstalk.

We have also simulated the effect of sampling statistics error using the Monte Carlo method, for a single Gaussian distribution. This is a very important issue since one can only measure the sampling statistics and use that to represent the underlining population statistics. If the sampling statistics are far from the population statistics, one cannot get the true value for the population statistics accurately, regardless of what kind of analysis tool one is using. This is true for any kind of measurement. The goodness of the sampling statistics is proportional to the total number of measurements used to compose the histogram. The bigger the total number of measurements, the better the sampling statistics. For a given total number of measurement per histogram, if one repeats the histogram measurement and DJ and RJ value deduction process, for a number of times, the DJ and RJ parameters deduced will compose distributions that are very close to Gaussian. The standard deviation for the DJ and RJ distributions obtained in this manner is proportional inversely to the total number of measurements in each histogram composition. Our simulation has shown a 1σ error of 4.5% for DJ and 17.2% for RJ, given 10,000 measurements per histogram and repeating the simulation 100 times. Of course, for different histograms, or for different measurement totals, these numbers can change. The point we are making here is that sampling statistics is very important to guarantee the accuracy of DJ and RJ. A minimum number of measurements is needed for good sampling statistics. That number can vary with different applications and accuracy specifications.

References

- [1] Y. Takasaki, "Digital transmission design and jitter analysis", Artech House, Inc., 1991
- [2] P. R. Trischitta, E. L. Varma, "Jitter in digital transmission systems", Artech House, Inc. 1992.
- [3] National committee for information technology standardization (NCITS), working draft for "Fiber channel - methodologies for jitter specification", Rev 6, 1998.
- [4] J. Wilstrup, "A method of serial data jitter analysis using one-shot time interval measurements", ITC Proceeding, 1998.
- [5] P. R. Bevington, D. K. Robinson, "Data reduction and error analysis for the physical sciences", McGraw-Hill, Inc, 1992.
- [6] Knuth, D. E. "Seminumerical Algorithm", 2nd edition, Addison-Wesley, 1981.