

# A Better Method than Tail-fitting Algorithm for Jitter Separation Based on Gaussian Mixture Model

Fangyuan Nan · Yaonan Wang · Fuhai Li ·  
Weifeng Yang · Xiaoping Ma

Received: 27 June 2008 / Accepted: 12 August 2009 / Published online: 10 September 2009  
© The Author(s) 2009. This article is published with open access at Springerlink.com

**Abstract** Jitter is roughly defined as the timing shaking of the square waveforms output from phase locked loops. It consists of two parts: deterministic jitter and random jitter. Separating and identifying each jitter component are important in understanding the root cause of jitter and further in improving on phase locked loop design. A popular method for jitter separation is so-called Tail-fitting Algorithm. A better method than Tail-fitting Algorithm for separating deterministic jitter (DJ) and random jitter (RJ) from total jitter (TJ) is presented in this Letter. The new method targets directly on the original total jitter series, instead of the histogram. Histogram is dependent on bin number and is uncertain, but is inappropriately selected as the starting point of Tail-Fitting algorithm. Our method is based on Gaussian mixture model (GMM). The mathematical relationship between this model and the quantities of DJ and RJ is established. The concept of kurtosis is used to determine the order of GMM, thereby rendering our method fully automatic, highly efficient. Our method circumvents the most

cumbersome difficulty in tail identification of Tail-Fitting Algorithm, because tails and peaks of the histogram, even after being filtered, are fundamentally ambiguously defined, both theoretically and practically. Our method also bypasses the problem of initial value selection.

**Keywords** Phase locked loop · Jitter · Gaussian mixture model (GMM) · Probability density function (pdf) · Signal estimation · Kurtosis

## 1 Introduction

Jitter is roughly defined as the timing shaking of the square waveforms output from phase locked loops; or, is generally defined as the timing-deviation with regard to its ideal position, which in this sense is strictly called instantaneous jitter. In addition to instantaneous jitter, there are several other types of jitter, such as period jitter, cycle-to-cycle jitter [5]. This paper considers period jitters (abbreviated as jitters in the following), which are simply periods from all cycles. But the methodology of jitter separation is the same for other kinds of jitter.

In practice, our measurement is total jitter (TJ). Statistically, total jitter can be classified into two parts: random jitter (RJ) and deterministic jitter (DJ). Random jitter is due to inherent random noise such as thermal noise, shot noise, random modulation, etc. Deterministic jitter changes in a deterministic fashion, as contrary to random jitter. It may come from reflections, cross-talk, electromagnetic interference, systematical modulation, etc. Separating and identifying each jitter component is important in understanding the root cause of jitter and further in improving on phase locked loop design [5].

---

Responsible Editor: K. K. Saluja

F. Nan (✉) · Y. Wang · F. Li  
College of Electrical and Information Engineering,  
Hunan University,  
Changsha 410082, China  
e-mail: nanfangy@hotmail.com

W. Yang  
Institute of Computer and Communication,  
Hunan University of Technology,  
Zhuzhou 412008, China

X. Ma  
School of Information and Electrical Engineering,  
China University of Mining and Technology,  
Beijing, People's Republic of China

## 2 Brief Review of Tail-fitting Algorithm

One popular method of jitter separation used by Wavecrest Company is called Tail-Fitting Algorithm [5]. It has been written to the relevant national standard [6]. The method consists of two steps: *tail identification* followed by *tail-fitting*. That is, to formulate the histogram of the jitter series first, and then to identify the left and right tails of the histogram, and finally to find two Gaussian curves to fit these two tails respectively. After the **two** Gaussian curves are found, the quantities of random jitter and deterministic jitter are given by:

$$\begin{cases} RJ = \frac{\sigma_1 + \sigma_M}{2} \\ DJ = \mu_M - \mu_1. \end{cases}$$

Here,  $\sigma_1$  and  $\sigma_M$  are the standard deviation of left and right Gaussian curves, and  $\mu_1$  and  $\mu_M$  are the mean of left and right Gaussian curves, respectively.

There are several serious drawbacks of the algorithm. First of all, the derivation of the above two formula was not established in [5] whatsoever. Further, fitting the histogram using just two curves is theoretically illegitimate. Generally, we need more than two curves to fit the overall histogram (refer to Appendix). Practically, it is very difficult to identify the correct portions for the left and right tails — the left and right tails were not mathematically defined whatsoever. The analysis in [5] is based on two peaks for the left and right parts of the histogram. The peaks are not clearly defined either, because histogram is not monotonous, but quite fluctuating. Even after the histogram passes through some specially designed filter, which is achieved by a sophisticated artificial intelligence algorithm in [5], there is still no guarantee that the two peaks will be prominently identifiable, particularly for small jitter samples. It is even more difficult to make this step fully automatic, *i.e.*, to use as less user intervention or visual inspection as possible. The difficulty of accurately identifying and distinguishing two tails becomes more serious when DJ portion is very small and thus the two “tails” of the histogram approach very closely to each other.

Another drawback which evaded the attention of [5] is that histogram is dependent on bin-number used in the generation of the histogram. Using histogram as starting point in this problem unnecessarily adds another variable (*more ambiguity*) to original noisy jitter. Clearly, we need try as much as possible to avoid using a variable histogram as our starting point for jitter separation.

The second step of *tail fitting* is relatively much easier: that is, to use a Gaussian curve to fit the tail. It is just a nonlinear optimization problem. The curve fitting algorithm used in [5] is  $\chi^2$  fitting. For nonlinear optimization problem, however, selection of initial parameters needs

attention. If initial values are not appropriate, it may either never converge to the globally optimal point or take a long time to converge.  $\chi^2$  fitting in [5] needs first estimate the initial parameters so that “the initial fitting parameters are close to the final converging values” to avoid those problems. This circuitous reasoning is itself flawed because we don’t know the final converging values. Therefore initial parameter selection is another annoying problem in tail-fitting algorithm.

## 3 Kurtosis—based GMM Approach

### 3.1 Theoretical Result

The main idea of our method differs radically from that of previous method. *The revolutionary change is to turn the wheel around —change starting point of jitter separation from histogram to original jitter waveform.* Instead of fitting histogram curve directly, we change consideration to the original jitter waveform, from which the histogram is generated. Why? We assume the input jitter series are independent realizations of one random variable. If we can estimate the probability density function (pdf) of this random variable, we then equivalently fit the histogram of this random variable. Our main result is stated in the following theorem, the derivation of which is in the Appendix at the end of this Letter.

**Theorem** Consider the situation that a deterministic jitter with unknown shape is immersed in random jitter, the distribution of which is Gaussian:

$$P_{RJ}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$

The general expression for  $P_{TJ}(x)$  is a then a Gaussian Mixture Model (GMM):

$$P_{TJ}(x) = \sum_{j=1}^M b_j \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left[-\frac{(x-\mu_j)^2}{2\sigma_j^2}\right] = \sum_{j=1}^M b_j p(x|j; \mu_j^k, \sigma_j^k) \quad (2)$$

And DJ and RJ are calculated as follows (suppose  $\mu_1 < \mu_2 < \dots < \mu_M$ ):

$$\begin{cases} RJ = \frac{\sigma_1 + \sigma_M}{2} \\ DJ = \mu_M - \mu_1. \end{cases} \quad (3)$$

RJ is characterized by standard deviation  $\sigma$ . In general setting, however, the  $\sigma$  values for the most left and most right components are not the same. Therefore the standard deviation (or RMS) value  $\sigma$  of RJ is taken as the average of these two. DJ is quantified by the peak to peak value, which

is calculated by the distance between two peaks of far left and far right Gaussian components. Based on above theorem, the jitter separation problem becomes parameter estimation of the above model (2).

### 3.2 GMM and Maximum Likelihood Estimation (MLE)

A set of optimal parameters is then sought usually under the criterion of maximum likelihood estimation (MLE), that is, to find the model parameters, which maximize the likelihood of the input data at hand, which are assumed to be a set of samples drawn independently and observing unknown density (2). For a waveform  $\mathbf{x}=[x_1 x_2 \dots x_N]$  of length  $N$ , the GMM likelihood can be written as

$$p(x|\lambda) = \prod_{i=1}^N p(x_i|\lambda)$$

where  $\lambda$  is the collection of all the unknown parameters in (2). That is,  $\lambda = \{(b_i, \mu_i, \sigma_i) \mid i=1, 2, \dots, M\}$ .

However, this expression is a nonlinear function of the parameter  $\lambda$  and direct maximization is not possible. Fortunately, these parameters  $(b_i, \mu_i, \sigma_i) \mid (i=1, 2, \dots, M)$  can be estimated iteratively using EM (Expectation Maximization) algorithm [4], given model number  $M$ .

### 3.3 EM Algorithm

The basic idea of the EM algorithm is, beginning with an initial model parameter set  $\lambda_0$ , to estimate a new model parameter set  $\lambda_1$ , such that  $p(\mathbf{x}|\lambda_1) \geq p(\mathbf{x}|\lambda_0)$ , that is, to monotonically increase the likelihood in each step. The new model parameter set then becomes initial model parameter set until some convergence criterion is reached. For a general description of EM algorithm, refer to [9]. For our GMM model, the algorithm simplifies to a set of iterative formula for obtaining parameters  $\lambda_1$  from initial parameters  $\lambda_0$  [4, 7]:

$$b_j^{k+1} = \frac{1}{N} \sum_{i=1}^N p^k(j|x_i),$$

$$\mu_j^{k+1} = \frac{\sum_{i=1}^N p^k(j|x_i)x_i}{\sum_{i=1}^N p^k(j|x_i)}$$

$$\sigma_j^{2(k+1)} = \frac{\sum_{i=1}^N p^k(j|x_i)(x_i - \mu_j^{k+1})^2}{\sum_{i=1}^N p^k(j|x_i)}$$

where

$$p^k(j|x_i) = \frac{b_j^k p(x_i|j; \mu_j^k, \sigma_j^k)}{\sum_{m=1}^M b_m^k p(x_i|m; \mu_m^k, \sigma_m^k)}$$

And  $p(x_i|j; \mu_j^k, \sigma_j^k)$  is the Gaussian distribution with argument  $x_i$  and parameterized by mean  $\mu_j^k$  and variance  $\sigma_j^k$ , i.e., the individual component in (2).

There are some drawbacks of EM algorithm. The algorithm requires an initialization of the unknown parameter vector  $\lambda$  near the solution, it may get stuck in a local maximum point; and it assumes that the total number of mixing kernels is known in advance. The following kurtosis-based EM algorithm effectively solves all these limitations.

### 3.4 Kurtosis-based EM Algorithm for Automatic Model Order Determination

Determining the number of components  $M$  in a mixture is thus an important, but difficult, problem.

To deal with this problem, we borrow the method in [7] and call it kurtosis—based EM algorithm. The value of the likelihood alone does not provide much information regarding the effectiveness of the fit [7]. A new measure for Gaussian mixtures, called *total kurtosis*, is defined. This measure provides a criterion of how well the estimated Gaussian mixture fits the real data. The smaller the total kurtosis is, the better the fit is. A lower bound for the total kurtosis is zero.

The new algorithm for Gaussian mixture density estimation starts with a small number of kernels ( $M=1$ , in our application), it then performs EM update in order to maximize the likelihood of the data, while at the same time monitors the value of total kurtosis. Based on the progressive change of the total kurtosis, the algorithm performs kernel splitting and increases the number of kernels in the mixture. This splitting aims at making the absolute value of the total kurtosis as small as possible [7]. In short, the algorithm iteratively updates both the number of unknown kernels and the unknown parameters of these kernels from smaller number of kernels by monitoring both the likelihood and the total kurtosis.

Another great advantage of this kurtosis-based EM algorithm is that it nicely bypasses the problem of initial parameter estimation in [5]. Since the algorithm always starts with one mixture component, the selection of mixture parameters greatly simplifies. In the following iterations, the initial parameters are automatically updated from previous results [7]. It always converges to the globally optimal point (within reasonable accuracy tolerance).

## 4 Results

In order to test the validity of our algorithm, we applied it to several sets of signals which were all collected by HP scope. The signals were all FM (frequency modulation)

signals with carrier frequency of  $f_c=100\text{KHz}$ ; modulating frequency  $f_m=1\text{KHz}$ ; and the sampling rate  $f_s=10\text{ MHz}$ . What change were the frequency (equivalently period) deviation from the modulating signal and thus the period jitter of the modulated signal.

Notice, there are two criteria which are to be observed and examined in this specially designed and controlled experiment. The first one is about the *deterministic jitter*, which ought to approximate the maximal deviation in the period of modulating signal. The second one is about random jitter. Because random jitter is one kind of characterization of one particular system, its value should be consistent regardless of the frequency deviations from the modulating signals; by corollary and to be more precisely, its value should approximate the random jitter without any modulating signal present.

To achieve the goal, we first collected results of RJ values from the above signal (with the above  $f_c$  and  $f_s$  setting) without any modulation. Because there was no modulation, the jitter was completely due to random jitter and we used single Gaussian curve to fit the histogram of the period jitter. The standard deviation of the Gaussian curve gave random jitter  $\text{RJ}=4.78\text{ ns}$  for this ideally pure sine wave. The result is illustrated in Fig. 1.

Next, we studied FM100K\_1K\_05K\_10M signal. This notation means that  $f_c=100\text{KHz}$ ,  $f_m=1\text{KHz}$ , maximal frequency deviation (single side)  $\Delta f=0.5\text{KHz}$ ,  $f_s=10\text{ MHz}$ . Therefore,  $f_{\min}=99.5\text{KHz}$ ,  $f_{\max}=100.5\text{KHz}$ ; periods  $T_{\max}=\text{10050ns}$ ,  $T_{\min}=9950\text{ns}$ ,  $\Delta T=100\text{ns}$ . Relative error for DJ was  $\text{REDJ}=(100-94.3)/100=5.7\%$ ; relative error for RJ was  $\text{RERJ}=(5.19-4.78)/4.78=8.6\%$ . The final result for GMM fitting is shown in Fig. 2.

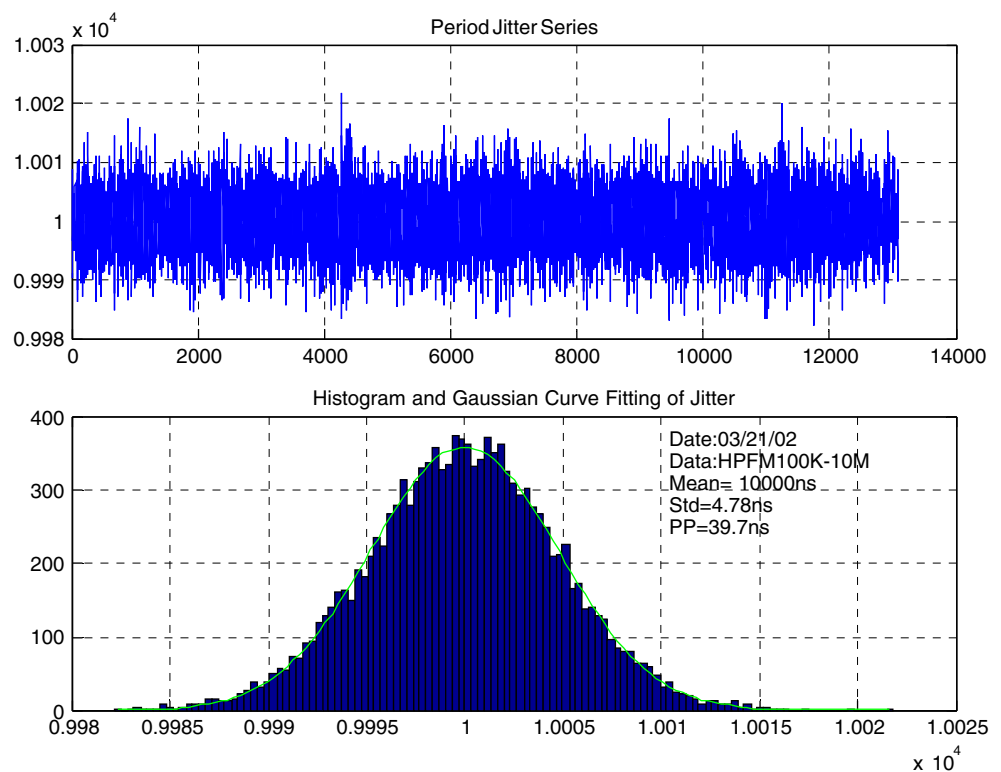
Similarly, keeping all other parameters the same, we studied three other cases (changed the frequency deviation to 0.8KHz, 1KHz, and 2KHz respectively) and summarized all results in Table 1. From Table 1, we conclude this algorithm is very productive, accurate to practically acceptable tolerance (the maximal relative error is within 10%). As far as speed is concerned, it takes about an average of 5 minutes to finish the separation calculation.

## 5 Conclusion

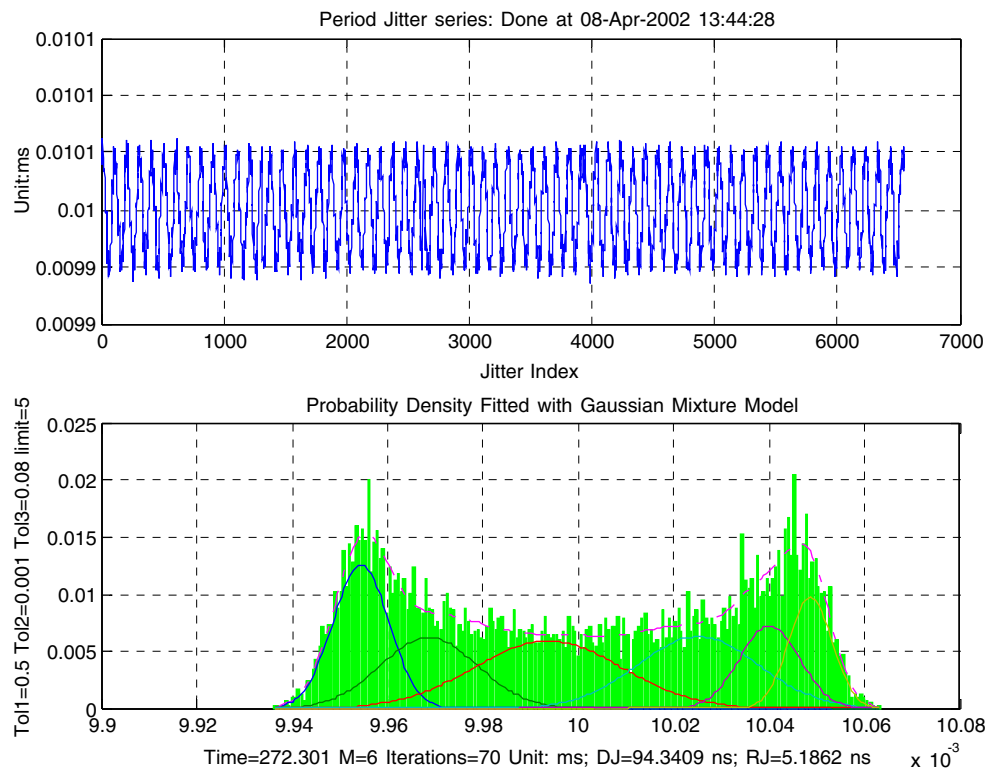
A better method than Tail-fitting Algorithm for deterministic jitter and random jitter separation based on Gaussian mixture model is developed in this paper. The method is productive both in theory and in practice. Theoretically, the mathematical foundation on the relationship between this model and the quantities of DJ and RJ is rigorously established. From this derivation, we further conclude that the method in this paper can be easily extended to deal with a more general problem. That is, to estimate the amplitude of an *unknown* deterministic signal immersed in Gaussian noise.

Practically, our algorithm displays multifold benefits compared with Tail-Fitting Algorithm in [5]. It does not use

**Fig. 1** Determination of RJ from a sine wave signal without modulation



**Fig. 2** Determination of DJ and RJ from a FM signal



the raw histogram, which depends on bin number. Using histogram as starting point in this problem unnecessarily adds another variable (*more ambiguity*) to original noisy jitter. It operates directly on the original jitter series and is therefore much more tractable and stable than Tail-Fitting Algorithm in [5], but the net effect is still fitting the overall histogram by multi-Gaussian curves. It does not distinguish the vague tails, which is most cumbersome in [5]. Our algorithm nicely bypasses the problem of initial parameter estimation in [5]. It always converges to the globally optimal point (within reasonable accuracy tolerance). It uses far less jitter samples for the jitter separation calculation: the average jitter length used in our algorithm is about 5,000, while the Tail-Fitting Algorithm in [5] uses an average jitter length of 50,000. This is a big reduction and therefore our algorithm is much faster.

Our algorithm needs appropriate selection of a set of parameters, particularly one parameter controlling the

convergence of the likelihood. The selection fashion is like tuning the parameters of a wide band antenna designed to receive signals with a wide range of frequencies. However, this is not difficult: the values fall within some reasonable region and can be set by experiment in the lab beforehand. The good news is that, when that parameter changes, the accuracy is still ensured, which is most important, although the number of kernels (which dictates speed) differs.

Our work was performed from fall 2001 to summer 2002. Because of some reasons, we did not consider publishing our work until recently. In order to check the timeliness of our work, we did a recent (up to 2008) literature survey. We searched extensively in some authoritative **journals** related to this area, such as, the Journal of Electronic Testing, Measurement, IEEE Journal of Solid-State Circuits, IEEE Transactions on Circuits and Systems (I and II), IEEE Trans. on Instrumentation and Measure-

**Table 1** Summary of DJ and RJ values and their relative errors from 5 experiments

$\Delta f$ (KHz)	$\Delta T$ (ns)	DJ (ns)	RelErrDJ	RJ (ns)	RelErrRJ
0	N/A	N/A	N/A	4.78	N/A
.5	100	94.3	5.7%	5.19	8.6%
.8	160	155.1	3.1%	4.91	2.7%
1	200	192.9	3.6%	5.03	5.2%
2	400	391.7	2.1%	4.86	1.7%



ment, etc. **We did not find any specific algorithms to overcome the shortcomings of Tail-fitting Algorithm.** However, indeed, we found five **conference papers** [1–3, 8, 10] which proposed new jitter decomposition techniques. **Unfortunately, they did not point out the drawbacks of Tail-fitting Algorithm either.** So we are confident that our work is still timely and novel at least in the aspect this Letter's title claims. Limited by the page length of this **short Letter**, a comprehensive comparison of our algorithm with those in [1–3, 8, 10] is beyond scope of this Letter and may be done in the future.

In other words, we confine our discussion in this Letter to the specific comparison of our algorithm with Tail-fitting Algorithm, nothing else.

### Appendix—derivation of the total Jitter's pdf from those of DJ and RJ

It is a fact that an unknown pdf of a random variable can be approximated by a weighted finite sum of Gaussian pdfs with different means and covariance matrices. While representing an unknown pdf as a GMM is nothing new, how the quantities of DJ and RJ are linked with the most and right Gaussian kernels, that is, how Eq. 3 come from, is rigorously established in this [Appendix](#), which is one contribution of this paper.

First of all, since total jitter TJ is the sum of random jitter (RJ) and deterministic jitter (DJ),

$$P_{DJ}(x) = \sum_{i=1}^M b_i \delta(x - \mu_i) \quad \mu_1 < \mu_2 < \dots < \mu_M$$

The pdf of TJ is the convolution of those of DJ and RJ. We regard DJ as realizations of a random variable, which is BOUNDED. The pdf of DJ can therefore be approximated as a sum of discrete probability function, that is, a series of  $\delta$  functions:

where  $b_i$ 's satisfy the constraint that

$$\sum_{i=1}^M b_i = 1$$

It is expected that the more number of kernels  $M$  is used, the more accurate this approximation and the better  $(\mu_M - \mu_1)$  approximates the peak-peak value of the unknown deterministic signal. In practice, only finite number of  $M$  is used.

The pdf of RJ is Gaussian as expressed in (1). Therefore,

$$P_{TJ}(x) = \sum_{i=1}^M b_i \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{(x-\mu_i)^2}{2\sigma^2}}$$

In general,  $\sigma$  may not be the same at different locations. Therefore, we generalize the form above to that in (2) as the pdf of the total jitter, and it is exactly so-called GMM.

This Appendix lays more general and more solid foundation for our method, and the related work in this field, e.g., it can substitute relevant portion in [6].

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

### References

1. Cai Y, Werner S, Zhang G, Olsen M, Brink R (2002) "Jitter Testing for Multi-Gigabit Backplane SerDes – Techniques to decompose and combine various types of jitter", IEEE Proc. International Test Conference, Oct. 2002, pp.7-10
2. Dou Q, Abraham J (2006) "Jitter decomposition by time lag correlation", IEEE Proc. Quality, Electronic Design, 7th International Symposium on , March 2006, p.6
3. Dou Q, Abraham J (2008) "Jitter Decomposition in High-Speed Communication Systems", 13th European Test Symposium, 2008, pp. 157–162
4. Reynolds Douglas A, Rose Richard C (1995) "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", IEEE Trans. On Speech and Audio Processing, Vol. 3, No. 1, pp 72–95, January 1995
5. Li Mike P, Wilstrup Jan, Jessen Ross, Petrich Dennis (1999) "A New Method for Jitter Decomposition Through Its Distribution Tail Fitting", IEEE Proc. International Test Conference Proceedings, 1999, pp. 788–794
6. National Committee for Information Technology Standardization (NCITS), working draft for "Fiber Channel — Methodologies for Jitter Specification", Rev. 10, June 1999.
7. Vlassis Nikos, Likas Aristidis, (1999) "A Kurtosis-Based Dynamic Approach to Gaussian Mixture Modeling", IEEE Trans. On Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 29, No. 4, pp 393–399, July 1999
8. Ong CK, Hong D, Cheng K.-T, Wang L.-C (2004) "Random jitter extraction techniques in a multi-gigahertz Signal", IEEE Proc. Design, Automation, and Test in Europe Conference and Exhibition, Feb. 2004, pp. 286–291
9. Srinath MD, Rajasekaran PK, Viswanathan R (1996) Introduction to Statistical Signal Processing with Applications. Prentice Hall, Englewood Cliffs
10. Yamaguchi TJ, Hou H X, Takayama K, Armstrong D, Ishida M Soma M (2007) "An FFT-based Jitter Separation Method for High-Frequency Jitter Testing with a 10X Reduction in Test Time", IEEE International Test Conference, 2007, paper 16.3, pp.1–8