Computer assisted mathematics, 2019
№ 1: 32–54
http://cte.eltech.ru

INFERRING TRIGGER NOISE FROM TRIGGER JITTER

Lamarche F.[1], PhD, ✉ francois.lamarche@teledyne.com

[1]Teledyne LeCroy, 700 Chestnut Ridge rd., Chestnut Ridge, NY 10977 USA

Abstract

We address the relationship between voltage noise at the input of a threshold comparator and random changes in the timing of the comparator output toggle timing (jitter). Models that go beyond the naïve linear relationship are presented and shown to match experiment. They help reconcile predictions with verifiable device performance; moreover, they allow an approximate measurement of the noise bandwidth without the help of high-frequency equipment. In the course our investigation, a PC/laptop toolset was constructed that lets a neophyte study noise via jitter. This toolset can also be used in education to teach the notion that a biased view of a random phenomenon can arise from triggering on the tails of its distribution.

Keywords: *jitter, noise, bandwidth, threshold, discriminator, hysteresis, skewness.*

Citation: F. Lamarche, "Inferring Trigger Noise from Trigger Jitter," *Computer assisted mathematics*, no. 1, pp. 32–54, 2019.

1. INTRODUCTION

A very useful technique for measuring noise present on a signal is to evaluate it from the jitter present in the digital signal generated when that signal is fed to a discriminator [1, 2]. In the context of an oscilloscope, the conversion of an analog signal to a digital signal marking the crossing of a threshold is known as a trigger, but this technique can be used during the characterization of any discriminator. Knowing the slope (e.g. in volts per second) of the signal near the discriminator threshold, one can naively estimate that the jitter is the noise divided by the slope, and that constitutes a correct order-of-magnitude estimate.

However, as we will discuss here in detail, the presence of hysteresis changes the proportionality between noise and jitter in a somewhat non-intuitive way, especially when the slope is small.

We start with a model of voltage changing quasi-linearly with time, $V = St$, where S is the slope. In practical terms, that voltage is very often coming from a sinusoidal signal generator and the slope near zero volts is then $S = 2\pi f\sqrt{2}V_{\mathrm{rms}}$.

After going through a series of simple cases, we will mathematically analyze the practical case of a discriminator and increasingly slow signals and compare theory with an example of experimental measurements. As will be shown, this technique allows one not only to estimate more correctly the voltage noise, but also predict skewness of the jitter distribution, and estimate, very approximately, the bandwidth of that noise.

Someone wanting to characterize noise in a comparator using this method can use the tools we have developed for this purpose, which are: script parameters for the oscilloscope used to collect the data, that allow to directly display skewness, EXCEL spreadsheets to compare Gaussian models with experimental data, MATLAB that simulates the trigger process, and VISUAL BASIC scripts that optimize the joining of data about the noise distribution acquired at several different slopes.

2. PARTICULAR LIMIT CASES

2.1. Limit case of small noise and no hysteresis, and white noise bandwidth is low

Let us write $St = \alpha W(t)$, α being a very small constant, and white noise function $W(t)$ is normalized by $< W(t)^2 >= 1$. Solving $St = \alpha W(t)$ for the crossing time gives $t_n = \alpha W(t)/S$. We can write this because, due to the low noise bandwidth, $W(t)$ is a constant which only varies from one threshold crossing to the next. The white noise function $W(t)$ having a root mean square of 1, the distribution of times t_n where the voltage meets the noisy threshold will have a root mean square of $|\alpha/S|$. As shown in figure 1, there is a distinct unique crossing point each time we run the threshold crossing experiment. As shown by the dashed lines, the threshold can vary, but it does not change on the time scale of the input voltage change. It is a valid approximation only if the white noise bandwidth is much less than α/S.

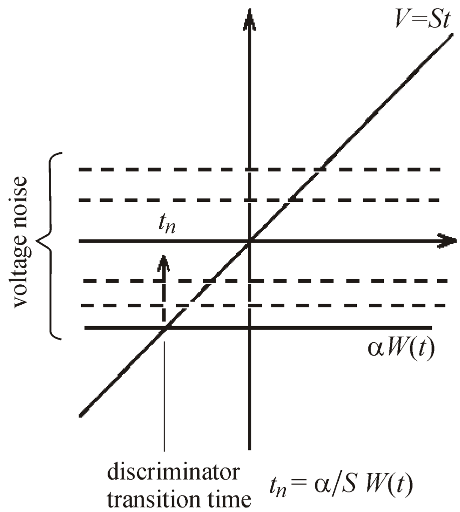

Figure 1. Flat threshold case

2.2. Limit case of small noise, no hysteresis, but high frequencies of white noise

The above estimate of jitter is nice and simple, but, if the noise is indeed white noise, it can never be assumed that the noise remains the same. Even in an arbitrarily small vicinity of the time origin, the white noise can have an infinity of different values, and therefore an infinite number of zero crossings. In the absence of hysteresis, each of those zero crossings contributes a point to the distribution of threshold crossing times. The probability of a crossing the threshold per unit time is still dependent on the noise-free voltage, St, and the probability that two consecutive random numbers from the probability distribution $W(t)$ are either both smaller than St/α or both larger than St/α can be estimated from the integral of the probability distribution $\mathfrak{L}(W)$ from St/α to infinity, also known as the cumulative distribution function. To illustrate with an example, let us assume a Gaussian probability distribution $\mathfrak{L}(W)$. Under that assumption, the probability that one of two consecutive points is smaller than St/α while the other is larger than St/α is proportional to a product of error functions:

$$\operatorname{erfc}\left(\frac{St}{\sqrt{2}\alpha}\right)\cdot\operatorname{erfc}\left(-\frac{St}{\sqrt{2}\alpha}\right). \tag{1}$$

The mean square jitter can be estimated by the second moment of this probability density function, i.e. by calculating:

$$\frac{2\alpha^2\int_{-\infty}^{+\infty}\left(1-\operatorname{erf}^2(\tau)\right)\tau^2 d\tau}{S^2\int_{-\infty}^{+\infty}\left(1-\operatorname{erf}^2(\tau)\right)d\tau}. \tag{2}$$

The primitive of $1-\operatorname{erf}^2(\tau)$ is $\tau-\tau\operatorname{erf}^2(\tau)-2/\sqrt{\pi}\operatorname{erf}(\tau)e^{-\tau^2}+(2/\pi)^{1/2}\operatorname{erf}(2^{1/2}\tau)$, which evaluates to $2(2/\pi)^{1/2}$ between $-\infty$ to $+\infty$.

The primitive of $2\left(1-\operatorname{erf}^2(\tau)\right)\tau^2$ is

$$2/3\Big(\tau^3(1-\operatorname{erf}^2(\tau))-2/\sqrt{\pi}\,\tau^2\operatorname{erf}(\tau)e^{-\tau^2}-2/\sqrt{\pi}\operatorname{erf}(\tau)e^{-\tau^2}$$
$$+(2/\pi)^{1/2}\operatorname{erf}(2^{1/2}\tau)-\tau e^{-2\tau^2}/\pi+1/(8\pi)^{1/2}\operatorname{erf}(2^{1/2}\tau)\Big)$$

which evaluates to $2/3(2(2/\pi)^{1/2}+2/(8\pi)^{1/2})=5/3(2/\pi)^{1/2}$ between $-\infty$ to $+\infty$. In this extreme assumption, there are several transitions of the discriminated signal each time the signal traverses the threshold; the jitter can still be defined as the root mean square of these transition times, and it is equal to $(5/6)^{1/2}|\alpha/S|$.

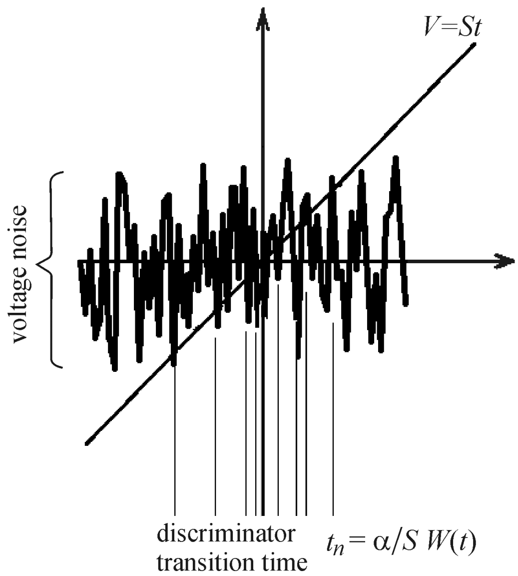

Figure 2. High Bandwidth case

A conceptual representation of the situation in this case is shown in figure 2. Note that, at times near the average crossing time, the frequency of discriminator output transitions per unit of time is of the order of the white noise bandwidth.

One can wonder how allowing several transitions can make the trigger jitter smaller in proportion to the voltage noise than allowing a single transition. This result actually depends on the probability distribution of the “white noise”. In the extreme case of a bi-modal distribution of noise, one can see the intuitive reason. For any time between the crossing of the threshold for the first hump of the distribution and the crossing of the threshold for the second hump of distribution, there is a uniform probability of transition. Therefore, using the well-known result [3] that the width of a “rectangular box” distribution is the width divided by the square root of twelve, and that the root mean square of the numbers in the set $\{-0.5, 0.5\}$ is 0.5, jitter is $(1/12)^{1/2}/0.5 = (1/3)^{1/2}|\alpha/S|$. In intuitive terms, transitions occur more near the zero-crossing compared with the vertical noise distribution having values near zero.

2.3. Handling finite hysteresis

A good discriminator will provide enough hysteresis to deal with noise, so that there is just one transition of each polarity per cycle of the generator. The question then becomes, what is the cumulative probability as a function of time, $P_{\mathrm{trig}}(t)$, that the discriminator fires? For that, we must consider a two-term differential equation:

$$\frac{dP_{\mathrm{trig}}}{dt} = \frac{S}{\alpha}\mathrm{PDF}\left(\frac{St}{\alpha}\right)SP + f_0\left(1 - P_{\mathrm{trig}}\right)\mathrm{CDF}. \tag{3}$$

CDF stands for cumulative distribution function, and PDF for probability density function. A factor $SP = (1 - P_{\mathrm{trig}})/\mathrm{CDF}$ is needed to consider the reduction of the available sample population due to the other term, but from a didactic point of view, it should be discussed only later as a detail; when the first term dominates, we can write $SP = 1$.

The first term represents the probability that the voltage change during a dt time interval causes is crossing over a threshold (hence the PDF, that measures the probability of the threshold being at that level). The second term represents the probability that, at constant voltage, a random process yields a different noise level, and that this new noise levels lands on a region that yields a trigger. In the next paragraph we consider the limit case of the first term dominating, and in the following one, the limit case of the second term dominating.

2.4. Limit case of small noise and finite hysteresis, and white noise bandwidth is low

We identify $W(t)$ of the first section with one random number pulled from the noise density “PDF” above.

We follow the discussion of that section, but instead of looking for $V_{\mathrm{c}} = 0$, we are looking for crossing a voltage $V_{\mathrm{c}} = \frac{1}{2}V_{\mathrm{hyst}}$, the hysteresis voltage V_{hyst} being the minimum amount of voltage change needed to overcome an opposite state of the comparator. After the threshold comparator output goes to the high state when the signal at the input of comparator exceeds $\frac{1}{2}V_{\mathrm{hyst}}$, it would have to go below $-\frac{1}{2}V_{\mathrm{hyst}}$ for the threshold comparator output to go to the de-asserted state, difference being $\frac{1}{2}V_{\mathrm{hyst}} - (-\frac{1}{2}V_{\mathrm{hyst}}) = V_{\mathrm{hyst}}$.

We then write the output transition time $t_T = (V_c - \alpha W(t))/S$. The white noise function $W(t)$ having a root mean square of 1, the distribution of times $t_T = V_{\mathrm{c}}/S$ will have a root mean square of $|\alpha/S|$. For a large enough hysteresis, and low enough noise and/or large enough slope, this crossing point is unique. Note that this perturbation theory approach which give us this very simple result has the underlying assumption that, for small typical time differences, $W(t)$ does not change. The conceptual representation of figure 1 still applies, the average position of the threshold (dashed lines) is simply shifted from zero to $\frac{1}{2}V_{\mathrm{hyst}}$. Under that assumption (corresponding to a white noise bandwidth much less than S/α), the noise $W(t)$ is a different “constant” at each crossing.

2.5. Limit case of small noise and finite hysteresis, and white noise bandwidth is large

What happens in this case is that the white noise, “riding” on a linearly increasing voltage, makes a very large number of low-probability attempts at exceeding the comparator’s threshold. Over time, the probability of firing the comparator per unit time is slowly increasing, and the cumulated probability is also increasing. Once this threshold has been reached, because of hysteresis, it is very unlikely that the white noise will cause the comparator output to switch again. The probability of having met the threshold is thus a function $P_{\text{trig}}(t)$ of time t, with an asymptote of 1, and it related to the probability $P(t)$ of not having met the threshold yet by the equation

$$P_{\text{trig}}(t) = (1 - P(t)).$$

It turns out it is more convenient to work with $P(t)$ than with $P_{\text{trig}}(t)$, because the probability $P(t)$ is satisfying this differential equation:

$$\frac{dP(t)}{dt} = -f_0 P(t) \int_{-tS/\alpha}^{+\infty} W(\tau)\, d\tau$$

where the integral runs from $\tau = -tS/\alpha$ to $+\infty$, and f_0 represents the “white noise bandwidth” (number of independent attempts at crossing the threshold per unit time).

This differential equation can be rewritten by setting $P(t) = \exp(L(t))$, $L(t) = \ln(P(t))$

$$\frac{dL(t)}{dt} = \frac{S}{\alpha}\frac{dL(\tau)}{d\tau} = -f_0 \int_{\tau}^{+\infty} W(\tau)\, d\tau = -f_0 C(\tau) \tag{4}$$

which ultimately means the logarithm of cumulative jitter distribution $L(\tau)$ is, up to a factor $-\alpha f_0/S$, the CDF (cumulative distribution function) of noise $C(\tau)$.

To find out the shape of the distribution of noise $W(\tau)$, via its cumulative distribution $C(\tau)$, the following steps should suffice:

a) Measure experimentally the distribution $T(t)$ of trigger times, by building a histogram of trigger times of a large quantity of trigger events,
b) Numerically integrate T (backward) to get the cumulative $P(t)$, note that $P(t) \to 1$ for $t \to -\infty$,
c) Take the logarithm of this function, call it $L(t)$,
d) Take the numerical derivative with respect to τ, τ being $(S/\alpha)t$,
e) Divide it by the dimension-less $-\alpha f_0/S$, and you obtain $C(\tau)$.

While in theory this works, in practice, the statistical errors involved in acquiring a function from a cumulative histogram cause either inaccuracy in the vertical axis, coarse granularity of the horizontal axis (time), or both. Taking the logarithm will give a relatively high importance to poorly populated regions of the histogram, and taking the numerical derivative is also a process that tends to increase the local uncertainty in relative terms. The statistical errors overwhelm the function value to such a point that the shape of the noise might not be recognizable. This is especially true in terms of the tails of that noise distribution, where statistical errors are worse.

What is interesting is that by changing the slope of the signal, the constant $\alpha f_0/S$ can be changed by orders of magnitude. As we will see later, this allows a special technique to “probe” the tails of the W distribution.

2.6. Gaussian noise assumption for case of finite noise, large noise bandwidth

For many practical purposes, most experimenters are ready to assume that the noise distribution is a Gaussian (a.k.a. Normal distribution). Owing to the Central Limit theorem, the noise distribution is often Gaussian as a good approximation. In any case, for the average experimentalist or engineer, the assumption of a Gaussian represents a useful approximation — he or she cares mostly about the standard deviation and not so much about the exact shape. For the most precise applications, one may use the n-degrees-of-freedom Student distribution (which becomes Gaussian in the large number-of-degrees-of-freedom limit), or linear combinations thereof — but one should always start with the Normal distribution.

Solving the above equation (4) in the case where $W(\tau)$ is Gaussian means the cumulative distribution function $C(\tau)$ is $\frac{1+\mathrm{erf}(\tau/\sqrt{2})}{2} = \frac{1}{2}\mathrm{erfc}(-\tau/\sqrt{2})$.

The primitive of that is of the form:

$$\frac{\tau}{2}\mathrm{erfc}\frac{-\tau}{\sqrt{2}} + \frac{e^{-\tau^2/2}}{\sqrt{2\pi}}.$$

And up to a coefficient and an additive constant, this is the $L(\tau)$ function for the Gaussian case.

For this Gaussian case, we can then write the cumulative distribution function $P(\tau) = \exp\left(-\frac{\alpha f_0}{\sqrt{2}S}(z\,\mathrm{erfc}(-z) + e^{-z^2}/\sqrt{\pi})\right)$ where $z = \tau/\sqrt{2}$.

$P(\tau)$ is cumulative starting at $+\infty$, the usual "cumulative starting a $-\infty$" is $P_{trig}(\tau) = 1 - P(\tau)$.

Then the probability density function is:

$$\frac{\alpha f_0}{\sqrt{2}S}\mathrm{erfc}(-z)\exp\left(-\frac{\alpha f_0}{\sqrt{2}S}\left(z\mathrm{erfc}(-z) + \frac{e^{-z^2}}{\sqrt{\pi}}\right)\right).$$

Writing $N = \frac{\alpha f_0}{\sqrt{2}S} = 0.7N_{\mathrm{eff}}$, where N_{eff} is the "effective number of threshold-crossing attempts per tau", we can then calculate the n-th moment for any value this parameter N:

$$I(n,N) = \int_{-\infty}^{+\infty} x^n N\big(1+\mathrm{erf}(x)\big)\exp\left(-Nx(1+\mathrm{erf}(x)) - Ne^{-x^2}/\sqrt{\pi}\right)dx.$$

The Cumulative Distribution Function (CDF) and Probability Density Function (PDF) that we obtain here are similar, but not equivalent, to the Gompertz CDF and PDF. Like the Gompertz, there exist only very complicated expressions for the moments — and numeric integration is likely the best way to evaluate these moments. Also, like the Gompertz distribution, the PDF has a relatively gradual increase followed by an abrupt fall [4]. In the limit of small "a" parameter of the Gompertz distribution, the shape of the distribution becomes identical to the shape of our distribution when the slope S tends to zero: the CDFs will share the form $1 - \exp(-\exp(\tau))$. The moments M_n of order n of that "extreme" Gompertz distribution have the particularly simple form $M_n = (-1)^n\zeta(n)$ where ζ is the Riemann zeta function.

The mode of threshold time distribution is obtained by taking the derivative and solving for it to be zero. This occurs at negative τ_m such that

$$N = \frac{2}{\sqrt{\pi}} \cdot \frac{e^{-\tau_m^2/2}}{\big(1+\mathrm{erf}(\tau_m/\sqrt{2})\big)^2}.$$

Using the approximation [6] for $1 + \mathrm{erf}(x) = \mathrm{erfc}(-x)$ in terms of $\exp(-x^2)$,

$$1 + \mathrm{erf}(x) \cong \frac{e^{-x^2}}{-x\sqrt{\pi}}.$$

We find an approximate expression for the mode:

$$\tau_m \cong -\sqrt{2\mathscr{W}\left(\frac{N}{2\sqrt{\pi}}\right)}.$$

$\mathscr{W}$ being the Lambert function [9].

To obtain this result, we have used $N = 2/\sqrt{\pi} \cdot e^{-\tau_m^2/2}/\left(1 + \mathrm{erf}\left(\tau_m/2\right)\right)^2 \cong 2e^{\tau_m^2/2}\sqrt{\pi}\tau_m^2/2$ then used the definition of $\mathscr{W}$ as the reciprocal of $x\exp(x)$.

The moments of the PDF for the Gaussian noise are best calculatable either numerically or with infinite series. However, it is sufficient to look at the second and third derivatives of the PDF at its mode to realize two important aspects:

1. As the N parameter is increased, the value of the PDF's second derivative,

$$-\left[N^3\mathrm{erfc}(-\tau)^3 - (6N^2\mathrm{erfc}(-\tau)e^{-\tau^2} + 4N\tau e^{-\tau^2})/\sqrt{\pi}\right]\exp\left(-N(\mathrm{erfc}(-\tau)\tau + e^{-\tau^2}/\sqrt{\pi})\right)$$

if evaluated at the mode, simplifies to:

$$-8\left(\frac{e^{-\tau_m^2}}{\sqrt{\pi}}\right)\left[2 + \tau_m\mathrm{erfc}\left(-\tau_m\right)/\left(\frac{e^{-\tau_m^2}}{\sqrt{\pi}}\right)\right]\exp\left(-N\left(\tau_m\mathrm{erfc}(-\tau_m) + \frac{e^{-\tau_m^2}}{\sqrt{\pi}}\right)\right)/\mathrm{erfc}(-\tau_m)^3$$

that we divide by the value of the PDF at the mode to get the "relative second derivative":

$$-4\frac{e^{-2\tau_m^2}}{\pi}\left[2 + \tau_m\mathrm{erfc}(-\tau_m)/\left(\frac{e^{-\tau_m^2}}{\sqrt{\pi}}\right)\right]/\mathrm{erfc}(-\tau_m)^2.$$

In this expression, the content of the square bracket converges to a value near unity, while the other factors converge to $-4\tau_m^2$. The inverse of the square root of the absolute value of the "relative second derivative" represents the root mean square value for the normal (a.k.a. Gaussian) distribution that fits the PDF near its mode. As such, it represents a coarse estimate of the r.m.s. jitter of the threshold crossing position, and we see that it decreases as

$$1\bigg/\sqrt{2\mathscr{W}\left(\frac{N}{2\sqrt{\pi}}\right)}.$$

In fact, in the "extreme Gompertz" limit, the second moment is $\zeta(2) = \pi^2/6$ times the value of the PDF at its maximum (mode) divided by its second derivative there. So, for very large N, an asymptotic value for the jitter is

$$1\bigg/\sqrt{\frac{12}{\pi^2}\mathscr{W}\left(\frac{N}{2\sqrt{\pi}}\right)}. \quad (5)$$

2. If the time distribution were symmetric, then both the skewness and the third derivative at the peak would be zero. If we calculate this third derivative at the point where the first derivative is zero, and that this third derivative is not zero there, it strongly suggests that the third moment (which is called skewness after normalizing it to the cube of the r.m.s.)

is non-zero. The evaluation of the third derivative at the point where the first derivative is zero yields a ratio, with a positive denominator, and this numerator:

$$4+8\tau\Big/\Big(\frac{e^{-\tau^2}}{\sqrt{\pi}}\Big)\mathrm{erfc}(-\tau)+(1-\tau^2)\Big/\Big(\frac{e^{-\tau^2}}{\sqrt{\pi}}\Big)\mathrm{erfc}(-\tau)^2.$$

This expression is negative for all values of τ smaller than -0.15405. This means that except for an extreme case, the PDF has a negative third derivative and negative skewness — and indeed, this is what numeric integration confirms. For positive values of τ, we would have positive skewness, but the hypothesis of (equation 2) has to be abandoned in favor of those of (equation 1), and indeed the skewness never goes positive.

3. ILLUSTRATION WITH SOME FITTED EXPERIMENTAL DATA

After describing an example of a data taking setup for jitter distribution data, we will describe a practical method how to piece together the data concerning the noise PDF acquired using very different frequencies (and therefore very different slopes at the threshold crossing). We will also measure the root mean square jitter in each frequency/slope/effective number setting, yielding a curve that we will attempt to fit with a model of jitter r.m.s. width originating from the assumption of a Gaussian noise PDF. The same fitting exercise will be repeated with the skewness of the distribution as a function of the frequency/slope/effective number setting. All these fits allow us to estimate roughly the bandwidth of the white noise, to which we will finally give a physical interpretation.

3.1. Experimental set-up

To take the measurements, as can be seen in figure below, a low-noise dual-output arbitrary-function generator (Siglent SDG1050) is connected via coaxial cables to the device under test. The device under test is a differential threshold comparator whose input noise we wish to characterize. The signal generators have a phase control for each output, the first output is programmed with a phase of 0°, while the second output is programmed with a phase of 180°. Using two “T” coaxial adapters, the two differential signals are also connected to the 1MΩ inputs labelled C3B and C4B of a LeCroy Wavemaster 820Zi-A. On this oscilloscope, a math function labelled “F1” computes C3-C4, reconstituting the differential signal present at the input of the device under test.

The output of the device under test is connected via a 50 Ω coaxial to the 50 Ω input labelled C2A of the oscilloscope. The oscilloscope is set to trigger on positive edges on C2A, i.e. on negative-to-positive transitions of the output of the Device Under Test. The output impedance of the device under test is also approximately 50 Ω, so that a reflection-free transmission of the discriminator output is available as a trigger source on the oscilloscope. On this oscilloscope, the complex Fourier transform is available as a math function labelled “F2”, a parameter labeled “P2” then evaluates the phase of the “F1” signal at the frequency which was programmed into the Arbitrary Function Generator. A math function labelled “F3” then records a hundred-bin histogram of “P2”, the phase of the signal. The FFT (Fast Fourier Transform) is used to calculate this transform, so that the millions of points (up to 10 million) that the oscilloscope has acquired can quickly yield the phase of the differential signal with excellent precision (better than $0.01\big/\sqrt{10^7}$ radians, or 0.57m°). Knowing the frequency of the sinusoidal output of the Arbitrary function generator, the distribution of phase recorded in the “F3” histogram then corresponds to a distribution of trigger times via the equation t.f=angle(°)/360°.

The oscilloscope has a parameter (labelled P3) that directly computes the root mean square deviation from the histogram of signal phases, and for high enough statistics in the histogram, it gives a precise measurement of the second moment of the threshold time distribution. The oscilloscope does not have a built-in skewness calculation, but the Param script capability makes it easy to add one (see appendix A).

Since we feed a differential sine wave of a known amplitude $\sqrt{2}V_{\text{rms}}$, the naïve/low noise bandwidth expectation for root mean square of the time jitter is very simply $\alpha\big/(2\pi f\sqrt{2}V_{\text{rms}})$, so the root mean square of the phase distribution width would naively be a constant: $\alpha\big/(\sqrt{2}V_{\text{rms}})$, in radians[1] Simply put, any change in the width of the angle distribution when changing the frequency is a manifestation of non-naïve models.

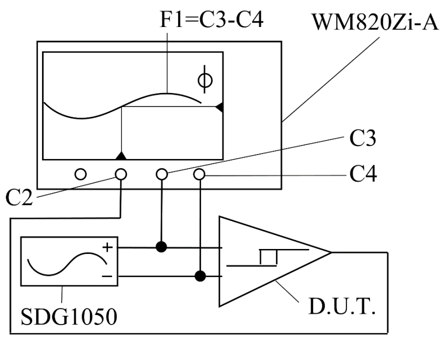

Figure 3. Setup

3.2. Direct reconstruction of the cumulative noise distribution: a practical methodology

The measurements taken with a high slope S_H describe well the center of the noise distribution. We apply the above steps “a” through “e”, taking as many points as possible during step “a”. Even if the distribution is measured with a very large number of points, there comes a point where the left-hand side of the distribution has large statistical errors.

The measurements taken with a smaller slope S_L also represent an attempt to evaluate $C(\tau)$. However, for similar statistics, we are obtaining smaller values of $C(\tau)$, corresponding to smaller values of the dimension-less “τ” variable. A smaller slope means comparatively more “early trigger opportunities”. When the statistical error associated with the “S_H” evaluation becomes larger than the statistical error associated with the “S_L” evaluation, “S_L” becomes the accurate source of the $C(\tau)$ evaluation instead of “S_H”. $C(\tau)$ being continuous, we can accomplish a harmonization of the two evaluations by τ-shifting the distributions until they have the same average in the region between the mode of one and the mode of the other (mode meaning bin of maximum population). The detailed algorithm finding the modes and locating the best shift is shown in figure 7 [Appendix B].

This can be repeated with measurements taken with a yet smaller slope S_{L2}, and then repeated with measurements taken with a yet smaller slope S_{L3}, etc...

An ideal choice of slopes is a geometric progression, with the ratio between successive slopes being about 2 to 3, to quickly span many orders of magnitude. An example of a

[1] To obtain the value in degrees of angle, multiply by $180°/\pi$.

composite-slope evaluation of $C(\tau)$ is shown in figure below, the twelve slopes being in the ratio {100:200:400:1400:3400:5000:10000:25000:55000:105000:205000:405000} owing to the choice of frequencies as 100, 200, 400, 1400, 3400, 5000, 10000, 25000, 55000, 105000, 205000, and 405000 Hertz.

On this figure, using a log scale for the vertical axis, a fit of $C(\tau)$ with the cumulative normal distribution $(1+\text{erf}(\tau))$ looks pretty good, however, one must keep in mind that, on a logarithmic scale, significant deviations can seem insignificant. In the presence of significant $1/f$ noise [7, 8], we can logically expect tails that deviate from the normal distribution. The vertical axis is in arbitrary units but could easily be normalized to unity. The horizontal axis is the phase of the histogram bins used in this reconstruction, in degrees of angle. Considering the systematics that affect both the channel and trigger of the oscilloscope, some of the distributions have been shifted in angle to let the curve elements match better. The fit has three free parameters, and its equation is:

$$7.9433 \times 10^4 \left(1 + \text{erf}\left(\frac{\phi - 94.43^\circ}{0.088^\circ}\right)\right).$$

Of the three parameters of the fit, only the denominator angle of 0.088° has physical meaning concerning the amount of noise present at the input of the discriminator.

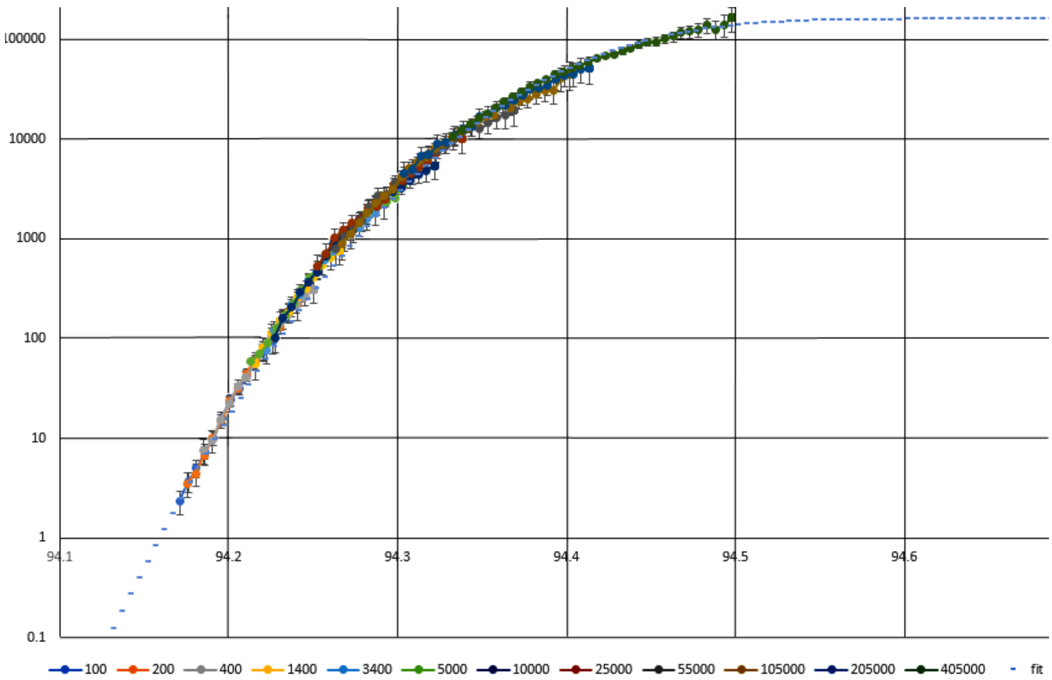

Figure 4. Composite CDF

Noise distribution naturally tend to be polarity symmetric. This method does not provide equal accuracy for the distribution of positive noise as it does for negative noise. Even after accumulating millions of points, at the highest frequency, the histogram we build is barely starting to probe the CDF beyond its central inflexion point. One method would be to assume that the noise is symmetric. A potentially better solution, if we want to see possible differences between the two sides of the noise distribution, would be to change the trigger polarity (e.g. set the oscilloscope trigger to negative edges on C2A, for the experimental setup described above).

3.3. Visual Basic tool to merge several histograms into one CDF fit file

In appendix D is a program written in VB script that reads several histogram data files and merges them. A vital information when wanting to assemble these into a common noise CDF is the frequency (equivalently period, slope, ...) at which each histogram was taken, so the file are labelled with the frequency, i.e. part of the file name of each histogram file is the frequency.

The important steps are:

1. The arguments of the script are C file names, loop “I” over them until argument count is C-1:
   a) We get the i-th frequency from the file name,
   b) We get the i-th histogram from the file contents.
2. Using indices j and k, we sort by increasing frequency both the frequencies and the histograms.
3. We write five “header” rows, comprising 10 columns per histogram:
   a) The first row contains groups of 10 cells, each group having, in the first cell, the frequency in Hz, and in the seventh cell a barycenter formula based on the sum of the products of 4c) below, and in the tenth cell, a formula calculating the square root of the second moment using the eighth column of each group of 10 columns,
   b) In the second row, the tenth cell in each group of 10 calculates the statistical error on the second moment calculation,
   c) In the third row, the tenth cell in each group of 10 calculates the period (in μs) of the sine source,
   d) In the fourth row, the tenth cell in each group of 10 is set to the 2nd moment calculated in a),
   e) In the fifth row, the tenth cell in each group of 10 is set to a scaled version of the error in b).
4. We write 100 histogram rows, in each group of 10:
   a) First represents the bin position harmonized with the algorithm of appendix B,
   b) Second column represents the bin population,
   c) Third has a formula multiplying the first and second column for the purpose of calculating the histogram’s barycenter,
   d) Fourth column has a formula cumulating the bin population from bottom to top,
   e) Fifth column has a formula for the statistical error on the reconstructed noise CDF, set to zero if there is too much relative error,
   f) sixth column has a formula for the reconstructed noise CDF, set to zero if above is zero,
   g) seventh column is left empty,
   h) The eighth column has a formula multiplying bin population by the square of the difference between the bin position and the barycente,
   i) The ninth column has a formula multiplying bin population by the square of the difference between the bin position and the barycenter,
   j) The tenth column is left empty.
5. We write two “footer” rows with formulae calculating the skewness and skewness statistical errors.

3.4. Width of the jitter distribution as a function of the period

During the experiment, when we apply several frequencies and measure the distributions of angle, we are varying $N = (\alpha f_0)/2S$, which represents "half the effective number of threshold-crossing attempts per tau". One simple and effective way to compare the time jitter with predictions of models is to plot the root mean square width $\phi_{\rm rms} = \sqrt{<(\phi - <\phi>)^2>}$ of the measured angles as a function of "N". If the frequency f is way above the noise bandwidth, we should observe the naïve expectation of a constant $\phi_{\rm rms}$, and models will predict shapes for $\phi_{\rm rms}$, as a function of N. The Figure 5 shows a plot of experimentally measured $\phi_{\rm rms}$, as a function of the period in μs. The abscissae (the period in μs) is a variable proportional to N — but the exact horizontal scale factor is a fit parameter, which depends on both noise α and noise bandwidth f_0. The ordinate contains the other fit parameter, a vertical scale, which is simply proportional to the size α of the noise.

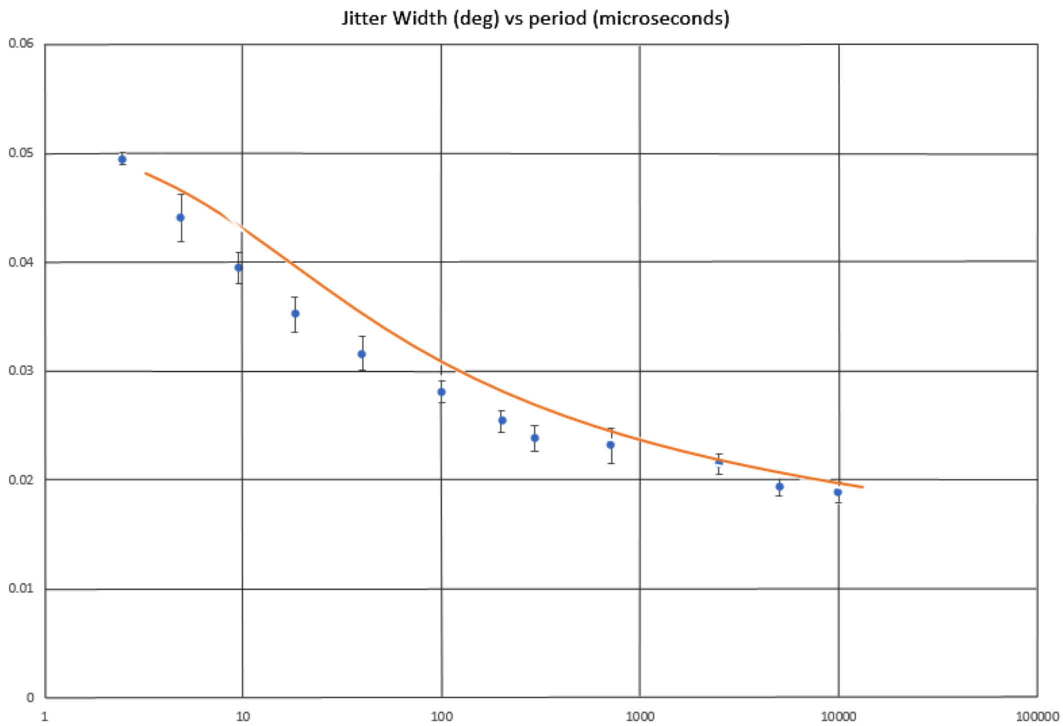

Figure 5. Angle Widths

The error bars represent just the statistical error on the measurement of the root-mean-square jitter. This example data illustrates several points. First, we see data cannot be fit by a constant, as the naïve model would mandate. Then, we can run a 2-parameter fit with the Gaussian noise assumption: this is represented by the orange solid line. One parameter is the width of the Gaussian, its value being 0.057°. The corresponding divisor in the erf argument of the cumulative distribution function is 0.08°, not too far off the result of the fit of the direct reconstruction algorithm result. The other parameter is the number of random number generations per tau, which reaches ~2000 at the lowest frequency, at a period T of 10000 μs, which means $f_0 = N_{\rm eff}/T\,(360°/0.057°) \cong 1.2$ GHz. Note that for the lowest frequency (last point) the Lambert-function based asymptotic formula (5) then predicts 0.0183°, which is approximately this data point.

In the above plot, the statistical errors are shown, but there are also systematic errors due to temperature variations of the Device Under Test.

If we wanted to eliminate sources of errors, we would acquire trigger jitter data immediately after frequent frequency changes, because this process would average-out the effect of temperature drifts and other possible environmental influences.

3.5. Skewness of the jitter, skewness of time distribution, as a function of the N parameter

Since time is proportional to angle in each of the measurements, the skewness of the time distribution can directly be measured from the skewness of the angles.

$$\phi_{\text{skewness}} =< (\phi - < \phi >)^3 > / \phi_{\text{rms}}^3$$

In the figure below, the skewness for the observed distribution of angles for our device under test is plotted as a function of the period T. The first observation is that the distributions are skewed and are more skewed as the period is increased. This is indeed a prediction of the models of N_{eff} independent random attempts at crossing the threshold.

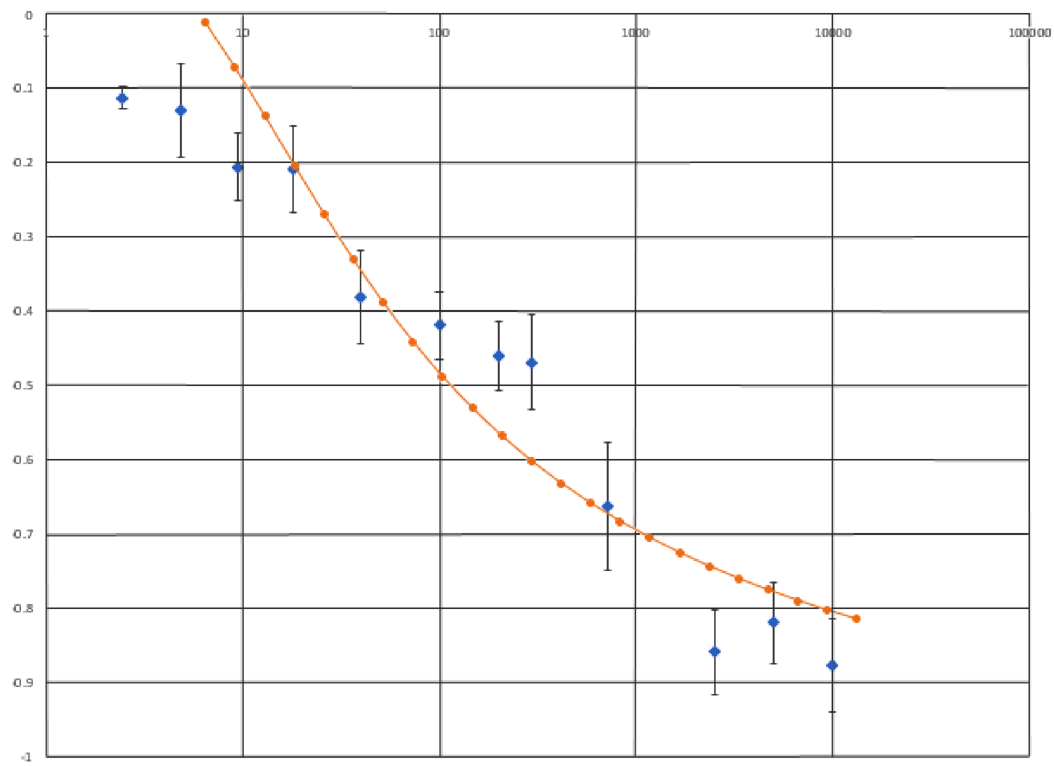

Figure 6. Skewness

The figure 6 also shows the skewness expected from the Gaussian model (solid orange line). In the limit of extremely long periods, the skewness of our measured trigger time histograms follows that of our Gaussian-hypothesis PDF, which will converge asymptotically to the same skewness as the extreme Gompertz distribution, namely $\frac{-12\zeta(3)\sqrt{6}}{\pi^3} \cong -1.14$ [4]. This result follows from the nth moments of the PDF corresponding to an $\exp(-\exp(x))$-shaped CDF being (up to signs) the Riemann zeta functions of n.

4. A DIDACTIC MATLAB SIMULATION GENERATING AN ANIMATION

With respect to a situation where the bandwidth of the noise can be neglected, the introduction of noise bandwidth causes two very noticeable and somewhat non-intuitive effects, namely a narrowing of the jitter distribution, and a more and more asymmetry of the same distribution.

The transition from and to this regime is described by the differential equation (3), but that differential equation is difficult to solve analytically.

One possibility would be to solve this differential equation precisely up to numeric errors of integration techniques. However, that would fail to capture the real underlying randomness.

So, we have developed a small MATLAB program that actually simulates a noise, which remains constant and then changes, at a chosen “noise bandwidth”, as the input voltage changes. This program actually histograms many instances of this random process, in red.

In parallel, the program also simulates the “second term only” random process, and histograms many instances of that random process, this is shown in blue, the green line showing the analytic average.

The animation clearly illustrates how the two descriptions becomes equivalent for small enough slopes (i.e. with enough tries at threshold crossing) even though they are quite different from one another initially for large slopes, as the first term dominates. A magenta oblique line intersecting a blue-and-cyan multiply-valued curve by symbolizes the signal trying to exceed a finite-bandwidth random threshold.

For a didactic point of view, a student can be taught to write the simulation without knowing that it will result in a histogram with negative skewness and that this histogram will also become narrower than the naïve expectation. They can discover these facts by running the simulation and watching the animation.

The listing of the simulation is in appendix C. The frequency of random numbers (noise bandwidth) is reduced by a factor d by copying one random numbers $d-1$ times. A factor called “lam_const” can optionally be non-zero. If enabled, this feature permits to partly reconcile the correct full simulation of equation (3) with the exact solution to equation (4) at the expense of an arbitrary number of “extra chances to attempt crossing the threshold”. The student can realize that this solution is not exact, because it can either reconcile the second moment, or the skewness, but not with the same “lam_const”.

5. OTHER TOOLS

An excel spreadsheet numerically integrates equation (4) for various values of N, and this tool is ready to turn on Student T distributions, if the data exhibit tails that significantly exceed the tail of the Normal distribution [5]. Up to 25 values of N are simulated simultaneously, and the resulting widths and skewness are calculated. This is appended as an attachment to this article in binary format. Columns B-Z hold the erf calculations, scaled by N_{eff}. Columns AB-AZ hold the integral of the previous. Coulmns BB-BZ take the exponential. Columns CB-CZ calculate the PDF. In columns DB-DZ, the PDF is normalized. Cells EB1-EZ1 are where we control the 25 N_{eff} values. Columns FB-FZ hold the sums for the normalization. Columns GB-GZ hold the PDF-abscissa products. Columns HB-HZ calculate the first moment by summing these products. Columns IB-IZ calculate PDF by square of distance to mean products. Columns JB-JZ calculate square root of the second moments from the previous. Columns KB-KL then calculate PDF by cube of distance to mean products, and finally columns LB-LZ computes the skewness from the above by summing them.

6. CONCLUSION

We started by assuming a proportionality between noise and jitter, via the local slope of a linear signal, and while we found that in certain limit cases this proportionality exists, in other cases, like in the presence of hysteresis and considering the noise bandwidth, we have found that the jitter tends to be smaller than could be expected from the noise. Perhaps even more surprisingly, we found a case where hysteresis is not needed to make a different distribution arise.

The physical origin of the phenomenon is that when noise bandwidth is large compared to the time scale of the signal variation, triggering happens with the extreme tails of the noise distribution. Owing to the central limit theorem, the noise will be approximately Gaussian, and the integral of the noise inside the tails will be given by the error function. In the tails, that function is steeper in relative terms than near the inflexion point, and that makes the onset of probability of transition more abrupt. When this probability reaches a maximum, there is then a rapid decay in transition probability because the transition has already occurred: this causes a definite skewness in the probability density function.

While we are able to fit the width and skewness with models, and we have some success with a strictly Gaussian model of noise, in general a more powerful method is to reconstruct the noise distribution by merging jitter data acquired with several value of the slope, spread across various orders of magnitude. One version of this method was shown, which is somewhat statistically inefficient, but very simple to implement.

Contrasting the theory and methods presented in this article with other methods of evaluating discriminators [10], we note that the concept of influence of tails of the noise distribution on threshold crossing has been amply studied, simulated, and compared to experiments. However, the net effect that these tails have on trigger jitter, and how it creates a skewness of the distribution of threshold crossing times even if the noise is strictly symmetric has not yet been made familiar to a wide audience. A particularly interesting feature is that it enables estimates of the noise bandwidth using equipment that is of much lower frequency. Maybe an inexpensive instrument can be built to display noise bandwidth as a teaching tool and convenient industrial debug tool. In any case, it is hoped that this article will help researchers be more familiar with the underlying mathematical causes of these phenomena.

ALGORITHM OF THE HISTOGRAM HARMONIZATION

Appendix A: skewness parameter for Teledyne LeCroy scopes

```
Function Update() 'VBS code
    ' this Measure function calculates skewness of a histogram
    ' (experimentaldistribution)
    numSamples = InResult.Samples ' this is the number of bins of the histogram
    ' start with histogram population and first moment
    sum = 0
    sumi = 0
    scaledData = InResult.DataArray
    For i = 0 To numSamples - 1
        sum = sum + scaledData(i)
        sumi = sumi + scaledData(i)*i
    Next
    ' the average of the distribution is the first moment sumi/sum
    ' use this to center the next moments on the center of gravity
    ' of the distribution
```

```
        sumi2 = 0
        sumi3 = 0
        scaledData = InResult.DataArray
        For i = 0 To numSamples - 1
            sumi2 = sumi2 + scaledData(i)*(i-sumi/sum)*(i-sumi/sum)
            sumi3 = sumi3 + scaledData(i)*(i-sumi/sum)*(i-sumi/sum)*(i-sumi/sum)
        Next
        ' Calculate the skewness as 3rd moment divided 2nd moment to the power 3/2
        OutResult.Value = sumi3/sum / (sumi2/sum)^1.5
        ' Set measurement resolution to 1m, and units to unitless
        OutResult.VerticalResolution = 0.001
        OutResult.VerticalUnits = ""
    End Function
```

Appendix B: Flowchart of the histogram harmonization

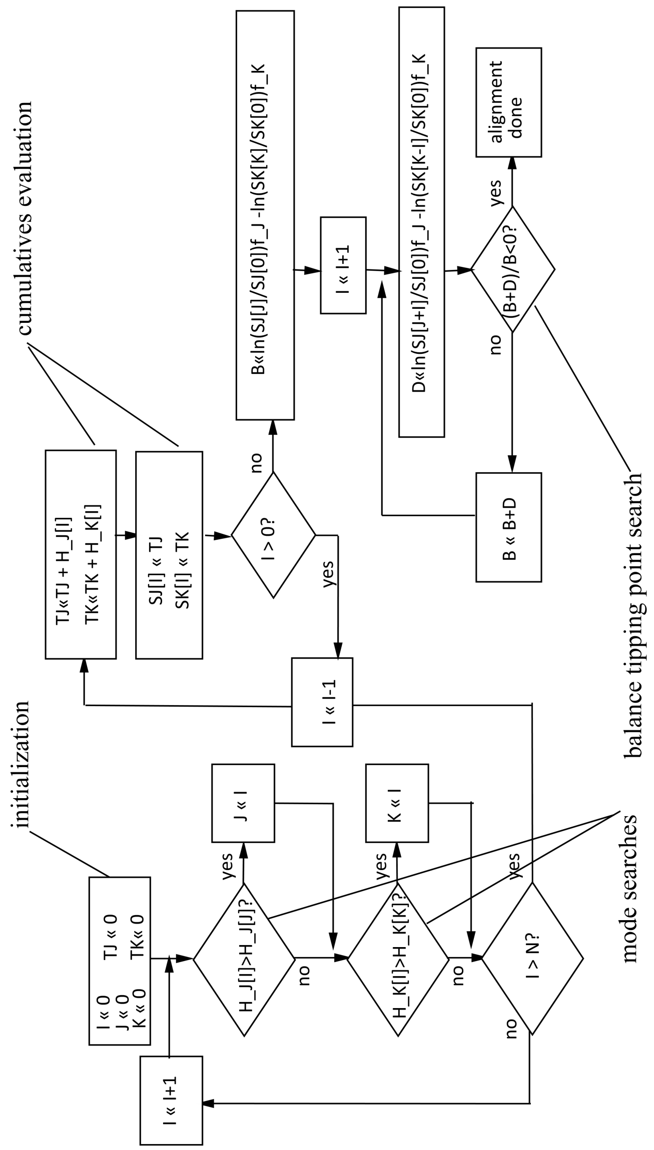

Figure 7. Flowchart of the histogram harmonization algorithm

Appendix C: MATLAB simulation program, producing animation

```
lam_const=2.8/2;
wobj=VideoWriter(sprintf('myv%.1f.avi',lam_const));
wobj.FrameRate = 10;
open(wobj);
N=5040;
nu=[1:N];
divs=nu(fix(N./nu)==N./nu);
granu=0.005;
resu=zeros(300,7);
resu_row=0;
for idiv=[prod(size(divs))-3:-1:1]
    if (idiv==1)
        nfr=20;
    else
        nfr=5;
    end
    for frp=[1:nfr]
        d=divs(idiv);
        simpr=rand(N);
        exper=sqrt(-2*log(rand(N))).*cos(2*pi*rand(N));
        for j=[0:d-1]
            for k=[0:d-1]
                if (j>k)
                    exper(1+[j:d:N-d+j],1+[k:d:N-d+k])=exper(1+[k:d:N-d+k],
                    1+[k:d:N-d+k]);
                end
                if (j<k)
                    exper(1+mod([j+d:d:N+j],N),1+[k:d:N-d+k])=exper(1+[k:d:N-d+k],
                    1+[k:d:N-d+k]);
                end
            end
        end
        level=cumsum(0*exper+granu)-8;
        cdft=exp( (-lam_const*sqrt(2)-sqrt(0.5)/d/granu)
        *(level/sqrt(2).*erfc(-level/sqrt(2))+exp(-level.*level/2)/sqrt(pi)));
        cdfn=cumprod(double(simpr>erfc(-level/sqrt(2))*(lam_const*granu+0.5/d)));
        cdfu=cumprod(double(exper>level));
        xvalues=diff(cumsum(mean(level')));
        plor=-diff(sum(cdfu'));
        plot(xvalues,plor,'r');
        moyn=sum(xvalues.*plor)/sum(plor);
        stdv=sqrt(sum((xvalues-moyn).*(xvalues-moyn).*plor)/sum(plor));
        skwn=sum((xvalues-moyn).*(xvalues-moyn).*(xvalues-moyn).*
        plor)/sum(plor)/stdv^3;
        xlim([-8 14]);
        ylim([0 50]);
        hold on;
        plot(xvalues,-diff(sum(cdfn')),'b');
        plog=-diff(sum(cdft'));
        plot(xvalues,plog,'g');
        aver=sum(xvalues.*plog)/sum(plog);
        stnd=sqrt(sum((xvalues-aver).*(xvalues-aver).*plog)/sum(plog));
```

```
            skuw=sum((xvalues-aver).*(xvalues-aver).*(xvalues-aver).
            *plog)/sum(plog)/stnd^3;
            whe=1+floor(rand(1)*d);
            for irp=[whe:d:N-1]
                plot(xvalues([irp,irp]),[41.8,43.2],'-b')
                plot(xvalues([irp,irp]),[40.5,41.8],'-c')
                plot(xvalues([irp,irp]),[43.2,44.5],'-c')
            end
            plot(xvalues([irp,irp]),[48.5,49.5],':k')
            plot([-2.5 2.5],[40,45],'-m')
            plot([-7.5 13.5],[41.8,41.8],'-b')
            plot([-7.5 13.5],[43.2,43.2],'-b')
            plot([-7.5 13.5],[40.5,40.5],'-c')
            plot([-7.5 13.5],[44.5,44.5],'-c')
            hold off
            text(5,38,sprintf('slope=%d (Ntry=%.1f)',d,200/d));
            writeVideo(wobj,getframe(gcf));
            drawnow;
            resu_row=resu_row+1;
            resu(resu_row,1)=d;
            resu(resu_row,2)=moyn;
            resu(resu_row,3)=stdv;
            resu(resu_row,4)=skwn;
            resu(resu_row,5)=aver;
            resu(resu_row,6)=stnd;
            resu(resu_row,7)=skuw;
        end
    end
    close(wobj);
```

Appendix D: listing of script for histogram-joining into excel-ready sheet

```
set fso=createobject("scripting.filesystemobject")
mypth=fso.getparentfoldername(wscript.scriptfullname)
N=wscript.arguments.count-1
redim fr(N)
redim frq(N)
redim lins(N)
redim xval(99)
redim nulin(99)
for i=0 to N
    fr(i)=mid(wscript.arguments(i),8+instrrev(wscript.arguments(i),"trigtdis"))
    frq(i)=left(fr(i),instr(fr(i),"Hz")-1) if (right(frq(i),1)="k")
    then frq(i)=1000*left(frq(i),len(frq(i))-1)
    frq(i)=frq(i)*1
    set f=fso.opentextfile(wscript.arguments(i))
    lins(i)=split(f.readall,vbcrlf)
    f.close
next
for k=0 to N
    for j=k+1 to N
        if (frq(j)<frq(k)) then
            ff=frq(k)
            frq(k)=frq(j)
```

```
                frq(j)=ff
                lines=lins(k)
                lins(k)=lins(j)
                lins(j)=lines
            end if
        next
    next
    ' plan of 10 columns        A:bin, B:pop, C:prod, D:cumsum, E:error
    '                           F:scaledlog G:sum(Prod) H:2nd, I:3rd, J: moments
    '
    '
    '                   header
    '
    '
    set fo=fso.createtextfile(mypth&"\MergedHistos.csv")
        for k=0 to N
        letb=chr(65+((10*k+1) mod 26))
        if 10*k+1>=26 then letb=chr((10*k+1)\26+64)&letb
        letc=chr(65+((10*k+2) mod 26))
        if 10*k+2>=26 then letc=chr((10*k+2)\26+64)&letc
        leth=chr(65+((10*k+7) mod 26))
        if 10*k+7>=26 then leth=chr((10*k+7)\26+64)&leth
        fo.write( frq(k) & "," & "Hz" & ",,,,,=SUM(" & letc & ":" & letc & ")/
sum(" & letb & ":" & letb & "),,,=SQRT(SUM(" & leth & ":" & leth & ")/
SUM(" & letb & ":" & letb & "))/0.9,")
    next
    fo.writeline()
    for k=0 to N
        letb=chr(65+((10*k+1) mod 26))
        if 10*k+1>=26 then letb=chr((10*k+1)\26+64)&letb
        letj=chr(65+((10*k+9) mod 26))
        if 10*k+9>=26 then letj=chr((10*k+9)\26+64)&letj
        fo.write( ",,,,,,,,,=" & letj & "1/SQRT(SUM(" & letb & ":" & letb & ")),")
    next
    fo.writeline()
    for k=0 to N
        leta=chr(65+((10*k  ) mod 26))
        if 10*k  >=26 then leta=chr((10*k  )\26+64)&leta
        fo.write( ",,,,,,,,,=1000000/" & leta & "1,")
    next
    fo.writeline()
    for k=0 to N
        letj=chr(65+((10*k+9) mod 26))
        if 10*k+9>=26 then letj=chr((10*k+9)\26+64)&letj
        fo.write( ",,,,,,,,,=" & letj & "1,")
    next
    fo.writeline()
    for k=0 to N
        letj=chr(65+((10*k+9) mod 26))
        if 10*k+9>=26 then letj=chr((10*k+9)\26+64)&letj
        fo.write( ",,,,,,,,,=" & letj & "2*5,")
    next fo.writeline()
    '
```

```
'
'
'                          end of header
'
'
for k=0 to N-1
    lines0=lins(k)
    lines1=lins(k+1)
    max0=0
    max1=0
    for i=5 to 104
        to0=split(lines0(i),",")
        lines0(i-5)=to0(1)*1
        if (k=0) then
            xval(i-5)=to0(0)*1
        end if
        if (lines0(i-5)>=lines0(max0)) then max0=i-5
        to1=split(lines1(i),",")
        lines1(i-5)=to1(1)*1
        if (lines1(i-5)>=lines1(max1)) then max1=i-5
    next
    scadiflog0=lines0
    scadiflog1=lines1
    for i=98 to 0 step -1
        scadiflog0(i)=scadiflog0(i+1)+lines0(i)
        scadiflog1(i)=scadiflog1(i+1)+lines1(i)
    next
    integ0=scadiflog0
    integ1=scadiflog1
    stats0=scadiflog0(0)
    stats1=scadiflog1(0)
    for i=99 to 0 step -1
        scadiflog0(i)=-log(scadiflog0(i)/scadiflog0(0)+1e-99)*frq(k)
        scadiflog1(i)=-log(scadiflog1(i)/scadiflog1(0)+1e-99)*frq(k+1)
    next
    'if (frq0<frq1) then
    sign=1
    'else
    'sign=-1
    'end if
    'if (1) then
    tot0=scadiflog0(max0)
    tot1=scadiflog1(max1)
    bal=tot1-tot0
    for mov=1 to 20
        balold=bal
        tot0=tot0+scadiflog0(max0+sign*mov)
        tot1=tot1+scadiflog1(max1-sign*mov)
        bal=tot1-tot0
        if (bal*balold<=0) then
            frac=1-abs(bal)/abs(bal-balold)
            exit for
        end if
```

```
        next
        'end if
        chisq=0
        mov=mov-1+frac
        for i=max0 to max0+sign*mov step sign
            j=i+max1-max0-sign*mov
            relerr=(scadiflog0(i)-scadiflog1(j))/(scadiflog0(i)+scadiflog1(j))
            chisq=chisq+relerr^2*lines0(i)*lines1(j)*integ0(i)*integ1(j)/ _
            (lines0(i)*integ0(i)*integ1(j)+lines1(j)*integ0(i)*integ1(j)+lines0(i)
            *lines1(j)*integ0(i)+lines0(i)*lines1(j)*integ1(j))
        next
        msgbox "peak " & max0 & " in " & frq0 & "(" & stats0 & ")" & vbcrlf &
        "peak " & max1 & " in " & frq1 & "(" & stats1 & "}" & vbcrlf & mov & "--->"
        & mov+(max0-max1)*sign & vbcrlf & "chi^2:" & chisq
        for i=0 to 99
            nulin(i)=nulin(i)&xval(i)&","
            if (i<2) then
                xval(i)=xval(i)-(max1-max0-sign*mov)*(xval(99)-xval(98))
            else
                xval(i)=xval(i)-(max1-max0-sign*mov)*(xval(1)-xval(0))
            end if
    nulin(i)=nulin(i)&lines0(i)&","
    leta=chr(65+((10*k) mod 26))
    if 10*k>=26 then leta=chr((10*k)\26+64)&leta
    letb=chr(65+((10*k+1) mod 26))
    if 10*k+1>=26 then letb=chr((10*k+1)\26+64)&letb
    letd=chr(65+((10*k+3) mod 26))
    if 10*k+3>=26 then letd=chr((10*k+3)\26+64)&letd
    lete=chr(65+((10*k+4) mod 26))
    if 10*k+4>=26 then lete=chr((10*k+4)\26+64)&lete
    letg=chr(65+((10*k+6) mod 26))
    if 10*k+6>=26 then letg=chr((10*k+6)\26+64)&letg
    nulin(i)=nulin(i) & "=" & leta & i+6 & "*" & letb & i+6 & ","
    'column C
    nulin(i)=nulin(i) & "=" & letd & i+7 & "+" & letb & i+6 & ","
    'column D
    nulin(i)=nulin(i) & """=" & leta & "$1*(LN(" & letd & i+6 & "+1E-99)-LN(" & letd
    & i+7 & "+1E-99))*IF(SQRT(1/(" & letd & i+6 & "+0.000000001)+1/(" & letd & i+6
    & "-" & letd & i+7 & "+0.000000001))>0.080123,0,SQRT(1/(" & letd & i+6 &
    "+0.000000001)+1/(" & letd & i+6 & "-" & letd & i+7 & "+0.000000001)))*4.0123"","
    nulin(i)=nulin(i) & """=IF(" & lete & i+6 & "=0,0,(LN(" & letd & i+6 & "+1E-99)-
LN(" & letd & i+7 & "+1E-99))*" & leta &"$1)"",,"
    nulin(i)=nulin(i) & "=" & letb & i+6 & "*(" & leta & i+6 & "-" & letg & "$1)*
(" & leta & i+6 & "-" & letg & "$1),"
    nulin(i)=nulin(i) & "=" & letb & i+6 & "*(" & leta & i+6 & "-" & letg & "$1)*
(" & leta & i+6 & "-" & letg & "$1)*(" & leta & i+6 & "-" & letg & "$1),,"  next
    next
    for i=0 to 99
    nulin(i)=nulin(i)&xval(i)&","
    nulin(i)=nulin(i)&lines1(i)&","
    leta=chr(65+((10*k) mod 26))
    if 10*k>=26 then leta=chr((10*k)\26+64)&leta
    letb=chr(65+((10*k+1) mod 26))
```

```
if 10*k+1>=26 then letb=chr((10*k+1)\26+64)&letb
letd=chr(65+((10*k+3) mod 26))
if 10*k+3>=26 then letd=chr((10*k+3)\26+64)&letd
lete=chr(65+((10*k+4) mod 26))
if 10*k+4>=26 then lete=chr((10*k+4)\26+64)&lete
letg=chr(65+((10*k+6) mod 26))
if 10*k+6>=26 then letg=chr((10*k+6)\26+64)&letg
nulin(i)=nulin(i) & "=" & leta & i+6 & "*" & letb & i+6 & ","
'column C
nulin(i)=nulin(i) & "=" & letd & i+7 & "+" & letb & i+6 & ","
'column D
nulin(i)=nulin(i) & """=" & leta & "$1*(LN(" & letd & i+6 & "+1E−99)−LN(" & letd
 & i+7 & "+1E−99))*IF(SQRT(1/(" & letd & i+6 & "+0.000000001)+1/(" & letd
 & i+6 & "−" & letd & i+7 & "+0.000000001))>0.080123,0,SQRT(1/(" & letd & i+6 & "
+0.000000001)+1/(" & letd & i+6 & "−" & letd & i+7 & "+0.000000001)))*4.0123""",
nulin(i)=nulin(i) & """=IF(" & lete & i+6 & "=0,0,(LN(" & letd & i+6 & "+1E−99)−
LN(" & letd & i+7 & "+1E−99))*" & leta &"$1)"",,"
nulin(i)=nulin(i) & "=" & letb & i+6 & "*(" & leta & i+6 & "−" & letg & "$1)*
(" & leta & i+6 & "−" & letg & "$1),"
nulin(i)=nulin(i) & "=" & letb & i+6 & "*(" & leta & i+6 & "−" & letg & "$1)*
(" & leta & i+6 & "−" & letg & "$1)*(" & leta & i+6 & "−" & letg & "$1),,"
next
fo.write(join(nulin,vbcrlf))
fo.writeline()
for k=0 to N
letb=chr(65+((10*k+1) mod 26))
if 10*k+1>=26 then letb=chr((10*k+1)\26+64)&letb
leti=chr(65+((10*k+8) mod 26))
if 10*k+8>=26 then leti=chr((10*k+8)\26+64)&leti
letj=chr(65+((10*k+9) mod 26))
if 10*k+9>=26 then letj=chr((10*k+9)\26+64)&letj
fo.write( ",,,,,,,,,=SUM(" & leti & ":" & leti & ")/SUM(" & letb & ":" & letb &
")/" & letj & "1^3/0.9^3,")
next
fo.writeline()
for k=0 to N
letb=chr(65+((10*k+1) mod 26))
if 10*k+1>=26 then letb=chr((10*k+1)\26+64)&letb
fo.write( ",,,,,,,,,=SQRT(40/SUM(" & letb & ":" & letb & ")),")
next
fo.close
```

References

1. M. Miller, *White paper*, Presented at DesignCon 2014, Santa Clara, CA, USA, 2014.
2. J. Res. Natl. Inst. Stand. Technol. 110, pp. 511–527, 2005.
3. E. W. Weisstein, "Uniform Distribution," *MathWorld–A Wolfram Web*. [Online]. Available: http://mathworld.wolfram.com/UniformDistribution.html
4. A. Lenart, "The moments of the Gompertz distribution and maximum likelihood estimation of its parameters," *Scandinavian Actuarial Journal*, no. 3, pp. 255–277, 2014; doi: 10.1080/03461238.2012.687697
5. N. M. Temme, "The asymptotic expansion of the incomplete Gamma functions," *SIAM Journal on Mathematical Analysis*, vol. 10, no. 4, pp. 757–766, 1979.

6. D. L. Decker, “Computer evaluation of the complementary error function,” *American Journal of Physics*, vol. 43, issue 9, pp. 833–834, 1975; doi: 10.1119/1.9699
7. R. Kiely, “Understanding and Eliminating 1/f Noise,” *Analog Dialogue*. [Online]. Available: http://www.analog.com/en/analog-dialogue/articles/understanding-and-eliminating-1-f-noise.html
8. J. F. Stephany, “A theory of 1/f noise,” *Journal of Applied Physics*, vol. 83, no. 6, pp. 3139–3143, 1998; doi: 10.1063/1.367071
9. J. H. Lambert, “Observationes variae in mathesin puram,” *Acta Helvetica, physico-mathematico-anatomico-botanico-medica*, vol. 3, pp. 128–168, 1758.
10. J. Kim, B. S. Leibowitz, J. Ren, and C. Madden, “Simulation and Analysis of Random decision Errors in Clocked Comparators,” *IEEE transactions on Circuits and Systems – I: Regular Papers*, vol. 56, no. 8, pp. 1844–1857, 2009; doi: 10.1109/TCSI.2009.2028449

Received 21.08.2019, the final version — 24.09.2019.

Francois Lamarche, Dr, (PhD Physics, McGill University, 1989) Metrologist at Teledyne LeCroy Inc, Chestnut Ridge, NY, USA, ✉ francois.lamarche@teledyne.com