The Jitter-Noise Duality and Anatomy of an Eye Diagram

Vladimir Dmitriev-Zdorov, Mentor Graphics
vladimir_dmitriev-zdorov@mentor.com

Martin T. Miller, Teledyne LeCroy
marty.miller@teledynelecroy.com

Chuck Ferry, Mentor Graphics
chuck_ferry@mentor.com
Abstract

We compare two approaches where the structure of the eye diagram is considered either as a timing uncertainty or alternatively a vertical uncertainty. We show that only the second view allows accurate BER evaluation when the eye is seriously impaired. It follows from BER definition that its proper evaluation should be delay lock loop based on considering the two portions of the eye diagram, and by their separate integration. With such understanding of BER, it turns out that the jitter-induced vertical noise can be considered same way as the signal’s ISI, although correlated with the latter. We propose a method of statistical simulation based on this view, where the effect of Tx jitter and ISI is considered together. We use the probability mass functions (PMF) to represent both ISI and jitter contributions, which makes the algorithm simple and physically clear. Some interesting effects can be explained based on the proposed approach, for example jitter increase with growing PRBS order of the input pattern.

Patent Disclosure

Portions of this document are the subject of patents applied for.

Biography

Dr. Vladimir Dmitriev-Zdorov is a principal engineer at Mentor Graphics Corporation. He developed a number of advanced models and novel simulation methods used in the company’s products. Current work includes development of the efficient methods of circuit/system simulation in time and frequency domain, transformation and analysis of the multi-port systems, statistical & time domain analysis of SERDES links. He received Ph.D. and D.Sc. degrees (1986, 1998) based on his work on circuit and system simulation methods. The results have been published in numerous papers, conference proceedings, including several in DesignCon, and a monograph.

Dr. Martin Miller is chief scientist at Teledyne-LeCroy. Marty received a doctorate in particle physics from the University of Rochester (1981). He has over 35 years experience at Teledyne-LeCroy in various R&D functions, including analog, digital and software design. Marty has spent the past 24 years focusing on algorithms and methods for display and measurements in digital oscilloscopes, most significantly in the area of signal integrity (jitter and timing). Presently he has more than 15 U.S. patents in this domain. He has participated in IEEE committees and JEDEC, contributing to test methods for digitizers and memories. He has actively participated in DesignCon since 2007 as both an “expert” panel member (tp_M1), as a paper presenter, tutorial presenter and is on the program committee for DesignCon.

Chuck Ferry is product marketing manager at Mentor Graphics Corporation. Chuck focuses on product definition and validation for signal integrity and power integrity solutions. He has spent the last 15 years tackling a broad range of high speed digital design challenges spanning from system level design to multi-gigabit channel analysis developing and incorporating detailed characterizations of the IC, packages, connectors and multiple boards. He has delivered signal integrity services and seminars using a wide range of EDA tools as well as performed modeling services creating and validating IC models. Chuck graduated Magna cum Laude from the University of Alabama with a BS in electrical engineering and continued graduate course work in the areas of signal processing and hardware description languages.
Introduction

Is “jitter” purely timing uncertainty or is noise the consequence of jitter? Which view is more productive for an accurate estimation of bit error ratio (BER)? In the field of radio frequency (RF) electronics the phenomenon of Amplitude to Phase conversion and Phase to Amplitude conversion are well known. This body of knowledge concerns mostly tones and clocks, but there are lessons to be learned for how these concepts translate to serial data signals. The information in an eye-diagram is composed of both jitter and noise, and often we speak of “open” eyes and “closed” eyes, but too often we consider only one axis of closure. As the technology of serial data transmission has developed there has been alternating emphasis on one or the other of the jitter problem or the noise problem. In the early days of optical links almost all the emphasis was on vertical noise then more recently, jitter became the focus as timing margins became the harder part of the problem. Today it would not be unfair to say we have pushed the envelope to the point that both jitter and noise have roughly equal importance. For Serial Data links, jitter manifests as both a non-uniformity in edge timing and as “random” variations in edge timing. Often intersymbol interference (ISI) is considered uniquely in terms of the systematic “jitter” resulting from a lossy transmission medium. However, the nature of ISI is to shape the trajectory of the transmitted signal and its effect on jitter is secondary. The primary cause of the transmit jitter is non-ideal transition times. ISI, noiselaboratory, crosstalk and other impairments on the receiver side, make the threshold level crossing times spread along the horizontal axis. Hence, we can use the probability density function (PDF) of these crossing times, to estimate BER through integration. Although the “jitter” view is more common, the “voltage” can be preferred in some jitter analysis techniques. We’ll demonstrate that proper BER evaluation requires vertical integration of the eye density or histogram, thus making the “vertical thinking” more productive. Furthermore, we’ll describe a simple but accurate statistical simulation algorithm where transmit jitter, originated at different times, is converted into vertical noise which at observation point is summed up so that the resulting eye density and BER plots acquire its true shape, both in vertical and horizontal dimension. We’ll describe a physically clear and accurate approach for statistical channel analysis in presence of Tx jitter with no simplifying assumptions on edge response linearization or statistical independence between ISI and jitter components.

The paper is organized as follows. First, we discuss a new way of finding BER, as a function of the sample time and voltage. Then, we introduce jitter at receive and transmit end, and explain how it affects the trajectories of the eye diagram. Then, we analyze the effect of constraining bit combinations, and the effect it has on the Eye Diagram and BER. Then, we consider the concept of probability mass function, and formulate the algorithm of statistical analysis in presence of Tx jitter, both random and deterministic. Finally, we compare our statistical analysis and measurements, for the purpose of correlation. We used measurement and extracted data from a laboratory vector network analyzer, a laboratory pattern generator and an oscilloscope.

How to find the BER from the Eye Diagram

The BER is a ratio of the total number of bit errors to the total number of bits correctly transferred. BER is metric for the probability of a signal to be distorted enough by the time it reaches the receiver that it is misinterpreted as the incorrect logical value. This very basic definition will help guide us in determining which method of BER calculation should be considered as the golden standard.
Figure 1: Waveform and Eye-diagram Example

(a) example waveform piece
(b) eye diagram built from example waveform piece

Figure 2: BER Plots

(a) positive eye density integrated upwards
(b) negative eye density integrated downwards
(c) positive branch of BER
(d) negative branch of BER
(e) combined eye
(f) combined BER
As mentioned earlier there are two distinctly different methods that can be used to evaluate the BER of a link. Both depend on an integral sum of the area below a PDF. One method depends on integrating along the horizontal/time axis and the other that we will discuss in detail depends on integrating along the vertical/voltage axis. We will compare the results from both of these methods.

Let’s first examine a method in which we use the vertical eye density distribution for BER evaluation. For convenience, consider a very short waveform - only a few bits long - as illustrated in Figure 1(a). The vertical lines indicate selected sampling times, which become horizontal centers of the 2UI long trajectories in the eye diagram. The blue and red brackets at the bottom show how the waveform is partitioned into trajectories; with color indicating the sign of the transmitted bit being sampled: red – positive, blue - negative. The trajectories in 1(b) are colored accordingly. These trajectory’s can be determined within the simulation domain fairly easily since the stimulus pattern is well known as well as the delay through the system channel. Likewise, there are methods within the measurement domain to determine these trajectory’s.

With colored trajectories in 1(b), we can evaluate BER. Select any point \( P_k \), with arbitrary coordinates, indicating timing and voltage offset of the sampling event. Since color indicates the sign of the transmitted bit, the probability of error can be estimated by counting the number of ‘red’ traces below \( P_k \) (‘1’ read as ‘0’) and the number of blue traces above \( P_k \) (‘0’ read as ‘1’). For the points selected in 1(b), such evaluation gives:

<table>
<thead>
<tr>
<th>Sample point</th>
<th>BER est.</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0</td>
<td>All bits correctly sampled</td>
</tr>
<tr>
<td>P2</td>
<td>1/7</td>
<td>2 of total 14 bits sampled incorrectly</td>
</tr>
<tr>
<td>P3</td>
<td>1/2</td>
<td>7 of 14 incorrectly sampled</td>
</tr>
<tr>
<td>P4</td>
<td>2/7</td>
<td>4 of 14 incorrectly sampled</td>
</tr>
<tr>
<td>P5</td>
<td>4/7</td>
<td>8 of 14 incorrect</td>
</tr>
</tbody>
</table>

Evidently, we can place as many sample points as needed and find the BER for all possible (x,y) coordinates, thus making it 2D BER plot. Finding BER is straightforward, regardless of the fact that the “eye” is open or closed. Similarly, we can build BER from eye diagram in more general case, when individual trajectories cannot be ‘counted’. However, instead of using colors, we should consider “density” of trajectories, separately accumulated for positive and negative bits being sampled. In most cases such selective accumulation is possible, because measurement device or a simulator know the type of the bit pattern it receives. After the two separate eye-density plots are accumulated, they are integrated to find the cumulative distribution, either upward or downward. Figure 2 illustrates this approach, where Figure 2(a) through 2(d) show the two branches of the eye density are separately integrated and produce the two summands of the BER. The summation of the branches gives us a conventional eye and BER plots, as in Figure 2(e) and Figure 2(f).

We can prove that the technique considered above is accurate and agrees with BER definition: BER is a probability of incorrect reading of a transmitted logical level. We can estimate BER as a ratio of incorrectly sampled bits to the total number of transmitted bits: \( BER = N_{errors}/N_{total} \). With two logical levels, incorrect reading means either reading ‘1’ as ‘0’ or reading ‘0’ as ‘1’. Denote those numbers as \( N_{10} \) and \( N_{01} \) respectively. The total number of bits is composed from the number of ‘ones’ \( N_1 \) and ‘zeros’ \( N_0 \). Therefore, \( N_{errors} = N_{10} + N_{01}, N_{totals} = N_1 + N_0 \), and:

\[
BER = \frac{N_{10} + N_{01}}{N_1 + N_0} = P_1 P_{10} + P_0 P_{01}
\]  

(1)
Here, $P_1 = N_1 / (N_1 + N_0)$ and $P_0 = N_0 / (N_1 + N_0)$ are estimates of probability of transmitting bit ‘1’ or ‘0’ respectively; $P_{01} = N_{01}/N_1$ and $P_{10} = N_{10}/N_0$ evaluate probability of reading ‘1’ as ‘0’ or ‘0’ as ‘1’. For a DC balanced signal, $P_1 = P_0 = 0.5$ and (1) becomes: $BER = 0.5 \cdot (P_{10} + P_{01})$.

BER is a function of time/voltage or (x,y) coordinates. For a fixed $x$, $P_{10}$ grows with $y$, but $P_{01}$ grows against $y$. By nature, $P_{10}$ and $P_{01}$ are cumulative probability distributions, and should be found by integration of the respecting probability densities, which in our case are normalized positive and negative eye density branches. Since integration should be performed in different directions, it is essential to keep two separate eye density plots. The proposed technique suggests, and Figure 1 prove that with increased simulation length, the eye density plot becomes more periodic in times, whereas BER is basically non-periodic function. Eye density is a view at the receiver end, but BER is a function of the distance between the two events: the signal being sent, and the sampling, with their respected time/voltage level coordinates.

Figure 3: Integrations of the Eye Density in Various Directions
Figure 4: Eye Density Plot

Figure 5: BER Plot
Will both methods always produce the same results?

In all operations we considered so far, the direction of comparisons (above or below), or eye density integration was made in vertical direction. Does jitter or noise make any difference to it? Perhaps not, as long as we are able to correctly build the trajectories or eye density plots considering Tx or Rx jitter. Now, what about horizontal integration of the eye density? Sometimes, it is reasonable as a rough BER evaluation. Typically, horizontal integration is applied to the combined (positive and negative) eye density.

Let’s however consider the case of separate integration, which gives more insight into the problem. In Figure 3, we evaluate BER for a certain point designated by a small white circle, lying at the median level that corresponds to the optimal sampling threshold. The white and blue arrows show the integration paths, horizontal or vertical, leading to the point. If the eye is open, as in Figure 3(a) and Figure 3(b), both horizontal and vertical integration may give the same result, since both integration paths go over the same trajectories. However, an implicit assumption here is that the horizontal path counts each trajectory only once that necessitates the ‘monotonic’ manner of rise or fall transitions within integration region. Evidently, we don’t have to worry about non-monotonic transitions in case of vertical integration. However, even if the raw integral value is correct, it is not sufficient. The integral itself gives us the total number of encountered errors. To get the error rate, this number should be divided onto the total number of transmitted ‘high’ or ‘low’ bits. In case of vertical integration, the total mass of the transmitted high or low bits can be found once by full-range vertical integration of the positive or negative ‘bunch’. Note that the result of this integration does not depend on the x-position: it always includes all the trajectories. In case of horizontal integration, normalization can be done by dividing each integral value on the number of counted positive or negative bits transmitted. It would be improper to divide this value on the sum found by horizontal integration from the eye center by 1UI to the right or left, because such integration will only count the trajectories that change the logic states, but not the ones on the top and bottom of the eye diagram. A possible way for the horizontal integration is to divide the result on the total number of transmitted bits, assuming that positive and negative are equally probable.

In case of the closed or partially closed eye, horizontal integration is greatly affected. As we see from Figure 3(c) and Figure 3(d), the horizontal integration path may miss some trajectories if they lie above or below it, or it can count some trajectories several times. Note, that vertical integration still produces the meaningful correct BER estimation.

Let’s now look at another example in which we measure and simulated a channel then computed the BER plot by both the vertical and horizontal methods. In Figure 4 which is an eye density plot for a channel you will noticed that we highlighted two areas of the eye density plot near the edges of the inside of the eye. There is an apparent “ripple” within the eye density plot.

Let’s now investigate the resulting BER plots that would be computed using the vertical and horizontal approach discussed to this point. In Figure 5 the BER plot in red was computed using the horizontal method. You will see that since the BER is calculated purely by integrating along the horizontal axis is purely monotonic. On the other hand the BER plot in blue was calculated using integration along the vertical axis and separately accumulated for positive and negative bits being sampled. For this example the vertical integration method produces non-monotonous BER curves that at first appear less intuitive but when comparing the BER plot to the eye density plot above it is clear that one can ascertain a deeper insight into the behavior of the channel using this method.
Figure 6: Horizontal jitter is equivalent to the vertical convolution with varied PDF

(a) Transmit Jitter “fuzzy” trajectory
(b) Receive Jitter creates vertical noise proportional to slope of trajectory

Figure 7: Transmit Jitter and Receive Jitter both manifest as vertical “Noise”.

**Tx and Rx jitter**

Let’s explain why vertical eye density distribution is of primary importance for jitter simulation. As shown e.g. in [1, 2, 3], in presence of jitter, the signal at the receive sampling location can be represented as:

\[
y(t + \eta_{RX}) = \sum_{k=-\infty}^{n} (b_k - b_{k-1}) \cdot S(t - kT + \eta_{TX,k})
\]

In (2), \(T\) is a bit interval, \(b_k\) are logical bit values, either +1 or −1, \(S(t)\) is the channel’s step response, and \(\eta_{TX}, \eta_{RX}\) represent phase jitter injected at transmitter and receiver. For a given moment of observation, \(t = nT + \tau\), with \(\tau \in [0, T]\) characterizing the timing position of the sample point within bit interval, (2) can be rewritten as:
Figure 8: Vertical Noise Effects on Jitter. Vertical noise is created by transmit jitter independently in every segment of the step response. On the eye diagram, the combined noise creates a “jitter increase” much larger than the transmit jitter alone.

\[ y_n (\eta_{RX}) = \sum_{k=-\infty}^{n} (b_k - b_{k-1}) \cdot S_{n-k} (\eta_{TX,k}) \]  

In (3) \[ y_n (x) = y (nT + \tau + x), S_{n-k} (x) = S [(n - k) T + \tau + x] \].

The receive jitter \( \eta_{TX,k} \) represents the difference between the intended and the actual sample time, and can be characterized by its single probability distribution. Transmit jitter is different. Since \( y (t) \) depends on many transmitted bits, with their unique jitter values \( \eta_{TX,k} \), the effect of transmit jitter is more complicated.

As we see from (2), in the absence of jitter, \( y (t) \) is expressed through samples of the step response, taken by bit interval space with an offset \( \tau \). There are two equivalent ways of considering jitter applied to a single transition. First is to convolve the step or edge response with the horizontal jitter PDF, as shown in Figure 4: the red curve at the bottom illustrates jitter PDF, the bold blue curve designates the original waveform, and the pair of dashed blue waveforms show the boundaries of the “jittered waveform”. All points of the original curve are being displaced in concert. The second way is to find the equivalent vertical PDFs \( (v_1 (y), v_2 (y), \ldots) \) by transforming horizontal distribution into vertical; vertical PDFs are shown in green. Since curvature of the edge response varies in time, so does the vertical distribution. If we take many of such vertical PDFs, together they will define the same ‘jittered’ body of the edge as the one found by horizontal displacement.

Both ways are mathematically equivalent but of course the first is easier and is preferred when considering the effect of receive jitter. When the jitter phase does not change along the time or x-coordinate, as is the case of receive or fully correlated transmit jitter, the same horizontal convolution can be applied to every trajectory of to the eye density as a whole.

In the presence of transmit jitter, each sample \( S_{n-k} \) in delay lock loop(3) experiences its own phase modification therefore horizontal modulation does not occur in ‘concert’, and convolution with jitter PDF is no more appropriate. In this case, only ‘vertical’ thinking gives the right answer. For all potentially important sample points of the response (excluding flat portions), the vertical distributions \( v_i (y) \) should be...
built and then combined together. This way, every summand in \( (3) \) becomes a random variable, or vertical noise: 
\[ v_{n-k}(y) \leftrightarrow S_{n-k}(\xi_k), \]
with its unique probability distribution. Depending on the bit combination and the difference \((b_k - b_{k-1})\), it may or may not contribute into the sum. In most cases, transmit jitter samples are considered statistically independent, and summation of the random values \( v_{n-k} \) means convolution of their PDFs. If this is not the case, the power of transmit jitter can often be separated into its slow (correlated) and fast (uncorrelated) portions. The first can be accounted for by a horizontal convolution, similar to receive jitter; the second can be described as a combination of vertical noise components.

Let’s now see how trajectories in the eye diagram are modified by transmit and receive jitter. Each trajectory is associated with a particular bit combination. If bit values in \( (3) \) are fixed, in absence of jitter each trajectory becomes a combination of segments of the step response. The more bit combinations we allow, the more different trajectories could be found in the eye diagram. For example, PRBS22 contains more different bit combinations than e.g. PRBS7, and hence creates more trajectories, many of them not possible with PRBS7 input. The eye therefore becomes more stressed with increasing pseudo-random bit sequence (PRBS) order. If in addition the channel gets certain transmit jitter, each trajectory becomes vertically ‘fuzzy’, as in Figure 7(a). Deviation of the vertical noise is not constant along unit interval, nor it is in inverse proportion to the slope of the trajectory, as in Figure 7(b) representing receive jitter. Let’s explain why. For example, our trajectory is a combination of the three step response segments, shown in Figure 6 in red and blue. The sum of these segments is nearly constant, which means almost horizontal trajectory. Still, the vertical noise is considerable, as it comes from two independent sources with relatively large deviation, as suggested by the slope of these segments. In Figure 7(b) we show this – much flatter - trajectory with associated vertical noise. Now, if we measure the effective horizontal jitter, we find it much larger than was the transmit jitter itself. This is the effect of ‘jitter amplification’, described and analyzed in a number of previous works [2, 3]. As we see, the two conditions are needed for it to happen: (A) relatively steep segments of the step response contribute to a certain trajectory, but their contributions are nearly canceled so that the slope of the trajectory (or some portion of it) is small, but vertical noise isn’t; and (B): bit combination that creates this trajectory minimizes the vertical eye opening thus making it susceptible to vertical noise. As we understand, with larger PRBS order, these two things have better chance to happen, that’s why we sometimes observe the effect of jitter increase with PRBS order.

### Statistical computation of eye density

Now, let’s look back to the expression \( (3) \). Jitter samples and the very coefficients \( b_k \) are random. Hence, \( y_n \) is also a random value. If we were able to find vertical PDF of \( y_n = y(nT + \tau), \tau \in [0, T] \), and then add the receive jitter, the eye diagram would be found. But how can we possibly do that? The expression
contains non-linear transformations of many random variables, and the multipliers are not only random, but mutually dependent. Indeed, the difference \((b_k - b_{k-1})\) indicates the sign of the transition: 0 means no transition, +2 - rising transition, and -2 - falling transition.

If we think of the edge response \(E(t)\), as a transition from low to high state, instead of the step response (transition from neutral/zero to high state), the factor ‘2’ is not needed since the swing of the latter is two times more. Similarly, the samples of the edge response are equal \(E_{N-k} = 2 \cdot S_{N-k}\).

Positive transitions can only happen from a negative state after a negative transition, and vice versa. The fact that the transition’s sign depends on the previous state, suggests that the algorithm should be organized as a state machine. Consider Figure 9(a). As the summation in (3) progresses, we move right and update the content associated with low (L) or high (H) states. The “content” should evidently be a partial PDF, updated by a series of summations. The content of \(L_k\) has ‘contributions’ from \(L_{k-1}\) (no transition) and \(H_{k-1}\) (negative transition). Likewise, \(H_k\) is contributed from \(H_{k-1}\) and \(L_{k-1}\). Transmit jitter should only be added at transitions, when the difference \((b_k - b_{k-1})\) is non-zero. The ‘weight’ of contributions coming from predecessors \((L_{k-1}, H_{k-1})\) equals the probability of positive, negative or no state transition. If the input pattern is uncorrelated, both contribute with identical weight, 0.5. Although the number of summands in (3) is infinite, it is enough to consider as many preceding bits as may give yields to a trajectory. Practically, this number depends on the duration of the non-flat portion of the step response divided on the bit interval. When computations end, \(L_N, H_N\) contains ‘negative’ and ‘positive’ portions of the eye density, suitable for conditional integration, as required by the BER definition.

In the absence of transmit jitter, the transition from \(L_{k-1}\) to \(H_k\) requires shifting the accumulated PDF up by \(E_{N-k}\). If the transmit jitter presents, the ‘shift’ becomes a convolution with vertical PDF \(v_{N-k}(y)\), as shown in Figure 4 for a particular sample point of the step or edge response \(t = (N - k) T + \tau\). Let \(P_{L,k}(y)\) and \(P_{H,k}(y)\) denote the partial eye-density (PDF) found in states \(L_k\), and \(H_k\) respectively. Then, the algorithm becomes:

1. Define initial states by setting: \(P_{L,0}(y) = \delta(y + PLE_{\infty}/2), P_{H,0}(y) = \delta(y - E_{\infty}/2)\).

2. For \(k = 1 \ldots N\),

\[
P_{L,k} = 0.5 [P_{L,k-1} + P_{H,k-1} \ast v_{N-k}(-y)]
\]

\[
P_{H,k} = 0.5 [P_{H,k-1} + P_{L,k-1} \ast v_{N-k}(+y)]
\]

3. The sum \(P_{L,N} + P_{H,N}\) is the resultant complete eye-density.

The latter is suggested by the structure shown in Figure 9(b). Expressions (4) are similar to those in [1], with the only difference that convolution (designated as \(\ast\)) is performed here with the vertical jitter noise functions, not the Dirac pulses representing jitter-less PDF.

**Convolution and the probability mass function (PMF)**

The algorithm described above is heavily based on convolution. All components, including samples of the step/edge responses must be represented by their PDFs. Numerical computation dictates that all distributions should be defined on a finite grid, or as a discrete PDF sometimes called probability mass function. Usually, fitting an arbitrary probability density onto a finite grid PMF requires preserving the first moments. The zero-order moment \(M_0\) is an integral of the PDF, should be equal 1; and the first moment \(M_1\)
(a) original value of $z$ positioned between the grid points

(b) the representation that keeps $M_0$ and $M_1$ unchanged

Figure 10: Values of $z$

(a) undesirable deviation caused by approximation of a discrete value by two points on the grid

(b) gaussian distribution with small sigma (less than grid size) suffering from same problem

Figure 11: Undesirable Deviations from Approximations

(a) dispersion creating non-zero values to the left of the bounded PDF

(b) it is possible to keep the boundary when the discrete PDFs are snapped to the grid

Figure 12: Effect of Dispersion Produced by the Approximation on a Discrete Mesh
red - grid size is 0.01V, blue - grid size is 0.001 V
eye density is computed for every point along one UI. As iterations progress, the PMF develops into the vertical cross-section of the eye density shown on the right.

is the mean value. Let’s consider how to represent discrete values with PMF, which could be for example, the initial PDFs in the algorithm above described by Dirac functions.

On a continuous coordinate axis, a discrete value $z$ is described by a Dirac function, as shown in Figure 10(a). However, on a discrete mesh, if we want to keep the first two moments, it should be represented by the two samples. The weights can be found from simple equations:

\[
M_0 = A_1 + A_2 = 1 \\
M_1 = A_1 Y_m + A_2 Y_{m+1} = z
\]  

At the same time, the second moment $M_2$ changes from 0 to $x (1 - x)$, where $x = (z - Y_m) / (Y_{m+1} - Y_m)$ is a relative position of $z$ between the points. The PDF acquires an additional standard deviation, $\sigma_{std} = \sqrt{M_2}$, shown on Figure 11(a), together with its ‘average’ shown by a horizontal line. The same effect applies to a Gaussian PDF. If the “sigma” of the continuous distribution is smaller than approximately half of the mesh size, but the center is positioned between the grid points, the apparent standard deviation of the discrete PDF can be much larger than this “sigma”, by an order or more (see Figure 11(b)).

When we convolve discrete PDFs, the additional ‘sigma’ may cause catastrophic consequences. Figure 12(a) illustrates parasitic dispersion of otherwise bounded PDF, after the discrete PDFs are combined by a series of convolutions. The expected PDF should be identically zero to the left of the green vertical line, however due to non-zero second moment in every convolved component, the resulting PDF spreads to the left.

Since low probability portion of the eye density and BER is of primary importance, the effect we observe in Figure 12(a) should not be allowed. Instead of replacing each discrete value by a pair, as illustrated in Figure 10, we snap it to the mesh, by rounding to the nearest mesh point. The second moment becomes zero, and the convolution of several entries gives properly bounded distribution, as in Figure 12(b), independently of the mesh size. Of course, rounding will not preserve the first moment. To prevent accumulation of displacements, we implemented a “balanced rounding” approach, where the direction of rounding is optimally selected by minimizing both individual and cumulative error.

Before we start analysis by the algorithm of Figure 9, every sample of the edge response should be rounded to the discrete mesh. Without transmit jitter, the computation does not require PDF transform-
(a) Sine phase jitter (red) makes the Tx transition time non-equidistant. (b) Accordingly, the step/edge response should be sampled in different moments, resulting in the different vertical PDFs.

Figure 14: Sine phase jitter

More effects and impairments

Non-Gaussian distribution of transmit jitter

Such seemingly important change requires only a minor modification to the Algorithm. For any given jitter distribution, Gaussian, uniform, Dual-Dirac, or anything else, the corresponding vertical PDFs can still be found in a manner depicted by Figure 6, and participate in convolution operations outlined by the algorithm.

Transmit jitter has deterministic component(s)

How is the algorithm modified if transmit jitter has both random and deterministic components? In this case, deterministic jitter can be considered as a modulation of the sampling positions of the step or edge response. The phase does not change linearly with time, but experience some disturbances (Figure 14(a), red) making the logic transitions non-equidistant. The modified timing affects the sampling of the step/edge response, and therefore, shapes and positions of the vertical PDFs $v_k(y)$, as in Figure 14(b).

Typically, an initial phase of the sine jitter is not known, therefore the main cycle of the algorithm should be performed several times, with different initial phase, to find the resulted average eye density and then BER. With several sine or duty-cycle jitter sources, one should organize several nested cycles in the same manner. Of course, this makes the entire simulation more expensive however the solution time increase is a matter of minutes.
Receive jitter and noise

If the receiver adds its own jitter or noise, uncorrelated with transmitted pattern, these can be simply accounted for by horizontal/vertical convolutions of the result eye density plots with jitter/noise PDF respectively. If the convolutions are applied to the accumulated eye densities, and they are converted into BER by integration, then the effect will be correctly passed to BER as well. But, the convolutions can be applied to BER after integration, if the eye densities have not been convolved with receive noise and jitter; the order of operations does not affect the result. If however receive jitter is correlated with the pattern, the algorithm becomes more complicated. Some approaches are considered in [4, 5] which are based on using the multi-step Markov chains or control system analysis, with jitter transfer linearization. In either case, computations cannot be performed in two separate stages.

Asymmetry of rising/falling transitions

So far, we assumed that rising and falling transitions are described by the same step or edge response. However, the Algorithm described above can work with two different responses. In this case, the two mirror PDFs $v_{N-k}(+y)$ and $v_{N-k}(-y)$ in (4) should be replaced by two independent PDFs, $v_{N-k,\text{rise}}(+y)$ and $v_{N-k,\text{fall}}(-y)$, found from rising and falling edges respectively.

Construction of Eye Diagrams from the Test and Measurement Perspective

Historically, eye diagrams have been used (and still are) to view and measure digital communications signals. The idea was to look at both setup and hold times by viewing the region of a Non-Return-to-Zero (NRZ) data signal that a logical 1 or 0 would be detected. The methods for setting-up an oscilloscope to present a picture of this zone was in many cases “crude”, especially if no “decision” clock was provided. The classical approach to constructing an eye diagram from digitized waveforms (such as for a digital oscilloscope) has not evolved very much in the last 2 decades. At the start of digital oscilloscopes, the idea was to simulate what had been traditionally viewed on an analog oscilloscope and interpreted visually. Today the usual procedure is waveforms are accumulated in a 2 dimensional (2D) histogram that can be colorized and viewed like a photo. Each sample point contributes at a vertical coordinate corresponding to its voltage value and a horizontal coordinate corresponding to its phase relative to a recovered (or presumed) clock that defines the boundaries of a single bit period. A 2D histogram Figure 23(c) is then incremented at the $\{v, \phi\}$ coordinate. Digital up-sampling can be used to more efficiently fill in the gaps between sample points. But as the bitmap accumulates, the process depends on the randomness of sampling phase with the phase of the digitized signal to fill in the gaps. You can analyze jitter and noise from such a 2D but it presents many problems.

This author spent much energy struggling to extract useful information from the eye diagram before realizing that too much information was lost in the manner the eye-diagram was constructed. When the vertical slice of a traditional eye diagram was analyzed to fit to a shape, the corners of the eye opening could not be treated unambiguously. So some years ago another approach was explored which for a short repeating test pattern amounts to no less than constructing a 2D histogram over the entire pattern. That is, each unit interval (UI) in the test pattern has it’s own eye-diagram. But each UI by reason of having its own nominal trajectory Figure 1(a) does not form what we usually imagine as an eye-diagram with an
eye-opening at the center. Rather the resulting eye diagram looks more like a blurry trajectory spanning multiple UIs Figure 15. Just imagining this kind of accumulation of data is very revealing since it offers a number of strategies that simplify analysis of the signal under observation.

So there is an approach to constructing an eye-diagram and related diagrams from acquired waveform data that is similar in spirit and conception to the previous discussions. There is a subtle paradigm shift since the focus is on a vertical analysis of data much like what is displayed in Figure 15. This one 2D PMF per UI is far easier to analyze, since the vertical distributions are quite well behaved compared to the composite of all UIs. Furthermore, the separation of UI’s that are decoded as a 1 or a 0 can be easily separated into two categories as indicated by the discussion of simulation of an eye diagram. There are no ambiguities in the “corners” of the eye diagram if each of these are analyzed before re-combining each UI to make a single eye diagram.

The fundamental nature of the trajectory of the analog signal that is being studied: \( v(t) \) is a single-valued function of time. For every time coordinate there is a voltage value that the signal has at that time. Viewed horizontally, \( t(v) \) is not single valued. Otherwise stated, some voltages may never occur, whereas every time has a voltage. So for an ideal PMF eye diagram, every vertical slice sums or integrates to the same probability as for every other vertical slice of the same width. The signal always passes through some portion of the vertical slice.

One problem with the classic method described above, is that it does not guarantee that the integral of the “density” for each vertical column of the 2D PMF is the same. That’s because we rely on statistical randomness to equalize the probabilities, and that is is only convergent in the limit of a very large amount of acquired data. There are other annoying characteristics of the classical eye-diagram. The process is basically non-convergent. That is to say, the diagram changes as more data is accumulated. This makes comparisons of eye-diagrams from a little tricky, since depending on how many UI and even how many digitized samples per UI figure into how the eye-diagram appears. This further makes traditional mask testing a dodgy matter.

A different approach is possible, an approach to constructing an Eye-diagram (and related diagrams) that is similar in spirit and conception to the simulation discussion above. There is a subtle paradigm shift. This methodology emphasizes vertical analysis and these are the steps: (for simplicity we will consider the case where there is a repeating data pattern, but methods exist for non-repeating data patterns and even “live” data)

• acquire waveform(s)
  – determine (or use a specified) threshold voltage
  – analyze threshold crossings (some may be missing)
  – perform clock data recovery (optionally using a software phase lock loop (PLL) or delay lock loop (DLL))
  – optionally simulate receiver equalization that can be trained (or manually adjust equalization) for purposes of decoding if the channel is “impaired”
  – decode the bit stream and determine if there is a pattern and what phase this acquisition has relative to it
  – for each UI in the acquisition, interpolate a voltage value over N uniformly spaced intervals and update a histogram for each of the N positions within the UI.

15
Figure 15: Multiple UI 2D histogram

(a) Classic Eye Diagram  
(b) New type 2D PMF Eye Diagram

Figure 16: 2-Classical (standard) Eye Diagram and the 2D PMF Eye Diagram using the proposed technique
when done processing all UI in the acquired waveform (or in multiple waveforms), we will next prepare two eye-diagram like 2D PMFs (on the same coordinate grid that would be used for an eye diagram), one for bits which are decoded as zero and one for bits which are decoded as 1

- parametrize each of the N histograms per UI in the pattern according to any of a number of models (including dual-Dirac)
- extrapolate each histogram to a detailed PMF
- for each coordinate between the N intervals, “morph” a new PMF between nearest neighbors and sum thisPMF into one or the other 2D PMF (according to whether this UI was decoded as a zero or a one)

now as was done for the simulation case, produce two new 2D BER plot on the same grid as the 2D PMF, summing the individual columns of the 2D PMF for ones from the bottom to top, and producing a 2D CDF or probability for a bit error over the whole surface of the Eye Diagram.

Now, I have left out many of the details about how each of these steps is accomplished, but in the industry, none of those steps are without existing examples \[6\] \[7, 8, 9\]. And this paper is not about a detailed recipe, but to show how such an approach gives some interesting and surprising results. The illustrations Figure 17(b) show the 2DPMF plots for ones and for zeros, and integrating these to obtain separate BER plots for ones and zeros Figure 17(b). (Lighter tones of color represent higher probability).

One satisfying result is that the 2D PMF can be created from this methodology which is what a classic eye diagram Figure 16(a) should be Figure 16(b). Unlike the classic eye diagram, it is “convergent”, meaning it does not grow into the central region as more and more waveforms are accumulated and analyzed. As such it can be used for a more objective kind of mask testing.

Perhaps the most interesting result, is that the two new 2D BER plots represent precisely the probability of a bit error for 1’s and for 0’s. and they may be combined (in correct proportion) to form what is commonly known as a “contour plot”. From this contour a horizontal slice at a given voltage level yields a “bathtub curve” such as the one usually obtained by jitter analysis.

**Test Cases**

In order to confirm this methodology gives results comparable to those obtained from the more current jitter-timing analysis, we consider 4 measurement test cases. In each case a differential laboratory generator
was used to produce a 17.1 Gbps data stream with a PRBS7 test pattern. (The 17.1 Gbps is arbitrary, but was chosen to give a mostly closed eye diagram when passed through a test fixture that was on hand with sufficient measured S-parameters as well as complete “stack-up” data for being used in the simulation part of this paper). The setup when using the “channel” is shown here Figure 18. A 36 GHz analog bandwidth 80 GS/s oscilloscope was used to capture test waveforms. The four test cases were comprised of:

1. Direct to scope (no test fixture or “channel”) with no induced random jitter.
2. Direct to scope (no test fixture or “channel”) with 1 ps RMS induced random jitter.
3. Test fixture interposed between generator and scope with no induced random jitter.
4. Test fixture interposed between generator and scope with 1 ps RMS induced random jitter.

In Figure 19 the 2D PMFs are shown for the 4 test cases. These are quite similar to classic eye diagrams when combined. Next, for each of the test cases the BER plots were obtained by first integrating the 2D PMFs from bottom to top for the ones, and from top to bottom for the zeros. In this case the BER-plots Figure 20 are shown in a temperature color-code rather than tones of blue and red. Additionally the colors are not continuous but “terraced” so bright red is for all probabilities between 1 and 1e-1, and for each decade in probability another color towards violet is used. The jitter bathtub curves can be obtained simply by taking a single row of the BER plot. These can be compared to the equivalent bathtub curve from a horizontal jitter analysis Figure 21. There are a number of caveats if you want good correlation between the two methods. The clock data recovery (CDR) method used for creating the new type eye diagrams must be the same. This is not much of an issue for simulation because the transmit frequency can be made perfect and is known. But the choice of PLL can dramatically affect the ISI jitter contribution, especially with longer test patterns containing longer run-lengths of zeros or ones. The number of columns used for
Figure 19: Both 2D PMF for ones (red) and for zeros (red), for 17.1 Gbps PRBS7. Lighter shades of red and blue represent higher probability density.
Figure 20: Composite BER plot, for 17.1 Gbps PRBS7 with and without Lossy Channel and with and without induced RMS Jitter. The Vertical scale is 72 mV for all cases.
Figure 21:

Figure 22: Jitter decomposition using bathtub curves from noise analysis and those from current timing analysis
the BER plot limits the precision of the vertical coordinate used for selection of a “row” of probabilities. Especially with a closed eye, this can be critical. In these examples the grid used is 1000 by 1000.

At the time of submission of this paper we are not yet satisfied that we understand why some of the jitter breakdown values shown in Figure 22 for random and deterministic jitter vary between the two methods. It is very encouraging that the Total Jitter is very consistence since this is of course the most important number.

**Pros and Cons for Signal Integrity Measurements**

There are several real benefits resulting from this method for creating these new kinds of eye diagrams and at least one drawback.

**Cons:**

1. This is a compute intensive kind of analysis requiring considerable computing power, memory resources and computation time

**Pros:**

1. The contour plot or BER plot is simple to obtain
2. The bathtub curve is easily obtained from the BER plot.
3. Jitter figures can be obtained with no explicit “timing analysis” from the bathtub curve
4. Because the new eye diagrams 2DPMF is “convergent”, it is more suitable for mask testing.
5. Because CDR is used, this analysis can be performed on spread spectrum clocked signals.
6. Equalization can be done strictly for CDR and decoding (sorting 1s and 0s) enabling analysis of noise and jitter for a completely “closed” eye.
7. Several by-products are readily accessible, like a noiseless version of the input signal and a noise only version of the waveform.

**Something New**

When working with this kind of analysis, there are a number of by-products that were unexpectedly interesting. Since the trajectories of every UI are known, they can be removed from the waveform under observation, leaving a “noise only” waveform. This waveform is extremely useful in tracking down non-random effects like cross-talk. You can also remove the trajectories of all UIs and for all of ones and zeros integrate from the top to the median and the bottom to the median yielding what the author calls a “flat eye diagram” Figure 23. What is remarkable about these is they reveal how much of the observed noise is produced from truly horizontal jitter sources. Note in the cases with no induced jitter Figure 23(a) and Figure 23(c), the image are less vertically fat at divisions 2 and 8 (crossing times) when compared to the induced jitter cases Figure 23(b) and Figure 23(d). Also note how much greater the effect is for the direct to scope case, since the jitter manifests as noise in proportion to the slope of the signal under observation and of course the channel data has much slower slopes.
(a) Direct to scope with no induced jitter
(b) Direct to scope with 1ps RMS induced jitter
(c) Lossy Channel with no induced jitter
(d) Lossy Channel with 1ps RMS induced jitter

Figure 23:
Conclusions:

There is a “duality” between Jitter and Noise at the heart of signal integrity simulation and measurement. The two physical phenomena each manifest as the other depending on how you look at the probability of a bit error. However, there is an advantage in focusing on vertical analysis and we have demonstrated how that approach is beneficial to both simulation and measurement. We have shown how it is both possible and useful to construct eye diagrams based on vertical analysis for simulation as well as for measurement. The methodology is easy to understand even if it does include some sophisticated methods to correctly incorporate noise assumptions and jitter assumptions for simulation purposes. For the measurement case, the incorporation of both noise and jitter is done for us by nature (physics) but the same approach to construction of the eye-diagram yields easy to understand results for estimating the BER of a signal under observation. The authors are pleased that we discovered our common views on these matters and that we’ve had the opportunity to share what we believe are valuable and exiting findings.
References


