Winter 2019

Efficient Characterization of Transmitter Output Jitter Components in 100GBASE-CR4 Ethernet

Paul Alan Willis
University of New Hampshire, Durham

EFFICIENT CHARACTERIZATION OF TRANSMITTER OUTPUT JITTER COMPONENTS IN 100GBASE-CR4 ETHERNET

BY

PAUL WILLIS

BS, Electrical Engineering, University of New Hampshire, USA, 2015

THESIS

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Master of Science

in

Electrical and Computer Engineering

December 2019

This thesis has been examined and approved in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering by:

Thesis Director, Michael J. Carter, PhD.
Associate Professor of Electrical Engineering

Nicholas J. Kirsch, PhD.
Associate Professor of Electrical Engineering

Richard A. Messner, PhD.
Associate Professor of Electrical Engineering

on 8/20/2019.

Original approval signatures are on file with the University of New Hampshire Graduate School.

This thesis dedicated to the University of New Hampshire InterOperability Laboratory.

ACKNOWLEDGMENTS

I would like to thank my friends at the lab: Hayden Hanes, Curtis Donahue, Mike Klempa, and Craig Chabot for making it a place I want to visit everyday. I would like to thank Blake Brown for inspiring me to become a better engineer and a better person.

TABLE OF CONTENTS

NOMENCLATURE

Math functions

erf	Gaussian error function
$erfc$	Complementary error function
$erfc^{-1}$	Inverse complementary error function

Symbols

$\delta - \delta$	dual-Dirac
f_0	$-3\,\mathrm{dB}$ frequency

Specifications

IEEE 802.3	Ethernet
IEEE 802.3 Clause 83D	Electrical characteristics of CAUI-4, 100 Gb/s Ethernet across four lanes over chip-to-chip channel
IEEE 802.3 Clause 92	Electrical characteristics of 100GBASE-CR4, 100 Gb/s Ethernet across four lanes over copper cable channel
IEEE 802.3 Clause 93	Electrical characteristics of 100GBASE-KR4, 100 Gb/s Ethernet across four lanes over backplane channel
IEEE 802.3 Clause 110	Electrical characteristics of 25GBASE-CR and 25GBASE-CR-S, 25 Gb/s Ethernet across a single lane over copper cable channel
IEEE 802.3 Clause 111	Electrical characteristics of 25GBASE-KR and 25GBASE-KR-S, 25 Gb/s Ethernet across a single lane over backplane channel
IEEE 802.3 Clause 120B.3	Electrical characteristics of 200GAUI-8 and 400GAUI-16, 200 Gb/s Ethernet across eight lanes and 400 Gb/s Ethernet across sixteen lanes over chip-to-chip channel

Abbreviations

BER	Bit error rate
BERT	Bit error rate tester
BUJ	Bounded uncorrelated jitter

CDF	Cumulative distribution function
CDR	Clock and data recovery
COM	Channel Operating Margin
DCD	Duty-cycle distortion
DDJ	Data dependent jitter
DJ	Deterministic jitter
DUT	Device under test
EBUJ	Estimated bounded uncorrelated jitter
ECC	Error correction code
EOJ	Even-odd jitter
ERJ	Estimated random jitter
ET	Equivalent time
ETUJ	Estimated total uncorrelated jitter
IEEE	Institute of Electrical and Electronics Engineers
ISI	Intersymbol interference
JTF	Jitter transfer function
NRZ	Non-return-to-zero
OJTF	Observed jitter transfer function
PAM	Pulse amplitude modulation
PDF	Probability density function
PJ	Periodic jitter
PLL	Phase-locked loop
PRBS	Pseudo random binary sequence
RJ	Random jitter
RT	Real time
RV	Random variable
SJ	Sinusoidal jitter

TIE	Time interval error
TJ	Total jitter
TOJ	Transmitter output jitter
UI	Unit interval
UNH	University of New Hampshire

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

ABSTRACT

Efficient Characterization of Transmitter Output Jitter Components in 100GBASE-CR4 Ethernet

by

Paul Willis

University of New Hampshire, December, 2019

Electrical data communication links less than 10 meters long are increasing link rates at a steady pace. As link rate increases, timing tolerances become more important to ensure low bit error rates (BER). Ensuring high performance links in scenarios with large amounts of crosstalk requires the characterization of transmitter output jitter (TOJ). Some jitter models are incapable of effectively separating Gaussian random jitter from jitter caused by crosstalk. To overcome this, IEEE 802.3-2018 Clause 92 100GBASE-CR4 defines TOJ tests using the dual-Dirac ($\delta - \delta$) model. Additionally the 100GBASE-CR4 TOJ test definition separates out components of jitter by selecting isolated edges in a test pattern to be sampled. Applying the $\delta - \delta$ model to specific edges in a test pattern is not widely present in other jitter test methodologies. In this thesis the IEEE 802.3-2018 Clause 92 TOJ tests are implemented, and issues related to measurement time, captured data size, and measurement accuracy are addressed. A set of measurements were taken of a signal generator with a set of expected worst case jitter components applied to the signal. These measurements are explored to validate the implementation and examine limitations of the test definition.

CHAPTER 1

INTRODUCTION

1.1 Rationale of testing telecommunications equipment

Implementers of high speed networks place value in reliability and interoperability among devices. It is useful to be able to plug two modules into each other and reliably have a low bit error rate (BER) link established. In order to guarantee this, there are performance metrics for transmitters and receivers that can be characterized and compared against some specification. It is the role of standards bodies to define the details of links and the performance metrics to be met. The Institute of Electrical and Electronics Engineers (IEEE) creates and maintains the Ethernet standard (IEEE 802.3). IEEE 802.3 is used in many high speed networks, as evidenced by vendors from various networking industries requesting the development of new clauses [1, p. 152].

Error correction coding (ECC) techniques are a common way to handle a certain amount of errors. If a link has a BER that is too high, then the effective throughput and latency of that link is impacted, even with the use of ECC. Receivers must be able to decode malformed signals and transmitters must be able to transmit clean signals. Jitter is an important component of transmitter characterization to meet BER goals. If the edges come too early or too late compared to the recovered clock's sample time then the symbol could be sampled too close to the edge. If a symbol is sampled too far along an edge then the receiver will sample a symbol at the incorrect voltage and an error will occur.

1.2 Rationale of research

The Ethernet standard includes test definitions for each specification, but not all the necessary details to implement the tests. This allows flexibility in test equipment, but also allows for implementations that have needlessly long test times or that can produce inaccurate results. Many of the IEEE 802.3-2018 [2, s. 6, p. 411-459] Clause 92 (100GBASE-CR4) test definitions are simple to

implement. However, the 100GBASE-CR4 transmitter output jitter (TOJ) measurement methodology is more involved compared to other test definitions in the 100GBASE-CR4 specification. The 100GBASE-CR4 TOJ tests use Gaussian fitting, which have specific assumptions of the measured data that must be recognized. While implementing 100GBASE-CR4 TOJ tests, it is possible to make decisions that result in inaccurate measurements or that result in the tests taking longer than necessary to perform. Additionally, while implementing 100GBASE-CR4 TOJ tests it is possible to misinterpret the test definition. The purpose of this thesis is to explore the choices that must be made by an implementer of the 100GBASE-CR4 TOJ tests and provides information on the behavior of the IEEE 802.3 clause 92 TOJ test definitions. The results of this thesis can be used to aid interpretation of device results and definitions of future jitter tests.

This thesis only examines the case of IEEE 802.3-2018 subclause 92.8.3.8.2 *Effective bounded uncorrelated jitter and effective random jitter* (EBUJ and ERJ). This test definition has been adopted by other IEEE 802.3-2018 clauses listed in Table 1.1. This test definition is important because it is used by all 25 Gbps non-return to zero (NRZ) Ethernet technologies that operate over copper cable or backplane channels.

Table 1.1. IEEE 802.3-2018 clauses that use Clause 92 TOJ tests.

Clause	Technology	Reference
83D	100 Gb/s CAUI-4 chip-to-chip	Table 83D-1
93	100GBASE-KR4	93.8.1.7
110	25GBASE-CR and 25GBASE-CR-S	110.8.3
111	25GBASE-KR and 25GBASE-KR-S	111.8.2
120B.3	200GAUI-8 and 400GAUI-16 chip-to-chip	120B.3.1

CHAPTER 2

THEORY

2.1 Concepts of jitter

2.1.1 Definition of jitter

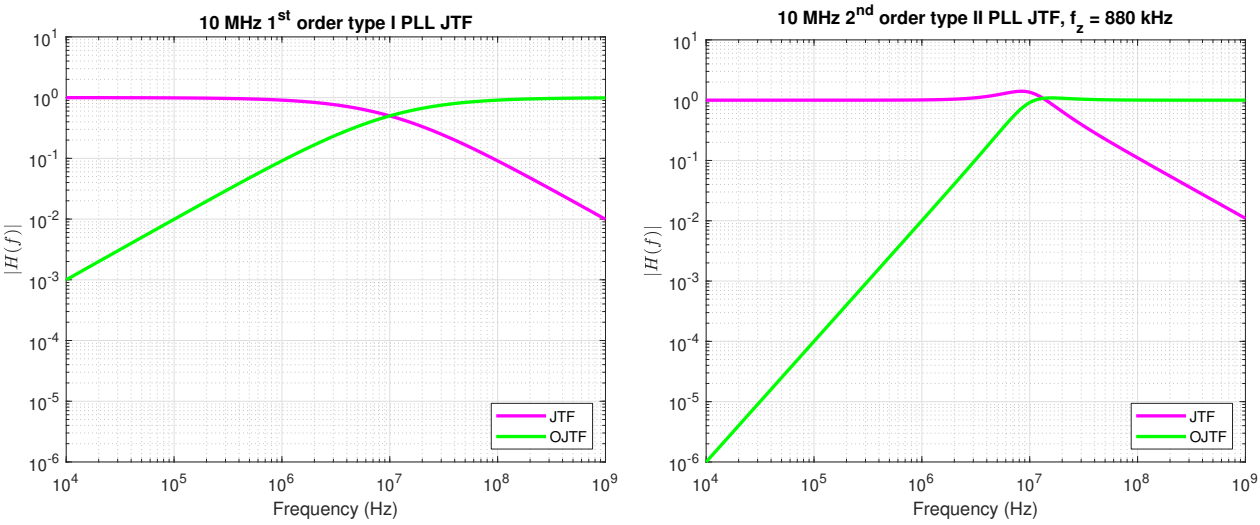
Jitter is the phase error of a signal compared against a reference clock with a spectrum greater than $10\,\mathrm{Hz}$. Wander is defined as phase error of a signal compared against a reference clock with a spectrum less than $10\,\mathrm{Hz}$ [3, p. 1-3]. Most baseband communications systems embed clock information in the data signal. This removes the requirement of having an extra channel to distribute a clock signal at the expense of requiring line coding that guarantees a minimum number of transitions in any sequence of symbols, also known as transition density. Receivers reconstruct a clock signal from a received data signal using a clock and data recovery (CDR) circuit that is typically composed of a phase-locked loop (PLL) [4, p. 47]. PLLs can track phase error at frequencies less than than their loop filter bandwidth. Jitter at frequencies greater than the PLL loop filter bandwidth results in phase error between the reconstructed clock and the signal that the receiver samples.

A CDR's jitter transfer function (JTF) is the ratio of the phase error at the CDR's output to the phase error present in CDR's input [5, p. 2]. The observed jitter transfer function (OJTF) is the inverse of the JTF. The JTF magnitude spectrum and OJTF magnitude spectrum of a given ideal system sum to unity (2.1) because a CDR should not apply gain to jitter. The OJTF magnitude spectrum is useful because it is the spectrum of jitter that is not tracked by a receiver's CDR. Jitter in the OJTF passband negatively affects BER.

$$|OJTF| = 1 - |JTF| \tag{2.1}$$

Figure 2.1 shows the JTF of two typical PLL implementations. 1st order PLLs have a less steep roll off, which results in high frequency jitter being tracked worse than 2nd order PLLs. Conversely,

2nd order PLLs track high frequency jitter better than 1st order PLLs, but typically have an underdamped response. This underdamped response causes peaking in gain below the cutoff frequency, which causes jitter in the peaking band to affect the signal more. The choice between 1st and 2nd order PLLs is influenced by the expected jitter spectrum.



(a) 1st order type I PLL JTF.

(b) 2nd order type II PLL JTF.

Figure 2.1. Example ideal PLL jitter transfer functions.

The ability of CDRs to track low frequency phase error means wander is only an issue if the maximum frequency deviation of the received signal is outside of the CDR's tracking range. The CDR tracking range is specified by two 100GBASE-CR4 tests:

- IEEE 802.3-2018 subclause 92.8.3.9 *Signaling rate range* for transmitter
- IEEE 802.3-2018 subclause 92.8.4.6 *Signaling rate range* for receiver

Eye diagrams are useful tools for evaluating characteristics of transmitters. Eye diagrams are constructed by stacking symbols on top of each other using a reference clock to define the length of each symbol. The jitter not tracked by the CDR used to make the reference clock is made apparent in the edge crossings of an eye diagram. Time interval error (TIE) is the difference between when an edge is expected to arrive relative to a reference clock and when it actually arrives. Figure 2.2

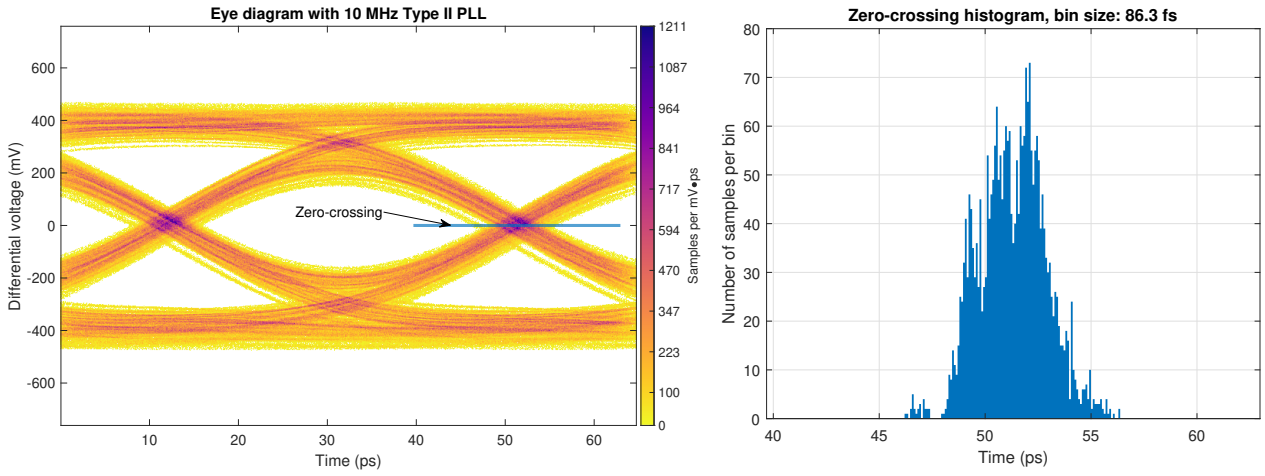
is an example eye diagram captured with an equivalent time (ET) oscilloscope. It is possible to generate eye diagrams with data captures from real time (RT) oscilloscope, however it requires the collection of long sequences of data and a CDR implemented in post processing.

This thesis will refer to the edge crossings of two-level NRZ signals as zero-crossings. "Edge crossings" is a more general term that should be used with modulation schemes and/or technologies that have edge crossings that are not necessarily at zero volts.

The zero-crossing histogram is taken from -0.2 unit interval (UI) to $+0.2$ UI centered around the right edge mean zero-crossing time. UI is a time unit that is equivalent to the length of time of one symbol at a given symbol rate. This is the same time range that all other measurements in this document use. This zero-crossing histogram is the distribution of TIE values for the observed edges. 1 UI @ 25.78125 GBaud is $38.\overline{78}$ ps. In the case of Figure 2.2, $t = 0$ on the left edge of the plot. The left zero-crossing is at $t = 0.\overline{3}$ UI ($12.\overline{92}$ ps). The mean of the right zero-crossing is nominally at $t = 1.\overline{3}$ UI ($51.\overline{71}$ ps). The -0.2 UI and $+0.2$ UI bounds used in the zero-crossing histogram are nominally at $43.\overline{95}$ ps and $59.\overline{47}$ ps, respectively. The actual bounds present in Figure 2.2 are slightly offset from these nominal values because the actual right edge zero-crossing mean is not at its nominal location of $1.\overline{3}$ UI from the left edge.

Measuring jitter from an eye diagram directly takes a long time because most of the samples collected by the oscilloscope are not zero-crossing samples. Working with eye mode on the Keysight 86108B also limits the time resolution to $86\,\mathrm{fs}$ per bin. When measuring jitter with an ET oscilloscope it is a good practice to reduce the horizontal range to only be as wide as the largest expected jitter value. Having a needlessly wide horizontal range on an ET oscilloscope will increase time because more samples are captured that are not adjacent to zero-crossings. The smaller horizontal range also allows for smaller histogram bin sizes on the oscilloscope histogram.

The center of an eye diagram is considered the point that a receiver samples the signal. In order for the receiver to slice the correct voltage level the previous edge must occur before the center of the eye and the next edge must occur after the center of the eye. The BER is increased by the probability that the TIE will be greater than 0.5 UI.



(a) Example eye diagram.

(b) Example zero-crossing histogram.

Figure 2.2. Example zero-crossing of a 100GBASE-KR4 calibrated stressed RX signal (pre-channel, no random interferer) generated by an Anritsu MP1900A and captured on a Keysight 86108B (CDR: 2nd order type II PLL, $f_0 = 10\,\mathrm{MHz}$, $f_{transition} = 880\,\mathrm{kHz}$).

2.1.2 Jitter decomposition

Jitter is typically decomposed into random and deterministic components [6, p. 27]. The decomposition of jitter is laid out in Figure 2.3. The random component is defined as a Gaussian distribution with unbounded tails. The deterministic component is modeled as bounded jitter composed of periodic jitter (PJ), sinusoidal jitter (SJ), data dependent jitter (DDJ), and bounded uncorrelated jitter (BUJ). PJ is jitter with a spectrum that is constant but uncorrelated with the data pattern. PJ may come from power supplies or crosstalk from adjacent clock signals. SJ is a subset of PJ that occurs only at one frequency. The definition of PJ has an overlap with the periodic portion of BUJ, but is separated because early jitter decomposition definitions separated SJ and BUJ. DDJ is jitter that is correlated with the data pattern and is composed of duty-cycle distortion (DCD) and intersymbol interference (ISI). DCD is a variation in pulse width dependent on the logic level. ISI is jitter caused by changes in a symbol's voltage dependent on the data of previous symbols. ISI is caused by nonconstant group delay in channels. Bounded uncorrelated jitter (BUJ) is all jitter that does not fall into other categories. BUJ is commonly caused by crosstalk aggressors that are not correlated with the victim signal [7, p. 1]. Crosstalk is electromagnetic coupling of outside

sources (aggressors) to a signal of interest (victim). BUJ is occasionally separated into periodic and non-periodic components, with non-periodic components being difficult to distinguish from RJ.

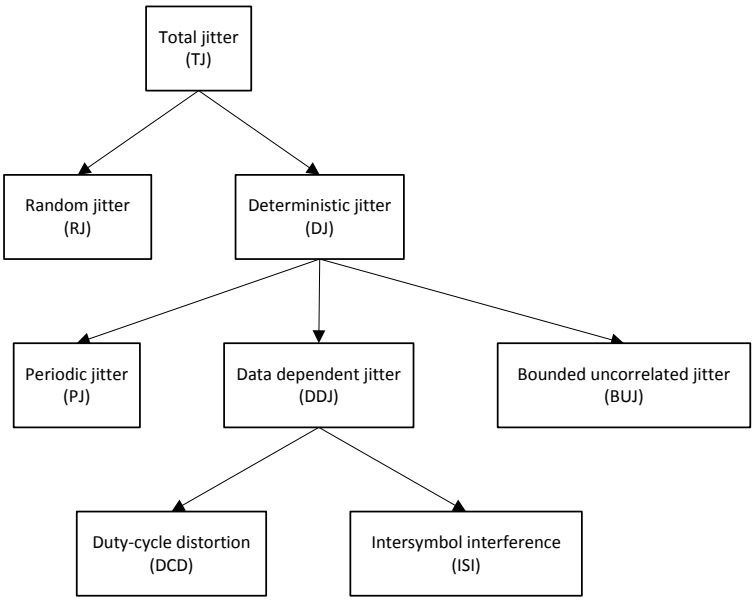


Figure 2.3. Jitter component hierarchy.

2.2 Transmitter output jitter tests

It is possible to gain a better understanding of the intention of the 100GBASE-CR4 TOJ tests by examining the factors that motivated the test methodology that IEEE 802.3bj (100 Gb/s Backplane and Copper Cable Task Force) adopted [8, p. 11]. Some jitter decomposition techniques do not strongly distinguish between non-periodic BUJ and RJ. Non-periodic BUJ is a large contributor to TJ in 100GBASE-CR4 due to the presence of adjacent lane crosstalk aggressors. The dual-Dirac model was considered in order to allow for convenient oscilloscope-based test setups without overly pessimistic estimations of RJ. The test methodology was first described at an IEEE meeting [9, p. 7-16] and adopted into IEEE 802.3bj-2014, which was officially adopted in IEEE 802.3-2015. IEEE 802.3-2018 subclause 92.8.3.8.2 *Effective bounded uncorrelated jitter and ef-*

fective random jitter has fewer details than the original presentation, so the presentation is useful to reference when making implementation choices. IEEE-802.3 likely removed the details in the original presentation to not show a preference towards certain hardware solutions in test and measurement.

2.2.1 Gaussian distributions

The Gaussian distribution probability density function (PDF) is defined in (2.2). The Gaussian PDF is greater than zero for all real inputs: the tails are unbounded.

$$f\left(x|\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.2}$$

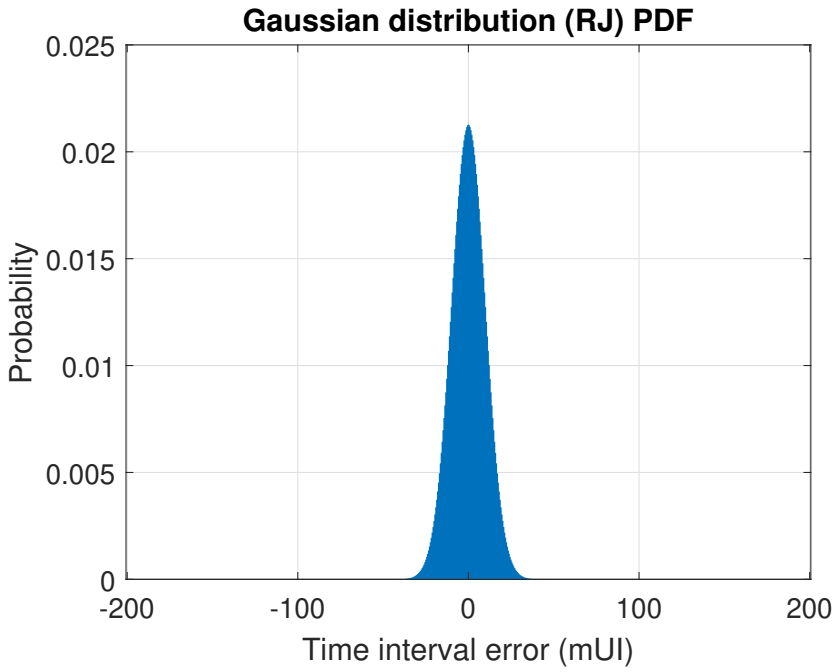


Figure 2.4. Example Gaussian PDF.

The Gaussian cumulative distribution function (CDF) is defined in (2.3).

$$\int_{-\infty}^{x} f\left(t|\mu,\sigma^2\right) dt = \frac{1}{2}\left[1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] \tag{2.3}$$

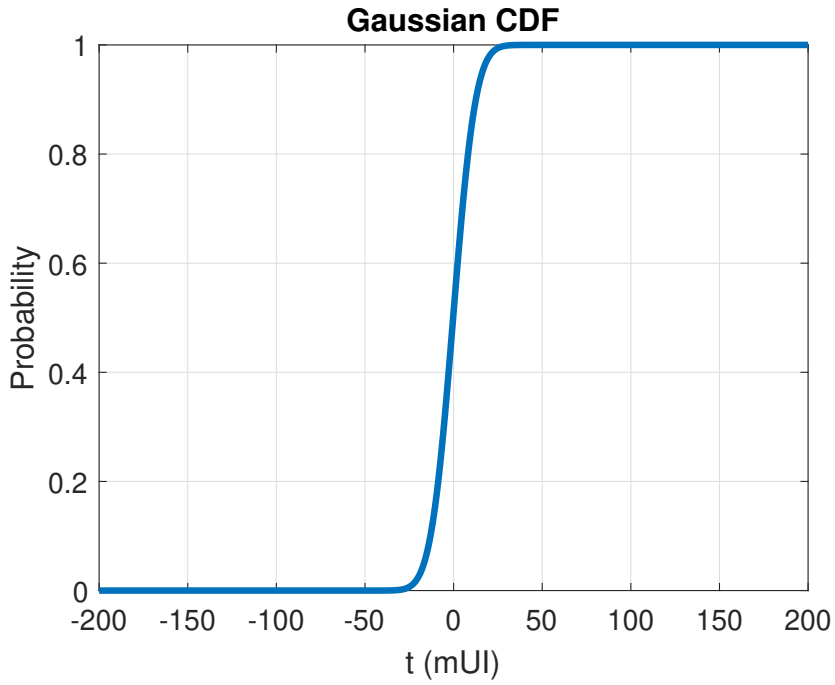


Figure 2.5. Example Gaussian CDF.

The Gaussian error function (erf) maps a range of a Gaussian distribution to the probability of that range. erf is defined in (2.4). erf, and by extension all functions based off of it, do not have closed form solutions.

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt \tag{2.4}$$

The complementary error function ($erfc$) is defined in (2.5). Due to Gaussian PDFs being symmetric about their mean, the $erfc$ is equivalent to taking the CDF of a Gaussian distribution with the horizontal axis reversed (right to left instead of left to right).

$$erfc(x) = 1 - erf(x) \tag{2.5}$$

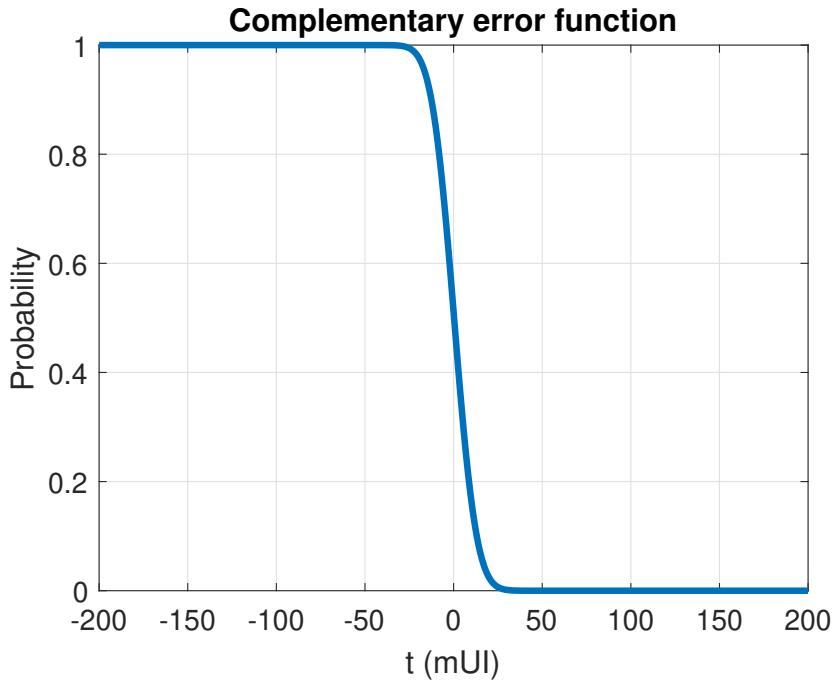


Figure 2.6. Example complementary error function.

The inverse complementary error function is defined in (2.6).

$$erfc(erfc^{-1}(x)) = x \tag{2.6}$$

$erfc^{-1}$ converts a Gaussian CDF to the range that it corresponds to. $erfc^{-1}$ is used as a metric of how Gaussian a distribution is. Using Equation (92-12) in IEEE 802.3-2018 (2.7), the BER scale of CDFs can be converted to Q.

$$Q(BER) = \sqrt{2} \times erfc^{-1}(2 \times BER) \tag{2.7}$$

$Q(x)$ is directly proportional to $erfc^{-1}(x)$. If a random variable (RV) has a Gaussian component in its PDF, then the tail of that Gaussian component corresponds to high Q values. The higher the Q curve value, the further along the Gaussian tail the CDF is. The higher Q curve values correspond to confidence in lower BER values [6, p. 465]. These higher Q curve values are filled in by observing more samples.

2.2.2 Dual-Dirac model

The dual-Dirac ($\delta - \delta$) model is a method of splitting TJ into RJ and DJ [10, p. 3] [11, p. 1]. The $\delta - \delta$ model assumes TJ is composed of two Gaussian distributions summed together. The $\delta - \delta$ model fits two Gaussian distributions from the tail regions on the left and right of the TIE histogram. The name "dual-Dirac" comes from the use of two Dirac delta functions convolved with Gaussian distributions. The time offset (mean of the Gaussian distributions) determines the magnitude of the measured DJ component. The standard deviation of the Gaussian distributions determines the magnitude of the measured RJ component. The $\delta - \delta$ model allows extrapolation of RJ to predict TJ values at very low BER values while only needing to observe enough samples to fit the tails of the Gaussian distributions.

The slope of a Q curve is increased by the presence of DJ that is Gaussian in shape but with truncated tails. The central limit theorem can be applied to summed components of DJ and may result in linear regions of the Q curves that still have DJ present. In order to separate RJ from DJ, the Q range used to characterize RJ must be linear and be outside the bounds of DJ. There is no currently known method of guaranteeing that a specific range of Q is outside of the bounds of DJ [10, p. 12]. However, using high Q value ranges makes it much more likely to be outside of the range of DJ. The 100GBASE-CR4 TOJ test definition defines a fixed Q range on which linear regressions are performed to characterize RJ. This assumes upper limits on how Gaussian the shape of a DJ distribution is expected to be and the amplitude of the DJ.

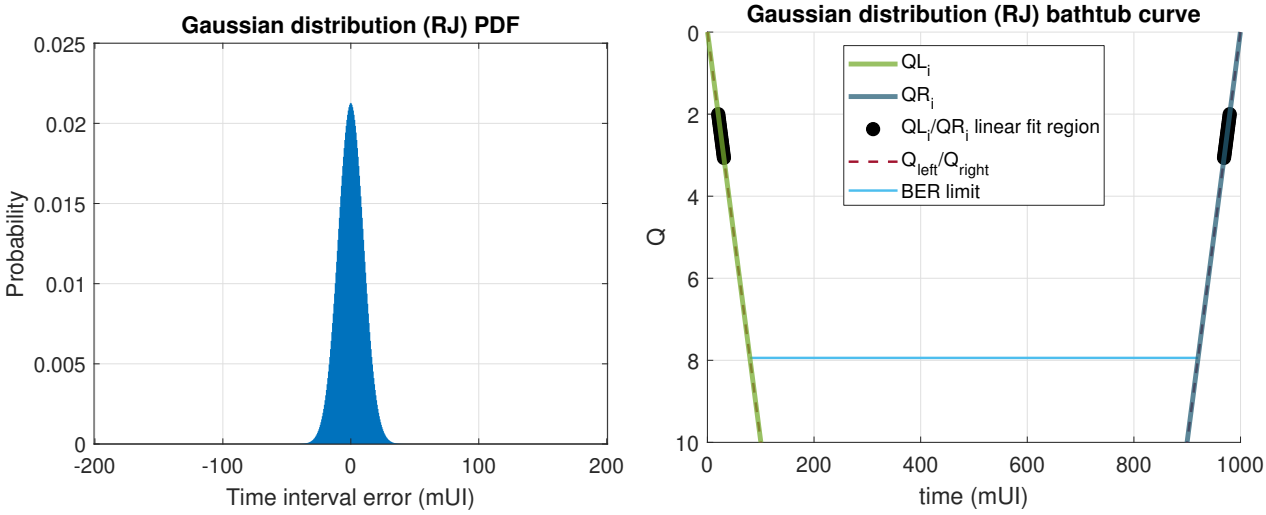
The $\delta - \delta$ model begins with an edge crossing PDF. Two CDFs are made from this PDF: one going from left to right and the other going from right to left (CDFL and CDFR). $erfc^{-1}$ is applied to the CDFs to generate Q curves: QL_i and QR_i.

If the region of the Q curve is high enough it should be outside of the bounds of DJ and be representative of histogram samples that are purely due to RJ. This purely RJ region can then be extrapolated to the low target BER without observing $\frac{3}{BER}$ samples. Observing $\frac{3}{BER}$ samples is necessary to be 95% confident that the actual BER is below the observed value [12, p. 15] when jitter is not decomposed into DJ and RJ. The ability to extrapolate RJ is the key benefit that the $\delta - \delta$ model provides. The extrapolated Q curves are denoted as Q_{left} and Q_{right}. The maximum Q value that DJ influences is dependent on the shape and amplitude of the DJ. The slope of the Q curve in the high Q region is determined by RJ. This is demonstrated in a simulated case where only RJ is present (Figure 2.7). In bathtub curves the right curve has 1 UI of time offset applied to it. The actual data used in calculations does not have this time offset applied to it and the mean of both left and right distributions is set to $0\,\mathrm{s}$.

Figure 2.7a is the PDF of a simulated edge crossing with Gaussian jitter applied to it. The horizontal axis is the time offset from the ideal crossing time in mUI: thousandths of a symbol time. The vertical axis is the probability of an edge arriving at a given time relative to a recovered clock.

Figure 2.7b is the bathtub curve of the simulated edge crossing histogram. QL_i is the likelihood of an edge on the left side of an eye diagram of occurring at a given time. QR_i is the likelihood of an edge on the right side of an eye diagram of occurring at a given time. The black dots indicate the points used to generate the linear fit curves: Q_{left} and Q_{right}. This linear fit region is defined by IEEE 802.3-2018 100GBASE-CR4 TOJ test definition. This region is assumed to be sufficiently outside of the bounds of DJ.

The slope of a Q curve is equivalent to RJ. This is demonstrated in a simulated case where only RJ is present (Figure 2.7).

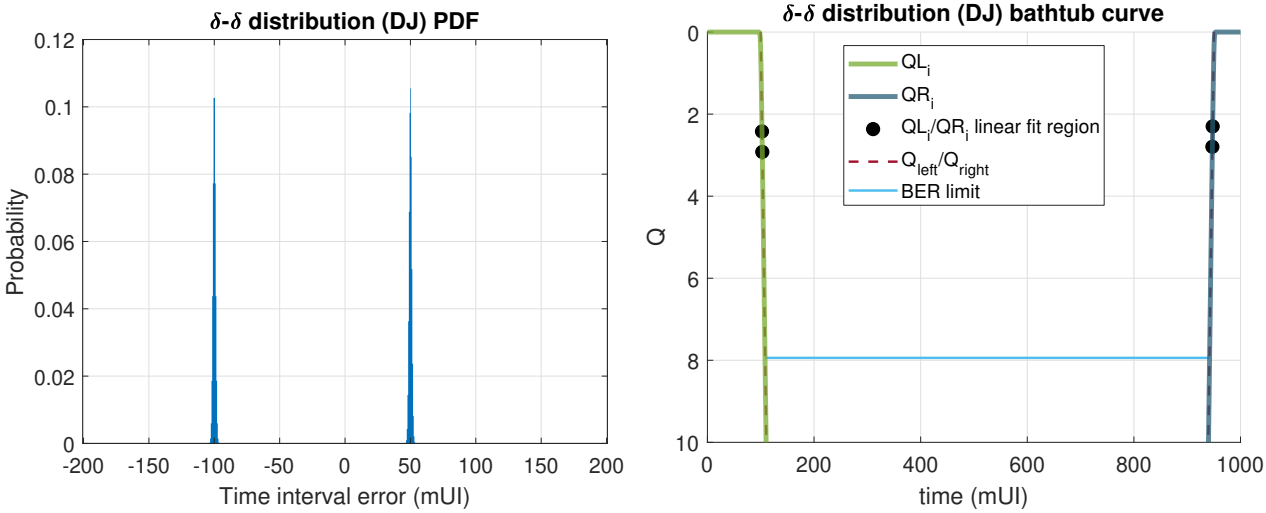


(a) Gaussian (RJ) PDF.

(b) Gaussian (RJ) bathtub curve.

Figure 2.7. Simulated Gaussian (RJ) PDF and bathtub curve ($\sigma = 10$ mUI, $\mu = 0$ mUI).

The time offset (x-intercept) of a Q curve is equivalent to DJ. This is demonstrated in a simulated case where only DJ is present (Figure 2.8). QL_i is shifted by 100 mUI and QR_i is shifted by 50 mUI, corresponding to the eye closure caused by the DJ distribution centered around each Dirac delta function.

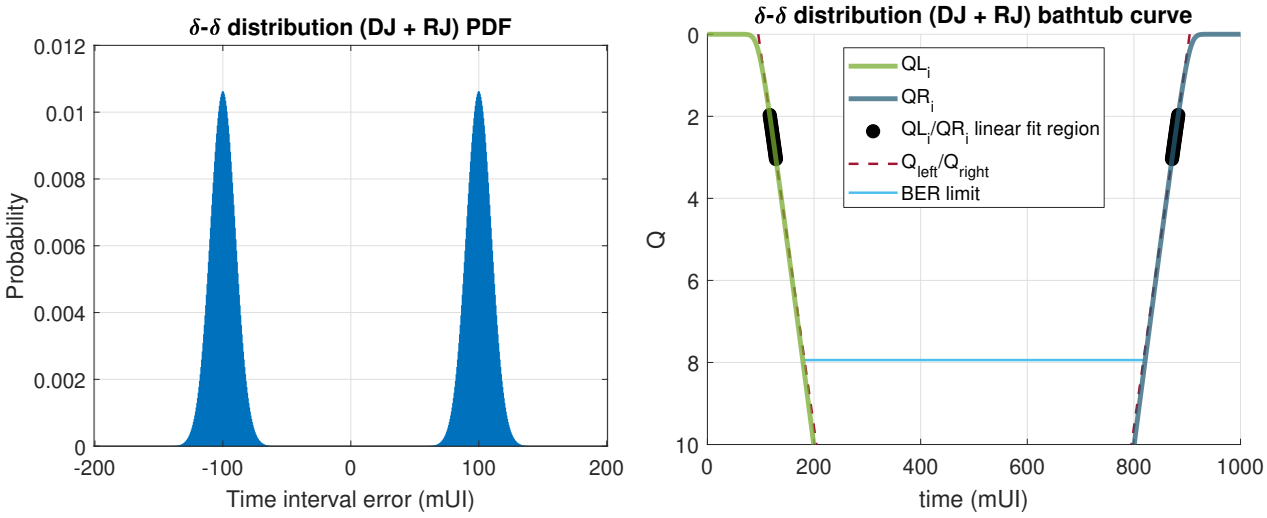


(a) $\delta - \delta$ (DJ) PDF.

(b) $\delta - \delta$ (DJ) bathtub curve.

Figure 2.8. Simulated $\delta - \delta$ (DJ) PDF and bathtub curve ($\sigma = 1$ mUI, $\mu = \{-100, +50\}$ mUI).

These two effects are independent of each other. This is demonstrated in a simulated case where both DJ and RJ is present (Figure 2.9). Both the slope and x-intercepts of the Q curves are influenced by the presence of RJ and DJ.

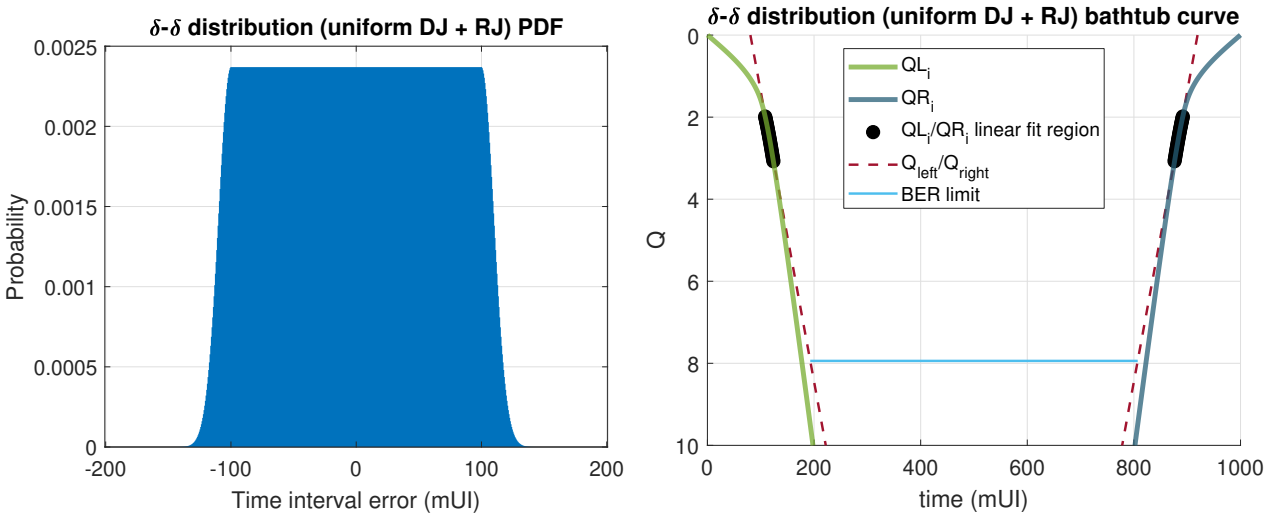


(a) $\delta - \delta$ (DJ + RJ) PDF.

(b) $\delta - \delta$ (DJ + RJ) bathtub curve.

Figure 2.9. Simulated $\delta - \delta$ (DJ + RJ) PDF and bathtub curve ($\sigma = 10$ mUI, $\mu = \{-100, +100\}$ mUI).

Different DJ distributions affect where the linear region of Q begins (Figure 2.10). When the bounded DJ distribution is similar to a truncated Gaussian distribution this can cause an under-estimation of BUJ and over estimation of RJ when the linear fit region is not in sufficiently high enough values of Q to be entirely outside of the bounds of DJ.



(a) $\delta - \delta$ (uniform DJ + RJ) PDF.

(b) $\delta - \delta$ (uniform DJ + RJ) bathtub curve.

Figure 2.10. Simulated $\delta - \delta$ (uniform DJ + RJ) PDF and bathtub curve ($\sigma = 10$ mUI, $\mu = \{-100, +100\}$ mUI).

The time between the Q curves at a given BER is the expected horizontal eye opening at that BER. This type of plot is typically referred to as a bathtub curve. Bathtub curves have horizontal units in seconds or UI. Bathtub curves have vertical scales in logarithmic BER when displaying CDFs. Bathtub curves have inverted vertical scales when displaying Q curves to mimic the appearance of a BER scale bathtub [10, p. 10].

Estimated bounded uncorrelated jitter (EBUJ) is defined in Equation (92-18) in IEEE 802.3-2018 (2.8). It is assumed that the Q curves will have the same characteristics as typical bathtub curves. That is to say: Q_{left} is a negatively sloped ($m_{left} < 0$) line with a positive x-intercept ($b_{left} > 0$) and Q_{right} is a positively sloped ($m_{right} > 0$) line with a negative x-intercept ($b_{right} < 0$).

$$EBUJ = \frac{b_{left}}{m_{left}} - \frac{b_{right}}{m_{right}} \tag{2.8}$$

Estimated random jitter (ERJ) is defined in Equation (92-18) in IEEE 802.3-2018 (2.9). ERJ is equivalent to the inverse of the average magnitude of the slopes. ERJ is representative of one standard deviation, commonly referred to as the RMS RJ value.

$$ERJ = \frac{m_{left} - m_{right}}{2 \times m_{left} \times m_{right}} \tag{2.9}$$

Estimated total uncorrelated jitter (ETUJ) is defined in Equation (92-19) in IEEE 802.3-2018 (2.10). The reason ERJ is multiplied by 7.9 is to extrapolate from the RMS RJ value to BER= 10^{-15} (2.11).

$$ETUJ = 7.9 \times ERJ \times EBUJ \tag{2.10}$$

$$Q\left(10^{-15}\right) \approx 7.9 \tag{2.11}$$

The 100GBASE-CR4 TOJ tests only define measurement limits for EBUJ and ETUJ. This thesis reports EBUJ, ETUJ, and ERJ. EBUJ and ETUJ are reported to evaluate measured results against the test limit. ERJ is reported to evaluate how well the $\delta - \delta$ model is working.

2.2.3 Test methodology

IEEE 802.3-2018 subclause 92.8.3.8.2 defines the math operations necessary to go from zero-crossing histograms to EBUJ, ERJ, and estimated total uncorrelated jitter (ETUJ). It is important to note that ETUJ is not the same as TJ. TJ includes jitter components that are correlated with the data signal: DDJ. TJ at a given BER is also directly observed with 95% confidence by observing $\frac{3}{BER}$ samples whereas ETUJ is an extrapolation based on the $\delta - \delta$ model. The IEEE 802.3-2018 subclause 92.8.3.8.2 test definition removes DDJ by only measuring jitter on specific edges and subtracting the mean time offset. Any DDJ should appear as a static time offset in the TIE for any one specific edge in a repeating data pattern. The mean jitter of the edges of interest are set to zero so, that any jitter component that tracks the data pattern is removed. This assumption is accurate because the test methodology involves only taking zero-crossing histograms on specific edges in a PRBS9 (Pseudo random binary sequence, $2^9 - 1$ bits long) test pattern:

- five zeros and four ones
- nine ones and five zeros

These selected edges are considered "lone edges" because there are no other transitions within a few symbols. In the PRBS9 test pattern these two subsequences have the longest runs of symbols preceding and following the transition of interest. EBUJ is an estimate of PJ and BUJ. ERJ is calculated by taking the linear regression of the Q curves at CDF ranges $2.5 \cdot 10^{-2}$ to 10^{-3}, establishing the expected bound of DJ in each of the two Gaussian distributions in the $\delta - \delta$ model. ETUJ is the combination of EBUJ and ERJ extrapolated to a BER of 10^{-15}. Using Equation (92-12) in IEEE 802.3-2018, the BER scale of CDFs can be converted to Q (2.7). The Q scale CDF range of interest is $1.96 < Q < 3.09$. The Q scale of the target BER is 7.94.

2.3 Interpretation of test definition

The first sentence in IEEE 802.3-2018 subclause 92.8.3.8.2 *Effective bounded uncorrelated jitter and effective random jitter* states:

> "Effective bounded uncorrelated jitter and effective random jitter are measured on each of two specific transitions in a PRBS9 pattern (see 83.5.10)."

The way that rising and falling edge histograms are intended to be used has some room for interpretation. Here are three interpretations that have been explored.

Interpretation A The zero-crossing histograms should be sample-wise summed together into a single PDF. The timebases of the two histograms should be combined by linearly interpolating the PDFs to a common set of bins. QL_i and QR_i is generated using the singular resultant histogram. This interpretation is explicitly described in a later TOJ definition: IEEE 802.3-2018 subclause 120D.3.1.8.1 *J_{4u} and J_{RMS} jitter*. This test definition goes into greater detail about how to do this combination for PAM4 signaling. **Interpretation A** is the equivalent of averaging the left and right zero-crossings of an eye diagram made from lone edges and duplicating the resultant zero-crossing on the left and the right of the eye diagram.

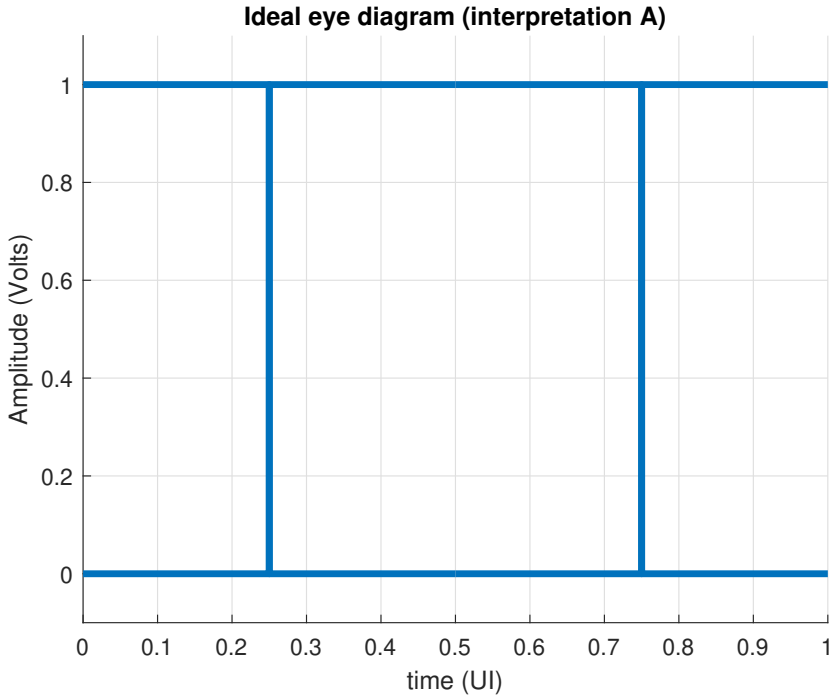


Figure 2.11. Interpretation A ideal eye diagram.

Interpretation B The zero-crossing histograms should be kept separate. QL_i is generated using one histogram and QR_i is generated by using the other histogram. There are two configurations for this interpretation:

1. The rising edge histogram is used to generate QL_i and the falling edge histogram is used to generate QR_i.
2. The falling edge histogram is used to generate QL_i and the rising edge histogram is used to generate QR_i.

Both configurations are analyzed and the worst case values are chosen. **Interpretation B** is the equivalent of making an eye diagram with the left zero-crossing as a lone rising/falling edge and the right zero-crossing as a lone falling/rising edge.

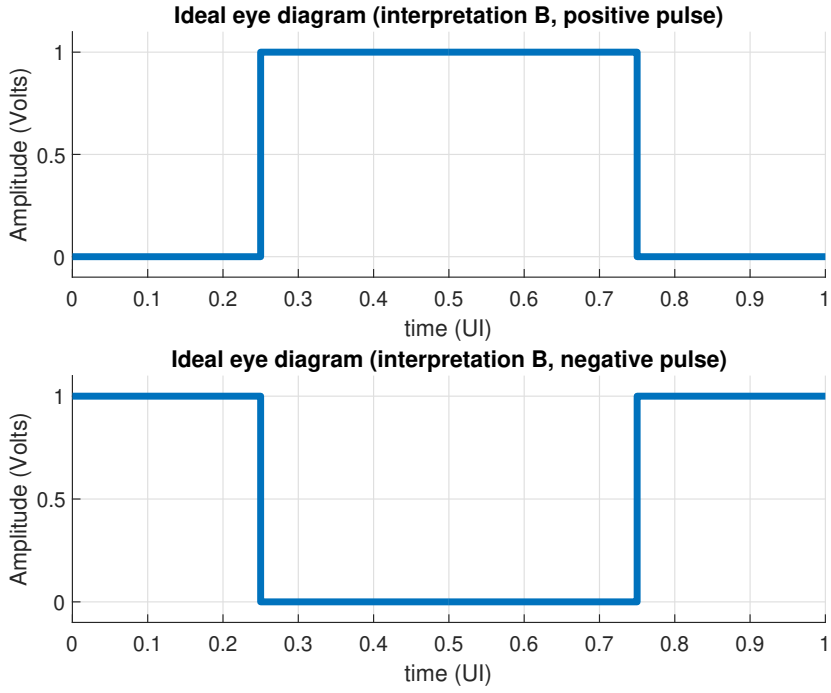


Figure 2.12. Interpretation B ideal eye diagram.

Interpretation C The zero-crossing histograms should be kept separate and the TOJ values should be calculated for each histogram. The worst case values for each measurement are the recorded results. **Interpretation C** is the equivalent of taking one lone edge and constructing an eye diagram by duplicating the edge on the left and the right.

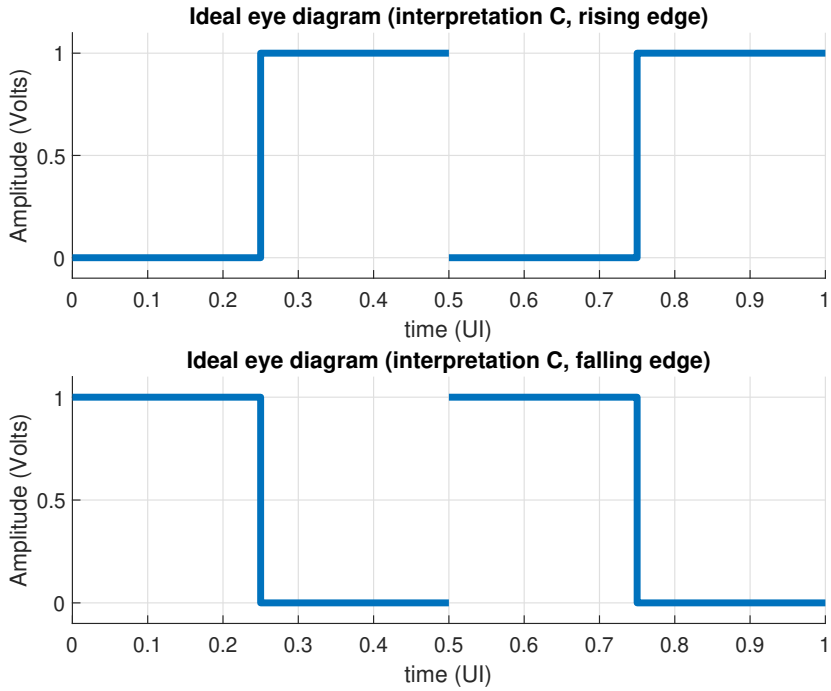


Figure 2.13. Interpretation C ideal eye diagram.

2.3.1 Comparison of interpretations

Interpretation B most accurately represents the scenario being tested. Asymmetries between rising and falling edges will result in **Interpretation C** having higher EBUJ values than **Interpretation A** and **Interpretation B**. Asymmetries between the left and right side of the of the TIE histograms present in only one edge will increase EBUJ values in **Interpretation C**. Asymmetries between rising and falling edges would also result in an increased measured DCD value.

CHAPTER 3

SETUP

3.1 Oscilloscope choice

It is possible to implement 100GBASE-CR4 TOJ tests on RT oscilloscopes. A CDR can be implemented in post processing to produce the clock signal necessary to generate zero-crossing histograms. There are two edges of interest in every PRBS9 pattern (511 bits). 766,500 symbols would have to be captured in order to collect 3,000 histogram samples. At 4 samples per UI, 8 bits per voltage sample, and 1 double floating point precision time sample this would result in a $27.6\,\mathrm{MB}$ uncompressed file. This is a reasonable file size, but the CDR code would have to be well optimized to keep the post processing computation time to a reasonable length.

ET oscilloscopes can trigger on a repeating pattern. The time range can be made very small so that the only sampled data is near the zero-crossings of interest. ET oscilloscopes are limited in how many samples per second they can store so reducing the horizontal range provides the benefit of reducing test time. The voltage scale can also be reduced to increase voltage precision. This was not done during this testing to avoid potential artifacts due to voltage rail clipping. Using an ET oscilloscope with a zoomed in time range allows for the collection of a high precision TIE histogram quickly made on the oscilloscope that is only a few kB in size and does not require post processing to reconstruct a reference clock. This reduces post processing computation and file size requirements, increases time domain precision, and potentially decreases overall test time. This technique comes at the cost of flexibility in implementing different CDRs in post processing. The exact time information of each sample is not kept, but only the zero-crossing histogram relative to the hardware CDR output is retained.

RT oscilloscopes are also capable of histogram functions. However, if the trigger is not supplied by a CDR that is locked onto the measured signal, then the measurement is not actually measuring the jitter of the device under test (DUT).

The speed of the test is limited by the trigger rate of oscilloscopes (typically less than a thousand a second). In order to directly observe a BER of 10^{-15} with a confidence of 95%, 3×10^{15} jitter samples need to be collected [12, p. 15]. With an oscilloscope with a sample fill rate of 1,000 samples/s, the tester would need to collect samples for 95,129 years. The $\delta - \delta$ model allows the jitter to be characterized with many fewer samples. The exact number of samples depends on the region of Q used to characterize RJ. In the case of 100GBASE-CR4 tests only a few thousand samples are needed, so the test time is reduced to less than a minute.

IEEE 802.3-2018 subclause 92.8.3.8 states:

> "The effect of a single-pole high-pass filter with a $3\,\mathrm{dB}$ frequency of $10\,\mathrm{MHz}$ is applied to the jitter."

This is interpreted as a specification for the CDR used to measure jitter. The effect of a high-pass filter is applied to the jitter by the ET oscilloscope's CDR OJTF.

3.2 Hardware used

Table 3.1. List of hardware.

Equipment	**Manufacturer**	**Model**	**Software version**	**Serial number**
Oscilloscope	Keysight	86108B	N/A	MY52130389
Oscilloscope mainframe	Keysight	DCA-X 86100D	A.06.01.68	MY57350354
Phase shifters	Spectrum Elektrotechnik	LS-P150-HFHM	N/A	CH1: 5306 CH2: 5307
DC blocks	Picosecond Pulse Labs	5501A	N/A	CH1: 8567 CH2: 8565
2.4mm to SMA adapter	Keysight	11901C	N/A	CH1: 20724 CH2: 35601
SMA cables	Huber+Suhner	Sucoflex 104PE	N/A	CH1: 255105016 CH2: 255105019
BERT	Anritsu	MP1900A	3.06.16	6261777850

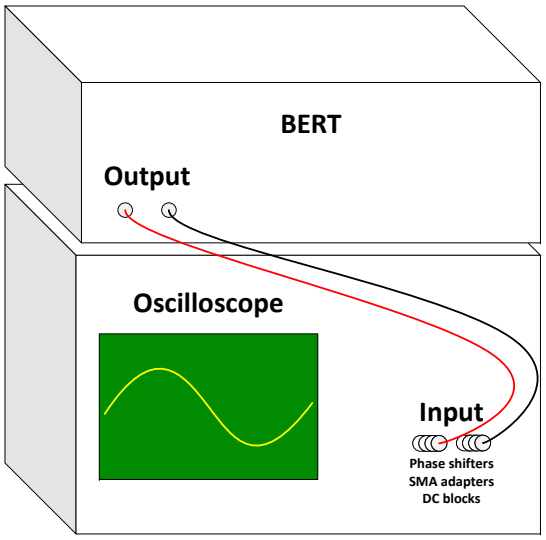


Figure 3.1. TOJ test setup diagram.

3.2.1 Explanation

A Keysight ET oscilloscope was used because it is the highest performance ET oscilloscope available to the University of New Hampshire InterOperability Laboratory (UNH-IOL) during this research. It is expected that comparable oscilloscopes from other manufacturers would yield comparable results to those in this thesis. The use of phase shifters is recommended by Keysight when measuring signals with symbol rates (SR) above 10 Gbps on the 86108B. The phase-matched SMA cables have approximately $5\,\mathrm{ps}$ of skew, which can be corrected to less than $200\,\mathrm{fs}$ of skew with phase shifters. The rated jitter noise floor of the 86108B is $< 70\,\mathrm{fs}$. DC blocks are used during 100GBASE-CR4 testing because the maximum allowed DC offset is above the oscilloscope's power rating. The DC blocks and SMA cables are rated for $26.5\,\mathrm{GHz}$ and limit the performance of the system. SMA's bandwidth limitation of $26.5\,\mathrm{GHz}$ is acceptable for this testing but $2.4\,\mathrm{mm}$ cables and DC blocks are more desirable, with bandwidth up to $50\,\mathrm{GHz}$. The Anritsu MP1900A was chosen as a signal source because it is capable of injecting specified amounts of independent jitter components at 25 Gbps. It is expected that comparable BERTs from other manufacturers would yield comparable results to those in this thesis.

According to the 86108B programmer's guide [13, p. 158] the order and type of the ET oscilloscope's built-in CDR is an underdamped 2nd order type 2 PLL and is not adjustable. This is not compliant with the 100GBASE-CR4 specification and does decrease the amount of low frequency jitter present in the measurement. The ET oscilloscope's built-in CDR has four available transition frequency settings at $f_0 = 10\,\mathrm{MHz}$ shown in Table 3.2. The 86108B programmer's guide states:

> "The Type-2 transition frequency indicates the frequency below which the second integrator in the loop starts to provide extra gain."

The transition frequency was chosen to be $880\,\mathrm{kHz}$ as a balance between OJTF peaking and rolloff. Note that the list of available transition frequency values present on the 86108B used are not the same as those listed in the 86108B programmer's guide. It is possible that other instances of this oscilloscope could have different transition frequency options than those listed in Table 3.2.

Table 3.2. 10 MHz PLL parameters reported on Keysight 86108B.

$f_{transition}$ (kHz)	Peaking (dB)
12	0
320	0.5
880	1.1
2800	2.1

Both ET and RT oscilloscopes may commonly have a jitter characterization mode that directly measures zero-crossing histograms with high sample rates. These modes often do not have a way to specify a specific edge in a pattern to measure.

3.3 Calibration values

The BERT was calibrated to meet the 100GBASE-CR4 interference tolerance test parameters in IEEE 802.3-2018 Table 92-8. A publicly available channel from the IEEE 802.3bj (100 Gb/s Backplane and Copper Cable Task Force) website was used in the Channel Operating Margin (COM) script to determine the target calibrated jitter values at a COM value of $3\,\mathrm{dB}$. No ISI generator (fitted insertion loss) or even-odd jitter was applied in measurements after this calibration was

performed. SJ was calibrated at $150\,\mathrm{MHz}$. BUJ was calibrated with a PRBS7 pattern transmitting at 12.5 Gbps. Table 3.3 is a list of the measured calibrated jitter values. Figure 3.2 is the eye diagram of the calibrated signal with all three jitter aggressors applied.

Table 3.3. Calibrated jitter component values.

jitter component	calibrated value (mUI)
RJ (RMS)	10
BUJ	35
SJ	100

These values are slightly above the limit for TOJ tests. With all three aggressors applied the applied BUJ is 135 mUI and applied TJ@BER$= 10^{-15}$ is 214 mUI. The EBUJ maximum limit is 100 mUI. The ETUJ maximum limit is 180 mUI.

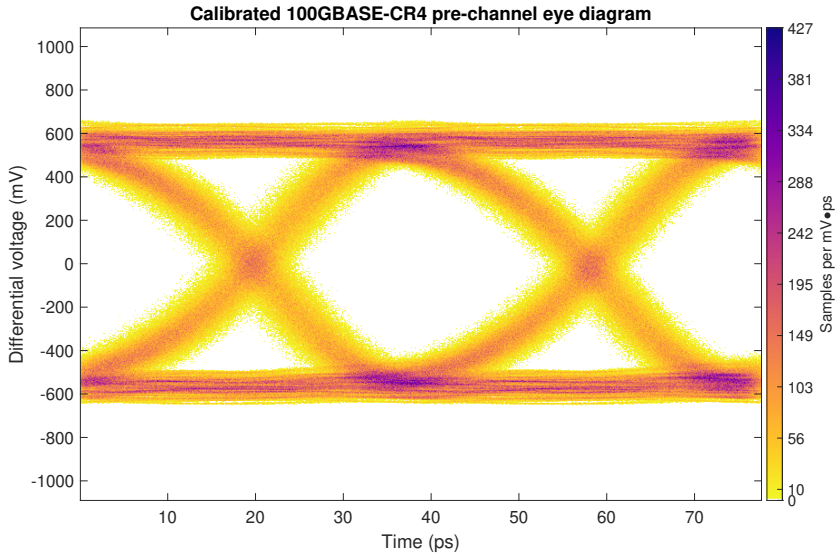


Figure 3.2. Eye diagram of a 100GBASE-CR4 calibrated stressed RX signal (pre-channel) generated by an Anritsu MP1900A and captured on a Keysight 86108B (CDR: 2$^{\text{nd}}$ order type II PLL, $f_0 = 10\,\mathrm{MHz}$, $f_{transition} = 880\,\mathrm{kHz}$).

3.4 Histogram optimization

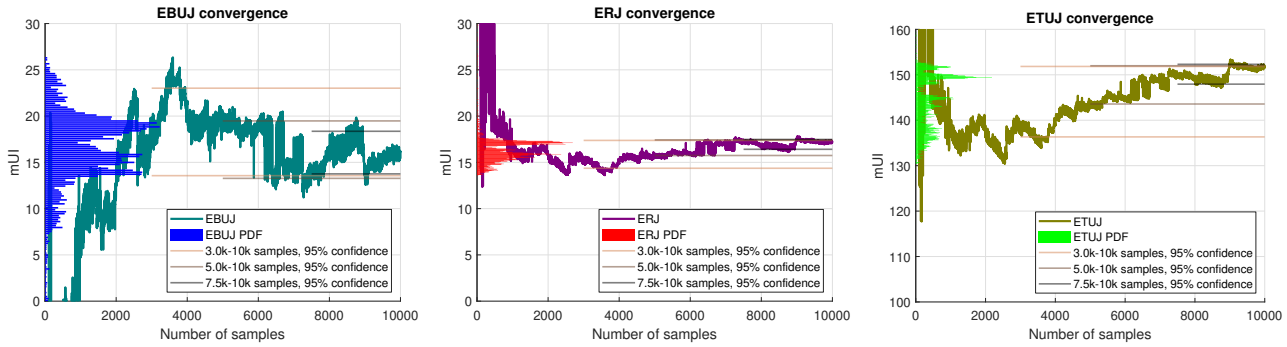
3.4.1 Number of samples

The original TOJ presentation [9, p. 10] says each zero-crossing histogram has at least 20,000 samples. A data set from a real DUT was collected during early research. This data set was used to perform a TOJ convergence analysis (Figure 3.3). The methodology of collecting histograms on ET oscilloscopes had not been discovered yet, so this data was built up from capturing many instances of the entire test pattern. A MATLAB based post processing CDR based on applying a linear regression to the count of edges over edge times was used.

The linear fit CDR works by drawing an ideal Baud line with a horizontal axis of time and a vertical axis of number of UI. Each edge time is assigned a UI value that places it closest to the ideal Baud line. A linear regression is calculated on the edge time points. The slope of this linear regression is the actual average SR. The linear regression CDR does not track low frequency jitter. The data is broken into blocks to remove low frequency jitter components. The block size was chosen to be $\frac{1}{10\,\mathrm{MHz}}$ to not have frequencies below $10\,\mathrm{MHz}$ in the OJTF. The methodology of capturing instances of the entire test pattern and applying a post processing CDR is less accurate than the direct histogram methodology but the conclusion from the convergence test applies to the direct histogram methodology.

The confidence limit bars were calculated by taking CDFs from the bottom and top of calculated jitter values between N and 10,000 samples where $N = \{3{,}000, 5{,}000, 7{,}500\}$. The jitter value where 95% of the samples fall within each CDF was used as the limit. Increasing the number of samples collected past 3,000 and 5,000 decreases the range of measured jitter values. This decreased range is desirable. However, during the data collected for this thesis contains 3,000 samples per zero-crossing histogram to keep test time to below five minutes per test case. **Interpretation A** was used in this convergence test. The horizontal axis is the number of samples in the combined edge zero-crossing histogram so the actual number of samples in the individual histograms is half as many as the horizontal axis indicates. Also note that when BUJ is very small it has been observed that EBUJ may potentially be a small negative value. This is possible with

real data even though negative jitter values cannot exist in the real world. This indicates that the mean of one of the Gaussian distributions in the $\delta - \delta$ model is negative when it is expected to be positive or vice-versa.



(a) EBUJ convergence. (b) ERJ convergence. (c) ETUJ convergence.

Figure 3.3. TOJ convergence of a real device over number of zero-crossing histogram samples (95% confidence over various sample ranges).

3.4.2 Bin size

The original TOJ presentation [9, p. 10] says that the histogram resolution shall be no coarser than $50\,\text{fs}$/bin but no finer than $5\,\text{fs}$/bin. A histogram width of 400 mUI was chosen to be able to observe histogram samples from signals with RJ values more than twice the specification limit. The horizontal resolution of the oscilloscope was set to its maximum: 65,536 points per waveform. However, histograms that the oscilloscope returns have a fixed number of bins: 751. This results in $20.7\,\text{fs}$/bin, which is in the middle of the range recommended in the original TOJ presentation.

3.5 Measurements taken

The BERT was configured to transmit PRBS9 with the calibrated applied jitter. Each of the three jitter aggressors: SJ, BUJ, and RJ, were toggled on and off on the BERT for a total of 8 different configurations. Two edges of interest, rising and falling, were captured for a total of 16 total histograms. The three interpretations were applied to each set of data. The following steps were taken to collect each histogram:

1. CDR lock and pattern lock were acquired on the ET oscilloscope.

2. Apply a fourth-order Bessel-Thomson low-pass filter to the signal as per IEEE 802.3-2018 92.8.3 using the oscilloscope built-in math functions.

3. The voltage range was set to $200\,\mathrm{mV}$/division to avoid clipping.

4. The edge of interest was located in the data pattern and placed in the display range.

5. A zero-crossing histogram was taken to determine the mean of the edge of interest relative to the pattern trigger.

6. The time scale was set to ± 200 mUI about the mean crossing time with 65,536 samples per waveform.

7. The histogram was saved after 3,000 samples had been collected.

CHAPTER 4

RESULTS

4.1 Data

The figures in this section compare values of EBUJ, ERJ, and ETUJ across the three different interpretations and eight cases of toggled jitter settings. Each column in the figures corresponds to a measured jitter value with a given interpretation. Each trio of columns corresponds to a specific jitter combination indicated at the bottom of the trio. The data is displayed in this way to make comparison of the three interpretations and identifying trends across jitter combinations easy. The full set of zero-crossing histograms is available in Appendix A. The full set of bathtub curves is available in Appendix B.

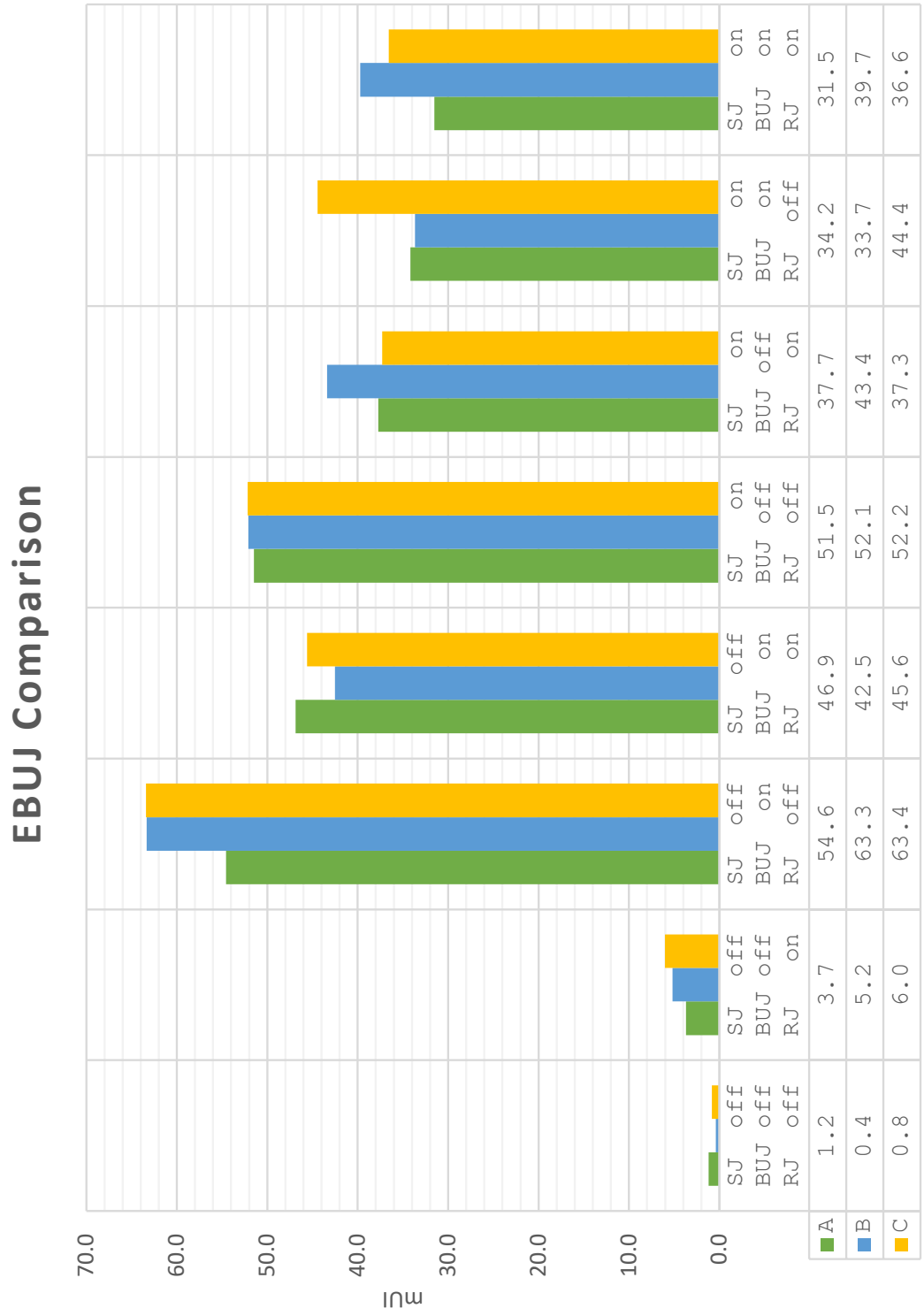


Figure 4.1. Comparison of EBUJ across combinations of jitter agressors and test definition interpretations.

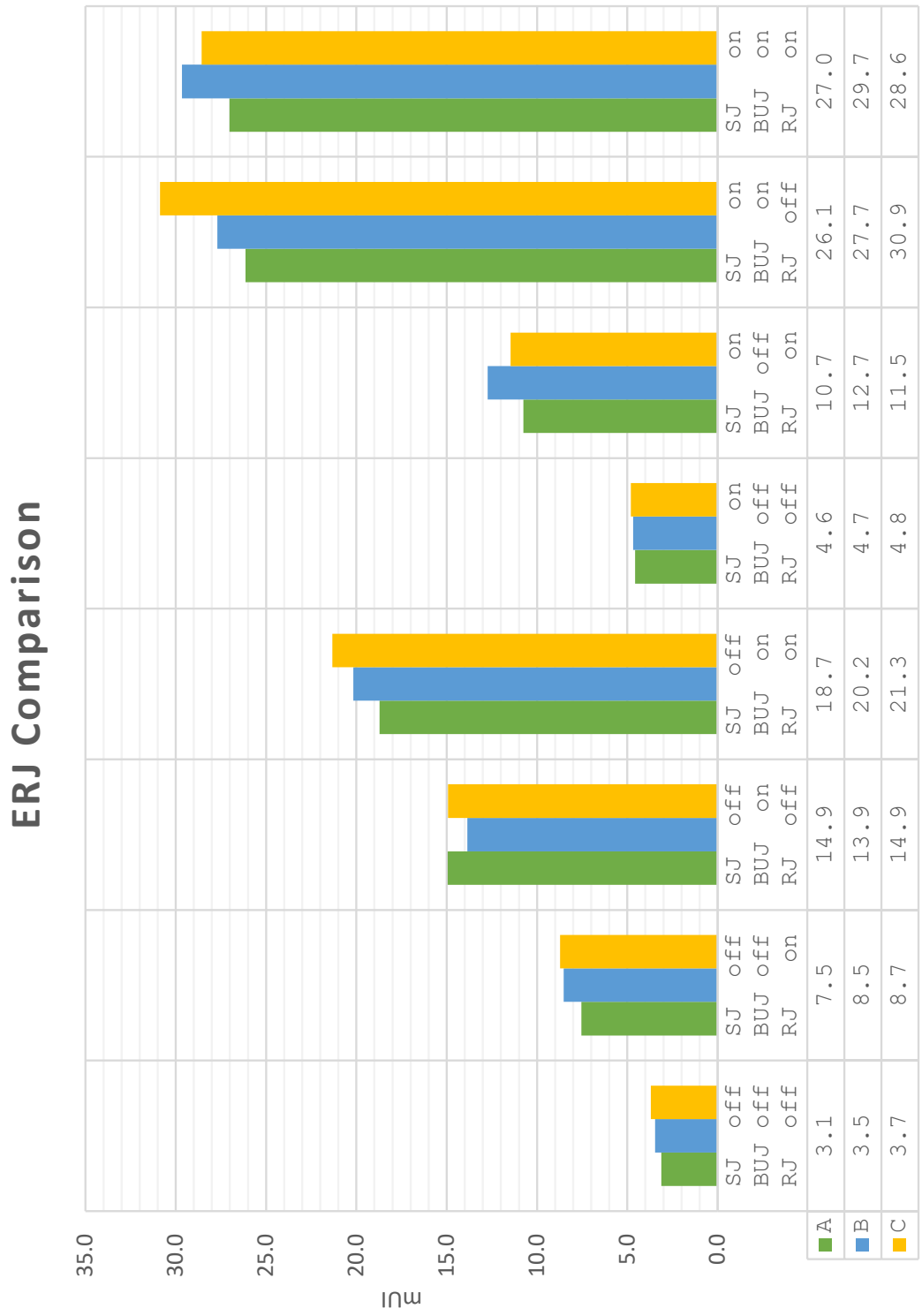


Figure 4.2. Comparison of ERJ across combinations of jitter agressors and test definition interpretations.

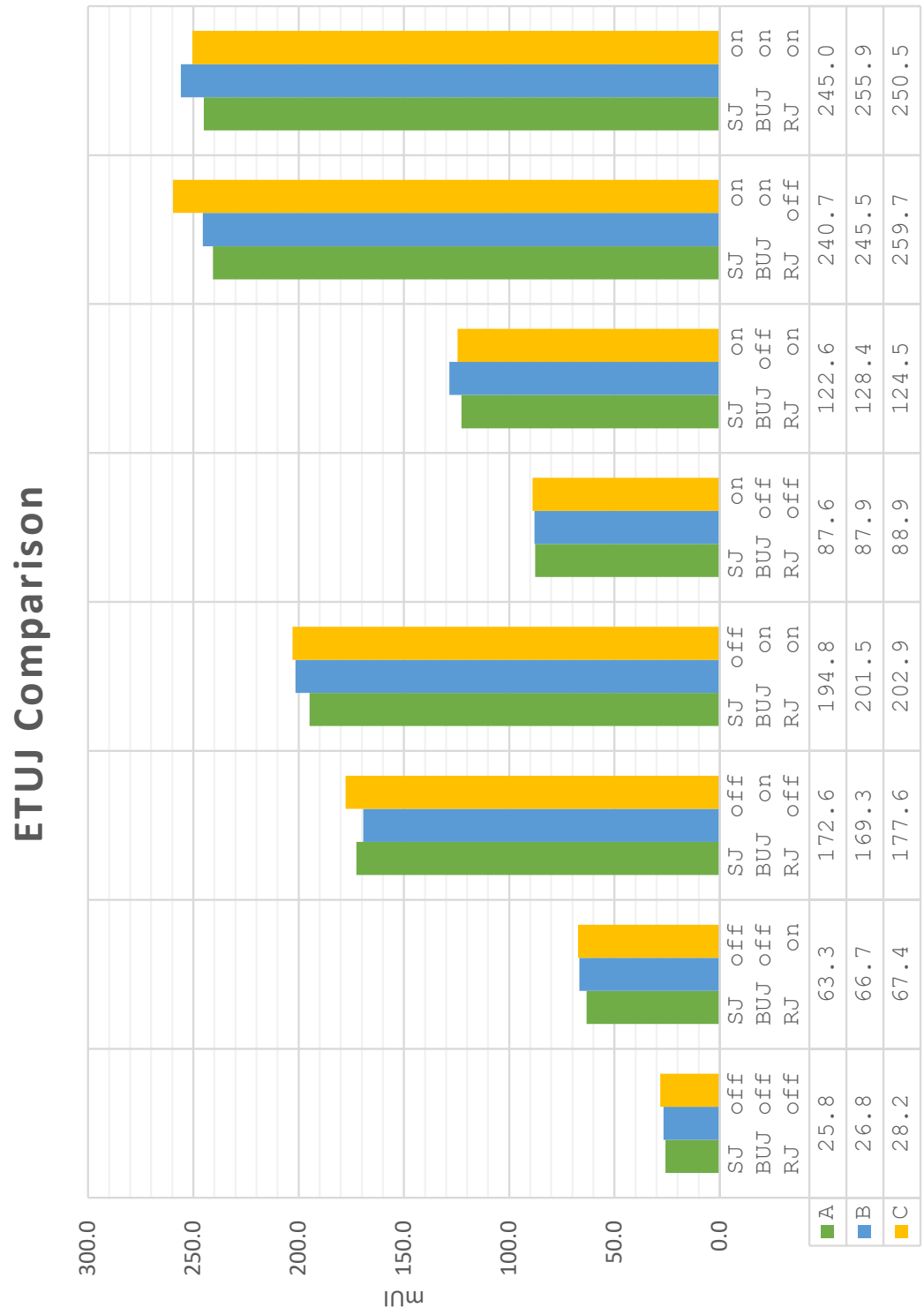
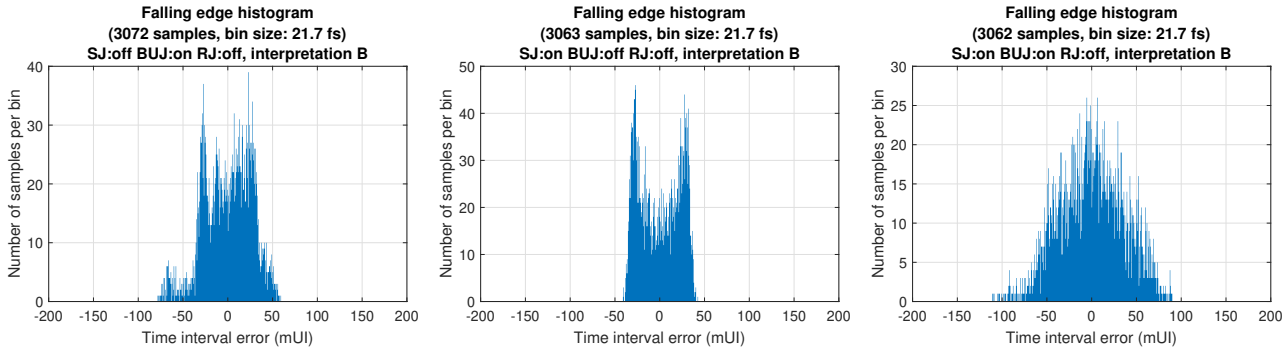


Figure 4.3. Comparison of ETUJ across combinations of jitter agressors and test definition interpretations.

4.2 Discussion

All three interpretations correlate with each other across all jitter aggressor configurations. For a given configuration of jitter aggressors all three interpretations are within 10 mUI of each other, with the largest deviations being in EBUJ. Note that SJ is a component of BUJ. EBUJ increased when SJ was applied. The $\delta - \delta$ model does not perfectly separate RJ from DJ. When SJ and BUJ were enabled, the ERJ was measured to be $10 - 20$ mUI higher than with all aggressors disabled. This observation corroborates simulation [14, p. 8].

In cases where large amounts of DJ are present (Figure A.7, Figure B.7, Figure A.8, Figure B.8) Q_{left} and Q_{right} are not symmetric. Q_{left} and Q_{right} should be symmetric if the Q linear fit region does not have DJ present because Gaussian distributions are symmetric about their mean. The presence of BUJ in the measurements taken on the rising and falling edges causes long uniform regions on the left side of their TIE histograms. The shape of the distribution that is influenced by BUJ is dependent on the data pattern used to produce the BUJ aggressor in the BERT. These uniform distribution regions cause the linear Q region to be pushed to higher values of Q than the test definition specifies. This effect results in an over-estimation of ERJ and ETUJ. The error can be quantified by taking the difference between ERJ values when RJ is disabled but DJ aggressors are toggled. In the cases examined during testing: the error is up to 26 mUI in ERJ, which results in an ETUJ error of 181 mUI. This is a significant effect and will cause false positives in test failure in DUTs with large amounts of DJ. Placing the Q linear fit region at higher values of Q would reduce this issue, but would require more samples in the zero-crossing histogram to define these regions. The amplitude of the applied BUJ is within the specified EBUJ limit. The amplitude of the applied DJ with both BUJ and SJ aggressors is 35% above the specified EBUJ limit.



(a) SJ: off, BUJ: on, RJ: off. (b) SJ: on, BUJ: off, RJ: off. (c) SJ: on, BUJ: on, RJ: off.

Figure 4.4. Falling edge zero-crossing histogram (SJ + BUJ comparison).

Figure 4.1 shows that adding both BUJ and SJ to the jitter distribution resulted in a lower EBUJ than when only SJ is present. Figure 4.4 shows a comparison of falling edge zero-crossing histograms with BUJ on and SJ off, BUJ off and SJ on, and BUJ on and SJ on. This comparison highlights the difference in jitter distribution shape between these three cases. The jitter distribution shape is more Gaussian when both deterministic jitter sources were present than when only one was present. The Dirac Delta functions of the $\delta - \delta$ model are closer to the center and the standard deviation of the Gaussian functions is larger. This effect is an example of the central limit theorem, however the deterministic components are still bounded in nature. If the Q curve was extrapolated from regions of the edge-crossing histogram that were outside of the deterministic jitter's bound, then this effect would not affect the estimated jitter values.

ERJ increased when DJ components were enabled much more than when RJ was applied. This observation is explained by the Q curve linear fit region being inside of the DJ bounds.

CHAPTER 5

CONCLUSION

5.1 Evaluation

This ET oscilloscope-based implementation satisfies IEEE 802.3-2018 subclause 92.8.3.8.2, with the exception of the JTF rolloff being 2nd order instead of 1st order. The TOJ test definition can cause pessimistic estimations of ERJ and EBUJ, and optimistic estimations of EBUJ. The magnitude of the error is dependent on the shape of the DJ distribution. The chance of this error could be reduced by changing the test definition to apply the linear fit in a different way. The range of Q values could be statically set to be higher, though this requires an exponential increase in the number of samples required to be observed to define the higher values of Q (Equation 2.7)

The estimate of 3,000 samples per histogram was made in the interest of reducing test time but pushes the limit of what is acceptable for **Interpretation B** and **Interpretation C**. 5,000 to 10,000 histogram samples would provide more confidence in scenarios where BUJ is large. It is possible that a device with large amounts of non-periodic BUJ that marginally fails ETUJ would pass an ETUJ test that used higher values of Q to characterize RJ. A failing TOJ result of a device could be a false positive indicator for interoperability issues.

An implementation based on post processing an RT oscilloscope capture would allow for implementing a CDR that is compliant to the specification regardless of hardware. This added flexibility would solve the issue of requiring an ET oscilloscope that meets the CDR requirements of the TOJ test definition. The increase in file size from kilobytes to megabytes and extra computation is not a serious concern for modern computers.

5.2 Future work

5.2.1 Post processing CDR

More advanced and efficient post processing CDR code would increase the JTF performance in RT oscilloscope-based implementations relative to windowed linear fit CDR. A high Q bandpass filter CDR [4, p. 54] is simple to implement in software. A high Q bandpass filter CDR is a bandpass filter of the received data signal with a center frequency of the SR fundamental frequency and a bandwidth of the loop filter bandwidth. For 100GBASE-CR4 this would be a bandpass filter with $Q = 1{,}289$. Issues related to high Q filtering in software are: the precision limit of double-precision floating-point, and the need to align the clock phase to the data. The clock offset could be calculated by generating an eye diagram and taking the mean of the zero crossing histogram across one UI (two edges).

More advanced linear fit CDR windowing techniques are also possible. A sliding window linear fit CDR that combines multiple results for a single clock time using a weighting function could be implemented to have tighter control over the JTF. The average symbol rate has the most error at the edges of the window, so those regions would be given the lowest weight. The width of the window would likely dictate the cutoff frequency of the JTF. The window function shape would likely influence the passband and stopband characteristics. This sliding window linear fit CDR should be explored mathematically and verified empirically to demonstrate if it is possible to generate a CDR with an arbitrary cutoff frequency and passband shape.

Periodograms could be used to empirically evaluate the JTF of post processing CDRs.

LIST OF REFERENCES

[1] R. Rabinovich, "40gb/s 100gb/s ethernet long-reach host board channel design," *IEEE Communications Magazine*, vol. 51, no. 11, pp. 152–158, Nov 2013.

[2] "IEEE standard for ethernet," *IEEE Std 802.3-2018 (Revision of IEEE Std 802.3-2015)*, pp. 1–5600, Aug 2018. [Online]. Available: https://doi.org/10.1109/IEEESTD.2018.8457469

[3] R. Neil, "Understanding jitter and wander measurements and standards," Agilent Technologies, Tech. Rep. 5988-6254EN, Feb 2003. [Online]. Available: http://literature.cdn.keysight.com/litweb/pdf/5988-6254EN.pdf

[4] M. Hsieh and G. E. Sobelman, "Architectures for multi-gigabit wire-linked clock and data recovery," *IEEE Circuits and Systems Magazine*, vol. 8, no. 4, pp. 45–57, Apr 2008.

[5] M. Schnecker, "Jitter transfer measurement in clock circuits," LeCroy Corporation, Tech. Rep., Feb 2009. [Online]. Available: http://cdn.teledynelecroy.com/files/whitepapers/designcon2009_lecroy_jitter_transfer_measurement_in_clock_circuits.pdf

[6] J. Moreira and H. Werkmann, *An Engineer's Guide to Automated Testing of High-Speed Interfaces*, 1st ed. Artech House, 2010.

[7] A. Kuo, R. Rosales, T. Farahmand, S. Tabatabaei, and A. Ivanov, "Crosstalk bounded uncorrelated jitter (BUJ) for high-speed interconnects," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 5, pp. 1800–1810, Oct 2005. [Online]. Available: https://doi.org/10.1109/tim.2005.855101

[8] P. Zivny, "Jitter and noise measurements in presence of crosstalk and 802.3bj," in *100 Gb/s Backplane and Copper Cable Task Force 2011 Chicago Meeting Materials*. IEEE, Sep 2011. [Online]. Available: http://www.ieee802.org/3/bj/public/sep11/zivny_01_0911.pdf

[9] P. Zivny and C. Moore, "802.3bj d2.1 transmitter output jitter specification for nrz pmds," in *100 Gb/s Backplane and Copper Cable Task Force 2013 Geneva Meeting Materials*. IEEE, Jul 2013. [Online]. Available: http://ieee802.org/3/bj/public/jul13/zivny_3bj_01a_0713.pdf

[10] R. Stephens, "White paper: Jitter analysis: The dual- dirac model, rj/dj, and q-scale," Agilent Technologies, Tech. Rep., Dec 2004. [Online]. Available: https://www.keysight.com/upload/cmc_upload/All/dualdirac1.pdf

[11] ——, "White paper: What the dual-dirac model is and what it is not," Tektronix, Tech. Rep. 55W-21988-0, Oct 2006. [Online]. Available: https://ransomsnotes.com/index_htm_files/RansomStephensJitter360Pt2DualDirac.pdf

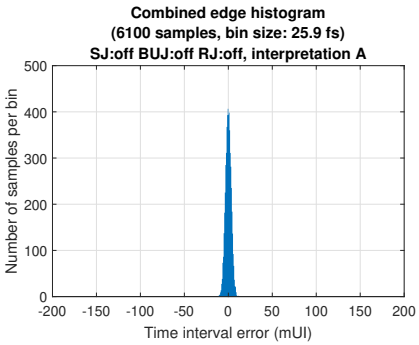
[12] A. Baldman, "Bit error ratio testing: How many bits are enough?" UNH InterOperability Lab, Tech. Rep., Mar 2003. [Online]. Available: https://www.iol.unh.edu/sites/default/files/knowledgebase/ethernet/BER-How_Many_Bits_18Mar2003.pdf

[13] *Keysight 86100A/B/C/D Wide-Bandwidth Oscilloscope Programmer's Guide*, 11th ed., Keysight Technologies, Apr 2015. [Online]. Available: https://www.keysight.com/upload/cmc_upload/All/86100_Programming_Guide.pdf

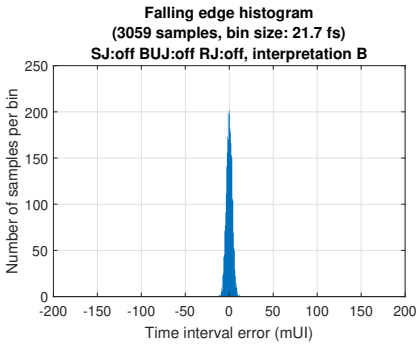
[14] C. Moore, "Experiments with simulated jitter," in *100 Gb/s Backplane and Copper Cable Task Force 2014 Indian Wells Meeting Materials*. IEEE, Jan 2014. [Online]. Available: http://www.ieee802.org/3/bj/public/jan14/moore_3bj_01_0114.pdf

APPENDIX A

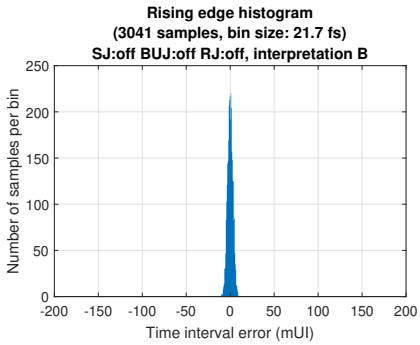
ZERO-CROSSING HISTOGRAM FIGURES



(a) Combined edge histogram.

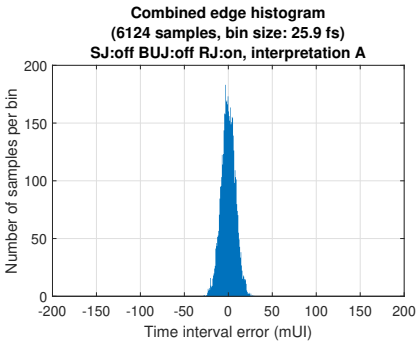


(b) Falling edge histogram.

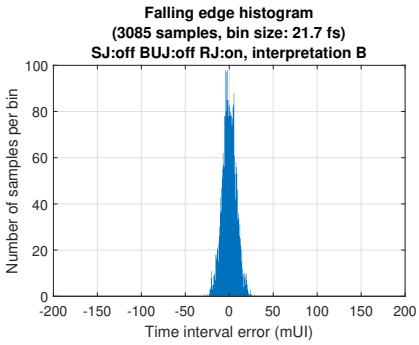


(c) Rising edge histogram.

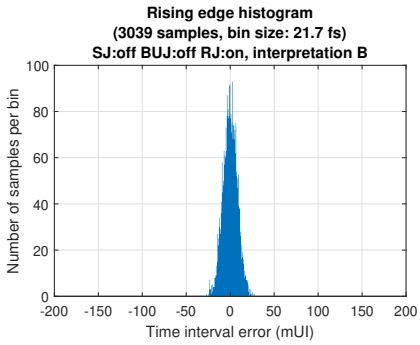
Figure A.1. Zero-crossing histogram (SJ: off, BUJ: off, RJ: off).



(a) Combined edge histogram.

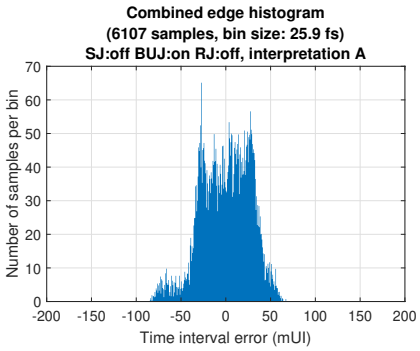


(b) Falling edge histogram.

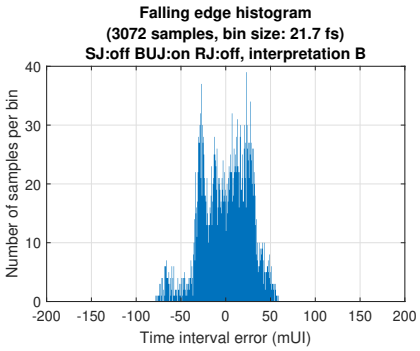


(c) Rising edge histogram.

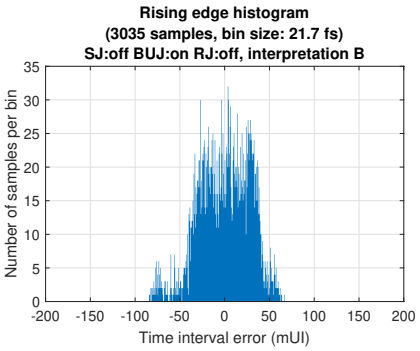
Figure A.2. Zero-crossing histogram (SJ: off, BUJ: off, RJ: on).



(a) Combined edge histogram.

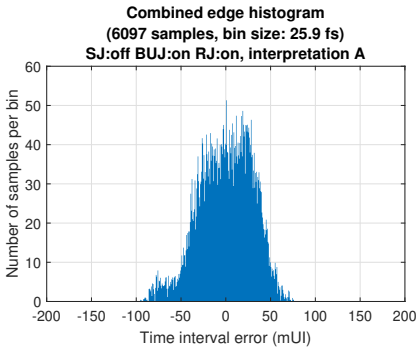


(b) Falling edge histogram.

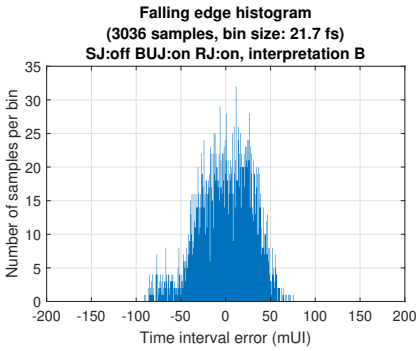


(c) Rising edge histogram.

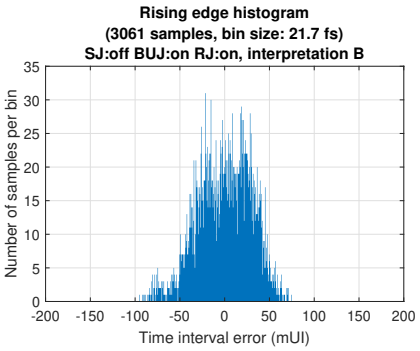
Figure A.3. Zero-crossing histogram (SJ: off, BUJ: on, RJ: off).



(a) Combined edge histogram.

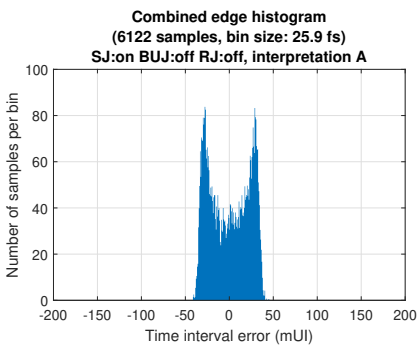


(b) Falling edge histogram.

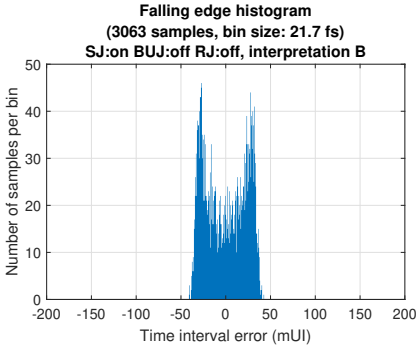


(c) Rising edge histogram.

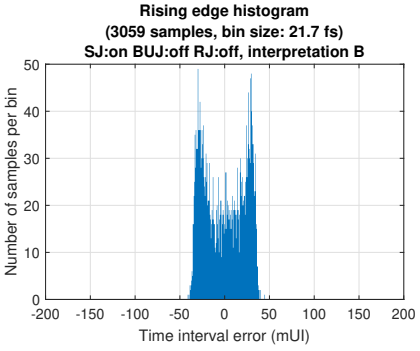
Figure A.4. Zero-crossing histogram (SJ: off, BUJ: on, RJ: on).



(a) Combined edge histogram.

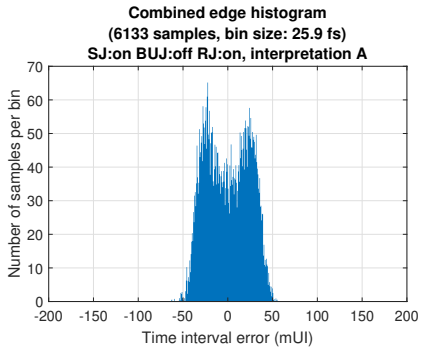


(b) Falling edge histogram.

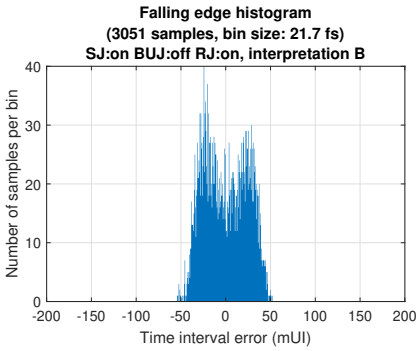


(c) Rising edge histogram.

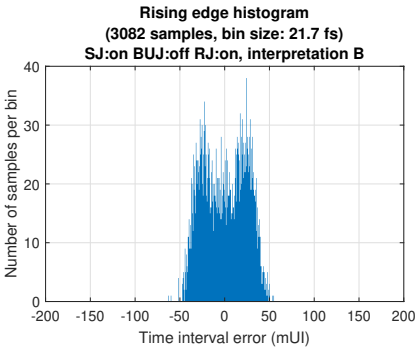
Figure A.5. Zero-crossing histogram (SJ: on, BUJ: off, RJ: off).



(a) Combined edge histogram.

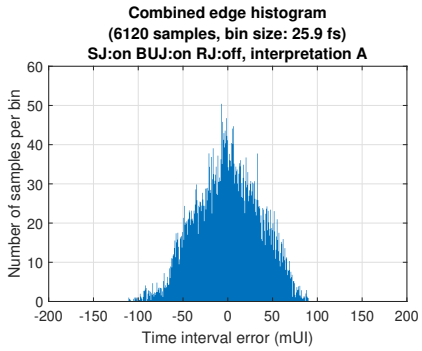


(b) Falling edge histogram.

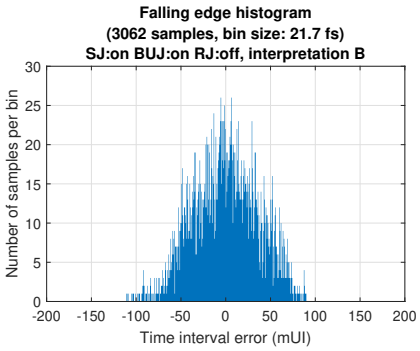


(c) Rising edge histogram.

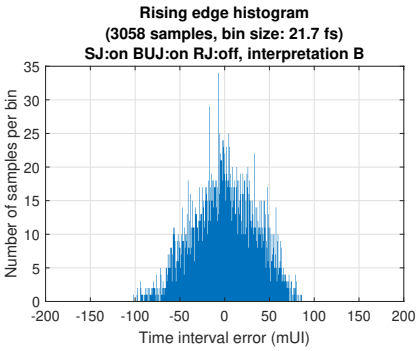
Figure A.6. Zero-crossing histogram (SJ: on, BUJ: off, RJ: on).



(a) Combined edge histogram.

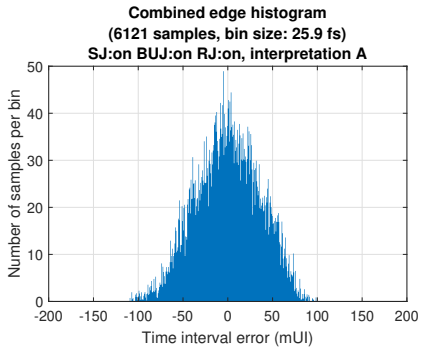


(b) Falling edge histogram.

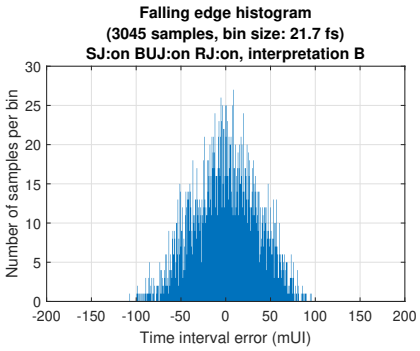


(c) Rising edge histogram.

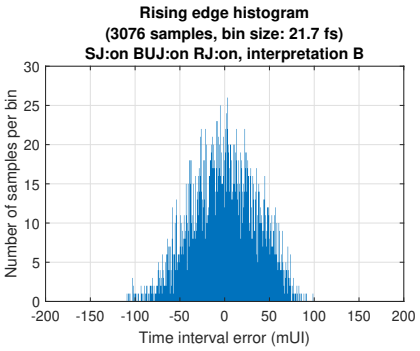
Figure A.7. Zero-crossing histogram (SJ: on, BUJ: on, RJ: off).



(a) Combined edge histogram.



(b) Falling edge histogram.

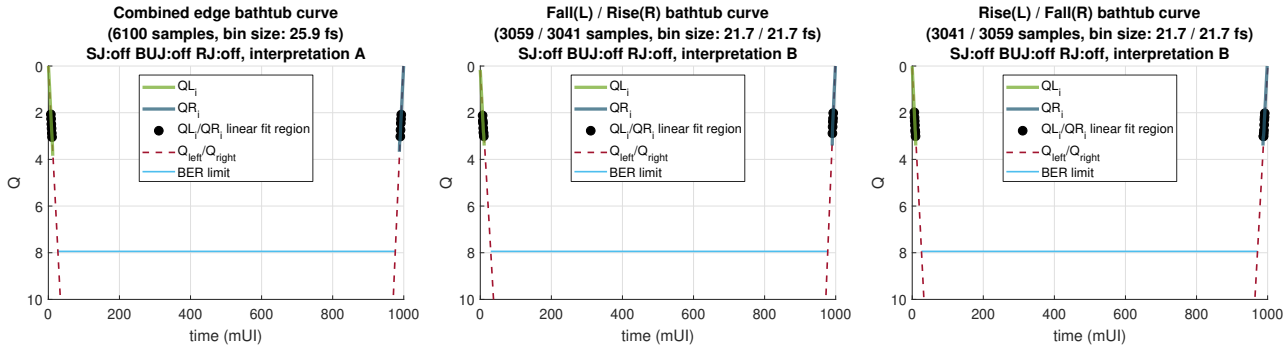


(c) Rising edge histogram.

Figure A.8. Zero-crossing histogram (SJ: on, BUJ: on, RJ: on).

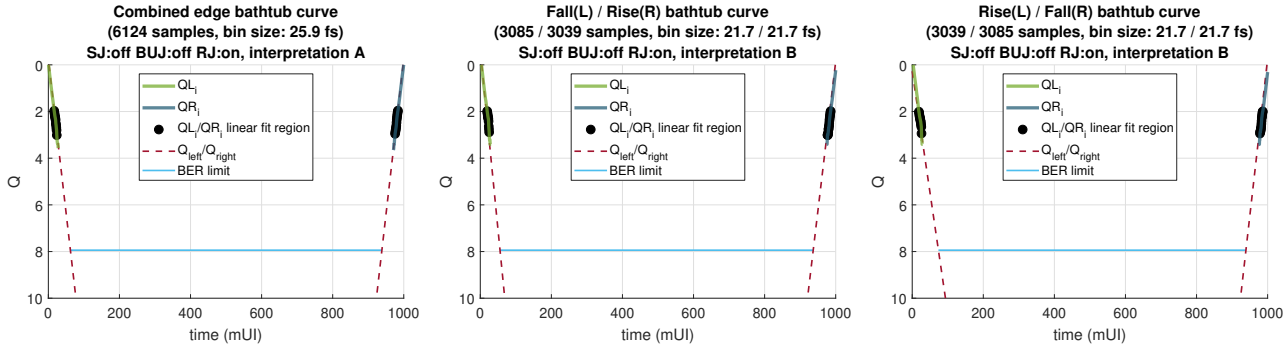
APPENDIX B

BATHTUB CURVE FIGURES



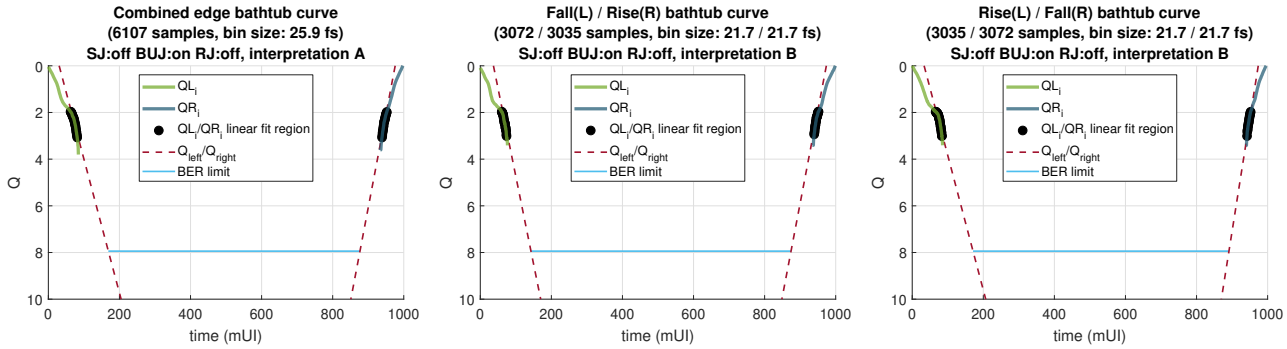
(a) Combined edge bathtub curve. (b) Fall(L) Rise(R) bathtub curve. (c) Rise(L) Fall(R) bathtub curve.

Figure B.1. Bathtub curve (SJ: off, BUJ: off, RJ: off).



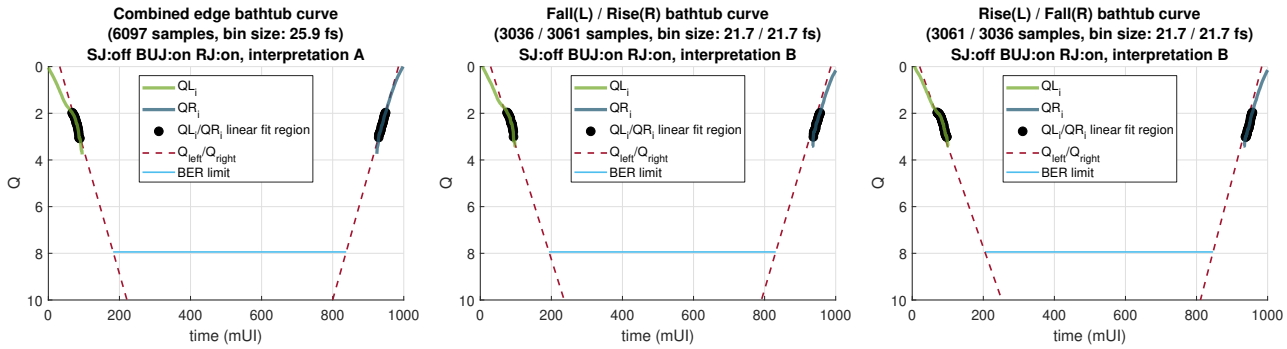
(a) Combined edge bathtub curve. (b) Fall(L) Rise(R) bathtub curve. (c) Rise(L) Fall(R) bathtub curve.

Figure B.2. Bathtub curve (SJ: off, BUJ: off, RJ: on).



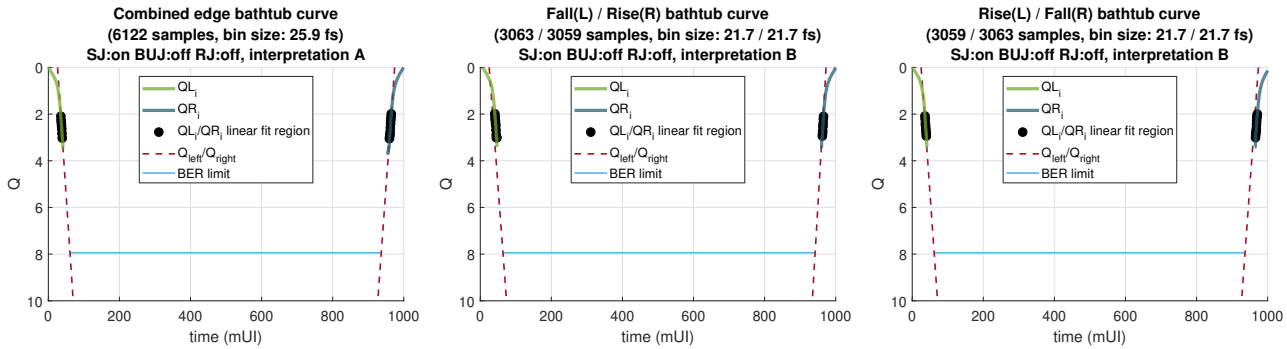
(a) Combined edge bathtub curve. (b) Fall(L) Rise(R) bathtub curve. (c) Rise(L) Fall(R) bathtub curve.

Figure B.3. Bathtub curve (SJ: off, BUJ: on, RJ: off).



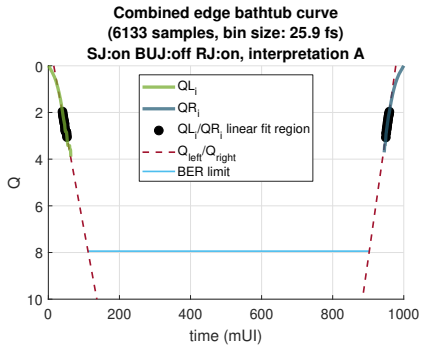
(a) Combined edge bathtub curve. (b) Fall(L) Rise(R) bathtub curve. (c) Rise(L) Fall(R) bathtub curve.

Figure B.4. Bathtub curve (SJ: off, BUJ: on, RJ: on).

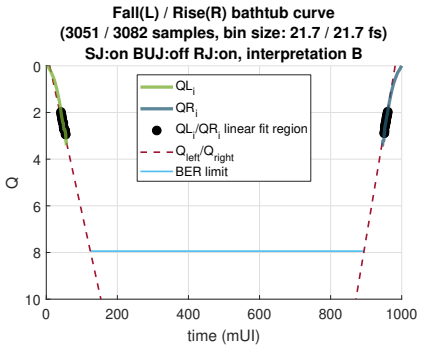


(a) Combined edge bathtub curve. (b) Fall(L) Rise(R) bathtub curve. (c) Rise(L) Fall(R) bathtub curve.

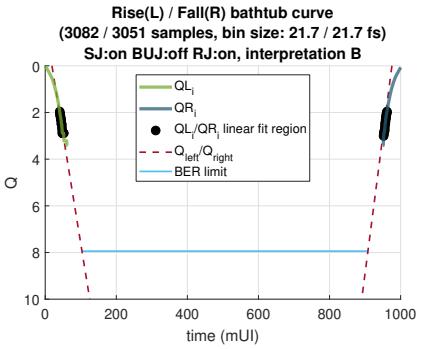
Figure B.5. Bathtub curve (SJ: on, BUJ: off, RJ: off).



(a) Combined edge bathtub curve.

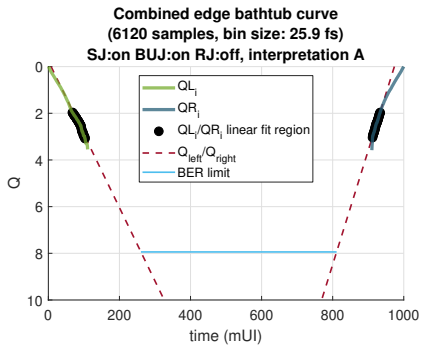


(b) Fall(L) Rise(R) bathtub curve.

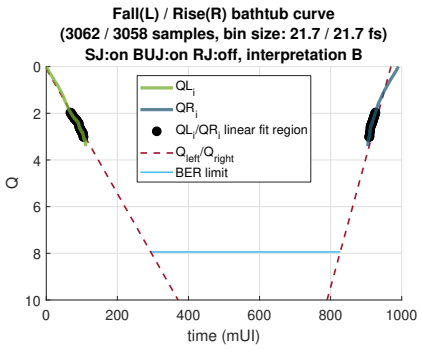


(c) Rise(L) Fall(R) bathtub curve.

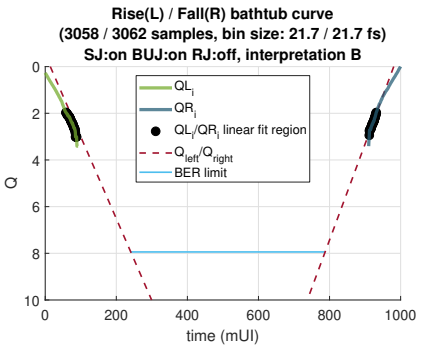
Figure B.6. Bathtub curve (SJ: on, BUJ: off, RJ: on).



(a) Combined edge bathtub curve.

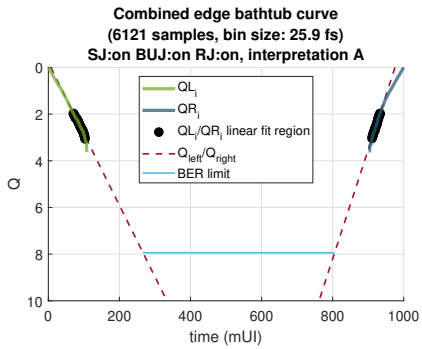


(b) Fall(L) Rise(R) bathtub curve.

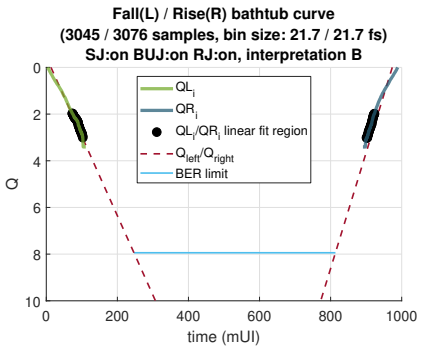


(c) Rise(L) Fall(R) bathtub curve.

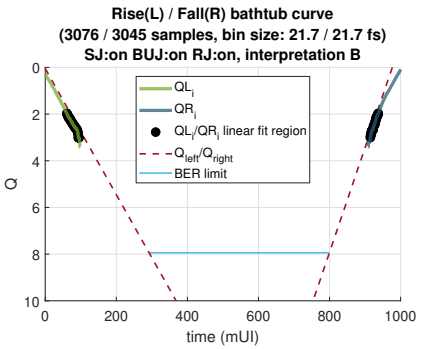
Figure B.7. Bathtub curve (SJ: on, BUJ: on, RJ: off).



(a) Combined edge bathtub curve.



(b) Fall(L) Rise(R) bathtub curve.



(c) Rise(L) Fall(R) bathtub curve.

Figure B.8. Bathtub curve (SJ: on, BUJ: on, RJ: on).

APPENDIX C

CODE

```
function [EBUJ, ERJ, ETUJ, m_left, b_left, m_right, b_right] = ...
    TIE2Jitter(TIE, bPlot, bitrate, iInterpretation)
%TIE2Jitter Calculates jitter parameters based from a TIE sequence
%           This code follows IEEE 802.3 Subclause 92.8.3.8.2
%           Effective bounded uncorrelated jitter and
%           effective random jitter
% TIE               - Time Interval Error (struct containing hist_rise and
%                     hist_fall: 2xN vectors containing time and PDF,
%                     hist_width: PDF range in UI, and
%                     NB: target number of bins if interpolating)
% doplots           - flag to enable plots
% bitrate           - bitrate (default: 25.78125E9)
% iInterpretation   - Interpretation to use (1-3, default: 3)
%                 1 - A combine falling and rising histograms
%                     interpolate into the same bins
%                 2 - B use one histogram for each edge of bathtub curve
%                 3 - C calculate TOJ for each histogram
%
%[EBUJ, ERJ, ETUJ, m_left, b_left, m_right, b_right] = ...
%    TIE2Jitter(TIE, bPlot, bitrate, iInterpretation)

if nargin < 3 || isempty(bitrate)
    bitrate = 25*66/64*1E9;
end

if nargin < 4 || isempty(iInterpretation)
    iInterpretation = 3;
elseif ~ismember(iInterpretation, 1:3)
    warning('iInterpretation must be between 1 and 3.  Defaulting to 3.');
    iInterpretation = 3;
end

iFall = 1;
iRise = 1;

% create error histogram of transitions (step a)
% TIE is a pair of histograms taken directly on an oscilloscope
switch iInterpretation
    case 1
        % Interpretation A
        hist_width = TIE.hist_width;
```

```
        NB{1} = TIE.NB;
        ti{1} = linspace(-(hist_width/2)/bitrate, ...
            (hist_width/2)/bitrate, NB{1});
        hist_fall = interp1(TIE.hist_fall(1, :), ...
            TIE.hist_fall(2, :), ti{1}, 'linear', 'extrap');
        hist_rise = interp1(TIE.hist_rise(1, :), ...
            TIE.hist_rise(2, :), ti{1}, 'linear', 'extrap');
        Ni{1} = hist_fall + hist_rise;
    case {2, 3}
        % Interpretations B and C
        iRise = 2;
        ti{iFall} = TIE.hist_fall(1,:);
        Ni{iFall} = TIE.hist_fall(2,:);
        ti{iRise} = TIE.hist_rise(1,:);
        Ni{iRise} = TIE.hist_rise(2,:);
end

strRF{iFall} = 'Fall';
strRF{iRise} = 'Rise';

nHist = length(ti);
if (iInterpretation == 2) && (nHist < 2)
    warning(['Only one histogram input.  ' ...
        'Using interpretation A instead of B.']);
    iInterpretation = 1;
end

for iHist = 1:nHist
    % convert from seconds to UI
    % this reduces numeric error in linear fit
    ti{iHist} = ti{iHist} * bitrate;

    % handle cases where the histogram doesn't cover the specified range
    ti{iHist} = ti{iHist}(~isnan(Ni{iHist})); %#ok<*AGROW>
    Ni{iHist} = Ni{iHist}(~isnan(Ni{iHist}));
    NB{iHist} = length(Ni{iHist});

    NS{iHist} = sum(Ni{iHist});

    % create CDF curves (step b)
    CDFLi{iHist} = @(i) sum(Ni{iHist}(1:i)/NS{iHist});
    CDFRi{iHist} = @(i) sum(Ni{iHist}(i:NB{iHist})/NS{iHist});
    QLi{iHist} = @(i) sqrt(2)*erfcinv(2*CDFLi{iHist}(i));
    QRi{iHist} = @(i) sqrt(2)*erfcinv(2*CDFRi{iHist}(i));

    % evaluate them over i
    CDFLi_i{iHist} = [];
```

```
    CDFRi_i{iHist} = [];
    QLi_i{iHist} = [];
    QRi_i{iHist} = [];
    for n = 1:NB{iHist}
        CDFLi_i{iHist}(n) = CDFLi{iHist}(n);
        CDFRi_i{iHist}(n) = CDFRi{iHist}(n);
        QLi_i{iHist}(n) = QLi{iHist}(n);
        QRi_i{iHist}(n) = QRi{iHist}(n);
    end

    % Determine m_left/b_left (step C)
    % Find values of i for which CDFLi/CDFRi is in the probability range
    probStart = 1.0E-3;
    probEnd = 2.5E-2;
    CDFLi_indx{iHist} = find(CDFLi_i{iHist} >= probStart & ...
        CDFLi_i{iHist} <= probEnd);
    CDFRi_indx{iHist} = find(CDFRi_i{iHist} >= probStart & ...
        CDFRi_i{iHist} <= probEnd);

    c = polyfit(ti{iHist}(CDFLi_indx{iHist}), ...
        QLi_i{iHist}(CDFLi_indx{iHist}), 1);
    m_left(iHist) = c(1);
    b_left(iHist) = c(2);

    c = polyfit(ti{iHist}(CDFRi_indx{iHist}), ...
        QRi_i{iHist}(CDFRi_indx{iHist}), 1);
    m_right(iHist) = c(1);
    b_right(iHist) = c(2);
end

strInterpretation = {'A', 'B', 'C'};
fprintf('Calculating TOJ with Interpretation %s.\n', ...
    strInterpretation{iInterpretation});

switch iInterpretation
    case {1, 2}
        % Interpretation A
        % A combine falling and rising histograms interpolate into
        % the same bins

        % Interpretation B
        % use one histogram for each edge of bathtub curve

        % try both configurations in Interpretation B
        if iInterpretation == 2
            iLR = {[iFall, iRise], [iRise, iFall]};
        else
```

```
            iLR = {[1, 1]};
        end

        for i = 1:length(iLR)
            m_leftTmp = m_left(iLR{i}(1));
            b_leftTmp = b_left(iLR{i}(1));
            m_rightTmp = m_right(iLR{i}(2));
            b_rightTmp = b_right(iLR{i}(2));

            % determine effective bounded uncorrelated jitter and
            % effective total uncorrelated jitter (step D)
            EBUJ(i) = (b_leftTmp / m_leftTmp) - ...
                (b_rightTmp / m_rightTmp);
            ERJ(i)  = (m_leftTmp - m_rightTmp) / ...
                (2 * m_rightTmp * m_leftTmp);
            ETUJ(i) = (7.9 * ERJ(i)) + EBUJ(i);
        end

        [~,iEBUJMax] = max(EBUJ);
        [~,iERJMax] = max(ERJ);
        [~,iETUJMax] = max(ETUJ);

        if iInterpretation == 2
            fprintf('EBUJ: Using %s(L) / %s(R) (worst case)\n', ...
                strRF{iLR{iEBUJMax}(1)}, strRF{iLR{iEBUJMax}(2)});
            fprintf('ERJ:  Using %s(L) / %s(R) (worst case)\n', ...
                strRF{iLR{iEBUJMax}(1)}, strRF{iLR{iEBUJMax}(2)});
            fprintf('ETUJ: Using %s(L) / %s(R) (worst case)\n', ...
                strRF{iLR{iEBUJMax}(1)}, strRF{iLR{iEBUJMax}(2)});
        end
    case 3
        % Interpretation C
        % calculate TOJ for each histogram
        for iHist = 1:nHist
            EBUJ(iHist) = (b_left(iHist) / m_left(iHist)) - ...
                (b_right(iHist) / m_right(iHist));
            ERJ(iHist) = (m_left(iHist) - m_right(iHist)) / ...
                (2 * m_right(iHist) * m_left(iHist));
            ETUJ(iHist) = (7.9 * ERJ(iHist)) + EBUJ(iHist);
        end
        [~,iEBUJMax] = max(EBUJ);
        [~,iERJMax] = max(ERJ);
        [~,iETUJMax] = max(ETUJ);

        fprintf('EBUJ: Using %s histogram (worst case)\n', ...
            strRF{iEBUJMax});
        fprintf('ERJ:  Using %s histogram (worst case)\n', ...
```

```
            strRF{iERJMax });
        fprintf('ETUJ: Using %s histogram (worst case)\n', ...
            strRF{iETUJMax});
end

% assign worst case values
if exist('iEBUJMax', 'var')
    EBUJ = EBUJ(iEBUJMax);
    ERJ = ERJ(iERJMax);
    ETUJ = ETUJ(iETUJMax);

    if iscell(m_left)
        m_left = m_left{iERJMax};
        b_left = b_left{iERJMax};
        m_right = m_right{iERJMax};
        m_right = m_right{iERJMax};
    end
end

% print results
fprintf('EBUJ: %6.2f mUI\n', EBUJ * 1E3);
fprintf('ERJ:  %6.2f mUI\n', ERJ * 1E3);
fprintf('ETUJ: %6.2f mUI\n', ETUJ * 1E3);

if bPlot
    sizeBinfs1 = mean(diff(ti{1})) / bitrate * 1E15;
    sizeBinfsFall = mean(diff(ti{iFall})) / bitrate * 1E15;
    sizeBinfsRise = mean(diff(ti{iRise})) / bitrate * 1E15;
    switch iInterpretation
        case 1
            figure;
            nHits = sum([TIE.hist_fall(2,:), TIE.hist_rise(2,:)]);
            plotErrorHistogram(ti{1}, Ni{1});
            title(sprintf(['Combined edge histogram\n' ...
                '(%d samples, bin size: %.1f fs)'], ...
                nHits, sizeBinfs1));

            figure;
            plotQCurves(ti{1}, ti{1}, QLi_i{1}, QRi_i{1}, ...
                CDFLi_indx{1}, CDFRi_indx{1}, ...
                m_left, b_left, m_right, b_right);
            title(sprintf(['Combined edge bathtub curve\n' ...
                '(%d samples, bin size: %.1f fs)'], ...
                nHits, sizeBinfs1));
        case 2
            figure;
            plotErrorHistogram(ti{iFall}, Ni{iFall});
```

```
230             title(sprintf(['Falling edge histogram\n' ...
231                 '(%d samples, bin size: %.1f fs)'], ...
232                 sum(Ni{iFall}), sizeBinfsFall));
233 
234             figure;
235             plotErrorHistogram(ti{iRise}, Ni{iRise});
236             title(sprintf(['Rising edge histogram\n' ...
237                 '(%d samples, bin size: %.1f fs)'], ...
238                 sum(Ni{iRise}), sizeBinfsRise));
239 
240             figure;
241             plotQCurves(ti{iFall}, ti{iRise}, QLi_i{iFall}, ...
242                 QRi_i{iRise}, CDFLi_indx{iFall}, CDFRi_indx{iRise}, ...
243                 m_left(iFall), b_left(iFall), ...
244                 m_right(iRise), b_right(iRise));
245             title(sprintf(['%s(L) / %s(R) bathtub curve\n' ...
246                 '(%d / %d samples, bin size: %.1f / %.1f fs)'], ...
247                 strRF{iFall}, strRF{iRise}, ...
248                 sum(Ni{iFall}), sum(Ni{iRise}), ...
249                 sizeBinfsFall, sizeBinfsRise));
250 
251             figure;
252             plotQCurves(ti{iRise}, ti{iFall}, QLi_i{iRise}, ...
253                 QRi_i{iFall}, CDFLi_indx{iRise}, CDFRi_indx{iFall}, ...
254                 m_left(iRise), b_left(iRise), ...
255                 m_right(iFall), b_right(iFall));
256             title(sprintf(['%s(L) / %s(R) bathtub curve\n' ...
257                 '(%d / %d samples, bin size: %.1f / %.1f fs)'], ...
258                 strRF{iRise}, strRF{iFall}, ...
259                 sum(Ni{iRise}), sum(Ni{iFall}), ...
260                 sizeBinfsRise, sizeBinfsFall));
261         case 3
262             figure;
263             plotErrorHistogram(ti{iFall}, Ni{iFall});
264             title(sprintf(['Falling edge histogram\n' ...
265                 '(%d samples, bin size: %.1f fs)'], ...
266                 sum(Ni{iFall}), sizeBinfsFall));
267 
268             figure;
269             plotErrorHistogram(ti{iRise}, Ni{iRise});
270             title(sprintf(['Rising edge histogram\n' ...
271                 '(%d samples, bin size: %.1f fs)'], ...
272                 sum(Ni{iRise}), sizeBinfsRise));
273 
274             figure;
275             plotQCurves(ti{iFall}, ti{iFall}, QLi_i{iFall}, ...
276                 QRi_i{iFall}, CDFLi_indx{iFall}, CDFRi_indx{iFall}, ...
```

```
                m_left(iFall), b_left(iFall), ...
                m_right(iRise), b_right(iRise));
            title(sprintf(['Falling edge bathtub curve\n' ...
                '(%d samples, bin size: %.1f fs)'], ...
                sum(Ni{iFall}), sizeBinfsFall));

            figure;
            plotQCurves(ti{iRise}, ti{iRise}, QLi_i{iRise}, ...
                QRi_i{iRise}, CDFLi_indx{iRise}, CDFRi_indx{iRise}, ...
                m_left(iRise), b_left(iRise), ...
                m_right(iFall), b_right(iFall));
            title(sprintf(['Rising edge bathtub curve\n' ...
                '(%d samples, bin size: %.1f fs)'], ...
                sum(Ni{iRise}), sizeBinfsRise));
    end
    drawnow;
end

return;


function plotQCurves(tiL, tiR, QLi, QRi, CDFLi_indx, CDFRi_indx, ...
    m_left, b_left, m_right, b_right)

ui_nom = 1;

tiL = fliplr(tiL);
tiR = fliplr(tiR);

% convert to milli UI
xL = tiL * 1E3;
xLFit = tiL(CDFLi_indx) * 1E3;
xR = (tiR + ui_nom) * 1E3;
xRFit = (tiR(CDFRi_indx) + ui_nom) * 1E3;

% set plot bounds
xBound = [-0.0*ui_nom, ui_nom*1.0] * 1E3;
yBound = [0, 10];

% set plot colors
cExtrap = [0.635, 0.078, 0.184, 1.000];
cLimit = [0.301, 0.745, 0.933, 1.000];
cFitRegion = [0, 0, 0, 0.5];
cQLi = [1.0*[0.466, 0.674, 0.188], 0.750];
cQRi = [0.5*[0.301, 0.745, 0.933], 0.750];

hold on;
```

```
set(gca, 'YDir', 'reverse');
% plot the BER limit
yLimit = sqrt(2)*erfcinv(2*(1E-15));
xLimit(1) = -((yLimit - b_left) / m_left);
xLimit(2) = 1 - ((yLimit - b_right) / m_right);
hp_lim = plot(xLimit*1E3, [1, 1] * yLimit, ...
    '-', 'LineWidth', 1.5, 'Color', cLimit);
% plot the linear extrapolation of Q
hp_QLe = plot(xBound, -(xBound .* m_left * 1E-3) + b_left, ...
    '--', 'LineWidth', 1.5, 'Color', cExtrap); %#ok<*NASGU>
hp_QRe = plot(fliplr(xBound), (xBound .* m_right * 1E-3) + b_right, ...
    '--', 'LineWidth', 1.5, 'Color', cExtrap);
% plot portion of Q that the linear fit was calculated on
hp_QLi_fitted = plot(xLFit, QLi(CDFLi_indx), '.', ...
    'MarkerSize', 6*3*1.5, 'Color', cFitRegion);
hp_QRi_fitted = plot(xRFit, QRi(CDFRi_indx), '.', ...
    'MarkerSize', 6*3*1.5, 'Color', cFitRegion);
% plot the actual Q curves
hp_QLi = plot(xL, QLi, 'Color', cQLi, 'LineWidth', 3);
hp_QRi = plot(xR, QRi, 'Color', cQRi, 'LineWidth', 3);


legend([hp_QLi, hp_QRi, hp_QLi_fitted, hp_QLe, hp_lim], ...
    {'QL_i', 'QR_i', 'QL_i/QR_i linear fit region', ...
    'Q_{left}/Q_{right}', 'BER limit'}, 'Location', 'North');
xlabel('time (mUI)');
ylabel('Q');
% set axis limits
ax = [xBound, yBound];
axis(ax);
grid on;
return;


function plotErrorHistogram(ti, Ni)

bar(ti * 1E3, Ni, 'BarWidth', 1);
title(sprintf('Error Histogram (%d samples)', round(sum(Ni))));
ylabel('Number of samples per bin');
xlabel('Time interval error (mUI)');
grid on;
return;
```