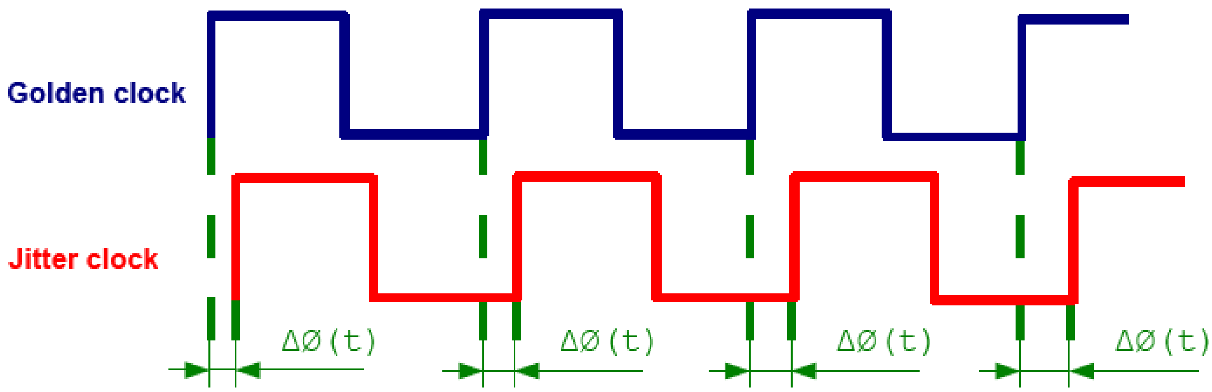
Outline

- Jitter definition - short vs long term(TIE)
- Jitter category - RJ, DJ and details
- Review of statistics - PDF, CDF, erf, erfc
- BERT vs Scope
- Dual Dirac Model - DJ_{pp} vs $DJ_{\delta\delta}$
- Decompose method - Spectrum vs Tail Fit
- Scope Jitter setting

TIE



- Deviation of the digital timing event from it is ideal position
- A specified number of observations.

Jitter definition

- Period Jitter (J_{PER})
  - Time difference between measured period and ideal period
- Cycle to Cycle Jitter (J_{CC})
  - Time difference between two adjacent clock periods
  - Important for budgeting on-chip digital circuits cycle time
- Accumulated Jitter (J_{AC})
  - Time difference between measured clock and ideal trigger clock
  - Jitter measurement most relative to high-speed link systems

Jitte Analysis Method

- Time domain
- Frequency domain

- Statistics domain
- Decompose to RJ and DJ

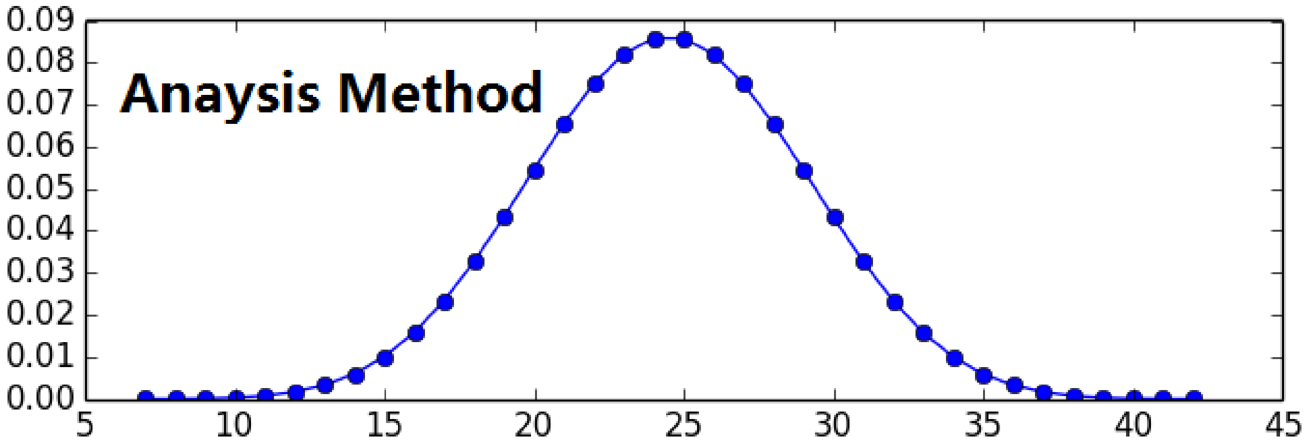
Monte Carlo

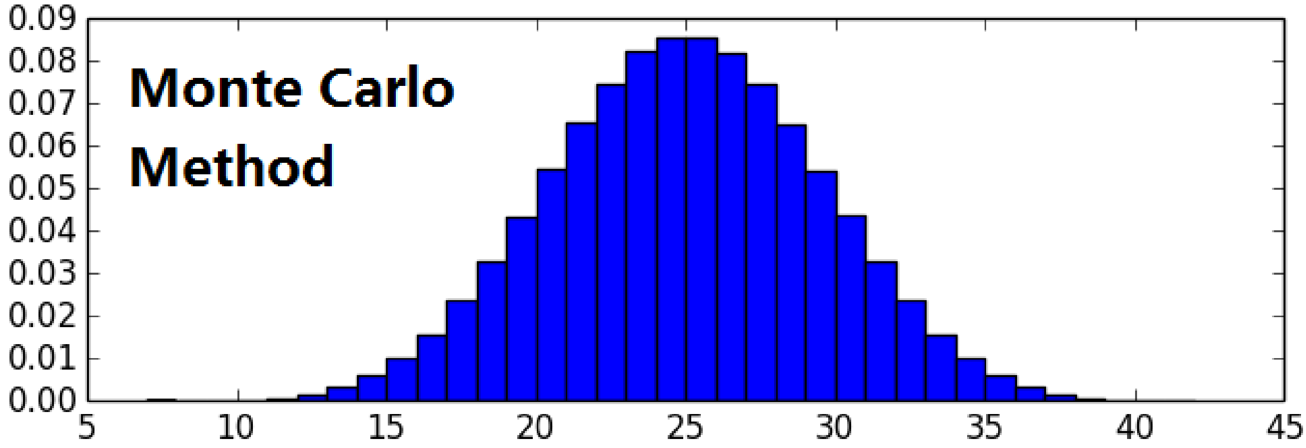
- Monte Carlo Simulation is a way of studying probability distributions with sampling
- Define a domain of possible inputs.
- Generate inputs randomly from a probability distribution over the domain.
- Perform a deterministic computation on the inputs.
- Aggregate the results.

Dice gambling

- 7 dices , 6 faces.
- summarize the 7 dices points
- what is the PDF?

Dice PDF





Code

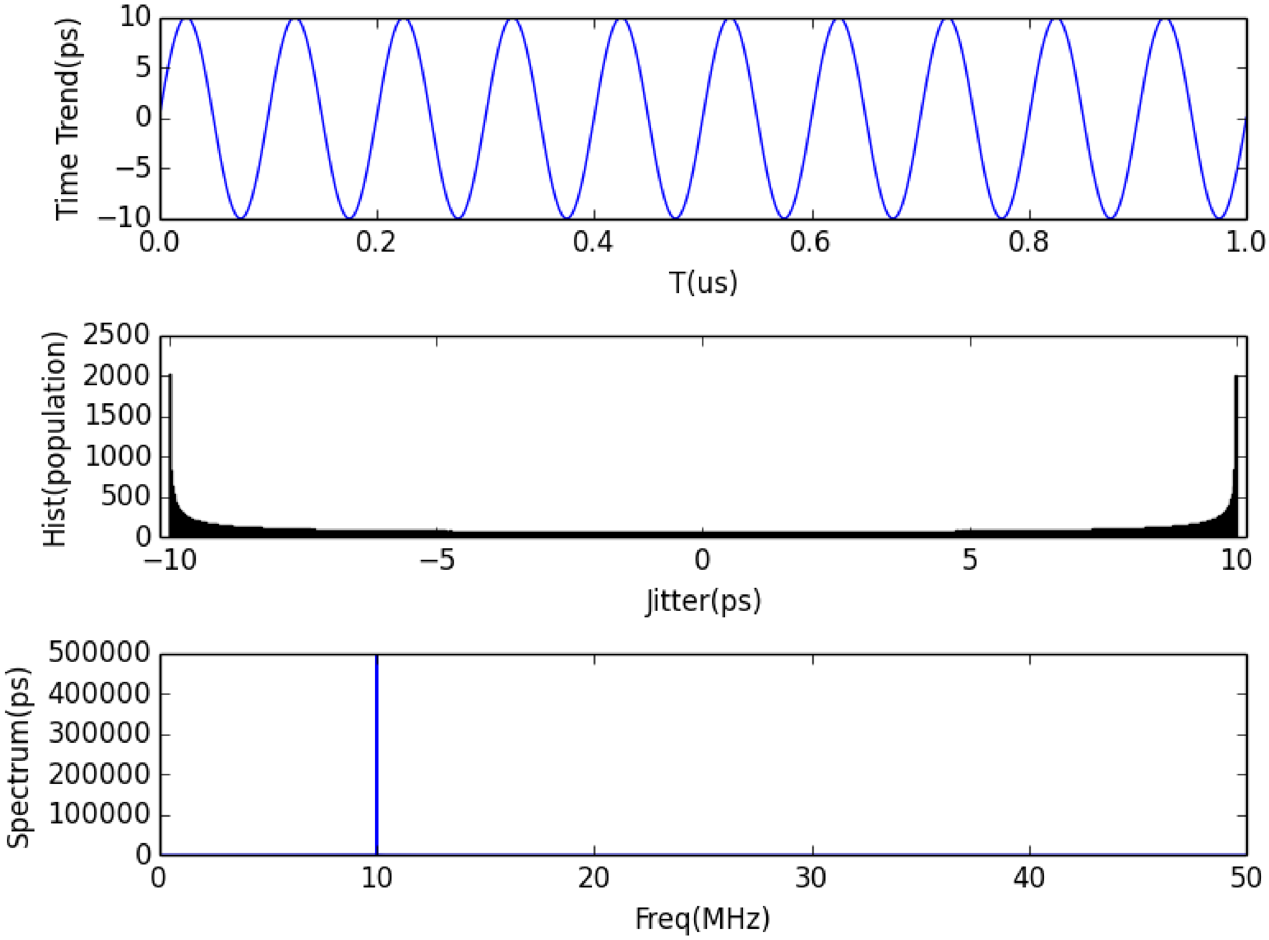
```
import numpy as np
from numpy.random import random_integers
import matplotlib.pyplot as plt

sample_len = 1000000
res = np.zeros(sample_len)
for i in range(sample_len):
    res[i] = np.sum(random_integers(1,6,7))
plt.hist(res,bins=35,normed=True)
plt.show()
```

Histogram(Simulation)

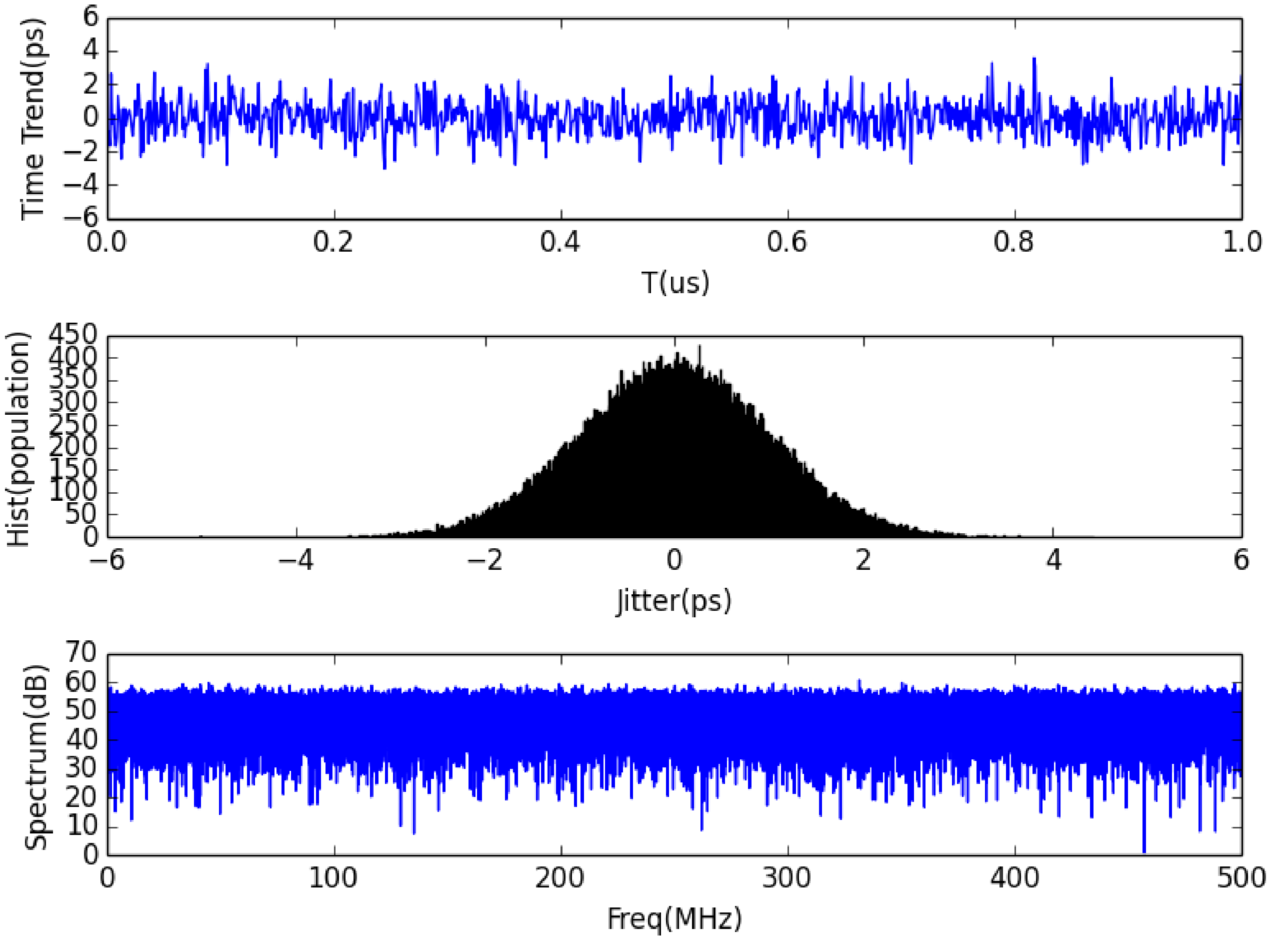
```python
import numpy as np
from numpy.fft import rfft, rfftfreq
from numpy.random import normal
import matplotlib.pyplot as plt

PJ_freq = 10e6 # 10MHz
PJ_amp = 10 # 10ps
RJ_rms = 1 # 1ps
sample_rate = 1e9 #1Gbps
sample_interval = 1./sample_rate
N_cycle =1000
pts = sample_rate/PJ_freq*N_cycle


t = sample_interval*np.arange(pts)*1E6

# sin jitter
plt.figure()
tie_sin = PJ_amp*np.sin(np.linspace(0,2*np.pi*N_cycle,sample_rate/PJ_freq*N_cycle))
plt.subplot(311)
plt.plot(t,tie_sin)
plt.xlabel("T(us)")
plt.ylabel("Time Trend(ps)")
plt.xlim([0,1]) # only plot 1us
plt.subplot(312)
plt.hist(tie_sin,bins=1000,normed=False)
plt.xlabel("Jitter(ps)")
plt.ylabel("Hist(population)")
plt.xlim([-10.2,10.2])
plt.subplot(313)
plt.plot(rfftfreq(len(tie_sin),1/sample_rate)/1e6,np.abs(rfft(tie_sin)))
plt.xlabel("Freq(MHz)")
plt.ylabel("Spectrum(ps)")
plt.xlim([0,50])
plt.tight_layout()

### random jitter
plt.figure()
tie_normal = normal(loc=0,scale=RJ_rms,size=pts)
plt.subplot(311)
plt.plot(t,tie_normal)
plt.xlabel("T(us)")
plt.ylabel("Time Trend(ps)")
plt.xlim([0,1]) # only plot 1us
plt.subplot(312)
plt.hist(tie_normal,bins=1000,normed=False)
plt.xlabel("Jitter(ps)")
plt.ylabel("Hist(population)")
plt.subplot(313)
plt.plot(rfftfreq(len(tie_normal),1/sample_rate)/1e6,20*np.log10(np.abs(rfft(tie_normal))))
plt.xlabel("Freq(MHz)")
plt.ylabel("Spectrum(dB)")

plt.tight_layout()
```

```
#Combined jitter
plt.figure()
tie_combine = tie_sin+tie_normal
plt.subplot(311)
plt.plot(t,tie_combine)
plt.xlabel("T(us)")
plt.ylabel("Time Trend(ps)")
plt.xlim([0,1]) # only plot 1us
plt.subplot(312)
plt.hist(tie_combine,bins=1000,normed=False)
plt.xlabel("Jitter(ps)")
plt.ylabel("Hist(population)")
plt.subplot(313)
plt.plot(rfftfreq(len(tie_combine),1/sample_rate)/1e6,20*np.log10(np.abs(rfft(tie_combine))))
plt.xlabel("Freq(MHz)")
plt.ylabel("Spectrum(dB)")
plt.xlim([0,50])
plt.tight_layout()


plt.show()
```

Sinusoid

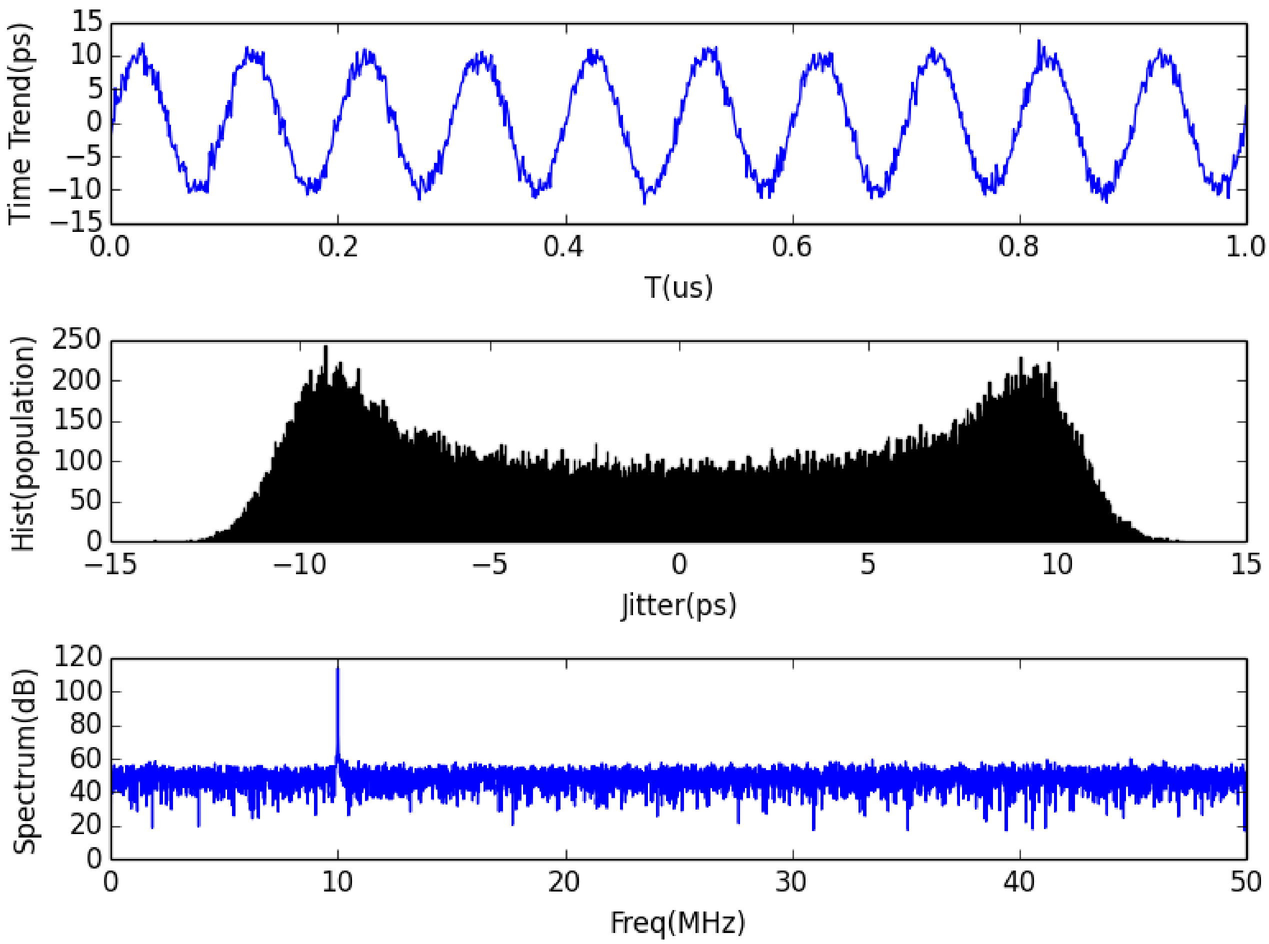


Random(Gaussian)

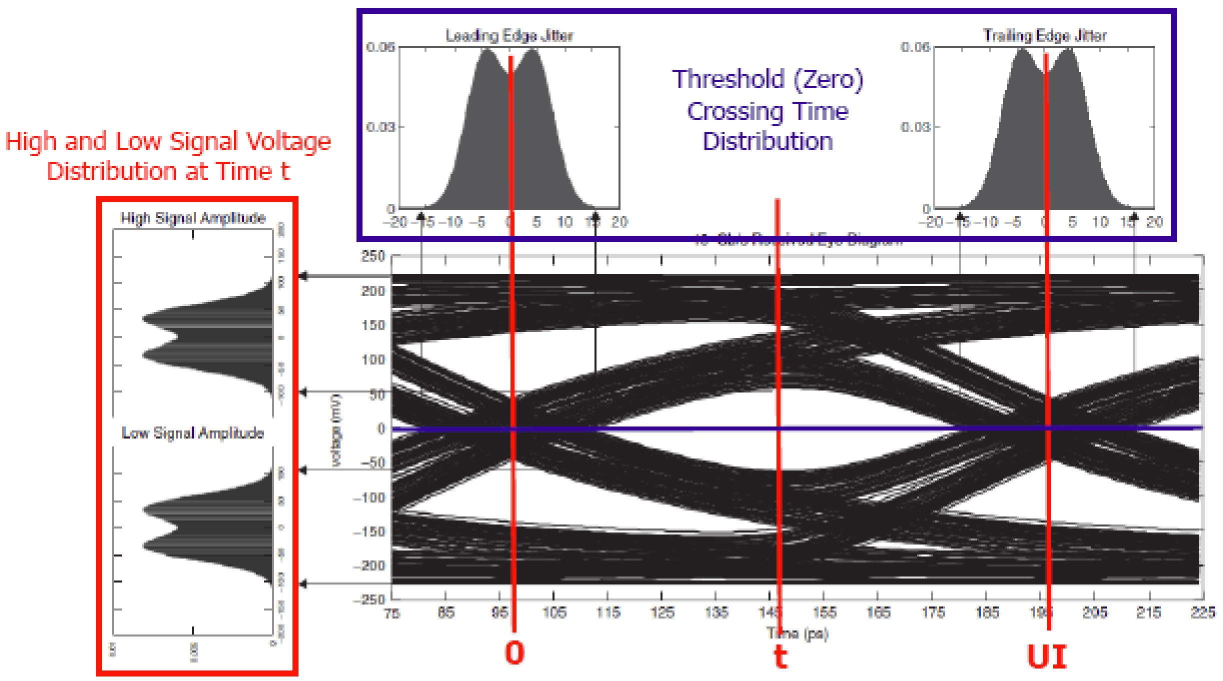


Time Trend(ps)
6
4
2
0
−2
−4
−6
0.0
0.2
0.4
0.6
0.8
1.0
T(us)
Hist(population)
450
400
350
300
250
200
150
100
50
0
−6
−4
−2
0
2
4
6
Jitter(ps)
Spectrum(dB)
70
60
50
40
30
20
10
0
0
100
200
300
400
500
Freq(MHz)

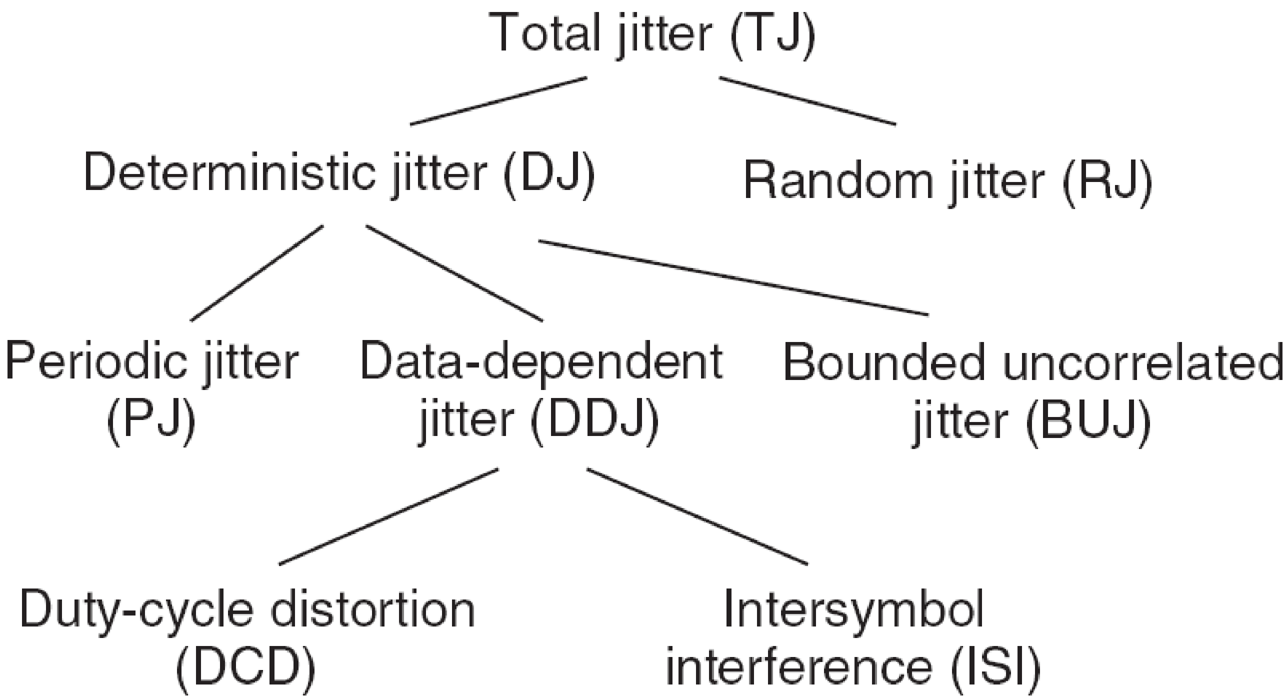
Combined



Jitter Histogram



Jitter category



Bounded vs Unbounded

- RJ is unbounded - long tail of Gaussian distribution
- Dj are bounded - must within a fixed range
- This is a significant differences, that's the reason why tail fit works

Uncorrelated vs Dependent

- Uncorrelated or dependent is relative to data stream
- Uncorrelated jitter need "**convolve**" , dependent jitter need "**add**".
- RJ and DJ is uncorrelated
- DCD,ISI is data dependent
- PJ is also uncorrelated to data, but still "add"
- Crosstalk is BUJ(Bounded uncorrelated jitter)

Review of statistics

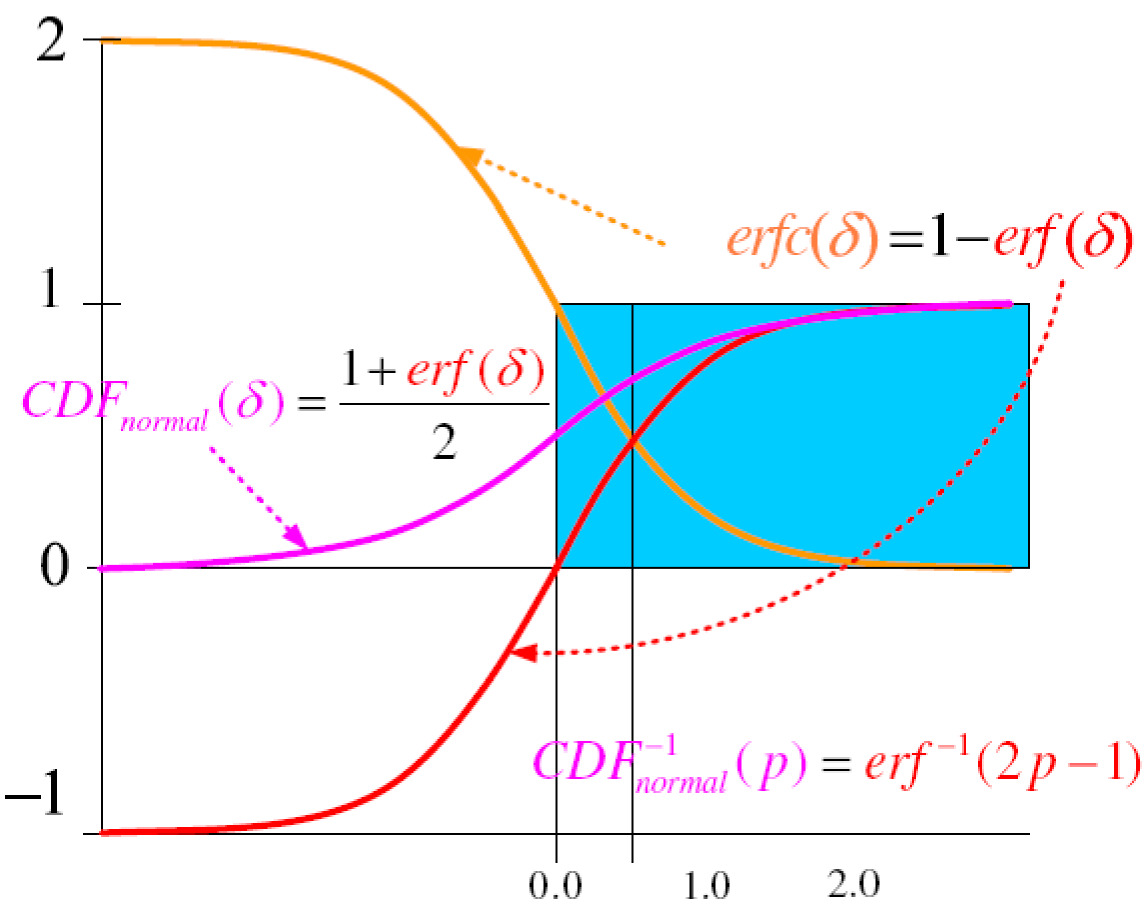
- PDF - The Probability density function, sometimes written $\rho(x)$.Histograms are closest in terms of a measurable probability density, and are said to approximate the PDF when normalized.
- CDF - The Cumulative Distribution Function is $CDF(x) = \int_{-\infty}^{x} PDF(u)du$

- PDF of Gaussion. $\rho(x) = \frac{w}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Statistics cont'd

- Error Function. $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du$. It is a basic a mathematical function closely related to the cumulative distribution function for a Gaussian. $1 - 2CDF_{Gaussian}(x) = erf(x)$
- Complimentary Error Function. $erfc(x) \equiv 1 - erf(x)$ or $erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-u^2} du$
- Inverse Error Function, $erf^{-1}(x)$, This function has no closed analytic form but can be obtained from numerical methods.

ERF relations

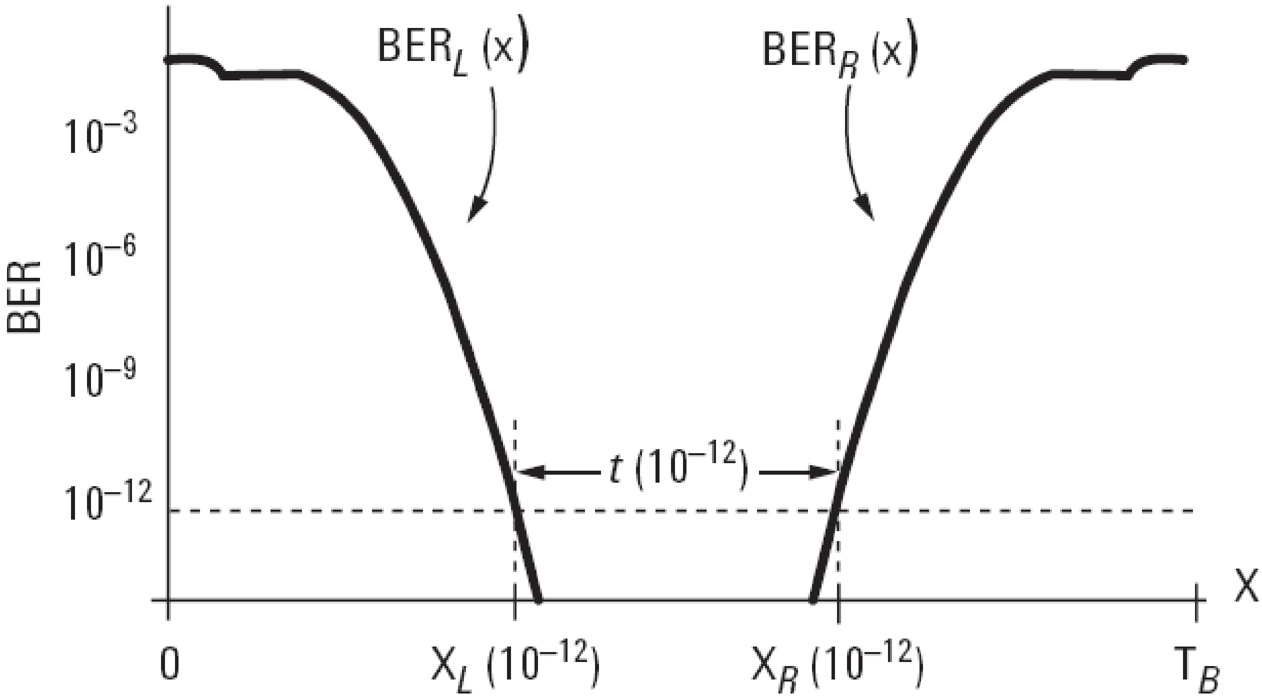


BERT vs Scope

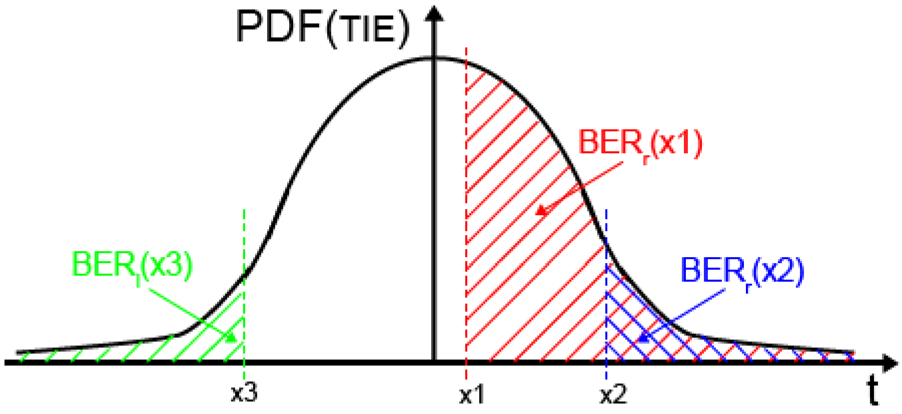
- BERT
  - Shift the sampling point across an entire bit time
  - Base on number of errors found
  - Accuratly measures the total jitter
- Scope

- Get a limited number of samples
- Good at determining the jitter components
- Need post-process to estimate RJ

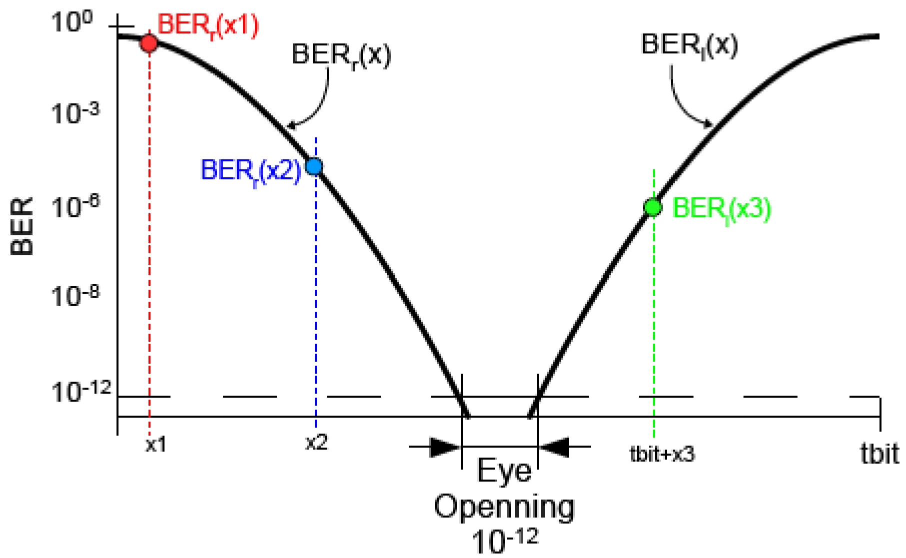
Bathtub



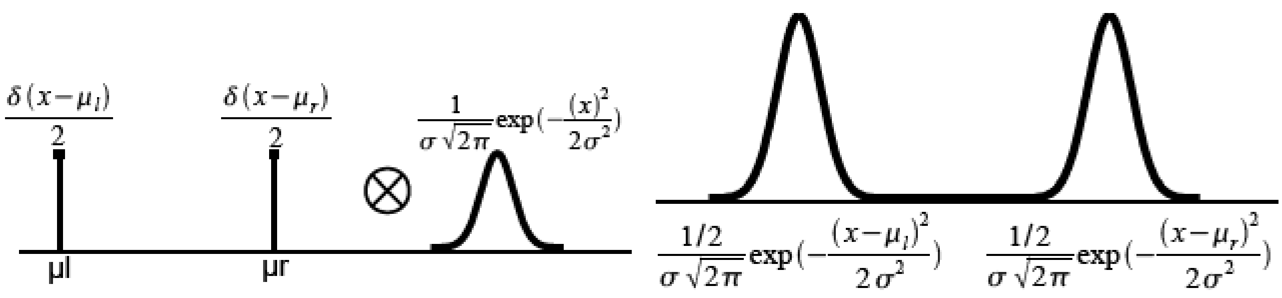
BER vs Jitter



(b) PDF of TIE



Dual Dirac Model



(a) PDF of each component

(b) $TJ_{(\delta\delta)}$ PDF

- An approximate method
- Quickly estimating of total jitter
- Not reflect the true jitter

Assumption

1. Jitter can be separated into two categories, random jitter (RJ) and deterministic jitter (DJ).

2. RJ follows a Gaussian distribution and can be fully described in terms of a single relevant parameter, the rms(σ) value of the RJ distribution.
3. DJ follows a finite, bounded distribution.
4. DJ follows a distribution formed by two Dirac-delta functions.
5. Jitter is a stationary phenomenon. Give the same result regardless of when that time interval is initiated.

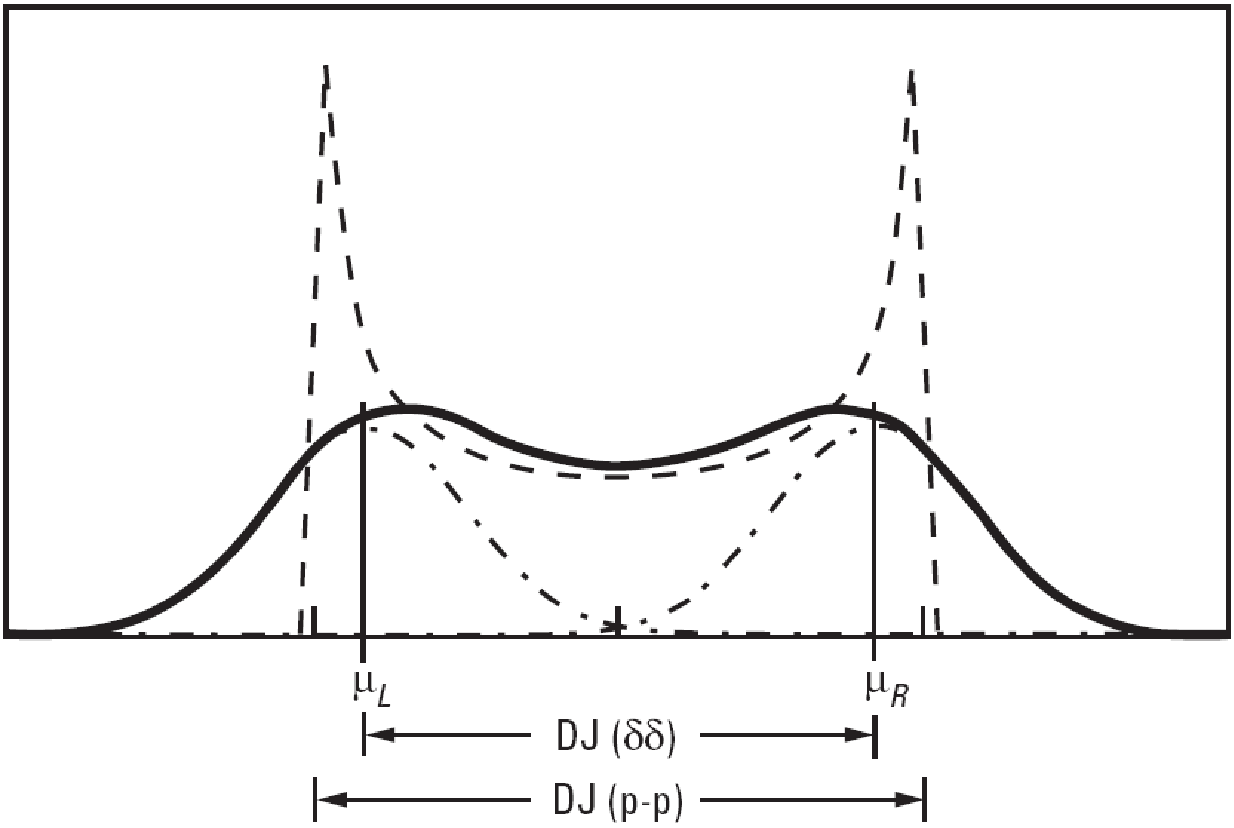
Math Expression

- PJ pdf is $A * \delta(x - \mu_l) + B * \delta(x - \mu_r)$
- RJ pdf is $\frac{1}{\sigma_{(\delta\delta)}\sqrt{2\pi}} \exp\left[-\frac{x^2}{2\sigma^2_{(\sigma\sigma)}}\right]$
- TJ combined by convolve, as it is uncorrelated $\frac{A}{\sigma_{(\delta\delta)}\sqrt{2\pi}} \exp\left[-\frac{(x-\mu_l)^2}{2\sigma^2_{(\sigma\sigma)}}\right] + \frac{B}{\sigma_{(\delta\delta)}\sqrt{2\pi}} \exp\left[-\frac{(x-\mu_r)^2}{2\sigma^2_{(\sigma\sigma)}}\right]$
- RJ RMS combine: $\sigma_{Total} = \sqrt{\sigma_1^2 + \sigma_2^2 + \ldots \sigma_n^2}$
- DJ pp combine: $DJ_{total}(\delta\delta) = DJ_1(\delta\delta) + DJ_2(\delta\delta) + \ldots + DJ_n(\delta\delta)$

DJ_{pp} vs $DJ_{\delta\delta}$

- Dual-Dirac DJ is a completely different quantity than the peak-to-peak DJ(very confused)
- To distinguish the two use the notation DJ_{pp} as the real jitter, $DJ_{\delta\delta}$ as dual-Dirac model DJ
- DJ_{pp} never follows the simple dual-Dirac distribution
- $DJ_{\delta\delta}$ is a model dependent quantity that must be derived under the assumption that DJ follows a distribution formed by two Dirac-delta. Generally, $DJ_{\delta\delta} < DJ_{pp}$

DJ_{pp} vs $DJ_{\delta\delta}$ cont'd



TJ vs BER

$TJ(BER) = 2 \cdot N(BER) \cdot \sigma_{total} + DJ_{total}$

$N = \sqrt{2} \cdot erfc^{-1}(2 \cdot BER)$

if BER if 1e-12, we could get the factor as 14.07 by the following code. Most of time we use a look-up table

```
from scipy.special import erfcinv
import numpy as np
BER = 1E-12
print 2*np.sqrt(2)*erfcinv(2*BER)
```

Jitter decompose

- Spectral Method
- Tail Fit method

Spectral Method

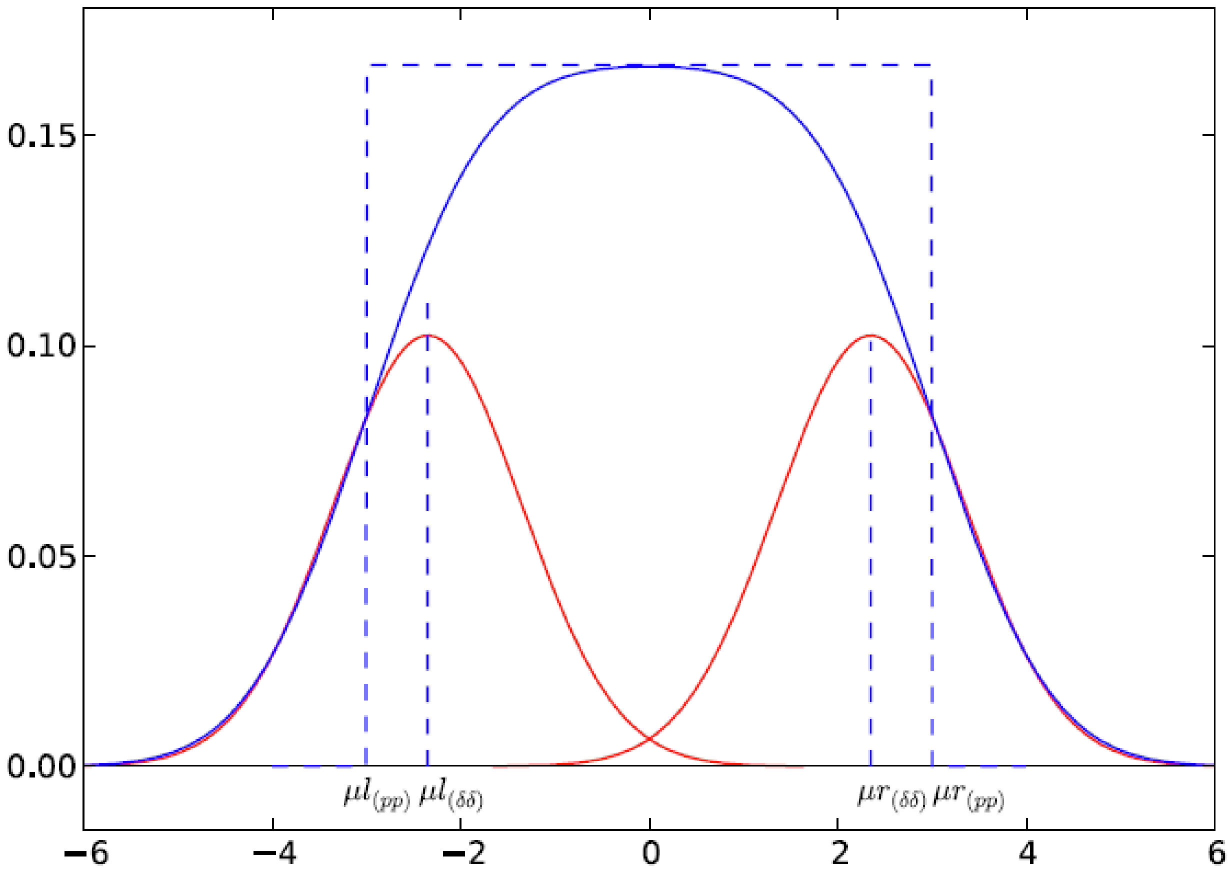
- FFT of TIE(t)

- spectral “peaks” are manifestations of deterministic jitter, and thus provide a measure of DJ
- The remaining spectrum, or “noise floor” accounts for all of RJ
- Remove the "peaks", and IFFT the "noise floor" get RJ
- Deconvolve and get DJ

Spectral problem

- BUJ like crosstalk is will impact noise floor
- Non-Stationary Periodic aggressors (like Spread Spectrum) manifest broadly in the jitter spectrum and cannot always be identified as “peaks”
- In effect, any wide-band aggressor which contributes bounded timing fluctuations will be indistinguishable from the noise floor, and consequently “counted” as RJ.
- Then it is easy to overestimations of Tj

Tail fit method



Tail fit difficulties

- Exponential waveform fit. - Q-scale

- Find the region of pure Gaussian PDF.   - Re-normalized Q-scale
- Need enough samples to get more long tail data.

Q-scale

- A smart method transform the exponential waveform to linear
- Q-scale is a simple coordinates transformation. Similar method like Cartesian coordinates to Polar coordinates, time domain to frequency domain. Different view point of the same physical object.
- Replace x with $Q = \frac{x-\mu}{\sigma}$ in BER expression, we get Q scale.

Math deduction

At the region far from DJ(long tail), the distribution is Gaussion, it also have left and right part. Just take left side as example.

$$BER_l(x) = K_l \frac{1}{\sigma_{l\delta\delta}\sqrt{2\pi}} \oint_x^\infty \exp\left[-\frac{(x' - \mu_{l\delta\delta})}{2\delta_{l\sigma\sigma}}\right] dx'$$

Let $Q = \frac{x-\mu_{l\delta\delta}}{\sigma_{l\delta\delta}}$, we get

$$BER_l(Q) = K_l \frac{1}{\sigma_{l\delta\delta}\sqrt{2\pi}} \oint_Q^\infty \exp\left[-\left(\frac{Q'}{\sqrt{2}}\right)\right] dQ'$$

Math cont'd 1

Remember that complementary error function is given by

$$erfc(x) = \frac{2}{\sigma\sqrt{2\pi}} \oint_x^\infty \exp(-u^2) du$$

then we could rewrite $BER_l(Q)$ as

$$BER_l(Q) = K_l erfc(\frac{Q}{\sqrt{2}}) = K_l(1 - erf(\frac{Q}{\sqrt{2}}))$$

inverse the function, $A_l = \frac{1}{K_l}$we get

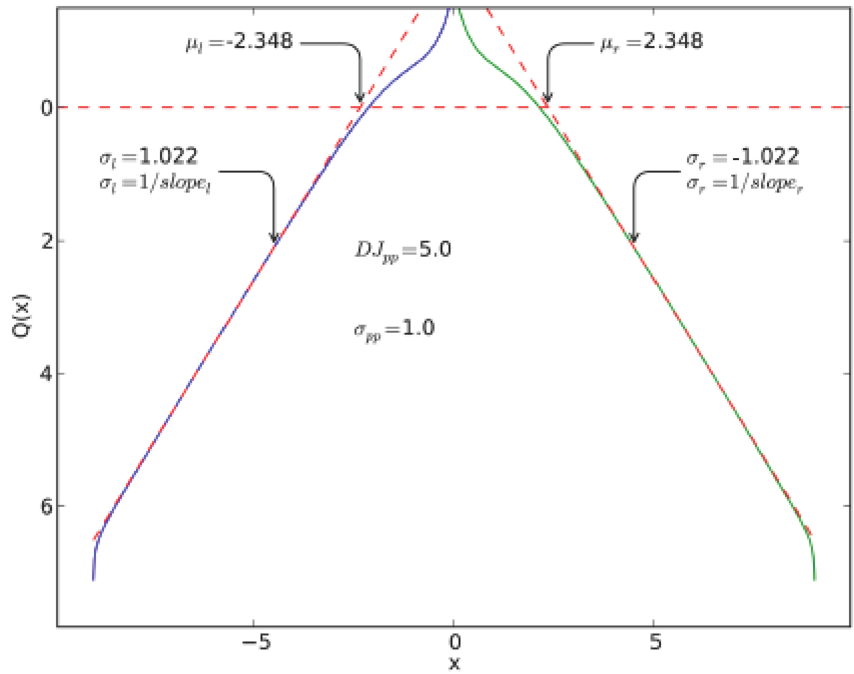
$$Q_l(x) = \sqrt{2} erf^{-1}\left[1 - BER_l \cdot A_l\right]$$

Math cont'd 2

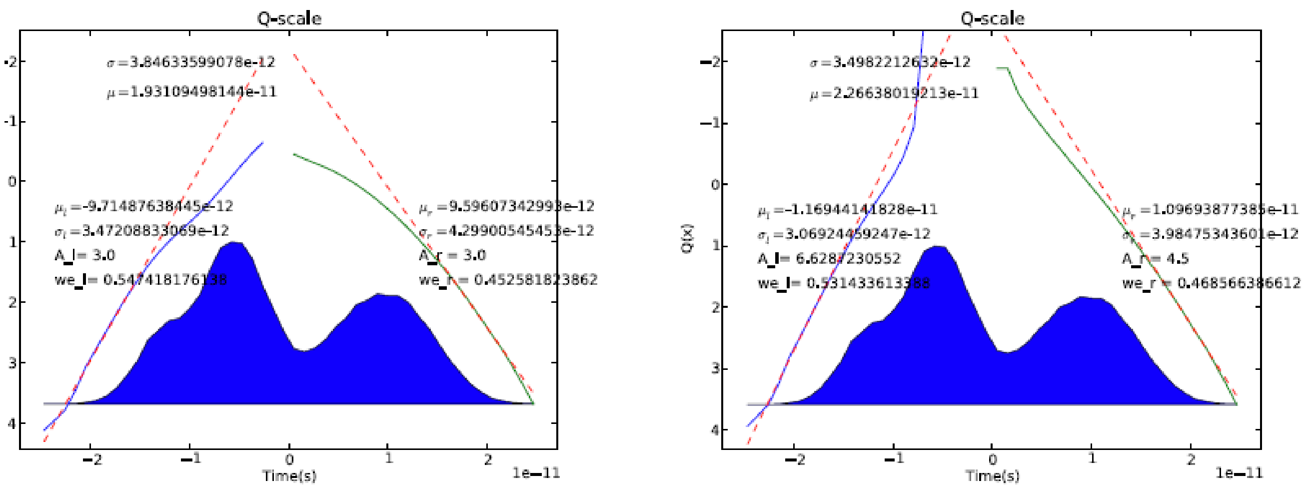
compare the 2 equation, the target is fit the 2.

$$Q_l(x) = \sqrt{2} erf^{-1}\left[1 - BER_l(x) \cdot A_l\right]$$

$$Q_l(x) = \frac{x - \mu_{l\delta\delta}}{\sigma_{l\delta\delta}}$$



Q-scale fit

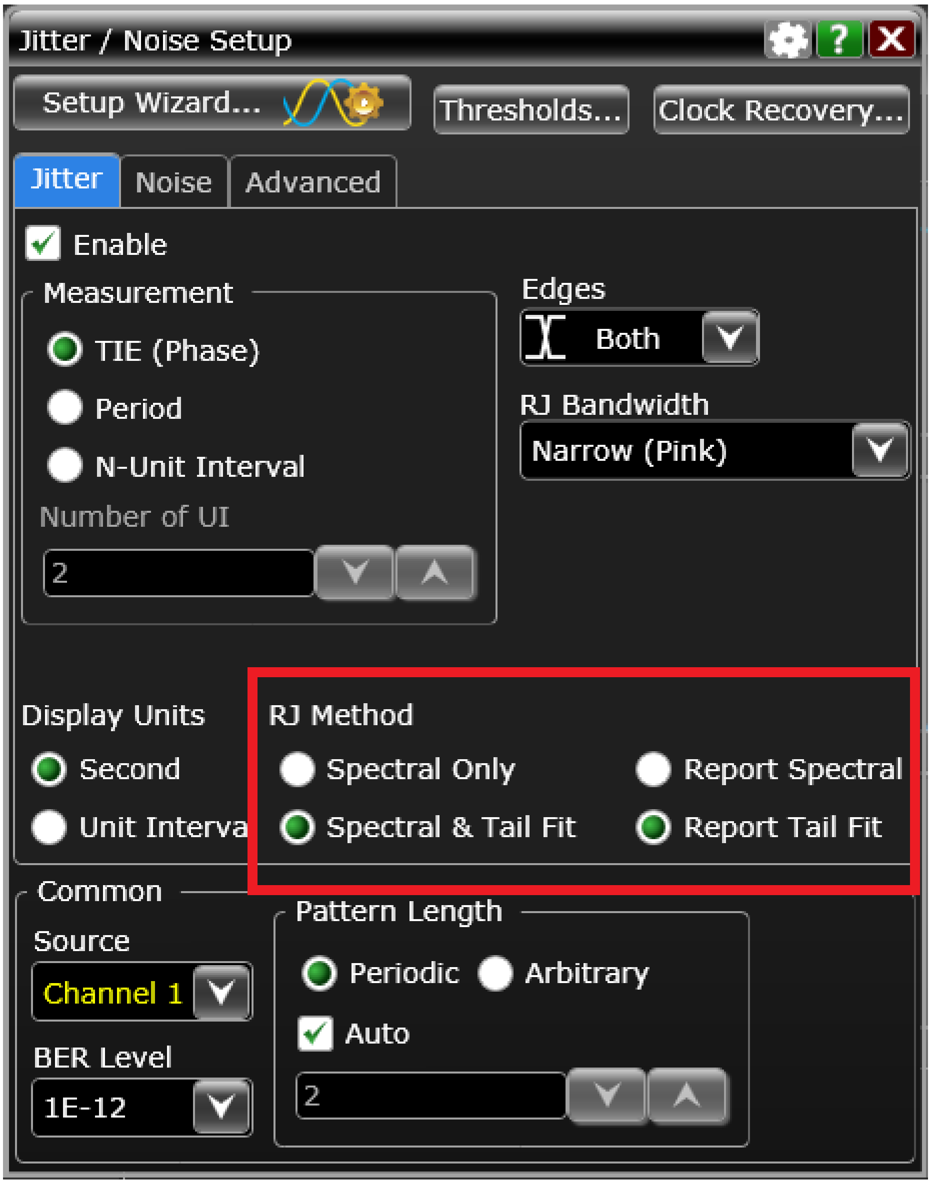


(a) Non-improved linear approximation (b) Improved linear approximation

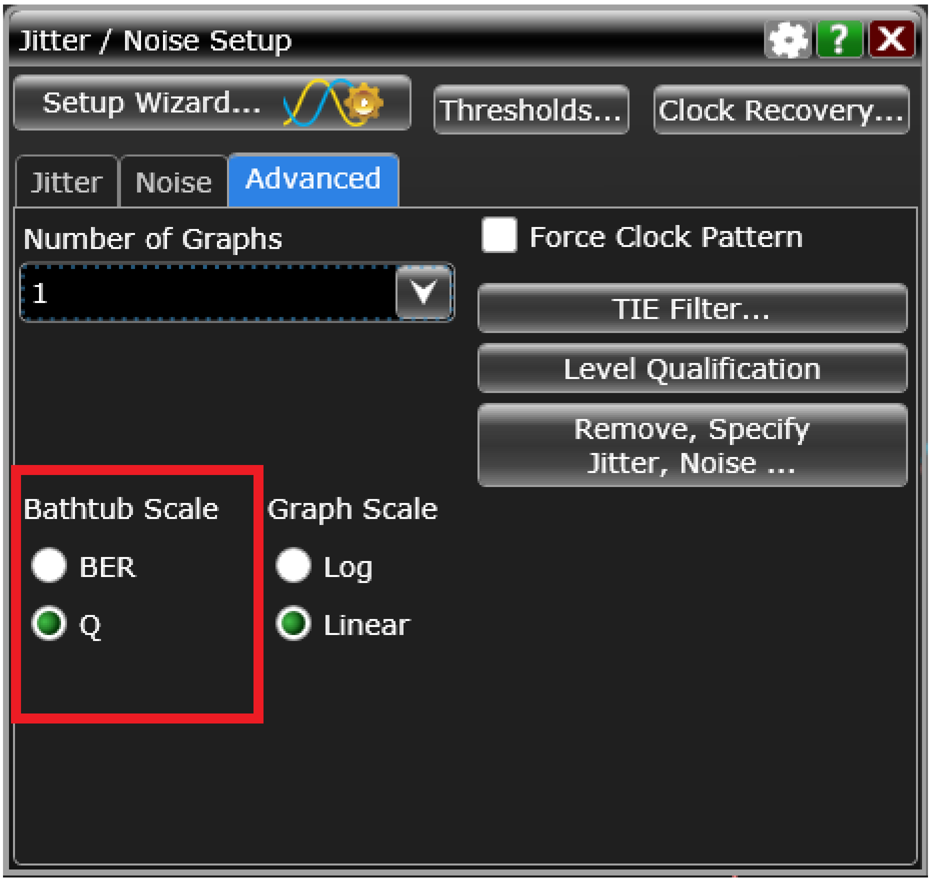
Pros of tail fit

- The noise like or flat spectrum jitter source like crosstalk and SSC could be remove from RJ
- If a system will too much crosstalk, it is better use tail fit method, otherwise spectrum method could be used.

Scope setting



Q-Scale option



Split threshold


Keysight Infiniium : Thursday, June 23, 2016 1:54:09 PM
File Control Setup Display Trigger Measure Math Analyze Utilities Demos Help
KEYSIGHT TECHNOLOGIES
Run Stop Single
80.0 GSa/s
4.11 Mpts
33 GHz
10 mV
Waveform Window 1
Jitter/Noise
Type All
Graphs 1
RJ,PJ Threshold (Periodic)
RJ+PJ Aliased Spectrum
RJ Aliased Spectrum
Separation Threshold
RJ Baseline
Power Spectral Density (Log)
0.0 Hz
1.250 MHz
2.490 MHz
3.740 MHz
4.980 MHz
Results (Measure All Edges)

BER scale

Keysight Infiniium : Thursday, June 23, 2016 2:02:51 PM

Q scale

Jitter Result

Spectrum

Results (Measure All Edges)	
Source	Channel 1
RJ Method	Spectral
Data Rate	3.0000005 GHz
Pattern Length	---------------
TJ(1E-12)	74.31 ps
RJrms,narrow	2.02 ps
DJδδ	45.51 ps
Transitions	308.492 k
PJrms	13.11 ps
PJδδ	45.24 ps
DDJpp	990 fs
DCD	990 fs
ISIpp	0.0 s
DDPWS	990 fs
Clock Recovery	Constant Freq
Edge Direction	Both
Measurement	TIE (Phase)
ABUJrms	--------------

Tail Fit

Results (Measure All Edges)	
Source	Channel 1
RJ Method	Tail Fit
Data Rate	3.0000005 GHz
Pattern Length	Clock
TJ(1E-12)	87.08 ps
RJrms	3.87 ps
DJδδ	31.87 ps
Transitions	308.492 k
PJrms	13.11 ps
BUJδδ	32.27 ps
DDJpp	990 fs
DCD	990 fs
ISIpp	0.0 s
DDPWS	990 fs
Clock Recovery	Constant Freq
Edge Direction	Both
Measurement	TIE (Phase)
ABUJrms	0.0 s

Summary

- Dual-Dirac model is for estimate RJ, aks. extrapolation.
- RJ is the easy split than DJ, as its PDF feature.
- Both spectrum and tail fit have pros and cons