A METHOD OF SERIAL DATA JITTER ANALYSIS USING ONE-SHOT TIME INTERVAL MEASUREMENTS

Jan Wilstrup, Corporate Consultant

Wavecrest Corporation, a Technologies Company
7275 Bush Lake Road, Edina, MN 55439

ABSTRACT

A method for measuring inter-symbol interference, duty cycle distortion, random jitter and periodic jitter is described.

The Blackman-Tukey method of signal analysis is used. This allows the application of jitter tolerance masks to ensure compliance to data communication standards.

Note: Mathematical analysis and computer simulations have been performed. The autocorrelation function and an FFT for clock jitter analysis have been implemented on our current product. A provisional patent application has been filed.

DEFINITIONS [1]

Jitter in serial data communication is a difference of data transition times relative to ideal bit clock active transition times.

UI is a Unit Interval. It is the average period of the bit clock.

ISI is Inter-Symbol Interference. ISI is caused by a data path propagation delay that is a function of past data history and occurs in all finite bandwidth data paths.

DCD is Duty Cycle Distortion. DCD is caused by differing propagation delays for positive and negative data transitions.

PJ is Periodic Jitter. This jitter is caused by one or more sine waves and its/their harmonics.

RJ is Random Jitter. It is assumed to be Gaussian (normal) and has a power spectral density that is a function of frequency.

ISI, DCD and PJ are all bounded. They may be described as a peak or peak to peak value in UI's or seconds. PJ in general, has a magnitude for each spectral line.

RJ is unbounded. It may be described by a standard deviation in UI's or seconds. A jitter tolerance mask is a function that is defined in the frequency domain. It has a magnitude that is a function of frequency.

In most serial data communications systems, the data/clock recovery circuit tolerates low frequency jitter more than high frequency jitter and the shape of the mask reflects this fact.

JITTER EXAMPLE (Figure 1)

The deltas in Figure 1 are the differences in time from data transitions and active bit clock transitions. d0 is the delta associated with data edge 0; d1 is the delta associated with data edge 1, etc.

If a given data pattern is repeated many times, ISI+DCD will be static relative to a pattern boundary (reference edge 0). Each delta will be a constant. The serial data edge positions, for ISI+DCD, have converged to a steady state. However, ISI+DCD will have jitter relative to a bit clock. The deltas will not be the same value.

PJ and RJ will cause each delta to vary in time. The variance of these deltas over N (an integer>0) UI time intervals yields an autocorrelation function of PJ and RJ (See appendix).

The means of the deltas allow an estimation of ISI+DCD. The variance of the deltas allows PJ and RJ estimations. In this way, ISI+DCD estimates may be separated from PJ and RJ estimates.

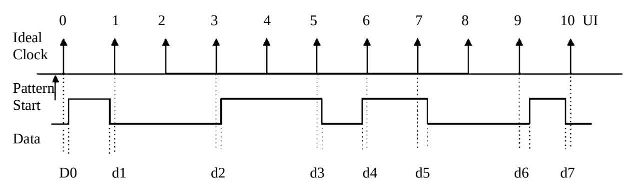

Figure 1. Jitter Example

SIGNALS REQUIRED

1. Serial data signal. This data must have a known pattern and it repeats.

2. Pattern start signal. It has a transition that has an unambiguous relationship to an arbitrary data reference edge. It may come from the equipment under test or from a hardware-based pattern recognizer that "triggers" on a word which has one unique position in the data pattern.

3. No bit clock is required or desired. In many cases, a low jitter bit clock is not available.

INSTRUMENT HARDWARE

1. Has a one-shot timer with low jitter, compared to the data signal jitter.
2. A one-shot measurement is taken from the *n*th start (Nstart) data edge after the pattern start signal to the *n*th stop edge (Nstop) after the pattern start signal. Nstart and Nstop are integers. This function requires two programmable counters. The pattern start signal is used as an external arming signal for the instrument.
3. The time between measurements is randomized. This is to ensure that the autocorrelation function performs correctly. This is random sampling in the statistical sense.[2] WAVECREST'S DTS-2075 presently takes one-shot measurements over a random time interval of about 21 us to 25 us + the measured time interval. An "and" function of the internal arm signal, which is randomized, and the external arm signal (pattern start) initiates a single one-shot time interval measurement.

PROCEDURE GOALS

1. High jitter tolerance. The method provides "good" estimates of jitter on systems that have almost complete eye closure.
2. Statistical checks are used to ensure jitter analysis integrity and accuracy.
3. The data provided by the procedure will form a basis for system diagnosis and characterization and compliance to data communication standards.

PROCEDURE OVERVIEW

In this sequence:

1. Measure a UI. This forms the basis for subsequent measurements and data analysis.
2. Pattern match. Test the measured data against an ideal image of the expected pattern. Estimate ISI+DCD.
3. Compute measurement sets needed to estimate PJ and RJ.
4. Take measurements and compute the variance of each set.
5. Run an FFT (Fast Fourier Transform) and apply an optional mask.
6. Separate PJ and RJ and compute their estimates.

DETAILED EXPLANATION OF PROCEDURE

In this sequence:

1. UI estimation. Take M1 (an integer) measurements of the start pattern signal and compute the mean and standard deviation. Divide the mean by the length of the pattern in UI (Lpatt): this is the UI estimate. Using the standard deviation, estimate the standard error of the mean: STD(measurements)/SQRT(M1). This estimate will work well if M1>100. [2] Test this against a default or user-defined constant. If this test fails, the following measurements and analysis may be in error. Increasing M1 may allow this test to pass.

2. Pattern match. Compare measured data with an image of the expected pattern. The image of this pattern must be rotated to perform this match as the relationship between the start pattern signal and the reference data edge is arbitrary. The match uses a least square criteria. The following material describes a pattern image, rotations and a pattern match.

The measurements follow this schedule:

A. Take M2 measurements from data edge 0 (reference edge) to data edge 1. Compute mean and error of the mean. Test error of the mean.
B. Repeat A for edge 0 to edge 2, edge 0 to edge 3...edge 0 to edge (end-1). "end" is the number of the edge where the pattern repeats.
C. The test in A may fail, especially in the presence of large amounts of PJ and RJ. Increasing M2 may allow this test to pass. M2 will usually be greater than 100.

INTRODUCTION

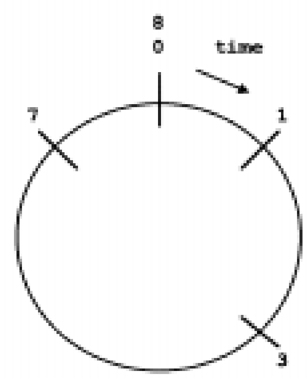

Figure 2. A Pattern Image

This circle shows a simple, serial data pattern. Time moves CW (clock-wise) as shown by the arrow. The numbered lines represent edge (transition) positions in UI (Unit Interval).

The top of the circle is the reference position of the pattern. This pattern starts on position 0 and ends on position 8. Eight (8) is the length of the pattern (Lpatt) in UI. This pattern is described by the set: [0 1 3 7 8] .

This pattern has 4 edges (end=4). The start and end positions are counted as one edge. From above: edge number 0 is the reference edge at 0 UI, edge number 1 at 1 UI, edge number 2 at 3 UI, etc. The number of edges in a pattern must always be even, otherwise its' start and end edges would have opposite polarities.

In the above circle, no polarities are shown. However, assume that a positive (+) transition occurs at 0 UI, negative (-) at 1 UI, + at 3 UI, - at 7 UI and + at 8 UI. The pattern has repeated. A general way of looking at patterns is to ignore polarity. This allows pattern definitions on both inverted and non-inverted data without concern about polarities.

The above circle displays no jitter. All of the edges have integer UI positions and they will have zero displacements relative to ideal bit clock transitions.

The reference pattern [0 1 3 7 8] is rotated CCW 1 edge number by subtracting the edge position of the first edge after the reference position from all of the edge positions. In this case, subtract 1. This results in the upper right circle of Figure 3: [0 2 6 7 8].

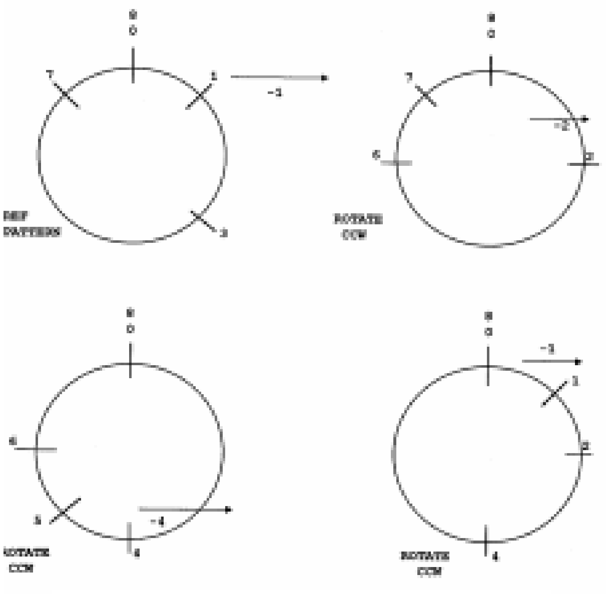

Figure 3. Rotating the Pattern Image

Edge numbers: [0 1 2 3 4]
The circles: [0 1 3 7 8] **Note**: All of these circles
[0 2 6 7 8] come from the same pattern.
[0 4 5 6 8] Only the reference edge
[0 1 2 4 8] has changed.

One more rotation will result in [0 1 3 7 8], the initial circle. Four rotations of the initial pattern (the number of edges in the pattern) will yield the initial pattern.

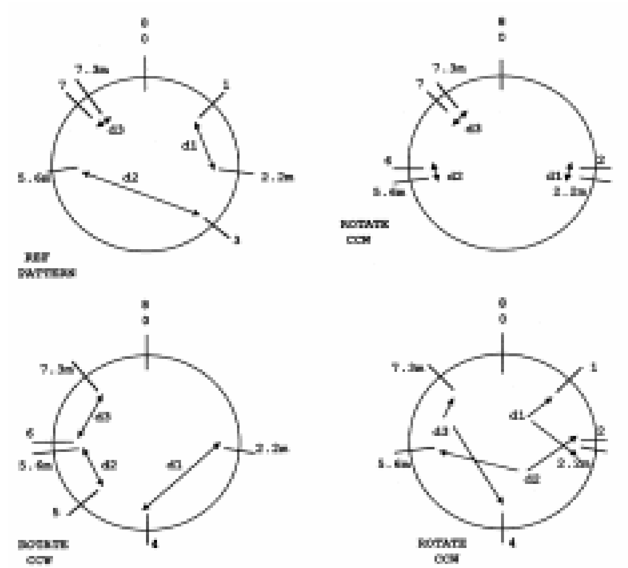

Figure 4. Pattern Matching

In Figure 4, the measured means 2.2m, 5.6m and 7.3m are placed on the four possible rotations of the pattern [0 1 3 7 8]. These means are from the reference edge to the first, second and third edges. The goal is to find out which rotation of the pattern will match the measured data. The *i*th delta, where i is the edge number, is the *i*th ideal position minus the *i*th measured mean. The deltas are squared and then summed for each rotation. The "match" rotation has the smallest sum (S).

[0 1 3 7 8] S=squar(1.0-2.2)+squar(3.0-5.6)+squar(7.0-7.3)=8.29

[0 2 6 7 8] S=squar(2.0-2.2)+squar(6.0-5.6)+squar(7.0-7.3)=0.29

[0 4 5 6 8] S=squar(4.0-2.2)+squar(5.0-5.6)+squar(6.0-7.3)=5.29

[0 1 2 4 8] S=squar(1.0-2.2)+squar(2.0-5.6)+squar(4.0-7.3)=25.29

[0 2 6 7 8] is the rotation that matches: it has the smallest sum. The quality of the match may be found by computing the standard deviation of the deltas that have the least sum. Our simulations, using many random data patterns and large amounts of jitter (ISI+DCD), show that a standard deviation less than 0.5 UI is a very good match.

The estimate for ISI+DCD peak to peak is:

MAX[-MIN(deltas), (MAX(deltas)-MIN(deltas)),MAX(deltas)]

The deltas are the set of deltas computed from the matched pattern. In the above equation, the -MIN(deltas) and the MAX(deltas) are needed when d0 (the reference edge delta) is at the lower or upper extreme of the overall delta distribution.

3. Compute measurement sets for Variance(tmeas(N)) (called VAR(N)) using the edge positions from the matched pattern. VAR(N) is used to feed data to the FFT, which provides PJ and RJ data. The goal is to compute measurement sets such that only one set of measurements is taken for each N: This speeds the following measurements.

A. Note: “end” is the last edge number in a matched pattern. Lpatt is the length of the pattern in UI. p(i) is an ideal edge position in UI for edge i in the matched pattern.

p(end)=Lpatt

Calculate expected N's from:

a. p(2*end)-p(0). p(2*end)-p(1).
p(2*end)-p(2) ....p(2*end)-p(2*end-1).

b. p(2*end-1)-p(0). p(2*end-1)-p(1).
p(2*end-1)-p(2)....p(2*end-1)-p(2*end-2).

c. p(2*end-2)-p(0). p(2*end-2)-p(1).
p(2*end-2)-p(2)....p(2*end-2)-p(2*end-3).

...
...
...
zzz. p(1)-p(0).

B. From all of the combinations in A, find measurement sets which "cover" N from 1 to Lpatt and have only one measurement set for each N. Each measurement set will have a unique edge pair and an expected N. Some patterns will not cover all N: there will be "holes" (gaps). The hole locations are stored.

C. Convert edge number pairs into a format for the DTS-2075. The data from B. is converted to Nstart +/- and Nstop +/- for the arm on *n*th event counters in the DTS-2075 circuitry.

4. Take M3 measurements for each edge pair, calculate variance and mean. Test mean against expected N. Store VAR(N) in location: "expected N" for that particular pair. Use interpolated data to "fill holes" (if any). A VAR(N) record has been created. This record is an autocorrelation function of PJ and RJ.

5. Run the FFT. This will provide RJ and PJ information. Note: Taking the FFT of the autocorrelation function of a signal is the basis of the Blackman-Tukey signal analysis method. This procedure uses this method to estimate PJ and RJ. See [3] for descriptions of the Blackman-Tukey method and FFT pre-processing practice.

A. Mirror VAR(N) to create MVAR(N). Var(0) is set to zero. The mirror function makes use of the symmetry of VAR(N) around N=0. VAR(N)=VAR(-N). This nearly doubles the length of the VAR(N) record and this improves the frequency resolution of the FFT output.

B. Force mean of MVAR(N) to zero.

C. Window MVAR(N). The Blackman-Tukey method usually uses a triangular window.

D. Padd record. This improves the FFT's resolution and accuracy.

E. Run the FFT. A radix 2 FFT is usually used.

F. Use a mask to weigh the FFT output as a function of frequency. Generally, serial data communication systems are more tolerant of low frequency jitter than high frequency jitter. This step is optional.

6. Separate PJ and RJ. Add amplitudes of the PJ spectral lines to give the magnitude of PJ in peak UI. Sum the RJ curve and take the square root to estimate RJ expressed as a standard deviation in UI's.

The method of separating PJ and RJ uses a technique called a constant false alarm filter that is used in radar. [4]

It consists of a sliding window that is applied to the FFT output bins. This window has a odd number of bins. Example: 9 bins. The lower 4 and upper 4 bins are averaged. If the central bin is larger than this average by a defined ratio, the magnitude and position of the central bin is stored away and later used to identify the spectral lines created by PJ. The window is moved over one bin and this process is repeated until the all of the FFT output bins have been processed.

See following section for a simulation of the constant false alarm filter.

CONSTANT FALSE ALARM FILTER SIMULATION RESULTS

Figure 5 displays the raw output of an FFT of PJ and RJ. The vertical axis is in dB and the horizontal axis is frequency. The spectral peaks in the display are PJ and the underlying envelope in the display is RJ.

At this point, the FFT raw data may be multiplied by a jitter tolerance mask for standards testing.

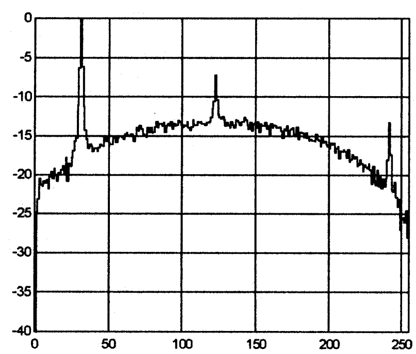

Figure 5. Raw Output

The constant false alarm filter is applied.

Figure 6 displays the constant false alarm filter output. The sliding window of the filter has a width of 9 bins. The magnitude and frequency of each PJ spectral line has been isolated from RJ. An estimate of the total PJ is the algebraic sum of the spectral line magnitudes. This gives an estimate of total PJ in peak UI or seconds.

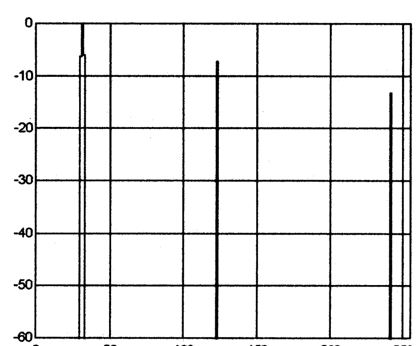

Figure 6. Filtered Output

An estimate of RJ is calculated by removing the spectral lines from the FFT raw data. The magnitude of the bins are summed and a square root is taken. The result is a one-sigma estimate of RJ in UI or seconds.

CONCLUSION

A method of measuring serial data communication jitter has been described. Further work is needed in:

1. Estimating the error in PJ and RJ estimations caused by the presence of ISI+DCD.
2. Reducing the error in 1. by choosing a better measurement schedule for VAR(N).
3. Refining the metrics for the statistical tests used in this procedure.

APPENDIX [1] [3]

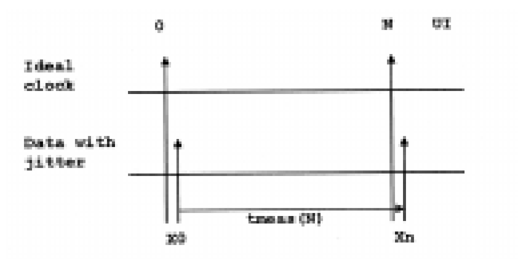

Figure. A-1

Figure A-1 shows a one-shot measurement spanning N*UI. X0 and Xn are the deltas for the jitter.

From graph:

(1.1) tmeas(N)=N*UI+Xn-X0

M measurements on a randomized time schedule are taken for each N.

Therefore:

$$(1.2)\quad \text{Variance}(\text{tmeas}(N)) = \frac{1}{M-1}\sum_{k=1}^{M}\left[\text{tmeas}(N)-\overline{\text{tmeas}(N)}\right]^2$$

$$(1.3)\quad \overline{\text{tmeas}(N)} = \frac{1}{M}\sum_{k=1}^{M}(N*UI+Xn-X0)$$

$$(1.4)\quad \overline{\text{tmeas}(N)} = N*UI + \frac{1}{M}\sum_{k=1}^{M}(Xn-X0)$$

If ISI+DCD is 'small' the mean of (Xn-X0) is small, then:

$(1.5)\quad \overline{\text{tmeas}(N)} \cong N*UI$. Substitute into (1.2) .

$$(1.6)\quad \text{Variance}(\text{tmeas}(N)) \cong \frac{1}{M-1}\sum_{k=1}^{M}(Xn-X0)^2$$

$$(1.7)\quad \text{Variance}(\text{tmeas}(N)) \cong \frac{1}{M-1}\sum_{k=1}^{M}(Xn^2+X0^2) + \frac{1}{M-1}\sum_{k=1}^{M}(-2*Xn*X0)$$

The first term in (1.7) is constant as a function of N and the second term is -2 times the autocorrelation function of PJ and RJ, see (1.8).

$$(1.8)\quad \text{Variance}(\text{tmeas}(N)) \cong C - 2*\left[\frac{1}{M-1}\sum_{k=1}^{M}(Xn*X0)\right]$$

N*UI is τ (tau) the lag time of the autocorrelation function.

The Fourier transform of the autocorrelation function from a signal gives the power spectral density of the signal.

When an FFT is performed on (1.8), the mean of the data record is forced to 0: This makes the value of C irrelevant. This is justified by the fact that PJ and RJ are not static, they have no "DC" component.

REFERENCES

[1] Information Transmission, Modulation, and Noise
Mischa Schwartz
Fourth Edition
McGraw-Hill Publishing Company 1990
ISBN 0-07-055909-0

[2] Statistical Methods
Rudolf J. Freund/William J. Wilson
Revised Edition
ACADEMIC PRESS, INC. 1997
ISBN 0-12-267472-3(alk. paper)

[3] Digital Signal Processing
A Practical Approach
Emmanuel C. Efeachor Barrie W. Jervis
Addison-Wesley Publishing Inc. 1993
ISBN 0 201 54413 X

[4] Private communication with:
J. M. Gwinn
Raytheon Company