Description, Installation, and Use of *jitterhist*:
A Program to Compute and Analyze Time Interval Error

Shawn M. Logan
May 15, 2025
smlogan@ieee.org

1. INTRODUCTION 1

2. BACKGROUND AND DEFINITIONS 2

2.1 TIME INTERVAL ERROR (TIE) 2
2.1.1 Definition 2
2.2 MEASURING TIME INTERVAL ERROR 2
2.2.1 Impact of High Frequency Modulation (fm >> fo) 3
2.2.2 Impact of Multi-Unit Interval, Low Frequency Phase Modulation 5
2.2.3 Estimating the Long Term Average Frequency 8
2.2.4 Impact of Reference Frequency Accuracy on Time Interval Error 8

3. *JITTERHIST* DESCRIPTION, INPUTS, AND OUTPUTS 11

3.1 PROGRAM DESCRIPTION 11
3.2 PROGRAM INPUTS 11
3.3 PROGRAM OUTPUTS 11
3.4 *JITTERHIST* TIME INTERVAL ERROR MEASUREMENT ALGORITHMS 14
3.4.1 Impact of High Frequency Modulation (fm >> fo) 14
3.4.2 Low Frequency, Multi-unit Interval Modulation (fm ≪ fo) 19
3.4.3 Long Term Average Frequency Offsets 19
3.5 PHASE NOISE FROM TIME INTERVAL ERROR 23
3.5.1 Phase Noise Background 23
3.5.2 jitterhist Methodology to Compute Phase Noise 26

4. EXAMPLES OF *JITTERHIST* OPERATION 28

4.1 PHASE MODULATED SQUARE WAVE AT 1 HZ: WITHOUT TIE SLOPE CORRECTION 28
4.2 PHASE MODULATED SQUARE WAVE AT 1 HZ: INCLUDES TIE SLOPE CORRECTION 32

5. PROGRAM INSTALLATION 35

6. SUMMARY 35

1. Introduction

Periodic timing signals are pervasive in modern electronics. Examples include the reference clock for a phase-locked loop, the sample clock for an analog-to-digital converter, the local oscillator (LO) in a modulator or demodulator circuit, processor and memory clocks, and the recovered clock generated by a clock and data recovery circuit. Common performance metrics to quantify its short term frequency stability[1] include time interval error or jitter in the time domain and phase noise in the frequency domain .

High performance digital sampling oscilloscope (DSO) manufacturers such as Keysight, Textronix, and Teledyne-LeCroy offer software packages that provide short term frequency stability measurements on captured timing signals. The packages analyze waveforms captured by the DSO and are not designed to analyze waveforms captured by other instruments or from a simulation. As a result, the ability to perform the same short term stability analysis on a captured waveform and a simulated waveform is not possible.

To avoid the cost and limitations of a DSO software package as well as provide additional flexibility in a timing measurement*, jitterhist*, was developed in the C programming language to analyze the short term frequency

[1] Other metrics, such as wander, are more appropriate for characterizing long term frequency stability.

stability of a measured or simulated periodic waveform. The command line based program computes the time interval error and phase noise of a periodic waveform and provides its outputs in both comma-separated variable file and graphical formats.

The program was developed for use with measured and simulated data and has proven insightful to understand and identify the sources of undesired components of phase noise and jitter. It is not designed to replace short term stability measurement solutions in a DSO package or simulation tool, but provides an additional analysis tool capable of customizing, automating, or studying short term frequency stability of a uniformly sampled periodic timing signal.

2. Background and Definitions

2.1 Time Interval Error (TIE)

2.1.1 Definition

When studying the temporal short term stability of a periodic signal, a common metric is its time interval error (TIE). The time interval error of a timing signal with long term frequency f_o is the difference in time between its threshold crossings and the corresponding threshold crossings of an ideal (jitter free) waveform having the same long term frequency f_o as the timing signal.

Figure 1 illustrates the definition of time interval error using a phase modulated waveform as the timing signal with a long term frequency f_o of 100 MHz. Its x-axis is normalized to the period number of the ideal waveform, and the y-axis is the amplitude relative to the threshold value of the ideal waveform.[2]

In Figure 1, the phase modulated signal is compared to an ideal (unmodulated) waveform of the same frequency f_o and ithe five positive edge based time interval errors are circled. There are two positive time interval errors, two negative time interval errors, and a single time interval error of approximately zero. Although not highlighted, there are also a set of five negative edge based time interval errors. The values of the negative and positive edge based TIE may not be the same nor have the same distributions. For a sample that contains N periods of a timing signal, there will be N positive edge based time interval errors and N negative edge based time interval errors.

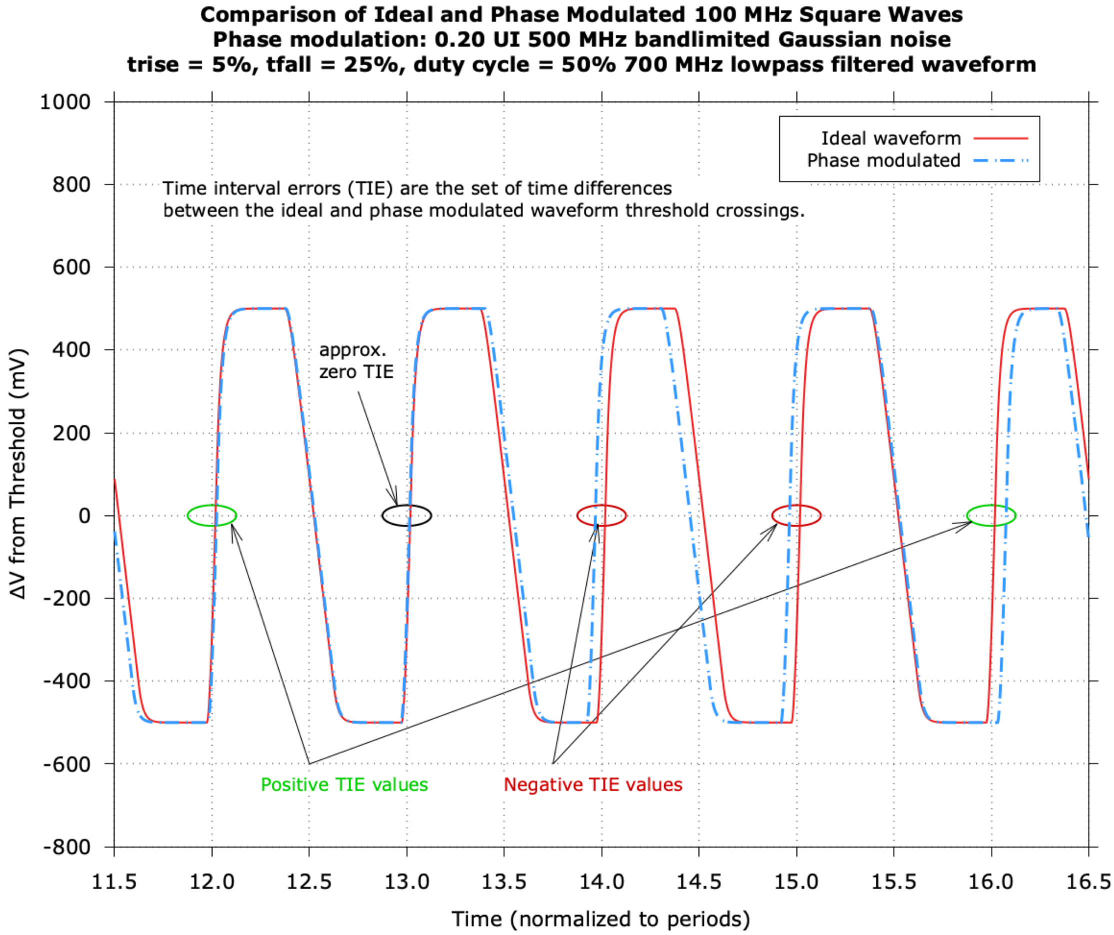
Time interval error is expressed as a either unit of time (sec) or relative to the period of the ideal waveform. In the latter case, a time interval error corresponding to 10% of the period of the ideal waveform is 0.10 unit interval or UI. When the time interval error exceeds the period of the ideal waveform, it will have a value of greater than 1.0 UI.

2.2 Measuring Time Interval Error

To measure the positive edge based time interval error at threshold crossing i, one locates the threshold value of the waveform on the i^{th} rising edge of the timing signal and takes the difference between the time of its threshold crossing and the time of the i^{th} rising edge of the ideal signal. Although this appears to be a simple computation, it does possess some subtleties that require care. Three cases are discussed: the impact of high frequency components of phase modulation (f_m >> f_o); the impact of multi-unit interval amounts of low frequency phase modulation ($f_m << f_o$); and the effect of a limited data sample when estimating the frequency of the ideal signal.

[2] In general, the switching threshold of a waveform will not be zero. Typically, the threshold is close to the average of its maximum and minimum values.

Figure 1



2.2.1 Impact of High Frequency Modulation (f_m >> f_o)

In some cases, high frequency modulation of a timing signal can obscure its actual threshold crossing instant. This effect is most common when the source of the modulation is some type of random noise or when the data samples are of a measured waveform.

For example, a measured waveform exported from a DSO will contain phase noise due to its inherent phase noise in addition to random measurement noise. If either noise contains significant frequency components close to those of its transition times, it can modulate the threshold crossings and even result in multiple threshold crossings over the course of a few waveform samples.

This phenomena is illustrated in Figure 2 with a 100 MHz square wave amplitude modulated by 1 GHz bandlimited random uniform noise with a modulation index of 0.25. The transition times of the waveform show evidence of modulation as the 1 GHz noise bandwidth is greater than the transition time bandwidths. The threshold crossing of the waveform about the 41,930th sample point is circled in Figure 2 and shown with greater detail in Figure 3 where each sample point is marked. Note that there are two positive edge threshold crossings over the course of a few sample points in the timing waveform but will only be a single positive edge threshold crossing in its ideal 100 MHz waveform. Therefore, if one were to compute the time interval error, the time interval error associated with the second positive edge threshold crossing near sample 41,930 will be about -1.0 unit interval (UI) since the next positive edge threshold crossing of the ideal 100 MHz waveform is near sample 42.930 (1000 samples in advance). Inspection of the waveform in Figure 2 indicates that a time interval error of -1.0 UI is not correct. Hence, to assure an accurate estimate of time interval error, a methodology is needed to detect and compensate for any multiple threshold crossings that occur due to noise as the waveform crosses its threshold.

Figure 2

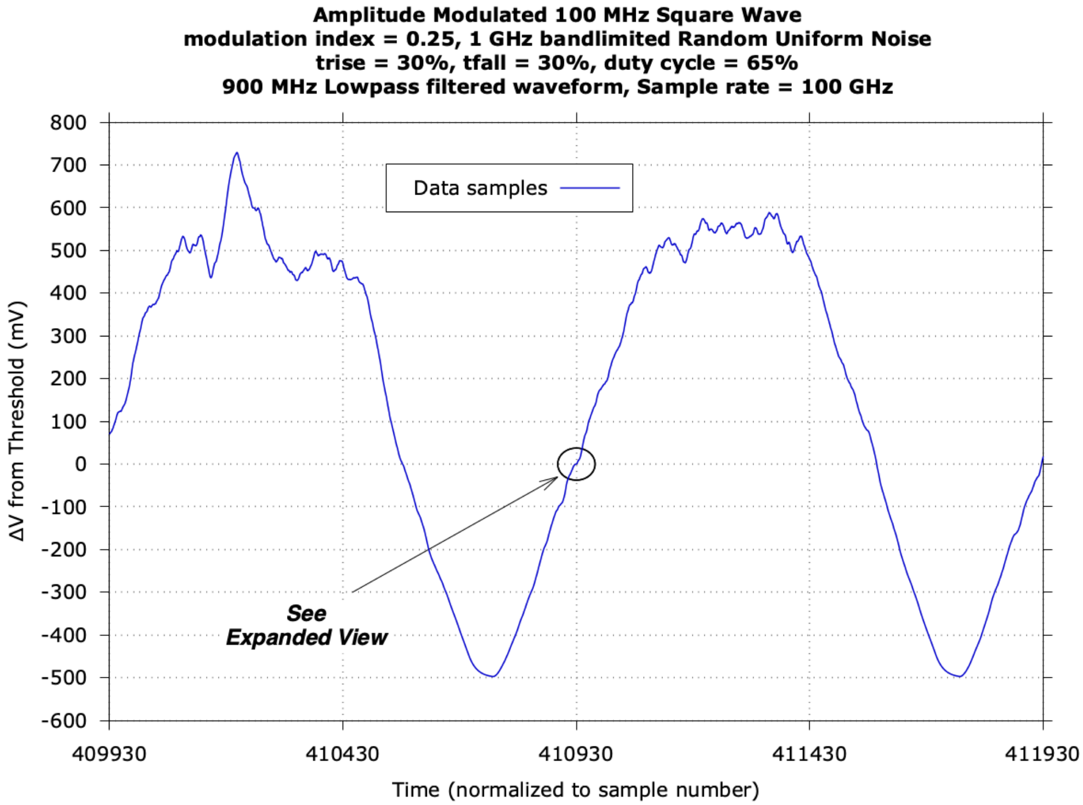
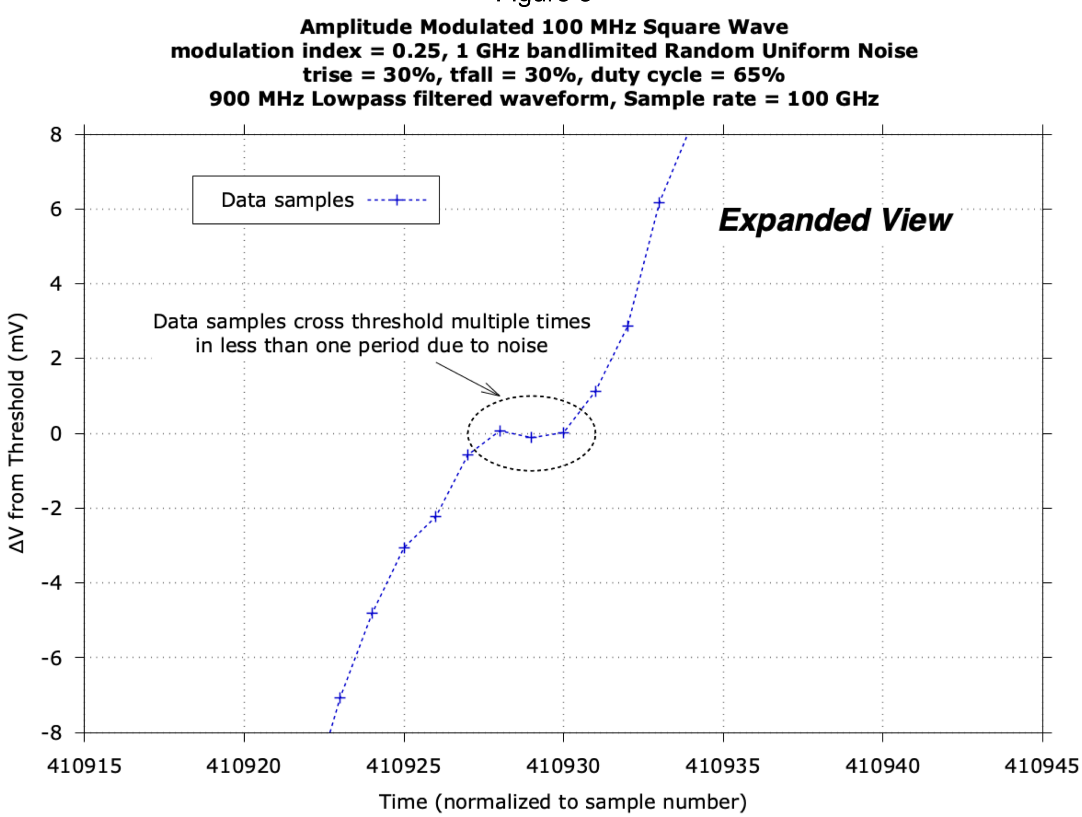


Figure 3



2.2.2 Impact of Multi-Unit Interval, Low Frequency Phase Modulation

A second time interval error measurement that requires care occurs if a timing signal of frequency f_o is phase modulated by a lower frequency f_m whose magnitude exceeds a unit interval.

The first issue to verify that the number of samples of the timing signal is sufficient to include at least one period of the lowest modulating frequency of interest. If the data sample does not include one or more periods of the modulating frequency, both the frequency and magnitude of the phase modulation cannot be accurately determined.

Secondly, the manner in which the multi-unit interval time interval error is computed requires some "bookkeeping". To illustrate its computation, a zero-mean sinusoidal signal with phase modulation is considered.

Equation [1] describes a sinusoid at frequency f_o with an initial phase φ_o that is sinusoidally phase modulated at a frequency f_m with a modulation amplitude A_m UI.

$$\begin{aligned}
&\sin\varphi(t) = \sin(\omega_o t + \varphi_o + 2\pi A_m \sin\omega_m t) \\
&where: \\
&\qquad \varphi_o = initial\ phase\ in\ radians \\
&\qquad \omega_o = 2\pi f_o \\
&\qquad A_m = amplitude\ in\ UI \\
&\qquad \omega_m = 2\pi f_m
\end{aligned} \qquad \text{Eq. [1]}$$

The phase modulation results in a frequency modulated waveform whose frequency deviation from f_o depends on both the magnitude and relative frequency of the phase modulation as shown in Equation [2]. Equation [2] suggests a limit exists for the maximum modulation amplitude A_m at a specific modulation frequency f_m as too large a value will result in an instantaneous frequency of zero.

$$\begin{aligned}
&\frac{\delta\varphi}{\delta t} = \omega_o + 2\pi A_m \omega_m \cos\omega_m t \\
&\frac{\Delta f}{f_o} = \frac{2\pi A_m f_m \cos\omega_m t}{f_o}
\end{aligned} \qquad \text{Eq. [2]}$$

The limit on the maximum phase modulation amplitude at a modulation frequency f_m to maintain an instantaneous frequency of greater than zero is derived in Equation [3].

$$\begin{aligned}
&\frac{\Delta f}{f_o} \geq -1 \\
&A_m \leq \frac{f_o}{2\pi f_m}
\end{aligned} \qquad \text{Eq. [3]}$$

Figure 4 illustrates a phase modulated sinusoid with f_o = 100 MHz, φ_o = 0, A_m = 2.0, and f_o = 3.3 MHz. In addition to the phase modulated sinusoid, Figure 4 also includes the 100 MHz unmodulated sinusoid and the phase signal $\varphi(t)$.

In Figure 4, the phase modulation $\varphi(t)$ is shown in green and its amplitude is referenced to the right hand side y-axis. The unmodulated 100 MHz sinusoid and the phase modulated waveforms are in blue and red respectively. and their amplitudes are shown on the left hand side y-axis. The x-axis spans a single phase modulation cycle of $0.303\ us$ ($\frac{1}{3.3\ MHz}$).

Consistent with Equation [3], the maximum frequency deviation from 100 MHz due to the 2 UI, 3.3 MHz phase modulation is 41.4 MHz as shown in Figure 5.

Figure 4

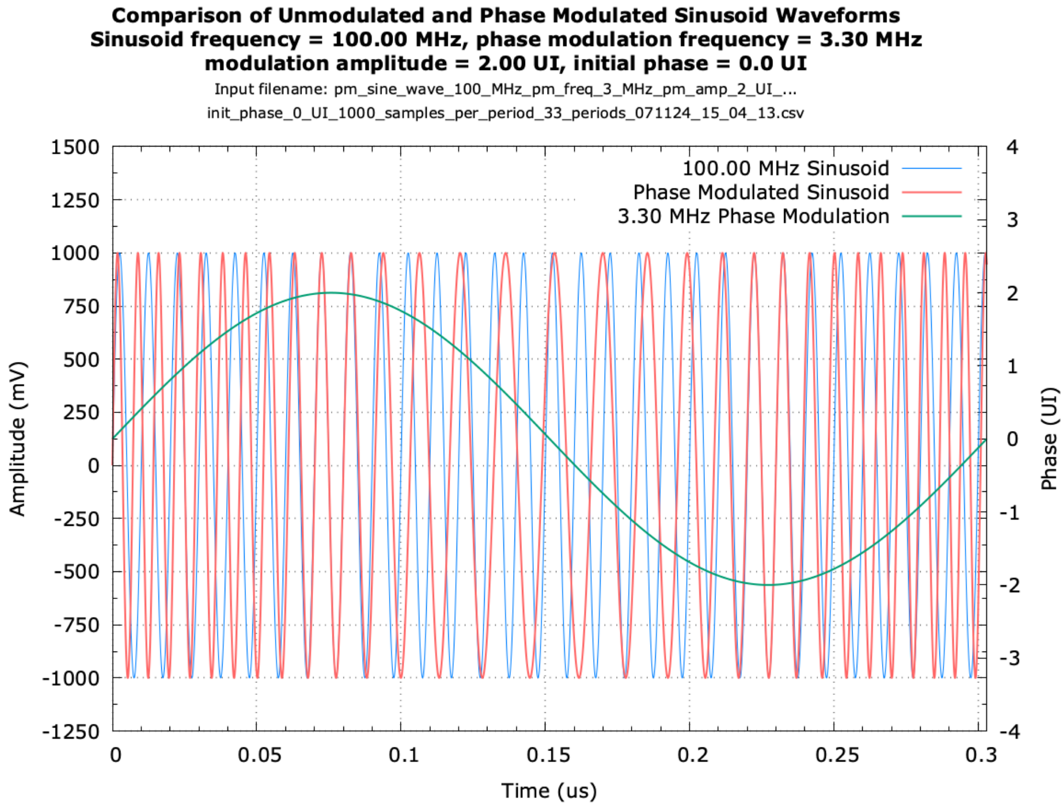
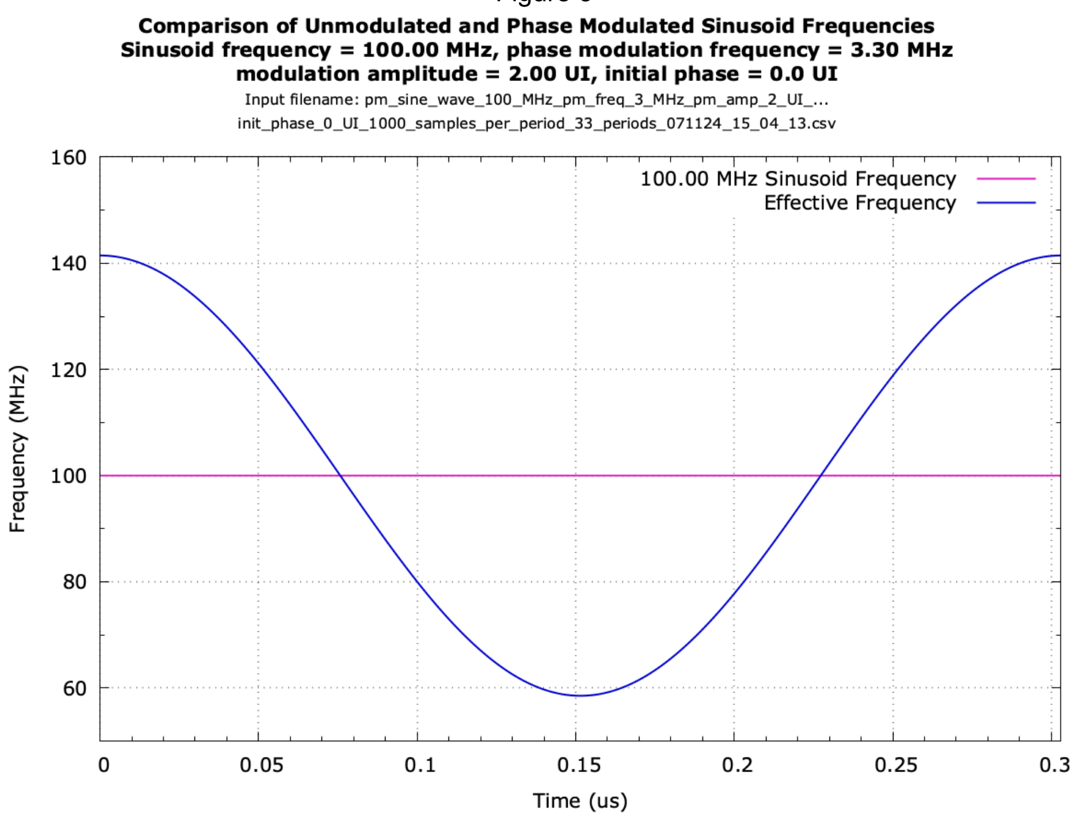


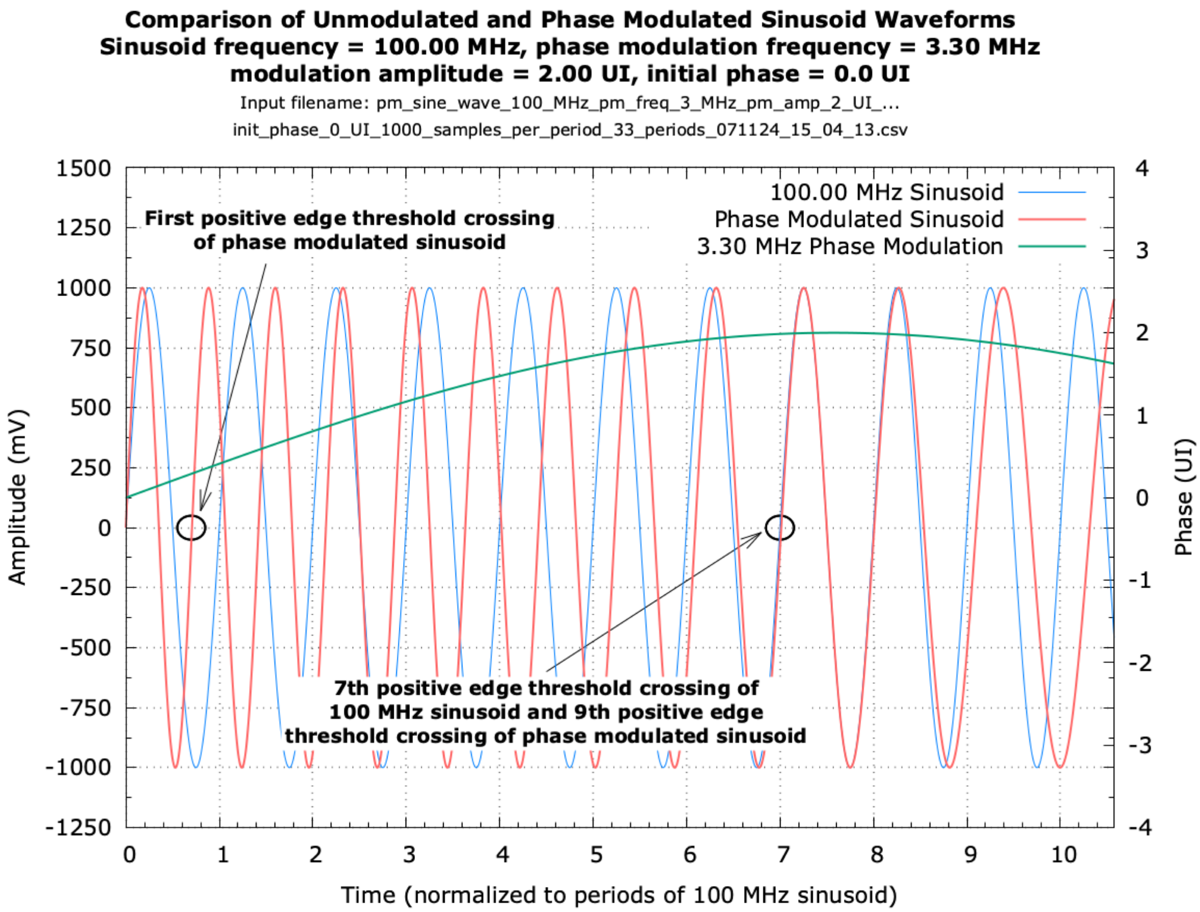
Figure 5



As the phase modulation approaches +2UI at about $0.070\ us$ in Figure 4, the phase of the ideal 100 MHz sinusoid and the phase modulated signal appear to overlap.

Figure 6 is an expanded view of Figure 4 where the time axis is normalized to periods of the 100 MHz unmodulated sinusoid. This illustrates the relative phase of the threshold crossings of the unmodulated and phase modulated waveforms more clearly. The first positive edge threshold crossing of the phase modulated waveform occurs at x = 0.7 while the first positive edge threshold crossing of the unmodulated sinusoid occurs at exactly x = 1.0 Hence, the positive edge time interval error of the phase modulated signal at 0.70 is 1.0 – 0.70 ≈ +0.30 UI.

Figure 6



However, if we compute the positive edge based time interval error of the phase modulated signal at $0.070\ us$[3] where the two signals appear to overlap in Figure 4 in the same manner, the time interval error will be approximately zero - which is incorrect. A closer examination of the ideal and modulated waveforms shows that the frequency of the phase modulated waveform increases relative to the ideal waveform between 0 and 7 of the 100 MHz unmodulated waveform and hence the 9^{th} positive edge of the ideal waveform that must be used in the time interval error calculation at x = 7 does not occur at x = 7 but is at x = 9 . Hence the correct time interval error computation for the phase modulated waveform at x = 7 is (9 UI – 7 UI) ≈ +2 UI (time of the 9^{th} threshold crossing of ideal waveform *minus* time of the 9^{th} threshold crossing of phase modulated waveform).

Therefore, to compute the time interval error, one must record both the number of each positive and negative edge threshold crossing event and its corresponding time for both the ideal and timing signal waveforms. It is not sufficient to take the difference between threshold crossing times of adjacent edges of the ideal and timing signal waveforms if the amplitude of the phase modulation is equal to or greater than 1 UI.

---

[3] The time of $0.07\ us$ in Figure 4 is 7 periods of the 100 MHz ideal waveform and corresponds to an x-axis value of 7 in Figure 6.

2.2.3 Estimating the Long Term Average Frequency

In Sections 2.2.1 and 2.2.2, the long term average frequency of the ideal waveform is referred to as f_o and is assumed to be a known quantity.

When its value is not known, it may be estimated as the average frequency, f_{ave}, of the timing signal using Equation [4].

$$f_{ave} \triangleq \frac{N}{T_S} \qquad \text{Eq. [4]}$$

where:

$$f_{ave} = \textit{average frequency of timing signal}$$

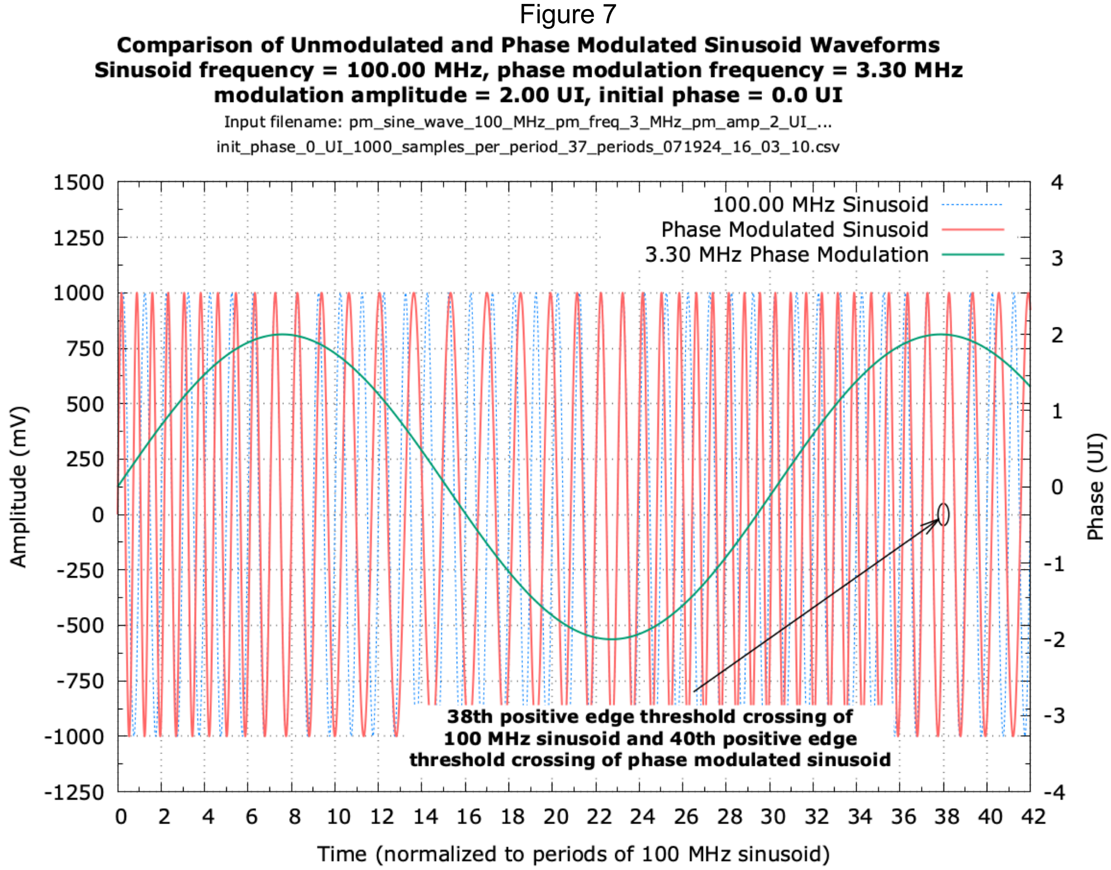
$$N = \textit{number of timing signal positive or negative edge threshold crossings}$$

$$T_S = \textit{total sample time for } N \textit{ positive or negative edge threshold crossings}$$

2.2.4 Impact of Reference Frequency Accuracy on Time Interval Error

If the timing signal contains phase modulation, the difference in time between its threshold crossings will not be constant. As was evident in the phase modulated sinusoid in Figure 6, its threshold crossings rarely align with the regular threshold crossings of an ideal 100 MHz waveform. As a result, depending on both the time span of the sample and its modulation frequency components, the average frequency will vary. For example, consider the average frequency of the phase modulated sinusoid from Section 2.2.2 between 0 and 38 in Figure 7. Using Equation [4] with N = 40 and $T_s = 38 \times \frac{1}{100MHz}$, the average frequency is $\frac{40}{38} \times 100MHz$ which is about 5% greater than the 100 MHz long term frequency f_o.

Figure 7



Since the time interval error of a waveform is computed relative to the long term average frequency, if the long term average frequency is not correct, the accuracy of the time interval error of the timing signal is compromised. Equation [5] defines the time interval error TIE(i) for the i^{th} threshold crossing of the timing signal and reference signal when the reference signal is at the long term frequency f_o of the timing signal.

Equation [6] provides insight into the impact on the time interval error when using a reference signal at frequency f_{ave} where f_{ave} differs from f_o by Δf. When f_{ave} is used to estimate the long term frequency f_o and there is a small difference between the two frequencies, the i^{th} time interval error measurement will include a linear term in i whose magnitude is a proportional to $\frac{\Delta f}{f_o}$. Since i is proportional to time, a plot of time interval error will include the sum of the actual values TIE(i) and a second term with a slope whose value is proportional to the frequency difference Δf. The presence of the slope will increase the peak-to-peak time interval error when compared to using the exact long term frequency f_{ave} to compute the time interval error.

$$TIE(i) = t_i - T_i$$

$where:$

$$t_i = i^{th}\ threshold\ crossing\ time\ of\ timing\ signal$$

$$T_{io} = i^{th}\ threshold\ crossing\ time\ of\ ideal\ reference\ signal\ with\ frequency\ f_o$$

$$TIE(i) = t_i - \frac{i}{f_o} \qquad \text{Eq. [5]}$$

$$TIE(i)_{\Delta f} = t_i - \frac{i}{f_o + \Delta f} = t_i - \frac{i}{f_o}\left[\frac{1}{1+\frac{\Delta f}{f_o}}\right]$$

$$\text{for} \frac{\Delta f}{f_o} \neq -1:$$

$$TIE(i)_{\Delta f} = t_i - \frac{i}{f_o}\left[1 - \frac{\Delta f}{f_o} + \left(\frac{\Delta f}{f_o}\right)^2 - \left(\frac{\Delta f}{f_o}\right)^3 + \ldots\right]$$

$$\text{for} \frac{\Delta f}{f_o} \ll 1:$$

$$TIE(i)_{\Delta f} \approx t_i - \frac{i}{f_o}\left[1 - \frac{\Delta f}{f_o}\right] = TIE(i) + i\frac{\Delta f}{f_o^2} \qquad \text{Eq. [6]}$$

The impact of a frequency difference between f_o, and f_{ove} for the 100 MHz phase modulated sinusoid over the 40 threshold crossings shown in Figure 7, Figure 8 shows the time interval error computed using the ideal reference frequency of 100 MHz and the average frequency of $\frac{40}{38} \times 100\ MHz = 105.26\ MHz$. Figure 9 illustrates the difference between the negative edge and positive edge TIE for the two reference frequencies. The slope of the difference waveforms is approximately equal to the 5.26 MHz frequency difference between the two reference frequencies as predicted by Equation [6].

When the reference frequency used for the TIE computation is not equal to 100 MHz, the rms and peak-to-peak TIE are greater than the TIE computed with the ideal reference frequency of 100 MHz.

Hence, an accurate estimate of the peak-to-peak or rms TIE requires an algorithm to correct the TIE for any difference between the average frequency of the timing signal and its long term average frequency. Alternately, if the number of samples of the timing signal is increased to include a greater number of modulation frequency periods, the difference between its average frequency and its long term frequency will be reduced and the accuracy of the computed TIE will improve.

Figure 8

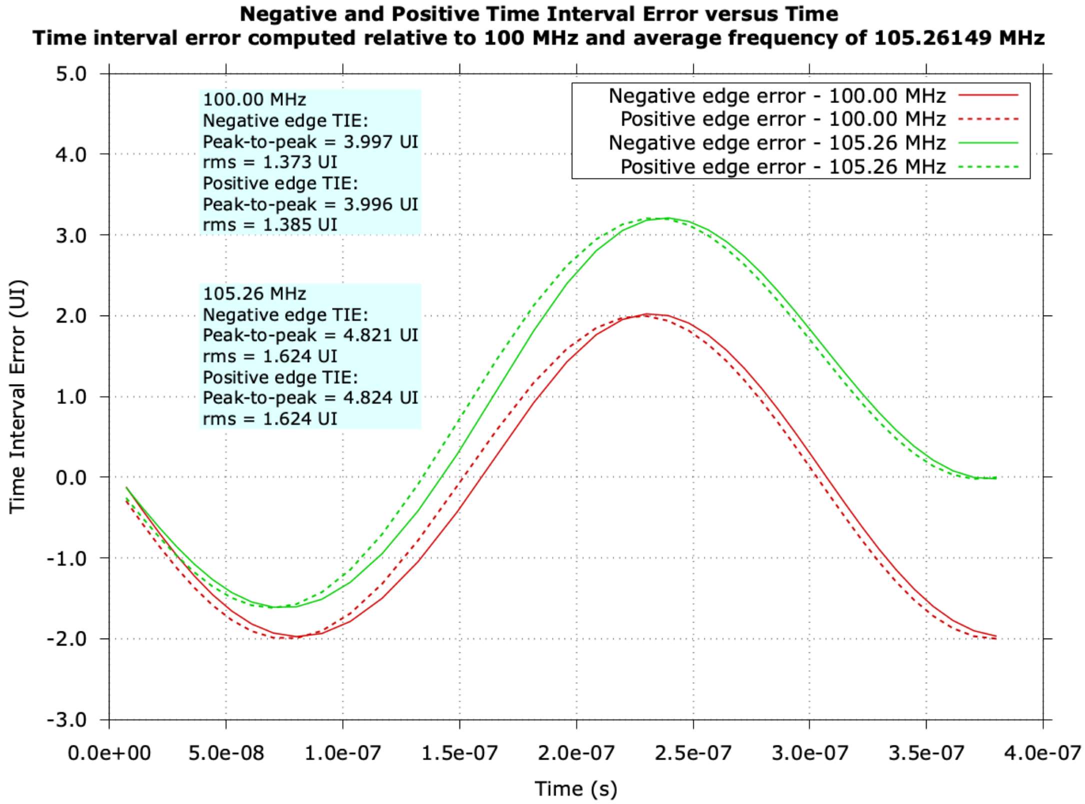
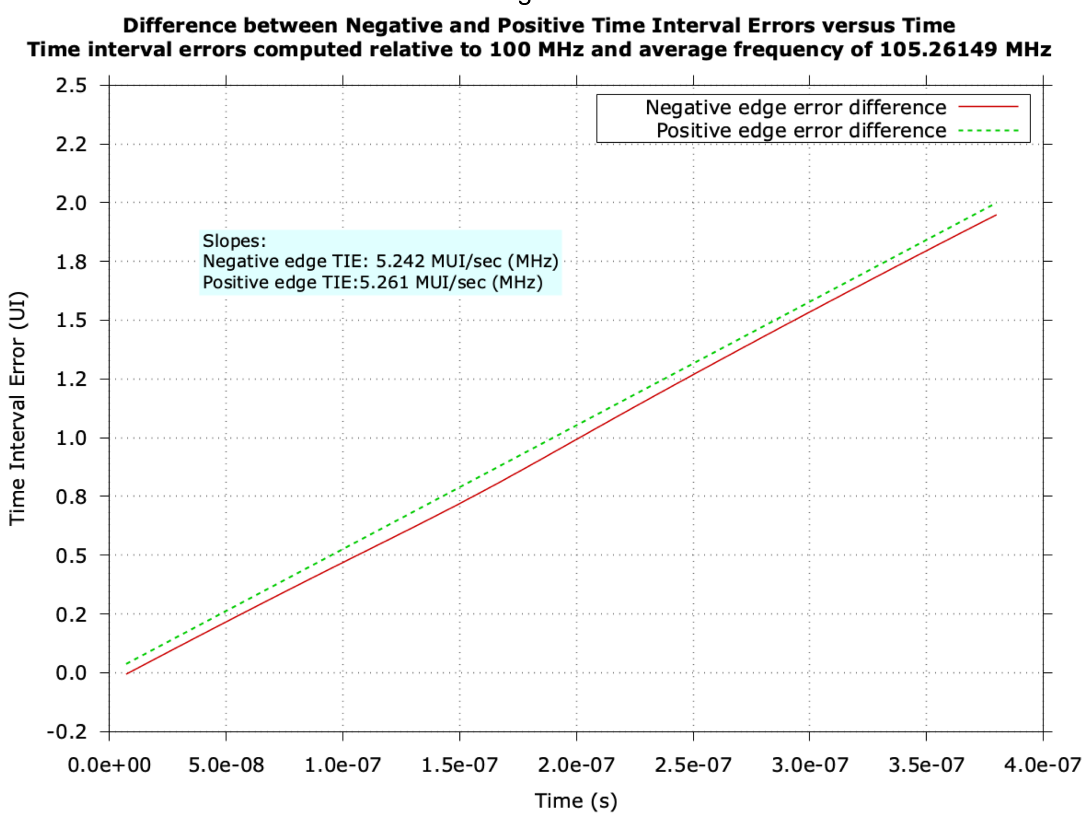


Figure 9



3. *jitterhist* Description, Inputs, and Outputs

3.1 Program Description

The custom C program, *jitterhist*, was developed to compute the time interval error of a set of uniformly spaced timing signal samples contained in a comma-separated variable file. The data samples may be in any column of the input data file, and the input file may have up to 12 columns. *jitterhist* computes the positive and negative edge based time interval error of the timing signal. If **Gnuplot**© is installed and in the UNIX search path, the program will create a plot of the time interval errors and save the plot as a Portable Network Graphics (png) file. A number of options are available to accommodate the nature of the input data and the desired analysis. *jitterhist* also computes the phase noise of the positive and negative edge based timing signal. if gnuplot is installed, the phase noise is plotted and saved to a png file.

The program has a command line interface in a UNIX based environment. If an incorrect number of command line arguments is provided, the program prints the list of its arguments and exits.

3.2 Program Inputs

Starting with version 1.80 of *jitterhist,* there are two means, or use cases, of entering input parameters.

Use case 1 is unchanged from earlier versions of *jitterhist.* Table 1 lists the command line inputs to the program and includes a brief description of each input and, when relevant, the range of its valid values. There is an additional optional input parameter in v1.80 for the window number to use in its phase noise analysis. This input is optional and may be omitted for compatibility with prior versions of *jitterhist.*

A second use case is now available where the input parameters are included in a single text file. The name of the text file is the only command line input as shown in Table 2. Use case 2 includes additional input parameters not available in use case 1. The format of the input text file is provided in Table 3. A sample text file "jitterhist_input_params.txt" is included in the Documentation folder and will be created by jitterhist when entering an incorrect number of command line entries or by issuing the command "*jitterhist* ?". The order of the input parameter lines within the file will not impact operation as the program searches the file for each initial keyword or phrase and then assigns its value.

Use case 2 allows one to optionally disable plots and algorithmically choose the optimal window function from a comma-separated list of window numbers. The algorithm integrates the phase noise computed from each window function in the list and chooses the window that sets the integrated values closest to their respective temporal TIE rms values. If the optimization feature is not selected, there is an entry for a single window number to use.

The comma-separated variable file containing the data samples and input parameters entered on the command line and the input text file are both audited for correctness. If not valid, a corrective message indicating the nature of the invalid input is displayed and the program exits gracefully.

3.3 Program Outputs

The program creates a comma-separated variable and graphics file containing the negative and positive edge based time interval errors as well as a comma-separated variable and graphics file containing the phase noise based on the negative and positive edges of the timing signal. The text and graphics output filenames are derived from name of the output file supplied on the command line (input 3 of Table 1) as shown in Table 4. If the feature to correct the slope of the time interval errors is selected (input 7 of Table 1), the filenames include the suffix "corrected" to distinguish them from filenames based on the uncorrected TIE results. When the feature to correct the slope is not selected, there will be a total of 4 output files. With the feature to correct the slope selected, there will be a total of 5 output files.

Table 1
jitterhist Input Parameters: Use Case 1

***jitterhist* Input Parameters: Use Case 1**		**Range of values**	**Units**	**Comments**
1	Comma-separated variable filename containing N data samples	< 164 characters		Input file may include any number of comment lines
2	Column number containing N data samples	between 1 and 12		
3	Output filename containing TIE results	< 164 characters		Comma-separated variable output file contains columns with time (sec) and negative/positive edge TIE
4	Sample rate in GHz	-	GHz	
5	Threshold voltage	Between minimum and maximum signal amplitudes	Volts	
6	Number of moving average samples (S)	$0 \leq S \leq 10\%$ of sample size N		Value of 0 does not use a moving average to estimate threshold crossing times
7	Correct for linear slope in TIE?	Yes (y) or No (n)		Yes: Computes optimum linear slope from TIE, removes slope and places in corrected TIE file. No: Makes no changes to computed TIE
8	Use average frequency to compute TIE?	Yes (y) or No (n)		Yes: Computes the TIE using the average frequency. No: Requires a eighth input with the frequency to use to compute the TIE expressed in MHz.
9	Enter frequency in MHz to use for TIE computation	-	MHz	Value is ignored unless input 8 is a "No".
10	Window number to use for phase noise analysis (optional input)	1 - 7		If no entry, default value is 1 (Rectangular window)

Table 2
jitterhist Input Parameters: Use Case 2

***jitterhist* Input Parameter: Use Case 2**		**Range of values**	**Units**	**Comments**
1	Text file containing list of jitterhist parameters or a single '?' to create sample input file	< 164 characters		Input file may include any number of comment lines and includes additional parameters not available in use case 1 in Table 1. Format of text file

Table 3
Sample Input File “jitterhist_input_params.txt” Containing *jitterhist* Input Parameters for Use Case 2

```
* Comment lines begin with * or do not contain a colon
* Enter a value for each input after the colon delimiter
filename_with_data_samples_(csv format): <Enter your csv input file with data samples>
column_number: <Enter column number of your csv input file with data samples>
output_csv_filename: <Enter your csv output filename with TIE results>
sample_rate_in_GHz: <Enter the sampling rate of your data samples in GHz>
signal_threshold_(V): <Enter the logic threshold in Volts>
num_moving_average_samples: <Enter the number of moving average samples to use for TIE computation (set to 0 for none)>
correct_slope?_(y/n): <Enter yes to correct slope of computed TIE or no to leave uncorrected>
Use_ave_frequency?_(y/n): <Enter yes to compute TIE using the computed average frequency or no to enter a frequency in MHz>
Average_frequency_to_use_if_no_(MHz): <If above entry is a no, enter the frequency in MHz to compute TIE>
Optimize_window_function?_(y/n): <Enter yes to optimize window used to compute phase noise or no to use a single window>
*
* Window numbers for available window types:
*
* Rectangular 1
* Tapered rectangular 2
* Triangular 3
* Hanning 4
* Hamming 5
* Blackman 6
* Blackman-hanning 7
*
List_of_window_numbers_for_optimization_if_yes: <Enter comma separated list of window numbers to use for optimization if optimization entry is a yes>
Window_number_if_no: <Enter window number if optimization entry is a no>
plot_outputs?_(y/n): <Enter yes to enable plots or no to disable plots>
```

Table 4
jitterhist Output Files

		jitterhist Output Filename	**Type of file**
	1	<output_filename>.csv	Text (CSV)
Slope correction not selected	2	<output_filename>.png	Portable Network Graphic
	3	<output_filename>_seg_sublength_1_num_segments_1_ovrlap_percent_0_window_7_pnoise.csv	Text (CSV)
	4	<output_filename>_seg_sublength_1_num_segments_1_ovrlap_percent_0_window_7_pnoise.png	Portable Network Graphic
Slope correction selected	5	<output_filename>_corrected.csv	Text (CSV)
	6	<output_filename>_corrected.png	Portable Network Graphic
	7	<output_filename>_corrected_seg_sublength_1_num_segments_1_ovrlap_percent_0_window_7_pnoise.csv	Text (CSV)
	8	<output_filename>_corrected_seg_sublength_1_num_segments_1_ovrlap_percent_0_window_7_pnoise.png	Portable Network Graphic

3.4 *jitterhist* Time Interval Error Measurement Algorithms

Section 2.2 outlined three factors that impact the accuracy of time interval error measurements: high frequency phase modulation effects on threshold voltage crossings; multi-unit interval low frequency modulation; and an offset between the actual and estimated long term average frequency of the timing signal. This section outlines algorithms in *jitterhist* to allow for each of these factors.

3.4.1 Impact of High Frequency Modulation (f_m >> f_o)

Several possible approaches were considered to accommodate timing signals whose threshold voltage crossings are modulated by high frequency noise. The method found to be most robust and computationally efficient includes a metric to detect if the threshold crossings are impacted by high-frequency noise and a moving average algorithm to estimate the threshold crossing in the presence of a “noisy” threshold crossing.

3.4.1.1 Detection Mechanism

In Figure 3, the rising edge of the timing signal crosses the voltage threshold, falls below the threshold on the following sample, and then resumes its rising slope and crosses the threshold a third time. As a result, the on-time between these two rising edge threshold crossings is the difference between the second and first threshold crossings or the sample time of 10 ps. This creates a duty cycle measurement of 10 ps divided by the long term period of the 100 MHz timing signal or 0.001%. If the noisy threshold crossings were to occur in the falling edge of the timing signal, the off-time would be 10 ps and the resultant duty cycle 99,999%.

jitterhist measures the duty cycle of each period of the timing signal and records both the maximum and minimum duty cycle values for the entire waveform sample. When the minimum duty cycle falls below 5% or the maximum duty cycle exceeds 95%, *jitterhist* assumes the threshold crossings occur too quickly and must be re-analyzed to estimate the actual threshold crossings.

3.4.1.2 Estimating the Threshold Crossing Time

In general, given a set of two data points (x1,y1) and (x3,y3), if one wants to determine the value of y2 corresponding to x2 where x1 ≤ x2 ≤ x3, some form of interpolation is utilized. With only two known data points, linear interpolation is commonly used to estimate y2.

In Figure 3, the use of two data points is not sufficient as it is not clear which two data points to choose for the interpolation. To estimate the actual threshold crossing requires something greater than two data points. Basically, a number of (x,y) data points are required to average the zero-mean high frequency noise that results in multiple threshold crossings.

jitterhist applies a moving average algorithm to the data samples to create a waveform from which the threshold crossings are computed using linear interpolation. This algorithm is numerically equivalent to performing linear interpolation but more efficient. With the moving average set to a value of S and with N input data samples, the moving average waveform will have N - 2S samples. This is illustrated in Figure 10.

Figure 10
Algorithm to Compute Moving Average of N Input Data Samples
with the Number of Moving Average Samples set to S

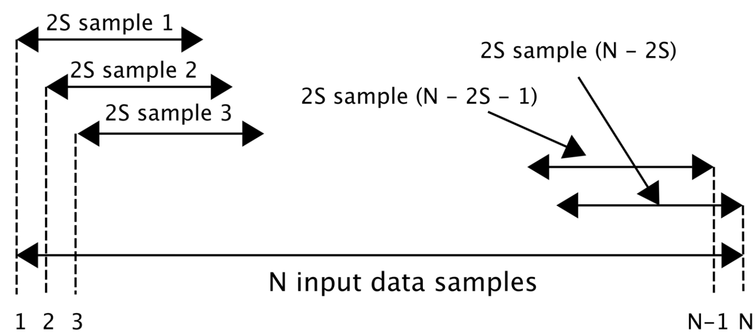


Figure 11 provides the timing waveform shown in Figure 2 and the moving average waveform created by *jitterhist*. In this case, the number of moving averages, S, was set to 12. Figure 12 is an expanded view of Figure 11 whose y-axis is normalized to the switching threshold. The moving average waveform crosses the switching threshold smoothly with a single switching threshold crossing event while the actual data samples have multiple threshold crossings between adjacent data samples.

Figure 11

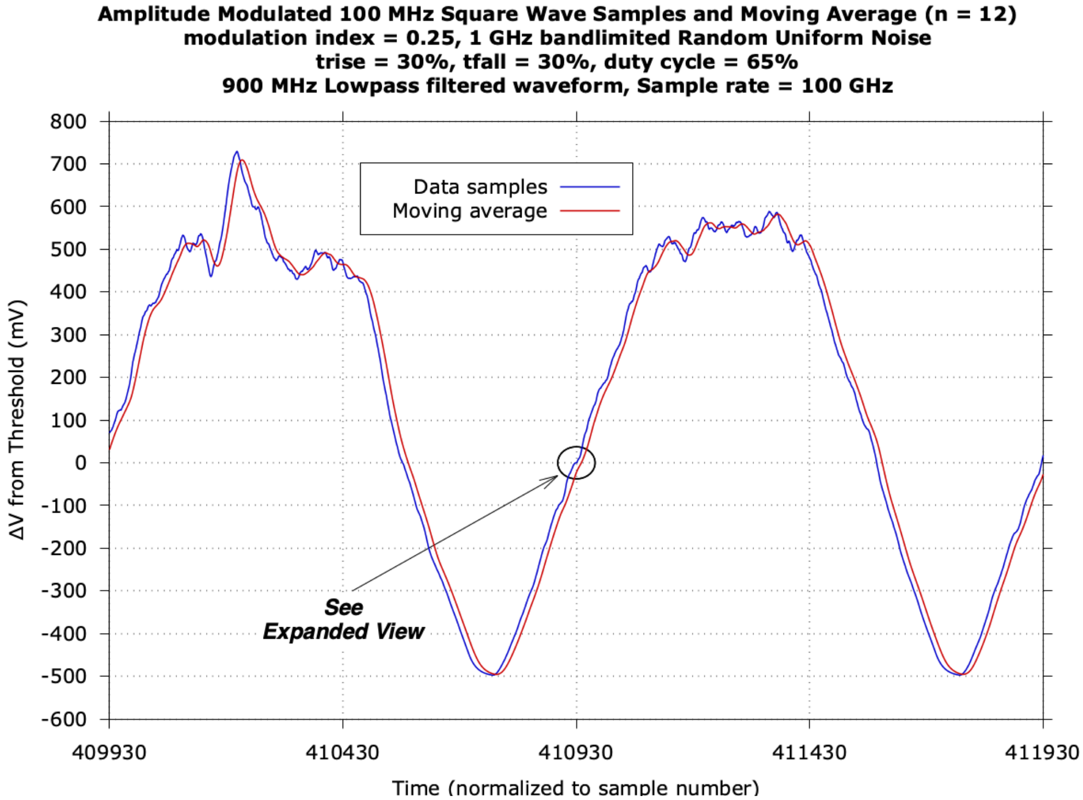
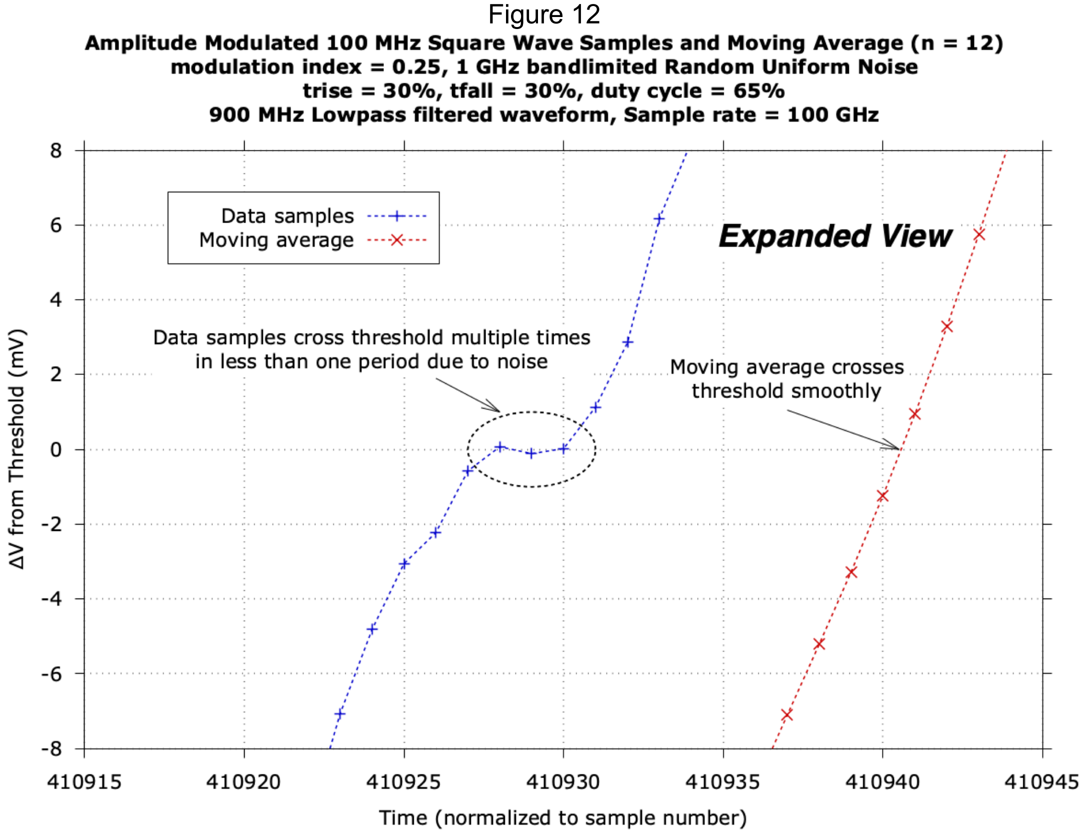


Figure 12



3.4.1.3 Combining the Detection and Threshold Crossing Algorithms

When the number of moving average samples, S, is set to zero, the moving average algorithm is not used to estimate threshold crossings. All threshold crossings are linear interpolated between the data samples.

If the number of moving average samples is set to a non-zero value, *jitterhist* combines both the detection algorithm and the threshold crossing algorithm in a single iterative feedback loop. A moving average waveform using 2S samples is created from the data samples and its time interval error is computed. If the maximum duty cycle exceeds 95% or the minimum duty cycle is less than 5%, the number of moving averages is increased from S to S + 1 and the time interval error and maximum/minimum duty cycles are re-computed.

For a non-zero moving average value of S and N input data samples, S is increased until any one of the following conditions is detected.

1. The number of moving averages is S + 20
2. Both the maximum and minimum duty cycle fall below 95% and above 5% respectively
3. If S becomes 10% of the output data sample size or $S \geq \frac{N}{12}$.

As an example of the terminal output during this iterative process, *jitterhist* was used to compute the time interval error of the 100 MHz amplitude modulated square wave shown in Figure 13. The number of moving averages, S, was set to 1 on the command line. Figure 14 shows the terminal output during execution as the algorithm increments the number of moving average samples from 1 to 6 to provide a more accurate estimate of the waveform threshold crossing times.

Figure 13

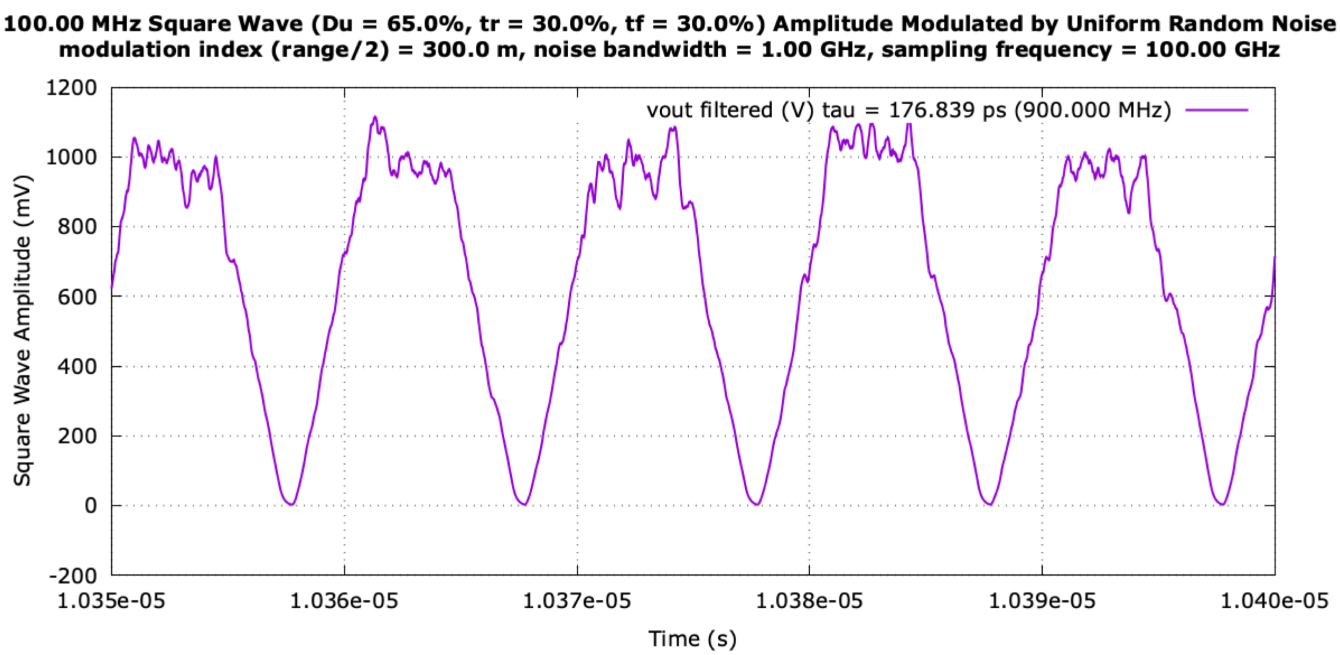


Figure 14
jitterhist Terminal Output During Execution

```
jitterhist v1.80 5/11/2025

Input file:
square_wave_100_MHz_ttran_rise_30_fall_30_percent_du_65_am_unoise_amp_300m_noise_bw_1_GHz_073124_19_52
_16.csv
Column number: 2
Output file: vout_jitter.csv
Sample frequency is 100.0000 GHz.
Threshold value is 500.000 mV.
Number of moving average samples = 1.
Entered average frequency is 100.0000 MHz.
First threshold crossing at time 9.31550801e-09 sec, interpolated_value is 5.00000000e-01.
Threshold crossings occur too quickly, increasing number of moving average samples to 2.
First threshold crossing at time 9.30533798e-09 sec, interpolated_value is 5.00000000e-01.
Threshold crossings occur too quickly, increasing number of moving average samples to 3.
First threshold crossing at time 9.29461100e-09 sec, interpolated_value is 5.00000000e-01.
Threshold crossings occur too quickly, increasing number of moving average samples to 4.
First threshold crossing at time 9.28364280e-09 sec, interpolated_value is 5.00000000e-01.
Threshold crossings occur too quickly, increasing number of moving average samples to 5.
First threshold crossing at time 9.27248028e-09 sec, interpolated_value is 5.00000000e-01.
Threshold crossings occur too quickly, increasing number of moving average samples to 6.
First threshold crossing at time 9.26112987e-09 sec, interpolated_value is 5.00000000e-01.
Read a total of 1039 periods in file
"square_wave_100_MHz_ttran_rise_30_fall_30_percent_du_65_am_unoise_amp_300m_noise_bw_1_GHz_073124_19_5
2_16.csv"
using a threshold value of 500.0 mV.
Average time period = 1.000009e-08 sec, frequency = 99.99907467 MHz.
Minimum period = 9.601901e-09 sec (104.14604863 MHz at 6.719140 us).
Maximum period = 1.040797e-08 sec (96.08018457 MHz at 1.239600 us, delta = 8.0659%).
Average on time = 6.4770e-09 sec, Duty cycle = 64.77 %.
Minimum on time = 6.0714e-09 sec (Duty cycle = 60.71 % at 7.619380e-06 sec).
Maximum on time = 6.8885e-09 sec (Duty cycle = 68.88 % at 3.809350e-06 sec, delta = 8.17%).
The calculated average frequency is 99.99907467 MHz.
Using an average frequency of 100.00000000 MHz to compute jitter.
Negative edge location variation = 605.254 ps (60.525 mUIpp +/- 1.00 mUI).
Positive edge location variation = 597.829 ps (59.783 mUIpp +/- 1.00 mUI).
Removing effect of residual slope in negative edge TIE:
slope_neg_edge = 2.254201e+02, intercept_neg_edge = 9.290174e-03
New negative edge based estimated frequency is 99.99977458 MHz.
Removing effect of residual slope in positive edge TIE:
slope_pos_edge = 9.771701e+01, intercept_pos_edge = 9.964680e-03
New positive edge based estimated frequency is 99.99990228 MHz.
Corrected positive and negative frequencies differ by 1.3 ppm (considered acceptable).
Slope corrected filename is "vout_jitter_corrected.csv".
Corrected negative edge location variation = 601.626 ps (60.163 mUIpp +/- 1000.00 uUI).
Corrected positive edge location variation = 597.399 ps (59.740 mUIpp +/- 1000.00 uUI).
Phase noise analyses complete and used 1 segment of size 1024 and each contains 512 output data
points.
Completed writing 512 data points to phase noise output file
"vout_jitter_corrected_seg_sublength_1_num_segments_1_ovrlap_percent_0_window_1_pnoise.csv".
Elapsed time: 34.375660 seconds
```

3.4.2 Low Frequency, Multi-unit Interval Modulation ($f_m \ll f_o$)

As discussed in Section 2.2.2, if a timing signal with long term average frequency f_o includes low frequency phase modulation that spans more than a single unit interval of its long term average frequency f_o, the time interval error computation will be incorrect if the time interval error computation uses time adjacent edges of the ideal and timing signal.

jitterhist allows for time interval error that spans multi-unit intervals by using separate transition edge arrays for the ideal signal and the timing signal. The index of each array for a specific threshold crossing time corresponds to its threshold crossing number. Arrays exist for both positive negative edge transitions. As a result, the time interval error of the i^{th} positive edge of the timing signal with respect to its ideal waveform is the difference between the i^{th} element of its positive edge transition time array and the i^{th} element of its ideal waveform positive edge transition time array. This avoids the potential for choosing the incorrect transition time of the timing signal or its ideal waveform when the phase modulation exceeds one unit interval.

3.4.3 Long Term Average Frequency Offsets

Since computing time interval error is a direct function of the long term average frequency of the timing signal's ideal waveform, any frequency offset between the estimated and true long term average frequency estimate will impact time interval error accuracy. *jitterhist* includes two algorithms to address an offset between the long term average frequency estimate and its actual value. The algorithms are optional and are enabled by setting its seventh command line argument, ("Correct for linear slope in TIE?", to "yes" in Table 1).

As shown in Section 2.2.3, when an offset exists between the average frequency used to compute the time interval error and its true long term average frequency, the time interval error will be the sum of the actual time interval error and a linear term dependent on the frequency offset and threshold crossing index. Graphically. the time interval error as a function of time will contain a positive or negative slope whose value is a function of the frequency offset.

Using the 100.0 MHz sinusoid phase modulated with 2 UI of 3.3 MHz sinusoidal modulation from Section 2.2.4 as an example, if the long term average frequency is assumed to be 100.5 MHz in lieu of the actual 100.0 MHz, the time interval error includes a positive slope as shown in Figure 15. The slope is estimated by inspection to be about 4 UI over 800 UI or:

$$\frac{4}{800} = 0.005\ (5000\ ppm)$$

and is consistent with the 5000 ppm frequency offset between the 100.5 MHz frequency used to compute the time interval error and the actual long term average frequency of 100 MHz.

To compensate for an offset between the ideal waveform's average frequency and its true long term average, *jitterhist* computes the slope and intercept of the time interval error samples and removes the linear term from the time interval error.

Although this solution appears to remove the error term due to an offset frequency, it is not robust with respect to the time span and modulation frequency content of the input data. If the time span of the samples is not sufficient to include a number of periods[4] of its low frequency phase modulation, the average value of the time interval error will be non-zero. The non-zero average will appear as a linear frequency term and will be removed from the time interval error in an effort to remove any offset between the ideal waveform's average frequency and its true long term average. However, this operation can increase the peak-to-peak time interval error over the time span of the input timing signal samples.

---

[4] The time span of the timing signal samples must include at least one period of any significant low frequency modulation to provide an accurate time interval error estimate.

This effect is evident in Figure 16 where the time interval error is computed on a data sample of a 100 MHz phase modulated sinusoid that spans 1.75 periods of its 3.3 MHz 2UI phase modulation. The time interval error is initially computed using a long term average frequency of 100.5 MHz and is shown in solid red (negative edge) and solid green (positive edge). The slope corrected version where the linear term is removed from the time interval error is shown is dashed curves. Note the corrected frequency has a greater difference from the exact 100 MHz long term average frequency than 100.5 MHz, and the peak-to-peak time interval error is increased from about 4.2 UIpp to about 4.4 UIpp.

To more accurately correct for any slope in the time interval error, *jitterhist* includes an optimization step following the initial slope correction. The initial estimate of the slope is used as the basis for the range of a two-step search that determines the linear correction polynomial that minimizes the peak-to-peak time interval error. The result of using this algorithm is shown in Figure 17. The long term average frequency is now estimated as 99.989 MHz and is within 200 ppm of the actual 100 MHz long term average frequency. The peak-to-peak time interval error is about 4 UIpp which accurately estimates the 2 UI phase modulation amplitude.

If, however, the time span of the timing signal does not include at least one period of its significant modulation frequencies, neither this algorithm nor any other algorithm can compensate for the offset between its estimated average frequency and true long term average frequency. As an example, if the time span includes ¾ of the 3.3 MHz modulation period in lieu of the time span of 1¾ 3.3 MHz modulation periods used for the analysis in Figure 16 and Figure 17, *jitterhist* estimates a corrected long term average frequency of far less than its true 100 MHz value and incorrectly predicts the peak-to-peak time interval error. The result of this analysis is shown in Figure 18.

If the true average long term frequency is known and used with *jitterhist* without any slope correction, the time interval error results are consistent with the expected results. This is shown in Figure 19.

This example illustrates the importance of choosing a timing signal sample length that includes as many periods as possible of its lowest significant modulation frequency

Figure 15

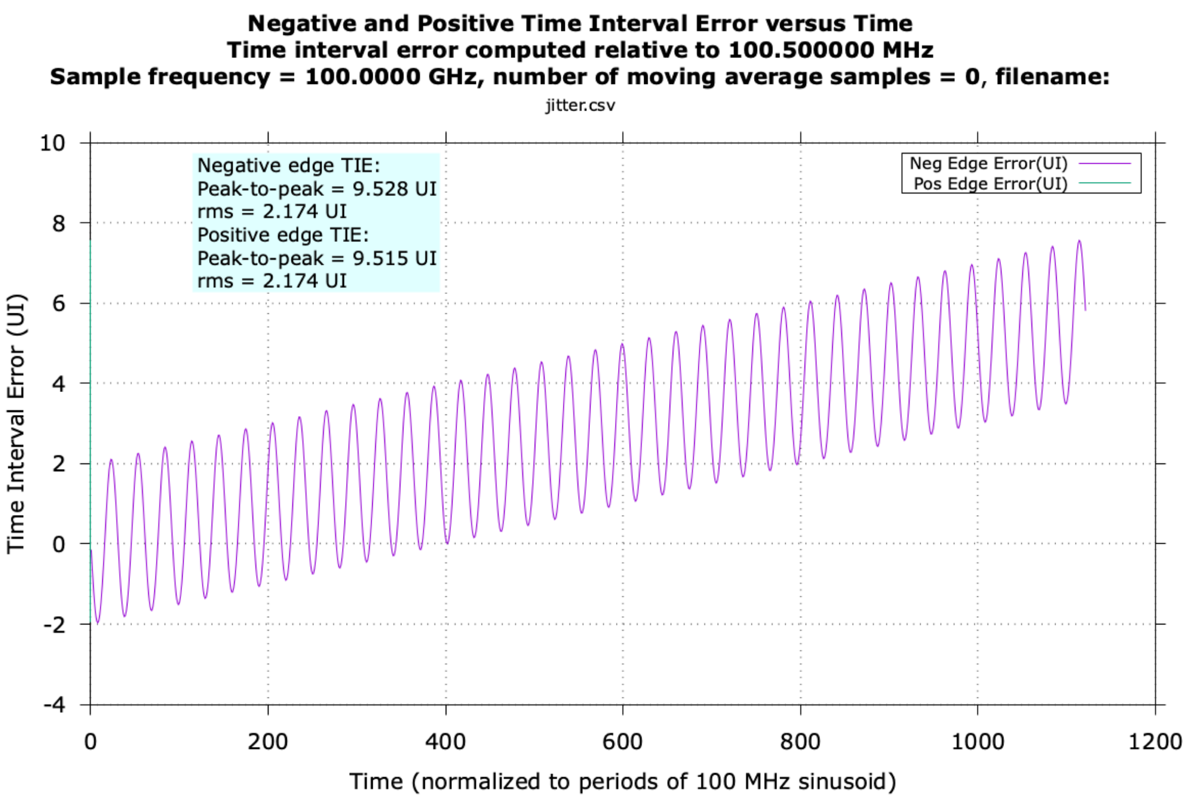


Figure 16

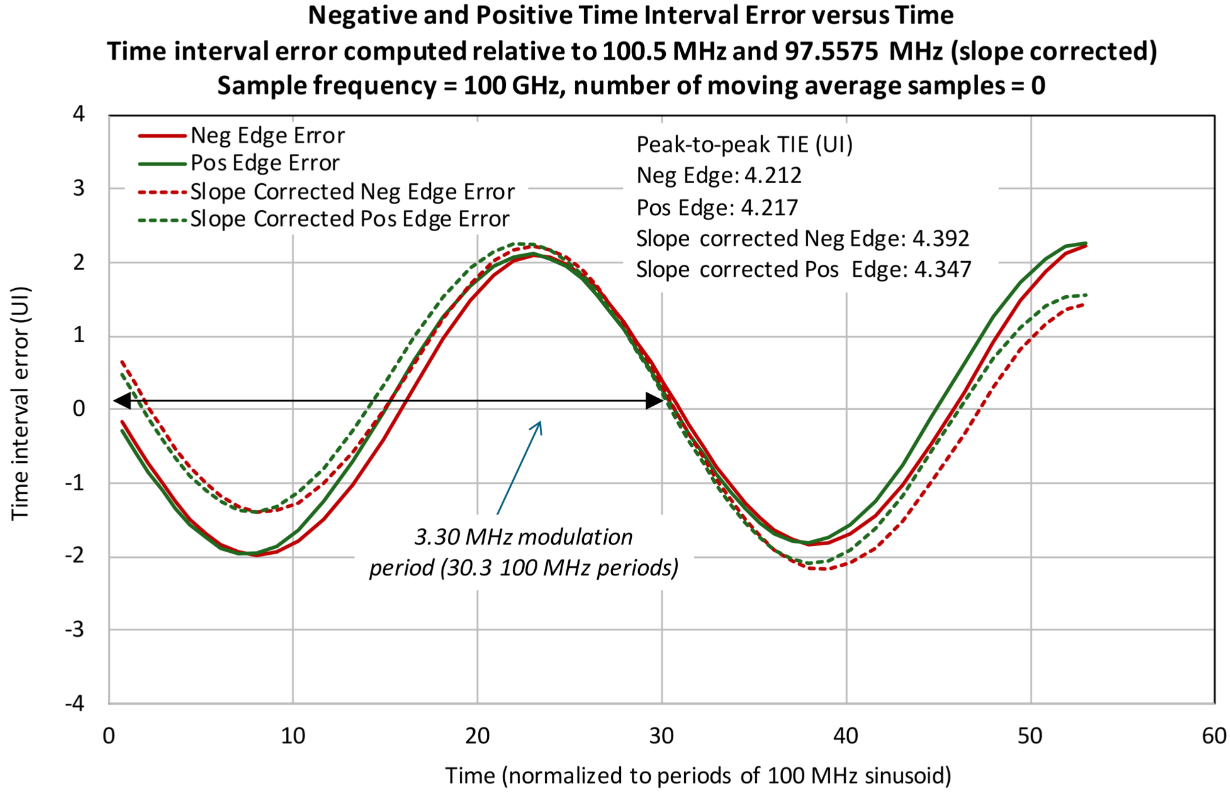


Figure 17

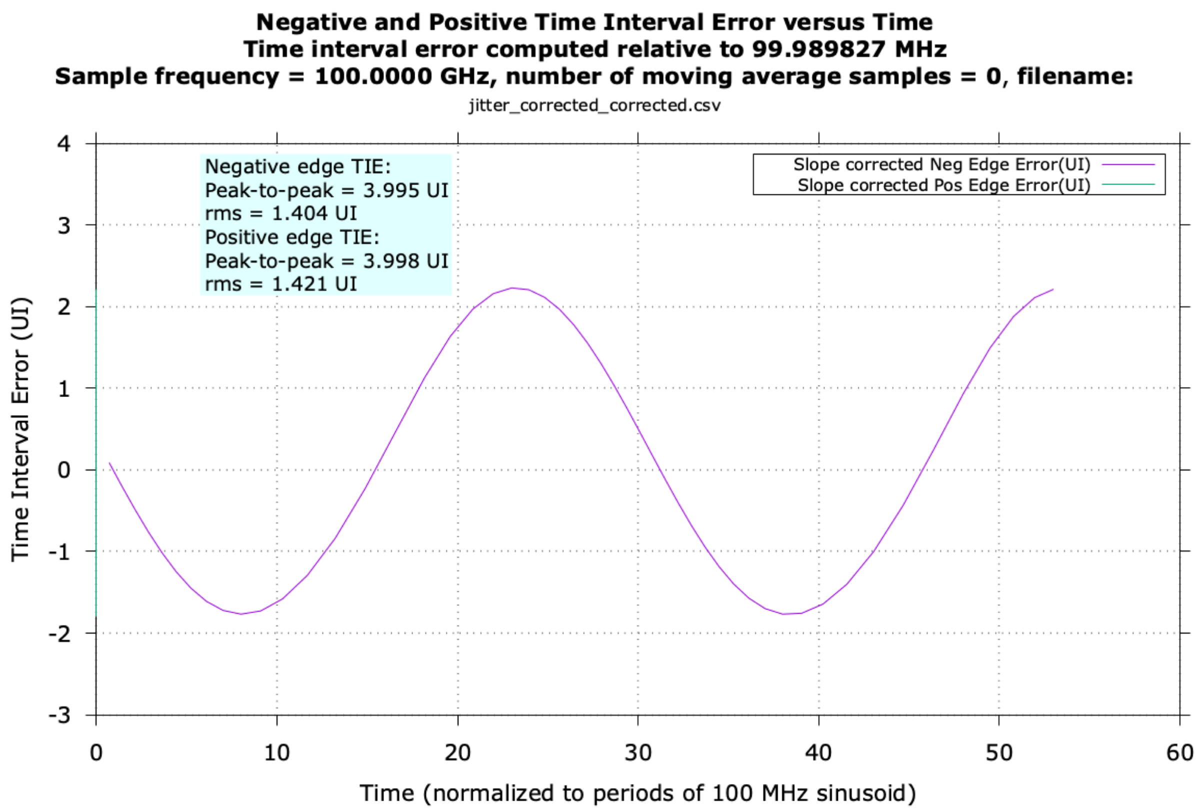


Figure 18

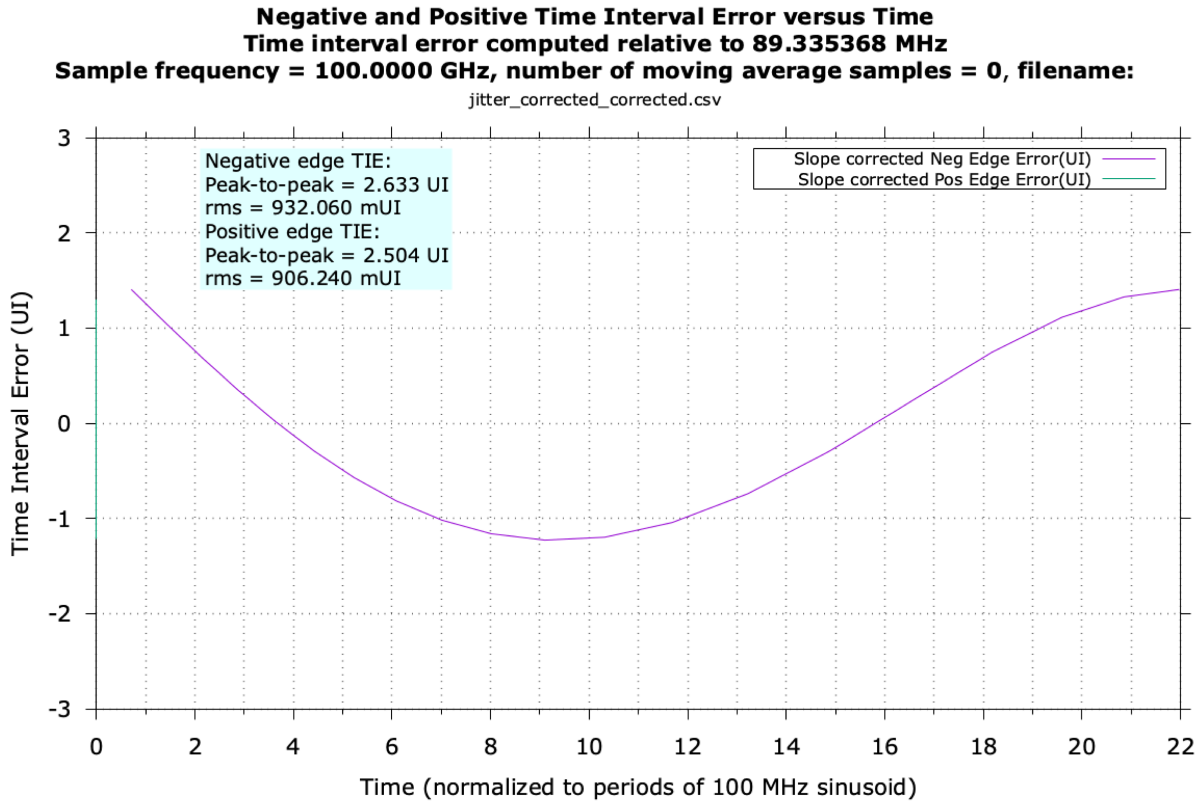
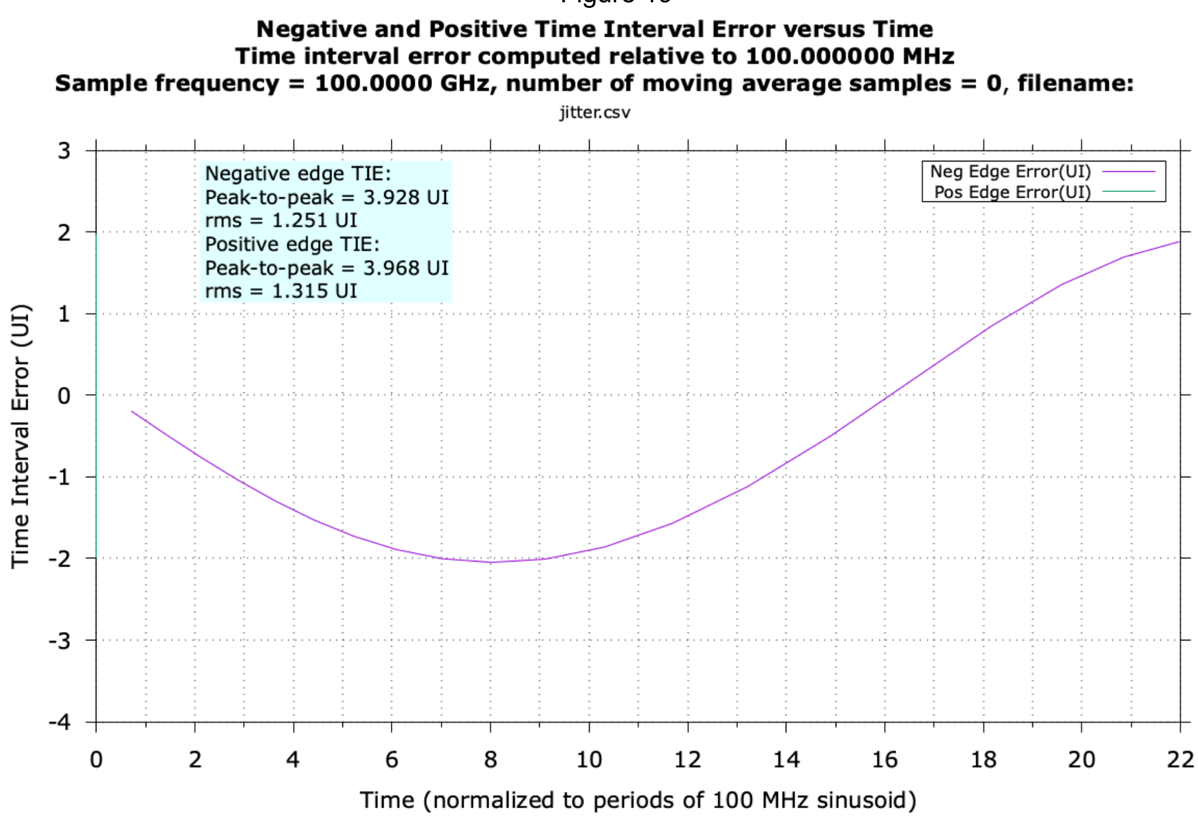


Figure 19



3.5 Phase Noise from Time Interval Error

3.5.1 Phase Noise Background

The terms phase and frequency noise are used to describe the undesirable variation in an oscillator's period *T* from its long term average period T_o due to noise effects. The terms first appear in the literature circa 1947 in conjunction with Doppler radar and later in the mid 1950 timeframe in numerous other applications (reference [1] Table II).

In the frequency domain, measures of the frequency stability include the spectral density of frequency fluctuations $S_{\Delta F}(f)$ and the spectral density of phase fluctuations $S_{\phi}(f)$.

The former is a measure of the mean square frequency variation in a 1 Hz bandwidth at an offset frequency *f* from the carrier and has the units of Hz^2/Hz. The latter is the mean square phase variation in a 1 Hz bandwidth at an offset frequency *f* from the carrier, and its units are $(radians)^2/Hz$. The relationship between these two measures is shown in Equation [7].

$$S_{\phi}(f) = \frac{S_{\Delta F}(f)}{f^2} \qquad [7]$$

For the specific case where the integral of $S_{\phi}(f)$ between frequencies f and greater is much less than 1 radian, one may use the fact that the Bessel function coefficients for narrow band phase modulation ($J_n(\beta)$, $|n| > 2$) are approximately 0. In this case, $S_{\phi}(f)$ is *approximately* the double-sideband noise-to-carrier ratio. The specific approximation is shown in equation [8].

$$\mathcal{L}(f) \approx 10\mathrm{Log}10\left[\left(\frac{1}{2rad^2}\right)S_{\phi}(f)\right] \ \phi \ll 1 \text{ radian} \qquad [8]$$

The expression for the ratio of the power in one phase noise sideband on a per Hertz basis in the units of dBc/Hz, $\mathcal{L}(f)$, is defined by Equation [9] and represents the IEEE preferred means of characterizing phase noise (references [2], [3]). It is very important to emphasize that this relationship is only valid for $\emptyset \ll 1$ radian.

$$\mathcal{L}(f) = 10\mathrm{Log}10\left(\frac{S_{\phi}(f)}{2}\right) \qquad [9]$$

Figure 20, Figure 21, and Figure 22 are provided to illustrate the accuracy of the approximation. Figure 20 compares the power spectral densities of a 1 Hz square wave phase modulated by a sinusoid at 10 mHz. The modulation amplitude is varied from 0.10 mUI to 500 mUI. When demodulated, sidebands of the 1 Hz carrier represent $S_{\phi}(f)$. An expanded view of Figure 20 about the square wave frequency of 1 Hz is shown in Figure 21. Note that as the amplitude of the modulation is increased, the number of sidebands spaced 10 mHz apart is increased. In addition, at the largest modulation amplitude of 500 mUI, the peak amplitude at 1 Hz is *less* than the amplitude of its adjacent sidebands. These observations are consistent with the Bessel coefficients that describe the amplitudes of phase modulation sidebands.

Figure 20

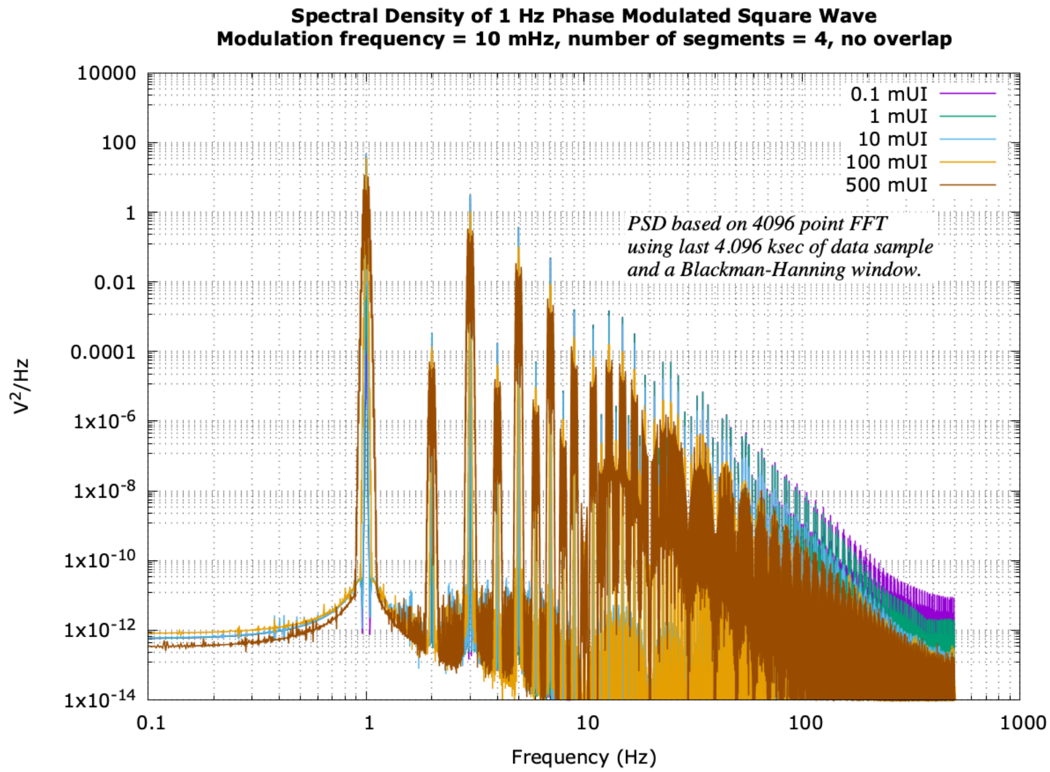


Figure 21

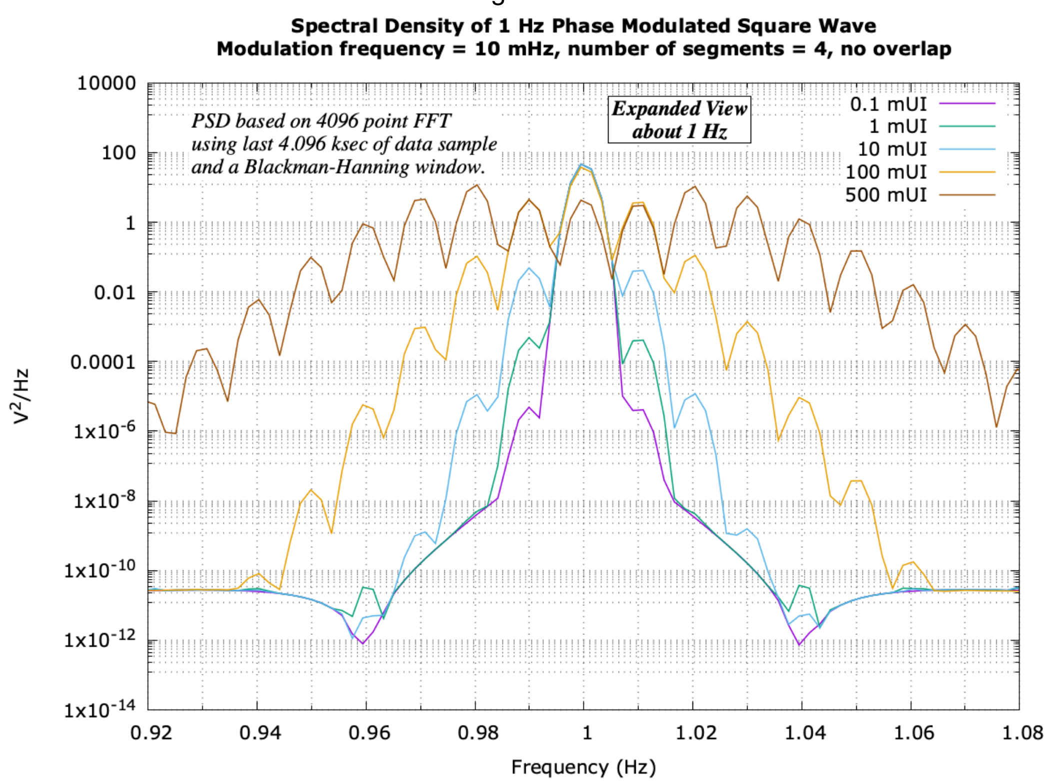
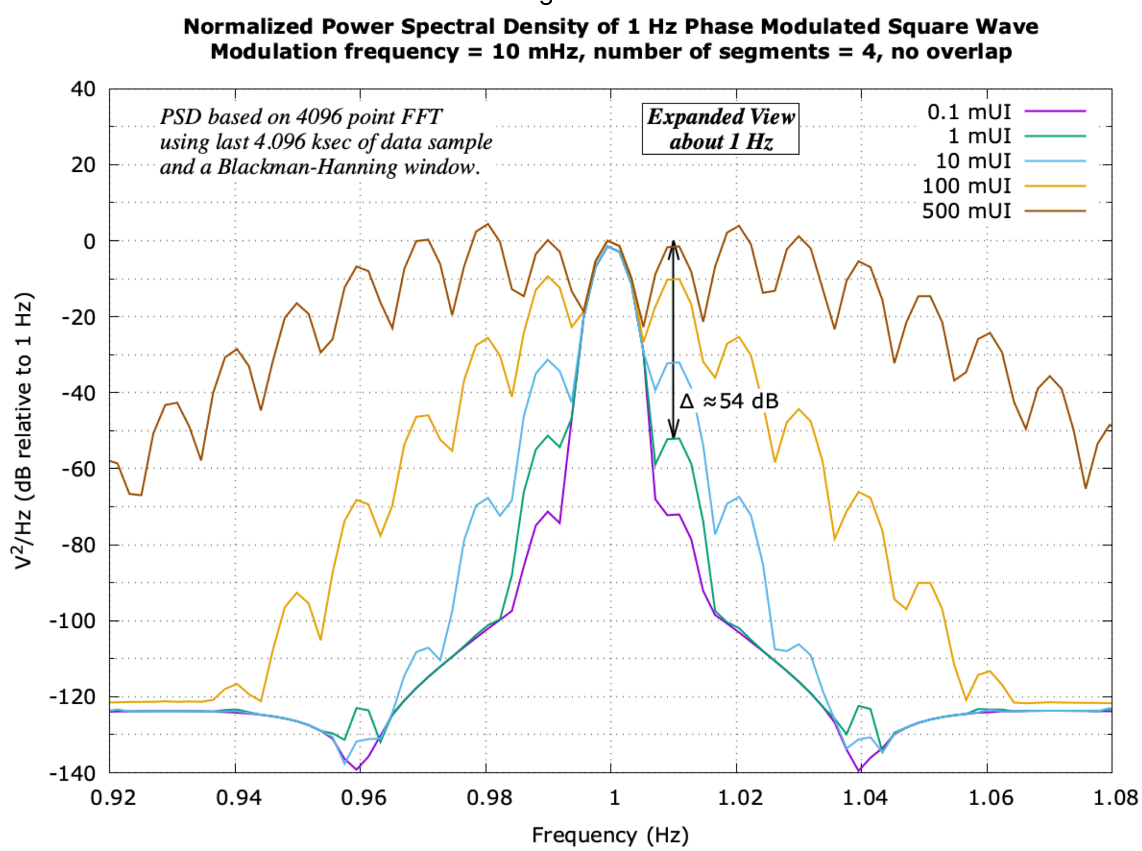


Figure 22 normalizes the power spectral densities in Figure 21 to their respective amplitudes at 1 Hz, Highlighted in Figure 22 is the approximate 54 dB difference between the power spectral density amplitude at 1 Hz at a modulation amplitude of 1 mUI and the peak of its only sideband at 10 mHz. from 1 Hz. The modulation amplitude of 1 mUI corresponds to a phase modulation of 60 dBc, and hence $\mathcal{L}(f)$ = -60 dBc/Hz. Using the approximate relationship of Equation [9], the spectral density $S_\phi(f)$ should be 6 dB greater than -60 dBc/Hz or -54 dBc/Hz. This is consistent with the results at 1 mUI shown in Figure 22. Similarly, when the phase modulation amplitude is reduced from 1 mUI to 0.10 mUI, the difference between the peak spectral density and sideband at 10 mHz is about -74 dB. However, when the phase modulation amplitude is increased to 10 mUI, the difference between its 1 Hz amplitude and its first10 mHz sideband amplitude is about -32 dB and suggests the value of $\mathcal{L}(f)$ is -38 dBc/Hz which is 2 dB different from the correct value of $\mathcal{L}(f)$ =-40 dBc/Hz. Using the difference in dB between the first 10 mHz sideband amplitudes and the peak amplitudes for modulation amplitudes of 100 mUI and 500 mUI to estimate $\mathcal{L}(f)$ results in even greater errors from the expected values of -20 dBc/Hz and -6 dBc/Hz respectively.

To avoid these errors, reference [4] suggests that the use of Equation [9] to estimate $\mathcal{L}(f)$ should be limited to values of $\mathcal{L}(f)$ less than approximately -20 dBc/Hz to preserve the small-signal approximation.

Figure 22



3.5.2 *jitterhist* Methodology to Compute Phase Noise

To avoid the conversion errors from $S_\phi(f)$ to $L(f)$ at larger amplitudes of phase modulation, *jitterhist* derives the phase noise from the power spectral density of the time interval error. As shown in Equation [10], for a timing signal with phase $\varphi(t)$ whose long term average frequency is $f_o = \frac{1}{T_o}$, its time interval error TIE is a sampled version of the phase error φ_e.

$$\varphi(t) = \omega_o t + \varphi_e(t)$$

$$where:$$

$$\omega_o = \text{ideal long term average frequency in rad/sec with period } T_o \text{ in sec}$$

$$\varphi_e(t) = \text{phase error in rad/sec}$$

$$\begin{aligned}
\varphi(iT_o) &= \omega_o iT_o + \varphi_e(iT_o) \\
\varphi(iT_o) &= 2\pi i + \varphi_e(iT_o) \\
TIE_i &= \varphi(iT_o) - 2\pi i \\
&= \varphi_e(iT_o)
\end{aligned} \qquad [10]$$

The single sided phase noise $L(f)$ in dBc/Hz is determined from the single sided power spectral density (PSD) of the time interval error in UI^2/Hz using Equation [11].

$$\begin{aligned}
\mathcal{L}(f) &= 10Log10\left[\left(\frac{1}{2}\right)S_\phi(f)\right] \\
&= 10Log10\left(\frac{1}{2}\right) + 10Log10[S_\phi(f)]
\end{aligned}$$

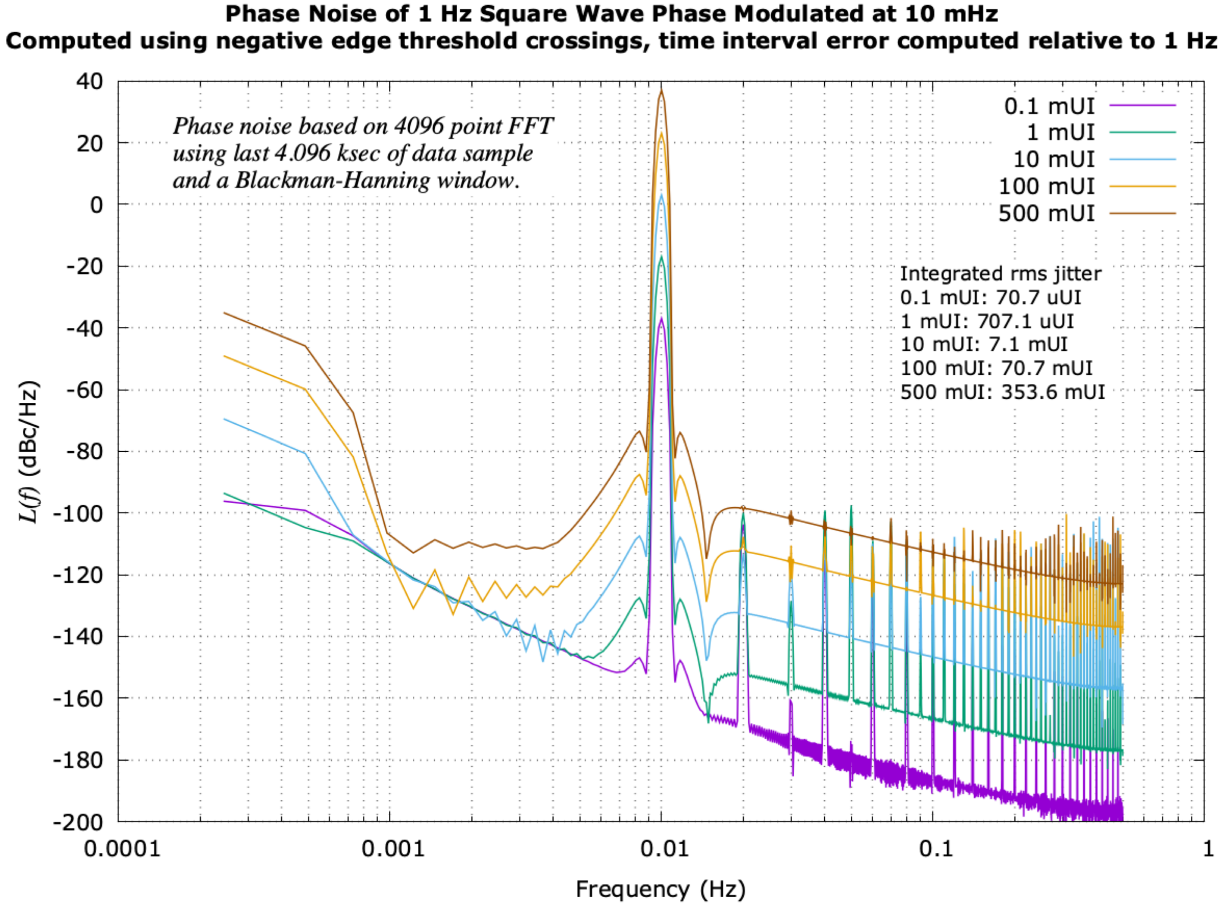
$$\text{For } S_\phi(f) \text{ expressed in } UI^2/Hz:$$

$$\begin{aligned}
\mathcal{L}(f) &= 10\log 10\,(S_\phi(f)) + 10\log 10[(2\pi)^2] + 10Log10\left(\frac{1}{2}\right) \\
&= 10\log 10\,(S_\phi(f)) + 12.95329741dB
\end{aligned} \qquad [11]$$

jitterhist initially computes the one-sided PSD from the time interval error in unit intervals and then performs the conversion to phase noise. The PSD algorithm is based on Welch's algorithm [5] and uses the last N samples of the M time interval samples with a single segment where $N = 2^{floor(log_2 M)}$. For example, if there are 5000 time interval error samples, the PSD is computed using the last 4096 time interval error samples. The phase noise will consist of 2048 phase noise values with its value at 0 Hz undefined (NaN).

Using the prior example of a set of 1 Hz phase modulated square waves whose modulation amplitudes are between 0.10 UI and 500 mUI, the phase noise computed using *jitterhist* for each square wave are shown in Figure 23. The integrated rms values for each phase modulation amplitude are consistent with their respective temporal rms values as expected from Parseval's Theorem.

Figure 23

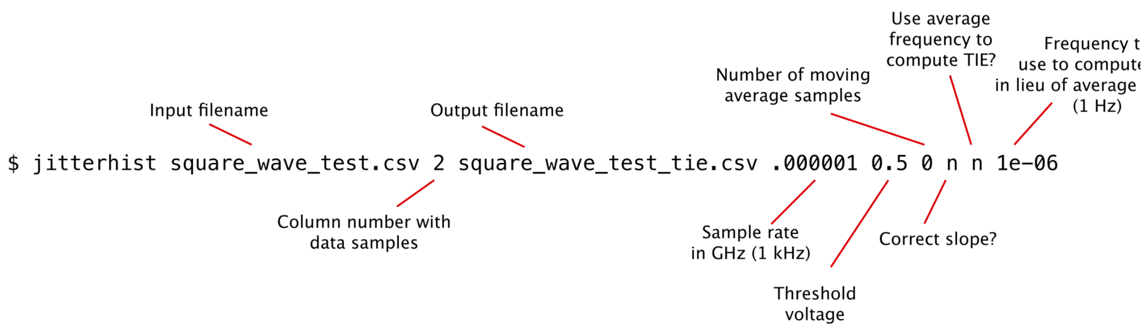


Phase Noise of 1 Hz Square Wave Phase Modulated at 10 mHz
Computed using negative edge threshold crossings, time interval error computed relative to 1 Hz
Phase noise based on 4096 point FFT
using last 4.096 ksec of data sample
and a Blackman-Hanning window.
0.1 mUI
1 mUI
10 mUI
100 mUI
500 mUI
Integrated rms jitter
0.1 mUI: 70.7 uUI
1 mUI: 707.1 uUI
10 mUI: 7.1 mUI
100 mUI: 70.7 mUI
500 mUI: 353.6 mUI
L(f) (dBc/Hz)
Frequency (Hz)
40
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
-200
0.0001
0.001
0.01
0.1
1

4. Examples of *jitterhist* Operation

4.1 Phase Modulated Square Wave at 1 Hz: Without TIE slope correction

As a simple example of its use, *jitterhist* is used to compute the time interval error and phase noise of a 1 Hz square wave phase modulated at 10 mHz with an amplitude of 100 mUI. The square wave is illustrated in Figure 24. The input file "square_wave_test.csv" contains 4100 periods of the square wave, and there are 1000 samples per period (sample rate of 1 kHz). Its first 11 lines are shown in Table 5. Example command line [1] is the command line used to perform the TIE analysis without slope correction. Since the input waveform does not contain any measurement noise or any added random noise, the number of moving average samples used to estimate the waveform threshold crossings about 500 mV is set to zero. In this example, the computed average frequency is not used in the analysis. Instead, the frequency of 1 Hz (1e-06 MHz) is included in the command line to compute the time interval error.



Example command line [1]

Figure 24

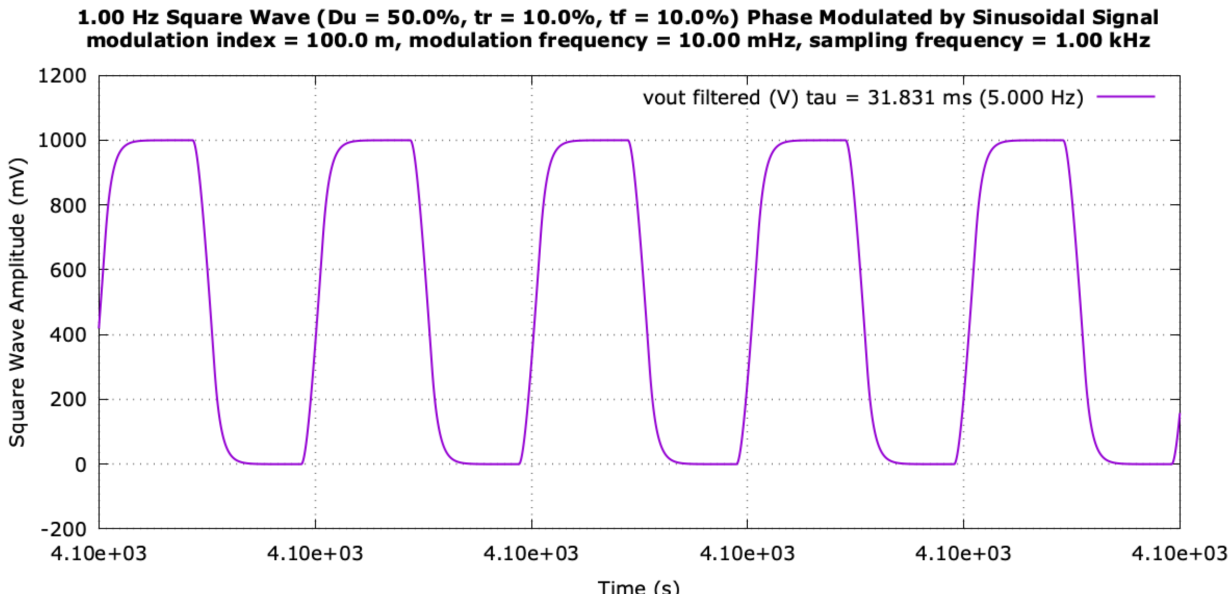


Table 5
Partial contents of file "square_wave_test.csv"

```
Time (s),vout filtered (V) tau = 31.831 ms (5.000 Hz)
0.000000000000e+00,1.649218549550e-01
1.000000000000e-03,1.720273076360e-01
2.000000000000e-03,1.792222816424e-01
3.000000000000e-03,1.865040082970e-01
4.000000000000e-03,1.938698045508e-01
5.000000000000e-03,2.013170703351e-01
6.000000000000e-03,2.088432859950e-01
7.000000000000e-03,2.164460098024e-01
8.000000000000e-03,2.241228755459e-01
9.000000000000e-03,2.318715901953e-01
1.000000000000e-02,2.396899316383e-01
```

As *jitterhist* is executing, a log is printed to the terminal. Figure 25 contains the output log in response to Example command line [1]. The plot of time interval errors as a function of time and the phase noise of the negative and positive edges of the timing signal are shown in Figure 26 and Figure 27 respectively.

Figure 25
jitterhist Output log: No slope correction

```
jitterhist v1.80 5/11/2025

Input file: square_wave_test.csv
Column number: 2
Output file: square_wave_test_tie.csv
Sample frequency is 1.0000 kHz.
Threshold value is 500.000 mV.
Number of moving average samples = 0.
Entered average frequency is 1.0000 Hz.
First threshold crossing at time 4.02214824e-02 sec, interpolated_value is 5.00000000e-01.
Read a total of 4099 periods in file "square_wave_test.csv"
using a threshold value of 500.0 mV.
Average time period = 9.999985e-01 sec, frequency = 1.00000153 Hz.
Minimum period = 9.937206e-01 sec (1.00631905 Hz at 1.949022 ks).
Maximum period = 1.006279e+00 sec (993.75981028 mHz at 2.298028 ks, delta = 1.2559%).
Average on time = 5.0100e-01 sec, Duty cycle = 50.10 %.
Minimum on time = 4.9785e-01 sec (Duty cycle = 49.79 % at 1.349022e+03 sec).
Maximum on time = 5.0415e-01 sec (Duty cycle = 50.41 % at 3.999034e+03 sec, delta = 0.63%).
The calculated average frequency is 1.00000153 Hz.
Using an average frequency of 1.00000000 Hz to compute jitter.
Negative edge location variation = 199.902 ms (199.902 mUIpp +/- 1.00 mUI).
Positive edge location variation = 200.000 ms (200.000 mUIpp +/- 1.00 mUI).
Phase noise analyses complete and used 1 segment of size 4096 and each contains 2048 output data
points.
Completed writing 2048 data points to phase noise output file
"square_wave_test_tie_seg_sublength_1_num_segments_1_ovrlap_percent_0_window_1_pnoise.csv".
Elapsed time: 20.893855 seconds
```

Figure 26

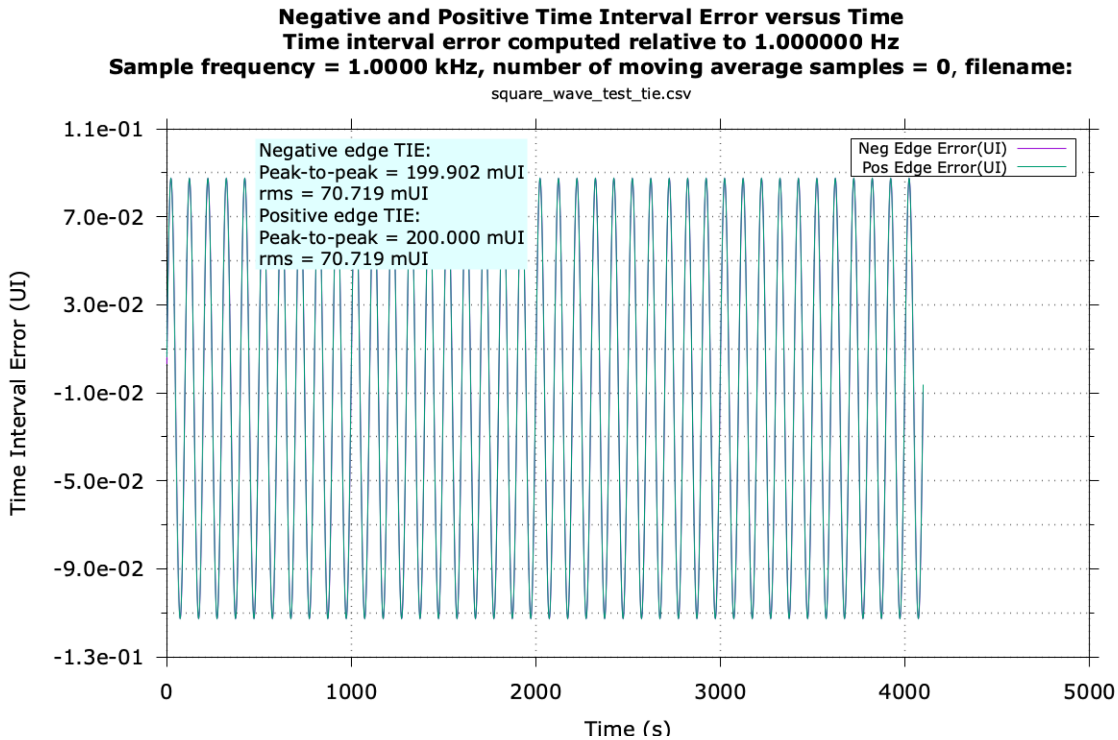
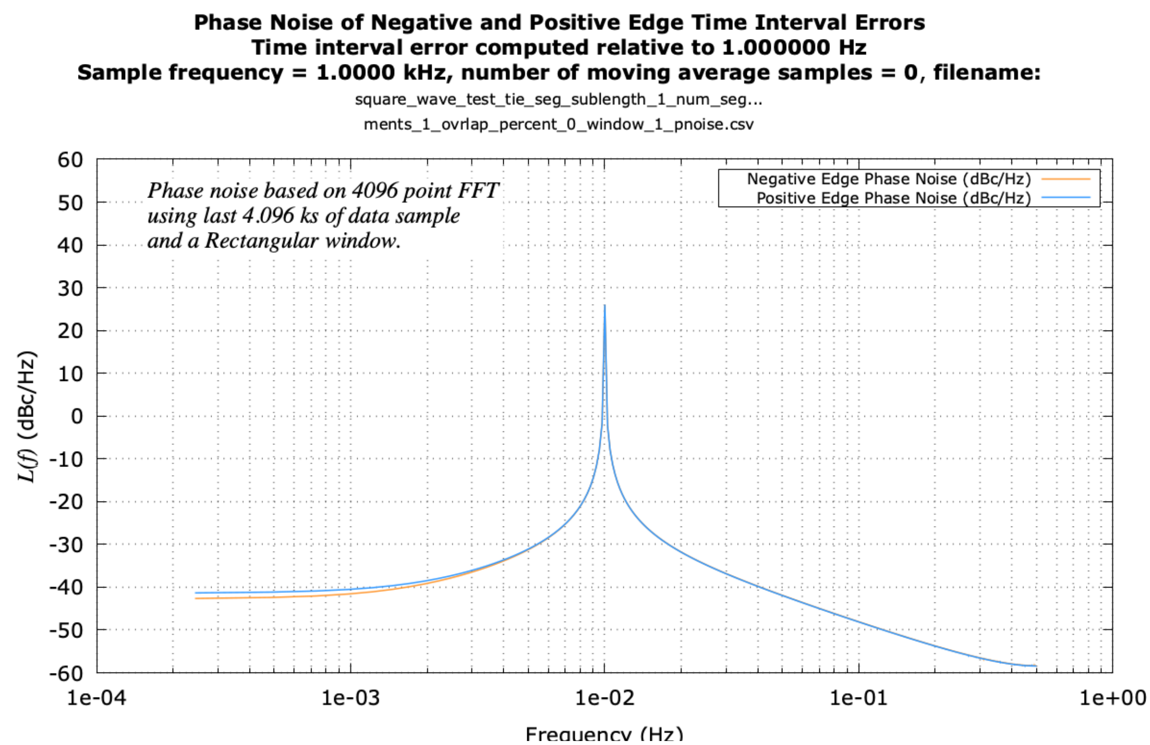
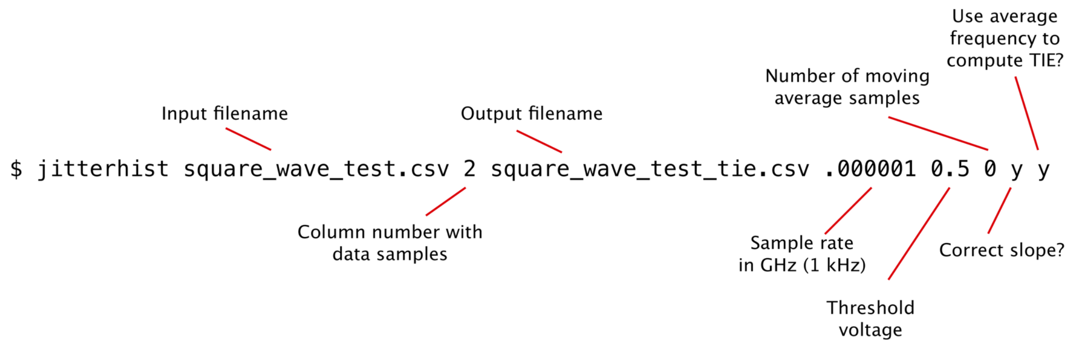


Figure 27



4.2 Phase Modulated Square Wave at 1 Hz: Includes TIE slope correction

Using the prior 1 Hz square wave phase modulated at 10 mHz with an amplitude of 100 mUI, ,*jitterhist* is used to compute the time interval error and phase noise of the signal with slope correction. In this example, Example command line [2] is used to invoke *jitterhist*. Note that the number of command line inputs is reduced by one when compared to Example command line [1] as it instructs the analysis to use the average frequency for the initial time interval error analysis. In Example command line [1], the frequency of 1 Hz was entered and used for the analysis. In addition, Example command line [2] instructs the executable to correct the slope of the time interval error.



Example command line [2]

The program log is provided in Figure 28. Following TIE computation using the average frequency, the slopes of the negative and positive edge TIE are computed and optimized. The updated TIE results are computed and included in the log. The corrected slope negative and positive edge TIE are plotted versus time in Figure 29, and the phase noise of each is provided in Figure 30.

Figure 28
jitterhist Output log: Includes slope correction

```
jitterhist v1.80 5/11/2025

Input file: square_wave_test.csv
Column number: 2
Output file: square_wave_test_tie.csv
Sample frequency is 1.0000 kHz.
Threshold value is 500.000 mV.
Number of moving average samples = 0.
First threshold crossing at time 4.02214824e-02 sec, interpolated_value is 5.00000000e-01.
Read a total of 4099 periods in file "square_wave_test.csv"
using a threshold value of 500.0 mV.
Average time period = 9.999985e-01 sec, frequency = 1.00000153 Hz.
Minimum period = 9.937206e-01 sec (1.00631905 Hz at 1.949022 ks).
Maximum period = 1.006279e+00 sec (993.75981028 mHz at 2.298028 ks, delta = 1.2559%).
Average on time = 5.0100e-01 sec, Duty cycle = 50.10 %.
Minimum on time = 4.9785e-01 sec (Duty cycle = 49.79 % at 1.349022e+03 sec).
Maximum on time = 5.0415e-01 sec (Duty cycle = 50.41 % at 3.999034e+03 sec, delta = 0.63%).
The calculated average frequency is 1.00000153 Hz.
Using an average frequency of 1.00000153 Hz to compute jitter.
Negative edge location variation = 205.929 ms (205.930 mUIpp +/- 1.00 mUI).
Positive edge location variation = 206.027 ms (206.027 mUIpp +/- 1.00 mUI).
Removing effect of residual slope in negative edge TIE:
slope_neg_edge = 1.177032e-06, intercept_neg_edge = -1.182157e-02
New negative edge based estimated frequency is 1.00000035 Hz.
Removing effect of residual slope in positive edge TIE:
slope_pos_edge = 1.188612e-06, intercept_pos_edge = -1.184531e-02
New positive edge based estimated frequency is 1.00000034 Hz.
Corrected positive and negative frequencies differ by -0.0 ppm (considered acceptable).
Slope corrected filename is "square_wave_test_tie_corrected.csv".
Corrected negative edge location variation = 201.279 ms (201.279 mUIpp +/- 1.00 mUI).
Corrected positive edge location variation = 201.332 ms (201.332 mUIpp +/- 1.00 mUI).
Phase noise analyses complete and used 1 segment of size 4096 and each contains 2048 output data
points.
Completed writing 2048 data points to phase noise output file
"square_wave_test_tie_corrected_seg_sublength_1_num_segments_1_ovrlap_percent_0_window_1_pnoise.
csv".
Elapsed time: 20.607368 seconds
```

Figure 29

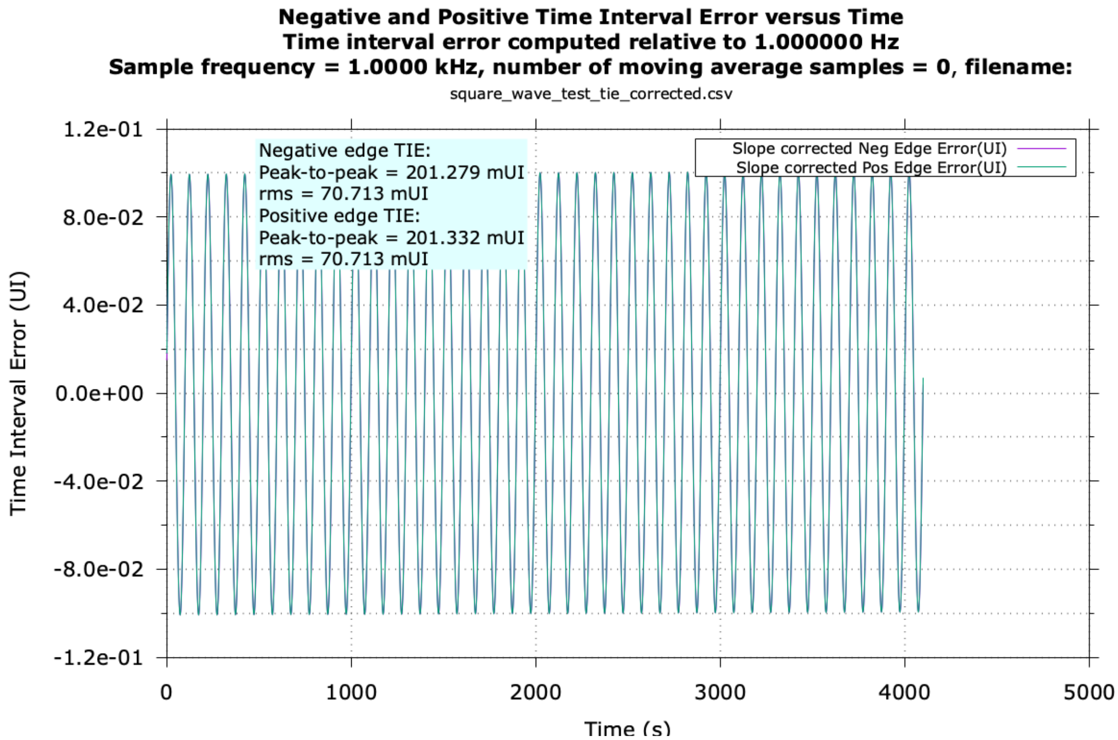
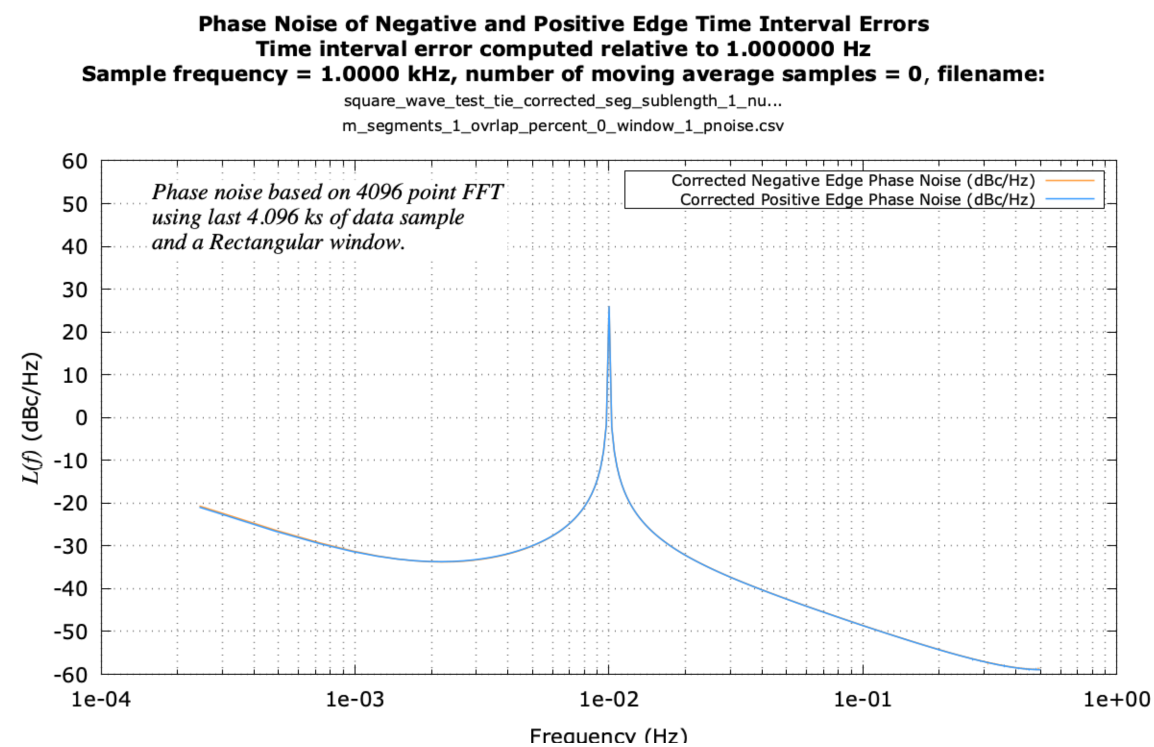


Figure 30



5. Program Installation

The current version of *jitterhist* is v1.80 and its source and documentation are available in the compressed tar file “jitterhist_v1p80_051125.tar.gz”. The most recent and stable version may be downloaded from reference [6]. To install the program and create the executable “*jitterhist*”, enter Example command line [3] in the directory in which you wish to locate the program.

$ tar -xvzf jitterhist_v1p80_051125.tar.gz   Example command line [3]

This will create a directory jitterhist_v1p80 and extract its directory structure:

Documentation/	include/	src/
README.txt	plotting_routines/	
example/	jitterhist@	

Navigate to the “src” subdirectory, and issue the following two UNIX commands:

$ make   Example command line [4]
$ make clean   Example command line [5]

Issuing these two commands will create the executable “*jitterhist*” and delete object files no longer needed. In addition, Example command line [4] will attempt to create a symbolic link to *jitterhist* in your $HOME/bin directory if this directory exists. Assuming your $HOME/bin directory is contained in your executable search path (UNIX PATH variable), this will allow you to execute *jitterhist* from any directory using a command line syntax similar to Example command line [1] in Section 4.1.

6. Summary

A C-based program, *jitterhist*, computes the time interval error and phase noise of a uniformly sampled periodic waveform contained in a comma-separated variable file. The program writes the time interval error and phase noise to comma-separated variable output files. If gnuplot is installed and in the executable search path, the program will also plot and save the time interval error and phase noise to Portable Network Graphic files (png) files.

jitterhist includes features to provide accurate time interval error results for periodic waveforms with significant amounts of high-frequency measurement/modulation noise and for an offset frequency between the estimated long term average frequency and computed frequency. It provides accurate results for waveforms having multiple unit intervals of low frequency modulation if the time span of the waveform samples includes more than one the period of the modulation.

Examples of program operation are included in Section 4, and instructions for downloading and installing the program are contained in Section 5.

Shawn Logan
August 12, 2024

Version History

v1.0 Initial document release 8/7/2024
v1.1 Release 8/8/2024

1. Syntax changes in Sections 1, 2, and 4 to improve readability
2. Corrected initial release date to 8/7/2024 in Version History
3. Corrected date below Summary Section 6 to 8/8/2024

v1.2 Release 5/15/2025

1. Updated Section 3.2 to reflect version 1.80 of *jitterhist* that accepts a single text file with all *jitterhist* input parameters as well as its former command line inputs.
2. Updated log file in Figure 14 to reflect log file of *jitterhist* v1.80.
3. Updated log files and time interval and phase noise plots in Section 4 to reflect outputs using *jitterhist* v1.80.

References

[1] D. B. Leeson, "Oscillator Phase Noise: A 50-Year Review," in IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 63, no. 8, pp. 1208-1225, Aug. 2016

[2] "IEEE Standard for Jitter and Phase Noise," in IEEE Std 2414-2020 , vol., no., pp.1-42, 26 Feb. 2021, doi: 10.1109/IEEESTD.2021.9364950.

[3] "IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology--Random Instabilities," in IEEE Std 1139-2008 (Revision of IEEE Std 1139-1999) , vol., no., pp.1-50, 27 Feb. 2009, doi: 10.1109/IEEESTD.2009.6581834.

[4] T. E. Parker, "Characteristics and Sources of Phase Noise in Stable Oscillators," 41st Annual Symposium on Frequency Control, 1987, pp. 99-110

[5] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," in IEEE Transactions on Audio and Electroacoustics, vol. 15, no. 2, pp. 70-73, June 1967.

[6] https://1drv.ms/u/s!AnM-GsAEZPoSsistH2ga2HiOVW6h?e=XOh71c