

# Learn Frequency Stability Analysis Using Stable32

W.J. Riley\*

Hamilton Technical Services

Beaufort, SC 29907 USA

[bill@wriley.com](mailto:bill@wriley.com)

March 4, 2020

## ABSTRACT

This document is a tutorial on frequency stability analysis using as its basis the Stable32 program and its documentation that is freely available from the International Electrical and Electronic Engineers (IEEE) Ultrasonics, Ferroelectrics, and Frequency Control (UFFC) Society. It guides the reader through various techniques for quantifying the stability of a frequency source in the time and frequency domains, providing information about both the theoretical background and practical methods for such an analysis.

## 1 INTRODUCTION

We start with an introduction to the Stable32 program and some frequency stability terminology.

### 1.1 Introduction

See: §1, [6], [9], [10], [14], [16], [22], [26], [27], [36], [37], [48], [52]<sup>1,2</sup>

This tutorial is intended to extend the knowledge of a casual Stable32 user toward a more complete understanding of the techniques

---

\* The author of this tutorial is the developer of the Stable32 program for frequency stability analysis.

<sup>1</sup> Annotations marked with § refer to the *Handbook of Frequency Stability Analysis*, and those inside square brackets [] refer to [references](#) at the end of this document. Open the *Handbook* from the Stable32 Help menu and keep it open as you read this tutorial.

<sup>2</sup> Suggested references are shown inside boxes below some section headings.

of frequency stability analysis. It presumes that the reader has already obtained and installed a copy of the Stable32 program and its documentation<sup>3</sup>, and that he/she has, at a minimum, followed the initial startup steps that demonstrate the basics of its operation in its Welcome screen and included test data<sup>4</sup>.

The tutorial then leads a Stable32 user through a number of examples emphasizing analysis techniques rather than program operation. A key aspect of the tutorial is the program's ability to generate power law noise as test data to use for exploring the various analysis methods.

Before going any further, we should say that what we are talking about is really frequency instability, but, of course, you know that.

Since you are interested in learning more about frequency stability analysis, you have no doubt heard about and probably used the Allan variance, AVAR, (or its square root, the more commonly-used form, the Allan deviation, ADEV). But there's much more to frequency stability analysis than that, as this tutorial will show.

This document contains basic information about most aspects of a frequency stability analysis,

---

<sup>3</sup> Stable32 can be downloaded at:

<https://ieee-uffc.org/frequency-control/frequency-control-software/stable32/>.

<sup>4</sup> You can reactivate this dialog by setting the ShowWelcome flag in the [Preferences] section of the Stable32.ini file to 1.

with references to further details (e.g., the math). Exercises are included for many topics to stimulate further study. It is recommended that, after reading about a topic in this document, the reader consult the referenced section of the *Handbook*, the references cited herein, and then their referenced documents as you get deeper into these topics.

Most analysts who use the techniques of frequency stability analysis have frequency sources they wish to characterize and measuring systems for doing so. But all users of precision frequency sources need to understand those techniques, and hands-on experience with them, using actual or simulated data, is the best way to become familiar with them.<sup>5</sup>

Do the exercises – don’t just look at them. That way you’ll learn more and be able to try different things. Treat this document as a workbook.

## 1.2 Documentation

The essential documentation for this tutorial is included with the Stable32 program, most importantly its *User Manual*<sup>6</sup> [36], the built-in context-sensitive help, and the associated *Handbook of Frequency Stability Analysis*<sup>7</sup>. The *User Manual* and *Handbook* can be accessed via the program’s Help menu. In addition, there are a number of papers in the Stable32 program folder and on the [stable32.com](http://stable32.com) web site that address specific topics related to frequency stability analysis (see [References](#)). All those documents contain many references to further information.

## 1.3 The Stable32 Paradigm

Since we will be using the Stable32 software to support this tutorial, it seems wise to first describe the way that program is organized.

<sup>5</sup> We highly recommend attending one of the annual [NIST Time and Frequency Seminars](#). The tutorials at the annual [Frequency Control Symposium](#), [European Frequency and Time Forum](#), and the [ION PTTI Meeting](#) are also opportunities to learn more about frequency control and stability analysis.

<sup>6</sup> Including its current [addendum](#), if any.

<sup>7</sup> References to *Handbook* sections are shown as §.

The basic paradigm is a pair of phase and frequency arrays whose data can be entered from a disk file or generated internally. Stable32 is a batch processing program – it does not directly support clock measurements. The phase and frequency arrays need not both be filled. If they are, they generally, but not necessarily, hold equivalent data (as indicated by a blue infinity-like symbol). The contents of these arrays are shown in small plots at the bottom of the screen (see Figure 1), along with their basic properties. Conversions are supported between phase and frequency data.



Figure 1. Stable32 Status Bar

A useful feature during an analysis step that changes the contents of these arrays is to activate the Tabs function so that the previous data is saved and can be restored.

## 1.4 Frequency Stability Terminology

See: §3

The time and frequency field employs a standard terminology to describe the stability of a frequency source [6].

A frequency source will have a certain nominal output frequency,  $f_0$ , and an actual output frequency,  $f$ , as measured with adequate precision against a suitable absolute standard via a traceable process. The output frequency is generally denoted by its *fractional frequency deviation*,  $\Delta f/f_0 = (f - f_0) / f_0$ . For example, for  $f_0 = 10$  MHz, if the actual frequency is  $f = 10.000001$  MHz, the frequency deviation is  $\Delta f = 1$   $\mu$ Hz, and the fractional frequency offset is 1 ppm, denoted as  $1 \times 10^{-6}$ ,  $1\text{pp}10^6$ , or  $1\text{e-}6$ , which are dimensionless quantities (not Hz) and are generally written simply as  $\Delta f/f$  and denoted as a function of time by the symbol  $y(t)$ . The corresponding phase value as a function of time is denoted as  $x(t)$  and has units of seconds.

## 1.5 Phase and Frequency Data

See: §3

Frequency stability analysis generally applies to equally-spaced discrete phase or frequency measurements (a *time series*) taken at a particular *measurement interval* denoted by the lowercase Greek letter *tau* ( $\tau$ ). Other words used for this quantity are *sampling interval*, *measurement time*, *sampling time* or *averaging time*. The measurement and sampling terms are usually associated with the measurement process itself, while the averaging time applies to the analysis. The basic measurement interval is often denoted as  $\tau_0$  while the analysis averaging time is simply called  $\tau$ . As noted above, phase data have units of seconds, while frequency data are dimensionless<sup>8</sup> fractional frequency.

## 1.6 Data Averaging

See: §3, [5]

Phase and frequency data taken at one sampling interval can be averaged to a longer tau (the opposite cannot be done, of course). Phase data are “averaged” by simply omitting the intermediate points, while frequency data are averaged to a longer tau by ordinary algebraic averaging. There are some subtleties associated with those processes [5], but for the most part they are straightforward (and supported by the Stable32 program). Data averaging is done transparently during a stability run, and may be done to a data file as it is read or later as a way to speed an analysis or save storage space when a finer time resolution is not needed.

## 1.7 Some Statistical Terminology

See: §2

Frequency stability analysis involves both deterministic and stochastic (random) factors. Examples of the former are frequency offset and drift, while statistics and spectral densities are employed to describe the latter. The noise properties can be examined using measures like variances in the time domain and power spectra in the frequency (Fourier) domain.

---

<sup>8</sup> Units of Hz/Hz are sometimes associated with fractional frequency values, but that seems rather awkward.

In the time domain, we analyze phase or frequency time series data that is generally assumed to exist “forever” and whose properties remain constant over time (are stationary)<sup>9</sup>. The unlimited extent of the data means that we apply sampling rather than population statistics. If the behavior of the frequency source does change due to jumps, etc. then we must consider that there has been a regime change.

In the frequency domain, the power spectrum of the frequency or phase variations is usually modeled as an integer power law function.

Alternatively, one can analyze the phase or frequency variations by their autocorrelation function. For white noise, adjacent samples are uncorrelated. For more divergent flicker, random walk, and random run noise, their samples are correlated over increasing distances.

## 2 STABILITY ANALYSIS

Next we cover some of the basics of frequency stability analysis.

### 2.1 Some Analysis Basics

See: §2, [10]

When analyzing the variations of a time series, it is important to distinguish between a statistic (e.g., the [Allan variance](#)) and its estimator, the computational formula used for its calculation. For example, while AVAR is defined as the expected value of an expression most similar to its original 1<sup>st</sup> differences of frequency non-overlapped estimator, it is best estimated by the more efficient fully overlapping formula since that provides more degrees of freedom and higher confidence. Both of those AVAR estimators are unbiased. Other even higher confidence biased estimators are available (e.g., the [Total](#) and [Théo1](#) statistics), or when results are needed at larger averaging factors, but they may require bias corrections that depend on the noise type (Stable32 does those automatically).

---

<sup>9</sup> The average frequency of a divergent flicker or random walk FM process is not stationary.

The [Modified Allan variance](#) is a special purpose statistic involving phase averaging that is used mainly for its ability to distinguish between white and flicker PM noise. It finds wider use as the basis for the Time variance, TVAR, which is a popular estimator for clock error caused by noise.

The [Hadamard variance](#), HVAR, is another special purpose statistic that is based on the 2<sup>nd</sup> differences of frequency. Although providing lower confidence than AVAR, it has the advantage of convergence for more divergent noise types, and, generally more importantly, ignores linear frequency drift.

Some basic principles to keep in mind:

1. Frequency is the rate of change of phase.
2. One converts phase data to frequency data by taking 1<sup>st</sup> differences.
3. One converts frequency data to phase data by numerical integration.
4. The exponent of power law noise is increased by 2 by differencing which makes it less divergent and raises its relative high frequency content.
5. The exponent of power law noise is decreased by 2 by integration which makes it more divergent and lowers its relative high frequency content.
6. White noise has a flat spectral density versus Fourier frequency.
7. Noise spectra can apply to either phase or frequency data, whose power law exponents differ by 2.
8. Always follow R.W. Hamming's admonition that "the purpose of computing is insight, not numbers" [3].

## 2.2 Power Law Noise

See: §3.2

Power law noise having an integer exponent,  $\alpha$ , between -4 and +2 corresponding to noise types from White PM to Random Run FM have been found to characterize the behavior of all types of frequency sources<sup>10</sup>. Some of these noise types

correspond directly to physical mechanisms in those devices (e.g., W FM in passive atomic frequency standards), while others are simply reasonable models for their behavior.

In addition, integer power law noise follows the +2 increase in  $\alpha$  that corresponds to the first differencing during the conversion between frequency and phase data, and vice versa. So, for example, W FM noise is the same as RW PM noise (both  $\alpha=0$ ).

The power law noise type is sometimes referred to by its color (e.g., white for a flat spectrum, pink for a 1/f or flicker spectrum, red for an  $f^2$  spectrum, brown for an  $f^2$  random walk spectrum, and black for an  $f^3$  flicker walk spectrum) although this terminology is mainly used only for white noise in the frequency stability analysis field.

Examples of the most common types of power law noise are shown in Figure 2.

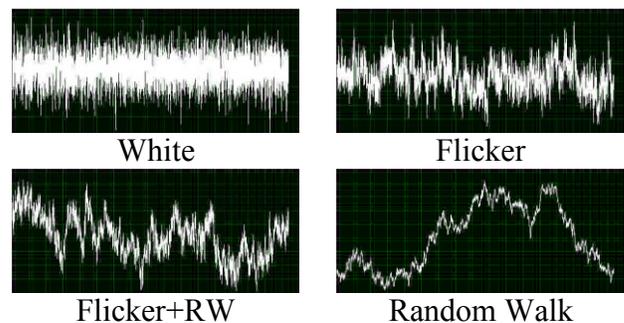


Figure 2. Common Types of Power Law Noise

The various noise types can apply to either phase or frequency data. Note that power law noise doesn't necessarily have to have an integer exponent – mixtures of noise types are possible.

Table 1. Summary of Stability Plot Noise Slopes				
Noise Type	$\alpha$	Stability Plot Noise Slope		
		ADEV	MDEV	TDEV
W PM	+2	-1	-3/2	-1/2
F PM	+1	-1	-1	0
W FM	0	-1/2	-1/2	+1/2
F FM	-1	0	0	+1
RW FM	-2	+1/2	+1/2	+3/2

<sup>10</sup> Plots of various power law noise types are shown in Table 1 of the *Handbook*.

## 2.3 Noise in Frequency Sources

The type of noise associated with each type of frequency source depends on the physical principles of its operation<sup>11</sup>. The source may be an active oscillator or a passive device like most atomic frequency standards where a frequency lock loop locks a crystal oscillator to an atomic reference. The dominant noise may vary with the spectral region (e.g., near or far from the carrier) or averaging time (short, medium, or long-term stability).

The simplest case is probably a classic rubidium gas cell or cesium beam tube passive atomic clock, or a passive H-maser, where white noise at its detector maps over to frequency variations in its frequency lock loop for averaging times longer than the loop time constant. Baseband flicker noise isn't much of an issue because the frequency lock servo operates with an audio frequency error signal. At shorter averaging times, the free-running crystal oscillator determines the stability. At much longer averaging times, environmental sensitivities and internal aging mechanisms may impose a so-called "flicker floor", beyond which the stability worsens. Behavior in that region probably isn't driven by any noise source in the usual sense of the word but rather by the operating environment.

Active oscillators like crystal oscillators and active hydrogen masers have an even more complex combination of noise sources. At low Fourier frequencies within the oscillator loop bandwidth, phase changes become frequency changes. Active device  $1/f$  flicker noise becomes  $f^3$  noise very close to the carrier, and device white noise becomes  $f^2$  noise further from the carrier. Outside the oscillator loop bandwidth, the device  $1/f$  and white noise is seen directly as additive noise, with the latter setting the far out noise floor. In the time domain, the more divergent noises tend to make the frequency record of a crystal oscillator rather un-white with flicker and random walk fluctuations. Likewise, the noise of an active H-maser tends varies from white/flicker PM at short averaging times to

more divergent random walk/flicker walk FM at longer averaging times (albeit very small).

The noise of time transfer systems is usually dominated by either white or flicker PM noise, while the noise floor of most clock measuring systems is set by their quantization or other white PM noise that improves with the averaging time.

## 2.4 Noise Generation

See: §8

The Stable32 Noise function can generate power law phase and frequency data to support the study of frequency stability analysis techniques. The resulting noise data has a selectable number of points, tau, power law noise type and level, and can include a frequency offset and/or drift, plus a sine component of selected level and period. The Noise dialog box is shown in Figure 3.

A different set of noise data is generated each time the function is invoked. This is usually convenient when performing a simulation, but it also means that your results will be slightly different than the examples herein.

You can generate and plot various types of integer power law noise using the Noise and Plot functions of Stable32. You'll notice that all white noise looks more-or-less the same, while there is a large variation in the appearance of the more divergent noise types<sup>12</sup>.

<sup>11</sup> See any reference to the [Leeson model](#) [4].

<sup>12</sup> The (little-used) Stable32 Audio function, found under the Utility menu, allows phase or frequency noise data to be converted to a .wav file and listened to. White PM frequency noise sounds like frying bacon. White FM frequency noise sounds like the noise from a radio between stations. Random walk FM frequency noise has a low rumbling sound. One can also examine the noise waveform with an oscilloscope or audio FFT analyzer.

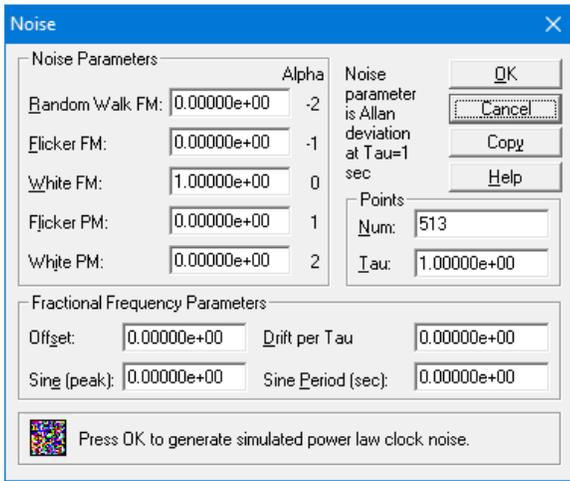


Figure 3. Noise Dialog Box

This function can be used to generate power law phase and frequency data having the desired properties that can be used as test data for a frequency stability analysis.

Figure 4 shows the results of ten sets of 513 points of simulated white FM noise having a nominal ADEV of 1.0 at their sampling tau.

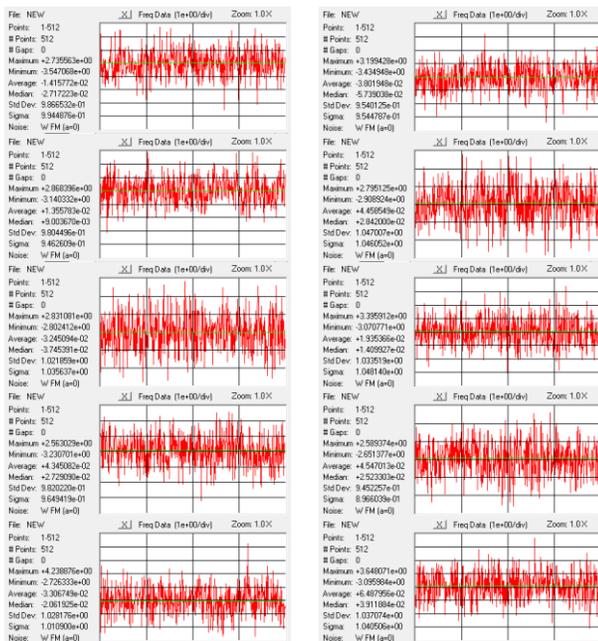


Figure 4. Ten Sets of Simulated W FM Noise

The ADEV and standard deviations values are nearly the same as expected for white FM noise. In all cases the ADEV values are well within the double-sided 95% confidence factor range (per the Sigma function not shown).

The Stable32 distribution package includes several sets of phase and frequency data, but these are quite small and do not represent a variety of

noise types. Thus, for the purposes of this tutorial, the reader should be prepared to generate appropriate data sets to illustrate various cases.

By the way, if you should ever need to generate samples of the even more divergent Flicker Walk FM ( $\alpha=-3$ ) and Random Run FM ( $\alpha=-4$ ) noise types, that is quite easy to do with the technique used in §3.4 and §8.3 of the *Handbook*. Generate some F FM noise phase data, save it, read it in again as FW FM frequency data, and convert it to FW FM phase data. Likewise, generate some RW FM noise phase data, save it, read it in as RR FM frequency data, and convert it to RR FM phase data.<sup>13</sup>

## 2.5 Noise Identification

See: §5.5

Many aspects of a frequency stability analysis require identification of the dominant noise type of the phase or frequency data under investigation, and several techniques have been devised to estimate the power law noise type automatically. Therefore the analyst generally does not have to worry about doing that him/herself. If you are interested in the methods used to ID power law noise, more information is available about that in Section 5.5 of the *Handbook*. One of the most effective uses the Lag 1 autocorrelation function [1].

## 2.6 Data Plotting

See: §10.12

Data plotting is a basic step for any frequency stability analysis. Visual examination of the phase and frequency data provides insight into its general character, and guides the rest of the analysis. A data plot will reveal any outliers, and, at various averaging factors, will show the noise characteristics, drift trend, and any anomalies. A formal, annotated data plot is not always necessary, but visual inspection should be the

<sup>13</sup> Note that the saved files should not have headers. Note also that the Stats function will not ID these very divergent noise types, but that the Autocorrelation function will.

first step in any frequency stability analysis<sup>14</sup>. This tutorial contains many examples of phase and frequency data plots.

## 2.7 Why the Allan Variance?

See: §5.2.1, [43]

The standard variance and its square root, the standard deviation is widely understood and used as a way to describe the scatter of a statistical variable. It was found, however, as you shall soon see for yourself, that it fails to adequately describe the more divergent forms of power law noise that commonly occur in frequency sources. For white noise (e.g., W FM) it's fine. But for flicker and other divergent noise it doesn't converge – the value grows with the number of samples analyzed. The Allan variance/deviation was devised in the early 1960's to solve that problem by using the first differences of the fractional frequency deviations as its basis. Let's see for ourselves:

Using Stable32, generate sets of 100, 1000, 10,000 and 100,000 samples of flicker FM noise, and calculate the standard and Allan deviations for each. It doesn't matter what noise level you use or its tau – let's use 1.0 for both. Generate the noise with the Noise function, analyze it with the Overlapping ADEV Sigma function, record and compare the results. Here's an example:

# Samples	Standard Deviation	Allan Deviation
100,000	2.80	1.20
10,000	2.71	1.20
1000	2.23	1.23
100	1.88	1.29

As you can see, the standard deviation grows with the number of samples while the ADEV remains nearly constant<sup>15</sup>. This behavior is shown graphically in Figure 5.

<sup>14</sup> Stable32 includes small phase and frequency data plots in its status bar, and offers the Stats function for more detailed data inspection.

<sup>15</sup> You can see this effect plotted in Figure 1 of the *Handbook* and herein as Figure 5.

Convergence of Standard & Allan Deviation for Flicker FM Noise

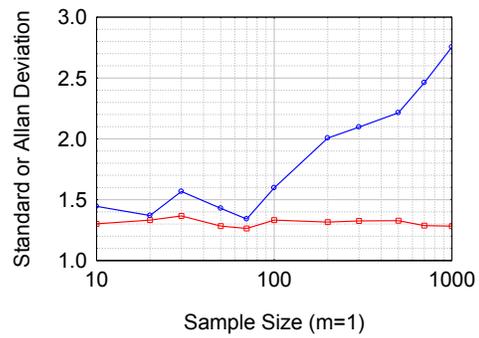


Figure 5. Convergence of Standard and Allan Deviation of Flicker FM Noise.

The standard deviation (blue) depends on the sample size for F FM noise, while the Allan deviation (red) does not and is therefore a better measure of frequency stability.

## 3 EXERCISES

Now it's time to try some of the techniques of frequency stability analysis using the Exercises listed below.

**Table 2.**  
**Frequency Stability Analysis Exercises**

#	Topic
<a href="#">1</a>	Generate and Analyze Power Law Noise
<a href="#">2</a>	Confidence Intervals
<a href="#">3</a>	Compare PSD Plots
<a href="#">4</a>	Check Spec Consistency
<a href="#">5</a>	Use MVAR to Distinguish Between White and Flicker PM Noise
<a href="#">6</a>	HDEV Rejects Frequency Drift
<a href="#">7</a>	HDEV Has Wider Error Bars Than ADEV
<a href="#">8</a>	Try the Dynamic AVAR Function
<a href="#">9</a>	Try the Autocorrelation Function
<a href="#">10</a>	Use an Autocorrelation Scatter Plot to Show Data Quantization
<a href="#">11</a>	Frequency Drift Model
<a href="#">12</a>	Find and Remove Outliers
<a href="#">13</a>	Frequency Jumps

### 3.1 Power Law Noise Characteristics

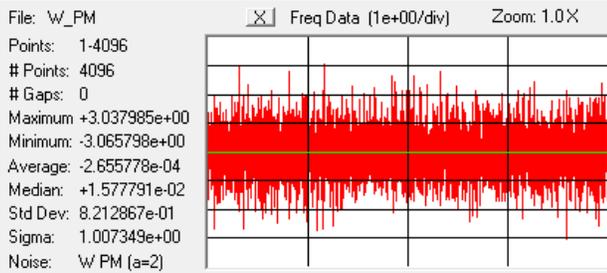
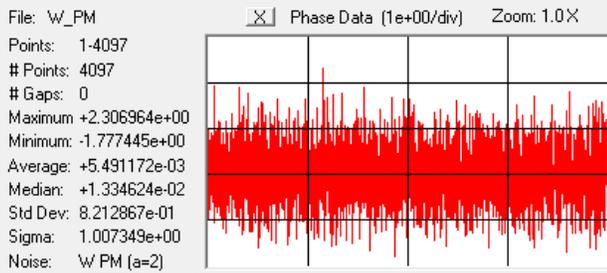
See: §3.2

A good first exercise is to use the Stable32 Noise function to generate and save five 4097-

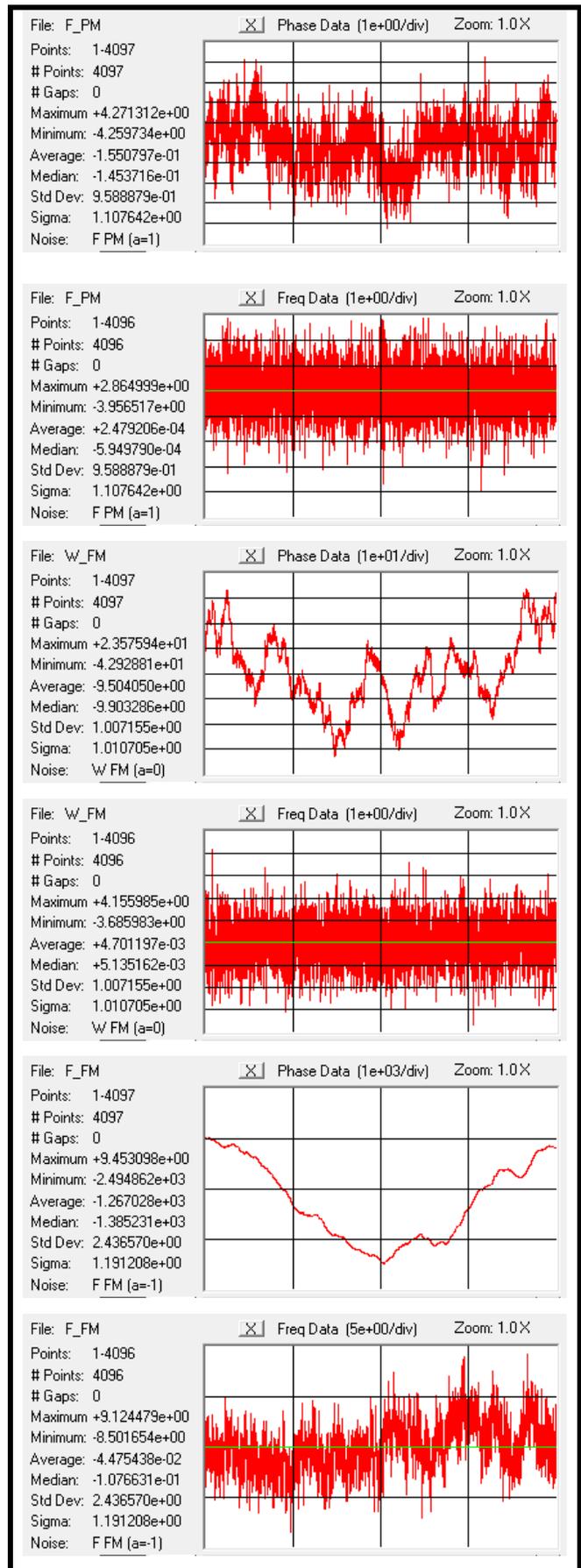
point integer power law noise phase and frequency data sets (10 files) and examine them with the Plot, Run and Power functions to observe their properties in both the time and frequency domains. The non-zero Sigma values can be set arbitrarily to 1.0 and the Tau to 1, with no frequency offset or drift, and no sine component. One can use filenames like RW\_FM, F\_FM, W\_FM, F\_PM, and W\_PM with .phd and .frd extensions to identify them.

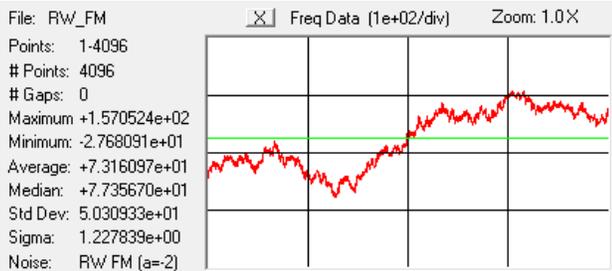
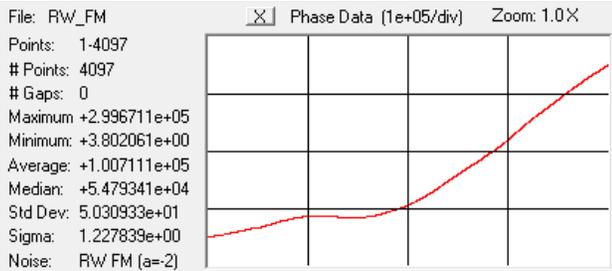
### Exercise #1: Generate and Analyze Power Law Noise

1. Launch the Noise function and generate a set of 4096 (# points=4097) of W PM noise. Use a value of 1.0 for the noise amplitude and 1 for the tau, and omit any drift or sinusoidal terms. Or, if you wish, you can set the amplitude to a more realistic value.
2. Examine the noise in the small plots on the status bar or with the Stats function. If you wish, launch the Plot function and produce and annotate larger plots. You may also want to print them.

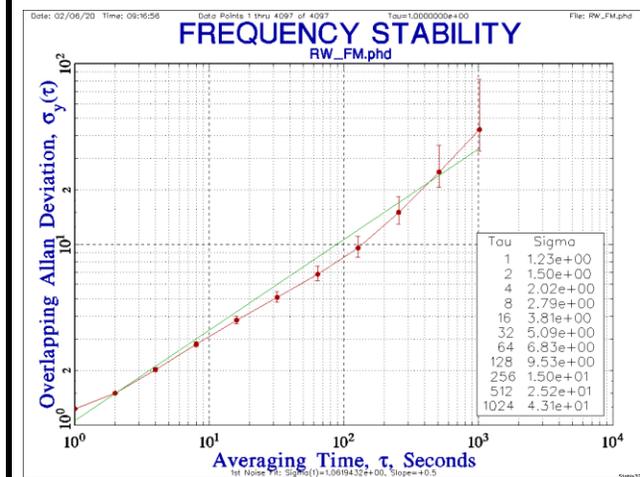
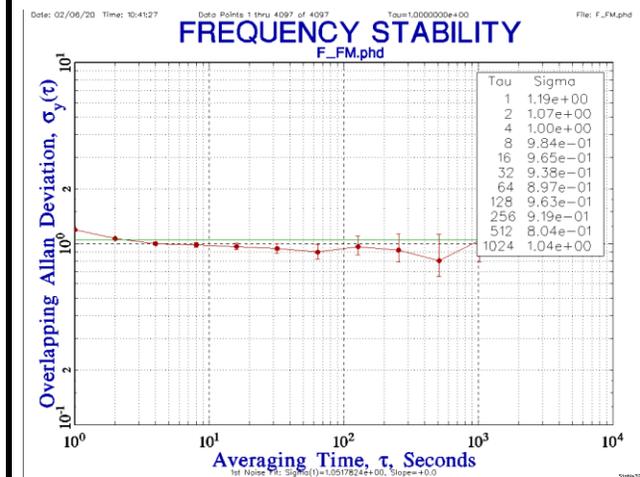
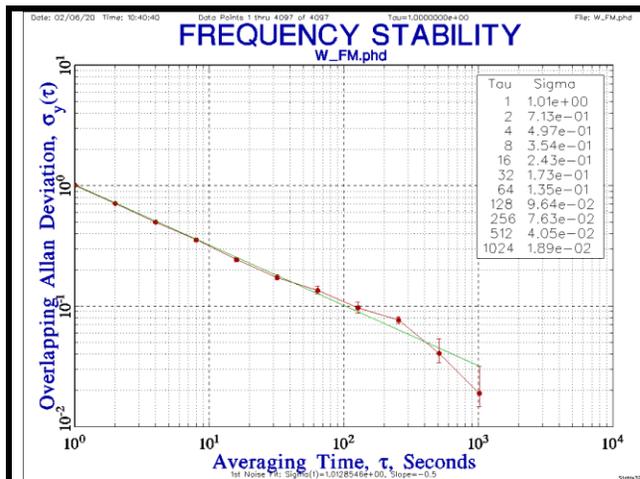
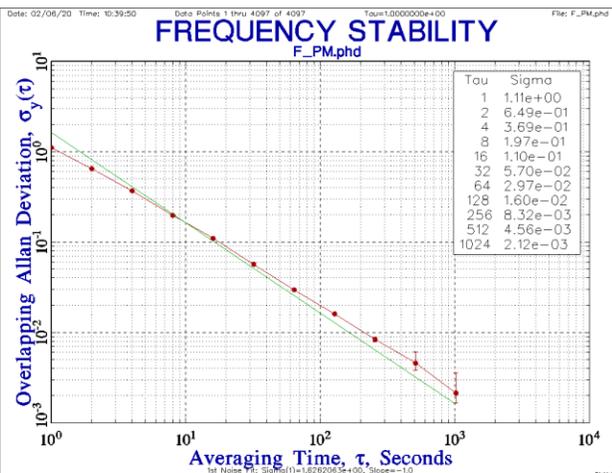
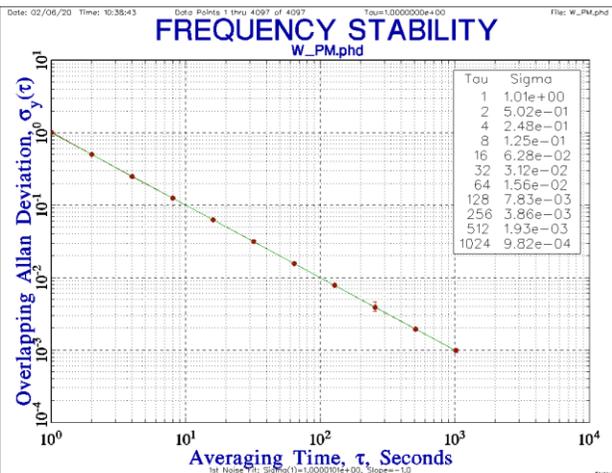


3. Save the resulting values as both phase (W\_PM.phd) and frequency (W\_PM.frd) data files at a convenient location on disk. You might also want to save these data sets in Tabs for easy recall.
4. Repeat the same process for the other four power law noise types (F PM, W FM, F FM & RW FM).

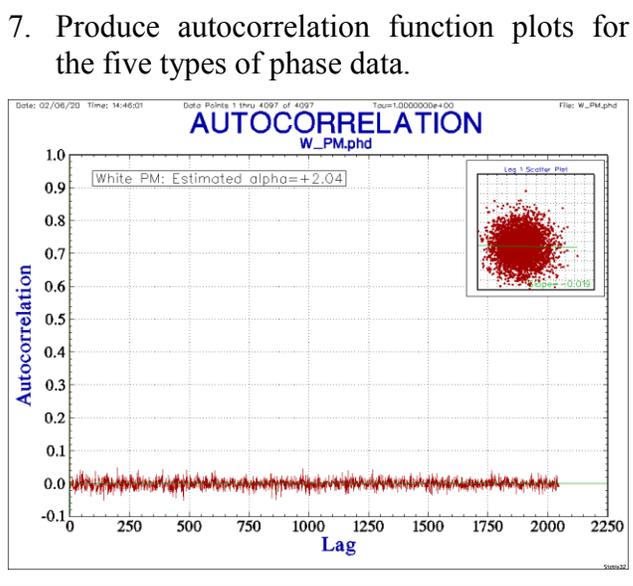
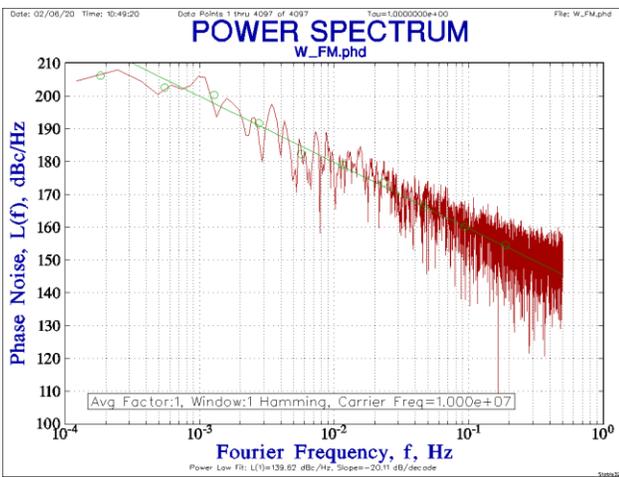
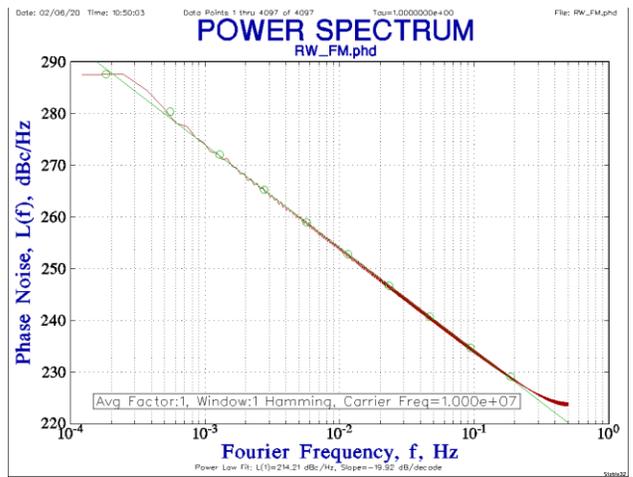
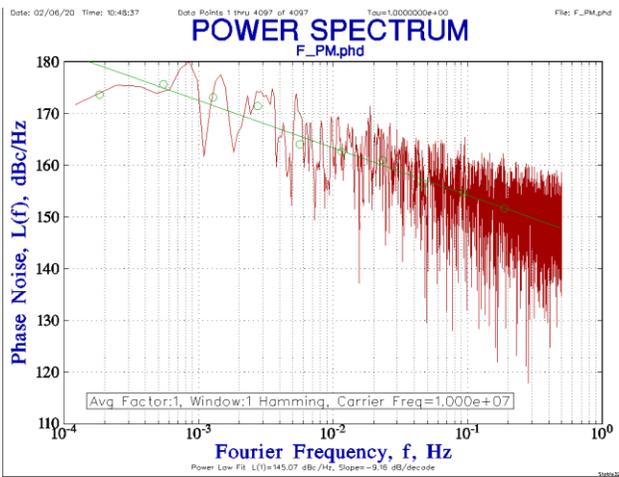
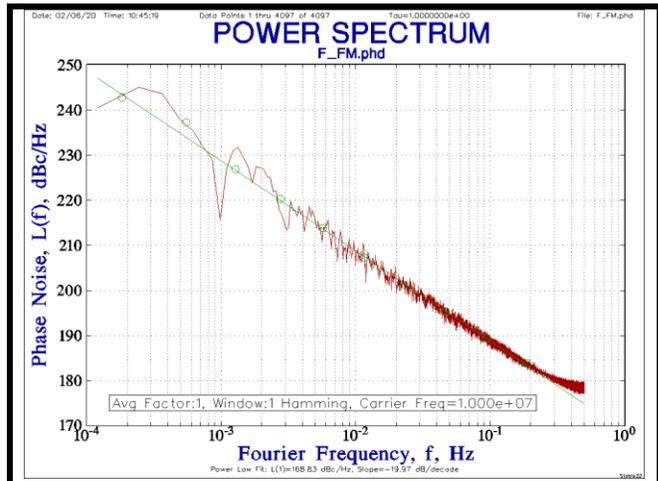
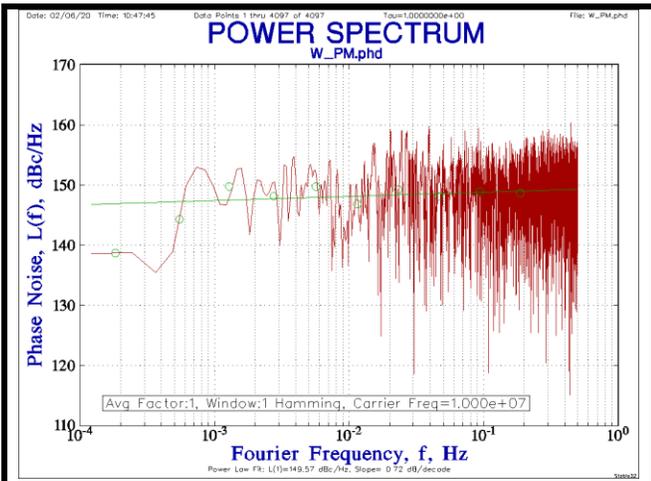




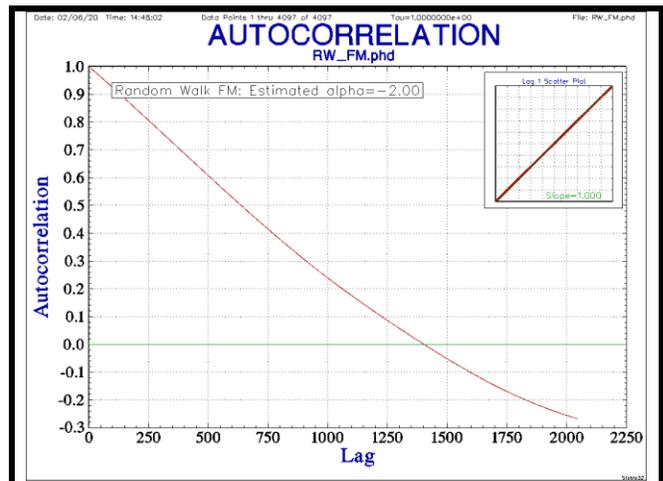
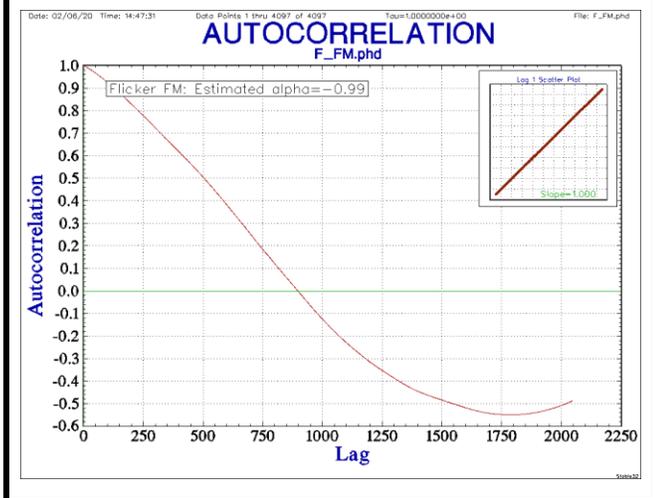
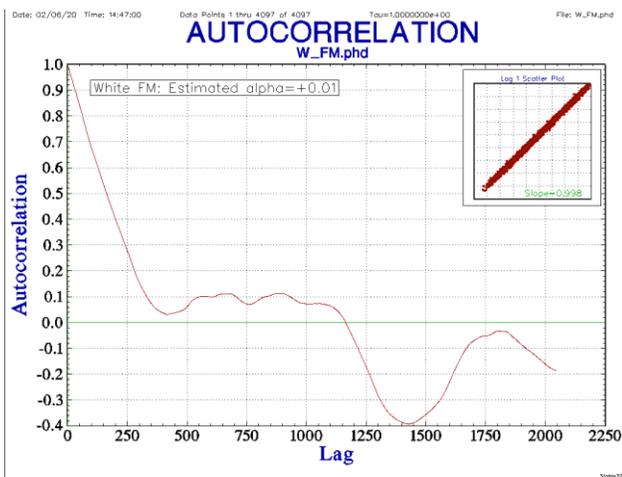
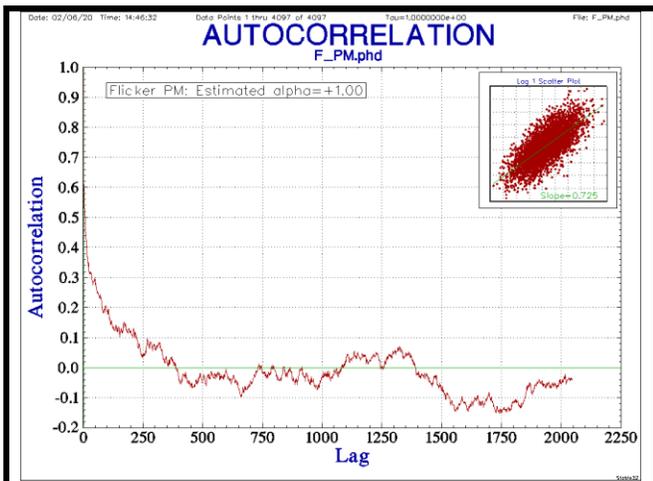
5. Produce ADEV plots for these five noise types. Include noise fits and observe their slopes.



6. Produce  $\mathcal{L}(f)$  PSD plots for these five noise types. Include power law fits and note their slopes.



7. Produce autocorrelation function plots for the five types of phase data.



- Notice how the data becomes increasingly correlated at longer lags as the data type becomes more divergent, and that the noise type is accurately identified by the Lag 1 ACF.

That's a lot of work, but it will provide considerable insight into the behavior of power law noise and the relationship between phase and frequency data, and time and frequency domains. A copy of the [Chart of Frequency Stability Analysis](#) included with Stable32 can be helpful<sup>16</sup>.

### 3.2 Confidence Intervals

See: §5.3, [7], [25]

The results of a stability analysis such as the Allan deviation should include an assessment of the confidence that its value is correct. This is usually done by showing error bars along with the nominal value, where the maximum and minimum limits correspond to a certain confidence factor<sup>17</sup>.

It is always better to show even approximate error bars than none at all. Simple error bars are typically set using a factor of  $\pm 1/\sqrt{N}$ , where n is the number of data points.

The setting of confidence limits and error bars for a particular stability variance is a bit complicated, but fortunately the details can be hidden from the analyst by appropriate software like

<sup>16</sup> See also Figures 3 and 31 of the *Handbook*.

<sup>17</sup> 68% double-sided confidence limits are generally used.

Stable32. The span of the error bars indicate the “variance of a variance” and therefore follow Chi-squared statistics that depend on the number of equivalent degrees of freedom (see §5.4) that applies to the analysis. That, in turn, depends on the variance type, the noise type, the number of data points used, and, of course, the chosen confidence factor. Of these attributes, the number of  $X^2$  degrees of freedom is the hardest to determine, but formulae have been developed for doing so, either analytically or empirically via simulation<sup>18</sup>.

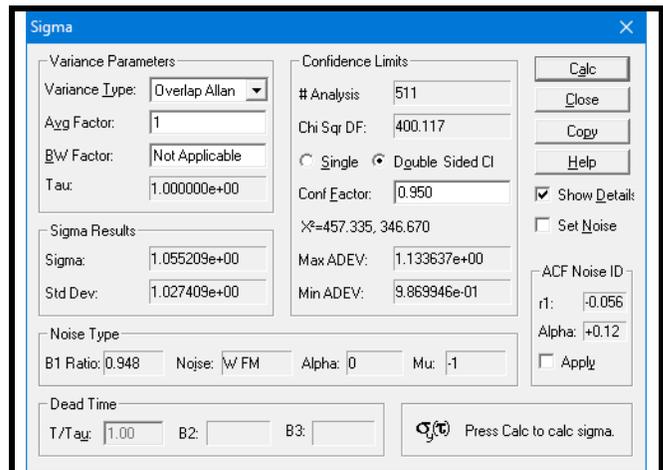
The most important observations about these confidence intervals and error bars are:

1. The progression from classic to overlapping ADEV, to the Total and Thêol statistics has significantly improved the confidence in stability determinations (see [Higher-Confidence Statistics](#) below and *Handbook* Figure 1).
2. The error bars grow with larger averaging factors as the number of analysis points becomes smaller.
3. The number of data points (and measurement run time) required is about 5 times shorter for Thêol than for the original ADEV.
4. The  $X^2$  error bars are asymmetric, larger in the upper direction.
5. Most stability analyses use double-sided 68% (1-sigma) confidence intervals.
6. Stability results are usually reported as their nominal values.
7. The Stable32 software hides most of this complexity.

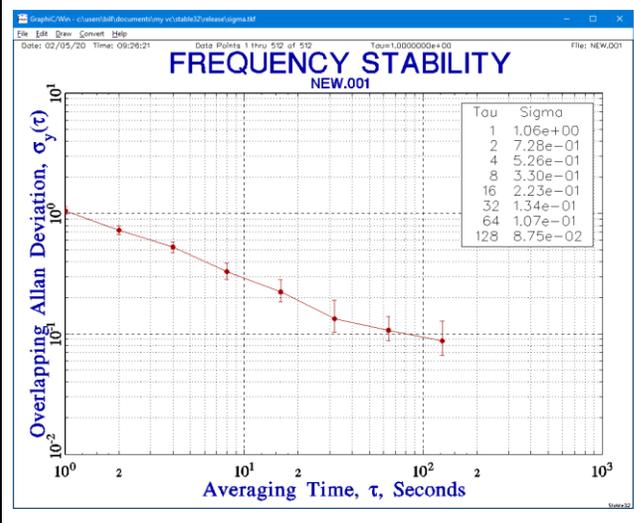
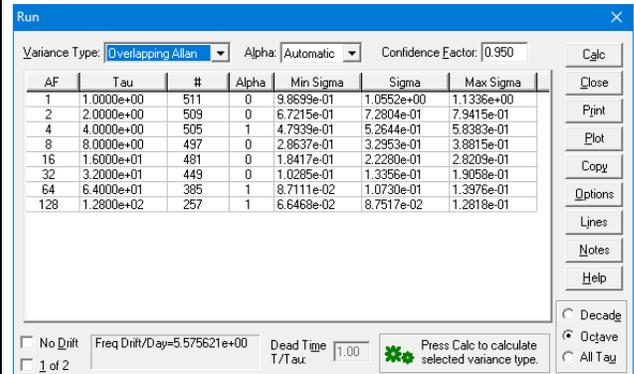
### Exercise #2: Confidence Intervals

1. Generate some simulated power law noise of your choosing using the Noise function. Here we show 512 points of W FM noise.
2. Open the Sigma function, choose and calculate a variance type, and examine the items related to confidence intervals. Here we show the overlapping Allan variance.

<sup>18</sup> The confidence intervals details are shown in the Stable32 Sigma function.



3. We see that, for these 511 analysis points, the number of equivalent  $X^2$  degrees of freedom is about 400 (due to the overlapping samples used), and, for a 95% confidence factor, the corresponding values from the  $X^2$  distribution are about 457 and 347, which determines ADEV error bars of about 0.98 and 1.13 around a nominal value of about 1.06.
4. You can see these error bars on the corresponding ADEV Run function table and plot.



5. Observe the same information for different variance types, noise types, numbers of data points, etc.

### 3.3 Higher-Confidence Statistics

See: §2.1, [46]

Considerable progress has been made since the introduction of the original Allan variance toward methods for improving the confidence estimators for it and related statistics. The original Allan variance was followed by its Overlapping version, the Total variance and then the Thêo1 along with its Overlapping/Thêo1 hybrid, ThêoH. Total versions of the Modified, Time, and Hadamard variances have also been devised. Figure 6 shows this progress for the Allan variance family. In cases where long measurement times are needed (e.g., 1-month runs), the savings in time and cost can be considerable.

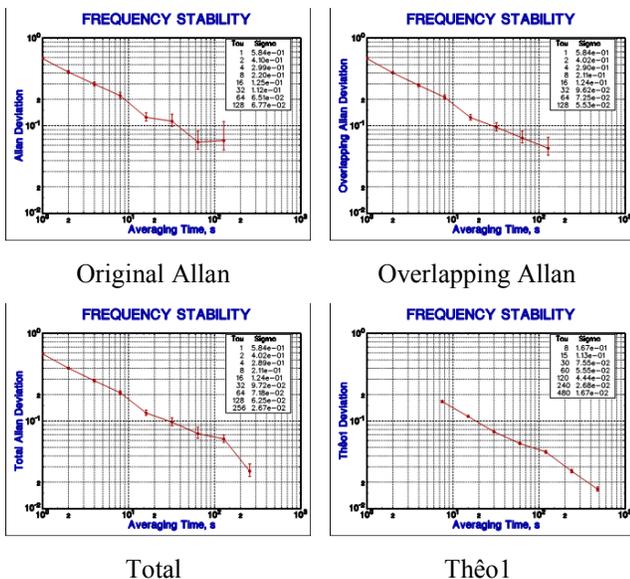


Figure 6 Progress in Frequency Stability Analysis

The confidence improves, the error bars shrink and the tau range expands as the stability measure evolves from the original Allan to the Overlapping Allan, Total and Thêo1 statistics.

### 3.4 Spectral Analysis

See: §5.3, [2], [28], [29], [30], [33], [49]

The field of spectral analysis is large and complex, and plays an important part in frequency

stability analysis. Unlike most spectral analysis however, the emphasis is on characterizing noise rather than discrete components. And unlike the calculation of statistics such as the overlapping Allan variance, there is no standard methodology for spectral analysis – it is a combination of art and science. While the relationship between a time series and its spectrum is exact and invertible using the Fourier transform or an FFT, a spectral analysis involves additional considerations such as windowing (see §6.5) and averaging (see §6.6) for which there is no single correct methodology.

Some aspects of spectral analysis may not be obvious or intuitive at first. The sampling interval,  $\tau$ , of the time domain data determines the upper Fourier frequency,  $1/2\tau$ , of the spectrum per the Nyquist criterion of 2 samples/cycle. Similarly, the number of data points,  $n$ , determines the Fourier frequency spacing,  $1/n\tau$ , and the lower Fourier frequency extent of the spectrum. More data does not reduce the variance of the spectrum; one must use averaging to accomplish that. Because the Fourier transform is linear in frequency, most of the spectral points are at the high frequency end of a log frequency plot. Some sort of log averaging or segmentation can improve that. Windowing (tapering) at the ends of the time series data is necessary to avoid [Gibbs phenomenon](#) distortion at low Fourier frequencies, although matching the time series end points by removing a trend line and/or performing pre-whitening can help. For the purposes of frequency stability analysis, it is important to get the magnitude and effective noise bandwidth of the spectrum correct.

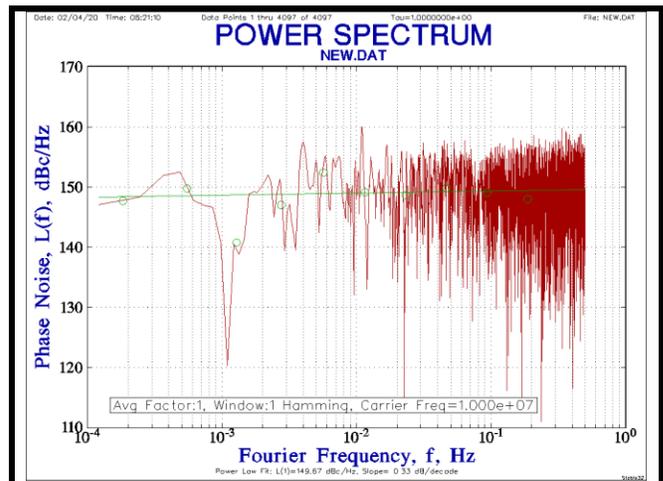
Frequency domain stability analysis uses several spectral measures for both phase and frequency data, and since they are all mathematically related, conversions are possible between them (see §6.1 and §6.2). For phase data, these are  $S_x(f)$ , the spectral density of the time variations,  $\text{sec}^2/\text{Hz}$ ,  $S_\phi(f)$ , the spectral density of the phase variations,  $\text{rad}^2/\text{Hz}$ , and  $\mathcal{L}(f)$ , the SSB phase noise,  $\text{dBc}/\text{Hz}$ . For frequency data, it is  $S_y(f)$ , the spectral density of the fractional frequency variations,  $1/\text{Hz}$ .

Stable32 supports all of these spectral measures and attempts to present them with reasonable default settings and in their proper engineering units. The frequency stability analysis student can use those plots to compare them and observe the spectral characteristics of various power law noises.

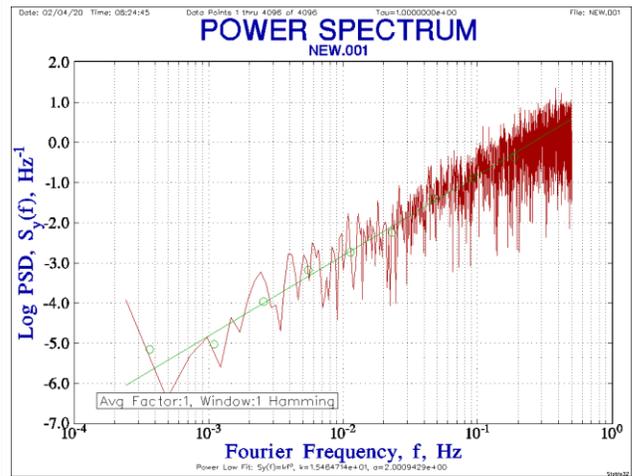
Try the various PSD plot types and their many options for windowing, smoothing and averaging. Try pre-whitening, filtering and trend removal as ways to improve PSD analysis.

### Exercise #3: Compare PSD Plots

1. In this exercise, we will compare various measures of power spectral density.
2. Generate 4096 points of white PM noise with the Noise function (set the # points to 4097). The magnitude and tau can both be set to 1.0 since we only care about the noise type.
3. Select the phase data.
4. Launch the Power function.
5. Select and plot the  $S_x(f)$  PSD plot. Observe the flat fit to the W PM noise with a power law exponent B near zero. Note: You may want to copy (Copy/Bitmap) and paste these PSD plots to more easily compare them.
6. Close the plot.
7. Select and plot the  $S_\phi(f)$  PSD plot. Observe the flat fit to the W PM noise with a power law exponent B near zero.
8. Close the plot.
9. Select and plot the  $\mathcal{L}(f)$  PSD plot. Observe the flat fit to the W PM noise with a slope near zero.



10. Close the plot.
11. Close the Power function.
12. Select the frequency data.
13. Launch the Power function again.
14. Plot the  $S_y(f)$  PSD plot.
15. Observe the power law exponent near +2. The 0 power law exponent of white PM noise is increased by 2 by the differencing that converts it to frequency noise (see §3.4).



16. Close the plot. Try other power law noise types.

### 3.5 Domain Conversions

See: §7

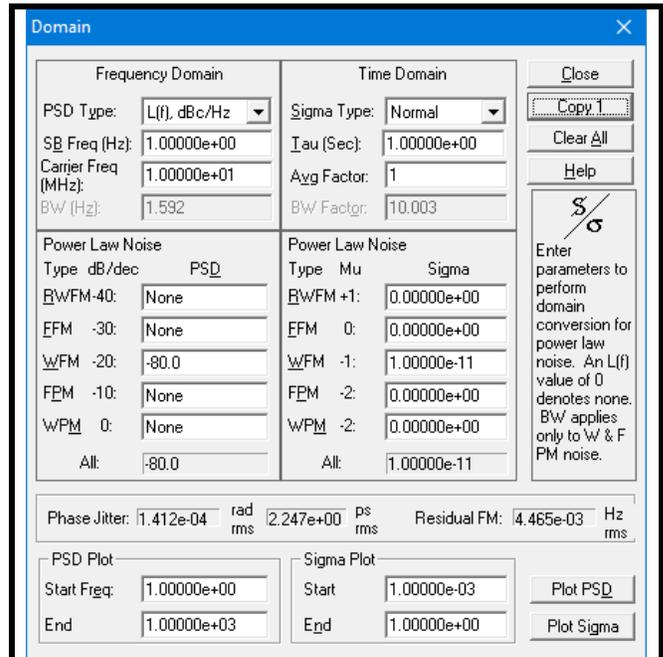
Conversions are possible between measures of frequency stability in the time and frequency domains. While mathematically related, performing those conversions can be quite complicated (see §7) in general, but formulae are available for the case of common power law noise types (see §7.1), and the later conversions

are supported by the Stable32 Domain function. The power law model works for multiple noise sources in most cases because the phase noise falls off versus higher sideband frequency. However, it can be inapplicable for some composite cases (e.g., where there is a narrowband filter in the source output).

One important domain conversion application is to verify that the specifications of a frequency source are consistent between domains. Phase noise specifications usually dominate for sideband frequencies of 1 Hz and above while time domain specifications usually cover averaging times longer than 1 second, so the two domains only overlap at those edges. But, increasingly, instruments are available to make time sub-second time domain measurements.

**Exercise #4: Check Spec Consistency**

1. Consider a 10 MHz rubidium frequency standard (RFS) that has a specified 1-second short-term stability (STS) of  $1 \times 10^{-11}$  and a specified phase noise of -80 dBc/Hz at 1 Hz from the carrier. Are these time and frequency domain values consistent?
2. The RFS STS can be presumed to be based on white FM noise, e.g.,  $1 \times 10^{-11} \tau^{-1/2}$ . Enter that noise type and level, along with the 10 MHz carrier frequency, into the Domain function as the time domain stability, and observe the corresponding frequency domain level of phase noise at 1 Hz, e.g.,  $\mathcal{L}(1) = -80$  dBc/Hz.



3. Yes, the specifications are consistent.
4. If you wish, other aspects of domain conversion, etc can be explored. For example, change the carrier frequency to 100 MHz and you will see that the x10 multiplication has degraded the phase noise by 20 dB.

**3.6 Sampling Functions and Frequency Response**

See: [5], [44], [45]

The time domain statistics used in frequency stability analysis involve the sampling of time series data in particular patterns that determine their impulse response and spectral properties in the frequency domain. For example, the impulse and frequency responses of the AVAR are shown in Figures 7 and 8 (other patterns and responses apply to the MVAR, HVAR and other statistics). These sampling functions relate to their autocorrelations and other underlying properties, and they also apply to the way certain instruments make phase and frequency measurements.

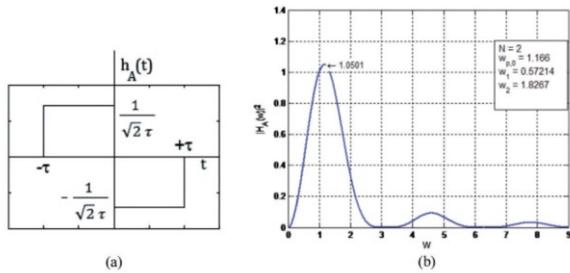


Fig. 13. – Impulse response (a) and power frequency response (b) of the AVAR algorithm.

Figure 7. Impulse and Frequency Response of the AVAR Sampling Function (From [44]).

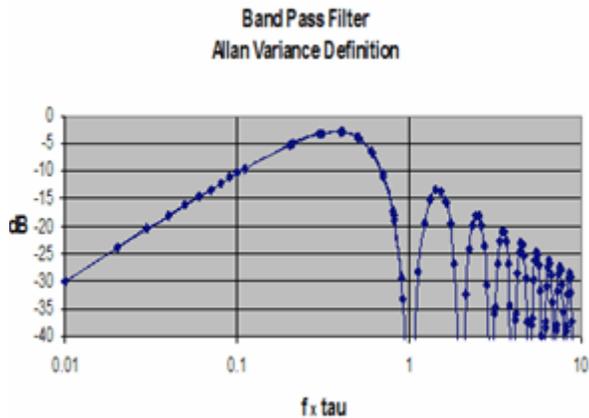


Figure 8. Frequency Response of the AVAR (From [5]).

The frequency response of AVAR looks like a 1/2-octave-wide band pass filter. The peak in the response is at  $f \cdot \tau = 0.5$ , or at  $\tau = 0.5/f = 0.5T$ , where  $f$  is a Fourier component of fractional frequency deviation  $y(t)$  and  $T$  is the period of that component.

### 3.7 Sigma-Tau Plots

See: §5.1

The most common way to show the stability of a frequency source in the time domain is by means of a sigma-tau plot. That is a log-log plot of some stability measure such as the Allan deviation versus averaging time. A sigma-tau diagram not only shows the stability as a function of the averaging time, but the slope of the plot indicates the type of power law noise that applies, as shown in Figure 9. Similar relationships apply to Modified and Time variances (see Figures 10 and 11). The Stable32 Run function generates sigma-tau tables and plots for phase or frequency data.

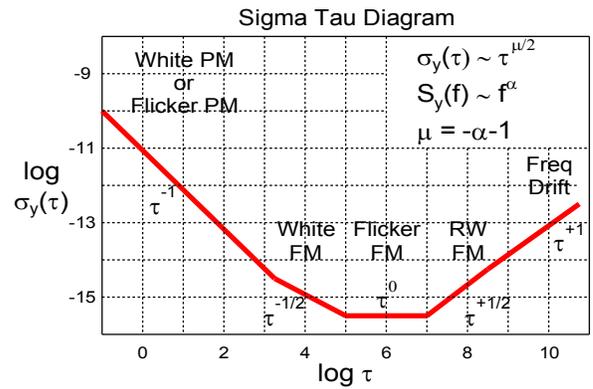


Figure 9. Sigma-Tau Diagram

The slope of the log sigma versus log tau plot indicates the type of power law noise that applies.

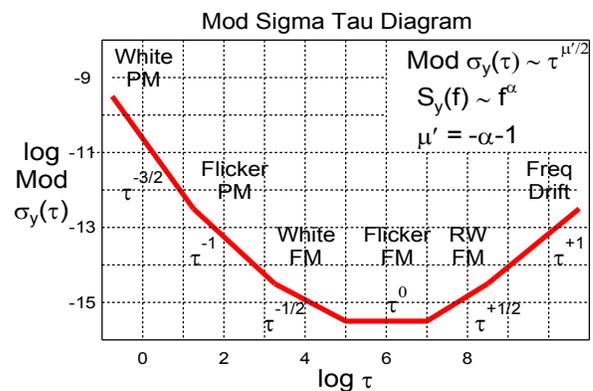


Figure 10. Mod Sigma-Tau Diagram

The slopes for white and flicker PM noise are different in a Mod sigma-tau plot.

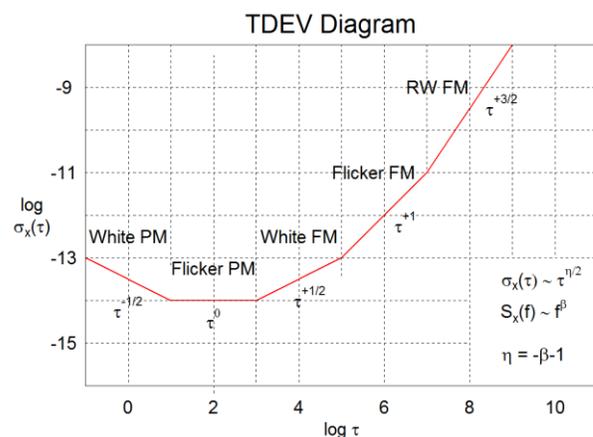


Figure 11. Time Sigma-Tau Diagram

The slopes of the TDEV sigma-tau diagram are those of MDEV +1 reflecting its multiplication by  $\tau$ .

An “All Tau” stability plot often provides the best information and is a form of spectral analysis since it shows small ripples.

### 3.8 L(f) Plots

See: §6.3

The most common way to show the stability of a frequency source in the frequency domain is by means of a  $\mathcal{L}(f)$  plot<sup>19</sup>. That is a plot of the SSB phase noise in logarithmic units of dBc/Hz versus log Fourier frequency in Hz. The Stable32 Power function generates  $\mathcal{L}(f)$  and other related power spectral plots for phase or frequency data, as shown in Figure 12.

Stable32 is mainly a time domain stability analysis tool in that it accepts that form of phase or frequency data. It can, however, convert the time domain data into several forms of spectral plots.

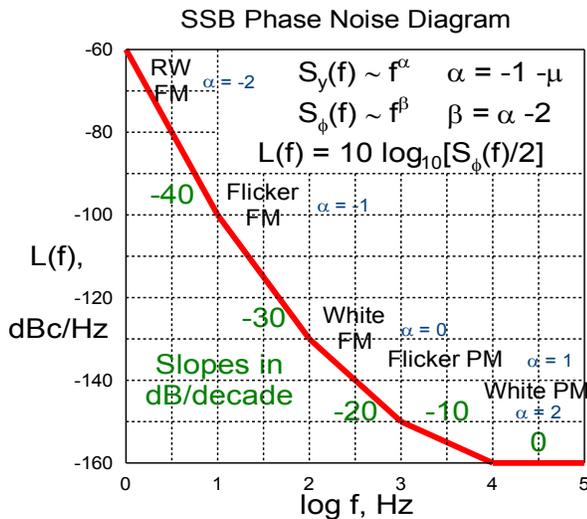


Figure 12. SSB Phase Noise Diagram

The slope of the  $\mathcal{L}(f)$  versus  $\log f$  plot indicates the type of power law noise that applies.

You will quickly see that the ADEV plots are identical for the same phase and frequency data. You can switch to the Octave or Decade option and see how the ADEV error bars increase at larger averaging factors as the number of sam-

<sup>19</sup> The script  $\mathcal{L}$  symbol that represents log SSB phase noise is commonly written as either L or  $\mathcal{L}$ . That notation was introduced by D. Halford.

ples becomes smaller. You can also change the confidence factor for the error bars.

### 3.9 Choice of Variance Type

See: §5.2, §10.13

Stable32 offers a complete set of variance types that you can try and compare as discussed below<sup>20</sup>.

### 3.10 Standard Variance

See: §5.2.1

The standard variance and its square root, the standard deviation, are familiar statistical measures of variability. Unfortunately, it does not converge for some of the power law noise types commonly associated with frequency sources, and is therefore not used for that purpose.

### 3.11 Allan Variance

See: §5.2.2

The Allan variance<sup>21</sup> was devised as a way to avoid the non-convergence of the standard variance for flicker FM ( $\alpha=-1$ ) and other divergent noise types. It, in effect, replaces comparisons against the average with 1<sup>st</sup> differences of the frequency data, as we saw earlier in this tutorial.

### 3.12 Overlapping Samples

See: §5.2.3

The original form of the Allan variance made its samples with a stride equal to the averaging factor thus using each value only once. In contrast, the overlapping form of the Allan variance uses a stride of one as it moves through the data, thereby using each point multiple times in making its samples. While not completely independent, these overlapping samples nevertheless contribute to a larger number of statistical degrees of freedom and improve the confidence of the estimate (see *Handbook* Figure 7). The software hides this complexity from the user,

<sup>20</sup> As you read these various variance descriptions, please refer to the referenced *Handbook* sections for their mathematical and other details.

<sup>21</sup> The [Allan variance](#) was devised by D.W. Allan in 1966.

and the additional time required for the more complicated calculation is negligible.

### 3.13 Overlapping Allan Variance

See: §5.2.4

The Allan variance, AVAR, and its square root, the Allan deviation, ADEV, is the most commonly specified and used measure of frequency stability, is recommended in most cases, particularly its Overlapping form. The original non-overlapped version is now deprecated because it is a less efficient estimator, and is included in Stable32 mainly for historic purposes or when it is explicitly specified.

The AVAR is based on the 1<sup>st</sup> differences of the fractional frequency values, or equivalently, 2<sup>nd</sup> differences of the phase.

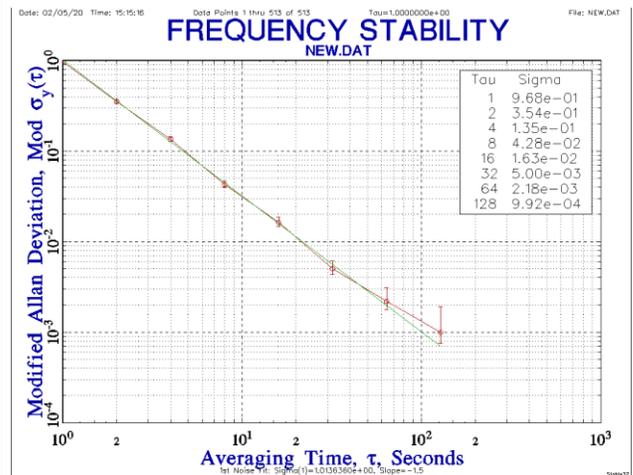
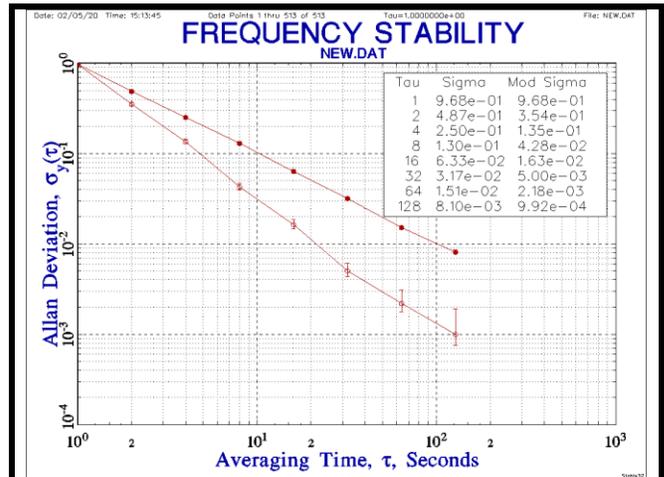
### 3.14 Modified Allan Variance

See: §5.2.5

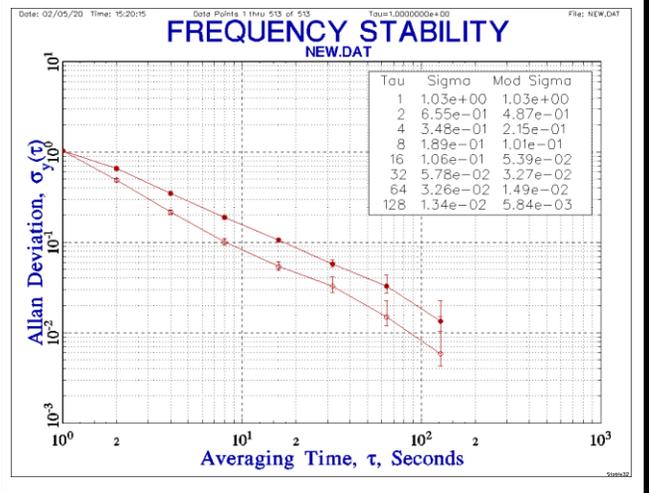
The Modified Allan variance<sup>22</sup>, MVAR, and its square root, the Modified Allan deviation, MDEV, use phase averaging in their estimation formula. The main use of the Modified Allan variance (MVAR) is to distinguish between white and flicker phase noise (something that can be important in analyzing a time transfer system). It is also the basis of the Time variance.

#### Exercise #5: Use MVAR to Distinguish Between White and Flicker PM Noise

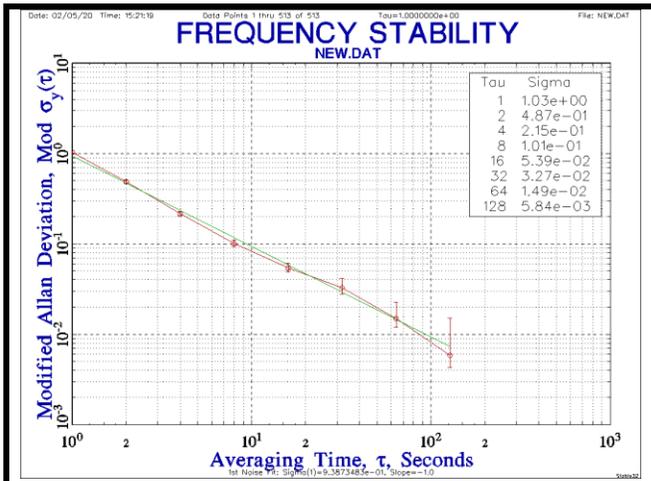
1. Generate some white PM noise.
2. Analyze and plot it with the combination of ADEV and MDEV and MDEV alone, the latter with a W PM -1.5 slope fit.



3. Generate some flicker PM noise.
4. Analyze and plot it with the combination of ADEV and MDEV and with MDEV alone, the latter with a F M -1.0 slope fit.



<sup>22</sup> The [Modified Allan variance](#) was devised by D.W. Allan and J.A. Barnes in 1981.



Note that the MDEV and ADEV are the same at unity averaging factor, and that MDEV is smaller at higher averaging factors. The ADEV slope is the same for these two noise types, and the MDEV slopes are different.

### 3.15 Time Variance

See: §5.2.6, [21]

The Time variance<sup>23</sup>, TVAR, is based on the Modified Allan variance, and the time deviation, TDEV, is its square root. A scaling factor makes TDEV equal to the standard variance of the time deviations for W PM noise at the basic sampling rate. Because of the close relationship between TDEV and MDEV, an MDEV plot can include loci of constant TDEV as shown in Figure 13.

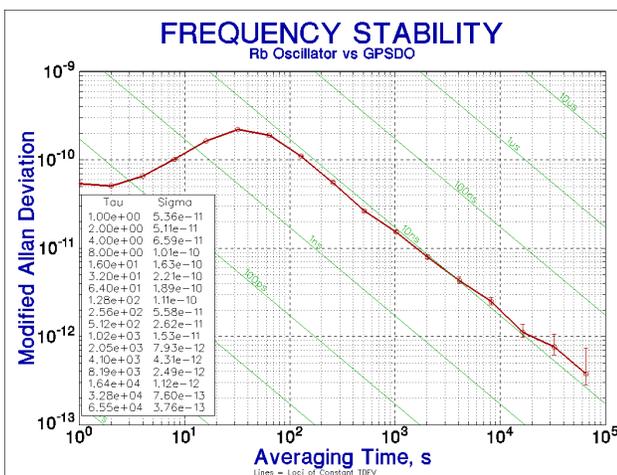


Figure 13. MDEV Plot with Lines of Constant TDEV

<sup>23</sup> The [Time variance](#) was devised by D.W. Allan, Marc Weiss, et al in 1990.

TDEV is an excellent measure of the noise of a timing source or time distribution system, for example, the 1-day TDEV nicely describes the stability of a GPS satellite clock between its daily updates.

### 3.16 Time Error Prediction

See: §5.2.7

A time error prediction typically includes terms representing the initial synchronization error, the linear time error caused by clock frequency offset, a quadratic term due to its frequency drift, and a TDEV noise estimate versus elapsed time. Depending on the application, additional time errors may need to be included due to environmental effects (e.g., temperature, barometric, etc. sensitivities). An example of such a time error budget would be for a GPS disciplined oscillator during holdover [47].

### 3.17 Hadamard Variance

See: §5.2.8, [17]

The Hadamard variance<sup>24</sup>, HVAR, and its square root, the Hadamard deviation, HDEV, is a 3-sample version of the 2-sample Allan variance<sup>25</sup>. As such, it extends its ability to handle more divergent noise Random Walk FM ( $\alpha=-3$ ) and Random Run ( $\alpha=-4$ ) FM noise, and it is insensitive to linear frequency drift.

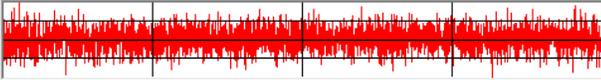
#### Exercise #6: HDEV Rejects Frequency Drift

1. In this exercise, we will see how the Hadamard deviation rejects linear frequency drift.
2. Normally, one would generate simulated noise data with frequency drift in the Noise function. But here we want to start without drift and add it later.
3. Use the Noise function to generate 4096 points of white FM noise.

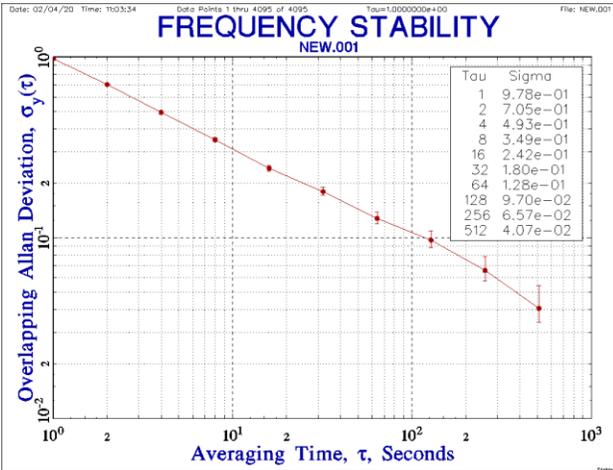
<sup>24</sup> Named for [Jacques Hadamard](#) (1865-1963), a French mathematician. The Hadamard variance was introduced to the frequency control field by R.A. Baugh in a 1971 FCS [paper](#).

<sup>25</sup> The [Picinbono variance](#) is a similar 3-sample statistic.

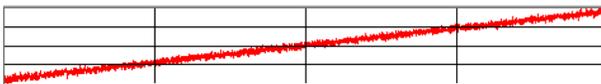
- Observe the resulting W FM noise data in its status plot.



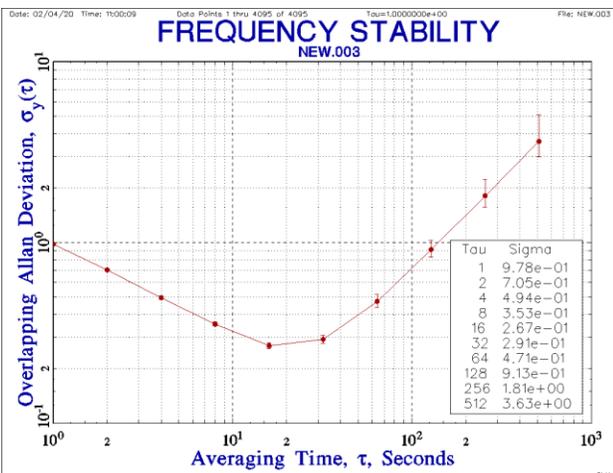
- Activate the Tabs feature and save this current drift-free frequency data in case you want to use it again.
- Perform an Overlapping ADEV Run and plot those drift-free results



- Open the Scale function and use it to add a drift slope of 0.01 to the frequency data.
- Observe the resulting frequency drift in its status plot.

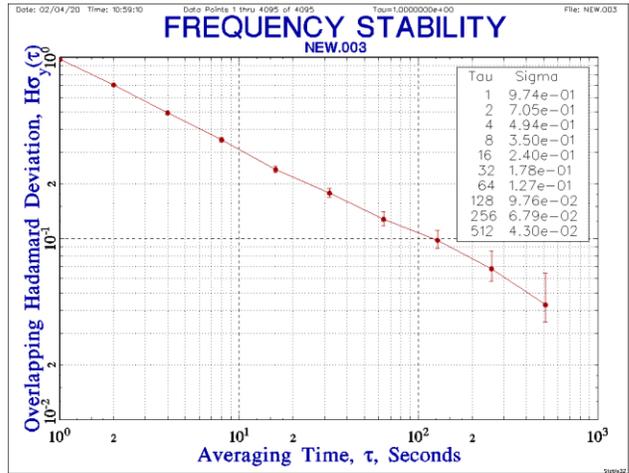


- Perform an Overlapping ADEV Run and plot those results. Note that the ADEV increases at longer tau as  $\tau^{+1}$  because of the drift.



- Perform an Overlapping HDEV Run and

plot those results. Note that the HDEV has rejected the frequency drift, and that it is essentially the same as the ADEV results without drift.



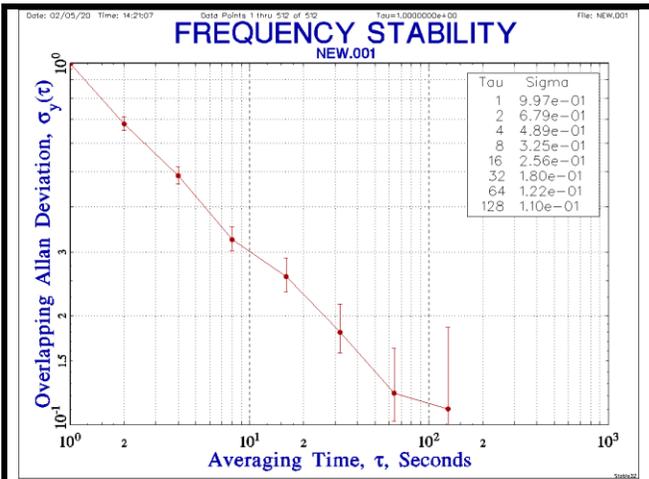
- If you wish to revisit the drift-free data, be sure to save the current drift data in another tab.

Why not use the Hadamard variance all the time since it gives essentially the same result as the Allan variance and you don't have to worry about the effect of (linear) frequency drift? Well mostly because the Allan variance (deviation) is what is generally specified and is the more familiar. Plus it is better practice to deal with any drift separately. But more importantly, ADEV is a more efficient estimator, providing a higher-confidence result.

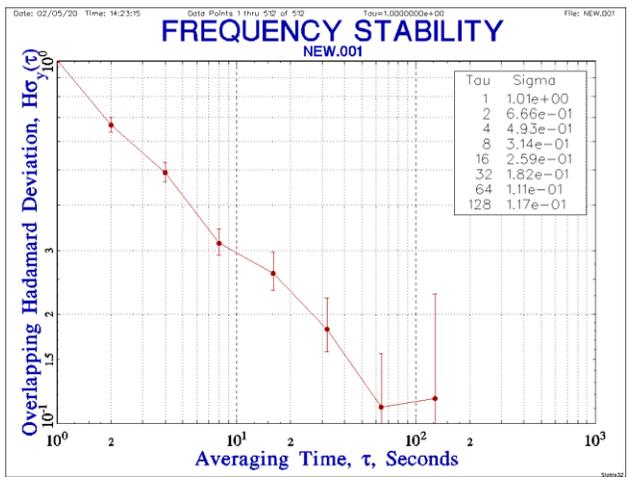
There are original, Overlapping, Modified and Total versions of the Hadamard variance; we used the overlapping version above since it is preferred over the original one that provides less confidence. The Modified Hadamard variance is a 3-sample statistic with phase averaging similar to the Modified Allan variance (see below) and the Total Hadamard variance is a version of it that has higher confidence at long averaging factors (see below).

### Exercise #7: HDEV Has Wider Error Bars Than ADEV

- Generate some test noise
- Calculate and plot ADEV



### 3. Calculate and plot HDEV



### 4. Observe that HDEV has wider error bars.

## 3.18 Overlapping Hadamard Variance

See: §5.2.9

Like the Overlapping version of the Allan variance, the Overlapping Hadamard variance uses fully overlapping samples to estimate that statistic with higher confidence.

## 3.19 Modified Hadamard Variance

See: §5.2.10

Like the Modified version of the Allan variance, the Modified Hadamard variance uses phase averaging in its computation to distinguish between white and flicker PM noise, to handle more divergent noise types, and to reject linear frequency drift.

## 3.20 Total Variance

See: §5.2.11, §5.11, [23]

The Total variance<sup>26</sup> is a higher-confidence estimator of the Allan variance, offering useful results at averaging factors that extend to a third of the record length by reflecting the data at its ends. It is a biased estimator for Flicker FM ( $\alpha=-1$ ) and random walk FM ( $\alpha=-2$ ) noise. One should not be concerned about using a biased estimator like the Total variance because the Stable32 software that supports its calculation also includes bias corrections<sup>27</sup>. There are also Total versions of the Modified, Time and Hadamard variances (see below).

The Total deviation is included in the analysis techniques recommended in IEEE Std 1139-1999 [6], but is not often used, mainly because most users are more familiar with the Allan and Hadamard deviations, and the newer Thêo1 statistic is even more advantageous.

Try these Total and Thêo1 stability measures with your own clock data and see for yourself how they can improve the confidence and extend the range of your analyses.

## 3.21 Modified Total Variance

See: §5.2.12, §5.12, [51]

The Modified Total variance is a higher-confidence estimator of the Modified Allan variance. It accomplishes that by extending the data at both ends by uninverted even reflection, allowing averaging factors up to one-third of the record length.

## 3.22 Time Total Variance

See: §5.2.13

The Time Total variance is a higher-confidence estimator of the Time variance based on the Modified Total variance.

<sup>26</sup> The [Total variance](#) was devised by D.A. Howe.

<sup>27</sup> These bias corrections generally depend on the noise type, and that determination is made at each averaging factor during a stability analysis.

### 3.23 Hadamard Total Variance

See: §5.2.14, [24], [50]

The Hadamard Total variance is a higher-confidence estimator of the Hadamard variance. It accomplishes that by extending the data at both ends by uninverted even reflection, allowing averaging factors up to one-third of the record length.

### 3.24 Thêo1

See: §5.2.15, §5.13, [54]

The Thêo1<sup>28</sup> statistic is another, even higher-confidence estimator of the Allan variance, supporting results out to three-quarters of the record length, where 10 terms contribute to the result. It is a biased estimator for all except W FM noise, and an empirical correction factor is applied for the other noise types.

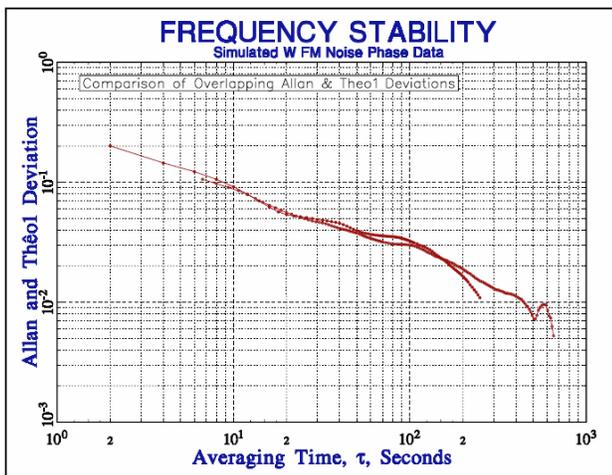


Figure 14. ADEV and Thêo1 Compared

Thêo1 provides useful stability results at larger averaging factors.

The information provided by a sigma-tau plot is logarithmic in time. It is one thing to add another point to a 1-second data run by doubling the 1-hour measurement time and quite another to do so for the hourly data of a 1-month duration. The latter involves significant monetary and schedule costs, and is where the advantage if the Thêo1 statistic becomes most apparent.

<sup>28</sup> The [Thêo1](#) and [ThêoH](#) statistics were devised by D.A. Howe in 2004-2006.

### 3.25 ThêoH

See: §5.2.16, §5.14

The ThêoH statistic is a hybrid of the Overlapping Allan variance and Thêo1, the latter used to extend the analysis seamlessly to longer averaging factors out to nearly 75% of the record length. An example of a ThêoH stability plot is shown in Figure 15 where a 30-day run provides a stability value at 21 days<sup>29</sup>.

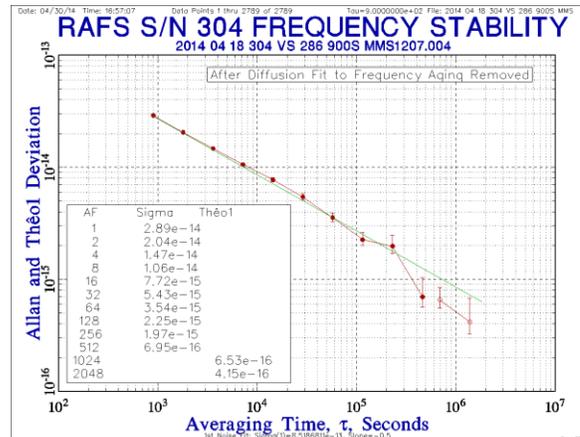


Figure 15. ThêoH Stability Plot for a GPS Satellite Rubidium Clock

The Thêo1 statistic provides a stability estimate out to 71% of the 30-day record length.

### 3.26 MTIE

See: §5.2.17, [52], [53]

The Maximum Time Interval Error, MTIE, is a different sort of statistic since it is based on the maximum value of the time error as a window is passed through the phase data. The MTIE statistic is mainly used within the telecommunications industry. MTIE is a peak deviation not closely associated with clock noise types, is sensitive to a single extreme value, and is computationally quite intensive<sup>30</sup>.

<sup>29</sup> This stability plot is actual data for an Excelitas GPS III Rb clock during its 30-day factory acceptance test. Because it is still stabilizing, the frequency drift is removed by a diffusion fit, which is an excellent model for the aging of its Rb physics package. It displays an  $8.5 \times 10^{-13} \tau^{-1/2}$  W FM noise characteristic that extends over 3 weeks without reaching any apparent “flicker floor”.

<sup>30</sup> The Stable32 Run/MTIE function includes a faster approximate MTIE calculation option.

### 3.27 TIE rms

See: §5.2.18

The rms Time Interval Error is another measure of time variability, the rms value of the 1<sup>st</sup> differences of the phase data. For no frequency offset, it is approximately equal to the standard deviation of the fractional frequency fluctuations multiplied by the averaging time. TDEV is a better measure of time deviation for divergent noise sources.

### 3.28 Integrated Phase Jitter and Residual FM

See: §5.2.19

The integrated phase jitter and residual FM are alternative ways to express phase and frequency jitter. The former integrates the phase noise over a certain bandwidth to produce a phase jitter value in rms radians. That is most often done using a power law noise model, often in a [spreadsheet](#). The residual FM is similar, giving a frequency jitter value of rms Hz.

### 3.29 Dynamic Stability

See: §5.2.20, [\[41\]](#)

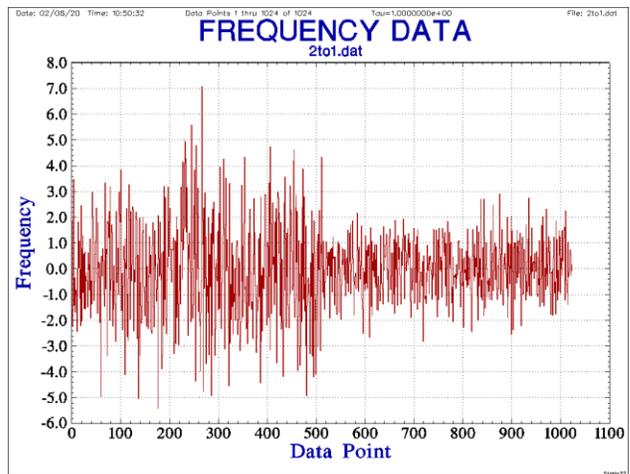
The Dynamic Stability<sup>31</sup>, DVAR function produced a 3D ADEV or HDEV plot that shows the variation in stability as a function of the data point, showing changes in log sigma versus log averaging factor as a function of time as an analysis window is moved through the data set.

#### Exercise #8: Try the Dynamic AVAR Function

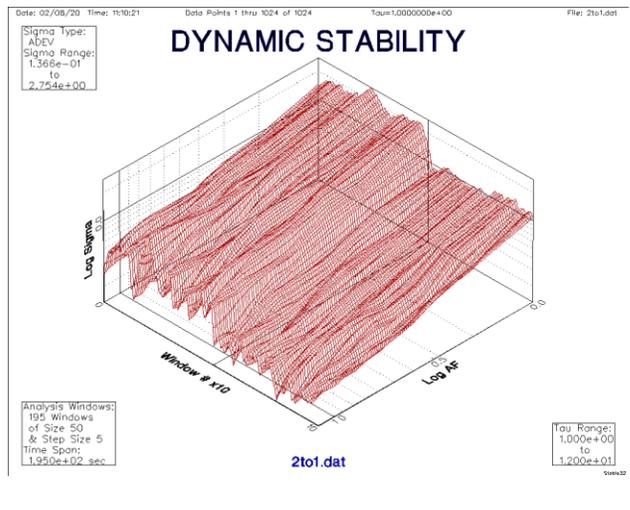
1. Generate 512 points of W FM noise with an ADEV of 1.0 (or some other simulated noise of your choosing).
2. Save the resulting frequency data.
3. Generate another set of the same type of noise but with twice the ADEV.
4. Use the Add function to add on the first set of data.
5. Plot the composite data. Clearly, the noise

<sup>31</sup> The [Dynamic Allan variance](#) was devised by L. Galleani and P. Tavella in 2005.

increases at the middle.



6. Launch the DVAR (Dynamic AVAR) function and plot the dynamic stability using the defaults. As expected, the stability changes at the middle of the plot.
7. You can experiment with different window and step sizes. Here smaller ones (50 and 5) appear to better resolve this sharp change



### 3.30 Dead Time

See: §5.6, §5.8, §5.9, §5.15, [\[11\]](#)

Dead time between successive frequency measurements can affect the results of a stability analysis. In cases of modest dead time, the bias can be removed by applying the Barnes B<sub>2</sub> and B<sub>3</sub> bias ratios (see Figure 16), and this can be done automatically during a Stable32 Run. Frequency measurements with extreme dead time (e.g., hourly 100 second measurements) are suitable only for drift determination, not ADEV

analysis. Dead time is not a factor for phase measurements.

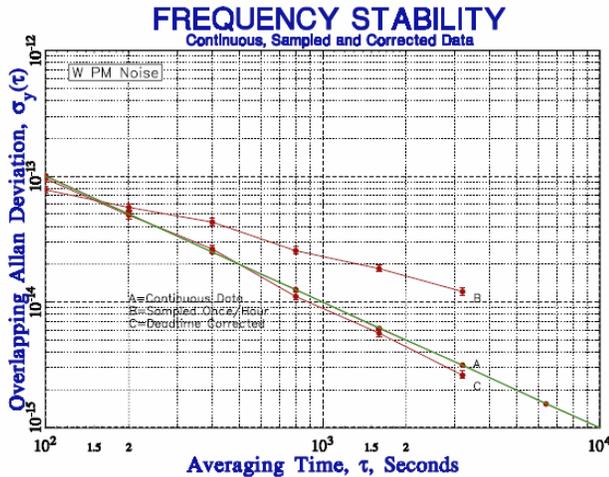


Figure 16. Dead Time Correction

The bias caused by measurement dead time can be corrected using the Barnes B2 and B3 ratios,

### 3.31 Unequally-Spaced Data

See: §5.16, [11]

Unevenly-spaced data is a problem for time series analysis. If the data are timetagged, and on a uniform grid, it is possible to automatically insert gaps to regularize the data<sup>32</sup>. That is effective only if the number of gaps is reasonable, and periodic gaps can cause strange results in an ADEV analysis. If the data are only slightly unevenly-spaced, that can simply be ignored, or the data can be forced to a uniform grid.

### 3.32 Two Identical Units

It is common to measure the stability of two nominally-identical frequency sources against each other, especially when a more stable reference is not available. In that case, it is reasonable to assume that their noise powers contribute equally, and to apply a correction factor of  $1/\sqrt{2}$  to obtain the stability of one unit<sup>33</sup>.

### 3.33 Three-Cornered Hat

See: §10.14, [18]

In the case multiple, not necessarily identical, units, the 3-cornered hat method<sup>34</sup> can be used to estimate the stabilities of the individual units.

### 3.34 Autocorrelation Function (ACF)

See: §5.5.3, §5.5.4, [1], [15]

Autocorrelation plots are seldom used to present final stability analysis results, but they can provide valuable insight into clock behavior, and the autocorrelation function serves as a theoretical underpinning for frequency stability analysis techniques<sup>35</sup>. A time-dependent autocorrelation implies non-stationarity. Inspection of an autocorrelation plot shows the extent that the adjacent data are correlated, indicates their power law noise type, and can show periodicities, as shown in the following exercise. In fact, the lag 1 autocorrelation function can be used to estimate the power law noise type quite accurately (see [1]). A lag 1 scatter plot shows the dispersion of the adjacent values around the diagonal and shows if the data are quantized (see [Exercise #8](#)).

#### Exercise #9: Try the Autocorrelation Function

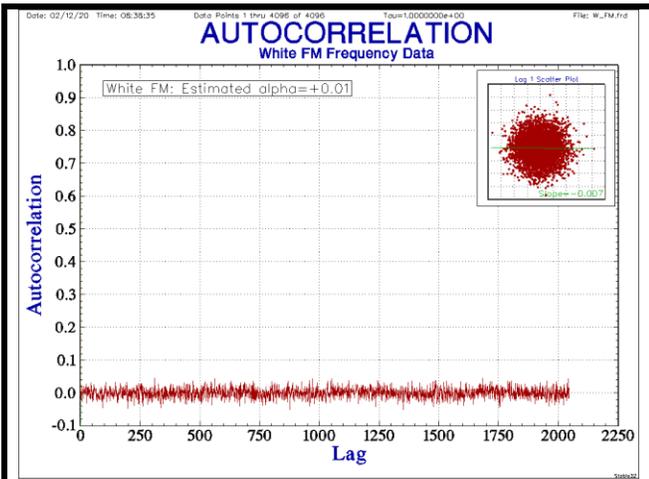
1. Generate or read W FM noise frequency data.
2. Open the Autocorrelation (ACF) function, activate the Lag 1 scatter plot insert, and plot the autocorrelation function.

<sup>32</sup> See the Stable32 Regularize function.

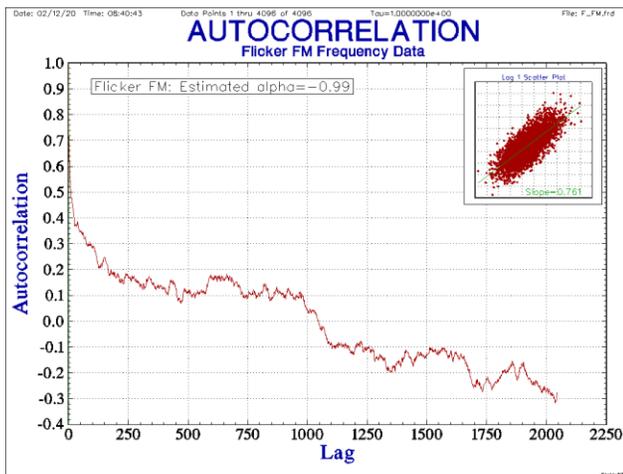
<sup>33</sup> The Stable32 Run function supports this correction.

<sup>34</sup> The term "3-cornered hat" was coined by J.E. Gray.

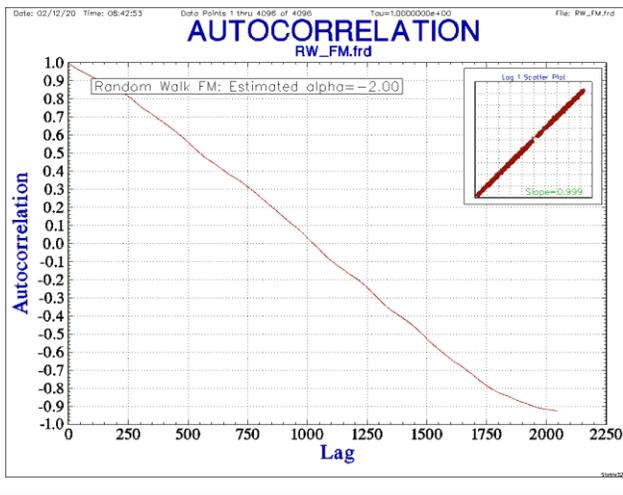
<sup>35</sup> The ACF and the power spectrum are related by the Fourier transform, and they are mathematically equivalent.



3. Repeat for F FM frequency data.

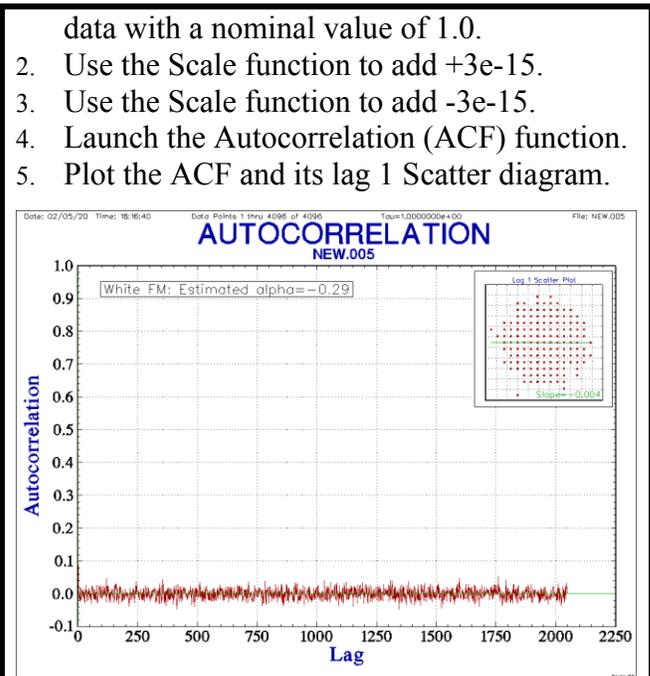


4. Repeat for RW FM frequency data.

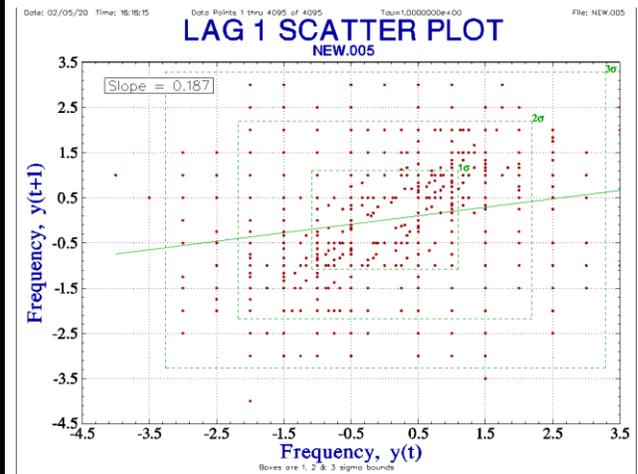


### Exercise #10: Use an Autocorrelation Scatter Plot to Show Data Quantization

1. Generate quantized data by exceeding its double-precision dynamic range. Start with 4096 points of simulated W FM frequency



6. The ACF plot shows that the noise is uncorrelated even at lag 1 indicating that it is white, and the estimated alpha confirms that it is white FM.



7. The scatter plot shows the 0.5 data quantization.

### 3.35 Histogram

See: §5.17

It can sometimes be useful to examine clock data with a histogram, perhaps to look for bimodality or quantization. A histogram works best when there is little or no slope to the data. Stable32 can fit the histogram to a normal distribution.

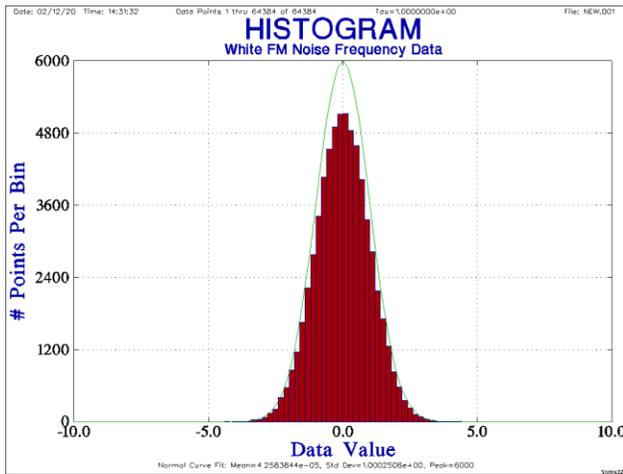


Figure 17. Histogram of W FM Frequency Data

### 3.36 Frequency Offset

See: §5.18

The fractional frequency offset of a source with respect to the measurement reference is equal to the average frequency. It can be estimated from phase data by any of the following three methods:

1. A least-squares linear fit to the phase data. This is optimum for white PM noise.
2. The average of the 1<sup>st</sup> differences of the phase data. This is optimum for white FM noise.
3. The difference between the first and last points of the phase data divided by their time span. This can be used to match the two endpoints.

### 3.37 Frequency Drift

See: §5.19, [55]

Frequency drift (see [Drift Analysis, Modeling, and Removal](#)) refers to the change in average frequency versus time and includes the frequency aging of the source due to internal effects plus those caused by environmental and other factors.

### 3.38 Performing a Stability Analysis

See: §10

Section 10 of the *Handbook* has subsections that delineate the main considerations and steps for a complete frequency stability analysis as follows:

1. Data Precision
2. Preprocessing

3. Gaps, Jumps and Outliers
4. Gap Handling
5. Uneven Spacing
6. Analysis of Data with Gaps
7. Phase-frequency Conversions
8. Drift Analysis
9. Variance Analysis
10. Spectral Analysis
11. Outlier Recognition
12. Data Plotting
13. Variance Selection
14. Three-Cornered Hat
15. Reporting

In many cases, one goes immediately to a variance analysis, but hopefully only after plotting the data to visually confirm that it has sufficient precision, is evenly spaced without gaps, jumps or outliers, and does not require prior drift removal. That visual inspection is generally best done on frequency data. One must also consider what variance type is best suited to the purpose at hand, and whether the data needs spectral analysis or frequency domain characterization. Some situations may call for post-processing by combining stability results at different sampling rates or the use of cross-correlation or three-cornered hat processing. Finally, the results are incorporated into a report to document and interpret them.

There is an example of a typical analysis procedure at the end of the *User Manual*<sup>36</sup>.

### 3.39 Preprocessing

See: §10.2

As mentioned above, the first steps in performing a stability analysis may involve preprocessing operations to prepare the phase or frequency data for detailed analysis. Generally, these steps are:

1. Visual inspection of the data
2. Outlier removal
3. Drift analysis and removal

<sup>36</sup> See pages 293-294 and 299-302.

### 3.40 Drift Analysis, Modeling, and Removal

See: §5.19, §5.20, [8], [31], [55]

The purpose of a frequency drift analysis can be either to determine the amount of drift or to model and remove it from the data (or both). Good practice usually dictates the removal of deterministic frequency drift before quantifying the stochastic noise. The decision whether drift removal is needed can usually be made by inspection of a plot of the frequency data. In some applications, the need for drift removal is avoided by using the Hadamard variance<sup>37</sup>. A linear model for frequency drift (or equivalently, a quadratic phase model) is the most obvious and most widely used, but other models such as log and diffusion fits are sometime better. The goal of removing drift from frequency data is to obtain white residuals to the greatest extent possible.

Stable32 supports a variety of drift models for either phase or frequency data as shown in Table 3, and it also includes polynomial fits and the ability to use an arbitrary fit equation. High-order polynomial fits are discouraged as non-physical and fragile. The most useful non-linear fits have proven to be a log and diffusion ( $\sqrt{t}$ ) fits to model a stabilization or actual gas diffusion process.

Table 3. Stable32 Drift Analysis Methods		
Data Type	Fit Method	Noise Model
Phase	Quadratic Fit	W PM
	Average of 2 <sup>nd</sup> Differences	RW FM
	3-Point Fit	W & RW FM
	Greenhall	All
	Linear Fit	Freq Offset
	Average of 1 <sup>st</sup> Differences	Freq Offset
	Phase Endpoints	Freq Offset
Freq	Linear Fit	W FM
	Bisection Fit	W & RW FM
	Log Fit	Stabilization
	Diffusion Fit	Diffusion
	Autoregression	AR(1)

Four methods are available for modeling the frequency drift of phase data:

1. The first method is a standard least-squares quadratic fit to the phase data. This method is optimum for white PM noise, and is the one most commonly used.
2. The second method is the average of the 2<sup>nd</sup> differences of the phase data. This method is optimum for random walk FM noise.
3. The 3-Point method uses the 3 points at the start, middle and end of the phase data. It is the equivalent of the bisection method for frequency data.
4. The Greenhall method uses 4 points at the start, 10%, 90% and ends of the phase data. It is applicable to all noise types.

Three methods are available for determining the frequency offset of phase data:

1. The first method is a least-squares linear fit to the phase data. This method is optimum for white PM noise.
2. The second method is the average of the 1<sup>st</sup> differences of the phase data. This method is optimum for white FM noise.
3. The third method simply uses the difference between the first and last points of the phase data. This method is intended mainly to match the two endpoints.

Five methods are available for modeling the frequency drift of frequency data:

1. The first method, the default, is a least squares linear regression to the frequency data. This is the optimum method for white FM noise.
2. The second method computes the drift from the frequency averages over the first and last halves of the data. This bisection method is optimum for white and random walk FM noise.
3. The third method, a log model of the form (see [MIL-PRF-55310](#)), applies to the frequency stabilization of crystal oscillators and rubidium frequency standards.
4. The fourth frequency drift method is a diffusion ( $\sqrt{t}$ ) model that has been found effective for high-performance rubidium frequency standards.
5. In addition, there is an autoregression frequency drift method that can fit and optionally remove AR(1) autoregressive noise from the data. That method calculates the lag 1 autocorrelation coefficient,  $r(1)$ , for a selected averaging factor and uses it to remove a first-order AR(1) noise process. It is useful for prewhitening data before a jump analysis.

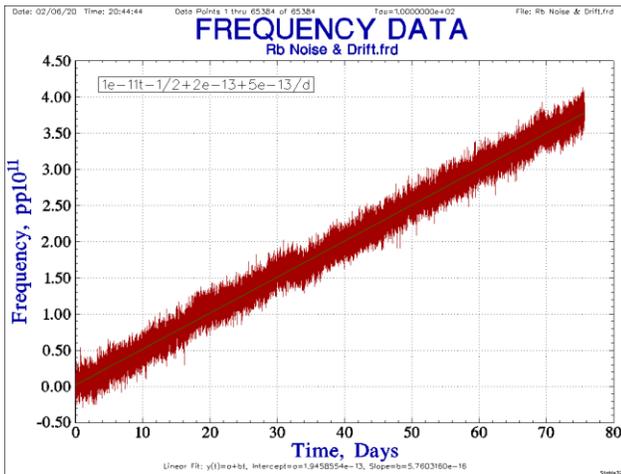
#### Exercise #11: Frequency Drift Model

1. Generate a set of white and flicker FM noise plus linear frequency drift using the Noise function. Simulate a medium-performance rubidium frequency standard having a 1-second W FM level of 1e-11 and a flicker

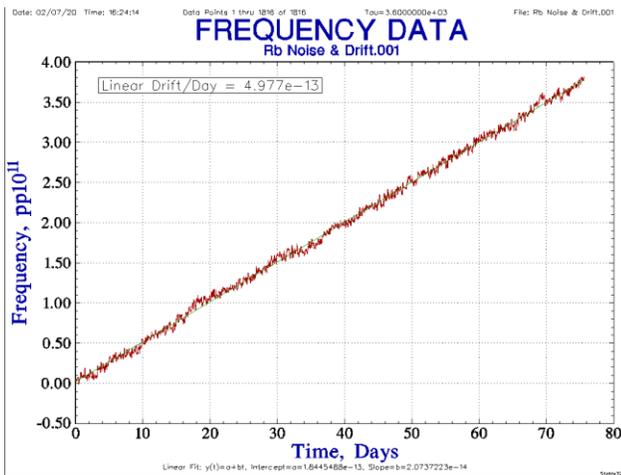
<sup>37</sup> For example, GPS Rb satellite clocks.

floor of  $2e-13$ , plus a frequency drift of  $5e-13$  per day. Use a 100 second sampling interval ( $\tau$ ) which makes the drift per tau equal to about  $5.787e-16$ . 65,384 points will represent a time span of about 75 days and support ADEV analysis out to about  $10^6$  seconds.

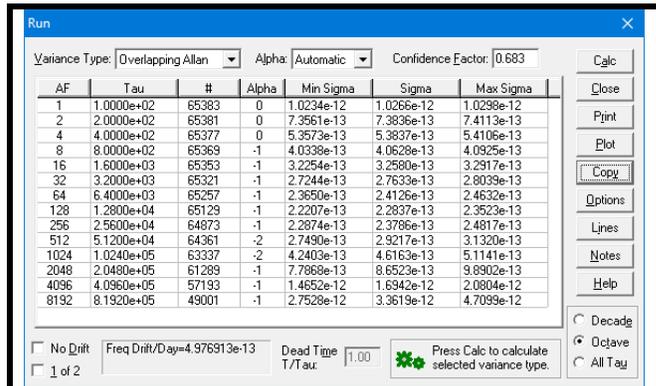
- Plot and annotate the resulting frequency data plot, including a linear fit.



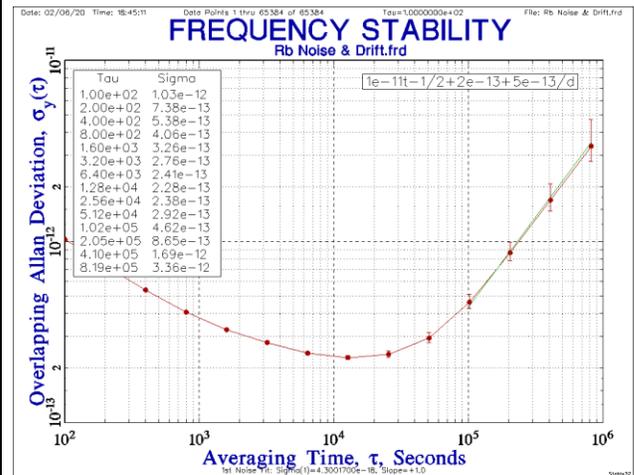
- The linear fit is about  $5.76e-16$  per 100 second tau interval as expected.
- Average the frequency data by a factor of 36 to a 1 hour tau to better show the flicker noise. Also add a message showing the daily drift.



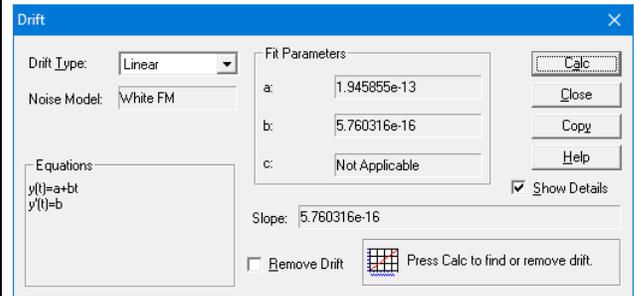
- Plot the ADEV table for the 100 second data.



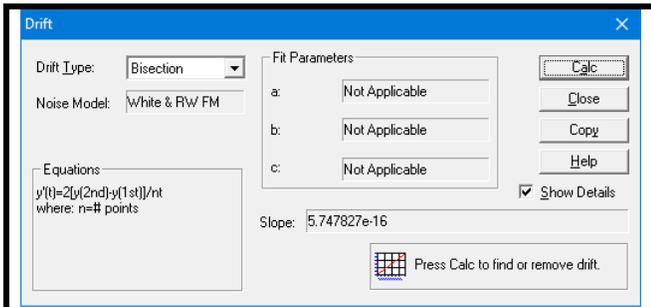
- Note that the reported linear frequency drift is  $4.98e-13$ /day, very close to that expected.
- Plot and annotate the ADEV results.



- The ADEV plot shows the W FM level of about  $1e-12$  at 100 seconds, a stability floor that approaches  $2e-13$ , and frequency drift with a  $+1$  slope at longer tau.
- Use the Drift function to calculate the linear frequency drift.



- The reported linear frequency drift is about  $5.76e-16$  per 100 second tau interval or about  $4.97e-13$  per day, again as expected.
- Repeat using the Bisection drift method.



12. The Bisection drift value is about  $5.75e-16$  per tau interval, essentially the same as for the linear fit.

The Stable32 Plot function provides access to a number of fitting methods via its Options/Fit Type control.<sup>38</sup>

### 3.41 Outliers

See: §10.3, [11], [13], [39]

It is possible to have outliers in a set of phase or frequency data, and those must be removed before performing a stability analysis. Outliers are best handled as frequency data, and the first step (as in any analysis) is to inspect the data for any problems.

A gross outlier is easy to spot since it will dominate the data plot, and it is quite easy to remove it by using the Edit function and replacing it with a value of 0. Stable32 treats all frequency zero values and interior phase zero values as gaps (first and last zero phase values are allowed). The zeros maintain the continuity of the data samples. If an actual value of zero is needed, a value of (say)  $1e-99$  can be used for it.

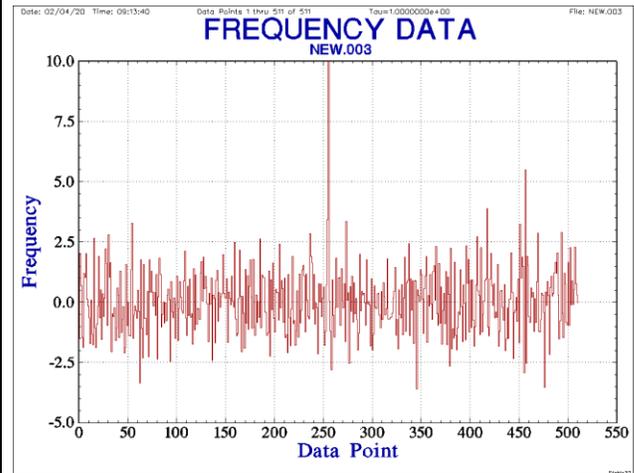
But a better way to identify outliers is to use the Check function which automatically finds and optionally removes outliers subject to a certain criterion based on the Median Absolute Deviation (MAD), a robust way to handle them.

<sup>38</sup> For phase data, these are Quadratic, Average, Polynomial, and Function. The latter two lead to Poly Order and  $x(t)=$  settings respectively. For frequency data, these are Line, Log, Diffusion, Average, Polynomial, Function, and Jumps. The latter lead to Poly Order,  $y(t)=$ , and Configure respectively. Configure opens a Frequency Jump Detection Parameters dialog box.

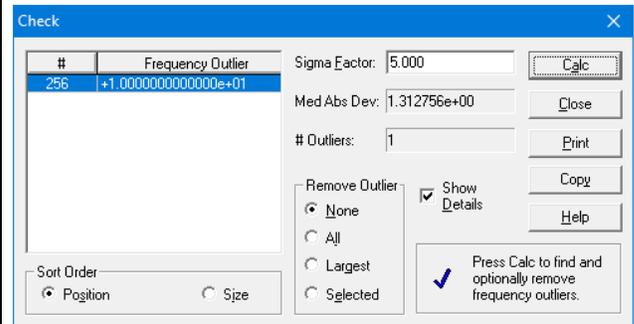
Good practice requires that all outliers removed be noted and explained to the extent possible.

### Exercise #12: Find and Remove Outliers

1. In this exercise, we will see how to identify and remove outliers.
2. Generate 512 points of simulated white FM noise data with the Noise function. The magnitude should be set to 1.0, and the tau can be left at 1.
3. Select the frequency data.
4. Open the Edit function and scroll to the middle of the frequency data (say point # 256) and click on it.
5. Change the value of the middle point to 10, press Enter, and then OK.
6. Use the Plot function to plot the frequency data. Note the outlier at the center.  
Note that the data filename extension has been incremented to indicate that it has changed (phase data extensions are even, frequency extensions are odd).



7. Launch the Check function, press Calc, and note that one outlier is detected. Note also that the point at about #470 is somewhat below the outlier threshold of about 6.6.



8. Try changing the Sigma Factor to see how it affects the outlier detection sensitivity.
9. Select Remove Outlier/All and press Calc to remove the outlier.
10. Explore multiple and other outlier options as desired. See how outliers affect the Statistics.

### 3.42 Jumps

See: §10.3, [19], [20], [32]

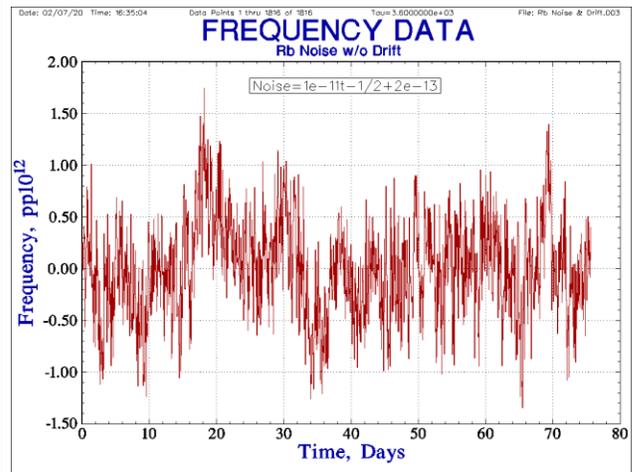
Jumps occur in phase and frequency data in the form of spikes and transients, back-and-forth excursions, steps, lurches, etc. These are generally more obvious in frequency data where as a jump rather than a change in phase slope. In interpreting them, one needs to keep in mind their integral/derivative relationship to help understand their underlying cause. In some cases, there is a obvious frequency jump, perhaps associated with a change in a device monitor signal; in other cases divergent noise can show a sudden lurch that only appears to be a jump. Besides an abrupt phase or frequency change, a frequency source can also experience a sudden change in its stability. Any of these phenomena can be a one-time isolated event, occur in bursts or a periodic way, or with some other pattern. One also needs to distinguish between the unit under test, the reference and the measuring system as the cause of the anomaly.

Jump analysis can be done by visual examination of a phase or frequency record, by some automated process, or a combination of the two. Some such algorithms involve comparing the mean values of two adjacent windows that are moved through the data, while others use cumulative sum charts and other similar techniques. Some of these methods require a data model whose parameters must be determined, while other methods are non-parametric. They can variously detect outliers, transients, level shifts and changes in variance for data having different noise characteristics and trends

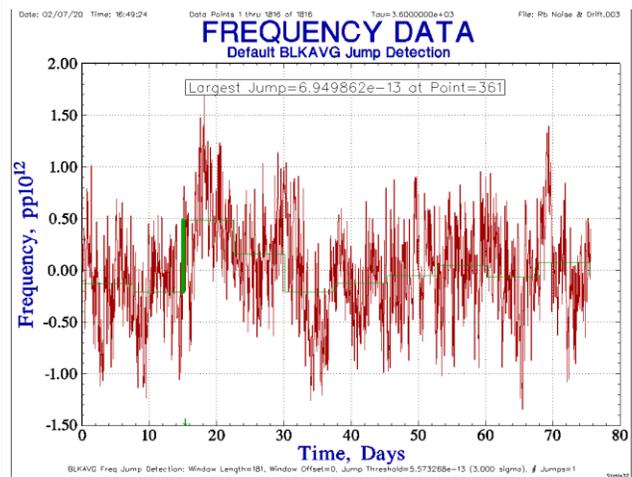
Besides phase and frequency plots, Stable32 has a jump detection feature, and also supports a dynamic stability function that can show ADEV or HDEV stability changes.

### Exercise #13: Frequency Jumps

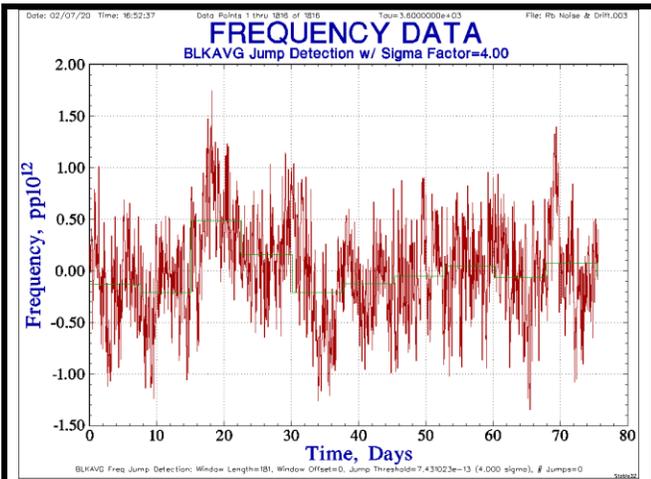
1. Start with the same simulated Rb clock noise as [Exercise #11](#) but with the linear drift removed, as shown in the following 1 hour tau frequency data plot. The white flicker FM noise has significant lurches but no obvious abrupt jumps.



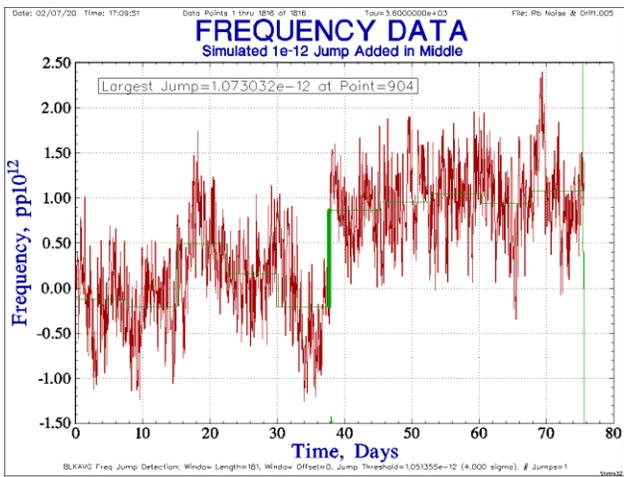
2. Nevertheless, the default BLKAVG jump detection function identifies one with its Sigma Factor=3.00. Note that the thin green lines indicate the detection blocks and the thick vertical line indicates the jump. The detection parameters are shown below the plot.



3. No jump is detected at Sigma Factor=4.00.



- A simulated jump can be inserted by adding an offset to part of the frequency data. Let's add an offset of  $1e-12$  in the middle at point 908. This is probably at about the threshold that one would detect a jump by eye, about half the peak-to-peak scatter but quite an obvious shift in the average.



- The jump is detected nicely, but we did have to adjust the sigma threshold up a bit to avoid a "false positive".

### 3.43 Filtering

See: [34]

Frequency domain filtering is seldom applied to time domain phase or frequency data. Low pass, band pass, high pass, or band stop filtration can nevertheless be performed as a preprocessing or investigative step as part of a frequency stability analysis.

Low pass filtration can be useful for removing high frequency noise that may otherwise obscure

underlying variations in the data. Its effect is similar to data averaging, but does not lengthen the sampling interval or reduce the number of data points.

High pass filtration can be useful for removing large amplitude low frequency fluctuation in the data due to divergent noise, drift or wandering in order to better see and analyze the high frequency noise. This is particularly effective when the drift or wandering does not fit a function to allow its removal.

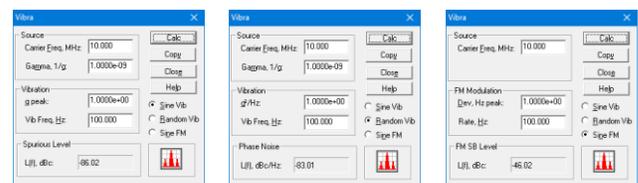
Band pass filtration can be useful for analyzing the amplitude variations of a discrete interfering component. Its function resembles that of a classic wave analyzer.

Band stop filtration can be useful for removing a discrete interfering component. By repeating this operation, multiple components may be removed without significantly affecting the underlying behavior.

### 3.44 Vibration

See: §11.5

Mechanical vibration can have a significant effect on the stability of a frequency source. This is particularly true for a quartz crystal resonator subjected to vibration [42]. The Stable32 Vibra function can calculate the level of spectral components caused by sinusoidal or random vibration applied to a crystal oscillator, based on narrowband FM theory that relates the level of sinusoidal frequency modulation and the resultant discrete spectral sidebands.



Spurious Level for crystal exposed to sinusoidal vibration. Phase noise for crystal exposed to random vibration. FM sideband level for a certain FM deviation and rate.

Figure 18. Stable32 Vibra Function Calculations

The ADEV of an oscillator subjected to sinusoidal FM will be degraded in a cyclic pattern as shown in Figure 19. This behavior can be simulated with the Noise function.

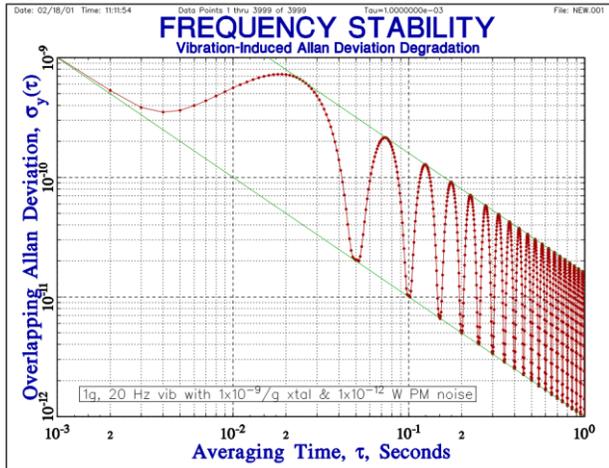


Figure 19. ADEV Plot with Vibration

Vibration-induced ADEV degradation of a crystal oscillator having a 1 second white PM noise level of  $1 \times 10^{-12}$  and an acceleration sensitivity of  $1 \times 10^{-9}/g$  exposed to sinusoidal vibration of 1g peak at 20 Hz. The W PM  $\tau^{-1}$  noise envelope is the lower green line and the vibration response  $\tau^{-1}$  envelope is the upper green line whose peaks occur at the modulation half cycles.

## 4 MEASUREMENTS & REPORTING

We present some suggestions for conducting and reporting clock measurements.

### 4.1 Measurements

See: §9, [37], [38], [40]

Except for simulations, clock measurements are obviously needed before a frequency stability analysis can be conducted. That means that a suitable measurement system must be available, along with an adequate reference, and that appropriate data must be captured and stored. Those matters are beyond the scope of this tutorial but are covered in Reference [40] and many other places. The resolution and noise floor of the measuring system are major considerations.

### 4.2 Clock Data

See: §9.6, §9.7

Phase, rather than frequency, data are preferred, and they generally should be formatted as double-precision exponential ASCII numeric-only values, one datum per line. Timetags are desirable (see below), preferably in the first, comma or space-delimited, column. The preferred units are time in seconds for phase data and dimensionless fractional frequency for frequency data.

When dealing with large amounts of clock data, it is recommended that they be stored in a formal database for easy retention and retrieval.

### 4.3 Timetags

See: §9.8

Phase and frequency data often needs to be timetagged or dated in some way. Within the time and frequency community, the 5-digit Modified Julian Date (MJD) is the most widely used format, having the convenient properties of a strictly numeric value that covers half of the 19<sup>th</sup> and into the 22<sup>nd</sup> centuries. It commonly includes a 5-digit decimal part that conveys time down to 1 second. The MJD is based on the longer Julian Date, the # of days since noon on January 1, 4713 BC, where  $MJD = \text{Julian Date} - 2,400,000.5$ . It starts at zero at midnight on November 17, 1858. The Stable32 program includes a number of features related to MJD timetags and corresponding date conversions, as well as a MJD Calendar (see its Help file and *User Manual* for more details).

It is easy to determine the current MJD by simply adding 40687 to the current UNIX computer time in seconds divided by 86400, where time starts at 0 hours on January 1, 1970. For example, in Python:

```
import time
...
# Get current MJD
now = time.time()
mjd = 40587.0 + now/86400.0
```

MJD timetags are usually written in 5.5 format as the 1<sup>st</sup> column of phase or frequency data, one row per datum. They are the recommended way to timetag such data.

## 4.4 Annotating Plots

See: §10.15

The plots shown in these examples are of the quick-and-dirty type that you'll get when doing the exercises. Real ones for a serious analysis can be more refined and either include additional information to document the results, or emphasize simplicity for a presentation. For the former, you'll want to include the header (date/time, points, tau, filename) and footer (organization name), add appropriate lines with their fit parameters, add notes, choose meaningful filenames, and store the data for future use (see Figure 14)<sup>39</sup>. For the latter, you'll want to minimize clutter for easier visibility. It is very easy to copy a plot and paste it into a document (the bitmap format is generally best). A small plot pasted into a document can be readily expanded on-screen for better readability.

## 4.5 Reporting

See: §10.15

Good practice dictates that the results of a significant frequency stability analysis be documented in a report<sup>40</sup>. Besides the usual items that describe the purpose, methodology, and results of the analysis, it is wise to retain the raw data in case it is needed again.

## 4.6 Other Examples

See: §11, [35]

Section 11 of the *Handbook* has several Case Studies showing various aspects of frequency stability analysis.

<sup>39</sup> One can even insert a small phase or frequency plot into a stability plot. See the Plot Inserts Help topic.

<sup>40</sup> Many Stable32 functions contain a Copy button to save the results to the Clipboard. Plots may be captured with the Edit/Copy(Bitmap) command. Most of the Convert options are now obsolete, but the bitmap image can be converted to another format (e.g., .png) by a Paint-type utility program. Clicking on a saved .tkf plot file will launch the Play utility to replay them.

## 5 CONCLUSIONS

We conclude by mentioning some leading contributors to the field of frequency stability analysis, and some suggestions for further study (see also Footnote 5).

### 5.1 Leaders of Frequency Stability Analysis

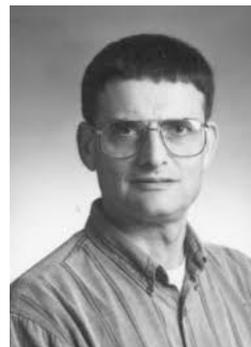
This tutorial has emphasized the Allan variance and related statistics that form the basis of time domain frequency stability analysis. Here we recognize some of the leaders of that field (see Figure 20). It's good to learn about the people in this field too.



James A. Barnes



David W. Allan



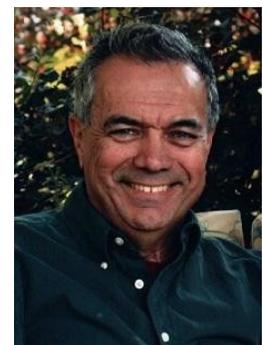
Charles A. Greenhall



François Vernotte



Marc Weiss



David A. Howe

Other contributors include: Claude Audion, Richard Baugh, Stefano Bregni, Andy Chi, Len Cutler, Jean-Jacques Gagnepain, David Leeson, Paul Lesage, Judah Levine, Demetrios Matsakis, Don Percival, Enrico Rubiola, Jacques Rutman, Sam Stein, Dick Sydnor, Patrizia Tavella, Jacques Vanier, John Vig, Fred Walls, Gernot Winkler, and many others.

Figure 30. Leaders of Frequency Stability Analysis

## 5.2 Summary

This document has hopefully served as a “mini-course” in frequency stability analysis, with the Stable32 program as its major teaching tool. Its “textbook”, the *Handbook of Frequency Stability Analysis*, is available from within the program with a click, as is its *User Manual*, and those documents contain the math and other details behind the analysis techniques. This document and those books, in turn, contain many references that provide opportunities for further study.

## ACKNOWLEDGMENT

The author acknowledges the helpful comments of Magnus Danielson regarding this tutorial<sup>41</sup>.

## ACRONYMS AND ABBREVIATIONS

ACF	Autocorrelation Function
ADEV	Allan Deviation
AVAR	Allan Variance
AFS	Atomic Frequency Standard
FCS	Frequency Control Symposium
FFT	Fast Fourier Transform
F FM	Flicker Frequency Modulation
F PM	Flicker Phase Modulation
FW FM	Flicker Walk Frequency Modulation
FW PM	Flicker Walk Phase Modulation
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPSDO	GPS Disciplined Oscillator
H	Hydrogen
HDEV	Hadamard Deviation
HVAR	Hadamard Variance
Hg	Mercury
IEEE	Institute of Electrical and Electronic Engineers
MAD	Median Absolute Deviation
MJD	Modified Julian Date

<sup>41</sup> Mr. Danielson leads a group that maintains and updates the Stable32 program.

NIST	National Institute of Standards and Technology
OCXO	Oven Controlled Crystal Oscillator
OCVCXO	Oven Controlled Voltage Controlled Crystal Oscillator
PLL	Phase Locked Loop
PM	Phase Modulation
PPS	Pulse Per Second
PSD	Power Spectral Density
PTTI	Precise Time and Time Interval
RAFS	Rubidium Atomic Frequency Standard
Rb	Rubidium
RbXO	Rubidium Crystal Oscillator
RF	Radio Frequency
RFS	Rubidium Frequency Standard
RW FM	Random Walk Frequency Modulation
RW PM	Random Walk Phase Modulation
RR FM	Random Run Frequency Modulation
RR PM	Random Run Phase Modulation
SSB	Single Sideband
S/N	Signal to Noise
STS	Short Term Stability
TDEV	Time Deviation
TVAR	Time Variance
T&F	Time and Frequency
UFFC	Ultrasonics Ferroelectrics and Frequency Control
VCXO	Voltage Controlled Crystal Oscillator
W FM	White Frequency Modulation
W PM	White Phase Modulation
XO	Crystal Oscillator

## REFERENCES

See: §14, [12]

1. W.J. Riley and C.A. Greenhall, “[Power Law Noise Identification Using the Lag 1 Autocorrelation](#),” *Proceedings of the 18<sup>th</sup> European Frequency and Time Forum*, April 2004.
2. D.B. Percival and A.T. Walden, *[Spectral Analysis for Physical Applications](#)*, ISBN 0521435412, Cambridge University Press, 1993.
3. R.W. Hamming, *[Numerical Methods for Scientists and Engineers](#)*, ISBN 007025887-2, McGraw-Hill Book Company, 1973.
4. D.B. Leeson, “[A Simple Model of Feedback Oscillator Noise Spectrum](#),” *Proceedings of the IEEE*, Vol. 54, No. 2, pp. 329–330, February 1966.
5. W.J. Riley, “[The Averaging of Phase and Frequency Data](#),” Stable32 web site, November 2011.
6. *IEEE Standard 1139*, “[IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology – Random Instabilities](#),” March 1999.
7. W.J. Riley, “[Confidence Intervals and Bias Corrections for the Stable32 Variance Functions](#),” Stable32 web site, August 2003.
8. W.J. Riley, “[Curve Fitting Features in Stable32](#),” Stable32 web site, July 2008.

9. W.J. Riley, "[The Basics of Frequency Stability Analysis](#)," Stable32 web site, August 2004.
10. W.J. Riley, "[Methodologies for Time Domain Frequency Stability Measurement and Analysis](#)," Stable32 web site, September 2007.
11. W.J. Riley, "[Gaps, Outliers, Dead Time, and Uneven Spacing in Frequency Stability Data](#)," Stable32 web site, October 2006.
12. W.J. Riley, "[References for Frequency Stability Analysis](#)," Stable32 web site, January 2007.
13. W.J. Riley, "[Outliers in Time and Frequency Measurements](#)," Stable32 web site, July 2013.
14. W.J. Riley, "[Techniques for Frequency Stability Analysis](#)," Stable32 web site, May 2003.
15. W.J. Riley, "[Use of the Autocorrelation Function for Frequency Stability Analysis](#)," Stable32 web site, August 2003.
16. W. J. Riley, "[The Calculation of Time Domain Frequency Stability](#)," Stable32 web site, July 2002.
17. W.J. Riley, "[The Hadamard Variance](#)," Stable32 web site, May 2007.
18. W.J. Riley, "[Application of the 3-Cornered Hat Method to the Analysis Of Frequency Stability](#)," Stable32 web site, January 2003.
19. W.J. Riley, "[Algorithms for Frequency Jump Detection](#)," Stable32 web site, December 2008.
20. W.J. Riley, "[Frequency Jump Detection and Analysis](#)," Stable32 web site, November 2008.
21. W.J. Riley, "[The Statistics of Time Transfer](#)," Stable32 web site, January 2016.
22. W.J. Riley, "[The Evolution of Frequency Stability Analysis Software](#)," Stable32 web site, July 2014.
23. C.A. Greenhall, D.A. Howe and D.B. Percival, "[Total Variance, an Estimator of Long-Term Frequency Stability](#)," Stable32 web site, August 1998.
24. D. Howe, et al., "[A Total Estimator of the Hadamard Function Used For GPS Operations](#)," Stable32 web site, November 2000.
25. C.A. Greenhall and W.J. Riley, "[Uncertainty of Stability Variances Based on Finite Differences](#)," Stable32 web site, October 2003.
26. W.J. Riley, "[Handbook of Frequency Stability Analysis](#)." You can buy a printed copy of this July 2008 book at [Handbook](#), and download it as [NIST Special Publication 1065](#). Please note that there is a typo in SP1065 Eq. 28 for TTOTVAR on p. 26: The tau exponent should be 2, not 3. And Eq. 9 is missing brackets around the inner summation – use Eq. 11 instead.
27. [D.B. Sullivan, D.W. Allan, D.A. Howe and F.L. Walls \(Editors\)](#), "[Characterization of Clocks and Oscillators](#)," *NIST Technical Note 1337*, U.S. Department of Commerce, National Institute of Standards and Technology, March 1990.
28. W.J. Riley, "[Stable32 Frequency Domain Functions](#)," Stable32 web site, August 2007.
29. W.J. Riley, "[Properties of FFT Windows Used in Stable32](#)," Stable32 web site, August 2007.
30. W.J. Riley, "[Evaluating the Level of Discrete Components in Stable32 PSD Plots](#)," Stable32 web site, August 2007.
31. W.J. Riley, "[Curve Fitting Features in Stable32](#)," Stable32 web site, August 2008.
32. W.J. Riley, "[Frequency Jump Detection in Stable32](#)," Stable32 web site, May 2008.
33. W.J. Riley, "[Spur Detection, Analysis and Removal in Stable32](#)," Stable32 web site, June 2008.
34. W.J. Riley, "[The Stable32 Filter Function](#)," Stable32 web site, July 2008.
35. W.J. Riley, "[The Strange Case of a Pseudo Flicker Floor](#)," Stable32 web site, May 2019.
36. [Stable32 User Manual](#), Hamilton Technical Services, LuLu Press, September 2008.
37. S.R. Stein, "Frequency and Time-Their Measurement and Characterization", Chap. 12, [Precision Frequency Control](#), E.A. Gerber and A. Ballato, Eds., Academic Press, Inc., Orlando, 1985, ISBN 0-12-280602-6 (Vol. 2).
38. D.A. Howe, D.W. Allan and J.A. Barnes, "[Properties of Signal Sources and Measurement Methods](#)," *Proc. 35th Annu. Freq. Control Symp.*, pp. A1-A47, June, 1981.
39. Gernot M.R. Winkler, "[Introduction to Robust Statistics and Data Filtering](#)," Tutorial at 1993 IEEE Freq. Contrl. Symp., June 1993.
40. W.J. Riley, "[Time and Frequency Measurements](#)," Stable32 web site, April 2019.

41. L. Galleani and P. Tavella, "[The Characterization of Clock Behavior with the Dynamic Allan Variance](#)," Proc. 2003 Joint FCS/EFTF Meeting, pp. 239-244, May 2003.
42. J.R. Vig, "[Introduction to Quartz Frequency Standards](#)," October 1992. [Section 6](#) has information about acceleration effects and g-sensitivity. See also [IEEE Standard 1193](#).
43. "[Allan Variance](#)," Wikipedia, In particular, see the "Educational and practical resources" section.
44. E. Bava, "[Frequency Instability: Characterization of Quasi-Periodic Signals](#)," Proceedings of the International School of Physics "Enrico Fermi" Course 185 "Metrology and Physical Constants", 2013.
45. E. Bava, "[Frequency Instability: Deterministic Contributions and Slowly Varying Noise](#)," Proceedings of the International School of Physics "Enrico Fermi" Course 185 "Metrology and Physical Constants", 2013.
46. W.J. Riley, "[The Evolution of Frequency Stability Analysis Software](#)," Stable32 web site, July 2014.
47. "[Miniature Atomic Clock Holdover](#)," White Paper, Microsemi Corporation, January 2016.
48. J. Rutman and F.L. Walls, "[Characterization of Frequency Stability in Precision Frequency Sources](#)," *Proceedings of the IEEE*, Vol. 79, No. 6, pp. 952-960, June 1991.
49. D.B. Percival. "[Characterization of Frequency Stability: Frequency-Domain Estimation of Stability Measures](#)," *Proceedings of the IEEE*, Vol. 79, No. 6, pp. 961-972, June 1991.
50. D.A. Howe, et al, "[Total Hadamard Variance: Application to Clock Steering by Kalman Filtering](#)," *Proceedings of the 2001 EFTF Conference*, March 2001.
51. D.A. Howe and F. Vernotte, "[Generalization of the Total Approach to the Modified Allan Variance](#)," Proceedings of the 31<sup>st</sup> PTTI Meeting, pp.267-276, December 1999.
52. Stefano Bregni, "[Clock Stability Characterization and Measurement in Telecommunications](#)," *IEEE Transactions on Instrumentation and Measurement* , Vol. 46, No. 6, Dec. 1997.
53. Stefano Bregni, "Measurement of Maximum Time Interval Error for Telecommunications Clock Stability Characterization," *IEEE Transactions on Instrumentation and Measurement* , Vol. 45, No. 5, October. 1996.
54. D.A. Howe and T.K. Pepler, "[Very Long-Term Frequency Stability: Estimation Using a Special-Purpose Statistic](#)," *Proceedings of the 2003 IEEE International Frequency Control Symposium*, pp. 233-238, May 2003.
55. W.J Riley, "[Frequency Drift Characterization in Stable32](#)," Stable32 web site, March 2020.