

Frequency Jump Detection in Stable32

W.J. Riley
Hamilton Technical Services

• Abstract

The detection of jumps in a frequency record is a challenging problem by either visual or mathematical means. The former takes considerable experience and judgment, and is therefore quite subjective, but has the advantage of providing insight into device behavior. The latter is more impartial and consistent, and can be automated. In combination, mathematical jump detection can be applied for automatic clock testing and monitoring. If a jump is detected, the record can then be inspected visually before deciding on the action required. Version 1.53 and higher of the Stable32 program for frequency stability analysis includes a frequency jump detection option in its frequency data plot functions (see Appendix I) [1].

• Introduction

There is a large body of technical literature on the subject of detecting discontinuities in time series (often referred to as regime shift or change-point analysis) [2, 4, 11]. Many 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 tools. Some of these methods require a data model whose parameters are 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 [10]. The emphasis here is on detecting jumps in reasonably white frequency residuals after outliers and deterministic drift have been removed from the data. Stable32 has a robust and effective means for outlier removal based on the median absolute deviation (MAD). It has several models for fitting and removing frequency drift. And it offers the dynamic Allan deviation (DAVAR) for examining clock data for stability changes [3]. This paper therefore covers the new Stable32 features for detecting, analyzing and displaying frequency jumps.

• The Stable32 Frequency Jump Detection Algorithms

Stable32 has a choice of two methods for frequency jump detection, using either a block averaging (BLKAVG) or a sequential averaging (SEQAVG) algorithm, both supplemented by the classic cumulative sum (CUSUM) method for analyzing data for changes in mean [4].

The Stable32 BLKAVG frequency jump detection algorithm is very intuitive. It simply compares the difference between the average values within a pair of adjacent windows of adjustable length against a selectable jump threshold. That threshold can be either an absolute fractional frequency limit or a sigma factor based on the Allan deviation at an averaging factor equal to the window length over the entire data set. The start of the data windows can be offset up to the amount that the data that exceeds the total span of the data windows.

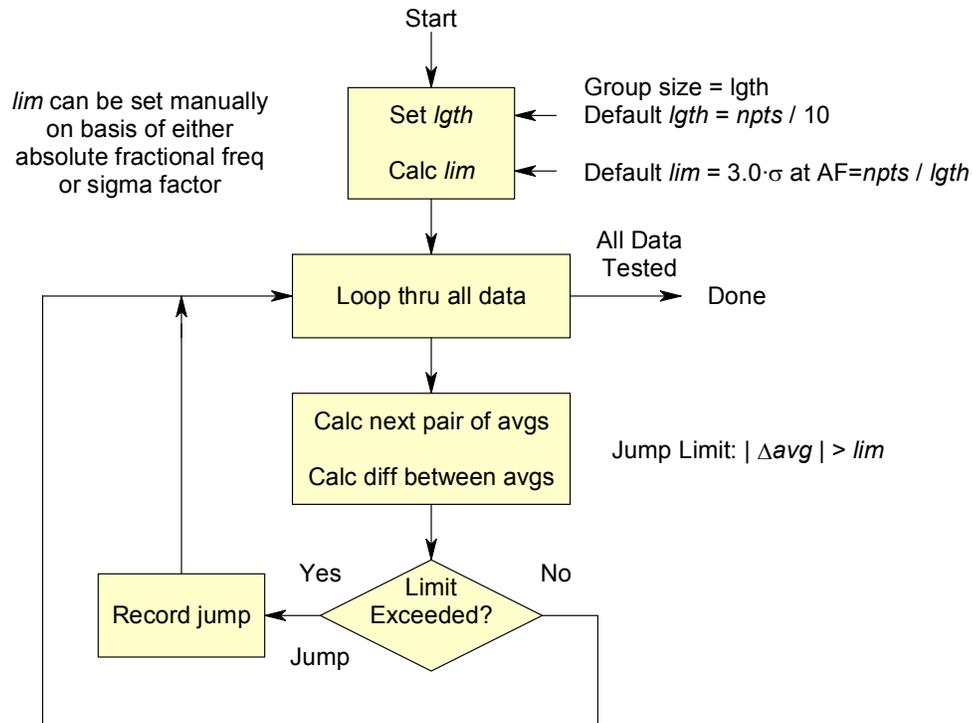
The SEQAVG method is based on the Rodionov Sequential T-test Analysis of Regime Shifts (STARS) algorithm [5]. It does not use fixed analysis windows, but rather scans the data sequentially. If a suspect point is found that exceeds the jump threshold, the next block of data is examined to confirm or reject the jump. Rather than using a jump threshold based on the Student's t-test [6], the SEQAVG method uses a fractional frequency jump limit.

The CUMSUM method is very effective for locating and quantifying single jumps in otherwise well-behaved frequency data. Interestingly, it is identical to the corresponding normalized (mean-removed) phase data, where a sharp slope change denotes a frequency jump.

• **Block Average Jump Detection Algorithm**

The Block Average (BLKAVG) frequency jump detection algorithm compares the difference between the average values within a pair of adjacent windows of adjustable length against a selectable jump threshold. That threshold can be either an absolute fractional frequency limit or a sigma factor based on the Allan deviation at an averaging factor equal to the window length over the entire data set. The start of the data windows can be offset up to the amount that the data that exceeds the total span of the data windows. The BLKAVG jump detection algorithm is shown in the following flowchart:

Block Average Jump Detection (BLKAVG) Algorithm

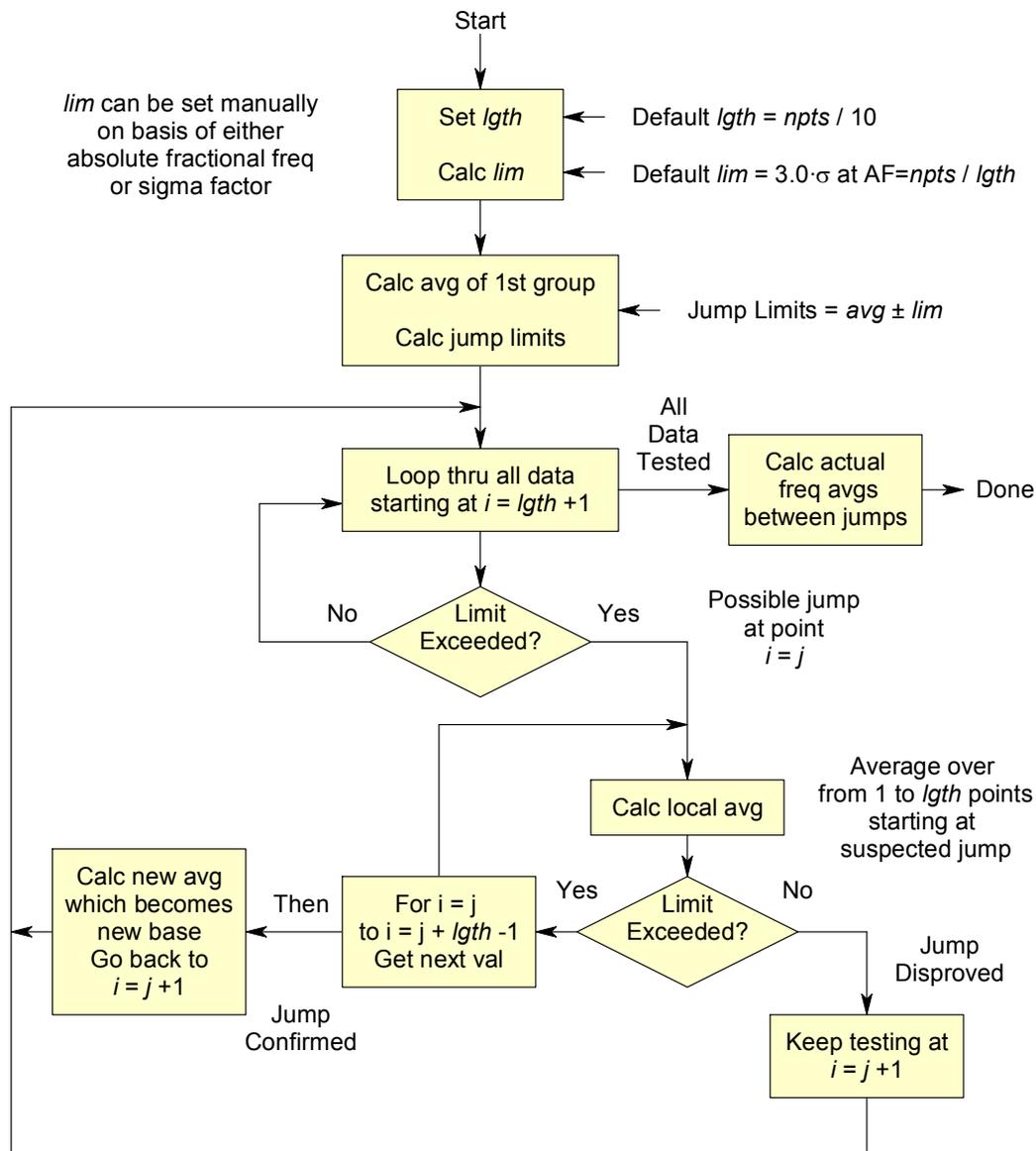


The visual presentation of a BLKAVG frequency jump analysis can be enhanced by plotting the average value of each analysis window along with the underlying frequency data (see Appendix I).

• **Sequential Average Jump Detection Algorithm**

The Sequential Average (SEQAVG) method is similar, but, instead of dividing the frequency record into fixed blocks, it examines the data sequentially. If a potential jump is detected, the following points are examined in an averaging window to accept or reject the jump. The SEQAVG plot shows the actual frequency averages in each jump regime, and is able to locate the jump more closely, but does not show trends in the data. The SEQAVG jump detection algorithm is shown in the following flowchart:

Sequential Average Jump Detection (SEQAVG) Algorithm



The visual presentation of a SEQAVG frequency jump analysis can be enhanced by plotting the average value of each jump regime along with the underlying frequency data (see Appendix I).

A comparison between the BLKAVG and SEQAVG methods is shown in the following table:

Comparison Between BLKAVG and SEQAVG Frequency Jump Detection Methods		
Detection Method	Advantages	Disadvantages
Block Average (BLKAVG)	<ol style="list-style-type: none"> 1. Simple, intuitive, easy to understand, obvious from plot. 2. Shows non-jump changes. 3. Handles drift better – shows it as staircase of block averages, not jumps. 	<ol style="list-style-type: none"> 1. Don't get exact jump location. 2. Don't get actual frequency averages. 3. May not get accurate jump size.
Sequential Average (SEQAVG)	<ol style="list-style-type: none"> 1. Get right jump location. 2. Get actual frequency averages within jump regimes. 3. Cleaner plot. 4. Better jump size estimates. 5. No offset parameter needed. 6. Window length less critical (if not too long). 	<ol style="list-style-type: none"> 1. More complex algorithm. 2. Doesn't show non-jump changes. 3. Depends on average at beginning of each regime. 4. Shows drift as staircase of jumps. 5. Slower (but not significantly).

• Cumulative Sum Jump Detection

The Cumulative Sum (CUSUM) is a classic change-point analysis technique that uses the cumulative sum of the differences between the current value and the overall average [4]. Starting with $S_0=0$,

$$S_i = S_{i-1} + (y_i - \bar{y}).$$

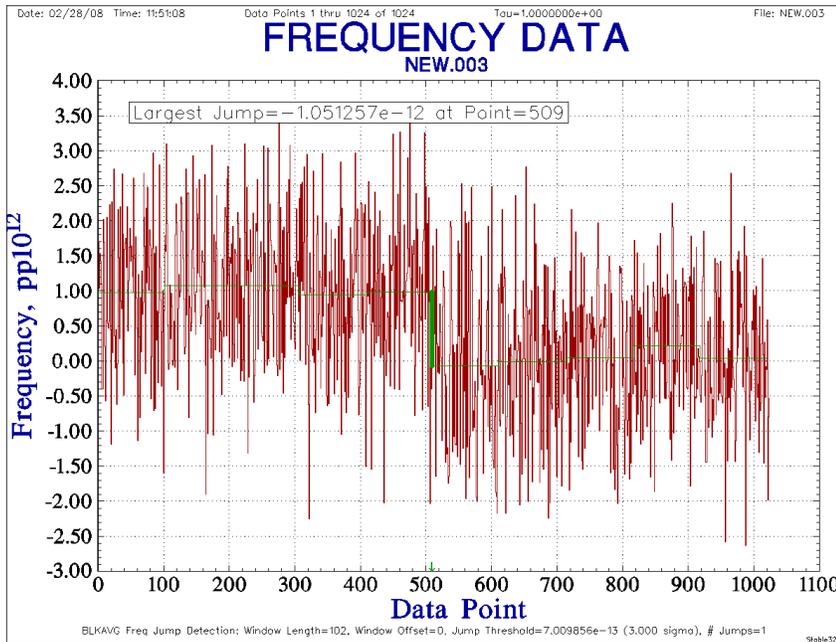
The CUSUM slope indicates the value of the data with respect to the overall average. A flat cumulative sum indicates that the data is near the average value, a straight cumulative sum indicates a period of constant data values, and, most importantly, a sudden change in the CUSUM direction indicates a jump in data. The CUSUM plot for a data set having a single jump will have a V or inverted V shape. The value of the jump can be determined by the sum of the two slopes of the CUSUM curve (see below).

• Other Algorithms

Recent progress in detecting "regime shifts" in ecological records has resulted in another jump detection algorithm called Sequential t-Test Analysis of Regime Shifts (STARS), which could be useful for analyzing clock frequency records [5]. The STARS algorithm uses the Student's t-test [6] to compare the mean values of two analysis windows that are moved sequentially through the data record, searching for differences that are associated with a significant jump. The problem with the STARS method is that it is too sensitive. Frequency stability mainly involves the analysis of noise. The t-test indicates whether, at a certain confidence level, there has been a shift in the mean, generally for small sample sizes. For the analysis of noise, this detects too many changes. The better criterion for noise is either an absolute frequency deviation, or a jump that exceeds the standard or Allan deviation by a certain factor.

• BLKAVG Jump Detection

An example of the Stable32 BLKAVG frequency jump detection function is shown in the plot below for a set of 1024 points of simulated frequency data having white FM noise at a level of 1×10^{-12} at a 1-second sampling time plus a step of -1×10^{-12} at the center of the record. The objective is to correctly detect this frequency jump in the presence of an equal amount of noise, and quantify both the location of the jump and its value, preferably using default detection parameters (an analysis window size based on the record length and a frequency jump threshold based on the Allan deviation of the entire frequency record at an averaging factor equal to the window size). The light horizontal green lines on the plot are the average values of the frequency data over the analysis windows, and the detected frequency jump is shown by the heavy vertical line at the center of the plot. Also shown below is the jump detection information that is automatically written to the clipboard.



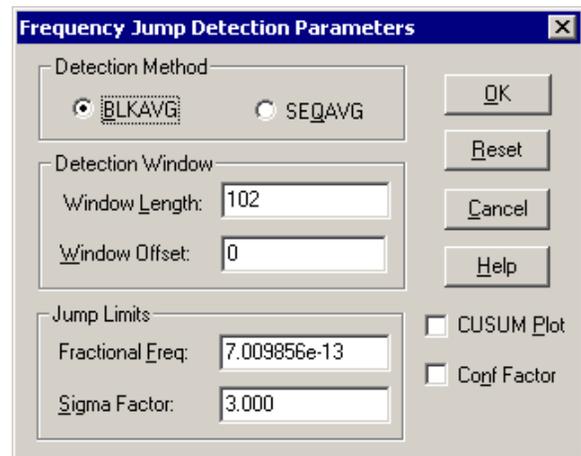
Stable32 Frequency Jump Detection Plot

```

File: NEW.003
BLKAVG Jump Detection Results:
Window Length = 102
Window Offset = 0
Jump Threshold = 7.009856e-13
# Jumps = 1
Largest Jump = 1.051257e-12
Point #      Jump Size
509         1.051257e-12
Scaled Cumulative Sum Results:
Min = -9.303689e-004 at Point # 1024
Max = 2.526259e+002 at Point # 506
Single Jump Estimate:
Jump = -9.879969e-013 at Point # 506
  
```

Jump Detection Results

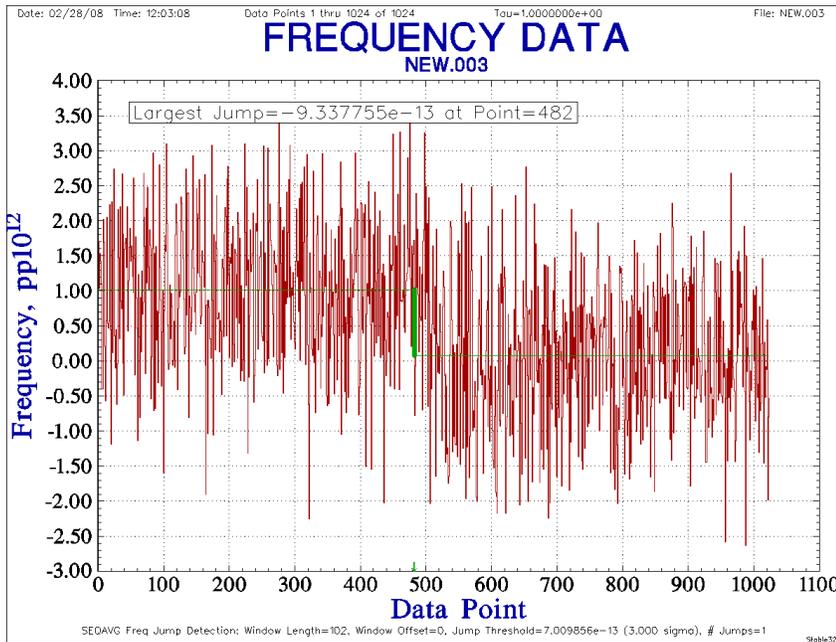
The frequency detection plot shows the average values of the analysis windows and a jump between them are indicated by a heavy vertical line. The default window length is one-tenth of the record length or five, whichever is larger. The default window offset is zero, and it can have a maximum value equal to the number of frequency data points modulo the window length. The default jump threshold is three times the value of the Allan deviation of the entire data set at an averaging factor equal to the window length. These jump detection parameters can be tuned for the particular application in the frequency jump detection parameter configuration dialog box.



Jump Detection Configuration Dialog Box

• SEQAVG Jump Detection

An example of the Stable32 SEQAVG frequency jump detection function for the same data set is shown in the plot below. Again, the objective is to correctly detect this frequency jump in the presence of an equal amount of noise, and quantify both the location of the jump and its value, preferably using default detection parameters (an analysis window size based on the record length and a frequency jump threshold based on the Allan deviation of the entire frequency record at an averaging factor equal to the window size). The light horizontal green lines on the plot are the average values of the frequency data over the entire jump regimes, and the detected frequency jump is shown by the heavy vertical line at the center of the plot. Also shown below is the jump detection information that is automatically written to the clipboard.



Stable32 Frequency Jump Detection Plot

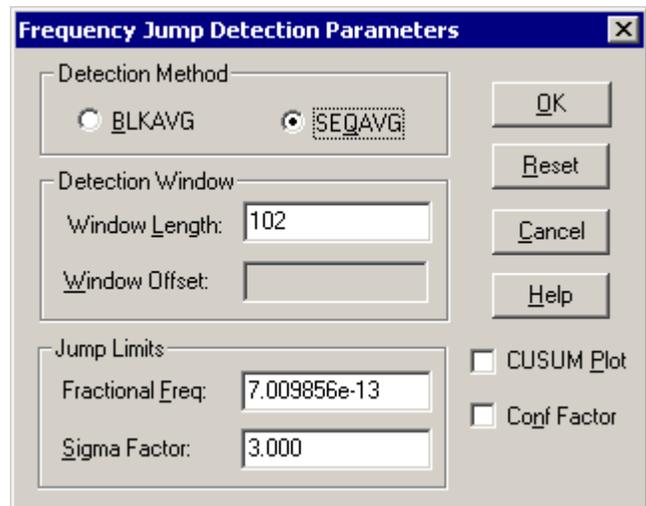
```

File: NEW.003
SEQAVG Jump Detection Results:
Window Length = 102
Jump Threshold = 7.009856e-13
# Jumps = 1
Largest Jump = -9.337755e-13
Point #      Jump Size
482         9.337755e-13
Scaled Cumulative Sum Results:
Min = -9.303689e-004 at Point # 1024
Max = 2.526259e+002 at Point # 506
Single Jump Estimate:
Jump = -9.879969e-013 at Point # 506
  
```

Jump Detection Results

The frequency detection plot shows the average values of the two jump regimes and the jump between them are indicated by a heavy vertical line.

The default window length is one-tenth of the record length or five, whichever is larger. The window offset parameter does not apply to the SEQAVG method. The default jump threshold is three times the value of the Allan deviation of the entire data set at an averaging factor equal to the window length. These jump detection parameters can be tuned for the particular application in the frequency jump detection parameter configuration dialog box. One jump is detected for window lengths between 28 and 220 for these data. Reversing the order of the data changes the reported jump magnitude by about 3.5% and the jump position by 4 points.



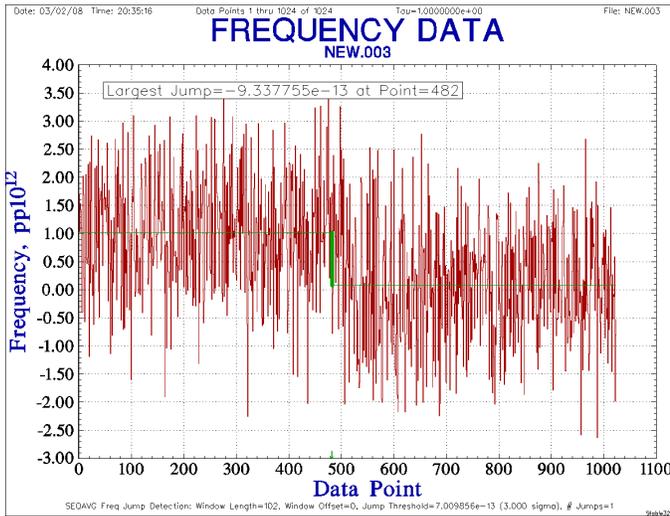
Jump Detection Configuration Dialog Box

• SEQAVG Jump Location

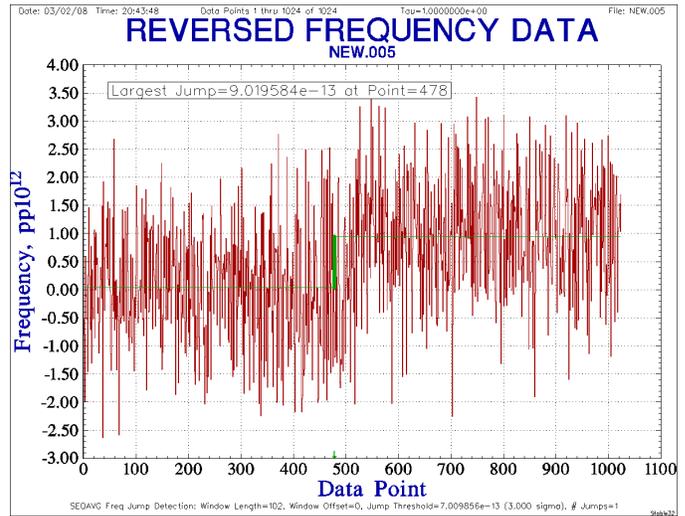
The SEQAVG frequency jump detection algorithm tends to estimate the jump location earlier than it actually is because reports it at the beginning of the averaging window in which the jump is confirmed. That bias can be reduced significantly by averaging the jump location estimates for the normal forward and reversed data sets. That improved jump location estimate is given by:

$$J = \frac{F + (N - R)}{2} \quad \text{where:} \quad \begin{array}{l} J = \text{Estimated Jump Location} \\ F = \text{Forward Jump Location} \\ R = \text{Reverse Jump Location} \\ N = \# \text{ Data Points} \end{array}$$

For example, consider the following forward and reverse SEQAVG jump detection plots for the same 1024-point simulated frequency data set having a jump of -1×10^{-12} at point 512 again using the default jump detection parameters:



Forward Jump Location = 482



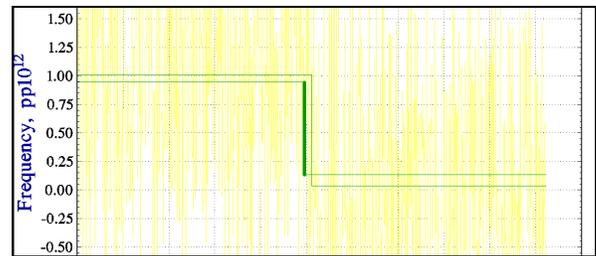
Reverse Jump Location = 478

$$J = (482 + 1024 - 478) / 2 = 514$$

The combined forward/reverse jump location estimate is very close to that expected, much closer than either the forward or reverse estimates alone. The Stable32 Scale function includes a Reverse feature that can be used to reverse the order of the data to perform this calculation.

• SEQAVG Jump Magnitude

The jump magnitudes are similar in each direction, and have an average value of 9.18×10^{-13} , reasonably close to the nominal value of 1×10^{-12} and the actual difference between the two halves of the data, 9.77×10^{-13} , as shown in the expanded forward data plot at the right. Reducing the window size to 50 yields an average jump magnitude of 9.54×10^{-13} , 2.4% below the actual.



Estimated and Actual Jump Magnitudes

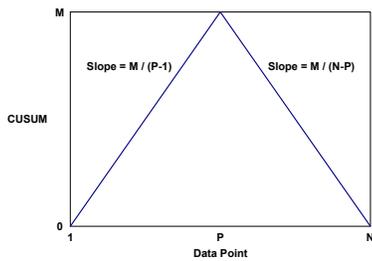
• CUSUM Plot

The Stable32 frequency jump detection function also makes use of a cumulative sum (CUSUM) calculation to help identify and estimate the location and value of frequency jumps. The CUSUM is a classic change-point analysis technique that uses the cumulative sum of the differences between the current value and the overall average [4]. Starting with $S_0=0$,

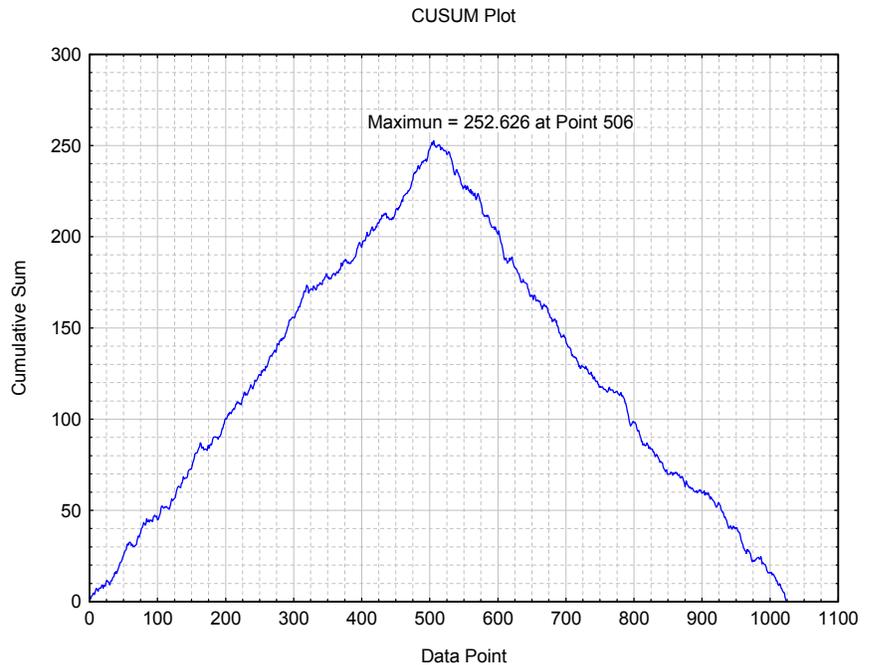
$$S_i = S_{i-1} + (y_i - \bar{y}).$$

The CUSUM slope indicates the value of the data with respect to the overall average. A flat cumulative sum indicates that the data is near the average value, a straight cumulative sum indicates a period of constant data values, and, most importantly, a sudden change in the CUSUM direction indicates a jump in data. The CUSUM plot for a data set having a single jump will have a V or inverted V shape. The value of the jump can be determined by the sum of the two slopes of the CUSUM curve. For N data points having a maximum or minimum cumulative sum value M at point P, the frequency jump is equal to:

$$\frac{M}{P-1} + \frac{M}{N-P}$$



Inspection of the CUSUM plot provides a good indication of the confidence in the jump determination and can show the presence of other smaller, below-threshold jumps. For a single dominant jump, it does the best job of locating and quantifying the jump. For frequency data, the CUSUM is identical to its corresponding phase data after frequency normalization.



CUSUM Plot for File: NEW.003

• Confidence Factor

A particularly nice aspect of the CUSUM method is its ability to provide an estimate of the confidence factor for a detected jump. This can be accomplished by randomly reordering the frequency data and calculating the CUSUM for many such cases, thereby establishing a basis of comparison for the actual jump estimation. For example, if in 995 out of 1000 such cases, the CUSUM range is less than that for the actual data, the confidence in the jump is 99.5%. In most cases for data sizes and thresholds appropriate for clock frequency data, the confidence level is above 99.9% when a jump is detected.

• Jump Detection Criteria

Important considerations are the ability to reliably detect jumps of a specified magnitude with minimal false alarms in the presence of common clock noises and slow frequency trends. Frequency jumps can be the most significant limitation to clock performance in many applications. While there is no substitute for human judgment in assessing clock behavior, automated jump detection can be a valuable supplement to that, especially in a large-scale production situation (e.g. commercial telecom clocks), or for critical applications (e.g. GPS or Galileo satellite clocks).

• Jump Display and Reporting

In Stable32, frequency jump detection is an option associated with a frequency data plot. The frequency averages are shown as horizontal lines extending over the analysis data windows. Those average values are connected by vertical steps that, in the case of a detected frequency jump, are shown as heavy lines. The jump detection parameters are shown as an annotation at the bottom of the plot, the largest jump magnitude and location (if any) is shown as an optional message, and complete information about the jump detection parameters and results is written to the Windows clipboard. A CUSUM sub-plot may also be displayed as an option. The values of the frequency average corresponding to each data point are automatically written to the file Jump.dat after every plot, and the cumulative sum results are written to

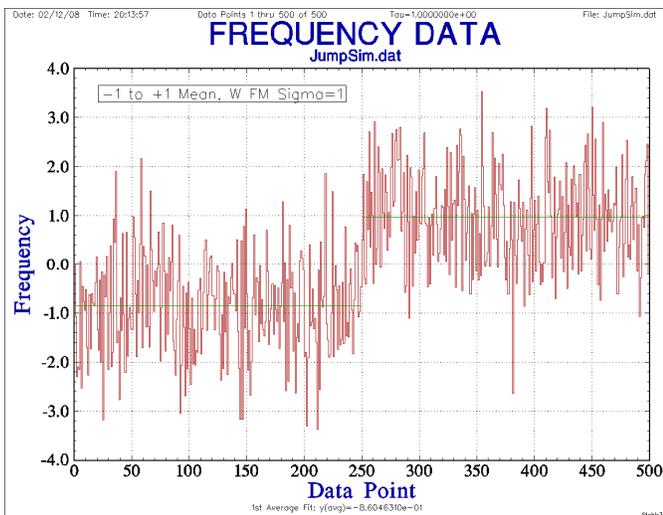
the Cusum.dat file. Note that the color of the frequency data in some of the BLKAVG and SEQAVG plots of this paper has been changed so that the jump analysis can be better seen.

• **Jump Detection Limitations**

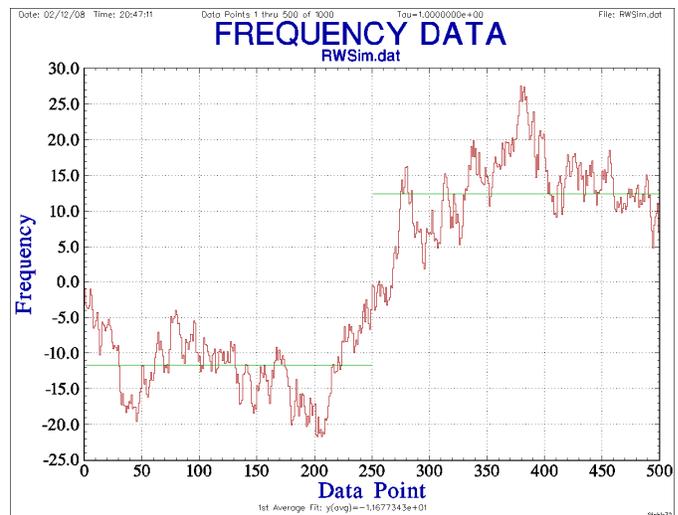
The most significant limitation of the BLKAVG frequency jump detection algorithm is that it uses analysis windows of a fixed (although selectable) size. That means that jumps can be detected only at certain places in the frequency record, and, accordingly, the magnitude of the detected jumps are not necessarily those of the underlying data but simply the change in the average value at the boundary of the two analysis windows. This limitation can be reduced by changing the size of the analysis window and/or their offset. The window size is a tradeoff between better time resolution (short) and better noise averaging (long). In contrast, the SEQAVG and CUSUM methods can detect a jump at any point in the data record. The SEQAVG jump location can be improved by analyzing the data in both directions. All of these jump detection methods are sensitive to drift and divergent noise.

• **Prewhitening**

It is often hard to tell the difference between uncorrected drift, divergent noise and an actual jump. Consider, for example, the following two plots. The one on the left is white FM noise having a sigma of 1 and a jump in mean value from -1 to +1 at the center, while the one on the right is random walk FM noise with the same sigma and no jump.



White Noise with Jump



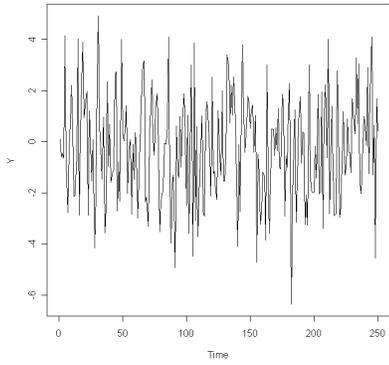
Random Walk Noise w/o Jump

Jump detection can sometimes be improved by removing not only outliers and deterministic drift, but also by removing flicker (“pink”) and random walk (“red”) noise from the data [7]. That prewhitening process can help to distinguish between true jumps and the lurching behavior caused by divergent noise. Such noise removal, although inexact, can be accomplished by determining the lag 1 autocorrelation coefficient, ρ_1 , of the data and using that to remove an AR(1) autoregressive noise component [8, 9]:

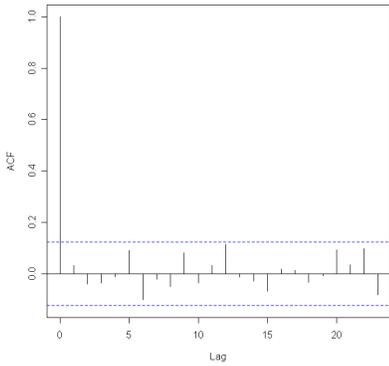
$$y(t) = y(t+1) - \rho_1 \cdot y(t).$$

A ρ_1 value of 0 corresponds to white noise, while increasingly positive ρ_1 values result from pink and red noise spectra, with a value of 1 associated with random walk noise. Examples of AR(1) noise processes and their autocorrelations and spectra are shown in the plots below (see [9] for more examples and discussion). The lurches of the more divergent noise processes can often resemble jumps.

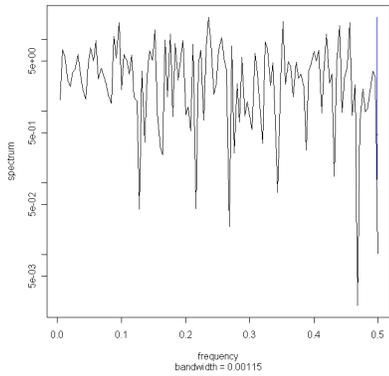
The Stable32 Drift function has provisions for removing AR(1) autoregressive noise from frequency data.



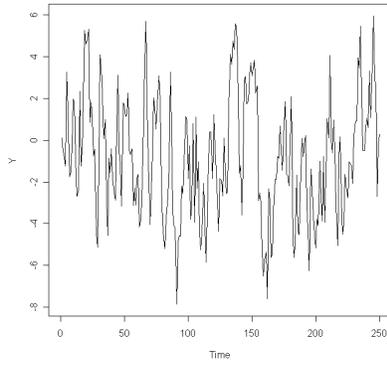
Series Y



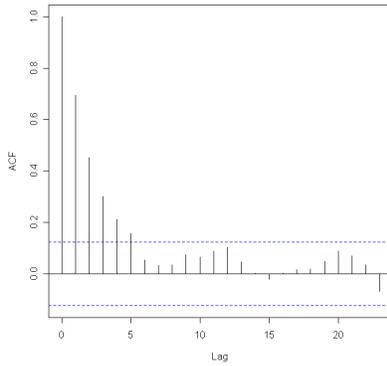
Series: x
Raw Periodogram



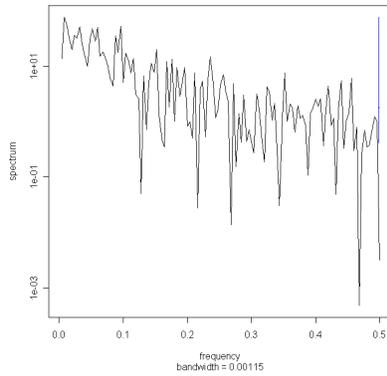
$\rho_1=0.000$ (White)



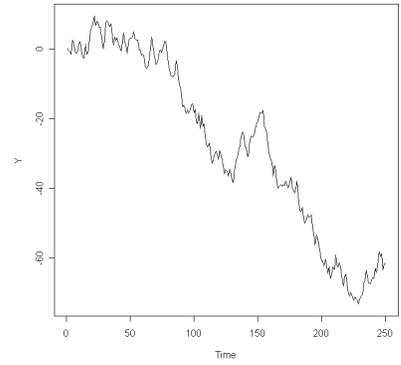
Series Y



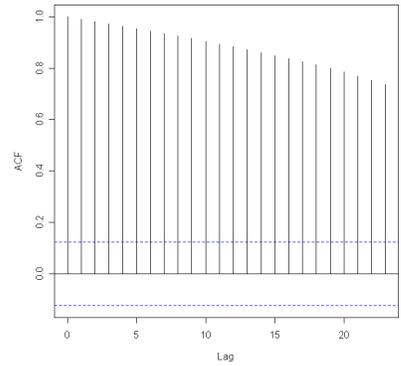
Series: x
Raw Periodogram



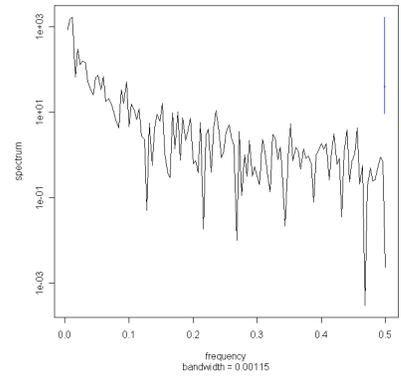
$\rho_1=0.700$ (\approx Flicker Noise)



Series Y



Series: x
Raw Periodogram

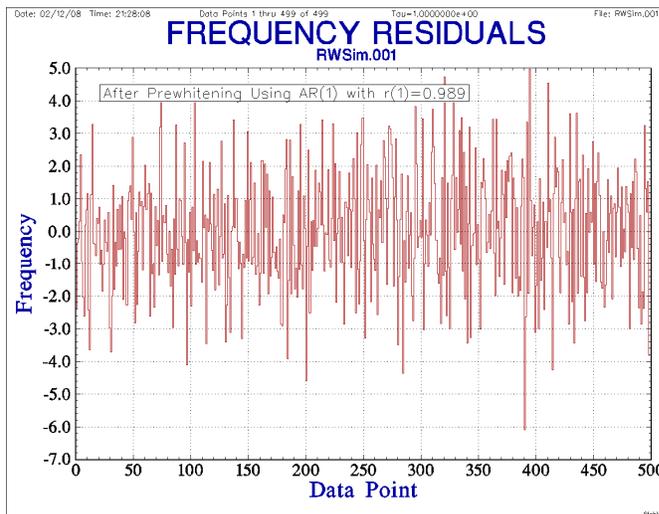


$\rho_1=1.000$ (Random Walk)

Examples of AR(1) Noise Processes: $y(t) = \rho_1 \cdot y(t-1) + \varepsilon(t)$ where $\varepsilon(t)$ is white noise

The sample of random walk noise shown above has a $\rho_1=0.989$ and a power law exponent $\alpha = -1.90$ at an averaging factor of 1. If that noise data is prewhitened as described above, the resulting plot is shown at the right. The residuals are white ($\rho_1 = -0.041$, $\alpha = +0.09$) and are without a jump.

A determination of ρ_1 can itself be affected by a jump, and techniques have been described for sectioning the data and correcting for bias [7]. The prewhitening operation will reduce the size of the jump, but the process is nevertheless reasonably effective in distinguishing between divergent noise and an actual jump, whose location is preserved, as shown in the following example.



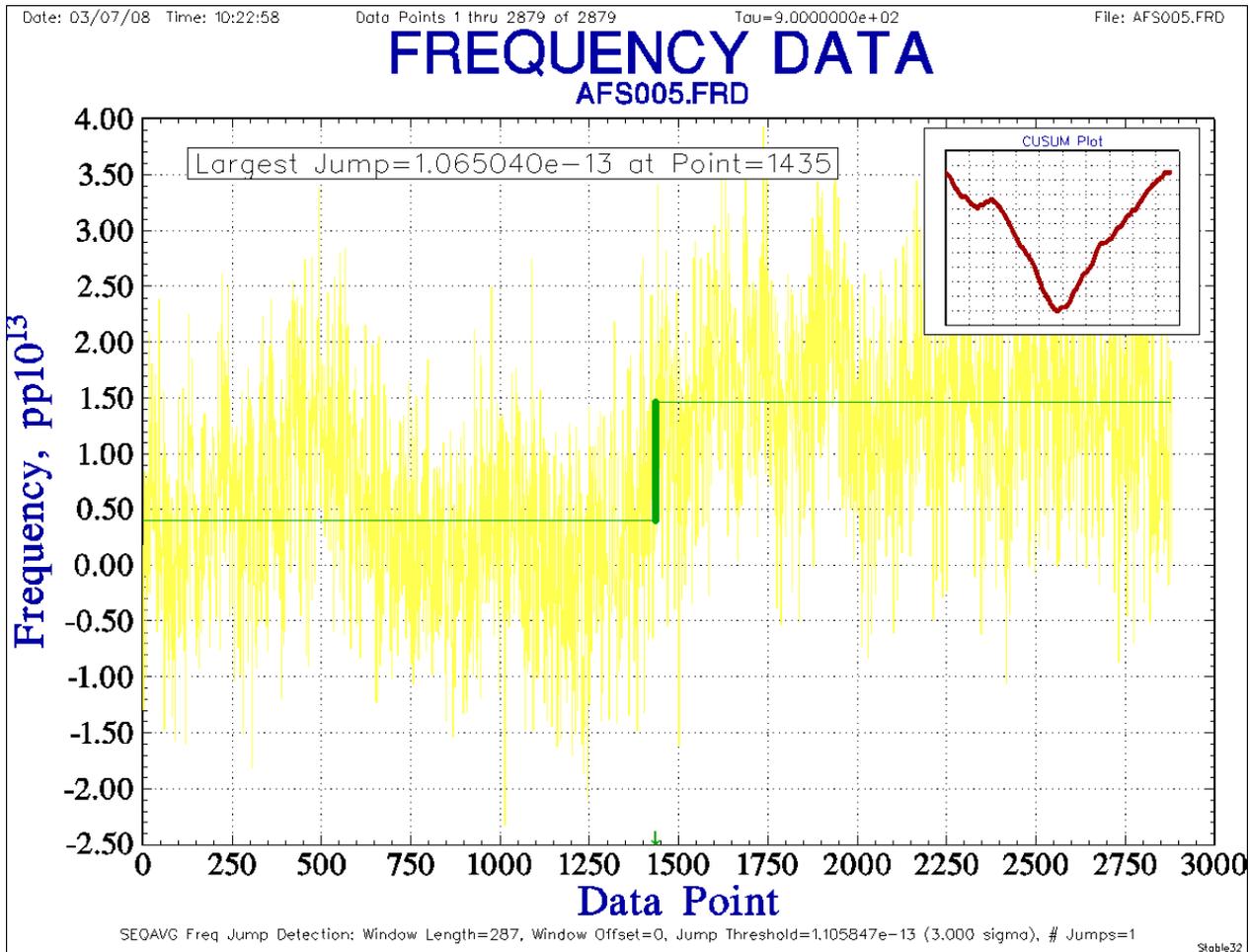
Prewhitened RW FM Noise

• Additional Examples

The previous examples of the Stable32 frequency jump detection methods showed the analysis an abrupt frequency change of 1×10^{-12} at the center of a 1024-point frequency record of simulated white FM noise having an Allan deviation of 1×10^{-12} at the 1-second sampling interval. The jump was correctly identified using the default detection parameters as having a magnitude of 1.05×10^{-12} at point 509. The calculated 1-second overlapping Allan deviation of the entire frequency record, 0.992×10^{-12} , is not significantly affected by the jump, which causes the stability plot to flatten and then go upward at longer averaging times. Varying the window offset over its allowable range of 0 to 24 had little effect on the magnitude of the detected frequency jump, which varied from 0.993 to 1.116×10^{-12} . The frequency jump is visually apparent in the plot, especially if the data are averaged by a factor of about ten, but the detection algorithm is able to quantify the jump automatically at least as well as can be done manually.

◆ Simulated Frequency Data with White and Flicker FM Noise Plus a Jump

Another example of Stable32 SEQAVG frequency jump detection is shown in the plot below, again using the default detection parameters except that this time the Show CUSUM plot box is checked. This is a 30-day set of 2880 points of $\tau = 900$ second (15 minute) simulated clock data having $2 \times 10^{-12} \tau^{-1/2}$ white FM noise and flicker FM noise at a level of 2×10^{-14} plus a jump of $+1 \times 10^{-13}$ at the center (point 1440). Again the jump is detected, located and quantified with reasonable accuracy. The CUSUM analysis characterizes the jump as having a value of $+1.06 \times 10^{-13}$ at Point 1435. The positive frequency “lurch” at Day 5 is not a jump but rather a result of the divergent flicker FM noise. It is visible in both the frequency and CUSUM plots, and its average frequency change is about half the threshold level.



Stable32 Frequency Data Plot with SEQAVG Jump Detection and CUSUM Plot Insert

SEQAVG Jump Detection Results:
 Window Length = 287
 Window Offset = 0
 Jump Threshold = 1.105847e-13
 # Jumps = 1
 Largest Jump = 1.379313e-13
 Point # Jump Size
 1434 1.379313e-13

Scaled Cumulative Sum Results:
 Min = -7.661024e+002 at Point # 1435
 Max = 1.378832e+000 at Point # 2819

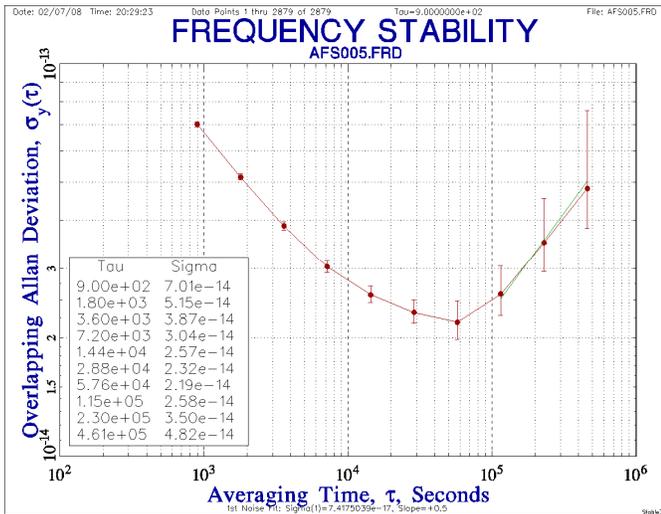
Single Jump Estimate:
 Jump = +1.064789e-013 at Point # 1435
 Confidence Factor > 99.9 %

SEQAVG Frequency Jump Detection Results

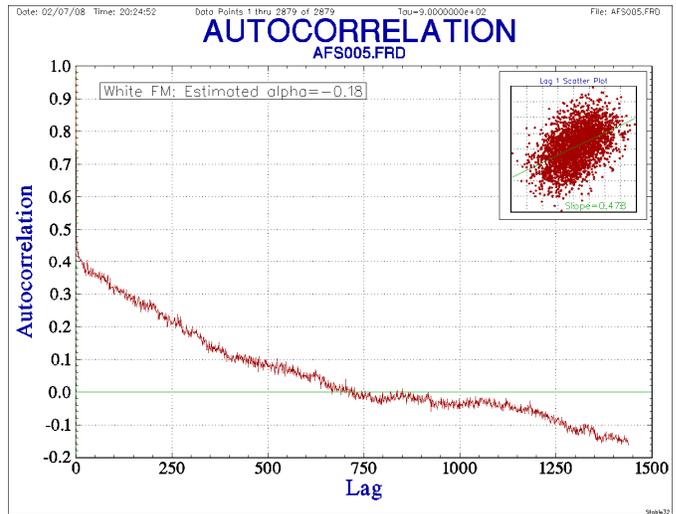
CUSUM Analysis Results

The same simulated noise without the frequency jump has a CUSUM range of 232, about one-third the jump value, and no jumps are found with the same detection parameters. A typical set of 1000 random CUSUMs has a range of 118, a mean of 82 and a standard deviation of 19.3, so the actual CUSUM range of 767 is many standards deviations larger, implying a high confidence in the jump detection.

Other Stable32 analysis functions can also provide insight into the presence and effect of a frequency jump, as shown in the Allan deviation and autocorrelation function plots below:



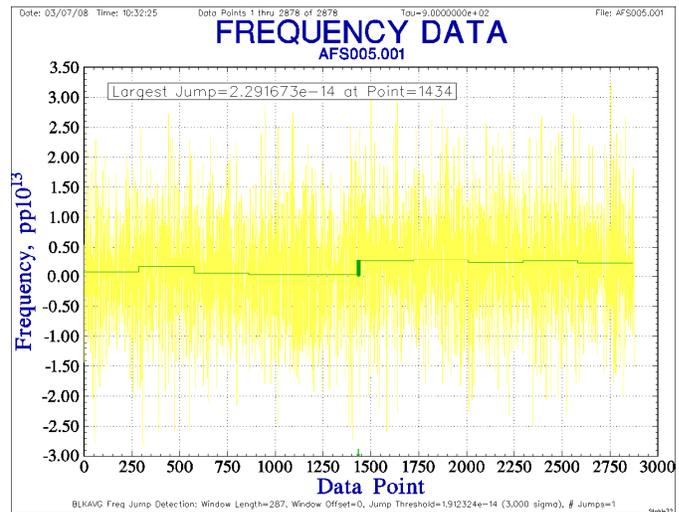
Allan Deviation Plot



Autocorrelation Function Plot

The upward slope at long τ in the stability plot indicates a frequency disturbance of some sort because the expected flicker FM noise would have a flat characteristic. Although usually associated with random walk FM noise, the $\tau^{1/2}$ slope is in this case caused by the step in average frequency. The long positive autocorrelation tail at large lags is a sign of either divergent noise or some sort of long-term frequency change. While both of these plots are inconsistent with the simulated noise, neither of them plots can be uniquely or simply related to the frequency jump.

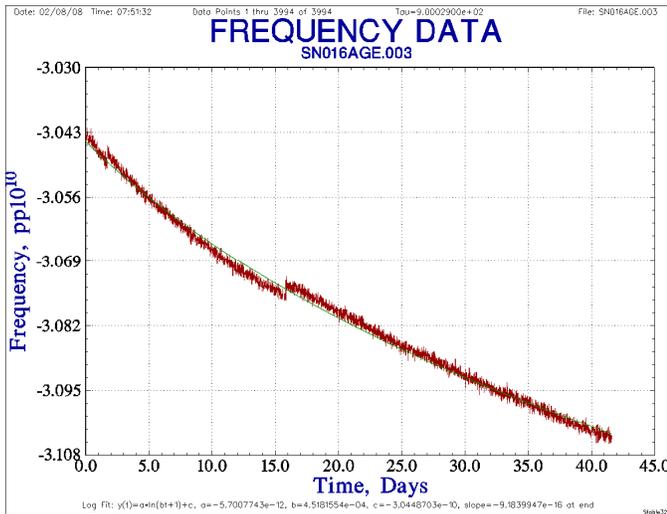
This data set can also be used as an example of the effect of prewhitening. Using an averaging factor of 10, the lag 1 autocorrelation coefficient for these data is 0.827, indicating a mixture of flicker and random walk FM noise. Prewhitening by removing the corresponding AR(1) autoregression process results in the much whiter frequency data plot shown at the right. Even though the jump is greatly reduced in magnitude, it is still detected with a 3-sigma threshold at the same place. This seems to indicate that it is actually a jump and not simply the result of the underlying divergent noise.



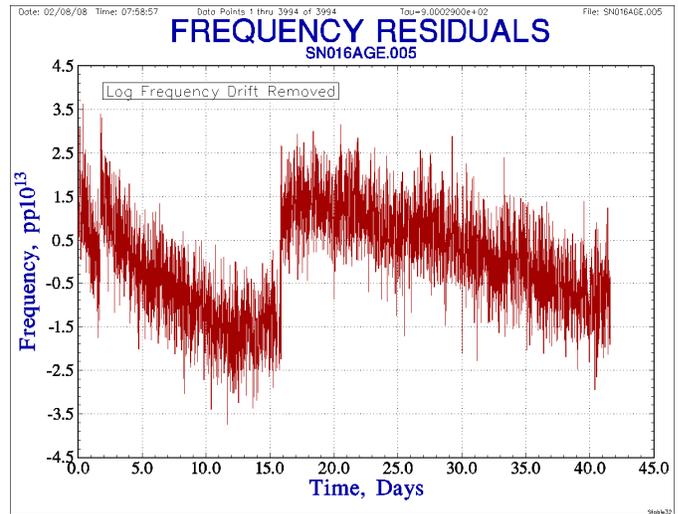
Jump Analysis of Prewhitened Data

◆ Actual Clock Stabilization Data with Jumps

Now let us examine some actual clock data, in this case from the early stabilization of rubidium frequency standard. The following two plots show the 42-day, $\tau = 900$ second frequency record and its residuals after removal of a log fit.

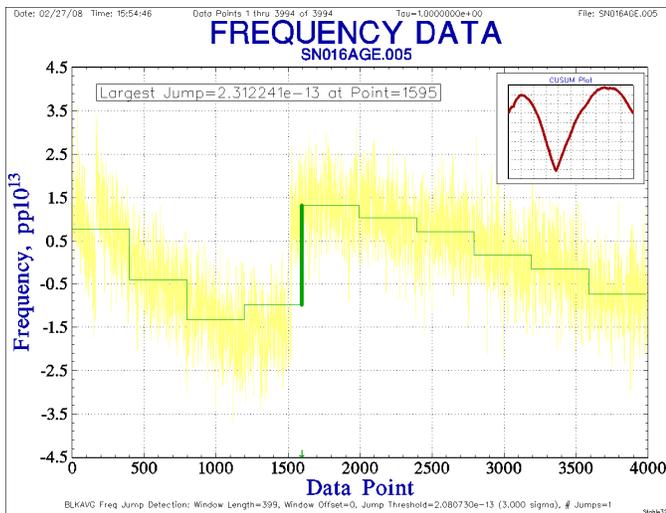


Frequency Data with Log Fit



Frequency Residuals after Log Fit Removed

These records show two frequency jumps, the first during the early rapid stabilization and a second more prominent jump at about Day 16. The next figure shows the results of a default 3-sigma jump analysis for the frequency residuals.

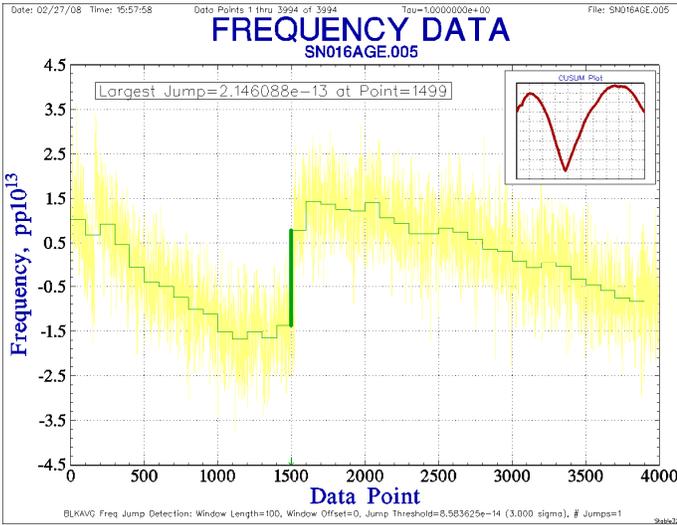


Default BLKAVG Frequency Jump Analysis Plot

File: SN016AGE.005
 BLKAVG Jump Detection Results:
 Window Length = 399
 Window Offset = 0
 Jump Threshold = $2.072076e-13$
 # Jumps = 1
 Largest Jump = $2.312241e-13$
 Point # Jump Size
 1595 $2.312241e-13$
 Scaled Cumulative Sum Results:
 Min = $-9.243454e+002$ at Point # 1524
 Max = $4.174905e+002$ at Point # 3065
 Single Jump Estimate:
 Jump = $+9.814003e-014$ at Point # 1524
 Confidence Factor > 99.9 %

Default BLKAVG Jump Analysis Results

The relatively long analysis window size (399) delays detection of the large jump and the default detection threshold ignores the first smaller jump. The CUSUM sub-plot shows only one sharp change because the first jump is actually more like a transient, and the multiple peaks distort its jump magnitude determination. Changing the window size to 100 yields the following results:



Tailored Frequency Jump Analysis Plot

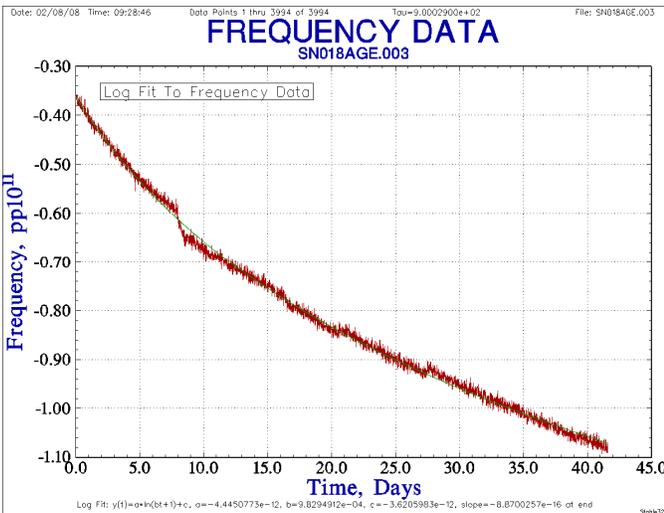
File: SN016AGE.005
 BLKAVG Jump Detection Results:
 Window Length = 100
 Window Offset = 0
 Jump Threshold = 8.366490e-14
 # Jumps = 1
 Largest Jump = 2.146088e-13
 Point # Jump Size
 1499 2.146088e-13
 Scaled Cumulative Sum Results:
 Min = -9.243454e+002 at Point # 1524
 Max = 4.174905e+002 at Point # 3065
 Single Jump Estimate:
 Jump = +9.814003e-014 at Point # 1524
 Confidence Factor > 99.9 %

Tailored Frequency Jump Analysis Results

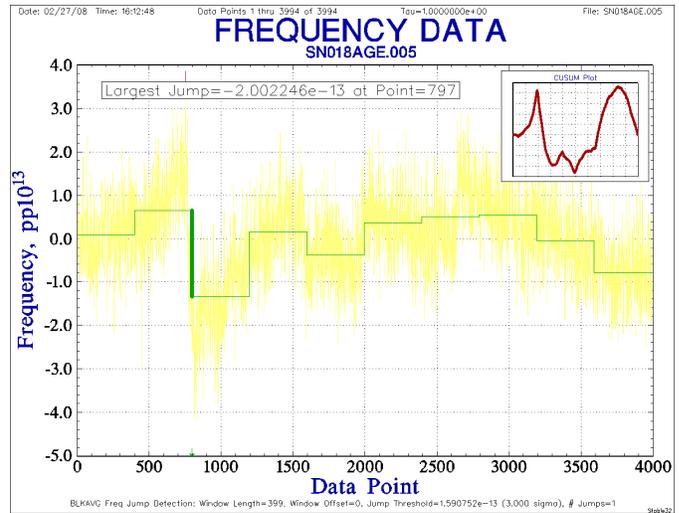
The analysis still detects a single jump, whose location is now determined more exactly. The smaller analysis window size still provides adequate noise averaging. Reducing the threshold to 5×10^{-14} produces 3 detected jumps (including the first one) but splits the second jump into two parts. Clearly, user judgment is needed when performing a jump analysis on a complex record.

◆ More Clock Stabilization Data with Jumps

As a final example, consider the following frequency record and corresponding jump analysis again using the residuals after removing a log fit.

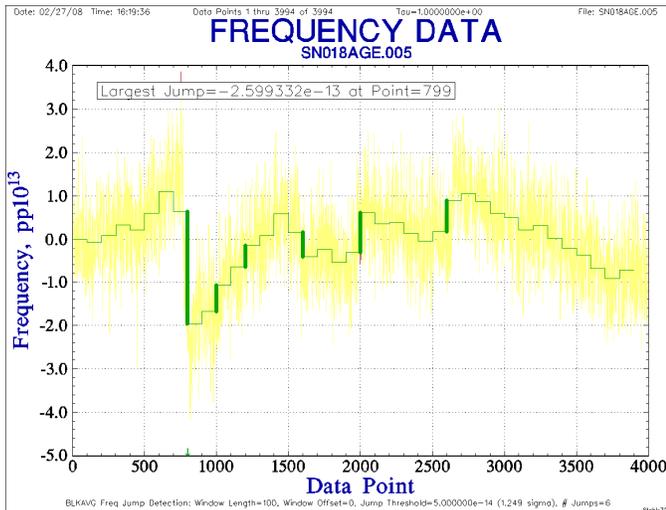


Frequency Data with Log Fit



Jump Analysis of Frequency Residuals

These data show a single large lurch and several smaller jumps. The default 3-sigma moving average jump analysis works well for this slower frequency discontinuity. The CUSUM sub-plot has its sharpest and largest peak at the large lurch, but is too complex to provide a meaningful jump estimate. The multiple smaller jumps can be detected by reducing the window size and threshold as shown below:



Jump Analysis with 5×10^{-14} Jump Threshold

File: SN018AGE.005
 BLKAVG Jump Detection Results:
 Window Length = 100
 Window Offset = 0
 Jump Threshold = 5.000000×10^{-14}
 # Jumps = 6
 Largest Jump = $-2.599332 \times 10^{-13}$
 Point # Jump Size
 799 2.599332×10^{-13}
 999 6.117441×10^{-14}
 1199 5.080865×10^{-14}
 1599 5.682200×10^{-14}
 1999 9.030250×10^{-14}
 2599 7.152933×10^{-14}
 Scaled Cumulative Sum Results:
 Min = -2.925799×10^2 at Point # 1974
 Max = 3.709917×10^2 at Point # 3362

Multiple Jump Analysis Results

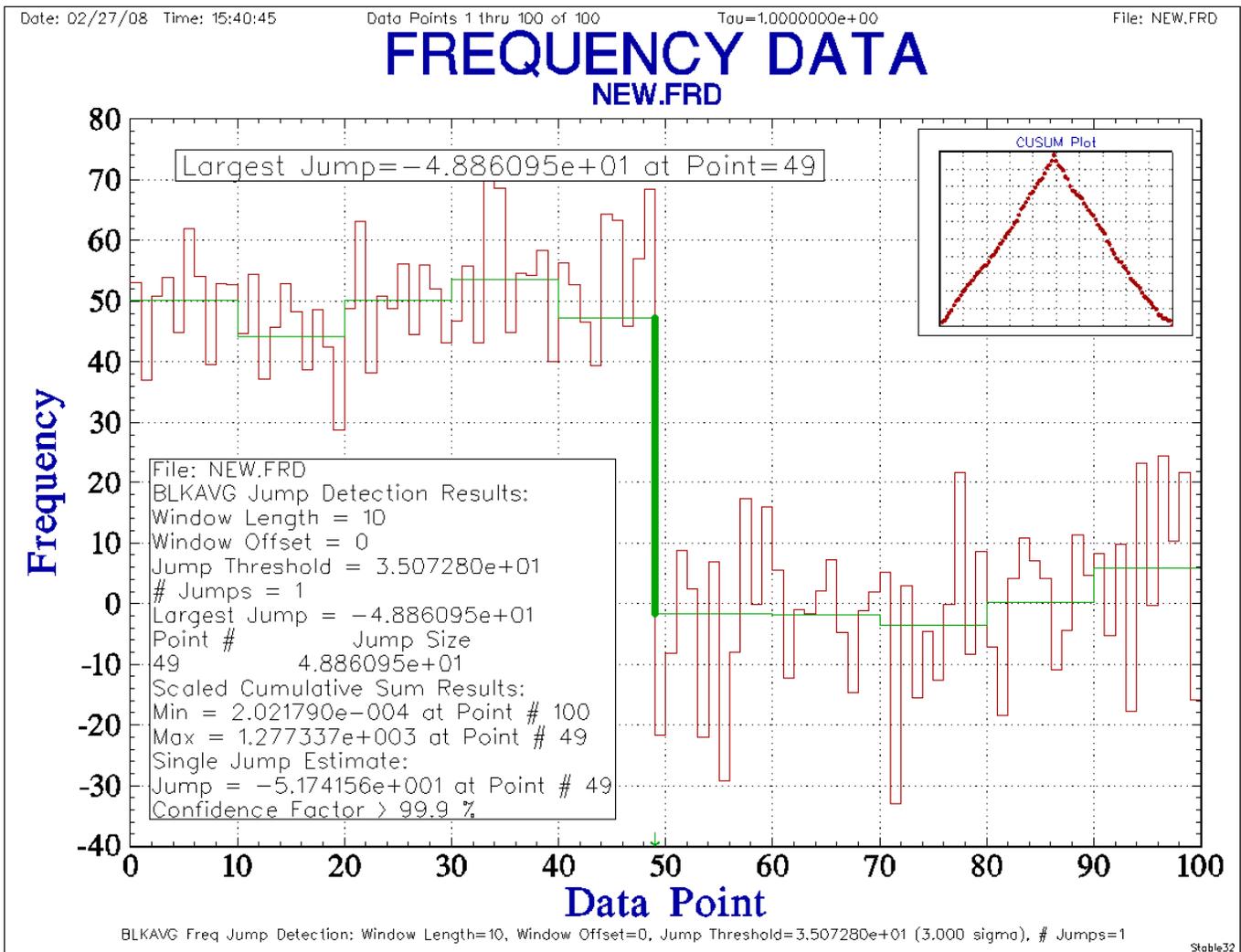
• Conclusions

This paper has described several methods for the detection and analysis of frequency jumps in clock data. The BLKAVG and SEQAVG algorithms, along with a CUSUM plot, are a consistent and effective way to examine such jumps. Those methods have been implemented in the Stable32 program for frequency stability analysis.

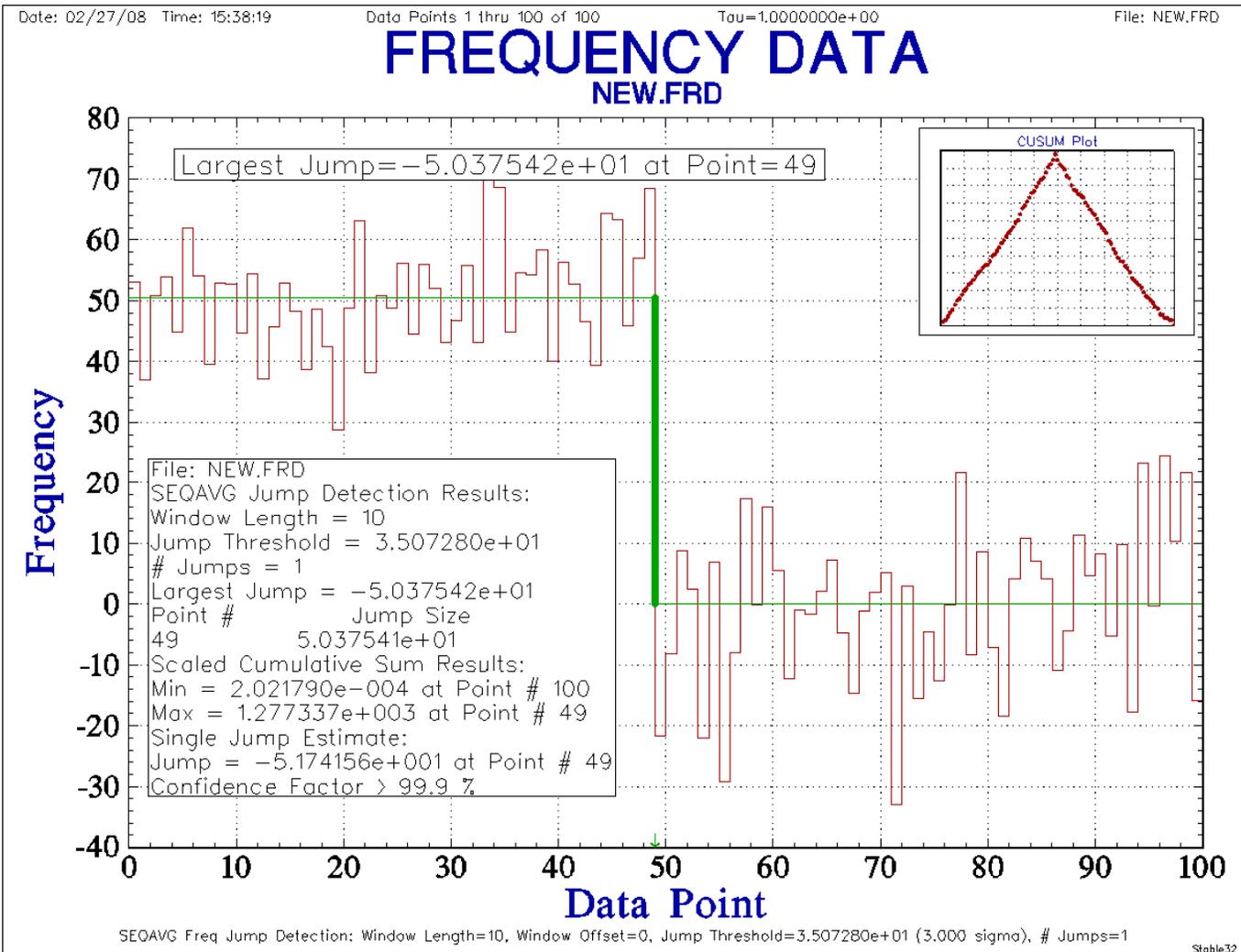
• References

1. [Stable32 Frequency Stability Analysis](#), Hamilton Technical Services, Beaufort, SC 29907.
2. S.N. Rodionov, "[A Brief Overview of the Regime Shift Detection Methods](#)", *Large-Scale Disturbances (Regime Shifts) and Recovery in Aquatic Ecosystems: Challenges for Management Toward Sustainability*, V. Velikova and N. Chipev (Eds.), UNESCO-ROSTE/BAS Workshop on Regime Shifts, 14-16 June 2005, Varna, Bulgaria, 17-24.
3. W.J. Riley, [Handbook of Frequency Stability Analysis](#), Hamilton Technical Services, March 2007.
4. W.A. Taylor, "[Change-Point Analysis: A Powerful New Tool for Detecting Changes](#)", Taylor Enterprises, Inc.
5. S.N. Rodionov, "[A Sequential Algorithm for Testing Climate Regime Shifts](#)", *Geophysical Research Letters*, Vol. 31, L09204, 2004.
6. "[Student's t-Test](#)", *Wikipedia*.
7. S.N. Rodionov, "[The Problem of Red Noise in Climate Regime Shift Detection](#)", *Geophysical Research Letters*, Vol. 31, L12707, 2006.
8. "[Autoregressive Moving Average Model](#)", *Wikipedia*.
9. E.J. Wagenmakers, S. Farrell and R. Ratcliff, "[Estimation and Interpretation of \$1/f^{\alpha}\$ Noise in Human Cognition](#)", *Psychonomic Bulletin & Review*, Volume 11, Number 4, August 2004, pp. 579-615 (37).
10. R.S. Tsay, "[Outliers, Level Shifts and Variance Changes in Time Series](#)", *Journal of Forecasting*, Vol. 7, 1-20, 1988.
11. M. Basseville and I.V. Nikiforov, [Detection of Abrupt Changes-Theory and Application](#), Prentice-Hall, ISBN 0-13-126780-9, April 1993.

Appendix I - Stable32 Version 1.53 Frequency Jump Analysis



This plot is an example of the Stable32 Version 1.53 Block Average (BLKAVG) frequency jump detection method that can be invoked from the Fit Type list box of the Frequency Plot Options function. It shows the results of a simulated frequency jump analysis for a relatively short frequency record. The red points on the main plot are the frequency data values, shown as boxes that denote the measurement interval. The light green horizontal lines are the frequency averages over the default moving average jump analysis windows of length 10, one-tenth of the record. The heavy vertical line at the center denotes the detected jump whose value of about -48.9 at point 49 exceeds the default 3-sigma jump threshold of 35.1. The cumulative sum (CUSUM) for these data is shown in the inset plot, whose sharp slope change is associated with the frequency jump. The inverted-V CUSUM plot is used to calculate another estimate of the frequency jump, about -51.7 at point 49. The actual value of the simulated frequency jump was 50. 1000 randomly-shuffled CUSUM ranges are used to obtain the > 99.9 % confidence factor for the frequency jump. The plot annotation at the bottom shows the analysis parameters.



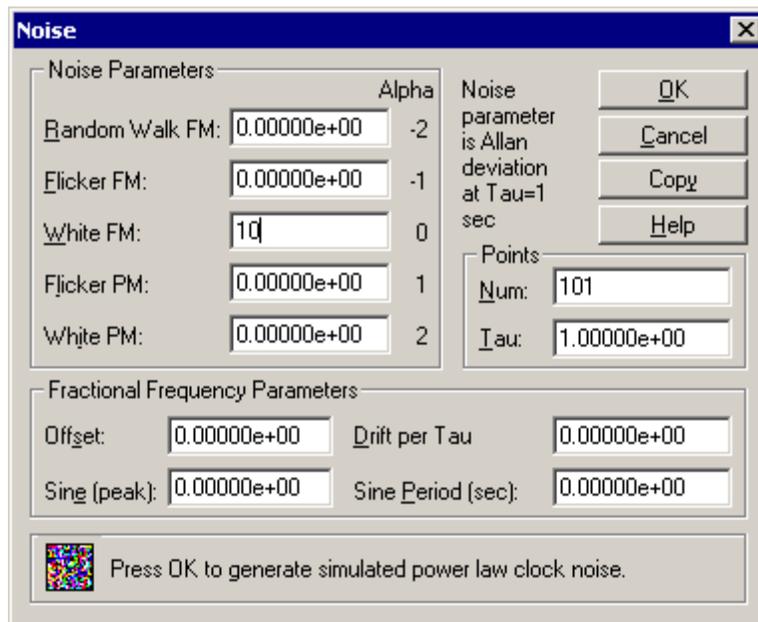
This plot is an example of the Stable32 Version 1.53 Sequential Average (SEQAVG) frequency jump detection method for the same frequency record. The red points on the main plot are the frequency data values, shown as boxes that denote the measurement interval. The light green horizontal lines are the frequency averages over the two jump regimes. The heavy vertical line at the center denotes the detected jump whose value of about -50.4 at point 49 exceeds the default 3-sigma jump threshold of 35.1. The same CUSUM results are shown in the inset plot. Data files Jump.dat and Cusum.dat are automatically written for the frequency averages and CUSUM values respectively, with frequency jumps indicated by zero values in the former. The note inset, copied from the Windows clipboard, shows the jump analysis results. The plot also contains the usual date, time, data points, tau, filename and organization markings, as well as a green arrow denoting the location of the largest jump. The jump detection options include the use of an alternate color for the frequency data to make the jump analysis lines more visible.

Appendix II – Generating Simulated Frequency Jump Data in Stable32

It is easy to create simulated frequency jump data in Stable32 using the Noise function to generate a set of frequency data of the desired size, sampling interval, drift and noise properties, and then using the Scale function to add a frequency offset to a portion of the data.

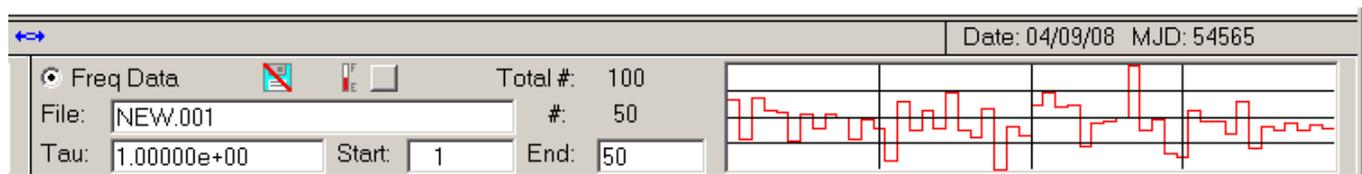
For example, to generate a set of frequency data points similar to those shown in the plots above:

1. Open the Noise function, enter these parameters, and press OK.

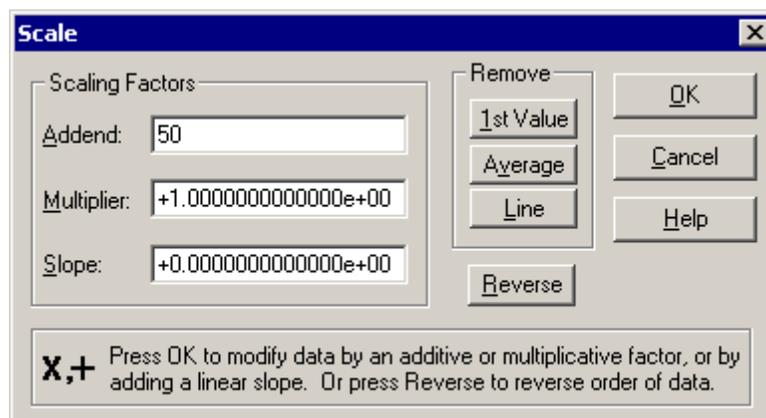


This will generate 100 frequency data points with a sampling interval of 1 second and having white FM noise with a nominal sigma of 10. These data will appear in the status plot at the bottom of the screen, and may be examined with the Stats function or plotted with the Plot function.

2. Set the frequency data analysis end limit to 50 in its status bar at the bottom of the screen.

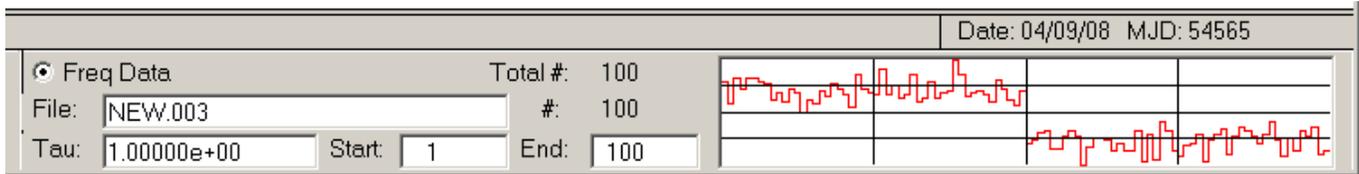


3. Open the Scale function, enter an addend of 50, and press OK.



This will add 50 to the first half of the frequency data, thus introducing a simulated negative frequency jump of that magnitude.

4. Restore the full frequency analysis limits by pressing the small button in the freq data status bar.



The desired set of simulated frequency jump data has now been created.

5. Plot the simulated frequency jump data using the SEQAVG frequency jump detection and analysis method with its default parameters.

