SANDIA REPORT
SAND2011-8265
Unlimited Release
Printed October, 2011

Component Evaluation Testing and Analysis Algorithms

B. John Merchant, Darren M. Hart

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2011-8265
Unlimited Release
Printed October, 2011

Component Evaluation Testing and Analysis Algorithms

B. John Merchant
Darren M. Hart
Dept. 5736, Ground-Based Monitoring R&E
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0404

Abstract

The Ground-Based Monitoring R&E Component Evaluation project performs testing on the hardware components that make up Seismic and Infrasound monitoring systems. The majority of the testing is focused on the Digital Waveform Recorder (DWR), Seismic Sensor, and Infrasound Sensor. In order to guarantee consistency, traceability, and visibility into the results of the testing process, it is necessary to document the test and analysis procedures that are in place. Other reports document the testing procedures that are in place (Kromer, 2007). This document serves to provide a comprehensive overview of the analysis and the algorithms that are applied to the Component Evaluation testing. A brief summary of each test is included to provide the context for the analysis that is to be performed.

ACKNOWLEDGMENTS

This document builds upon previous descriptions of Component Evaluation testing (Kromer, 2007), a previous software tool user's manual, and discussions with individuals involved in the Component Evaluation testing program. Special thanks for their input go out to Mark Harris, Dick Kromer, Tim McDonald, and Megan Slinkard.

CONTENTS

1 Introduction 9

2 Algorithms 11
2.1 Complex Numbers 11
2.1.1 Complex Operations 11
2.2 Discrete Fourier Transform 13
2.3 Windows 14
2.3.1 Rectangular 14
2.3.2 Hann 15
2.3.3 Hamming 15
2.3.4 Kaiser Bessel 16
2.3.5 Bartlett 17
2.4 Power Spectral Density 18
2.4.1 PSD Confidence 20
2.4.2 RMS Estimates 23
2.4.3 Dynamic Range Estimation 24
2.5 Response 27
2.5.1 Pole Zero 27
2.5.2 Frequency-Amplitude-Phase 33
2.6 Filters 37
2.6.1 Analog Filter Design 37
2.6.2 Frequency Warping 43
2.7 Coherence 46
2.7.1 Two Channel Coherence 46
2.7.2 Three Channel Coherence 51
2.7.3 N Channel Coherence 52
2.8 Sine Fit 53
2.8.1 Three Parameter Sine Fit 53
2.8.2 Four Parameter Sine Fit 54
2.9 Interpolation 57
2.9.1 Linear Interpolation 57
2.9.2 Whittaker Interpolation 57
2.10 Time Tag Measurements 61

3 Test Analysis 73
3.1 Generic Component Tests 73
3.1.3 Power Consumption 74
3.2 Digital Waveform Recorder Tests 77
3.2.3 AC Accuracy 77
3.2.4 AC Clip 79
3.2.5 Analog Bandwidth 81
3.2.6 Common Mode Rejection Ratio 83
3.2.7 Crosstalk 85
3.2.8 DC Accuracy 87
3.2.9 Infrasound System Noise & Bandwidth Limited Dynamic Range 90

3.2.10 Input Impedance ... 91
3.2.11 Input Shorted Offset ... 92
3.2.12 Input Terminated Noise and Maximum Potential Dynamic Range ... 93
3.2.13 Modified Noise Power Ratio ... 95
3.2.14 Relative Transfer Function ... 97
3.2.15 Seismic System Noise & Bandwidth Limited Dynamic Range ... 99
3.2.16 Sine Calibrator Amplitude ... 100
3.2.17 Sine Calibrator Current Amplitude ... 101
3.2.18 Sine Calibrator Frequency ... 103
3.2.19 Sine Calibrator THD ... 104
3.2.20 Time Tag Accuracy ... 105
3.2.21 Time Tag Drift ... 106
3.2.22 Total Harmonic Distortion ... 108
3.3 Seismic Sensor Tests ... 112
3.3.1 Seismic Sensor Self-Noise ... 112
3.3.2 Seismic Sensor Response Verification ... 114
3.4 Infrasound Sensor Tests ... 116
3.4.1 Infrasound Sensor Frequency Response ... 116
3.4.2 Infrasound Sensor Amplitude Response ... 119
3.4.3 Infrasound Sensor Response Verification ... 122
3.4.4 Infrasound Sensor Self Noise ... 124
3.4.5 Infrasound Sensor Isolation Noise ... 126

Appendix ... 129
Window Comparison ... 129
PSD Confidence ... 130
Filters ... 132

References ... 133

Distribution ... 135

NOMENCLATURE

dB	Decibel
DOE	Department of Energy
DFT	Discrete Fourier Transform
DUT	Device Under Test
DWR	Digital Waveform Recorder
FFT	Fast Fourier Transform
GNEMRE	Ground-based Nuclear Explosion Monitoring Research & Engineering (Program within NA22)
NA22	Office of Non-proliferation Research & Development (Office within NNSA)
NNSA	National Nuclear Security Administration (Office within DOE)
PDF	Probability Density Function
PSD	Power Spectral Density
RMS	Root Mean Square
SNL	Sandia National Laboratories
Tester	The individual responsible for performing a given test and the associated analysis

1 INTRODUCTION

The purpose of this document is to capture the algorithms employed by the Component Evaluation project for processing and analyzing test data. These algorithms are documented in sufficient detail so that they can be unambiguously reproduced. Where appropriate, MATLAB® scripts are provided as examples.

2 ALGORITHMS

A description of the core algorithms that are commonly applied to the test analysis modules are provided here. These algorithms form the basis around which the testing is performed.

2.1 Complex Numbers

Complex numbers and their operations are used extensively throughout the algorithms described in this document. Complex operations form the basis of much of signal processing and descriptions of their use are easily found in many available texts. Their description is included simply as an easy reference.

Complex numbers have both a real and an imaginary component. In their Cartesian form, they can be expressed as:

$$x + j * y$$

Where j is defined to be the imaginary unit:

$$j^2 = -1$$
$$j = \sqrt{-1}$$

In their Polar form, complex numbers can be expressed as:

$$r * e^{j\theta}$$

Complex numbers can be converted between their Cartesian and Polar forms using the following relationships:

$$r = \sqrt{x^2 + y^2}$$
$$\theta = \tan^{-1}(y, x)$$

$$x = r * \cos(\theta)$$
$$y = r * \sin(\theta)$$

2.1.1 Complex Operations

Given two complex numbers, defined as:

$$z_1 = x_1 + j * y_1 = r_1 * e^{j\theta_1}$$
$$z_2 = x_2 + j * y_2 = r_2 * e^{j\theta_2}$$

The following operations may be performed:

	Operation	Cartesian	Polar
Addition	$z_1 + z_2$	$(x_1 + x_2) + j * (y_1 + y_2)$	N/A
Subtraction	$z_1 - z_2$	$(x_1 - x_2) + j * (y_1 - y_2)$	N/A

Multiplication	$z_1 * z_2$	$(x_1 * x_2 - y_1 * y_2) + j * (x_1 * y_2 + x_2 * y_1)$	$r_1 * r_2 * e^{j(\theta_1+\theta_2)}$
Division	$\frac{z_1}{z_2}$	$\left(\frac{x_1 * x_2 + y_1 * y_2}{x_2^2 + y_2^2}\right) + j * \left(\frac{y_1 * x_2 - x_1 * y_2}{x_2^2 + y_2^2}\right)$	$\frac{r_1}{r_2} * e^{j(\theta_1-\theta_2)}$
Square Root	$\sqrt{z_1}$	$\sqrt{\frac{r_1 + x_1}{2}} + j * \sqrt{\frac{r_1 - x_1}{2}}$	$\sqrt{r_1} * e^{j\theta_1/2}$
Inverse	$\frac{1}{z_1}$	$\left(\frac{x_1}{x_1^2 + y_1^2}\right) - j * \left(\frac{y_1}{x_1^2 + y_1^2}\right)$	$\frac{1}{r_1} * e^{-j\theta_1}$
Negation	$-z_1$	$-x_1 - j * y_1$	$-r_1 * e^{j\theta_1}$
Complex Conjugate	z_1^*	$x_1 - j * y_1$	$r_1 * e^{-j\theta_1}$
Exponent	e^{z_1}	$e^{x_1} * \cos(y_1) + j * e^{x_1} * \sin(y_1)$	N/A
Logarithm	$\log(z_1)$	$\log(r_1) - j\theta_1$	N/A
Power	$z_1^{z_2}$	$e^{z_2 * \log(z_1)}$	N/A

Often, certain mathematical operations may be simpler to perform in either Cartesian or Polar form. If that is the case, then the most straight forward option may be to transform to the other form, perform the operation, and then transform back.

2.2 Discrete Fourier Transform

Given a discrete time series shown below:

$$x[n], 0 \leq n \leq N-1$$

It is possible to estimate the frequency content of the time series using the Discrete Fourier Transform (DFT) (Oppenheim, 1999, Pg. 542-543):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, 0 \leq k \leq N-1$$

The Fourier Transform, $X[k]$, is a complex sequence giving magnitude and phase offsets for each component of the signal at frequency element k.

The inverse transform is thus:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}, 0 \leq n \leq N-1$$

In general, both $x[n]$ and its Fourier Transform $X[k]$ are sequences of complex values containing both real and imaginary components. However, in practice for this application, $x[n]$ is limited to purely real sequences. If $x[n]$ is real, then its Fourier Transform is known to be symmetric (Oppenheim, 1999, Pg. 576):

$$X[k] = X^*[(-k)_N], \qquad where\ (n)_N\ denotes\ a\ modulo\ by\ N$$

The Fast Fourier Transform (FFT) is used to compute the DFT and its inverse using a Cooley-Tukey decimation-in-time radix-2 FFT algorithm (Oppenheim, 1999, Pg. 635-6). This method of computing the DFT is limited to sequences that have a length that is a power of 2. However, this limitation is not restrictive for this application since the time series being analyzed are typically divided into windows that have been chosen to be a power of 2 in length.

2.3 Windows

A number of windowing functions are employed in the test analysis. The windows serve to segment a section of the waveform time series. The windowed signal comes from multiplying the signal with the window. In addition, the windows generally taper the ends of the time segment so as to minimize spectral aliasing. The specific type of taper influences the resulting frequency content of the windowed signal.

The effect of a window in the frequency domain is the convolution of the frequency response of the window with the frequency response of the signal under study. The simplest case of a rectangular window results in considerable spectral leakage. An ideal window would consist of only an impulse in the frequency domain, as the convolution of an arbitrary signal with an impulse is simply itself. That is unattainable, as it would require a window of infinite length in the time domain, defeating the purpose of a window. However, windows can be shaped with that ideal in mind.

Assume each window is of length N. Along with an equation for the computation of the window, an example plot of the window function and its frequency response are shown below the equation for N=128. A Matlab® script for generating the window and response plots is provided (see 0 Window Comparison).

2.3.1 Rectangular

The rectangular window (Oppenheim, 1999, Pg 468) is defined as:

$$w[n] = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}$$

```
window_comparison(rectwin(128), 'Rectangular');
```

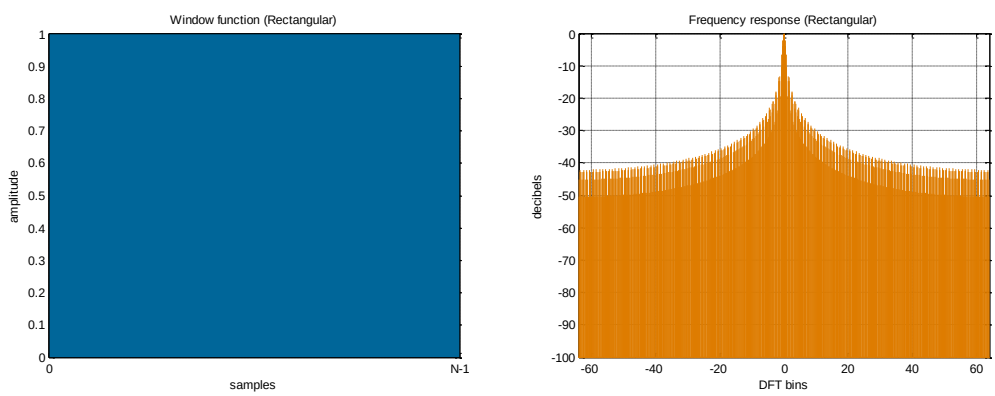

Figure 1 Rectangular Window

The frequency response of the rectangular window has side lobes with high amplitude. This leads to spectral smearing when this window is used to examine the frequency response of a signal under study.

2.3.2 Hann

The Hann window (Oppenheim, 1999, Pg 468), also known as a Hanning window, is defined as:

$$w[n] = \begin{cases} 0.5 - 0.5\ \cos\left(\frac{2\pi n}{N-1}\right), & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}$$

```
window_comparison(hann(128), 'Hann');
```

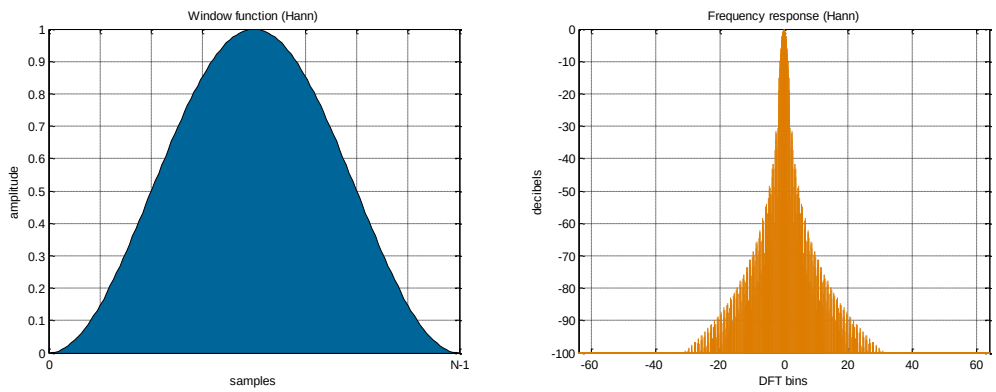

Figure 2 Hann Window

The Hann window has rapidly decaying side lobes. However, its main lobe is still fairly broad

2.3.3 Hamming

The Hamming window (Oppenheim, 1999, Pg 468) is defined as:

$$w[n] = \begin{cases} 0.54 - 0.46\ \cos\left(\frac{2\pi n}{N-1}\right), & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}$$

```
window_comparison(hamming(128), 'Hamming');
```

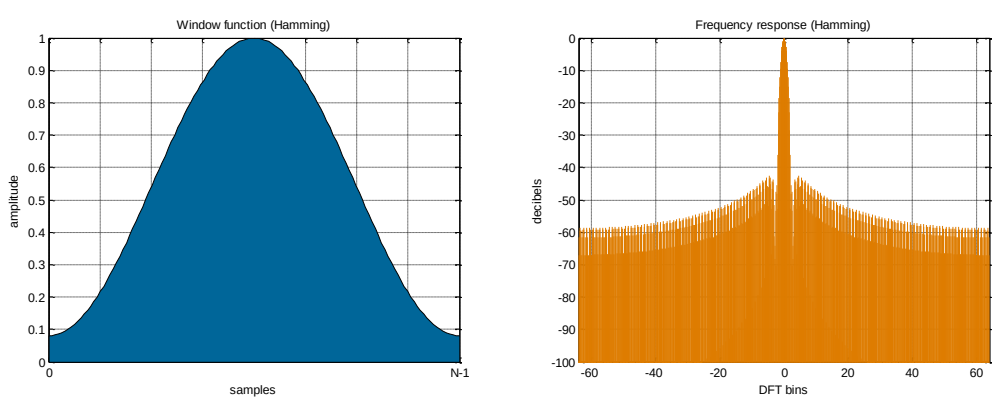

Figure 3 Hamming Window

The Hamming window has a narrower main lobe and fairly low amplitude side lobes.

2.3.4 Kaiser Bessel

The Kaiser Bessel window (Oppenheim, 1999, Pg 474) is defined as:

$$w[n] = \begin{cases} \dfrac{I_0\left(\beta\sqrt{1-\left[\dfrac{n-\alpha}{\alpha}\right]^2}\right)}{I_0(\beta)}, & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}$$

where:

$$I_0 \; is \; the \; zeroth \; order \; modified \; Bessel \; function$$

$$\alpha = \frac{N-1}{2}$$

$$A = -20 \, log_{10}\delta$$

$$\beta = \begin{cases} 0.1102\,(A-8.7), & A > 50 \\ 0.5842\,(A-21)^{0.4} + 0.07886\,(A-21), & 21 \le A \le 50 \\ 0, & A < 21 \end{cases}$$

For the purposes of this application, we have chosen the following peak approximation error:

$$\delta = 10^{-10}$$

This approximation error value results in an A and β of:

$$A = 200$$
$$\beta = 21.0813$$

```
window_comparison(kaiser(128, 21.0813), 'Kaiser Bessel');
```

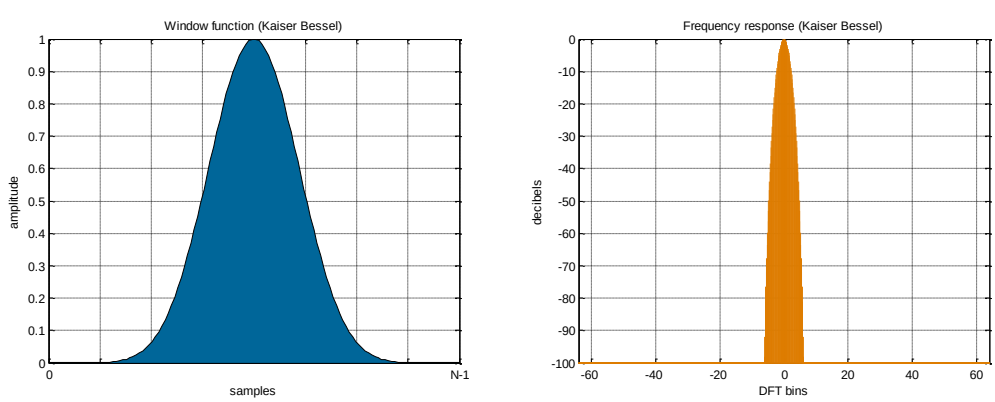

Figure 4 Kaiser Bessel Window

The Kaiser Bessel window has a slightly broader main lobe than the Hamming window. However, it has minimal side lobes.

2.3.5 Bartlett

The Bartlett window (Oppenheim, 1999, Pg 468), also known as a triangular window, is defined as:

$$w[n] = \begin{cases} 2n/(N-1), & 0 \le n \le (N-1)/2 \\ 2 - 2n/(N-1), & (N-1)/2 < n \le N-1 \\ 0, & otherwise \end{cases}$$

```
window_comparison(bartlett(128), 'Bartlett');
```

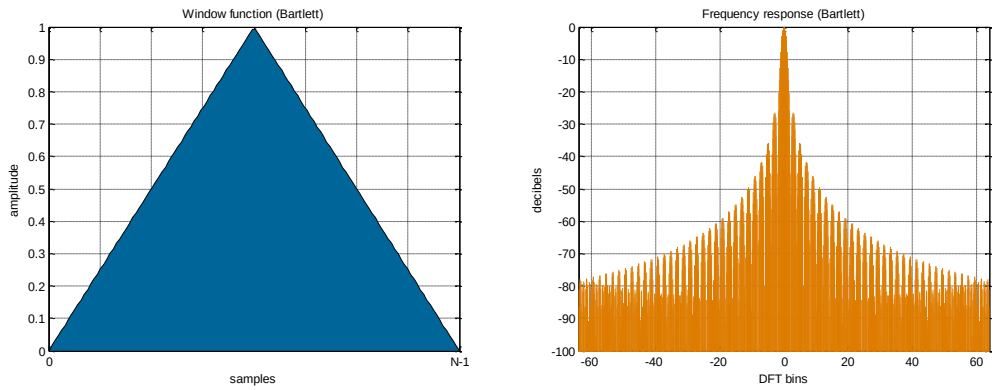

Figure 5 Bartlett Window

2.4 Power Spectral Density

The Power Spectral Density (PSD) is a method of describing how the power of a stationary signal is distributed over frequency. The PSD is estimated using Welch's method for computing a modified periodogram (Oppenheim, 1999, Pg 737-739). In summary, we have a digital time series of length N with a sampling period of T seconds such that:

$$x[n] = x(nT), \qquad 0 \le n \le N-1$$

The sequence is divided and windowed into segments of length L samples with a length R sample step size, using the window function $w[n]$ (see 2.3 Windows):

$$x_r[n] = (x[rR+n] - dc[r]) \cdot w[n], \qquad 0 \le n \le L-1$$

There are three methods by which any dc offset may be removed from the signal:

$$dc[r] = \begin{cases} 0, & None \\ \frac{1}{N}\sum_{n=0}^{N-1} x[n], & Entire \\ \frac{1}{L}\sum_{n=0}^{L-1} x[rR+n], & Block \end{cases}$$

In total, there are K segments where:

$$(K-1)\,R + (L-1) \le N-1$$

$$K \le (N-L)\,/R + 1$$

The segmented time series are represented graphically below:

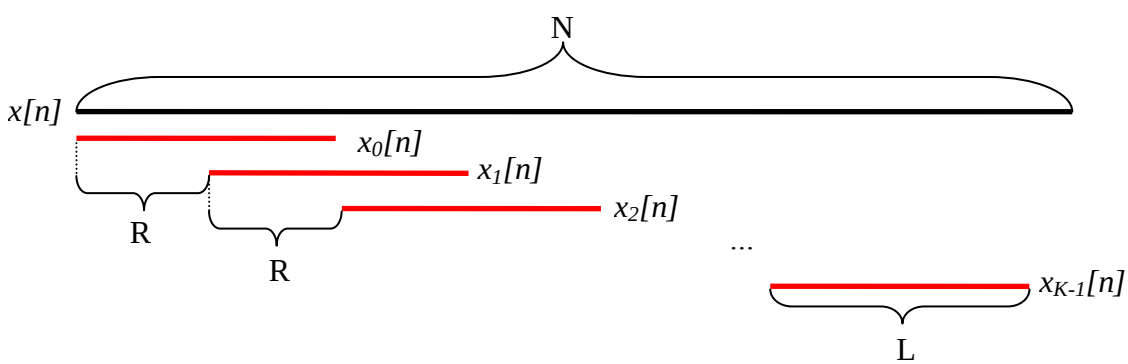

Figure 6 PSD Windowing

Note that there may be a small number of data points, less than R, at the end of the time series that are insufficient to form a full segment of length L. Some applications of the PSD choose to zero-pad the time series so as to be able to include those data points in an additional segment. That is not the approach favored by this application of the PSD. Zero-padding would only serve to introduce a sharp transition in the time domain, especially in the case of tonal data, which would result in spurious frequency content. Instead, we choose to simply not include those data points.

The Fourier Transform (see 2.2 Discrete Fourier Transform) of the rth segment is:

$$X_r[k] = FT\{x_r[n]\}, \qquad 0 \le r \le K-1, 0 \le k, n \le L-1$$

The normalization to account for the attenuation due to the windowing function is:

$$U = \frac{1}{L}\sum_{n=0}^{L-1}(w[n])^2$$

The periodogram of the individual data segment is defined as:

$$I_r[k] = \frac{T}{LU}|X_r[k]|^2, \qquad 0 \le r \le K-1, 0 \le k \le L-1$$

Note that the individual periodogram must be scaled by the sampling period T in order to estimate its continuous time equivalent (Oppenheim, 1999, Pg 731-732).

The averaged periodogram of the entire sequence is:

$$\bar{I}[k] = \frac{1}{K}\sum_{r=0}^{K-1} I_r[k], \qquad 0 \le k \le L-1$$

Because the data we are analyzing is composed of purely real sequences, the FFT and the periodogram are symmetric about the positive and negative frequencies. Thus, we can decrease the storage and any resulting computation by approximately half. Note that it is necessary to double the PSD values (other than at the DC and Nyquist frequencies), so that the power contribution of the negative frequencies are not lost.

$$Pxx[k] = \begin{cases} \bar{I}[k], & k = 0 \\ 2\,\bar{I}[k], & 0 < k < \frac{L}{2} \\ \bar{I}[k], & k = \frac{L}{2} \end{cases}$$

Thus, the length of the sequence $Pxx[k]$ is equal to M where:

$$M = \frac{L}{2} + 1$$

The unit of a PSD is U^2/Hz where U is the unit of the time series. For example, if the time-series samples are in units of Volts, then the unit of the PSD is V^2/Hz.

Typically, the PSD will be converted to decibels for display, for example $dB\ rel\ 1\ V^2/Hz$:

$$10 * \log_{10}(P_{xx}[k])$$

2.4.1 PSD Confidence

The accuracy of the PSD estimate increases as the number of time segments that are included in the spectral averaging grows. An important metric of the PSD is the 90% confidence interval measured in dB. The confidence interval indicates the estimated PSD has a 90% probability of falling within a band, expressed in dB, centered on the actual underlying PSD.

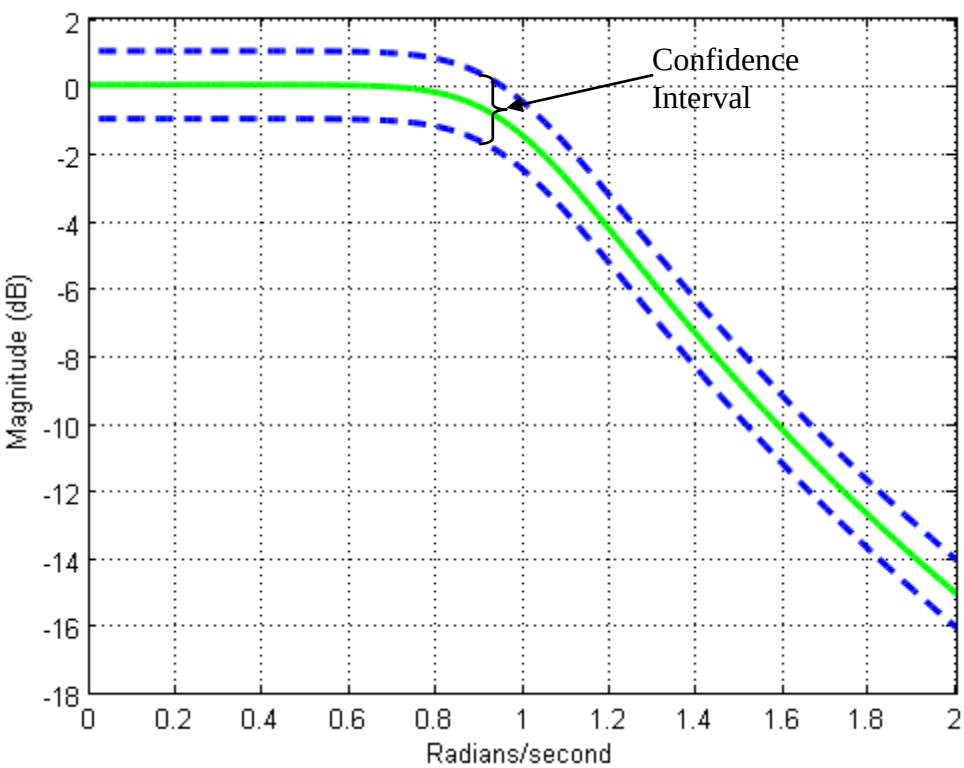

Figure 7 PSD Confidence Interval

For the case of non-overlapping data segments, if the entire data record is of length N and the time window is of length L, then the 90% confidence, in dB, of the PSD estimate is (Stearns, 1990, Pg 378):

$$C_{90\%} = \frac{14.1}{\sqrt{\frac{N}{L/2} - 0.833}}$$

In general, K, the number of data segments for a length R step size, is:

$$K = (N - L)\ /R + \ 1$$

For the non-overlapping case where R = L, the 90% confidence interval becomes:

$$C_{90\%} = \frac{14.1}{\sqrt{2K - 0.833}}$$

Therefore, the confidence interval is observed to decrease as the number of data segments in the PSD estimate increases.

However, this method of estimating the confidence interval does not take into account any potential overlap in the data segments or the windowing function used in constructing the data segments. It is insufficient to assume that the increased number of data segments, K, due to overlapping can simply be substituted into the equation for the confidence interval. The increased overlap will result in a decreased independence between the data segments that will limit the reduction in spectral variance. Depending upon the windowing function being used, the decrease in independence may be limited to some extent.

When estimating a PSD via the periodogram method, the variance of the PSD estimate is proportional to the number of independent time segments, K, that went into the spectral averaging (Oppenheim, 1999, Pg 738). Estimating the contributions of the degree of overlap and the window function to the PSD variance are provided for (Welch, 1967) by computing a weighting of the window function, w[n], where the data segment step size is R samples:

$$\rho[j] = \left[\sum_{n=0}^{L-1} w[n]w[n+jR]\right]^2 \Bigg/ \left[\sum_{n=0}^{L-1} w[n]^2\right]^2, 1 \le j \le K-1$$

Then, the overall reduction in variance in the PSD estimate due to the chosen overlap and window function is equal to:

$$\frac{K}{\left[1 + 2\sum_{j=1}^{K-1} \frac{K-j}{K} \rho(j)\right]}$$

Substitution this equation back into the 90% confidence equation results in:

$$C_{90\%} = \frac{14.1}{\sqrt{\frac{2K}{\left[1 + 2\sum_{j=1}^{K-1} \frac{K-j}{K} \rho(j)\right]} - 0.833}}$$

As would be expected, for the case of non-overlapping, rectangular windows, this updated confidence interval equation degenerates back to the original equation.

As an example, the plot below of theoretical confidence intervals was generated for an arbitrary time series with 100,000 data points, a 1024 point window length, and over a range of step sizes for various window functions (see 0 PSD Confidence for the Matlab® script to generate this plot).

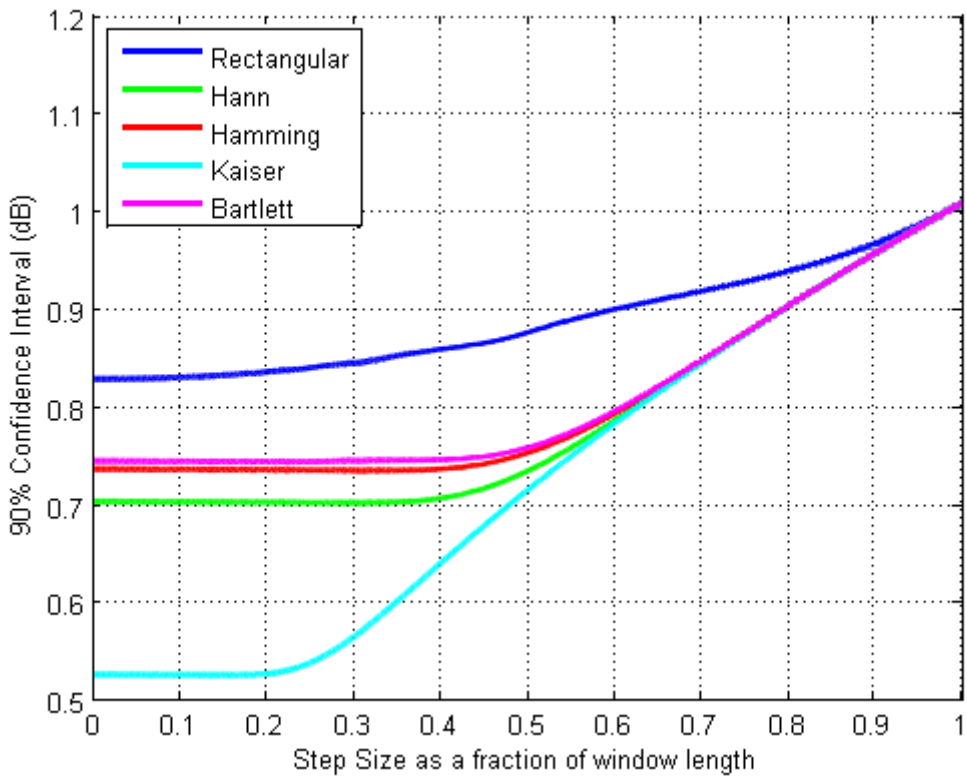

Figure 8 PSD Confidence Interval versus Window Length

From this confidence interval plot, it is observed that considerable gains may be made in reducing the uncertainty of the PSD estimate by allowing for an overlap in the data segments.

However, beyond an overlap of approximately one-half of the window length, there are diminishing returns, due to the decreased independence of the data in successive time segments. Also, for window functions that have greater amounts of taper at the extremes (such as Hann or Kaiser), there are greater potential gains in confidence interval that can be achieved by reducing the step size. For these window functions, the step size can be reduced further without any decrease in the independence between the data in successive time segments.

2.4.2 RMS Estimates

A Root Mean Square (RMS) is a statistical estimate of the signal magnitude of a time varying signal. The RMS may typically be computed in either the time domain or frequency domain.

In the time domain, with a time sampling period of T_s, the RMS estimate is equal to (IEEE Std 100-2000, Pg. 990-991):

$$rms = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}|x[n]|^2}$$

In the frequency domain, computing an RMS estimate across a frequency band of a PSD involves numerically integrating the PSD using the rectangle method.

For a given PSD of length M and window and Fourier Transform of length L:

$$M = \frac{L}{2} + 1$$

The PSD is defined:

$$Pxx[k], \qquad 0 \leq k \leq M - 1$$

Over frequencies (in Hertz):

$$f[k] = \frac{k}{T_s L}, \qquad 0 \leq k \leq M - 1$$

Note that PSD length is equal to approximately half the length of the Fourier Transform due to the method of storing the symmetric PSD of real signals.

Employing Parseval's theorem (Oppenheim, 1999, Pg. 621), the computation of the RMS power across the entire frequency band is:

$$rms = \sqrt{\frac{1}{T_s L}\sum_{k=0}^{M-1}|Pxx[k]|}$$

Where the term $\frac{1}{T_s L}$ is the spacing in Hertz between successive points in the PSD.

For the band-limited case, the RMS value over the frequency range $f[n]$ to $f[m]$, inclusive, is:

$$rms = \sqrt{\frac{1}{T_s L}\sum_{k=n}^{m}|Pxx[k]|}$$

If the given power spectra $Pxx[k]$ has units of U^2/Hz, then the rms estimate has the unit U_{rms}.

2.4.3 Dynamic Range Estimation

A dynamic range estimate is the ratio, typically expressed in decibels, between the largest and smallest signal values that a device can either output or accept as an input. For the purpose of this test analysis, we are choosing to define dynamic range as the signal to noise ratio between the RMS value of a full-scale sinusoid and the RMS of a device's self-noise within a specified bandwidth.

In the time domain, the dynamic range can be estimated using either peak or RMS estimates of the largest and smallest time domain signals.

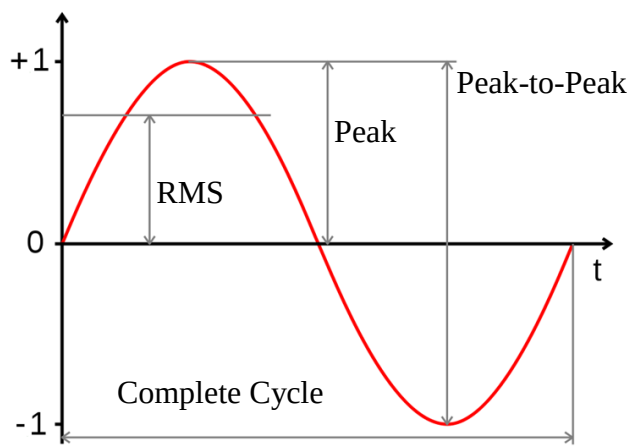

Figure 9 Time Domain Signal Measurements

In determining the dynamic range for a device under test, the low signal level is typically obtained from a frequency domain measurement of the device's self-noise. The high signal level is either obtained from the device's datasheet or determined experimentally by testing. The high signal level is typically expressed as a peak time domain value.

The first step in estimating a frequency domain measurement of self-noise is to determine the frequency band over which the signal levels will be computed. The choice of frequency band is dependent upon the desired application. An example power spectrum from a digitizer Input Terminated Noise test (see 3.2.12 Input Terminated Noise and Maximum Potential Dynamic Range) is shown below:

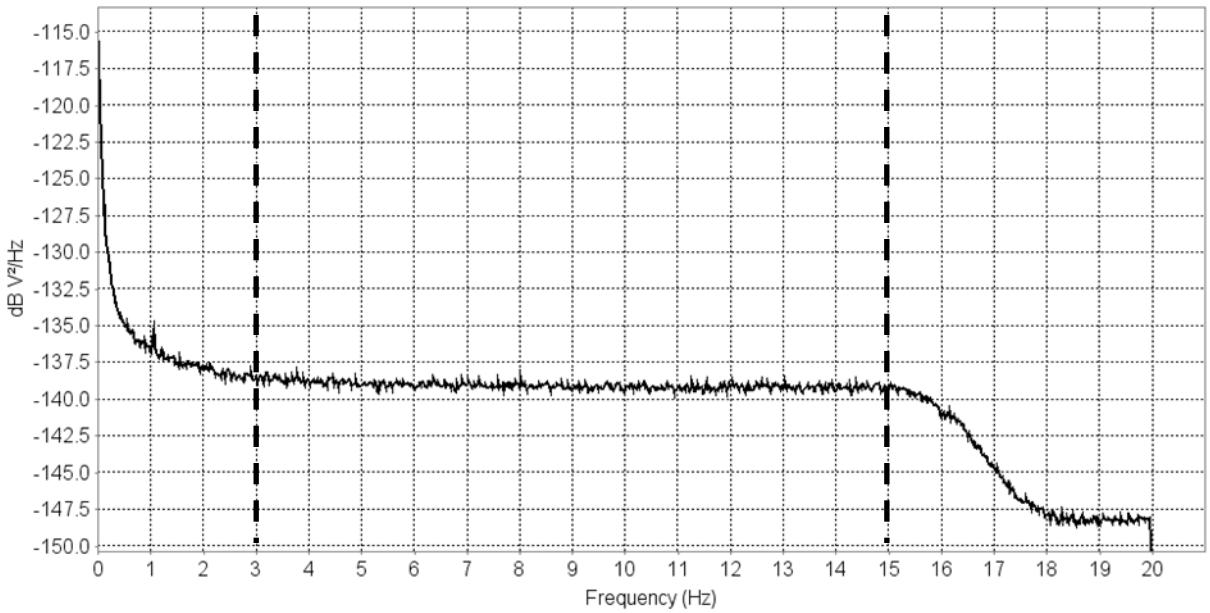

Figure 10 Dynamic Range Power Spectra

An RMS estimate of noise is computed (see 2.4.2 RMS Estimates) over the defined frequency band:

$$V_{noise\ (rms)}$$

For example, in the plot shown above, over a 3 to 15 Hz band there is approximately 0.38 μV_{rms} of noise.

Note that the size of the chosen frequency band will have a significant impact on the magnitude of the noise estimate. As the size of the frequency band increases or decreases, the noise estimate will also increase or decrease as either more or less signal content, respectively, is captured by the frequency band. This variability in the noise estimate due to the frequency band size directly relates to the dynamic range. Therefore, in general, a smaller frequency band will result in a higher dynamic range and a larger frequency band will result in a lower dynamic range. However, it is crucial that the chosen frequency band fully encompasses the band of interest for the desired application, regardless of the impact on dynamic range.

The maximum peak signal level may be obtained from the device's datasheet, verified experimentally:

$$V_{max\ (peak)}$$

In the case of a signal that is assumed to be a pure sinusoid with peak amplitude $V_{max\ (peak)}$, the relationship between its peak and RMS values is:

$$RMS = \frac{Peak}{\sqrt{2}}$$

In this case, the dynamic range in decibels would be:

$$dynamic\ range = 10 \cdot \log_{10}\left(\frac{V_{max\ (peak)}}{V_{noise\ (rms)}} \cdot \frac{1}{\sqrt{2}}\right)^2$$

2.5 Response

A response characterizes how a linear system transforms an input to produce an output. Responses are often used in applications such as modeling the behavior of a sensor or filtering digital time series. There are several ways in which a response may be specified: Pole Zero, Frequency Amplitude Phase values, and Finite Impulse Response values.

2.5.1 Pole Zero

A response may be defined parametrically in terms of its Amplitude (A), Poles (p_i), and Zeros (z_i). A pole-zero response may be either analog (a function of s) or digital (a function of z).

Analog:

$$H_a(s) = A\frac{\prod_{k=1}^{N_{zeros}}(s - \mathrm{z}_k)}{\prod_{k=1}^{N_{poles}}(s - p_k)}$$

Digital:

$$H_d(z) = A\frac{\prod_{k=1}^{N_{zeros}}(z - z_k)}{\prod_{k=1}^{N_{poles}}(z - p_k)}$$

2.5.1.1 Evaluating an Analog Pole Zero Response

An analog pole-zero response, such as the one defined below, may be evaluated at a given frequency *f* by using the following substitution:

$$H_a(s)|_{s=j2\pi f}$$

The resulting value from the response at the defined frequency is a complex value with both a magnitude and phase response.

2.5.1.2 Evaluating a Digital Pole Zero Response

A digital pole-zero response, such as the one defined below, may be evaluated at a given frequency *f* by using the following substitution where F_s is the digital sampling rate (see Oppenheim, 1999, Pg. 95-96):

$$H_d(z)|_{z=e^{\frac{j2\pi f}{F_s}}}$$

The resulting value from the response at the defined frequency is a complex value with both a magnitude and phase response.

2.5.1.3 Analog to Digital Conversion

In order to apply an analog response to a digital time series, it must first be transformed into a digital response. This analog to digital conversion is also used when designing digital filters. This conversion is performed using a bi-linear transform (Oppenheim, 1999, Pg 450-454).

The bi-linear transform allows for a simple substitution to convert between the S-domain and the Z-domain, accounting for the sampling period T_d:

$$s = \frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)$$

So, given an analog response:

$$H_a(s)$$

The discrete-time response is:

$$H_d(z) = \left. H_a(s)\right|_{s=\frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

Re-arranging the z terms in the bi-linear substitution yields:

$$s = \frac{2}{T_d}\left(\frac{1 - Z^{-1}}{1 + z^{-1}}\right)\frac{Z}{z}$$

$$s = \frac{2}{T_d}\left(\frac{z - 1}{z + 1}\right)$$

So, given the definition of an S-domain response, the Z-domain response becomes:

$$H_d(z) = \left. A\frac{\prod_{k=1}^{N_{zeros}}(s - \mathrm{z}_k)}{\prod_{k=1}^{N_{poles}}(s - p_k)}\right|_{s=\frac{2}{T_d}\left(\frac{z-1}{z+1}\right)}$$

Rearranging the terms of the response function results in:

$$H_d(z) = A\frac{\prod_{k=1}^{N_{zeros}}\left(\frac{2}{T_d} - z_k\right)}{\prod_{k=1}^{N_{poles}}\left(\frac{2}{T_d} - p_k\right)}\frac{\prod_{k=1}^{N_{zeros}}\left(z - \frac{\left(\frac{2}{T_d} + z_k\right)}{\left(\frac{2}{T_d} - z_k\right)}\right)\prod_{k=1}^{N_{poles}}(z + 1)}{\prod_{k=1}^{N_{poles}}\left(z - \frac{\left(\frac{2}{T_d} + p_k\right)}{\left(\frac{2}{T_d} - p_k\right)}\right)\prod_{k=1}^{N_{zeros}}(z + 1)}$$

Therefore, the bi-linear transformation may be easily performed using the following rules:

1. For each S-domain zero, z_k:
    a. There is a Z-domain pole at -1
    b. There is a Z-domain zero at $\frac{\left(\frac{2}{T_d}+z_k\right)}{\left(\frac{2}{T_d}-z_k\right)}$
    c. The gain is scaled by $\left(\frac{2}{T_d}-z_k\right)$
2. For each S-domain pole, p_k:
    a. There is a Z-domain zero at -1
    b. There is a Z-domain pole at $\frac{\left(\frac{2}{T_d}+p_k\right)}{\left(\frac{2}{T_d}-p_k\right)}$
    c. The gain is scaled by $\frac{1}{\left(\frac{2}{T_d}-p_k\right)}$

Naturally, many of the Z-domain poles and zeros introduced at -1 will cancel out.

2.5.1.4 Applying a response to digital time series data

The relationship between the input and output of a digital pole-zero response may be represented in the z-domain as shown in the diagram and equation below.

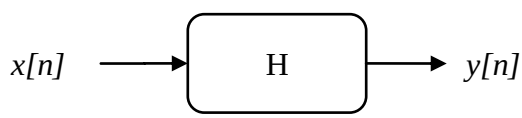

Figure 11 Response Diagram

$$Y = H \cdot X$$

If we have the response, $H(z)$, written in pole-zero form, we can represent the system as a difference equation:

$$H_d(z) = A\frac{\prod_{k=1}^{N_{zeros}}(z-z_k)}{\prod_{k=1}^{N_{poles}}(z-p_k)}$$

The numerator and denominator products may be expressed as a sum of exponents with coefficients:

$$H_d(z) = \frac{\sum_{k=0}^{N_{zeros}} b_k z^k}{\sum_{k=0}^{N_{poles}} a_k z^k}$$

Adjusting the z^k terms to have a negative exponent quantity so that the coefficients can be expressed as delays results in the following, since there are inherently more poles than zeros as required for a causal system:

$$H_d(z) = \frac{\sum_{k=0}^{N_{zeros}} b_k z^{-N_{poles}+k}}{\sum_{k=0}^{N_{poles}} a_k z^{-N_{poles}+k}}$$

Since $Y(z) = H_d(z)\, X(z)$:

$$H_d(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N_{zeros}} b_k z^{-N_{poles}+k}}{\sum_{k=0}^{N_{poles}} a_k z^{-N_{poles}+k}}$$

$$Y(z) \sum_{k=0}^{N_{poles}} a_k z^{-N_{poles}+k} = X(z) \sum_{k=0}^{N_{poles}} a_k z^{-N_{poles}+k}$$

Recall the definition of the Z transform of $x(n)$ and $x(n-n_o)$:

$$Z\{x(n)\} = \sum_n x(n) z^{-n}$$

$$Z\{x(n-n_o)\} = z^{-n_o} X(z)$$

Thus, the Z transform definition may be applied to re-write the response equation in the time domain:

$$\sum_{k=0}^{N_{poles}} a_k y[n - N_{poles} + k] = \sum_{k=0}^{N_{zeros}} b_k x[n - N_{poles} + k]$$

Rearranging the terms to solve for $y[n]$ results in the equation below

$$y[n] = \frac{1}{a_{N_{poles}}} \left[\left(\sum_{k=0}^{N_{zeros}} b_k x[n - N_{poles} + k] \right) - \left(\sum_{k=1`}^{N_{poles}} a_k y[n - N_{poles} + k] \right) \right]$$

2.5.1.5 Removing a response from digital time series data

Removing a response from a digital time series *y[n]* to obtain an estimate of *x[n]* may be viewed as the inverse operation of applying a response (Havskov, 2006, Pg 165-167):

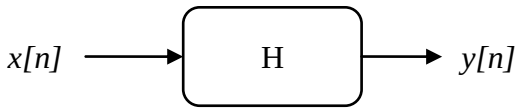

Figure 12 Response Diagram

$$X = \frac{Y}{H}$$

However, from an implementation standpoint, this approach is problematic. By inverting the response, the poles and zeros are reversed. Except for the simplest of responses, this will result in an unstable system (Scherbaum, 2001, Pg 139-146).

The second issue is that removing a response will not be able to reliably reconstruct any signal content outside of the response pass band. There are fundamental limits on the signal resolution that restrict recovery of the attenuated portions outside of the pass band. In addition, if there is any noise that has been added to the signal *y[n]* after the response had been applied, removing the response will serve to severely amplify that noise outside of the response pass band.

The solution we have taken to removing a response from data depends upon whether the correction is being performed in the time domain or in the frequency domain.

For the time domain correction, the response is evaluated at its calibration frequency (see 2.5.1.1 Evaluating an Analog Pole Zero Response) to get a scalar value as an estimate of the response sensitivity. The time series data is divided by the sensitivity to get back to the original units. For example, if the time series data is in volts and the response has the unit of Volt/Meter, then dividing by the response sensitivity will result in a time series whose unit is Meters.

$$x[n] = \frac{y[n]}{H(s)|_{s=j2\pi f_{cal}}}$$

This will yield a time series that is accurate within the pass band of the response, assuming that the response is flat across the pass band and the calibration frequency falls within the pass band.

The response correction must be performed in the frequency domain when the power spectra of the data are to be analyzed (see 2.4 Power Spectral Density). The PSD is shaped by the response evaluated at each of the discrete frequencies that make of the PSD.

The PSD is defined:

$$P_{yy}[k], 0 \leq k \leq N - 1$$

Over frequencies (in Hertz):

$$f[k], 0 \leq k \leq N-1$$

The input PSD is estimated as:

$$P_{xx}[k] = \frac{P_{yy}[k]}{H(s)|_{s=j2\pi f[k]}}, 0 \leq k \leq N-1$$

This will yield a power spectral density that has been corrected by the response

2.5.1.6 Converting a seismic response between unit types

Seismic sensors are transducers that measure the motion of a proof mass relative to a reference frame. Depending upon the sensors design, the position of the proof mass is proportional to Displacement, Velocity, or Acceleration (Havskov, 2001, Pg 23). The response functions below represent the relationship between the transfer function $H(s)$, proof mass displacement $Z(s)$, and the ground displacement $U(s)$:

$$\text{Displacement:} \quad H_d(s) = \frac{Z(s)}{U(s)}$$

$$\text{Velocity:} \quad H_v(s) = \frac{Z(s)}{\dot{U}(s)}$$

$$\text{Acceleration:} \quad H_a(s) = \frac{Z(s)}{\ddot{U}(s)}$$

The relationship between ground displacement, velocity, and acceleration can be obtained by either integration or differentiation (Poularikas, 1999, Pg. 3-3):

$$\ddot{U}(s) = s\,\dot{U}(s) = s^2 U(s)$$

Substituting this relationship back into the equations for the response yields:

$$H_d(s) = s\,H_v(s) = s^2 H_a(s)$$

$$H_v(s) = \frac{1}{s} H_d(s) = s\,H_a(s)$$

$$H_a(s) = \frac{1}{s} H_v(s) = \frac{1}{s^2} H_d(s)$$

Thus, a given response can be converted from its native unit to an equivalent form in a different unit by either multiplying or dividing by some multiple of s.

Often, seismic sensor testing involves making comparisons between the outputs of multiple co-located sensors. There is no guarantee that these sensors will all be of an equivalent type. For example, one of the seismometers may output signals proportional to velocity while another outputs signals proportional to acceleration. However, in order to make a comparison between the data collected from the sensors, it is necessary to convert the sensor output to a common measurement of some earth unit. Therefore, prior to performing a response correction (see 2.5.1.5 Removing a response from digital time series data), the seismic sensor responses must all be converted into a common unit using the equations above.

2.5.2 Frequency-Amplitude-Phase

A response may be defined non-parametrically in terms of a set of Frequency, Amplitude, and Phase values:

$$\{f_i, A_i, P_i\}, \qquad 0 \leq i \leq N-1$$

These non-parametric values may be the result of testing a sensor whose response is otherwise unknown. The number of Frequency, Amplitude, and Phase data points that may be defined is arbitrary and the points may not uniformly sample the desired frequency range of the response.

2.5.2.1 Estimating Analog Pole/Zero locations

The analog pole and zero locations of a FAP response may be estimated by performing a least-squared linear fit between the defined Frequency, Amplitude, and Phase values and a response function of the form:

$$H(s) = A\frac{\sum_{k=0}^{N_{zeros}} b_k s^k}{\sum_{k=0}^{N_{poles}} a_k s^k}$$

It is first necessary to assume the number of poles and zeros in the response model to be estimated. If the number of poles and zeros are unknown, then it is also possible to iterate over a range of combinations to find a response with a minimal misfit. This approach requires the involvement of someone familiar with the expected response characteristics of the sensor to ensure that a reasonably valid response is being estimated. Otherwise, it is possible to generate a response that is unstable or varies considerably outside of the defined response points.

For each Frequency, Amplitude, and Phase value compute the complex response:

$$H_l = A_l e^{jP_l}, \qquad 0 \leq l \leq N-1$$

$$G = \begin{bmatrix} -H_0(j\omega_0)^{N_{zeros}} \\ \vdots \\ -H_{N-1}(j\omega_{N-1})^{N_{zeros}} \end{bmatrix}$$

$$\omega_l = 2\pi F_l, \qquad 0 \le l \le N-1$$

$$D = \begin{bmatrix} H_0(j\omega_0)^0 & \cdots & H_0(j\omega_0)^{N_{poles}} & -(j\omega_0)^0 & \cdots & -(j\omega_0)^{N_{zeros}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_{N-1}(j\omega_{N-1})^0 & \cdots & H_{N-1}(j\omega_{N-1})^{N_{poles}} & -(j\omega_{N-1})^0 & \cdots & -(j\omega_{N-1})^{N_{zeros}} \end{bmatrix}$$

Solve for the coefficients of the response H(s):

$$[\, a;\ b\,] = (D' * D)^{-1} * \ (D' * G)$$

Since we know that the coefficients are real, the solution may be simplified:

$$[\, a;\ b\,] = \Re(D' * D)^{-1} * \ \Re(D' * G)$$

Note that the entries in the matrices above are complex and so the real portion of the complex matrix multiplies must be solved for.

The a and b vectors represent the coefficients of the digital response.

$$H_a(s) = \frac{\sum_{k=0}^{N_{zeros}} b_k s^k}{\sum_{k=0}^{N_{poles}} b_k s^k}$$

The poles and zeros of the response may be obtained by factoring the polynomial to obtain its roots and gain:

$$A = \frac{b_{N_{zeros}}}{a_{N_{poles}}}$$

$$H_a(s) = \ A\frac{\prod_{k=1}^{N_{zeros}}(s - z_k)}{\prod_{k=1}^{N_{poles}}(s - p_k)}$$

2.5.2.2 Estimating Digital Pole/Zero locations

The digital pole and zero locations may be estimated by performing a least-squared linear fit between the defined Frequency, Amplitude, and Phase values and a response function of the form:

$$H(z) = \frac{\sum_{k=0}^{N_{zeros}} b_k z^k}{\sum_{k=0}^{N_{poles}} a_k z^k}$$

It is first necessary to assume the number of poles and zeros in the response model to be estimated. If the number of poles and zeros are unknown, then it is also possible to iterate over a

range of combinations to find a response with a minimal misfit. This approach requires the involvement of someone familiar with the expected response characteristics of the sensor to ensure that a reasonably valid response is being estimated. Otherwise, it is possible to generate a response that is unstable or varies considerably outside of the defined response points.

For each Frequency, Amplitude, and Phase value compute the complex response:

$$H_k = A_k e^{jP_k}, \qquad 0 \le k \le N-1$$

$$G = \begin{bmatrix} -H_0 \\ \vdots \\ -H_{N-1} \end{bmatrix}$$

$$\omega_k = 2\pi F_k T_d, \qquad 0 \le k \le N-1$$

$$D = \begin{bmatrix} H_0 e^{j0w_0} & \cdots & H_0 e^{jN_{poles}w_0} & -e^{j0w_0} & \cdots & -e^{jN_{zeros}w_0} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_{N-1} e^{j0w_{N-1}} & \cdots & H_{N-1} e^{jN_{poles}w_{N-1}} & -e^{j0w_{N-1}} & \cdots & -e^{jN_{zeros}w_{N-1}} \end{bmatrix}$$

Solve for the coefficients of the response $H(z)$:

$$[\, a;\ b \,] = (D' * D)^{-1} * \ (D' * G)$$

Since we know that the coefficients are real, the solution may be simplified:

$$[\, a;\ b \,] = \Re(D' * D)^{-1} * \ \Re(D' * G)$$

Note that the entries in the matrices above are complex.

The a and b vectors represent the coefficients of the digital response:

$$H_d(z) = \frac{\sum_{k=0}^{N_{zeros}} b_k z^k}{\sum_{k=0}^{N_{poles}} a_k z^k}$$

The poles and zeros of the response may be obtained by factoring the polynomial to obtain its roots and gain:

$$A = \frac{b_{N_{zeros}}}{a_{N_{poles}}}$$

$$H_d(z) = A \frac{\prod_{k=1}^{N_{zeros}} (z - z_k)}{\prod_{k=1}^{N_{poles}} (z - p_k)}$$

2.5.2.3 Evaluating a FAP Response

A FAP response may be evaluated at a given frequency by first estimating the poles and zeros for such a response, as described in 2.5.2.1 Estimating Analog Pole/Zero locations or 2.5.2.2 Estimating Digital Pole/Zero locations.

Once a pole-zero estimate of the FAP response is obtained, the estimated response may be evaluated as described in 2.5.1.1 Evaluating an Analog Pole Zero Response or 2.5.1.2 Evaluating a Digital Pole Zero Response.

2.6 Filters

A digital filter is applied to waveform time series to selectively remove portions of the frequency content. The design of a digital filter is fairly straight forward, given the initial selection of a few key parameters: The filter type, order, and band pass frequencies.

First, an analog low-pass filter with a cutoff frequency at 1 radian/second is designed (see 2.6.1 Analog Filter Design).

Second, the analog low-pass filter is pre-warped for the chosen frequency band (see 2.6.2 Frequency Warping).

Finally, the pre-warped filter is converted from analog to digital using a bilinear transformation (see 2.5.1.3 Analog to Digital Conversion).

Once the digital filter has been completed, it may be applied to the digital time series (see 2.5.1.4 Applying a response to digital time series data).

2.6.1 Analog Filter Design

The first step in designing a digital filter is to create an analog low-pass filter with a cutoff frequency at 1 radian/second. The type of filter and its order may be selected in order to optimize various parameters such as the sharpness of the cutoff frequency, ripple inside the pass band, ripple outside the pass band, phase response, etc.

The sections for each type of analog filter include several pole-zero and magnitude-phase plots for various orders.

2.6.1.1 Butterworth

A Butterworth filter (see Oppenheim, 1999, Pg 824-826) is designed such that its magnitude response is flat within the pass band. An analog, low-pass, N^{th}-order Butterworth filter with a cutoff frequency at 1 radian/second has no zeros and its poles are equally spaced around the left half of the unit circle:

$$H_a(s) = \frac{1}{\prod_{k=1}^{N}\left(s - e^{\frac{j\pi(2i+N-1)}{2N}}\right)}$$

The pole-zero plots and magnitude-phase plots are shown below for several different orders.

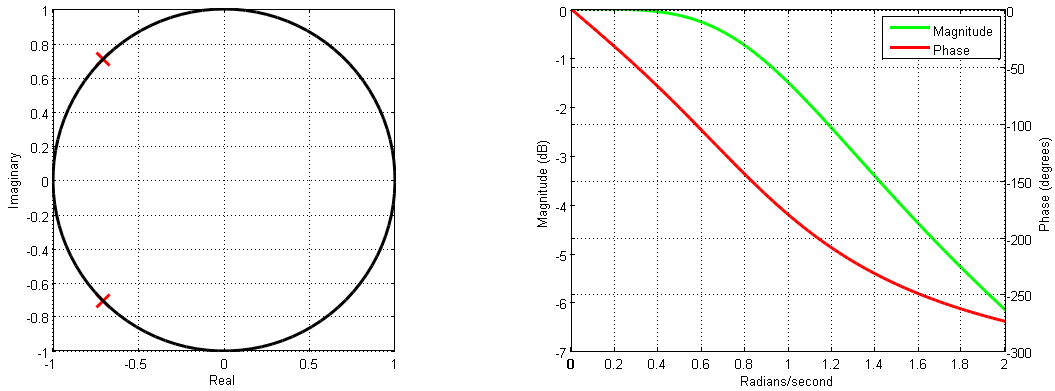

Figure 13 Butterworth Filter – 2nd Order

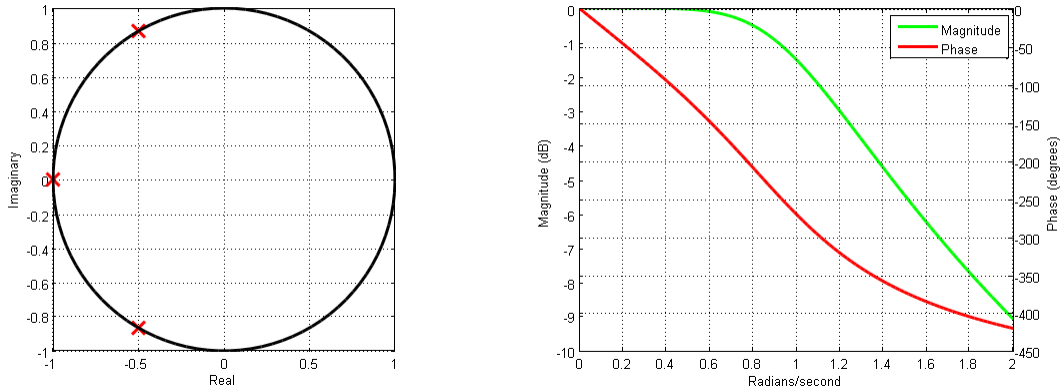

Figure 14 Butterworth Filter – 3rd Order

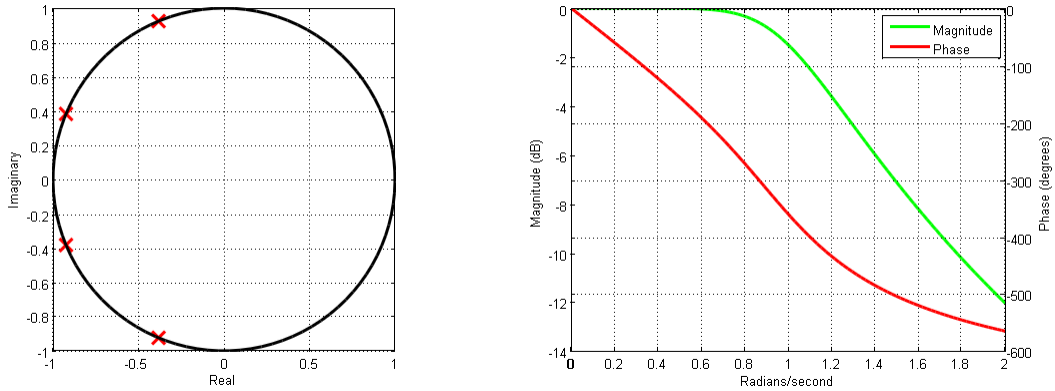

Figure 15 Butterworth Filter – 4th Order

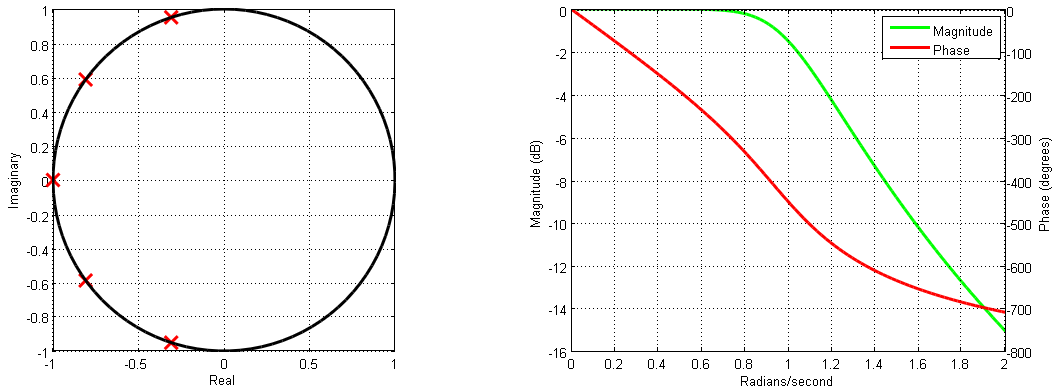

Figure 16 Butterworth Filter – 5th Order

2.6.1.2 Chebyshev 1

A Chebyshev Type 1 filter (see Oppenheim, 1999, Pg 826-828) is designed such that its magnitude response error is distributed across the passband, which is readily visible as ripple. An analog, low-pass, N^{th}-order Chebyshev 1 filter with a cutoff frequency at 1 radian/second has no zeros and its poles are equally spaced around an ellipse contained within the unit circle.

The pole-zero plots and magnitude and phase responses are shown below for several different orders.

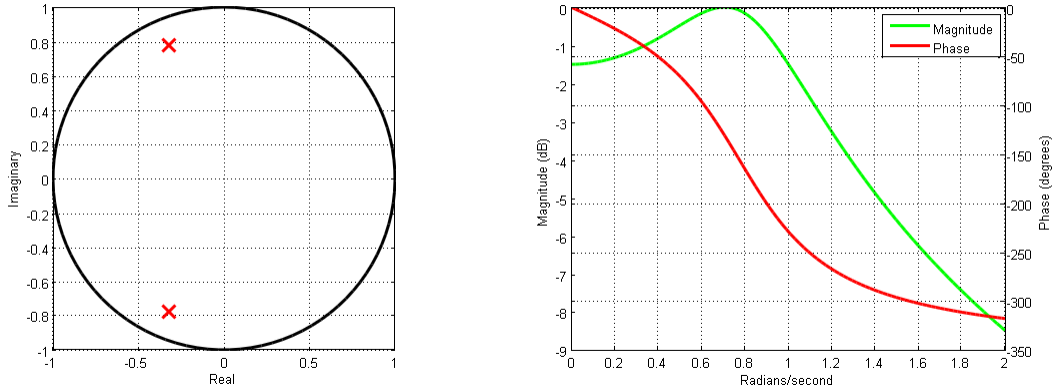

Figure 17 Chebyshev Type 1 Filter – 2nd Order

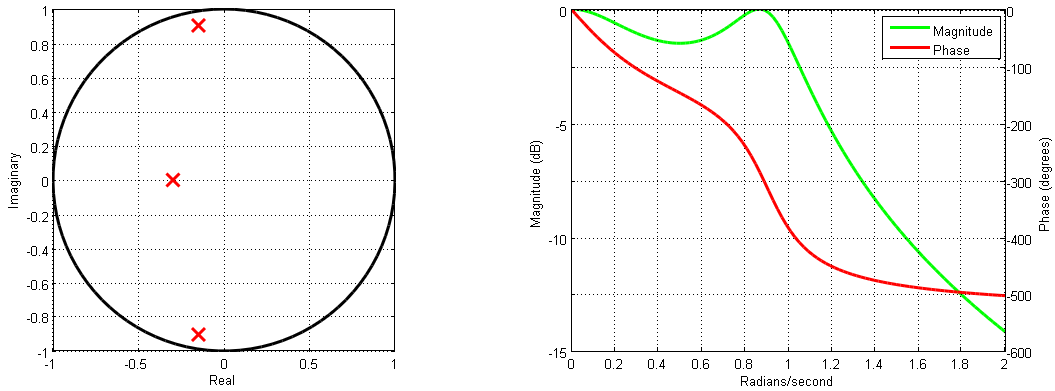

Figure 18 Chebyshev Type 1 Filter – 3rd Order

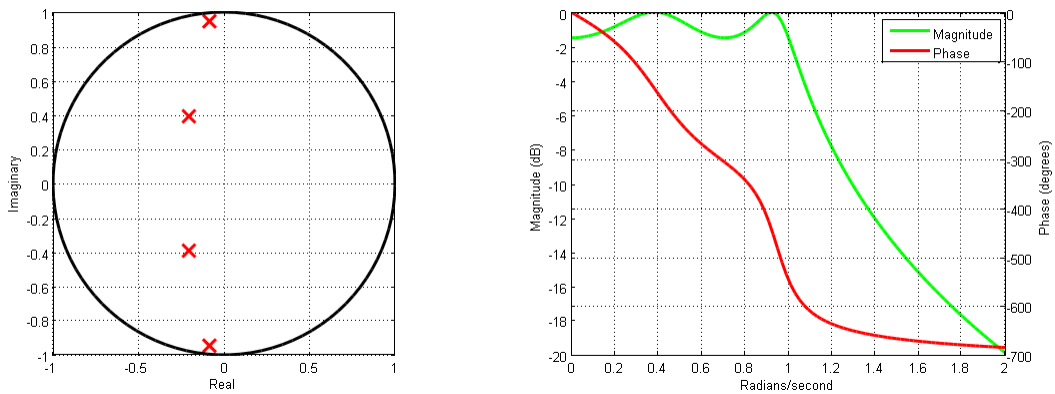

Figure 19 Chebyshev Type 1 Filter – 4th Order

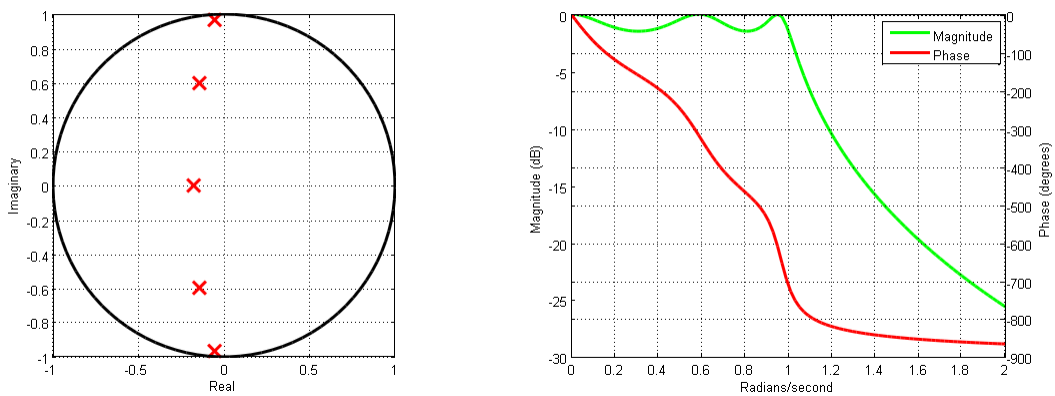

Figure 20 Chebyshev Type 1 Filter – 5th Order

2.6.1.3 Chebyshev 2

A Chebyshev Type 2 filter (see Oppenheim, 1999, Pg 826-828) is designed such that its magnitude response error is distributed across the stopband, which is readily visible as ripple. An analog, low-pass, N^{th}-order Chebyshev 2 filter with a cutoff frequency at 1 radian/second is obtained by transforming a Chebyshev 1 filter.

The pole-zero plots and magnitude and phase responses are shown below for several different orders.

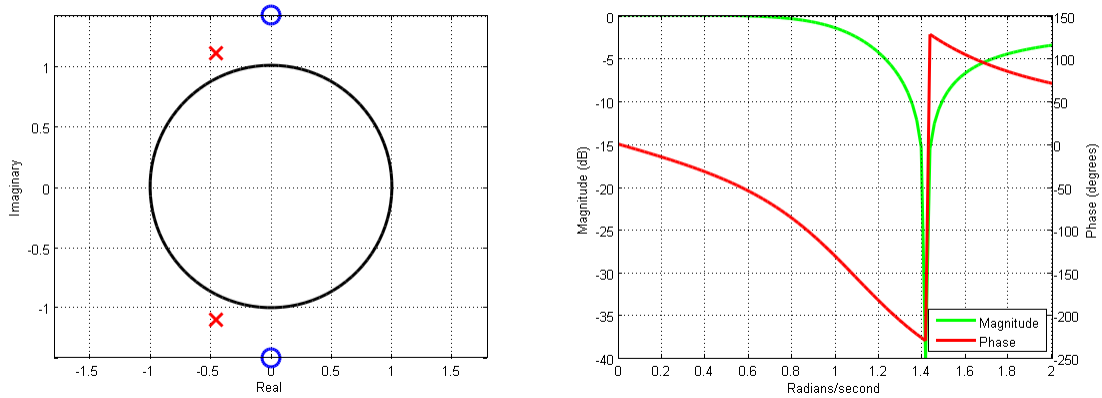

Figure 21 Chebyshev Type 2 Filter – 2nd Order

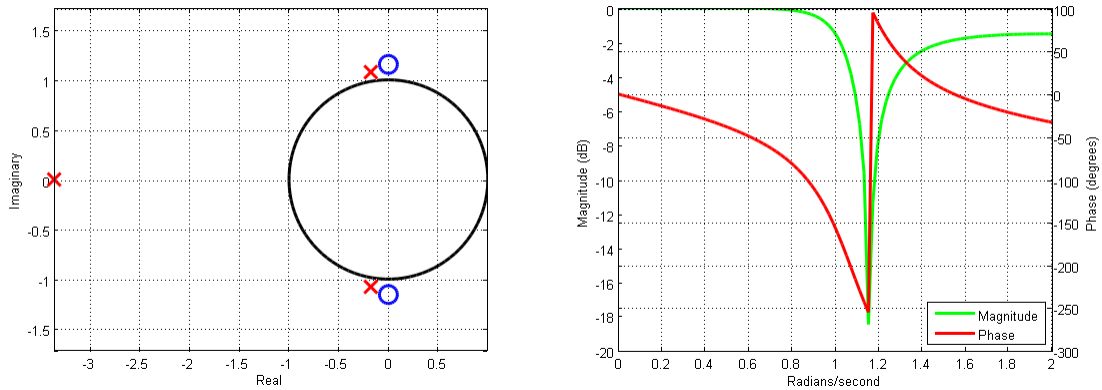

Figure 22 Chebyshev Type 2 Filter – 3rd Order

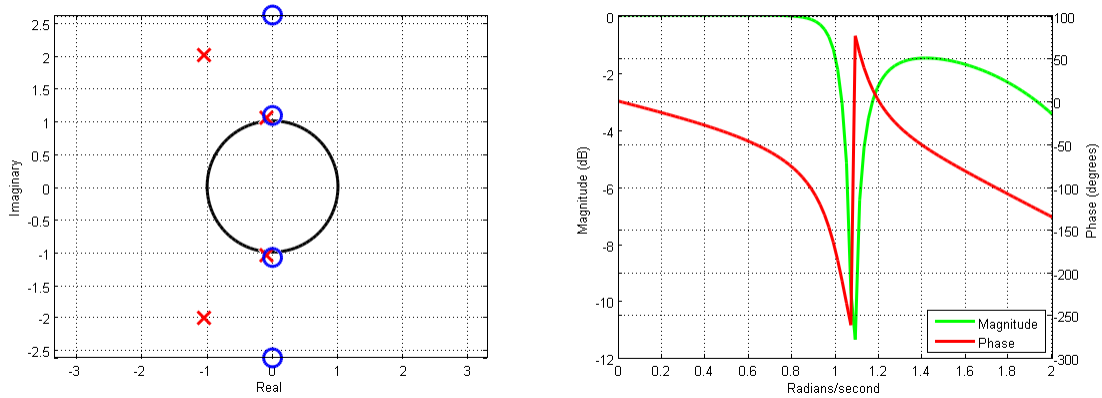

Figure 23 Chebyshev Type 2 Filter – 4th Order

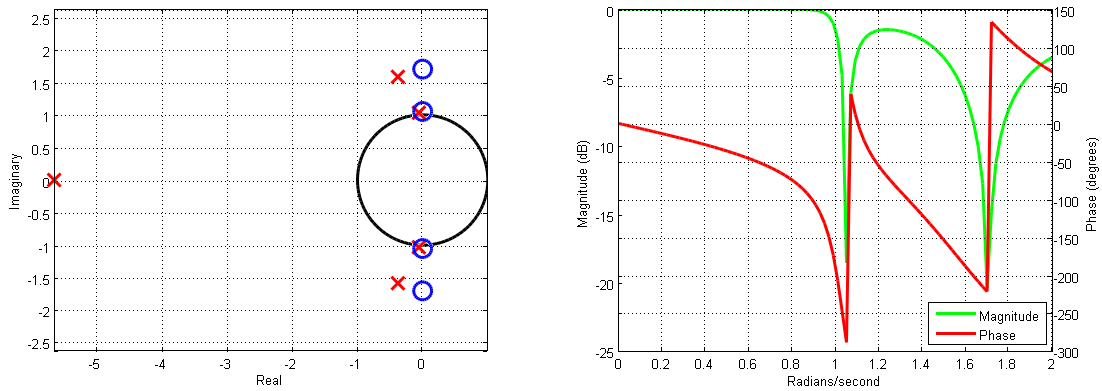

Figure 24 Chebyshev Type 2 Filter – 5th Order

2.6.1.4 Bessel

A Bessel filter is designed to have a phase response that is flat at zero frequency and linear phase through its pass band. A Bessel filter has no zeros and the poles can be solved through analytic modeling. However, they are more typically stored within a lookup table for faster computation (see Matlab command "besselap").

The pole-zero plots and magnitude and phase responses are shown below for several different orders.

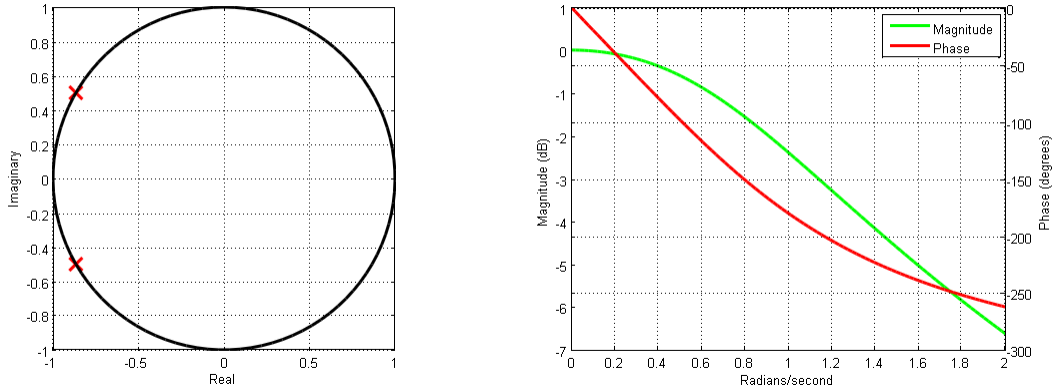

Figure 25 Bessel Filter – 2nd Order

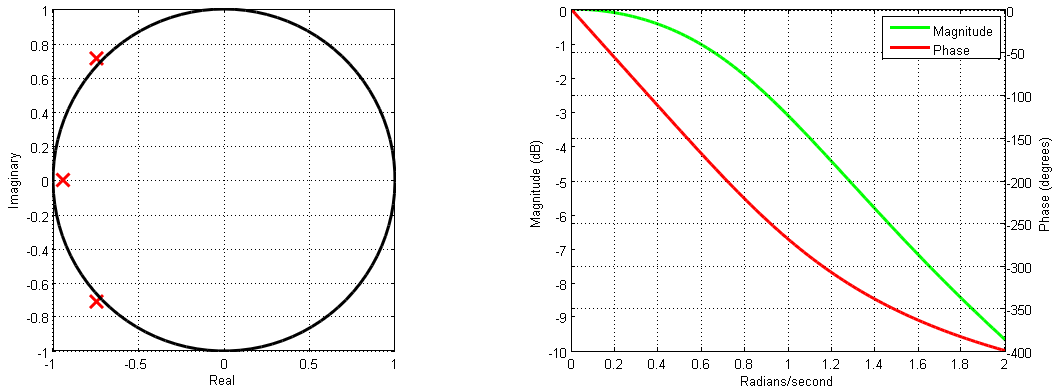

Figure 26 Bessel Filter – 3rd Order

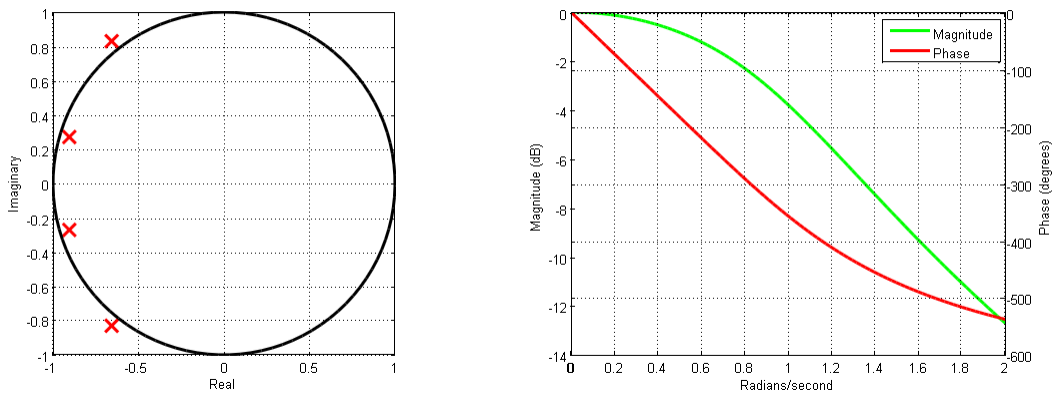

Figure 27 Bessel Filter – 4th Order

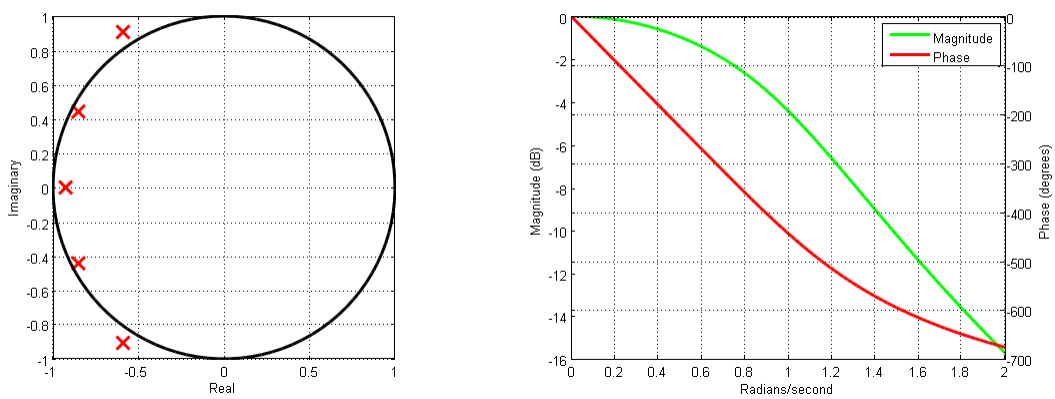

Figure 28 Bessel Filter – 5th Order

2.6.2 Frequency Warping

The second step in designing a digital filter is to warp the analog low-pass filter depending upon the desired frequency band limits of the filter. Frequency warping accomplishes two primary tasks. First, it converts the generic low pass analog filter into the specific band type that is desired. Second, it corrects for the distortion in frequency that is introduced by the bi-linear transform that will be used in a later stage of the filter design.

The warping is performed by translating the locations of the poles and zeros. For all of these types of filter bands, the frequency warping first begins with an analog low-pass filter with a cutoff frequency at 1 radian/second, as constructed in the previous section:

$$H(s)$$

2.6.2.1 Low Pass

A Low Pass filter has a single cutoff frequency. Any frequency content below that cutoff is retained while any frequency above that cutoff is removed. If the desired cutoff frequency in Hertz is at F_c with a sampling rate of F_s, then the Low Pass filter may be obtained from the following substitution (see Stearns, 1993, Pg 126-127*):

$$\Omega_c = 2F_s \tan\left(\frac{\pi F_c}{F_s}\right)$$

$$H_{LP}(s) = H(s)|_{s=s/\Omega_c}$$

*Note that the additional $2F_s$ factor that must be applied to the cutoff frequency is due to Stearn's definition of the bilinear transformation on page 126 that does not account for the sampling rate as the standard bilinear transformation definition does (see 2.5.1.3 Analog to Digital Conversion).

2.6.2.2 High Pass

A High Pass filter has a single cutoff frequency. Any frequency content above that cutoff is retained while any frequency content below that cutoff is removed. If the desired cutoff frequency in Hertz is at F_c with a sampling rate of F_s, then the High Pass filter may be obtained from the following substitution (see Stearns, 1993, Pg 129):

$$\Omega_c = 2F_s \tan\left(\frac{\pi F_c}{F_s}\right)$$

$$H_{HP}(s) = H(s)|_{s=\Omega_c/s}$$

2.6.2.3 Band Pass

A Band Pass filter has low and high cutoff frequencies. Any frequency content within the cutoffs is retained while any frequency content outside the cutoffs is removed. If the desired cutoff frequencies in Hertz are $F_L$and $F_H$with a sampling rate of F_s, then the Band Pass filter may be obtained from the following substitution (see Stearns, 1993, Pg 129):

$$\Omega_L = 2F_s \tan\left(\frac{\pi F_L}{F_s}\right)$$

$$\Omega_H = 2F_s \tan\left(\frac{\pi F_H}{F_s}\right)$$

$$\Omega_0 = \sqrt{\Omega_L \Omega_H}$$
$$W = \Omega_H - \Omega_L$$

$$H_{BP}(s) = H(s)\Big|_{s=\frac{(s^2+\Omega_0^2)}{Ws}}$$

2.6.2.4 Band Stop

A Band Stop filter has low and high cutoff frequencies. Any frequency content outside the cutoffs is retained while any frequency content within the cutoffs is removed. If the desired cutoff frequencies in Hertz are $F_L$and $F_H$with a sampling rate of F_s, then the Band Stop filter may be obtained from the following substitution (see Stearns, 1993, Pg 129):

$$\Omega_L = 2F_s \tan\left(\frac{\pi F_L}{F_s}\right)$$
$$\Omega_H = 2F_s \tan\left(\frac{\pi F_H}{F_s}\right)$$

$$\Omega_0 = \sqrt{\Omega_L \Omega_H}$$
$$W = \Omega_H - \Omega_L$$

$$H_{BS}(s) = H(s)|_{s=1/s}\Big|_{s=\frac{(s^2+\Omega_0^2)}{Ws}}$$

2.7 Coherence

2.7.1 Two Channel Coherence

The Two Channel Coherence technique (Holcomb, 1989) allows for a limited estimate of the channel noise and the relative response between the two channels. The generic system model for two systems with a common input, independent responses, and linearly additive noise sources is shown below.

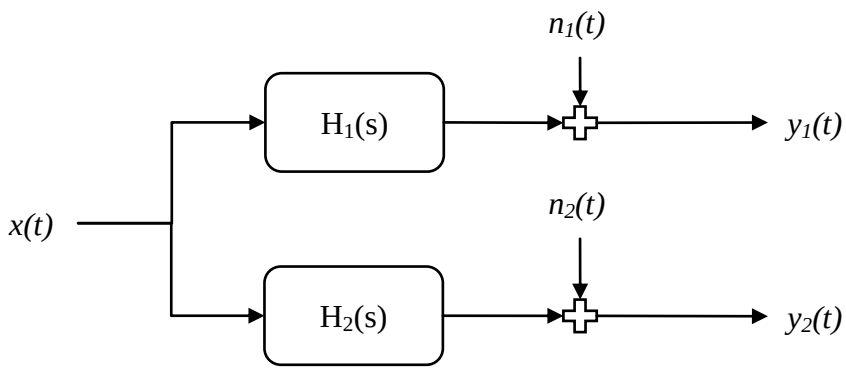

Figure 29 Two Channel Coherence System Diagram

The common input to the two systems is represented by the time varying quantity *x(t)*. The system responses $H_1(s)$ and $H_2(s)$ describe the effect of the system on the amplitude and phase of the input. $n_1(t)$ and $n_2(t)$ represent additive noise sources coming after the responses. $y_1(t)$ and $y_2(t)$ represent the recorded signals.

The frequency domain equations, in which capitalized variables represent the Fourier Transform of the time domain signals and responses, describing the observed outputs for the systems depicted above are:

$$Y_1 = H_1X + N_1$$
$$Y_2 = H_2X + N_2$$

The corresponding auto and cross power spectra are:

$$\begin{aligned} P_{y_1y_1} &= Y_1Y_1^* \\ &= (H_1X + N_1)(H_1X + N_1)^* \\ &= H_1H_1^*XX^* + H_1XN_1^* + H_1^*X^*N_1 + N_1N_1^* \end{aligned}$$

$$\begin{aligned} P_{y_2y_2} &= Y_2Y_2^* \\ &= (H_2X + N_2)(H_2X + N_2)^* \\ &= H_2H_2^*XX^* + H_2XN_2^* + H_2^*X^*N_2 + N_2N_2^* \end{aligned}$$

$$\begin{aligned} P_{y_1y_2} &= Y_1Y_2^* \\ &= (H_1X + N_1)(H_2X + N_2)^* \\ &= H_1H_2^*XX^* + H_1XN_2^* + H_2^*X^*N_1 + N_1N_2^* \end{aligned}$$

Since it is assumed that the linearly additive noise sources are independent of each other and the input signal, we can assume that $N_1N_2 = 0$, $XN_2 = 0$, and $XN_2 = 0$. The equations above simplify to:

$$\begin{aligned} P_{y_1y_1} &= H_1H_1^*XX^* + N_1N_1^* \\ &= |H_1|^2P_{xx} + P_{n_1n_1} \end{aligned}$$

$$\begin{aligned} P_{y_2y_2} &= H_2H_2^*XX^* + N_2N_2^* \\ &= |H_2|^2P_{xx} + P_{n_2n_2} \end{aligned}$$

$$\begin{aligned} P_{y_1y_2} &= H_1H_2^*XX^* \\ &= H_1H_2^*P_{xx} \end{aligned}$$

The mean squared coherence between the output channels is computed from the auto and cross power spectra (see 2.4 Power Spectral Density) of the two channels:

$$\gamma^2 = \frac{\left|P_{y_1y_2}\right|^2}{P_{y_1y_1}P_{y_2y_2}}$$

In order to compute an estimate of the relative response and noise, some assumptions must be made about the systems. First, one of the channels, which is designated the reference channel, is assumed to have a unity system response, H(s). Second, constraints must be placed on the linearly additive noise. These constraints are referred to as being either "Lumped" or "Distributed".

2.7.1.1 Lumped Noise

In the case of the lumped noise model, it is assumed that there is no noise present in the first reference channel and that all of the noise is present in the second channel. This model is useful when the reference device is assumed to have noise characteristics that are significantly lower than that of the test device.

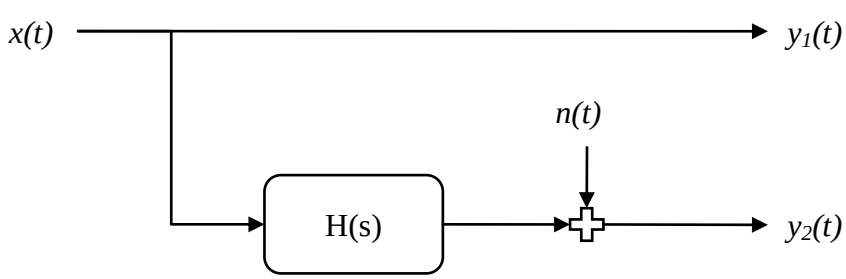

Figure 30 Two Channel Coherence Lumped System Diagram

The system model diagram above may be expressed as the frequency equations below:

$$Y_1 = X$$

$$Y_2 = XH + N$$

Using the system model frequency equations, the auto and cross spectra may be expressed as the following:

$$\begin{aligned} P_{y_1 y_1} &= Y_1 Y_1^* \\ &= XX^* \\ &= P_{xx} \end{aligned}$$

$$\begin{aligned} P_{y_2 y_2} &= Y_2 Y_2^* \\ &= (XH + N)(XH + N)^* \\ &= XX^* HH^* + XN^* H + NX^* H^* + NN^* \\ &= |H|^2 P_{xx} + P_{nn} \end{aligned}$$

$$\begin{aligned} P_{y_1 y_2} &= Y_1 Y_2^* \\ &= X(XH + N)^* \\ &= XX^* H^* + XN^* \\ &= H^* P_{xx} \end{aligned}$$

The noise power estimate for the lumped noise model is:

$$\begin{aligned} P_{nn} &= P_{y_2 y_2} - |H|^2 P_{xx} \\ &= P_{y_2 y_2} - HH^* P_{xx} \\ &= P_{y_2 y_2} - HP_{y_1 y_2} \\ &= P_{y_2 y_2} - \frac{P_{y_1 y_2}^*}{P_{y_1 y_1}} P_{y_1 y_2} \\ &= P_{y_2 y_2} \left(1 - \frac{P_{y_1 y_2}^* P_{y_1 y_2}}{P_{y_1 y_1} P_{y_2 y_2}} \right) \\ &= P_{y_2 y_2} (1 - \gamma^2) \end{aligned}$$

The estimate for the relative linear transfer function is:

$$\begin{aligned} H &= \frac{P_{y_1 y_2}^*}{P_{xx}} \\ &= \frac{P_{y_1 y_2}^*}{P_{y_1 y_1}} \frac{P_{y_1 y_2} P_{y_2 y_2}}{P_{y_1 y_2} P_{y_2 y_2}} \\ &= \gamma^2 P_{y_2 y_2} / P_{y_1 y_2} \end{aligned}$$

2.7.1.2 Distributed Noise

In the case of the distributed noise model, it is assumed that the noise is equally distributed between the two channels. This model is useful when the reference device is assumed to have approximately equal performance to the test device. In this case, the two noise sources, $n_1(t)$ and $n_2(t)$, are assumed to be independent but possessing identical power spectra.

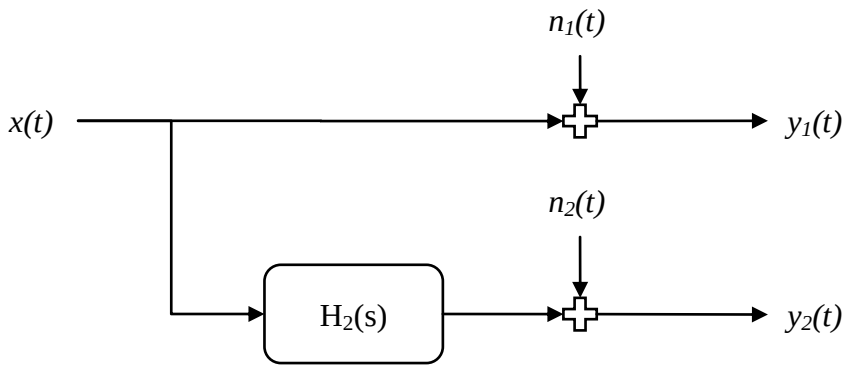

Figure 31 Two Channel Coherence Distributed System Diagram

The system model diagram above may be expressed as the frequency equations below

$$Y_1 = X + N$$

$$Y_2 = XH + N$$

Using the system model frequency equations, the auto and cross spectra may be expressed as the following

$$\begin{aligned} P_{y_1 y_2} &= Y_1 Y_2^* \\ &= (X + N_1)(XH + N_2)^* \\ &= XX^*H^* + XN_2^* + N_1X^*H^* + N_1N_2^* \\ &= P_{xx}H^* \end{aligned}$$

$$\begin{aligned} P_{y_1 y_1} &= Y_1 Y_1^* \\ &= (X + N_1)(X + N_1)^* \\ &= XX^* + XN_1^* + N_1X^* + N_1N_1^* \\ &= P_{xx} + P_{nn} \end{aligned}$$

$$\begin{aligned} P_{y_2 y_2} &= Y_2 Y_2^* \\ &= (XH + N_2)(XH + N_2)^* \\ &= XX^*HH^* + XN_2^*H + N_2X^*H^* + N_2N_2^* \\ &= P_{xx}|H|^2 + P_{nn} \end{aligned}$$

Solving these three equations for the noise power estimate for the lumped noise model:

$$P_{nn} = \frac{1}{2}\left(P_{y_1y_1} + P_{y_2y_2}\right) - \sqrt{\frac{1}{4}\left(P_{y_1y_1} - P_{y_2y_2}\right)^2 + \left|P_{y_1y_2}\right|^2}$$

The estimate for the relative transfer function is:

$$\begin{aligned} H &= \frac{P^*_{y_1y_2}}{P_{xx}} \\ &= \frac{P^*_{y_1y_2}}{\left(P_{y_1y_1} - P_{nn}\right)} \end{aligned}$$

2.7.2 Three Channel Coherence

The Three Channel Coherence technique (Sleeman, 2006) allows for the noise estimate of each channel and the relative response between each pair of channels to be uniquely determined. There is no assumption of how the noise is distributed between the channels, as there is with the Two Channel Coherence technique.

The three-sensor coherence analysis technique assumes that the three channels are all measuring a common input signal. Each channel has its own response, H_i, and some amount of independent, incoherent noise, n_i.

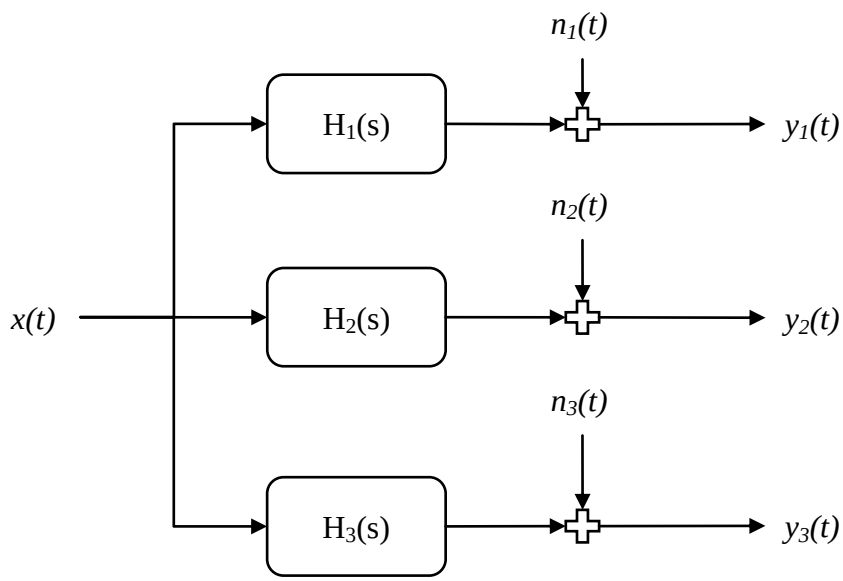

Figure 32 Three Channel Coherence System Diagram

From the system diagram above, the frequency domain equations relating the common input and the 3 channels of output are:

$$Y_1 = XH_1 + N_1$$
$$Y_2 = XH_2 + N_2$$
$$Y_3 = XH_3 + N_3$$

The cross power spectrum between each pair of channel outputs can be computed as:

$$P_{ij} = Y_i Y_j^*$$

If we assume that the internal noise, N_i, and the input signal, X, are independent such that $X \cdot N_i = 0$, then the cross power spectrum becomes:

$$P_{ij} = P_{xx} H_i H_j^* + N_{ij}$$

If we further assume that the noise in each channel is uncorrelated such that $N_{ij} = 0$ for $i \neq j$, then the relative transfer function between any two channels (for $i \neq j \neq k$) then becomes:

$$\frac{H_j}{H_k} = \frac{P_{ji}}{P_{ki}}$$

It is also possible to show that the noise auto power spectrum is:

$$N_{ii} = P_{ii} - \frac{P_{ji}P_{ik}}{P_{jk}}$$

Note that that this method of computing the noise spectra often results in a complex estimate. It is necessary to compute the magnitude of the noise estimate to obtain its value.

2.7.3 N Channel Coherence

If we have an arbitrary number of channels, N, where N >= 3, then it is possible to simultaneously solve for the all of the noise auto power spectra and relative transfer functions by examining the signal cross and auto power spectra.

As in the 3 Channel Coherence case, the underlying assumption is that the noise in each channel is uncorrelated with the input signal and with the noise in the other channels.

The relative transfer function is computed by averaging across all possible combinations

$$\frac{H_j}{H_k} = \frac{1}{N-2} \sum_{i \in 1..N}^{i \neq j \neq k} \frac{P_{ji}}{P_{ki}}$$

The auto power noise spectrum is computed by averaging across all possible combinations of the remaining channels taken two at a time.

$$N_{ii} = P_{ii} - \frac{1}{\binom{N-1}{2}} \sum_{j,k \in 1..N}^{i \neq j \neq k} \frac{P_{ji}P_{ik}}{P_{jk}}$$

2.8 Sine Fit

The Sine Fit algorithm performs a least-squared fit to determine the sinusoid parameters (DC Offset, amplitude, frequency, and phase) that most closely matches the supplied time series data. There are two methods for computing a Sine Fit. The first method is a three parameter Sine Fit in which the frequency is known beforehand and there is a simple close-formed least-squared solution for the amplitude, phase, and DC Offset. The second method is a four parameter Sine Fit algorithm that solves for frequency, amplitude, phase, and DC Offset. The four parameter algorithm requires an iterative solution with initial guesses for the parameters.

Note that although the algorithm is termed "Sine Fit", the parameters that are estimated are actually for a cosine function.

2.8.1 Three Parameter Sine Fit

The Three Parameter Sine Fit algorithm (IEEE Std 1241-2000, Pg. 26) solves for the three parameters (A_0, B_0, and C_0) for which the model below best fits the supplied data:

$$y[n] = A_0 \cos(\omega_0 t_n) + B_0 \sin(\omega_0 t_n) + C_0$$

by minimizing the RMS error between the supplied data and the model:

$$\varepsilon_{rms} = \frac{1}{M} \sum_{n=1}^{M} [y_n - y[n]]^2$$

Where the supplied data:

$$y_n, \qquad 1 \le n \le M$$

Is taken at times:

$$t_n, \qquad 1 \le n \le M$$

The angular frequency, $\omega_0 = 2\pi f$, is assumed to be known. Let:

$$D = \begin{bmatrix} \cos(\omega_0 t_1) & \sin(\omega_0 t_1) & 1 \\ \cos(\omega_0 t_2) & \sin(\omega_0 t_2) & 1 \\ \vdots & \vdots & \vdots \\ \cos(\omega_0 t_M) & \sin(\omega_0 t_M) & 1 \end{bmatrix}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}$$

And

$$X_0 = \begin{bmatrix} A_0 \\ B_0 \\ C_0 \end{bmatrix}$$

Set up the linear system of equations:

$$Y = DX_0$$

Solve for the A_0, B_0, and C_0 parameters:

$$X_0 = (D^T D)^{-1}(D^T Y)$$

Convert the three parameters into the form:

$$A\ \cos(\omega_0 t_n + \theta) +\ C$$

Where:

$$A = \sqrt{A_0^2 + B_0^2}$$

$$\theta = \begin{cases} \tan^{-1}\left(-\dfrac{B_0}{A_0}\right), & for\ A_0 \geq 0 \\ \tan^{-1}\left(-\dfrac{B_0}{A_0}\right) + \pi, & for\ A_0 < 0 \end{cases}$$

$$C =\ C_0$$

2.8.2 Four Parameter Sine Fit

The Four Parameter Sine Fit algorithm (IEEE Std 1241-2000, Pg. 27) solves for the four parameters (A_0, B_0, C_0, and ω_0) for which the model below best fits the supplied data:

$$y[n] = A_0 \cos(\omega_0 t_n) +\ B_0 \sin(\omega_0 t_n) +\ C_0$$

by minimizing the RMS error between the supplied data and the model:

$$\varepsilon_{rms} = \frac{1}{M}\sum_{n=1}^{M}[y_n - y[n]]^2$$

Where the supplied data:

$$y_n, \qquad 1\ \leq n\ \leq M$$

Is taken at times:

$$t_n, \qquad 1 \le n \le M$$

This algorithm must be solved iteratively due to the necessity to perform a non-linear estimation of the frequency. Because of the iterative method of solution, the four parameter sine fit algorithm can be unstable if a poor choice is made for the initial frequency.

Follow sequence of steps below:

a) Set index the index *i* to an initial value of 0. The initial estimate of ω_i is made by counting the number of zero crossings in the supplied data y_n and normalizing that count by the sampling rate. Perform a pre-fit using the Three Parameter Sine Fit to determine initial estimates of A_0, B_0, and C_0.

b) Increment the index *i* by one for the next iteration.

c) Update the angular frequency

$$\omega_i = \omega_{i-1} + \Delta\omega_{i-1}$$

d) Create the following matricies:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}$$

$$D_i = \begin{bmatrix} \cos(\omega_i t_1) & \sin(\omega_i t_1) & 1 & -A_{i-1} t_1 \sin(\omega_i t_1) + B_{i-1} t_1 \cos(\omega_i t_1) \\ \cos(\omega_i t_2) & \sin(\omega_i t_2) & 1 & -A_{i-1} t_2 \sin(\omega_i t_2) + B_{i-1} t_2 \cos(\omega_i t_2) \\ \vdots & \vdots & \vdots & \vdots \\ \cos(\omega_i t_M) & \sin(\omega_i t_M) & 1 & -A_{i-1} t_M \sin(\omega_i t_M) + B_{i-1} t_M \cos(\omega_i t_M) \end{bmatrix}$$

e) Solve the linear system of equations:

$$X_i = \begin{bmatrix} A_i \\ B_i \\ C_i \\ \Delta\omega_i \end{bmatrix}$$

$$Y = DX_i$$

$$X_i = (D^T D)^{-1}(D^T Y)$$

f) Go to step b and repeat until the changes in A_i, B_i, C_i, and ω_i are suitably small. For our case, a normalized tolerance of less than 10^{-12} is sufficient:

$$\left|\frac{A_i - A_{i-1}}{A_i}\right| \leq 10^{-12}$$

$$\left|\frac{B_i - B_{i-1}}{B_i}\right| \leq 10^{-12}$$

$$\left|\frac{C_i - C_{i-1}}{C_i}\right| \leq 10^{-12}$$

$$\left|\frac{\omega_i - \omega_{i-1}}{\omega_i}\right| \leq 10^{-12}$$

g) Convert the parameters to the form:

$$A\ \cos(\omega_0 t_n + \theta) +\ C$$

Where:

$$A = \sqrt{A_i^2 + B_i^2}$$

$$\theta = \begin{cases} \tan^{-1}\left(-\frac{B_i}{A_i}\right), & for\ A_0 \geq 0 \\ \tan^{-1}\left(-\frac{B_i}{A_i}\right) + \pi, & for\ A_0 < 0 \end{cases}$$

$$C = C_0$$

2.9 Interpolation

2.9.1 Linear Interpolation

Linear interpolation is a method of identifying any arbitrary intermediate data point located between two known data points, assuming that a straight line connects the two points, as shown in the figure below:

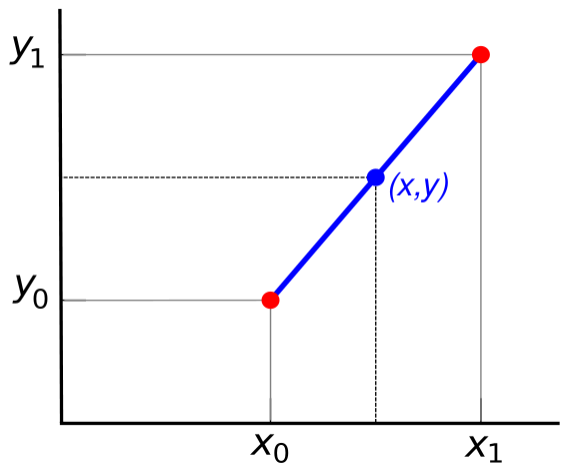

Figure 33 Bilinear Interpolation

The following equation describes the relationship between the two known data points and the intermediate point:

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$$

Solving this equation for *y*, given a known value of *x*, results in the following:

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0}$$

Solving this equation for *x*, given a known value of *y*, results in the following:

$$x = x_0 + (y - y_0)\frac{x_1 - x_0}{y_1 - y_0}$$

2.9.2 Whittaker Interpolation

Whittaker interpolation is a method of reconstructing a bandlimited signal from sampled values of the signal (Oppenheim, 1999, Pg 150). This method of interpolation has the advantage of perfectly reconstructing a bandlimited signal without adding any additional frequency content. However, this result relies upon the assumption that the signals are of infinite duration. Also, the

computational requirements for this method of interpolation are greater than for other simpler methods, especially for longer signals.

Whittaker interpolation may be performed in either the time or frequency domains.

2.9.2.1 Time Domain

In the time domain, Whittaker interpolation is conceptually equivalent to inserting zero values between the existing data points. The effect of inserting the zero values in the frequency domain is to generate spectral copies of the existing frequency content.

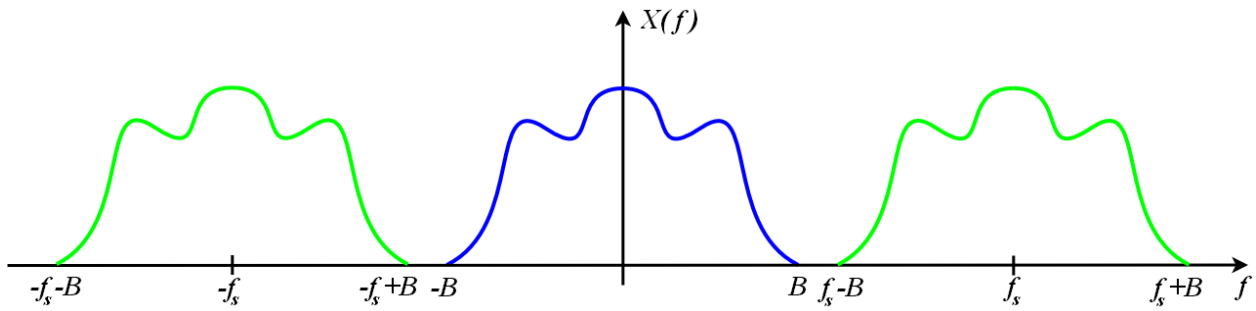

Figure 34 Frequency Domain Spectral Copies

A sinc filter may be applied to low-pass filter the signal and remove the spectral copies.

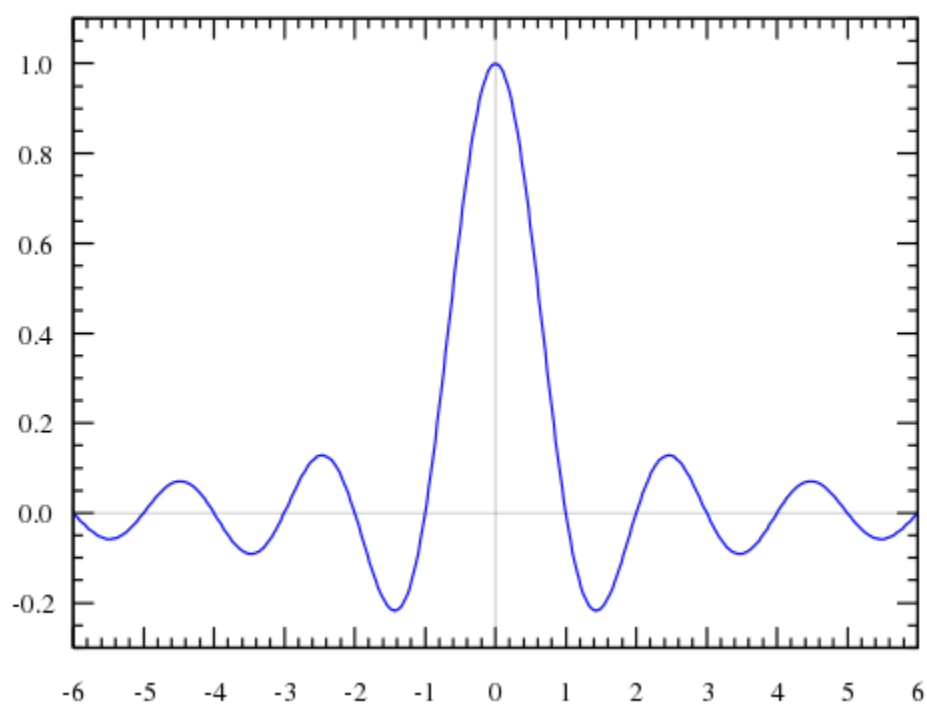

Figure 35 Sinc Filter

The frequency domain equivalent to the time domain sinc filter is shown below.

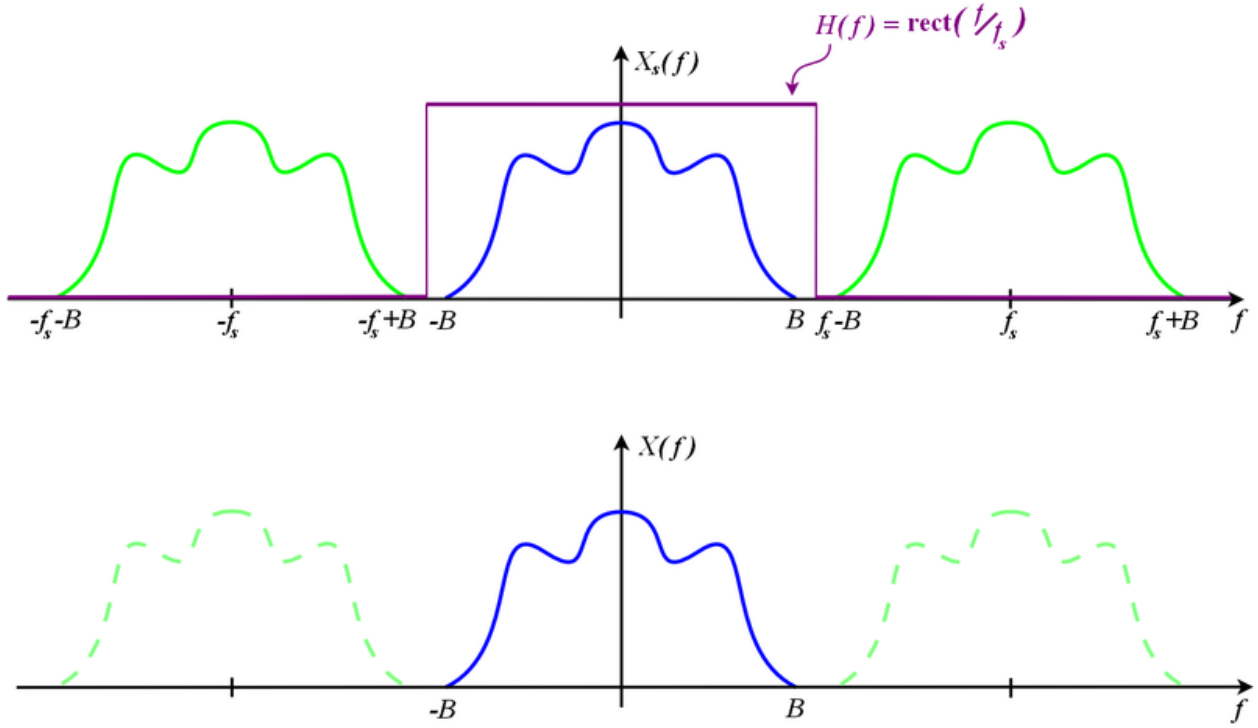

Figure 36 Low-pass Filter to remove spectral copies

For a sequence, $y[k]$, of length N interpolating by a factor of D may be performed by applying the following equation:

$$\tilde{y}[n] = \sum_{k=0}^{N-1} y[k] \frac{\sin\left[\pi\left(\frac{n}{D} - k\right)\right]}{\pi\left(\frac{n}{D} - k\right)}$$

2.9.2.2 Frequency Domain

Whittaker interpolation may also be performed in the frequency domain for increased performance. The two methods (time and frequency domain) are theoretically equivalent to one another, using the Fourier Transform. However, floating point errors may result in slight differences between the two results. In the frequency domain, the interpolation algorithm is shown below.

The sequence of data is zero padded to have a length that is a power of 2 and the Fourier Transform of the data is performed (see 2.2 Discrete Fourier Transform):

$$L = Next\ Power\ of\ 2\ (N)$$

$$Y[k] = FT\{y[n]\}, \qquad 0 \le k, n \le L - 1$$

The Fourier Transformed data has zeros inserted in the middle to simulate an increase in the sampling rate. This is conceptually equivalent to inserting zeros between samples and performing an idealized sinc filter in the time domain.

$$\tilde{Y}[k] = \begin{cases} Y[k] & , 0 \leq k \leq \frac{L}{2} - 1 \\ 0 & , otherwise \\ Y[k + L - L * D] & , L * D - \frac{L}{2} \leq k \leq D * L - 1 \end{cases}$$

Finally, the inverse Fourier Transform (IFT) is performed:

$$\tilde{y}[n] = IFT\{\tilde{Y}[k]\}, \qquad 0 \leq k, n \leq D * L - 1$$

2.10 Time Tag Measurements

Time Tag measurements are made by recording on a digitizer the on-the-minute or hour analog pulse triggers that can be outputted from a GPS receiver. The transition time of such pulses from a typical GPS receiver have a timing error that is less than 1 microsecond. By comparing the digitizers claimed time-tag for the trigger to the expected minute or hour transition, the amount of timing error can be computed. Therefore, such timing triggers can be used to measure the amount of timing error in the recorded signal of the digitizer.

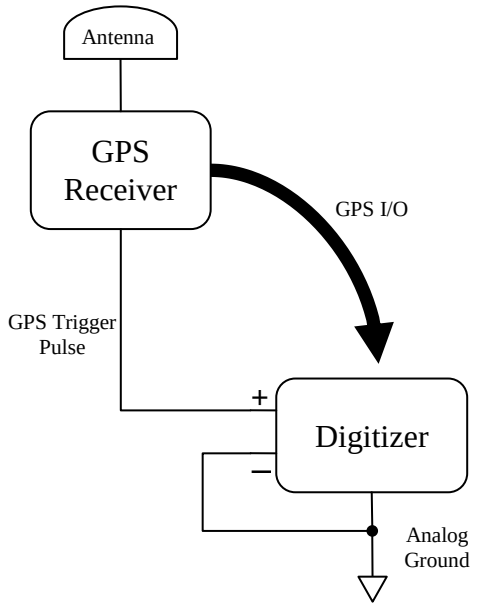

Figure 37 GPS Timing System Diagram

An important consideration is to recognize that the digitizer time tag errors (generally on the order of 10's of microseconds) that are estimated using this technique do not represent an absolute accuracy value. The estimated error values can vary considerably (by 100's of microseconds) simply by modifying the methods used for estimating the various parameters used. There is no suitable justification for why a given result is more accurate than another. However, so long as the estimation parameters are held constant, then the time tag error values that are estimated are consistent with one another. This relative accuracy of the error results is crucial for the application of the time tag error measurements when examining quantities such as the sample-to-sample jitter of the digitizer or the digitizer's clock drift and recover when GPS lock has been lost.

In addition, waveform time stamps are generally stored in a double precision floating point number as seconds since January 1, 1970. For the range of time values observed at the time this document was written, this numeric format restricts the resolution of the timestamp to approximately 0.2 microseconds.

First, to define terms related to the rising edge of the trigger pulse, the figure below (IEEE STD181-2003, Pg. 37) represents such a pulse.

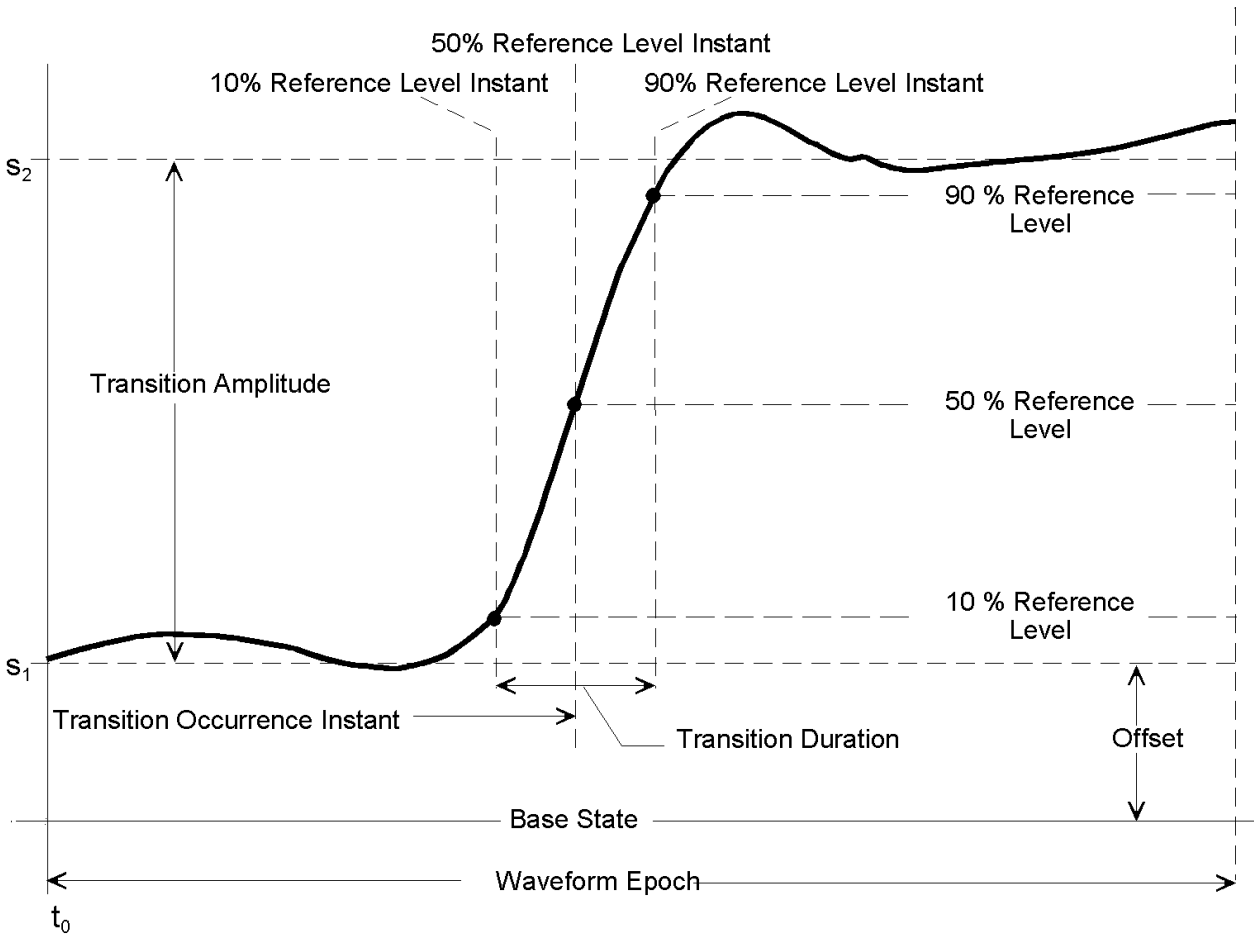

Figure 38 Timing Trigger Definition

A trigger has a low baseline level prior to the pulse represented by S1 and a high level baseline level after the pulse represented by S2. The mean point of the trigger is half-way between S1 and S2 and is the 50 % Reference Level. The 50% Reference Level occurs at the 50% Reference Level Instant, or trigger time. The trigger time, as identified by the digitizer's time tag, should occur on the appropriate time transition (minute or hour). Any difference between the measured trigger time and expected trigger time is defined to be the digitizers timing error for that pulse.

In order to measure the timing error, there are several steps that must be taken in the analysis:

1. Identify the rising transition region.

   The first step in measuring timing error is to identify the trigger pulse. Identification of the time period encompassing the trigger pulse is performed by examining the first difference, or numeric derivative, of the time series. The transition region is defined as being anywhere that the first difference is above a threshold of 0.001 Volts. This threshold was selected to ensure that it was greater than the sample-to-sample variability observed in the low and high baseline flat regions being outputted from the timing source. In addition, the transition region is expanded to include other transition regions that are within 100 time samples. This merging of the transition regions is performed to

ensure that there are valid and adequately long low and high level baselines both before and after the trigger, respectively.

To illustrate, the pulse below shows a triggers with the transition regions colored blue and the non-transition, or flat, regions colored magenta.

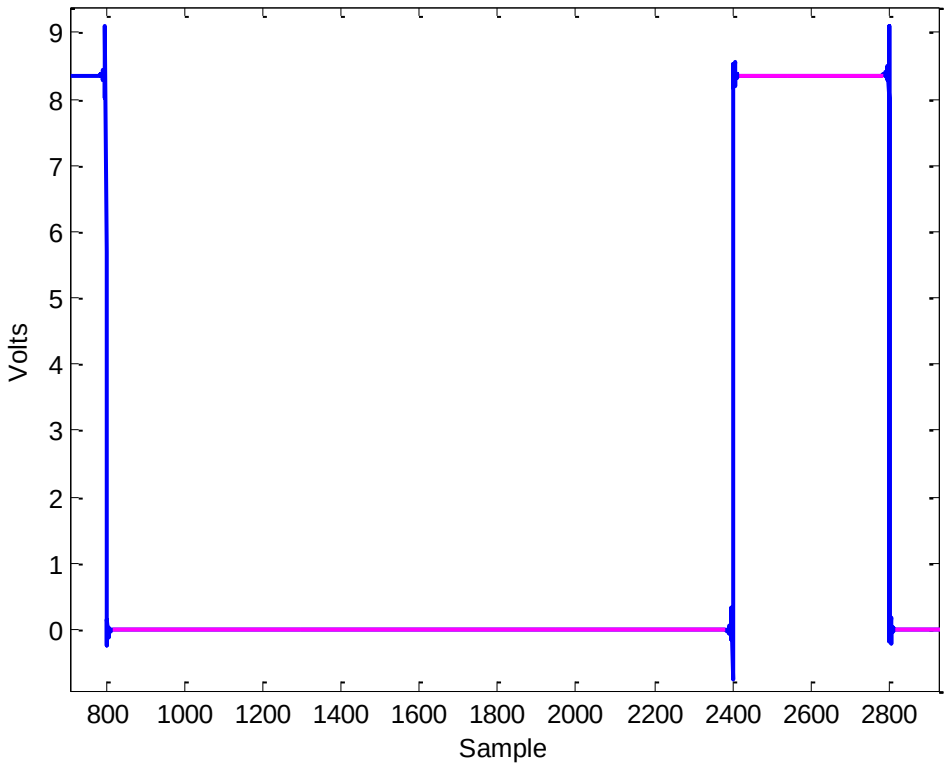

Figure 39 Example Timing Trigger

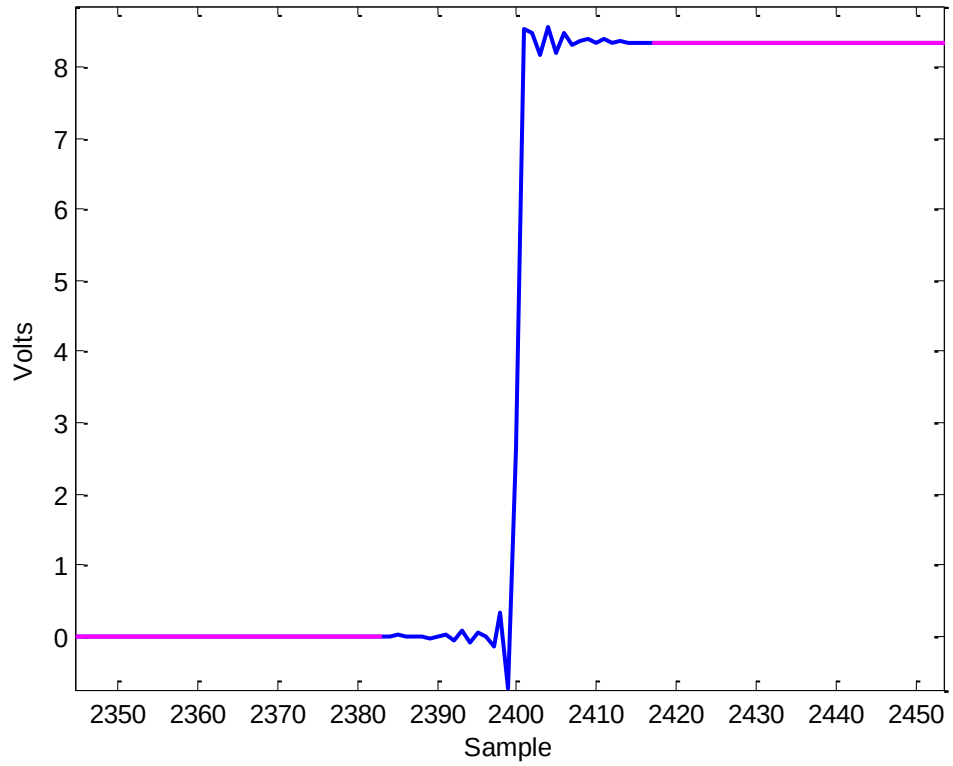

Figure 40 Example Timing Trigger

2. Estimate the low and high baseline levels that occur before and after the pulse (defined within the transition region).

   The low baseline level is identified by examining the first difference starting 5 samples prior to the start of the transition region and looking backwards up to 500 samples. The same threshold of 0.001 Volts is applied to ensure that the low baseline level is flat.

   The high baseline level is identified by examining the first difference starting 5 samples after the end of the transition region and looking forwards up to 500 samples. The same threshold of 0.001 Volts is applied to ensure that the high baseline level is flat.

   It is highly desirable to obtain a large time window for the low and high baseline levels in order to compute an accurate estimate of the low and high baseline levels. Even though the baselines appear to be flat, looking at the baselines more closely reveals that the GPS timing source contains some amount of variability. The plots below reveal the baseline regions to have a peak-to-peak range of approximately 0.001 Volts, which is also the threshold level that was chosen for the sample-to-sample differencing. In order to ensure the most accurate averaging, the baseline windows are made as long as possible before the sample-to-sample difference exceeds the defined threshold.

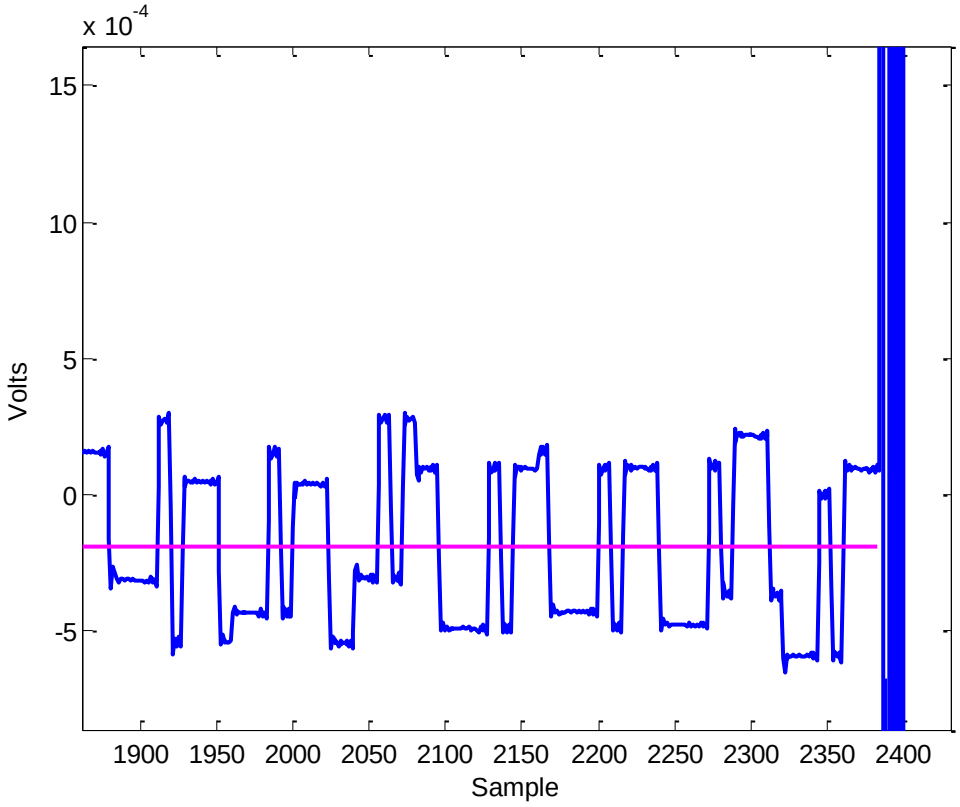

Figure 41 Low baseline level

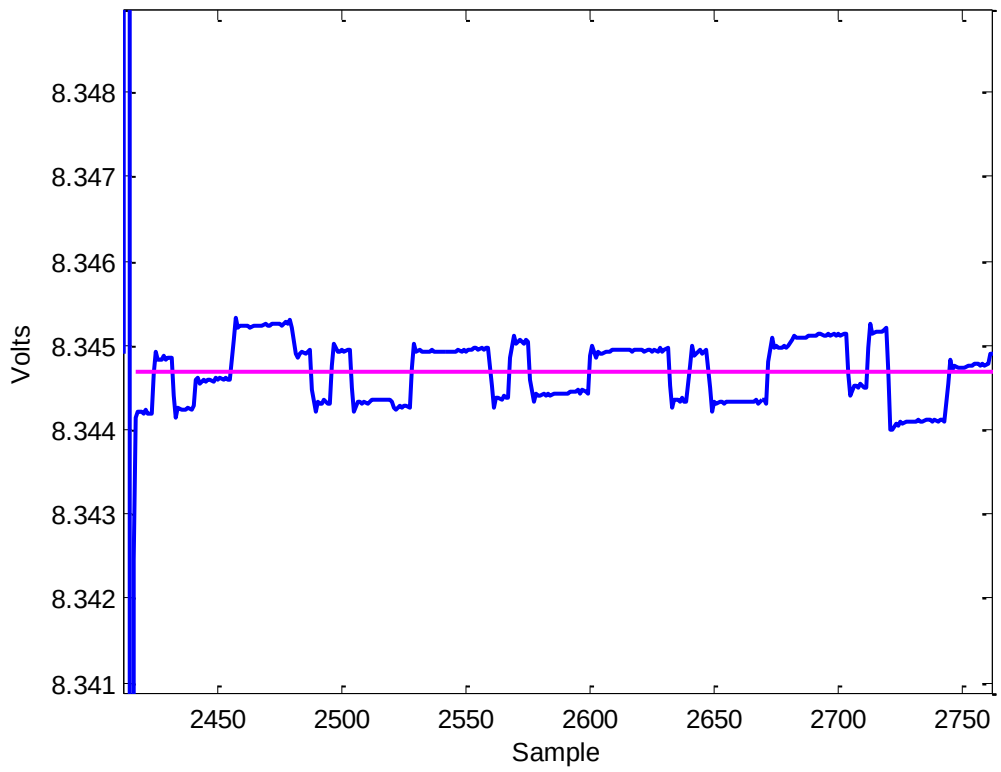

Figure 42 High baseline level

3. Compute the 50% Reference Level as being the average of the low and high baseline levels.

   The estimates of the low and high baseline levels can have a dramatic affect on the measured timing error. Any variability in these values will directly affect the 50 % Reference Level and thus shift the time instant at which the 50 % Reference Level is crossed.

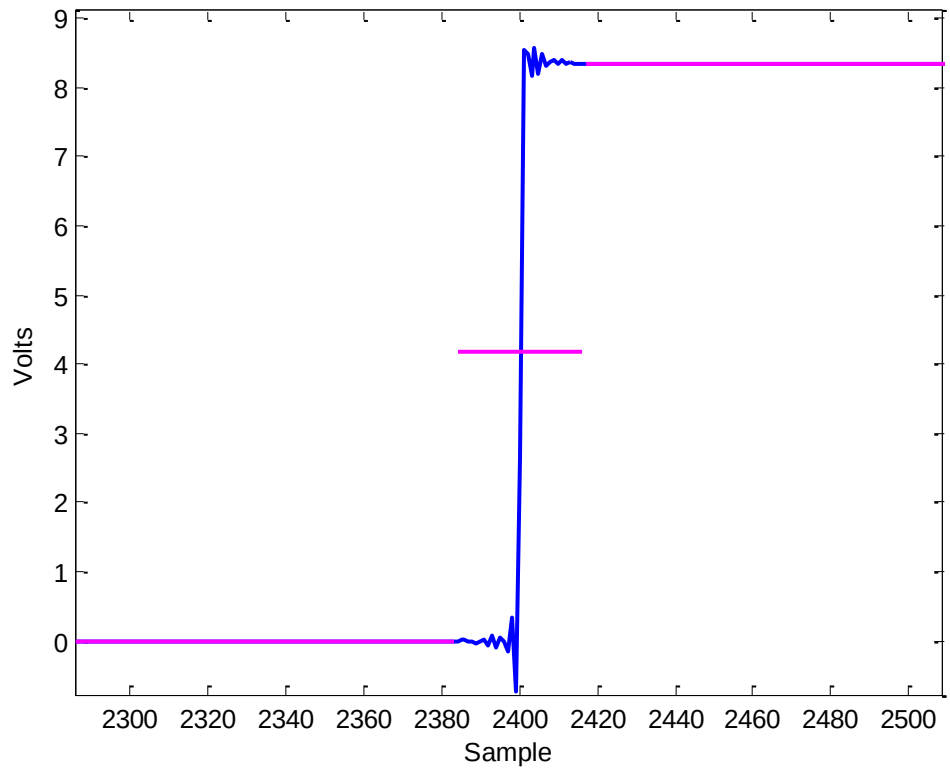

Figure 43 50% Reference Level

4. Perform a reconstruction of the rising transition region of the pulse (see 2.9.2 Whittaker Interpolation).

   The window of data that is interpolated is selected to be a total of 10 seconds in length. The window starts 5 seconds prior to the theoretical trigger time and extends 5 seconds past the trigger time. The data samples are interpolated such that there is at least one interpolation sample every millisecond.

   In the case of a timing pulse that is sampled at 40 Hz by the digitizer, a data sample is recorded every 25 milliseconds. The number of data samples included in the interpolation would be 400, with 200 prior to the trigger time and 200 after the trigger time. The interpolation would be performed to achieve an interpolated sample spacing of at least 1 millisecond. For performance concerns when performing the interpolation in the frequency domain, a factor of two interpolation scaling factor must be used. For this example, the interpolation scale factor would be 32. This method of timing estimation is intended for timing errors that are on the order of 10's of microseconds. This represents a timing error of roughly 1 part in 1000. Although there may be discrepancies in the absolute level of timing error for a given timing pulse, so long as consistent methods and parameters are used in estimating the timing errors, it is possible to make relative comparisons between the error measurements.

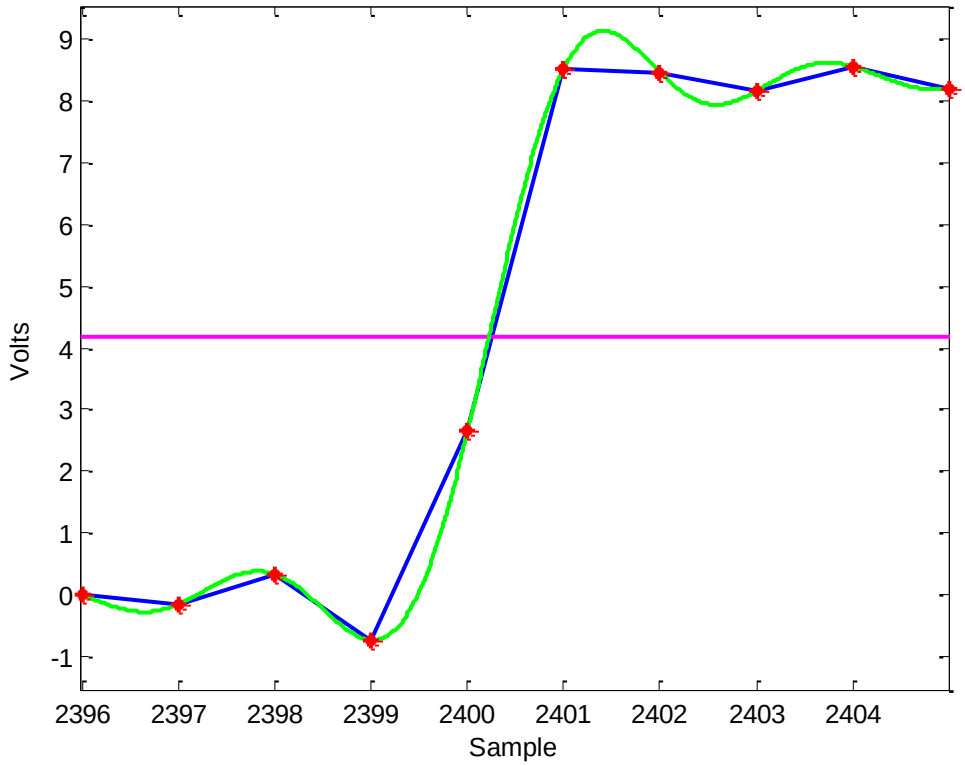

Figure 44 Timing Trigger Whittaker Interpolation

5. Identify the two points on the interpolated trigger that bound the 50 % Reference Level.

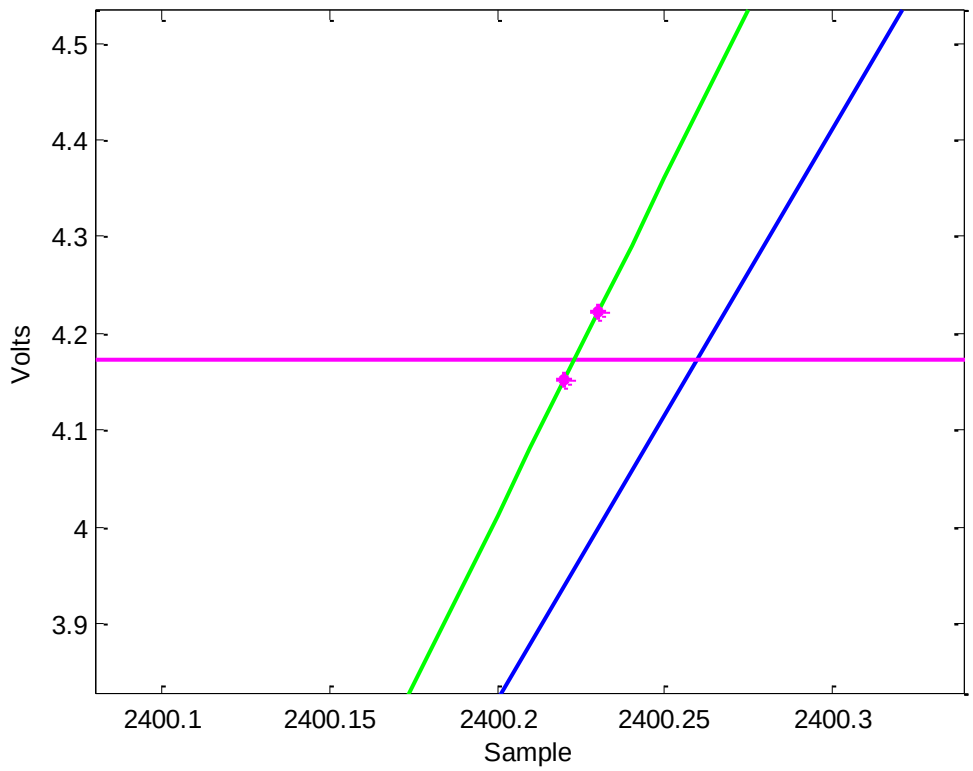

Figure 45 Timing Trigger nearest points

6. Perform a linear interpolation between those two points to identify the 50 % Reference Level Instance (see 2.9.1 Linear Interpolation).

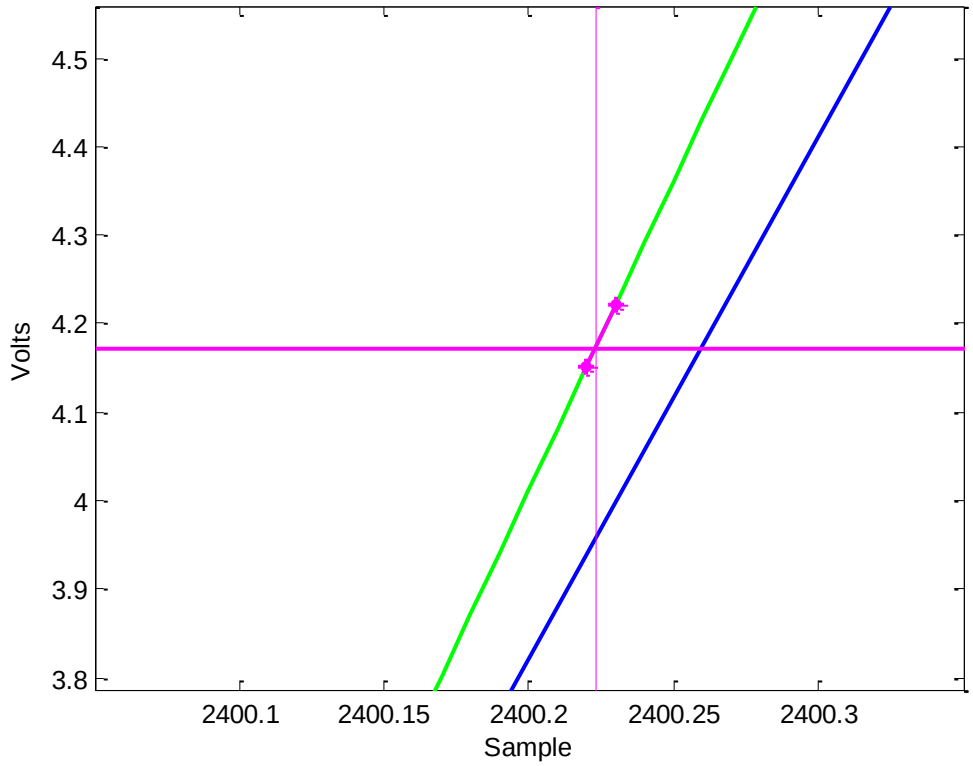

Figure 46 Timing Trigger bilinear interpolation

7. The timing error is then taken to be the difference in time between the 50 % Reference Level Instance and the minute or hour transition.

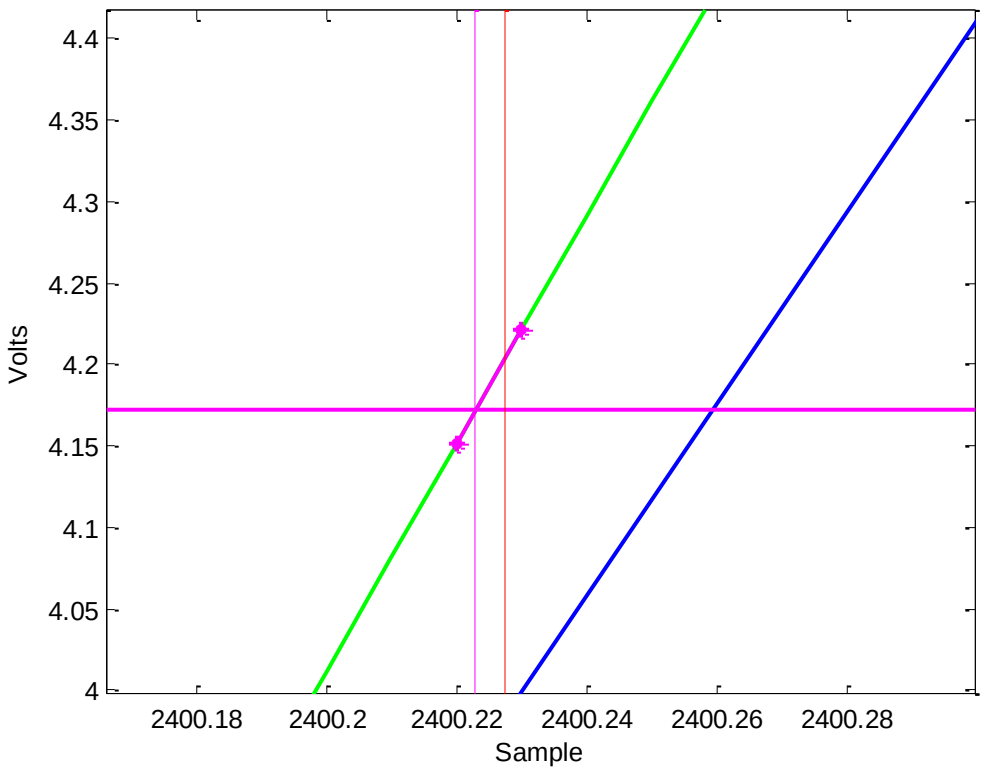

Figure 47 Timing Trigger Error

3 TEST ANALYSIS

This section describes the analysis that is performed for each of the test modules for the Component Evaluation project. For a general description of the tests and their procedures, see the appropriate Digitizer, Seismic, or Acoustic test definition document (Kromer, 2007).

Each test module section provides an overview of the analysis that includes equations and plots where relevant. In addition, tables are included at the end of each section listing the relevant waveforms to be collected, user defined input parameters, and the critical results of the analysis.

3.1 Generic Component Tests

Generic component tests are tests that can be applied to any component, regardless of its type (digitizer, seismometer, infrasound sensor, etc).

3.1.3 Power Consumption

The Power Consumption test is used to measure the amount of power that an actively powered electrical component consumes during its operation.

The power supply voltage level is first quantified by connecting the power supply to a meter and collecting a time series of the voltage value. This measurement is made prior to connecting the electrical component to the meter.

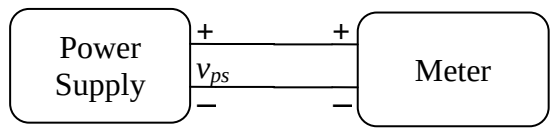

Figure 48 Power Supply Calibration Diagram

$$v_{ps}[n], \qquad 0 \leq n \leq N-1$$

Then, the power supply is disconnected from the meter and connected to the resistor and component as shown in the diagram below. The voltage across the resistor is measured with the meter.

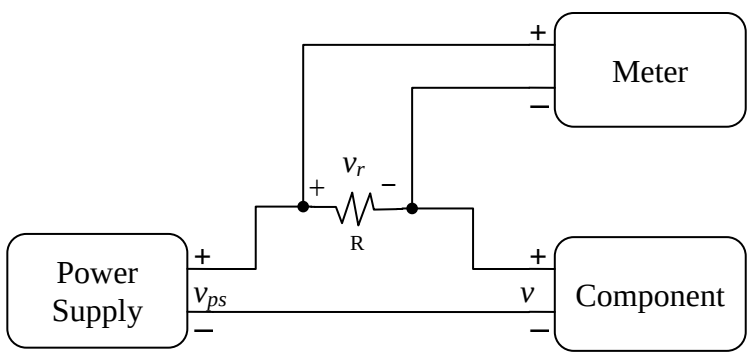

Figure 49 Power Consumption Diagram

The voltage is sampled for the same duration as when collecting the power supply voltage values so as to obtain equivalent statistics for the two quantities. Note that it is assumed that the power supply is of sufficient quality and capacity that its output voltage values do not change significantly when the electrical component's load is placed upon it.

$$v_r[n], \qquad 0 \leq n \leq N-1$$

Even though $v_{ps}, v_r,$ and v are not independent, they are related to one another by the resistor and the load impedance of the component being tested. This load impedance is unknown and may not even be purely resistive. Therefore, sample estimates for the mean and standard deviations are obtained from the collected power supply and resistor voltage time series:

$$\mu_{v_{ps}} = \frac{1}{N}\sum_{n=0}^{N-1} v_{ps}[n]$$

$$\mu_{v_r} = \frac{1}{N}\sum_{n=0}^{N-1} v_r[n]$$

$$\sigma_{v_{ps}} = \sqrt{\frac{1}{N-1}\sum_{n=0}^{N-1}\left(v_{ps}[n] - \mu_{v_{ps}}\right)^2}$$

$$\sigma_{v_r} = \sqrt{\frac{1}{N-1}\sum_{n=0}^{N-1}\left(v_r[n] - \mu_{v_r}\right)^2}$$

The average current through the resistor with resistance R is estimated using Ohm's law (Thomas, 1994, Pg. 25):

$$\mu_i = \frac{\mu_{v_r}}{R}$$

The current standard deviation may be estimated as a scaling of the resistor voltage standard deviation (need a reference here):

$$\sigma_i = \frac{\sigma_{v_r}}{R}$$

The average voltage across the electrical component may be estimated using Kirchoff's Voltage Law (Thomas, 1994, pg. 34-35):

$$\mu_v = \mu_{v_{ps}} - \mu_{v_r}$$

The standard deviation of the voltage across the electrical component may be estimated by treating it as the sum of two independent random variables:

$$\sigma_v = \sqrt{\sigma_{v_{ps}}^2 + \sigma_{v_r}^2}$$

Finally, the average power consumption of the electrical component may be estimated by employing the equation for electrical power (Thomas, 1994, Pg. 9):

$$\mu_P = \mu_v \cdot \mu_i$$

The standard deviation of the power consumption is somewhat more complicated as it must be modeled as the product of two independent random variables:

$$\sigma_P = \sqrt{\sigma_v^2 \sigma_i^2 + \sigma_v^2 \mu_i^2 + \mu_v^2 \sigma_i^2}$$

Table 1: Test Waveforms

Waveform	**Unit**	**Source**
Power Supply Waveform	Volt	Meter
Resistor Waveform	Volt	Meter

Table 2: Test Parameters

Parameter	**Unit**
Resistance	Ohm

Table 3: Test Results

Result	**Unit**
Power Consumption Mean	Watt
Power Consumption Standard Deviation	Watt

3.2 Digital Waveform Recorder Tests

3.2.3 AC Accuracy

The AC Accuracy test is used to measure the DC offset and bitweight of a digitizer channel by recording a sinusoid with known frequency and amplitude of the form:

$$V_{ref}\sin(\omega_0 t_n)$$

The function generator is connected to the digitizer as shown in the block diagram below:

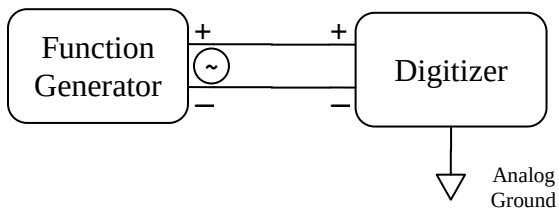

Figure 50 AC Accuracy Diagram

Compute a Sine Fit (see 2.8 Sine Fit) on the recorded data to determine the measured amplitude, frequency, and DC offset.

$$V_{meas}\sin(\omega_0 t_n) + V_{dc}$$

Compute the DC offset, ensuring that it is expressed as an integer multiple of the bitweight:

$$DC\ Offset = round\left(\frac{V_{dc}}{bitweight}\right) * bitweight$$

Compute the corrected bitweight:

$$Corrected\ Bitweight = \frac{V_{meas}}{V_{ref}} * Bitweight$$

Compute the % error in the bitweight:

$$\%\ Error = 100 * \frac{Corrected\ Bitweight - Bitweight}{Corrected\ Bitweight}$$

Table 4: Test Waveforms

Waveform	Unit	Source
Sinusoid Waveform	Volt	DUT

Table 5: Test Parameters

Parameter	Unit

Reference Voltage	Volt
Reference Frequency	Hz

Table 6: Test Results

Result	**Unit**
Sinusoid Amplitude	Volt
DC Offset	Volt
Corrected Bitweight	Volts/count
Percent Error	%
Sine Fit Error	Volt rms

3.2.4 AC Clip

The AC Clip test measures the digitizer clip level and behavior by feeding a sinusoid signal into a channel of the digitizer. The amplitude of the sinusoid is set to a level slightly below (no less than 10%) the digitizer clip level. The function generator is connected to the digitizer as shown in the block diagram below:

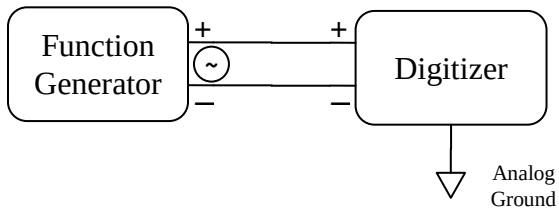

Figure 51 AC Clip Diagram

Warning, applying a voltage input greater than digitizer's full scale voltage level may result in irreparable damage to the digitizer. Therefore, the amplitude is slowly increased while monitoring the digitizers output. When the sinusoid begins to flatten, proceed to increase the amplitude until both the peak and trough of the sinusoid flatten. Allow 5 – 10 cycles to be recorded and then decrease the amplitude of the sinusoid to below the clip level.

The waveform is loaded and displayed to the tester. Only a few cycles of the clipped sinusoid are necessary. The digitizer behavior is observed by inspection whether it flattens out, ripples, wraps around between positive and negative, etc.

The tester defines the waveform segments that represent the positive and negative clip regions of the sinusoid:

$$x_{pos}[n], \qquad 0 \le n \le N-1$$

$$x_{neg}[n], \qquad 0 \le n \le N-1$$

Using these segments, the positive and negative mean and standard deviation are computed:

$$\mu_{pos} = \frac{1}{N}\sum_{n=0}^{N-1} x_{pos}[n]$$

$$\mu_{neg} = \frac{1}{N}\sum_{n=0}^{N-1} x_{neg}[n]$$

$$\sigma_{pos} = \sqrt{\frac{1}{N-1}\sum_{n=0}^{N-1}\left(x_{pos}[n] - \mu_{pos}\right)^2}$$

$$\sigma_{neg} = \sqrt{\frac{1}{N-1}\sum_{n=0}^{N-1}\left(x_{neg}[n] - \mu_{neg}\right)^2}$$

The tester qualitatively determines whether the clipping behavior has passed.

Table 7: Test Waveforms

Waveform	**Unit**	**Source**
Sinusoid Waveform	Volt	DUT

Table 8: Test Parameters

Parameter	**Unit**
Digitizer Clip Level	Volt
Sinusoid Amplitude	Volt
Sinusoid Frequency	Hz

Table 9: Test Results

Result	**Unit**
Positive Mean	Volt
Positive Standard Deviation	Volt
Negative Mean	Volt
Negative Standard Deviation	Volt
Passed	boolean

3.2.5 Analog Bandwidth

The Analog Bandwidth test measures the -3 dB roll-off point of the digitizer in order to estimate its bandwidth. This test is performed by feeding white noise from a function generator into one or more of the digitizer channels as shown in the block diagram below.

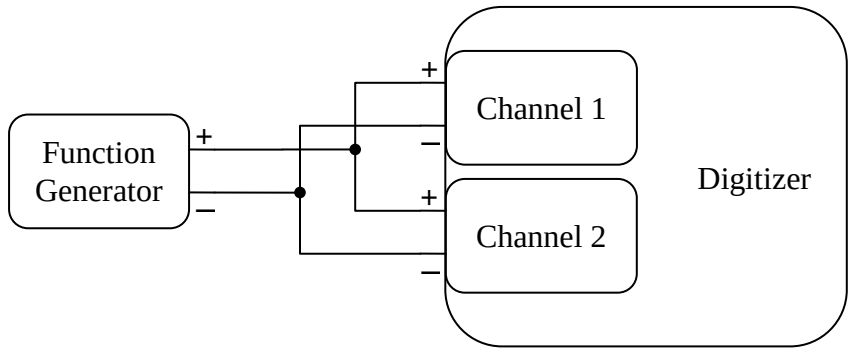

Figure 52 Analog Bandwidth Diagram

The first step in the test analysis is to compute the power spectra using a Hann window and 5/8 overlap (see 2.4 Power Spectral Density) of the white noise channels:

$$P[k], \qquad 0 \le k \le N-1$$

Next, the power spectra are smoothed by averaging over an odd number of frequency bins, M, specified by the tester. The smoothing is performed to ensure that the estimation of the -3 dB point is more accurate. Otherwise, the estimate may be overly impacted by random variability in the power spectra.

$$P_{smooth_i}[k] = \sum_{i=ceil\left(k-\frac{M}{2}\right)}^{floor\left(k+\frac{M}{2}\right)} P_i[k], \qquad 0 \le i \le N-1$$

For each of the waveforms, a reference frequency is selected that is equal to ½ of the Nyquist rate.

$$k_{ref\ i} = \frac{k_{nyquist\ i}}{2}, \qquad 0 \le i \le N-1$$

The -3 dB, or ½, value is computed for each of the channels:

$$P_{3db\ i} = \frac{Psmooth_i\left[k_{ref\ i}\right]}{2}, \qquad 0 \le i \le N-1$$

The -3 dB frequency is computed for each of the channels:

$$k_{3db\,i} = k\ for\ which\ \left|P_{3db\,i} - Psmooth_i\left[k_{ref\,i}\right]\right|\ is\ minimized$$

Finally, the power and attenuation (relative to the reference frequency) at the Nyquist rate are computed:

$$P_{nyquist\,i} = P\left[k_{nyquist\,i}\right], \qquad 0 \le i \le N-1$$

$$attenutation = 10\ \log_{10}\left(\frac{P_{nyquist\,i}}{Psmooth_i\left[k_{ref\,i}\right]}\right), \qquad 0 \le i \le N-1$$

Table 10: Test Waveforms

Waveform	**Unit**	**Source**
White Noise Waveforms	Volt	DUT

Table 11: Test Parameters

Parameter	**Unit**
PSD Parameters	
White Noise Amplitude	Volt
White Noise Bandwidth	Hz
Number of smoothing bins	

Table 12: Test Results

Result	**Unit**
Reference Frequency	Hz
Reference PSD	dB (V^2/Hz)
-3 dB Frequency	Hz
-3 dB PSD	dB (V^2/Hz)
Nyquist PSD	dB (V^2/Hz)
Attenuation at Nyquist	dB

3.2.6 Common Mode Rejection Ratio

The Common Mode Rejection Ratio test measures the ability of a digitizer to reject a common mode signal on a differential input channel. For this test, a function generator is used to generate a sinusoid with known frequency and amplitude of the form:

$$V_{ref}\sin(2\pi f_0 t_n)$$

The function generator is connected to the digitizer as shown in the block diagram below:

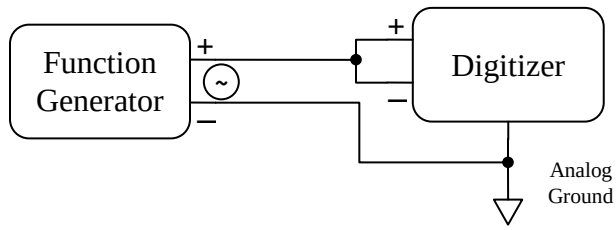

Figure 53 Common Mode Rejection Ratio Diagram

The positive and negative lines on the digitizer input channel are shorted as close to the digitizer as possible to minimize the introduction of any possible non-common mode noise from interference. The positive terminal of the function generator is connected to the shorted digitizer input channel. The negative terminal of the function generator is connected to the digitizer's analog ground.

The waveform, in volts, from the digitizer under test is collected:

$$x[t], 0 \leq t \leq N-1$$

Since the digitizer input channels are differential and are shorted together, the digitizer should not be recording any signal. However, some amount of common mode signal will still be present on the digitizer input channel. To measure the amount of residual common mode signal, compute a Sine Fit (see 2.8 Sine Fit) on the recorded data to determine the measured amplitude, frequency, and DC offset:

$$V_{meas}\sin(2\pi f_0 t_n) + V_{dc}$$

The Common Mode Rejection Ratio, in dB, is then:

$$CMR_{dB} = 10 * \log_{10}\left(\frac{V_{ref}}{V_{meas}}\right)^2$$

Table 13: Test Waveforms

Waveform	Unit	Source
Common Mode Waveform	Volt	DUT

Table 14: Test Parameters

Parameter	Unit
Reference Frequency	Hz
Reference Amplitude	Volts (peak)

Table 15: Test Results

Result	Unit
Measured Amplitude	Volts (peak)
CMR Ratio	dB

3.2.7 Crosstalk

The Crosstalk test measures how much of a signal recorded on one channel of a digitizer is also present on another channel as noise. For this test, a function generator is used to generate a sinusoid with known frequency and amplitude of the form:

$$V_{ref} \sin(2\pi f_0 t_n)$$

One of the channels is terminated with a resistor, typically 50 Ohms, and the remaining digitizer channels are connected to the function generator as shown in the block diagram below:

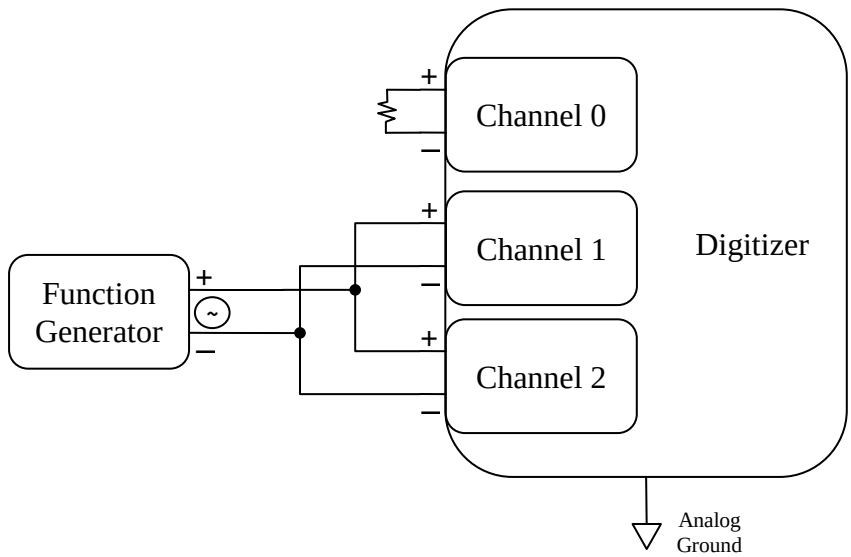

Figure 54 Crosstalk Diagram

Different configurations of the input sinusoid and the terminating resistor can be used to test either the best case or worst case levels of cross talk. The configuration represented here represents the worst case configuration.

Typically, this test will be repeated a sufficient number of times to allow for each channel on the digitizer to be input terminated while the remaining channels are fed a tonal signal.

The first step in the test analysis is to compute the power spectra using a Hann window (see 2.3.2 Hann) and 5/8 overlap (see 2.4 Power Spectral Density) of the input terminated channel and all of the tonal channels:

$$P_i[k], \qquad 0 \leq i \leq N-1$$

For the purposes of convention, the resistor terminated channel is assumed to be the 0th channel and the tonal channels are 1 through N-1. The frequency and RMS value of the maximum peak in each of the power spectra are identified and computed as described in 3.2.22 Total Harmonic Distortion:

$$f_{\max i}, \qquad 0 \le i \le N-1$$
$$V_{rms\,i}, \qquad 0 \le i \le N-1$$

The crosstalk (expressed in dB) between the ITN channel and each of the tonal channels is computed:

$$Crosstalk_i = 10\ \log_{10}\left(\frac{V_{rms\,0}}{V_{rms\,i}}\right)^2, \qquad 1 \le i \le N-1$$

The mean crosstalk value is also computed:

$$Mean\ Crosstalk = 10\ \log_{10}\left[\frac{1}{N-1}\sum_{i=1}^{N-1}\frac{V_{rms\,0}}{V_{rms\,i}}\right]^2$$

Table 16: Test Waveforms

Waveform	**Unit**	**Source**
Input Terminated Waveform	Volt	DUT
Tonal Waveforms	Volt	DUT

Table 17: Test Parameters

Parameter	**Unit**
PSD Parameters	
Terminator Resistance	Ohms
Signal Voltage	Volt
Signal Frequency	Hz

Table 18: Test Results

Result	**Unit**
Peak Frequencies (one per channel)	Hz
RMS of each peak (one per channel)	Volt RMS
Crosstalk ratio(ratio between ITN and each Tonal channel)	dB
Mean Crosstalk ratio	dB

3.2.8 DC Accuracy

The DC Accuracy test is used to measure the DC offset and bitweight of a digitizer channel by recording a known positive and negative dc signal at a reference voltage from a precision voltage source:

$$V_{ref}$$

There are two waveform time series associated with the DC Accuracy test. The positive waveform time series, $x_{pos}[n]$, is at a DC voltage of:

$$V_{ref}$$

To collect the positive waveform time series, the voltage source is connected to the digitizer as shown in the block diagram below.

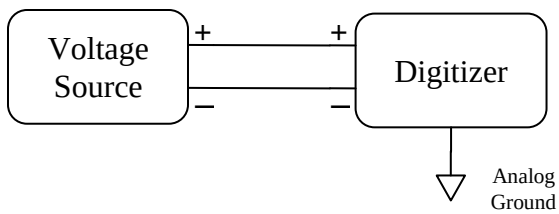

Figure 55 DC Accuracy Positive Measurement Diagram

The negative time series, $x_{neg}[n]$, is at a DC voltage of:

$$-V_{ref}$$

To collect the negative waveform time series, the voltage source is connected to the digitizer with its polarity reversed as shown in the block diagram below.

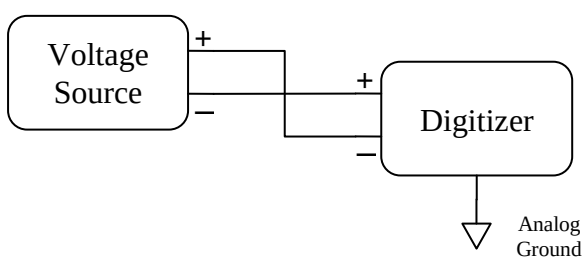

Figure 56 DC Accuracy Negative Measurement Diagram

The positive and negative waveform time series segments should be of equal length so that estimates of mean and standard deviation are statistically equivalent.

Compute the mean and standard deviation for both the positive and negative segments where N is the number of data samples:

$$\mu_{pos} = \frac{1}{N}\sum_{n=0}^{N-1} x_{pos}[n]$$

$$\mu_{neg} = \frac{1}{N}\sum_{n=0}^{N-1} x_{neg}[n]$$

$$\sigma_{pos} = \sqrt{\frac{1}{N-1}\sum_{n=0}^{N-1}\left(x_{pos}[n] - \mu_{pos}\right)^2}$$

$$\sigma_{neg} = \sqrt{\frac{1}{N-1}\sum_{n=0}^{N-1}\left(x_{neg}[n] - \mu_{neg}\right)^2}$$

Compute the range between the positive and negative voltages:

$$reference\ range = 2 * V_{ref}$$
$$range = \mu_{pos} - \mu_{neg}$$

Compute the DC offset, ensuring that it is expressed as an integer multiple of the bitweight:

$$DC\ Offset = round\left(\frac{\mu_{pos} + \mu_{neg}}{2 * Bitweight}\right) * Bitweight$$

Compute the corrected bitweight:

$$Corrected\ Bitweight = \frac{range}{reference\ range} * Bitweight$$

Compute the % error:

$$\%\ Error = 100 * \frac{range - reference\ range}{range}$$

Table 19: Test Waveforms

Waveform	**Unit**	**Source**
Positive DC Waveform	Volt	DUT
Negative DC Waveform	Volt	DUT

Table 20: Test Parameters

Parameter	**Unit**

Reference Voltage	Volt

Table 21: Test Results

Result	**Unit**
Positive Mean	Volt
Positive Standard Deviation	Volt
Negative Mean	Volt
Negative Standard Deviation	Volt
Voltage Range	Volt
DC Offset	Volt
Corrected Bitweight	Volts/count
Percent Error	%

3.2.9 Infrasound System Noise & Bandwidth Limited Dynamic Range

The Infrasound System Noise test measures the digitizer's noise and dynamic range expressed in the units of an infrasound sensor. The test is performed identically to the Input Terminated Noise and Maximum Potential Dynamic Range test (see 3.2.12).

The only difference comes in the analysis. An appropriate response is selected for the digitizer and a response corrected PSD is computed (see 2.4 Power Spectral Density and 2.5.1.5 Removing a response from digital time series data):

$$P_{yy}[k], 0 \leq k \leq N-1$$

Over frequencies (in Hertz):

$$f[k], 0 \leq k \leq N-1$$

Compute the rms value over a user defined frequency band of the PSD (see 2.4.2 RMS Estimates):

$$Pressure_{rms}$$

Estimate the Bandwidth Limited Dynamic Range (BLDR) for over the frequency band, using a user defined value for the sensor peak output (see 2.4.3 Dynamic Range Estimation):

$$BLDR = 10\ \log_{10}\left(\frac{Pressure_{peak}/\sqrt{2}}{Pressure_{rms}}\right)^2$$

Table 22: Test Waveforms

Waveform	Unit	Source
Test Infrasound Waveform	Pascal	DUT

Table 23: Test Parameters

Parameter	Unit
Frequency(s)	Hz
Peak Sensor Output	Pascal

Table 24: Test Results

Result	Unit
Noise RMS	Pascal
BLDR	Decibel

3.2.10 Input Impedance

The Input Impedance test measures the input impedance of a digitizer channel. A meter is configured to measure impedance and is connected to the digitizers input channel.

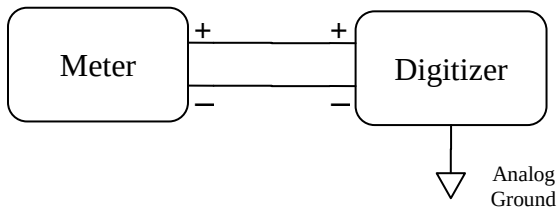

Figure 57 Input Impedance Diagram

The waveform, in ohms, is collected from the meter and analyzed.

$$x[t], \qquad 0 \leq t \leq N-1$$

The average of the waveform time series is estimated to be the digitizers input impedance

$$\mu_{ohms} = \frac{1}{N}\sum_{n=0}^{N-1} x[n]$$

Table 25: Test Waveforms

Waveform	Unit	Source
Input Shorted Waveform	Ohm	Meter

Table 26: Test Parameters

Parameter	Unit
None	

Table 27: Test Results

Result	Unit
Input Impedance	Ohm

3.2.11 Input Shorted Offset

The Input Shorted Offset test measures the amount of DC offset present on a digitizer by collecting waveform data from an input channel that has been shorted. Thus, any signal present on the recorded waveform should be solely due to any internal offset of the digitizer.

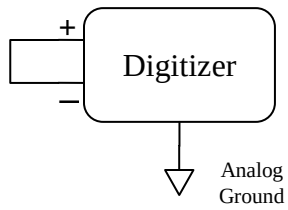

Figure 58 Input Shorted Offset Diagram

The waveforms, in volts, from the sensor under test are collected and analyzed.

$$x[t], \qquad 0 \leq t \leq N-1$$

The average of the waveform time series is estimated to be the digitizers internal offset

$$\mu_{offset} = \frac{1}{N}\sum_{n=0}^{N-1} x[n]$$

Table 28: Test Waveforms

Waveform	Unit	Source
Input Shorted Waveform	Volt	DUT

Table 29: Test Parameters

Parameter	Unit
None	

Table 30: Test Results

Result	Unit
DC Offset	Volt

3.2.12 Input Terminated Noise and Maximum Potential Dynamic Range

The Input Terminated Noise test measures the amount of zero state noise present on a digitizer by collecting waveform data from an input channel that has been terminated with a resistor. Thus, any signal present on the recorded waveform should be solely due to noise generated internally by the digitizer. The resistor should be chosen to match the load impedance of a known sensor. The resistance, R, is often in the range of 10 – 100 Ohms. However, the resistance value can also be as low as 0 ohms, representing a shorted connection.

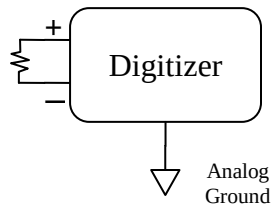

Figure 59 Input Terminated Noise Diagram

The tester specifies a frequency pass-band over which the RMS voltage and dynamic range will be computed.

In addition, the tester specifies what the full scale (peak) voltage is for this configuration of the digitizer. This peak voltage, V_{peak}, is used to estimate the dynamic range over specified frequency bands.

First, compute the power spectra over the waveform segment (see 2.4 Power Spectral Density):

$$P_{xx}[k], 0 \leq k \leq N - 1$$

Over frequencies (in Hertz):

$$f[k], 0 \leq k \leq N - 1$$

Compute the RMS voltage for the selected frequency band (see 2.4.2 RMS Estimates):

$$V_{rms}$$

The noise level may also be reported in counts:

$$count_{rms} = \frac{V_{rms}}{bitweight}$$

Estimate the maximum potential dynamic range (MPDR) for the frequency band, ensuring to convert the digitizers peak voltage to an RMS quantity (see 2.4.3 Dynamic Range Estimation):

$$MPDR = 10\ \log_{10}\left(\frac{V_{peak}/\sqrt{2}}{V_{rms}}\right)^2$$

For the purpose of comparison, the theoretical level of noise due to quantization in an ideal digitizer (Oppenheim, 1999, Pg 193-197), assuming a uniformly distributed random quantization error and a digitizer with a peak full scale of $V_{peak}$and B bits, is:

$$Quantization\ noise = \frac{\left(V_{peak}/2^{B-1}\right)^2}{12}$$

Since this noise is assumed to be uniformly distributed, it may also be overlaid against the input terminated power spectral density, where F_s is the sampling frequency, at the level:

$$Spectral\ Noise = \frac{\left(V_{peak}/2^{B-1}\right)^2}{12 * \frac{F_s}{2}}$$

Note that the quantization noise is spectrally distributed across one half of the sampling frequency due to the fact that our power spectra density estimates are single sided with the negative frequencies doubled over.

Table 31: Test Waveforms

Waveform	**Unit**	**Source**
Input Terminated Waveform	Volt	DUT

Table 32: Test Parameters

Parameter	**Unit**
PSD Parameters	
Terminator Resistance	Ohms
PSD Frequency Band	Hz
Digitizer Full scale	Volt

Table 33: Test Results

Result	**Unit**
Input Terminated Noise	Volt RMS
Input Terminated Noise	Count RMS
Maximum Potential Dynamic Range	dB

3.2.13 Modified Noise Power Ratio

The Modified Noise Power Ratio test (McDonald, 1994) compares the performance of a pair of identical digitizers to the performance of an ideal digitizer at various bit lengths by calculating the SNR as a function of signal power. This test is able to demonstrate the linearity of a digitizer across a range of amplitudes.

For this test, a function generator is connected to two different digitizers or digitizer channels as shown in the block diagram below.

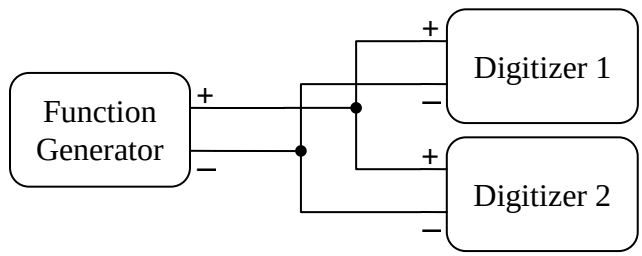

Figure 60 Modified Noise Power Ratio Diagram

The function generator is configured to output bandwidth limited, typically low-pass filtered below 20 Hz, Gaussian white noise over a range of user configurable amplitudes. The digitizers have a manufacturer's defined full scale input voltage range, V_{fs}, and the amplitude of the function generator output should ideally span that range over approximate 10 dB intervals. The tester collects time series data from the two digitizers for each of the amplitudes:

$$V_i, \qquad 0 \le i < \#\, amplitudes$$

$$x_{1,i}[n], \qquad 0 \le i < \#\, amplitudes, 0 \le n \le N-1$$
$$x_{2,i}[n], \qquad 0 \le i < \#\, amplitudes, 0 \le n \le N-1$$

The corresponding pairs of time series are analyzed to determine the noise spectra using coherence analysis with a distributed noise model (see 2.7.1 Two Channel Coherence):

$$P_{i,nn}[k], \qquad 0 \le i \le \#\, amplitudes, 0 \le k \le N-1$$

The tester defines an appropriate frequency band over which to compute the RMS estimates of signal and noise. The noise RMS voltage, $V_{i,noise}$, is computed from the noise spectra and the signal RMS voltage, $V_{i,signal}$, is computed from either digitizers signal spectra (see 2.4.2 RMS Estimates). The noise power ratio data point for this amplitude is then computed as the ratio between the signal and noise voltages:

$$NPR_i = \frac{V_{i,signal}}{V_{i,noise}}$$

A "loading factor", K_i, is computed as the ratio between the digitizers full scale voltage and the signal RMS voltage:

$$K_i = \frac{V_{fs}}{V_{i,signal}}$$

The noise power ratio data point is plotted in dB, 20log10(NPR_i) versus the loading factor in dB, -20 $\log_{10}(K_i)$.

The theoretical noise levels for a digitizer, where n is the number of bits, assume that the only two sources of noise are quantization noise:

$$quantization\ noise\ ratio = 2 * \frac{K^2 * \left(\frac{1}{2} - Q(K)\right)}{12 * (2^{n-1} - 1)^2}$$

and saturation noise:

$$saturation\ noise\ ratio = 2 * \left[(K^2 + 1) * Q(K) - \frac{K * e^{-\frac{K^2}{2}}}{\sqrt{2\pi}}\right]$$

Where $Q(K)$ is the partial area under the normalized Gaussian curve (due to the signal source being Gaussian):

$$Q(K) = \frac{1}{\sqrt{2\pi}} \int_K^{\infty} e^{-x^2/2} dx$$

The theoretical noise power ratio is then the inverse of the sum of the quantization and saturation noise ratios. The noise power ratio, in dB, is then plotted versus -20*log10(K).

$$NPR_{theoretical} = \frac{1}{quantization\ noise\ ratio + saturation\ noise\ ratio}$$

Table 34: Test Waveforms

Waveform	Unit	Source
Time Pulse Waveform	Volt	DUT

Table 35: Test Parameters

Parameter	Unit
PSD Parameters	
Low Frequency Band	Hz
High Frequency Band	Hz

Table 36: Test Results

Result	Unit
K values	dB
NPR values	dB

3.2.14 Relative Transfer Function

The Relative Transfer Function test measures the amount of channel-to-channel timing skew present on a digitizer. The test is performed by feeding white noise from a function generator into two of the digitizer channels as shown in the block diagram below.

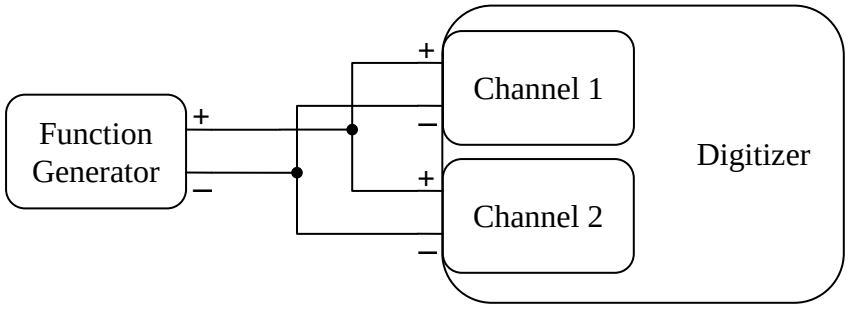

Figure 61 Relative Transfer Function Diagram

The relative response function between the two channels (see 2.7.1 Two Channel Coherence) is computed assuming a distributed noise model:

$$H[k], \qquad 0 \leq k \leq N-1$$

The tester defines a frequency range over which to measure the skew:

$$f[k], \qquad n \leq k \leq m$$

The amount of skew, in seconds, is computed by averaging the relative phase delay between the two channels (Oppenheim, 1999, Pg 242):

$$skew = \frac{1}{m-n+1} \sum_{k=n}^{m} \frac{\measuredangle(H[k])}{2\pi\, f[k]}$$

Table 37: Test Waveforms

Waveform	Unit	Source
Channel 1 White Noise	Volt	DUT
Channel 2 White Noise	Volt	DUT

Table 38: Test Parameters

Parameter	Unit
PSD Parameters	
White Noise Amplitude	Volt
White Noise Bandwidth	Hz
PSD Frequency Band	Hz

Table 39: Test Results

Result	Unit
Skew	Seconds

3.2.15 Seismic System Noise & Bandwidth Limited Dynamic Range

The Seismic System Noise test measures the digitizer's noise and dynamic range expressed in the units of a seismic sensor. The test is performed identically to the Input Terminated Noise and Maximum Potential Dynamic Range test (see 3.2.12 Input Terminated Noise and Maximum Potential Dynamic Range).

The only difference comes in the analysis. An appropriate response is selected for the digitizer and a response corrected PSD is computed (see 2.4 Power Spectral Density and 2.5.1.5 Removing a response from digital time series data):

$$P_{yy}[k], 0 \leq k \leq N-1$$

Over frequencies (in Hertz):

$$f[k], 0 \leq k \leq N-1$$

Compute the rms value over a user defined frequency band of the PSD (see 2.4.2 RMS Estimates):

$$Noise_{rms}$$

Estimate the Bandwidth Limited Dynamic Range (BLDR) for over the frequency band, using a user defined value for the sensor peak output (see 2.4.3 Dynamic Range Estimation):

$$BLDR = 10\ \log_{10}\left(\frac{Sensor\ Ouput_{peak}/\sqrt{2}}{Noise_{rms}}\right)^2$$

Table 40: Test Waveforms

Waveform	Unit	Source
Test Infrasound Waveform	Seismic Earth Unit	DUT

Table 41: Test Parameters

Parameter	Unit
PSD Parameters	
Frequency(s)	Hz
Peak Sensor Output	Seismic Earth Unit

Table 42: Test Results

Result	Unit
Noise RMS	Seismic Earth Unit
BLDR	Decibel

3.2.16 Sine Calibrator Amplitude

The purpose of the Sine Calibrator Amplitude test is to determine whether the digitizer's calibrator is outputting correct amplitude levels. The digitizer's calibrator output is configured by the tester to output sinusoids with a known frequency, f_0, and over a range of amplitudes, A_i:

$$A_i \ \cos(2\pi f_0 t + \theta)$$

The output from the digitizer's calibrator is connected to a calibrated meter and the output is recorded.

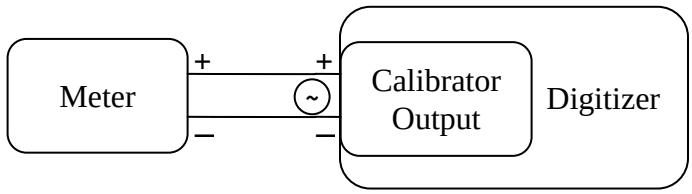

Figure 62 Sine Calibrator Amplitude Diagram

The tester provides the expected frequency and amplitude values as well as the waveform segments for each of the sinusoids. The analysis is then to perform a sine fit (see 2.8.1 Three Parameter Sine Fit) on each of the waveform segments. The resulting sine fit amplitudes are then compared to the expected amplitudes to compute a percent error:

$$a_i \ \cos(2\pi f_0 t + \theta) + V_{dc}$$

$$\%\ Error_i = \ 100 * \frac{a_i - A_i}{a_i}$$

Table 43: Test Waveforms

Waveform	Unit	Source
Test Digitizer Waveform	Volt	DUT

Table 44: Test Parameters

Parameter	Unit
Frequency	Hz
Amplitude Value(s)	Volt

Table 45: Test Results

Result	Unit
Sine Fit Amplitude(s)	Volt
% Error(s)	%

3.2.17 Sine Calibrator Current Amplitude

The purpose of the Sine Calibrator Current Amplitude test is to determine the current output capacity of the digitizer's calibrator. The positive terminal of the calibrator output is connected to ground through a high precision resistor with a known resistance value. The negative terminal of the calibrator output is connected directly to ground. A calibrated Meter is then used to measure the voltage drop across the resistor, as shown in the diagram below.

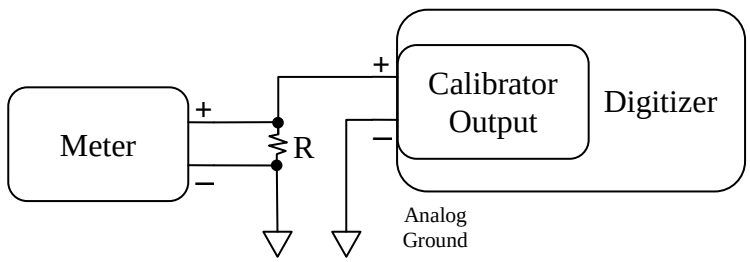

Figure 63 Sine Calibrator Current Amplitude Diagram

The digitizer's calibrator output is configured by the tester to output sinusoids with a known frequency, f_0, and amplitude, A:

$$A\ \cos(2\pi f_0 t + \theta)$$

The resistance value is selected relative to the sinusoid amplitude, A, such that the current across the resistor is slightly less than the rated capacity of the calibrator output. For example, if the selected output sinusoid has a peak amplitude of 10 Volts and the calibrator output is rated to have a current output limit of 1 mA, then the desired resistance value would be no less than:

$$R = \frac{V}{I} = \frac{10\ Volts}{1\ mA} = 10\ k\Omega$$

A segment of data from the meter is recorded and a sine fit is applied (see 2.8.1 Three Parameter Sine Fit) on to the waveform segment to determine the amplitude of the voltage drop across the resistor:

$$a\ \cos(2\pi f_0 t + \theta) + V_{dc}$$

The resulting sine fit amplitude, a, and resistance, R, is then used to determine the output current from the calibrator using Ohm's law (Thomas, 1994, Pg. 25):

$$Current = \frac{a}{R}$$

The results of this test verify that the calibrator can output at least the amount of current indicated by the voltage across the resistor. Note that if the selected voltage, A, and measured voltage, a, have considerably different values or if there is considerable RMS error in the sine fit,

then this would tend to indicate that the calibrator output is current limited. The solution in this case would be to either reduce the selected sinusoid amplitude or increase the load resistor.

Table 46: Test Waveforms

Waveform	Unit	Source
Test Digitizer Waveform	Volt	DUT

Table 47: Test Parameters

Parameter	Unit
Frequency	Hz
Amplitude Value	Volt
Resistance	Ohm

Table 48: Test Results

Result	Unit
Sine Fit Amplitude	Volt
Sine Fit RMS Error	Volt RMS
Current	Amp

3.2.18 Sine Calibrator Frequency

The purpose of the Sine Calibrator Amplitude test is to determine whether the digitizer's calibrator is outputting correct frequencies.

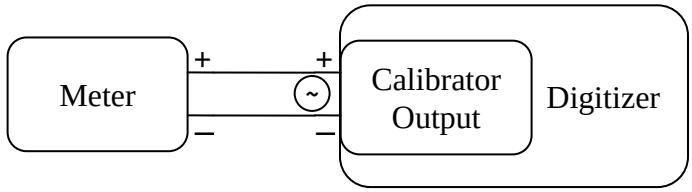

Figure 64 Sine Calibrator Frequency Diagram

The output from the digitizer's calibrator is connected to a calibrated meter and the output is recorded. The digitizer's calibrator output is configured by the tester to output sinusoids with a known amplitude, A_0, and over a range of frequencies:

$$A_0 \ \cos(2\pi F_i t + \theta)$$

The tester provides the expected frequency and amplitude values as well as the waveform segments for each of the sinusoids. The analysis is then to perform a sine fit (see 2.8.1 Three Parameter Sine Fit) on each of the waveform segments. The resulting sine fit amplitudes are then compared to the expected amplitudes to compute a percent error:

$$A_0 \ \cos(2\pi f_i t + \theta) + V_{dc}$$

$$\%\, Error_i = \ 100 * \frac{f_i - F_i}{f_i}$$

Table 49: Test Waveforms

Waveform	Unit	Source
Test Digitizer Waveform	Volt	DUT

Table 50: Test Parameters

Parameter	Unit
Frequency(s)	Hz
Amplitude Value	Volt

Table 51: Test Results

Result	Unit
Sine Frequency(s)	Hz
% Error(s)	%

3.2.19 Sine Calibrator THD

The Sine Calibrator THD test is performed identically to the THD test (see 3.2.22 Total Harmonic Distortion). The only difference is that instead of using a low distortion oscillator, the digitizer is configured to generate its own sine wave from its calibrator and this signal is recorded by an independent calibrated meter. The reason for performing this test is to quantify the expected THD levels when using the digitizer's own calibrator. This way, the results of later in place field testing of the digitizer can be compared to a known reference value.

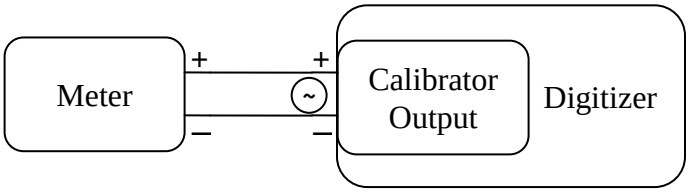

Figure 65 Sine Calibrator THD Diagram

3.2.20 Time Tag Accuracy

The Time Tag Accuracy test measures the digitizer's timing accuracy for the pulse per hour trigger output from an independent GPS Timing reference. The timing trigger output from a GPS receiver is connected to an input channel of the digitizer.

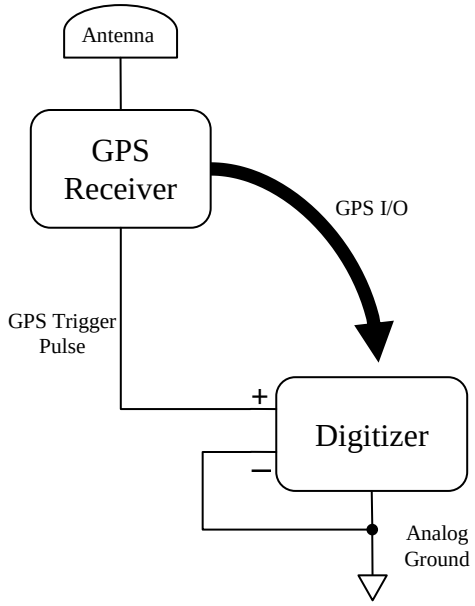

Figure 66 Time Tag Accuracy Diagram

A single timing trigger pulse at an hour crossing from the GPS Receiver is recorded and analysed to determine the timing error. See 2.10 Time Tag Measurements for a description of how the time tag measurements are analyzed.

Table 52: Test Waveforms

Waveform	Unit	Source
Time Pulse Waveform	Volt	DUT

Table 53: Test Parameters

Parameter	Unit

Table 54: Test Results

Result	Unit
Trigger Time	Seconds
Trigger Error	Seconds

3.2.21 Time Tag Drift

The Time Tag Drift test examines the digitizers timing accuracy while it has a stable GPS lock, how the timing accuracy drifts once it has lost GPS lock, and how the timing accuracy recovers once the digitizer has re-acquired GPS lock.

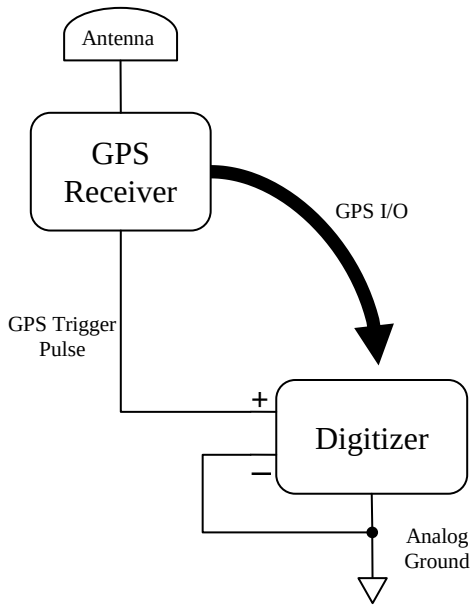

Figure 67 Time Tag Drift Diagram

For this test, the input channel of the digitizer is connected to the timing trigger output of an external GPS Timing Reference. The digitizer is initially allowed to lock onto the GPS and achieve a stable timing error. Next, the digitizers GPS receiver is disabled by either disconnecting the GPS receiver from the digitizer or covering the antenna, depending upon the digitizer model. The digitizer is allowed to run for a period of approximately 1 hour without a GPS lock. After one hour, the digitizer's GPS receiver is enabled and the timing is allowed to recover.

The pulse-per-minute triggers from this entire test are analyzed as described in 2.10 Time Tag Measurements. The set of trigger times and errors are:

$$t[n], \qquad 0 \le n \le N-1$$
$$e[n], \qquad 0 \le n \le N-1$$

The tester defines the start and end times for the drift and recovery periods:

$$i_{d\ start}, i_{d\ end}, i_{r\ start}, i_{r\ end}, \qquad 0 \le i \le N-1$$

The slope of the drift and recovery periods are computed by solving the linear system of equations for the slope a:

$$Y = a * X + b$$

$$Y = [X|1] * \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = [X|1] \setminus Y$$

For the drift period:

$$X = \begin{bmatrix} t[i_{d\ start}] \\ \vdots \\ t[i_{d\ end}] \end{bmatrix}$$

$$Y = \begin{bmatrix} e[i_{d\ start}] \\ \vdots \\ e[i_{d\ end}] \end{bmatrix}$$

For the recovery period:

$$X = \begin{bmatrix} t[i_{r\ start}] \\ \vdots \\ t[i_{r\ end}] \end{bmatrix}$$

$$Y = \begin{bmatrix} e[i_{r\ start}] \\ \vdots \\ e[i_{r\ end}] \end{bmatrix}$$

Table 55: Test Waveforms

Waveform	**Unit**	**Source**
Time Pulse Waveform	Volt	DUT

Table 56: Test Parameters

Parameter	**Unit**
Drift start time	Seconds
Drift end time	Seconds
Recovery start time	Seconds
Recovery end time	Seconds

Table 57: Test Results

Result	**Unit**
Drift Rate	Microseconds/hour
Recovery Rate	Microseconds/hour

3.2.22 Total Harmonic Distortion

The Total Harmonic Distortion (THD) test measures the amount of distortion present in a pure sine wave from an ultra-low distortion oscillator that is fed into an input channel of a digitizer. It is assumed that the oscillator is capable of generating a higher quality sinusoid than the digitizer is capable of sampling. The function generator is connected to the digitizer as shown in the block diagram below:

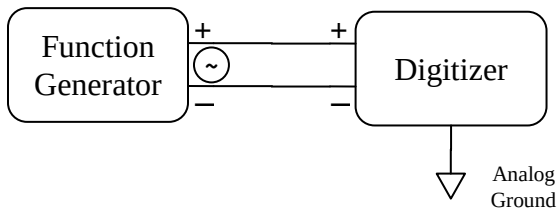

Figure 68 Total Harmonic Distortion Diagram

The power-spectra of the input signal will contain a single fundamental peak at the frequency of the tonal input signal. Any distortion of the sinusoid will be present in the power-spectra as peaks at integer multiples, or harmonics, of the fundamental frequency.

First, compute the power spectra of the waveform segment, using a Kaiser Bessel window (see 2.3.4 Kaiser Bessel) and 5/8 overlap (see 2.4 Power Spectral Density).

$$P_{xx}[k]$$

Over frequencies (in Hertz):

$$f[k]$$

The Kaiser Bessel window is used because of its relatively narrow main lobe and minimal side lobes. These qualities serve to minimize the amount of spectral smearing, allowing for greater frequency resolution.

Next, identify the start and stop locations of all of the peaks in the power spectra using the algorithm represented in the following diagram that looks for at least two consecutive increases followed by at least two consecutive decreases. In addition, the maximum value of the peak must exceed an average baseline value of the prior power spectra values plus the 90 % confidence interval (see 2.4.1 PSD Confidence). Finally, the peak must be located at an integer multiple of the fundamental frequency.

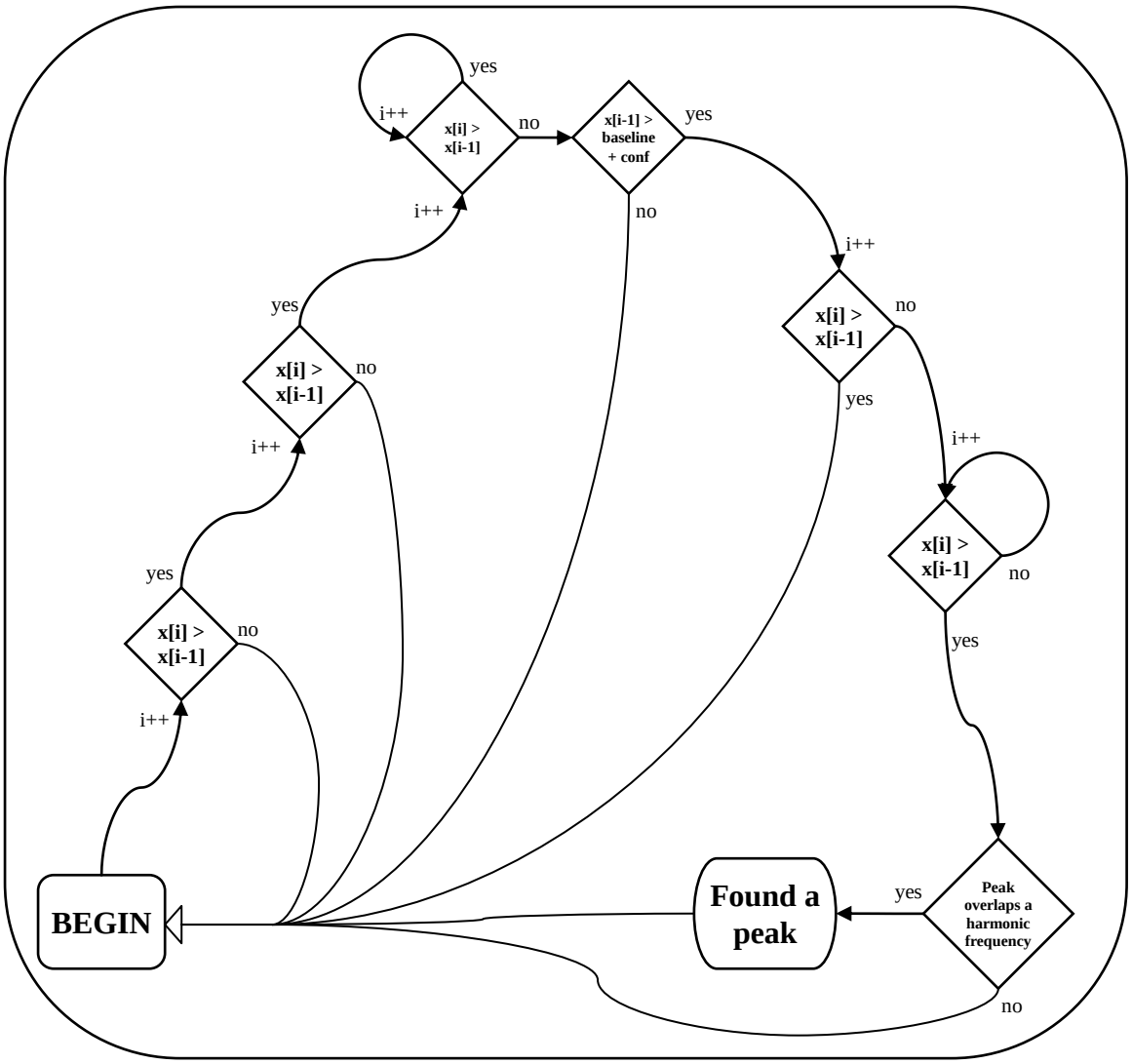

Figure 69 THD Peak Identification Algorithm

The result of this algorithm is that the start and end indices of each peak are identified and stored in the following vectors where M is the number of peaks identified:

$$start[l], \qquad 0 \le l \le M-1$$
$$end[l], \qquad 0 \le l \le M-1$$

The frequencies corresponding to these start and end indicies are:

$$f\big[start[l]\big] \ldots f\big[end[l]\big]$$

Estimate the frequency at which each peak occurs using a trapezoidal fitting function:

$$k_{max}[l], \qquad where\ P_{xx}\big[k_{max}[l]\big] = \max_{start[l] \le i \le end[l]} P_{xx}[k]\,, \qquad 0 \le l \le M-1$$

$$f_{peak}[l] = \Delta f\left(k_{max}[l] - 2 + \frac{1.5P_{xx}[k_{max}[l]-1] - 2P_{xx}\left[k_{max}[l]\right] + 0.5P_{xx}[k_{max}[l]+1]}{P_{xx}[k_{max}[l]-1] - 2P_{xx}\left[k_{max}[l]\right] + P_{xx}[k_{max}[l]+1]}\right)$$

Next, compute the RMS voltage in the frequency bands identified around each of the peaks (see 2.4.2 RMS Estimates) where L is the length of the FFT:

$$rms[l] = \sqrt{\frac{1}{T_s L}\sum_{k=start[l]}^{end[l]} Pxx[k]}\ , \qquad 0 \le l \le M-1$$

The fundamental peak is identified as being the first peak within the power spectra. All of the remaining peaks are assumed to be harmonics of the fundamental.

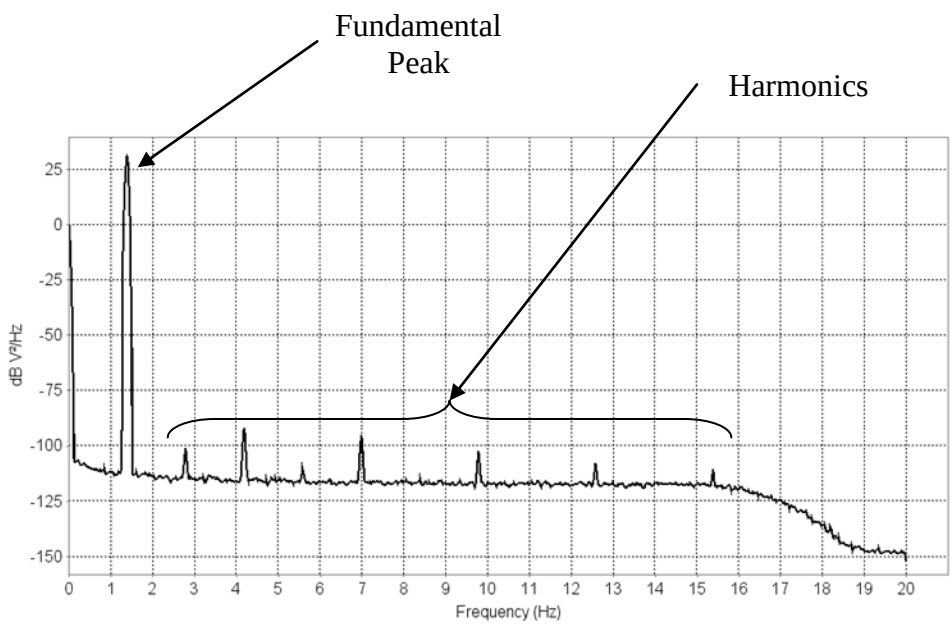

Figure 70 Example THD

If the peak search algorithm is unable to identify a peak for at least one harmonic of the fundamental frequency, then the value of the PSD at the next integer multiple of the fundamental frequency is used:

$$peak[1] = \sqrt{\frac{1}{T_s L}Pxx[k]}, \qquad k = presumed\ location\ of\ first\ harmonic$$

Compute the Total Harmonic Distortion, in dB, as the ratio between the power in all of the identified harmonics and the power in the primary frequency (IEEE Std 100-2000, Pg. 1191):

$$THD_{dB} = 10\log_{10}\left(\frac{\sqrt{\sum_{l=1}^{M-1}(rms[l])^2}}{rms[0]}\right)^2$$

Table 58: Test Waveforms

Waveform	Unit	Source
THD Waveform	Volt	DUT

Table 59: Test Parameters

Parameter	Unit
PSD Parameters	

Table 60: Test Results

Result	Unit
Peak Frequencies	Hz
Peak RMS values	Volt RMS
Total Harmonic Distortion	dB

3.3 Seismic Sensor Tests

3.3.1 Seismic Sensor Self-Noise

The Seismic Sensor Self Noise test measures the noise level of a seismic sensor under test relative to either one or two reference seismic sensors. The seismic sensor under test and the reference sensors with known response characteristics are co-located so that they are both measuring a common seismic signal.

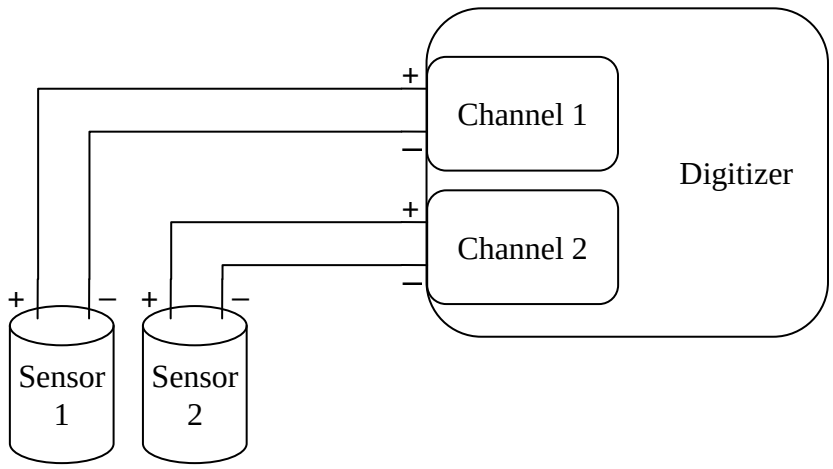

Figure 71 Seismic Sensor Self-Noise Diagram

The waveforms, in volts, from the reference sensors and the sensor under test are collected and analyzed.

$$x_{test}[t], \qquad 0 \le t \le N_{samples} - 1$$
$$x_{ref1}[t], \qquad 0 \le t \le N_{samples} - 1$$
$$x_{ref2}[t], \qquad 0 \le t \le N_{samples} - 1$$

The appropriate response is selected for each of the reference and test sensors and the response corrected auto and cross PSDs are computed (see 2.4 Power Spectral Density, 2.5.1.6 Converting a seismic response between unit types, and 2.5.1.5 Removing a response from digital time series data):

If there is one reference sensor, then the 2 Channel Coherence technique (see 2.7.1 Two Channel Coherence) is applied to the PSDs to compute the coherence between the two sensors and the amount of noise present in the sensor under test. If the reference sensors noise is lower than the sensor under test, then a lumped noise model is used. If the reference sensor and the sensor under test are of a common sensor type and have equivalent noise levels, then a distributed noise model is used.

If there are two reference sensors, then the 3 Channel Coherence Technique (see 2.7.2 Three Channel Coherence) is applied to the PSDs to compute the coherence and the amount of noise present in the sensor under test. Note that the 3 Channel Coherence Technique should only be applied if the sensors have similar response and noise characteristics.

If the seismic test sensor response is correct, then across the sensors common pass bands the coherence should be 1, the relative gain should be 0 dB, and the relative phase should be 0 degrees. Otherwise, the response is not correct.

Table 61: Test Waveforms

Waveform	Unit	Source
Reference Infrasound Waveform(s)	Pascal (Pressure)	Reference Sensor
Test Infrasound Waveform	Pascal (Pressure)	DUT

Table 62: Test Parameters

Parameter	Unit
PSD Parameters	

Table 63: Test Results

Result	Unit
Coherence	Unitless
Relative Gain	dB
Relative Phase	degrees

3.3.2 Seismic Sensor Response Verification

The Seismic Sensor Response Verification test measures the response of a seismic sensor under test relative to a reference seismic sensor. The seismic sensor under test and the reference sensor with known response characteristics are co-located so that they are both measuring a common earth motion.

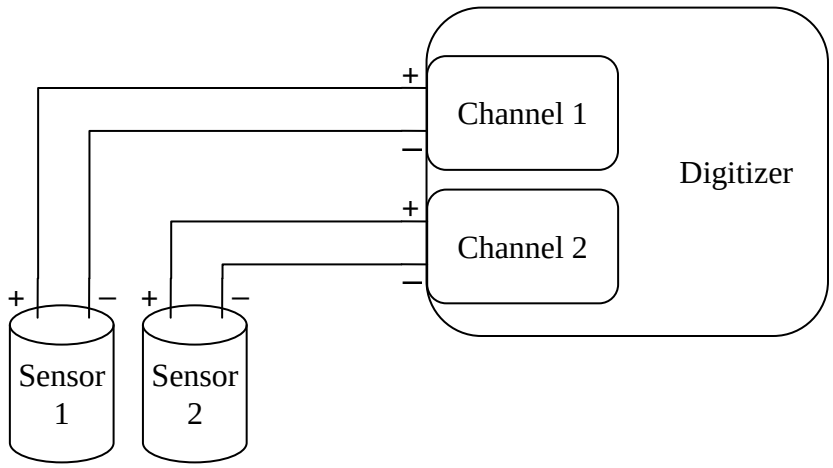

Figure 72 Seismic Sensor Response Verification Diagram

The waveforms, in volts, from the reference sensor and the sensor under test are collected and analyzed.

$$x_{test}[t], \qquad 0 \le t \le N_{samples} - 1$$
$$x_{ref}[t], \qquad 0 \le t \le N_{samples} - 1$$

The appropriate response is selected for each of the reference and test sensors and the response corrected auto and cross PSDs are computed (see 2.4 Power Spectral Density, 2.5.1.6 Converting a seismic response between unit types, and 2.5.1.5 Removing a response from digital time series data):

$$Pxx[k], Pyy[k], Pxy[k]$$

The 2-Channel Coherence technique (see 2.7.1 Two Channel Coherence) is applied to the PSDs to compute the coherence between the two sensors and the relative gain and phase. If the seismic test sensor response is correct, then across the sensors common pass bands the coherence should be 1, the relative gain should be 0 dB, and the relative phase should be 0 degrees. Otherwise, the response is not correct.

Table 64: Test Waveforms

Waveform	Unit	Source
Reference Seismic Waveform	Earth Motion	Reference Sensor
Test Seismic Waveform	Earth	DUT

	Motion	

Table 65: Test Parameters

Parameter	Unit
PSD Parameters	

Table 66: Test Results

Result	Unit
Coherence	Unitless
Relative Gain	dB
Relative Phase	degrees

3.4 Infrasound Sensor Tests

3.4.1 Infrasound Sensor Frequency Response

The Infrasound Sensor Frequency Response test measures the sensitivity of an infrasound sensor under test at multiple discrete frequency values. The infrasound sensor under test and a reference sensor with known response characteristics are placed inside of a pressure isolation chamber. The isolation chamber serves to attenuate any external ambient variations in pressure.

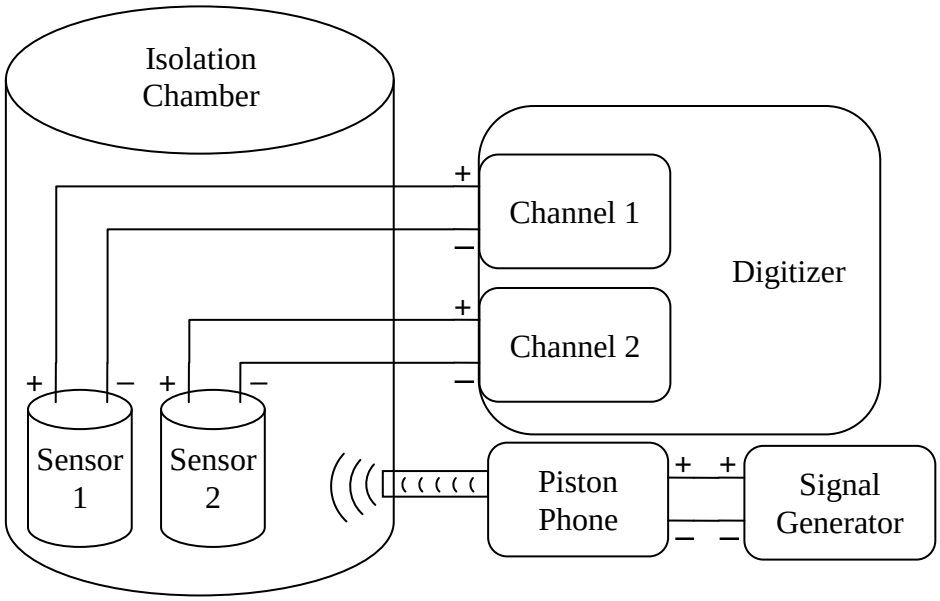

Figure 73 Infrasound Sensor Frequency Response Diagram

A piston-phone is attached to an inlet port on the isolation chamber. The piston-phone is driven with a sinusoid from a signal generator or the analog calibration output from a digitizer. The piston-phone serves to generate a pressure wave with characteristics defined by the signal generator. This pressure wave is recorded by both the reference sensor and the sensor under test.

The waveforms, in volts, from the reference sensor and the sensor under test are collected and analyzed.

$$x_{ref}[t], \qquad 0 \le t \le N_{samples} - 1$$
$$x_{test}[t], \qquad 0 \le t \le N_{samples} - 1$$

The frequency of the signal being outputted by the signal generator is adjusted across a range of discrete, user defined frequency values so as to sample the response of the test sensor.

$$f[n], \qquad 0 \le n \le N_{frequencies} - 1$$

The tester provides data windows for the reference and test sensors for each of the frequencies being tested. The length of the data segments is a user-defined parameter expressed as the

number of cycles for each defined frequency value. The number of cycles is divided by the frequency being examined to obtain the window length in seconds.

For each data segment, the 3-parameter sine fit algorithm (see 2.8.1 Three Parameter Sine Fit) is applied to successive windows of the time series data collected from the reference sensor.

$$A\ \cos(\omega_0 t + \theta) +\ C$$

$$\begin{aligned} A_{test}[n], A_{ref}[n], &\quad amplitude\ in\ volts \\ \theta_{test}[n], \theta_{ref}[n], &\quad phase\ offset\ in\ radians \\ C_{test}[n], C_{ref}[n], &\quad dc\ offset\ in\ volts \\ \varepsilon_{test}[n], \varepsilon_{ref}[n], &\quad RMS\ error\ in\ volts \end{aligned}$$

The phase offset that is stored is relative to the reference sensor:

$$Phase[n] = \theta_{ref}[n] - \theta_{test}[n]$$

From the RMS error, an SNR value in dB can be computed:

$$SNR[n] = 10 * \log_{10}\left(\frac{A_{test}[n]/\sqrt{2}}{\varepsilon_{test}[n]}\right)^2$$

In addition, the user selects an instrument response that defines the sensitivity in Volts/Pa of the reference sensor.

$$H_{ref}(s)$$

$$H_{ref}[n] = \left|H_{ref}(s)\right|_{s=j2\pi f[n]}$$

Given the peak amplitude values, in volts, for the reference sensor and the sensitivity, the peak pressure may be computed:

$$P_{ref}[n] = \frac{A_{ref}[n]}{H_{ref}[n]}$$

Using the peak pressure from the reference sensor, the sensitivity values, in Volts/Pa, for the test sensor may then be estimated:

$$H_{test}[n] = \frac{A_{test}[n]}{P_{ref}[n]}$$

Table 67: Test Waveforms

Waveform	**Unit**	**Source**
Reference Infrasound Waveform	Pascal (Pressure)	Reference Sensor

Test Infrasound Waveform	Volt	DUT

Table 68: Test Parameters

Parameter	**Unit**
Frequency(s)	Hz
Amplitude Value	Volt

Table 69: Test Results

Result	**Unit**
Sine Frequency(s)	Hz
Sine Fit Amplitude	Volt
Sine Fit Phase	Radian
Sine Fit DC Offset	Volt
Sine Fit Error	Volt RMS
Sine Fit SNR	dB
Peak Pressure	Pascal
Sensitivity	Volt/Pascal

3.4.2 Infrasound Sensor Amplitude Response

The Infrasound Sensor Amplitude Response test measures the sensitivity of an infrasound sensor under test at multiple discrete amplitude values at a fixed tonal frequency. The infrasound sensor under test and a reference sensor with known response characteristics are placed inside of a pressure isolation chamber. The isolation chamber serves to attenuate any external ambient variations in pressure.

A piston-phone is attached to an inlet port on the isolation chamber. The piston-phone is driven with a sinusoid from a signal generator or the analog calibration output from a digitizer. The piston-phone serves to generate a pressure wave with characteristics defined by the signal generator. This pressure wave is recorded by both the sensor under test and the reference sensor.

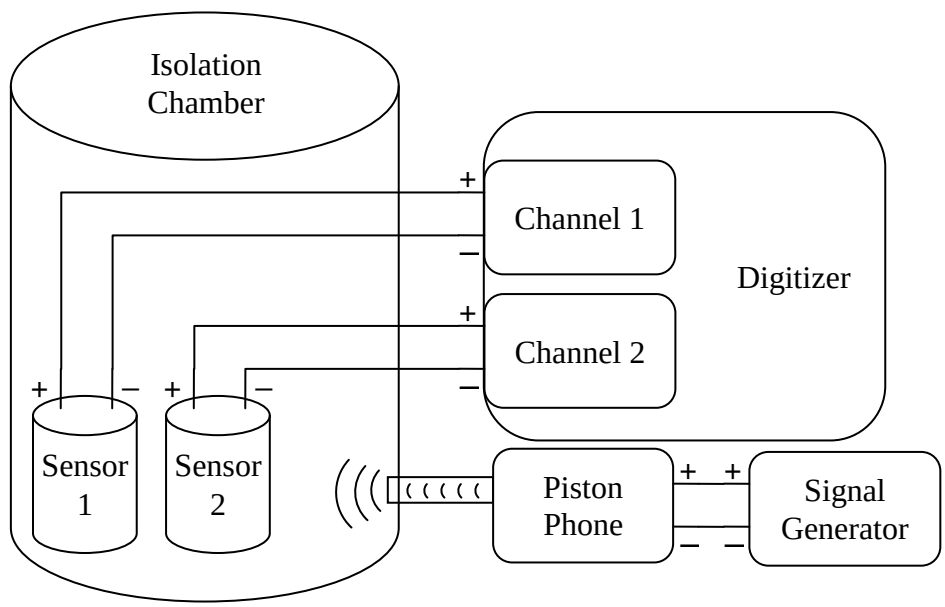

Figure 74 Infrasound Sensor Amplitude Response Diagram

The waveforms, in volts, from the reference sensor and the sensor under test are collected and analyzed.

$$x_{ref}[t], \qquad 0 \le t \le N_{samples} - 1$$
$$x_{test}[t], \qquad 0 \le t \le N_{samples} - 1$$

The amplitude of the signal being outputted by the signal generator is adjusted across a range of discrete, user defined amplitude values so as to sample the sensitivity of the test sensor.

$$A[n], \qquad 0 \le n \le N_{amplitudes} - 1$$

The tester provides data windows for the reference and test sensors for each of the frequencies being tested. The length of the data segments is a user-defined parameter expressed as the

number of cycles for each defined frequency value. The number of cycles is multiplied by the frequency being examined to obtain the window length in seconds.

For each data segment, the 3-parameter sine fit algorithm (see 2.8.1 Three Parameter Sine Fit) is applied to successive windows of the time series data collected from the reference sensor.

$$A\ \cos(\omega_0 t + \theta) +\ C$$

$$\begin{aligned} A_{test}[n], A_{ref}[n], &\quad amplitude\ in\ volts \\ \theta_{test}[n], \theta_{ref}[n], &\quad phase\ offset\ in\ radians \\ C_{test}[n], C_{ref}[n], &\quad dc\ offset\ in\ volts \\ \varepsilon_{test}[n], \varepsilon_{ref}[n], &\quad RMS\ error\ in\ volts \end{aligned}$$

The phase offset that is stored is relative to the reference sensor:

$$Phase[n] = \theta_{ref}[n] - \theta_{test}[n]$$

From the RMS error, an SNR value in dB can be computed:

$$SNR[n] = 10 * \log_{10}\left(\frac{A_{test}[n]/\sqrt{2}}{\varepsilon_{test}[n]}\right)^2$$

In addition, the user selects an instrument response that defines the sensitivity in Volts/Pa of the reference sensor.

$$H_{ref}(s)$$

$$H_{ref}[n] = \left|H_{ref}(j2\pi f[n])\right|$$

Given the peak amplitude values, in volts, for the reference sensor and the sensitivity, the peak pressure may be computed:

$$P_{ref}[n] = \frac{A_{ref}[n]}{H_{ref}[n]}$$

Using the peak pressure from the reference sensor, the sensitivity values, in Volts/Pa, for the test sensor may then be estimated:

$$H_{test}[n] = \frac{A_{test}[n]}{P_{ref}[n]}$$

Table 70: Test Waveforms

Waveform	Unit	Source
Reference Infrasound Waveform	Pascal (Pressure)	Reference Sensor

Test Infrasound Waveform	Volt	DUT

Table 71: Test Parameters

Parameter	**Unit**
Frequency	Hz
Amplitude Value(s)	Volt

Table 72: Test Results

Result	**Unit**
Sine Frequency(s)	Hz
SineFit Amplitude(s)	Volt
SineFit Phase(s)	Radian
SineFit DC Offset(s)	Volt
SineFit Error(s)	Volt RMS
Peak Pressure(s)	Pascal
Sensitivity(s)	Volt/Pascal

3.4.3 Infrasound Sensor Response Verification

The Infrasound Sensor Response Verification test measures the response of an infrasound sensor under test relative to a reference infrasound sensor. The infrasound sensor under test and the reference sensor with known response characteristics are co-located so that they are both measuring a common pressure field.

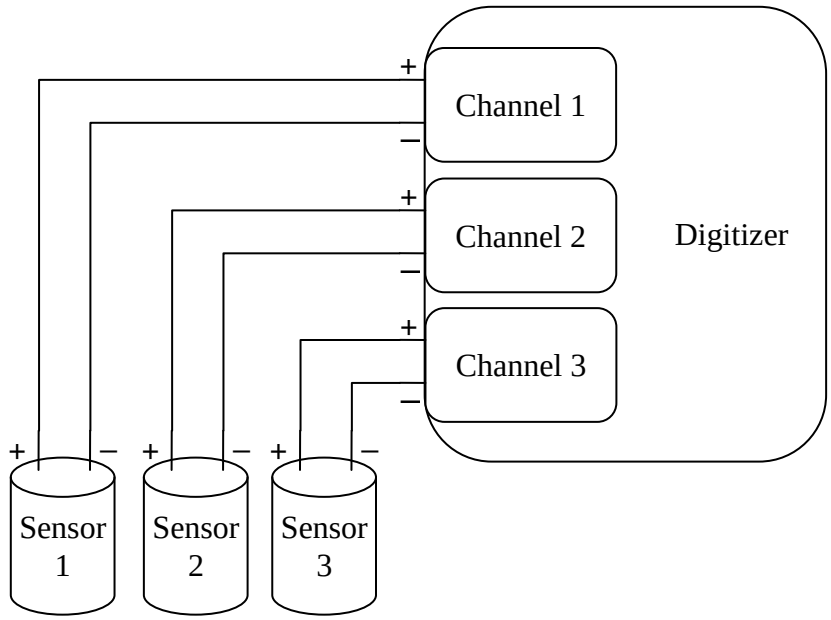

Figure 75 Infrasound Sensor Response Verification Diagram

The waveforms, in volts, from the reference sensor and the sensor under test are collected and analyzed.

$$x_{ref}[t], \qquad 0 \le t \le N_{samples} - 1$$
$$x_{test}[t], \qquad 0 \le t \le N_{samples} - 1$$

A response is selected for each of the reference and test sensors and the response corrected auto and cross PSDs are computed (see 2.4 Power Spectral Density and 2.5.1.5 Removing a response from digital time series data):

$$Pxx[k], Pyy[k], Pxy[k]$$

If there is one reference sensor, then the 2 Channel Coherence technique (see 2.7.1 Two Channel Coherence) is applied to the PSDs to compute the coherence, relative gain, and relative phase between the two sensors. If the reference sensors noise is lower than the sensor under test, then a lumped noise model is used. If the reference sensor and the sensor under test are of a common sensor type and have equivalent noise levels, then a distributed noise model is used.

If there are two reference sensors, then the 3 Channel Coherence Technique (see 2.7.2 Three Channel Coherence) is applied to the PSDs to compute the coherence, relative gain, and relative

phase between the three sensors. Note that the 3 Channel Coherence Technique should only be applied if the sensors have similar response and noise characteristics.

If the infrasound test sensor response is correct, then across the sensors common pass bands the coherence should be 1, the relative gain should be 0 dB, and the relative phase should be 0 degrees. Otherwise, the response is not correct.

Table 73: Test Waveforms

Waveform	Unit	Source
Reference Infrasound Waveform	Pascal (Pressure)	Reference Sensor
Test Infrasound Waveform	Pascal (Pressure)	DUT

Table 74: Test Parameters

Parameter	Unit
PSD Parameters	

Table 75: Test Results

Result	Unit
Coherence	Unitless
Relative Gain	dB
Relative Phase	degrees

3.4.4 Infrasound Sensor Self Noise

The Infrasound Sensor Self Noise test measures the response of an infrasound sensor under test relative to either one or two reference infrasound sensors. The infrasound sensor under test and the reference sensors with known response characteristics are co-located so that they are both measuring a common pressure field.

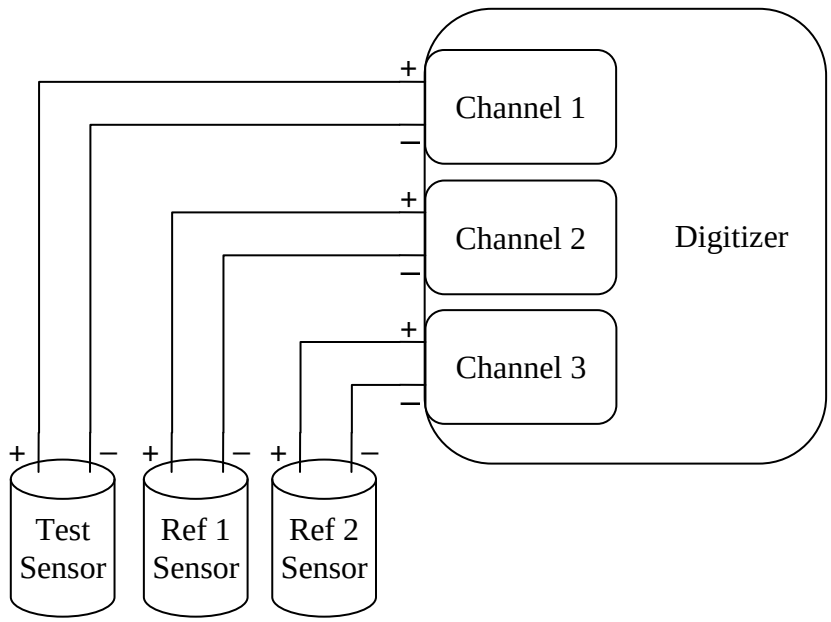

Figure 76 Infrasound Sensor Self Noise Diagram

The waveforms, in volts, from the reference sensors and the sensor under test are collected and analyzed.

$$x_{test}[t], \qquad 0 \le t \le N_{samples} - 1$$
$$x_{ref1}[t], \qquad 0 \le t \le N_{samples} - 1$$
$$x_{ref2}[t], \qquad 0 \le t \le N_{samples} - 1$$

The appropriate response is selected for each of the reference and test sensors and the response corrected auto and cross PSDs are computed (see 2.4 Power Spectral Density and 2.5.1.5 Removing a response from digital time series data):

If there is one reference sensor, then the 2 Channel Coherence technique (see 2.7.1 Two Channel Coherence) is applied to the PSDs to compute the noise present in the sensor under test. If there are two reference sensors, then the 3 Channel Coherence Technique (see 2.7.2 Three Channel Coherence) is applied to the PSDs to compute the noise present in the sensor under test.

The tester defines the sensors peak pressure and a frequency band over which to compute the RMS:

$$Pressure_{peak}$$

$$f_\min, f_max$$

Compute the rms pressure over a user defined frequency band of the PSD (see 2.4.2 RMS Estimates):

$$Pressure_{rms}$$

Estimate the dynamic range for over the frequency band, using a user defined value for the sensor peak output (see 2.4.3 Dynamic Range Estimation):

$$Dynamic\ Range = 10\ \log_{10}\left(\frac{Pressure_{peak}/\sqrt{2}}{Pressure_{rms}}\right)^2$$

Table 76: Test Waveforms

Waveform	**Unit**	**Source**
Reference Infrasound Waveform(s)	Pascal	Reference Sensor
Test Infrasound Waveform	Pascal	DUT

Table 77: Test Parameters

Parameter	**Unit**
PSD Parameters	
Peak Sensor Output	Pascal
Frequency (min)	Hz
Frequency (max)	Hz

Table 78: Test Results

Result	**Unit**
Noise	Pascal RMS
Dynamic Range	Decibel

3.4.5 Infrasound Sensor Isolation Noise

The Infrasound Sensor Isolation Noise test measures the internal noise of an infrasound sensor under test when the ambient infrasound signals present are below the sensor noise. The infrasound sensor is placed inside of a pressure isolation chamber. The isolation chamber serves to attenuate any external ambient variations in pressure that would otherwise be recorded.

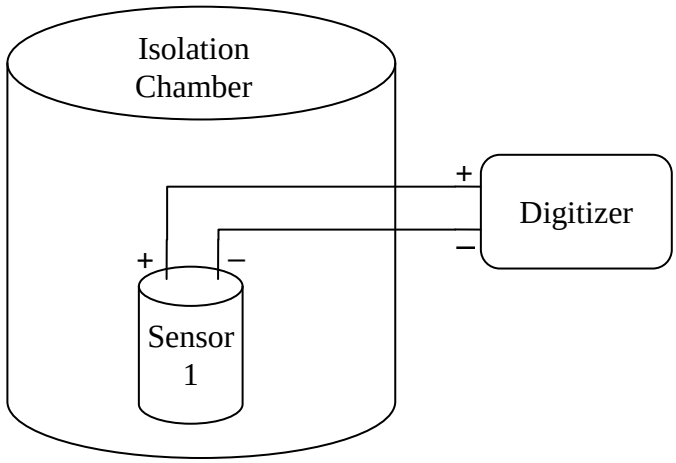

Figure 77 Infrasound Sensor Isolation Noise Diagram

The waveforms, in volts, from the sensor under test are collected and analyzed.

$$x[t], \qquad 0 \leq t \leq N_{samples} - 1$$

An appropriate response is selected for the sensor and the time series is converted to Pressure (see 2.5.1.5 Removing a response from digital time series data):

$$y[t], \qquad 0 \leq t \leq N_{samples} - 1$$

A response corrected PSD is computed (see 2.4 Power Spectral Density):

$$P_{yy}[k], 0 \leq k \leq N - 1$$

Over frequencies (in Hertz):

$$f[k], 0 \leq k \leq N - 1$$

The tester defines the sensors peak pressure and a frequency band over which to compute the RMS:

$$Pressure_{peak}$$

$$f_\min, f_max$$

Compute the rms pressure over a user defined frequency band of the PSD (see 2.4.2 RMS Estimates):

$$Pressure_{rms}$$

Estimate the dynamic range for over the frequency band, using a user defined value for the sensor peak output (see 2.4.3 Dynamic Range Estimation):

$$Dynamic\ Range = 10\ \log_{10}\left(\frac{Pressure_{peak}/\sqrt{2}}{Pressure_{rms}}\right)^2$$

Table 79: Test Waveforms

Waveform	**Unit**	**Source**
Test Infrasound Waveform	Pascal	DUT

Table 80: Test Parameters

Parameter	**Unit**
PSD Parameters	
Peak Sensor Output	Pascal
Frequency (min)	Hz
Frequency (max)	Hz

Table 81: Test Results

Result	**Unit**
Noise	Pascal RMS
Dynamic Range	Decibel

APPENDIX

Window Comparison

Matlab script to display the time and frequency domain characteristics of a given window.

```
function window_comparison(w, name)
%
%  Display the time and frequency domain characteristics
%  of the provided window w.
%

w = w(:)';

N=length(w);
k=0:N-1;

dr = 100;

B = N*sum(w.^2)/sum(w)^2;    % noise bandwidth (bins)

H = abs(fft([w zeros(1,7*N)]));
H = fftshift(H);
H = H/max(H);
H = 20*log10(H);
H = max(0,dr+H);

figure;
area(k,w,'FaceColor', [0 .4 .6]);
xlim([0 N-1]);
set(gca,'XTick', [0 : 1/8 : 1]*(N-1));
set(gca,'XTickLabel','0| | | | | | | |N-1');
grid on;
ylabel('amplitude');
xlabel('samples');
title(['Window function (' name ')']);

figure;
stem(([1:(8*N)]-1-4*N)/8,H,'-');
set(findobj('Type','line'),'Marker','none','Color',[.871 .49 0]);
xlim([-4*N 4*N]/8);
ylim([0 dr]);
set(gca,'YTickLabel','-100|-90|-80|-70|-60|-50|-40|-30|-20|-10|0');
grid on;
ylabel('decibels');
xlabel('DFT bins');
title(['Frequency response (' name ')']);
```

PSD Confidence

Matlab scripts to compare the effects of window function and overlap on the PSD confidence interval.

```
function psd_confidence
%
%  Display a plot of various confidence intervals
%

L = 1024;
overlap = L:-1:1;
N = 100000;

windows = [ rectwin(L) hann(L) hamming(L) kaiser(L,21.0813) bartlett(L) ];
windows_title = { 'Rectangular', 'Hann', 'Hamming', 'Kaiser', 'Bartlett' };
window_color = { 'b', 'g', 'r', 'c', 'm', 'y', 'b', 'w' };

figure;
grid on;
box on;
hold on;

xlabel('Step Size as a fraction of window length');
ylabel('90% Confidence Interval (dB)');

for i = 1:size(windows,2)

    conf = zeros(size(overlap));

    for j = 1:length(overlap)

        conf(j) = confidence_90(windows(:,i), overlap(j), N);

    end

    plot(overlap/L, conf, [ window_color{i} '-' ], ...
        'LineWidth', 2, 'LineSmooth', 'on');
end

legend(windows_title{:}, 'Location', 'NorthWest');



function conf = confidence_90(w, R, N)
%
%  Compute the 90% confidence interval, in dB, for
%  the provided window and overlap over a range of sample lengths
%

L = length(w);
weight = window_weight(w,R);

%
%  Compute K, the number of windows
%
K = ceil((N-L)/R+1);

%
%  Compute the scale factor
%
scale = ones(length(N),1);
for j = 1:length(weight)
    scale = scale + 2 * (K-j)./K * weight(j);
end

conf = 14.1 ./ sqrt(2.*K./scale-0.8333);
```

```
function weight = window_weight(w, R)
%
%  Compute the window weighting function given the window w
%  and step size R
%

L = length(w);
K = ceil(L/R)-1;

power = sum(w.^2)^2;

weight = zeros(K,1);

for j=1:K

   index = 1:(L-j*R);

   weight(j) =  sum(w(index).*w(index+j*R))^2 / power;

end
```

Filters

Matlab script to display poles, zeros, amplitude response, and phase response of a filter.

```
function filters(z,p,k)
%
%  Display information for the provided zeros, poles, and gain
%

%
%  Display the pole-zero plot
%
figure;
hold on;
box on;
grid on;
xlabel('Real');
ylabel('Imaginary');
axis equal;
%axis square;
plot(real(p), imag(p), 'rx', 'MarkerSize', 14, 'LineWidth', 2, 'LineSmooth', 'on');
plot(real(z), imag(z), 'bo', 'MarkerSize', 14, 'LineWidth', 2, 'LineSmooth', 'on');
plot( ...
    sin(linspace(0,2*pi,100)), ...
    cos(linspace(0,2*pi,100)), ...
    'k-', 'LineWidth', 2, 'LineSmooth', 'on');

%
%  Display the magnitude/phase plot
%
[ b, a ] = zp2tf(z,p,k);

W = linspace(0,2,100);
H = freqs(b,a, W);

fig = figure;
a1 = axes( ...
    'Parent', fig);
a2 = axes( ...
    'Parent', fig, ...
    'YAxisLocation', 'right', ...
    'Color', 'none');

hold( a1, 'on');
hold (a2, 'on');

xlabel(a1, 'Radians/second');
ylabel(a1, 'Magnitude (dB)');
ylabel(a2, 'Phase (degrees)');

box on;
grid on;

%
%  Plot the magnitude and phase
%
b = plot(a1, W, 10*log10(abs(H)), 'g-', 'LineWidth', 2, 'LineSmooth', 'on');
c = plot(a2, W, unwrap(angle(H))*360/pi, 'r-', 'LineWidth', 2, 'LineSmooth', 'on');

legend([b,c], 'Magnitude', 'Phase');
```

REFERENCES

Havskov, J., Gerardo, A. (2006), *Instrumentation in Earthquake Seismology*, Springer

Holcomb, L.G., *A direct method for calculating instrument noise levels in side-by-side seismometer evaluations*, U.S. Geol. Surv. Open-File Rept. 89-214, 1989.

IEEE Std 100-2000, *The Authoritative Dictionary of IEEE Standards Terms Seventh Edition.*

IEEE Std 181-2003, *IEEE Standard on Transitions, Pulses, and Related Waveforms*.

IEEE Std 1057-1994, *IEEE Standard for Digitizing Waveform Recorders*.

IEEE Std 1241-2000, *IEEE Standard for Terminology and Test method for Analog-to-Digital Converters*.

Kromer, R.P., Hart, D.M., Harris, J.M., *Test Definitions for the Evaluation of Digital Waveform Recorders*, SAND2007-5037, Sandia National Laboratories, Albuquerque, NM, July 2007.

Kromer, R.P., Hart, D.M., Harris, J.M., *Test Definitions for the Evaluation of Infrasound Sensors*, SAND2007-5038, Sandia National Laboratories, Albuquerque, NM, July 2007.

Kromer, R.P., Hart, D.M., Harris, J.M., *Test Definitions for the Evaluation of Seismic Sensors*, SAND2007-5025, Sandia National Laboratories, Albuquerque, NM, July 2007.

McDonald, T.S., *Modified Noise Power Ratio Testing of High Resolution Digitizers*, SAND94-0221, Sandia National Laboratories, Albuquerque, NM, May 1994.

Oppenheim, A.V., Schafer, R.W. (1999), *Discrete-Time Signal Processing*, Prentice Hall.

Poularikas, A.D. (1999), *The Handbook of Formulas and Tables for Signal Processing*, CRC Press LLC.

Scherbaum, F. (2001), *Of Poles and Zeros: Fundamentals of Digital Seismology*, Kluwer Academic Publishers.

Sleeman, R., Wettum, A., Trampert, J. (2006), *Three-Channel Correlation Analysis: A New Technique to Measure Instrumental Noise of Digitizers and Seismic Sensors*, Bulletin of the Seismological Society of America, Vol. 96, No. 1, pp. 258-271, February 2006.

Sterns, S. D. (1990), *Digital Signal Analysis*, Hayden Book Co.

Thomas, R. E., Rosa, J. R. (1994), *The Analysis and Design of Linear Circuits*, Prentice Hall.

Welch, P.D., “The Use of the Fast Fourier Transform for the Estimation of Power Spectra,” *IEEE Trans. Audio ElectroAcoustics*, Vol. AU-15, No. 2, pp. 70-73, June 1967.

DISTRIBUTION

1	MS0404	John Merchant	05736
1	MS0404	Darren Hart	05736
1	MS0404	Tim McDonald	05736
1	MS0671	Mark Harris	05627
1	MS0899	RIM-Reports Management	9532 (electronic)

Sandia National Laboratories