

Tracking Spectral Noise Lines in Advanced LIGO Data

Gillian Dora Beltz-Mohrmann, Wellesley College

Mentors: Alan Weinstein & Jonah Kanner, California Institute of Technology

LIGO SURF 2015

LIGO-T1500415

August 30, 2015

Abstract

The Advanced LIGO detectors are expected to make gravitational wave observations possible within the next few years. However, sharp spectral noise lines continue to obscure the data, and it is unknown if or how these lines wander over time. Therefore, we are developing a method that will track the frequencies of the various noise sources which appear in our data. Using Python for scripting, we utilize various signal processing techniques to identify the exact frequencies of the noise sources present in our time series. We then heterodyne to determine if and how a given spectral line wanders in frequency over time. This technique will provide beneficial insight for improving the quality of the data and the sensitivity to gravitational waves from spinning neutron stars and other astrophysical sources.

1 Introduction

Gravitational waves are ripples in space-time, first predicted by Einstein’s theory of general relativity in 1916. They can be produced by a number of sources, including compact binary coalescences, galactic supernovae, non-axisymmetric spinning neutron stars, and the stochastic gravitational wave background left over from the Big Bang. Now, nearly a century after the existence of gravitational waves was predicted, the Advanced LIGO detectors are expected to finally make gravitational wave observations possible.

LIGO is designed to detect gravitational waves using a laser interferometer, which measures with high precision the time it takes light to travel between suspended mirrors¹. If a gravitational wave passes by, the distance measured by the light will change, and a photodetector will produce a signal. The LIGO instruments are modeled after Michelson interferometers with Fabry-Perot arm cavities. The detectors are operated in coincidence at the different locations to rule out false signals. The first generation of LIGO ran from 2002 to 2010, and although it did not detect anything, it proved that the experiment was technologically possible². Advanced LIGO is designed to be ten times more sensitive due its many technical improvements, and should soon be able to detect gravitational waves. However, the data are confounded by various sources of noise.

We can examine LIGO data with amplitude spectral density (ASD) plots, which describe how the power of a signal or time series is distributed over different frequencies. From these plots (eg. Figure 1), we can see that there are two components to LIGO strain noise: a broadband component and sharp spectral lines. One of the sources of the largest line features is the electrical power

system, which operates at a frequency of 60 Hz and which produces spectral lines in the LIGO strain data at 60 Hz and all multiples (harmonics). Another chief source is the vibration of the suspension fibers which hold up the mirrors due to thermal noise. These noise lines, of which there are about a dozen, are called violin modes, and they typically produce spectral lines in the data around 500 Hz (and all harmonics). There are also line features from calibration; these are deliberately placed in the data by moving the mirrors. (These lines are very close to pure sinusoids, so their harmonics do not show up noticeably in the data.) There are also line features produced by parametric instabilities³. We would like to determine if and how the frequencies of the lines drift.

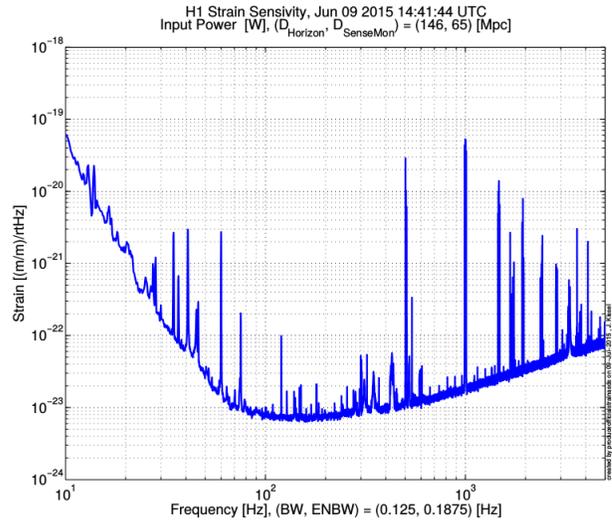


Figure 1: An example of a typical amplitude spectral density plot of LIGO strain data: H1 Calibrated Sensitivity Spectra Jun 9 2015 (Best for ER7)⁵

These spectral lines pose a particular chal-

lence for the detection of non-axisymmetric spinning neutron stars, since signals from such sources also appear as spectral lines in our ASDs. (The gravitational wave from a non-axisymmetric spinning neutron star would have a frequency of twice the rotation rate of the star.) For example, the Crab Pulsar spins with a frequency of 29.65 Hz⁸, so a gravitational wave produced would have a frequency of 59.3 Hz, which is tantalizingly close to the 60 Hz power line. If the power line drifted, it would be even more of a challenge to distinguish the noise from the signal. Therefore, it is very important to understand how the noise lines behave. Thus, we are developing a method for tracking the frequencies of these noise sources.

2 Methods

2.1 Scripting and Data

We use Python¹⁰ for scripting, making extensive use of a package called GWpy⁶, which provides specific tools for studying data from gravitational-wave detectors. Preliminary data were taken from Advanced LIGO’s ER7 run (specifically June 7, 2015), from both the Livingston (L1) and Hanford (H1) detectors.

2.2 Frequency identification

In order to identify the frequencies present in our time series, we generate a high resolution power spectral density (PSD) of our data using numpy⁹ and matplotlib⁴. (See Figure 2.) A PSD is simply the square of an ASD; it also illustrates how a signal’s power is distributed over different frequencies. In order to reduce spectral leakage, a Blackman window was applied. We use this

PSD to determine the exact frequencies of our spectral noise lines.

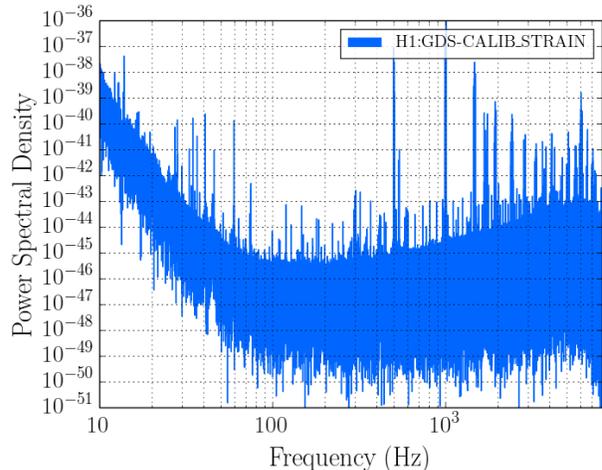


Figure 2: Power spectral density of strain data from the H1 detector on June 7, 2015 (ER7).

2.3 Heterodyning

Heterodyning is a signal processing method that involves multiplying an incoming signal (in our case, gravitational wave strain, $h(t)$) by a designed signal (in our case a complex exponential) to isolate a desired frequency. The complex exponential is given by $H(g, t) = e^{2i\pi gt}$, where g is chosen to be the frequency of one of the lines identified from the PSD. We then integrate over time and divide by the length of time to get an average: $Z = \frac{1}{T} \int_0^T h(t)H(g, t)dt$. When we do this for a stretch of data, Z becomes $Z(t)$. This technique allows us to pick out frequencies close to the heterodyne frequencies, suppressing all others because the heterodyne averages to zero for them. (See Appendix for details.)

The output of the heterodyne is a complex number, which can be separated into magnitude

and phase components. We examine the heterodyne magnitude and phase plots to determine how the signal frequency changes over time.

The magnitude, as shown in Figure 2, decreases as the difference between the heterodyne frequency and signal frequency increases. The bandwidth of this magnitude is equal to $\frac{2}{T}$; thus, the longer the heterodyne time, the narrower the bandwidth. (In this particular example, the heterodyne time is one second, so the bandwidth is 2 Hz.)

The heterodyne phase can be used to track frequencies within the bandwidth, as depicted in Figure 4. It is important to note that if the signal frequency does not change over time, the phase should not change over time.

If the signal frequency does change over time, both the magnitude and phase will change over time. The magnitude should peak when the signal frequency and heterodyne frequency are equal (see Figures 5 and 7), while the phase should have a slope that depends linearly on the frequency difference (see Figures 6 and 8). Unfortunately, the sign of the slope does not depend on whether the signal frequency increases or decreases. We have not determined yet why the slope that we see is not linear.

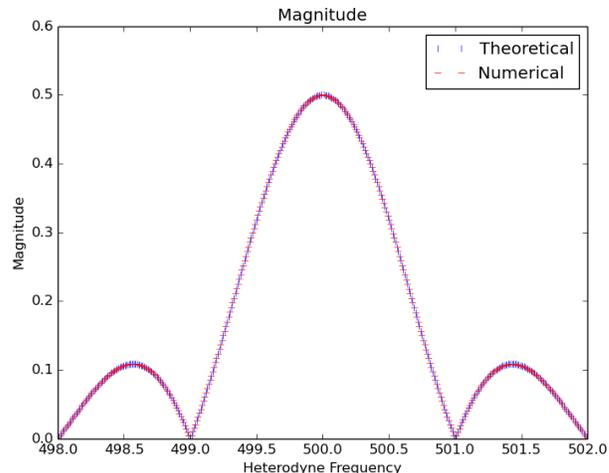


Figure 3: Theoretical and numerical calculations of heterodyne magnitude for various frequencies.

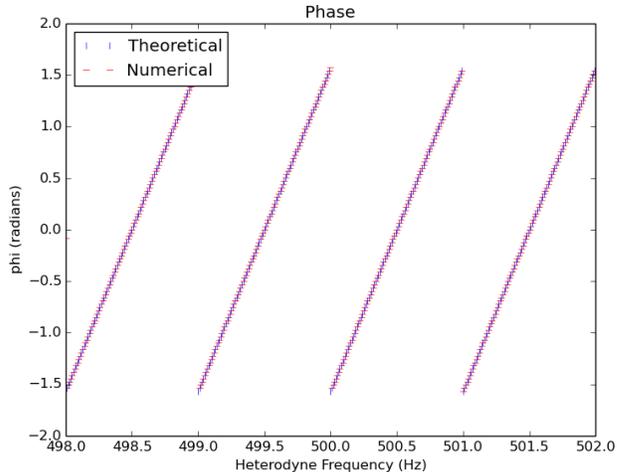


Figure 4: Theoretical and numerical calculations of heterodyne phase for various frequencies. We use unwrap to put all phase points between $-\pi/2$ and $\pi/2$.

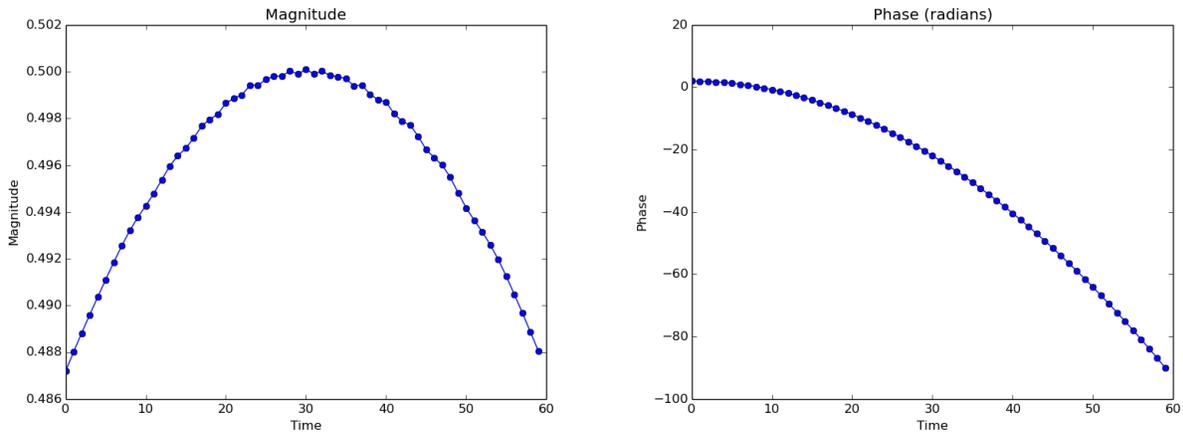


Figure 5: Magnitude for a signal frequency that increases from 500 to 500.25 Hz over 60 seconds; heterodyned at 500.125 Hz. Figure 6: Phase for a signal frequency that increases from 500 to 500.25 Hz over 60 seconds; heterodyned at 500.125 Hz.

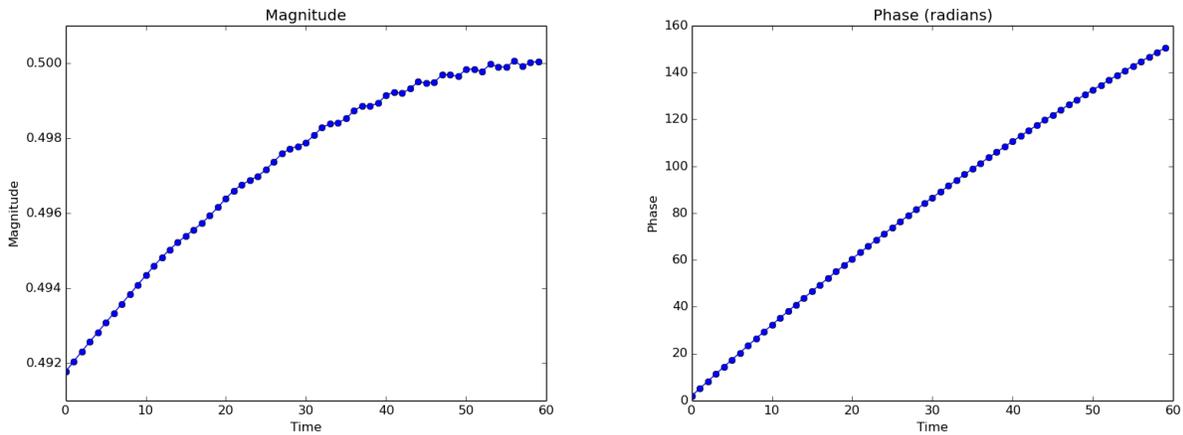


Figure 7: Magnitude for a signal frequency that increases from 505.5 to 505.6 Hz over 60 seconds; heterodyned at 505.6 Hz. Figure 8: Phase for a signal frequency that increases from 505.5 to 505.6 Hz over 60 seconds; heterodyned at 505.6 Hz.

3 Results

One of the first conclusions that can be drawn from the heterodyne phase plots is that the two detectors are quite different. It appears that the H1 power line wanders by about 0.01 Hz steadily over 4096 seconds (Figure 9), while the L1 power line jumps in frequency sporadically (Figure 10). (It is also possible that Python’s unwrap function is causing this effect, because all of the jumps are 2π Hz. This requires further investigation.)

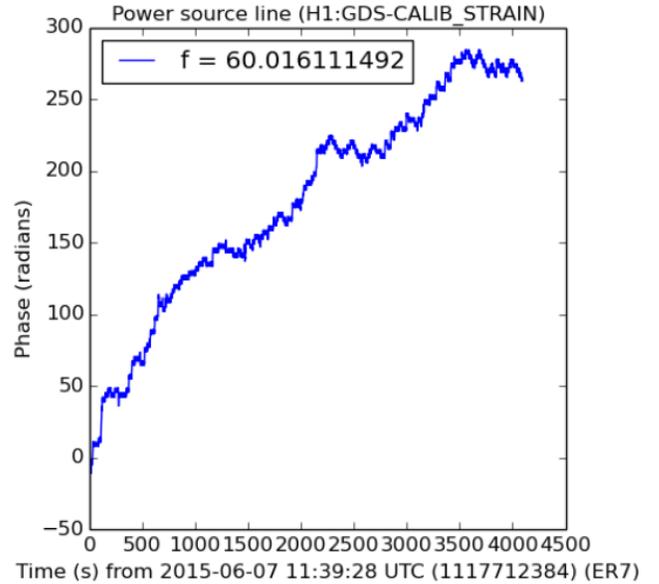


Figure 9: Heterodyne phase of the H1 power line.

We do not expect the photon calibrator lines to wander, so it is surprising to see that their phases have non-zero slopes, indicating that they do in fact wander. The H1 low frequency photon calibrator lines all seem to wander by about 0.14 Hz over 4096 seconds (Figure 11), while the L1 low frequency lines stay relatively constant (Figure 12). The H1 results are suspicious, and require further investigation.

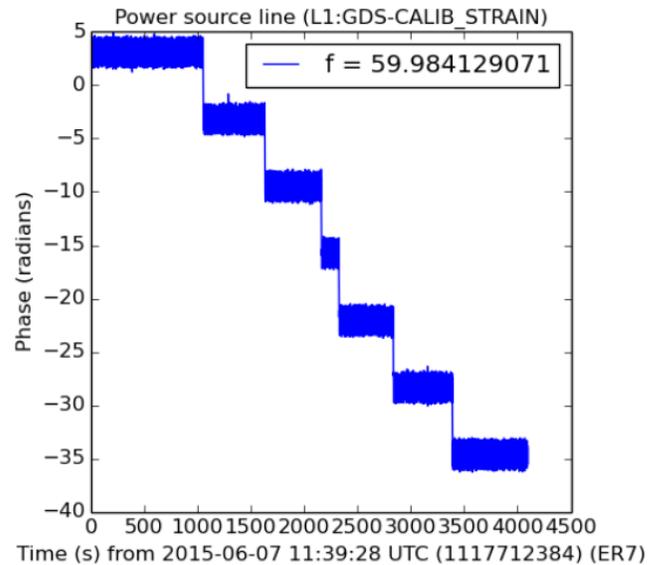


Figure 10: Heterodyne phase of the L1 power line.

Figures 13 and 14 show the first few violin mode lines for the H1 and L1 detectors. Three of the H1 lines appear to wander by 0.4 Hz over 4096 seconds, while the fourth wanders by 0.2 Hz. Two of the L1 lines seem to wander by 0.05 Hz over 4096 seconds, while the third wanders by 0.09 Hz. This is a reasonable result, but further investigation is needed to confirm its validity.

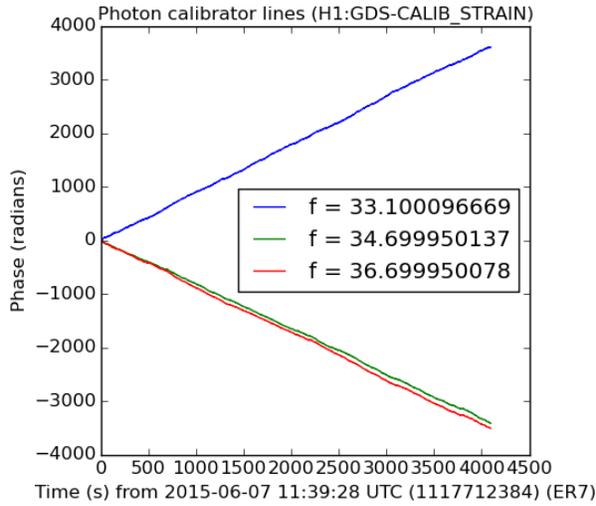


Figure 11: Heterodyne phase of the H1 low frequency photon calibrator lines.

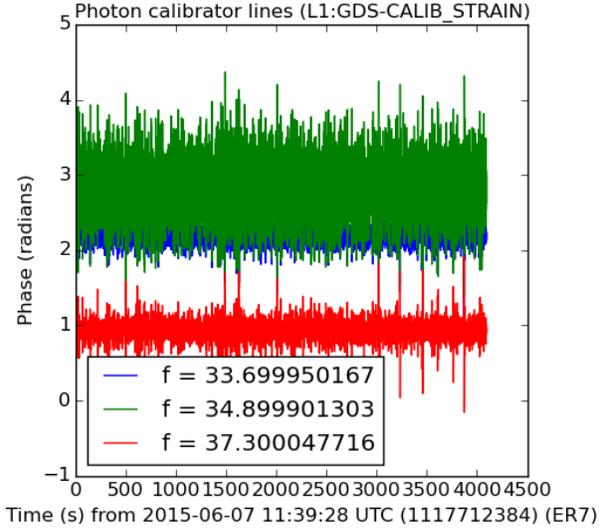


Figure 12: Heterodyne phase of the L1 low frequency photon calibrator lines.

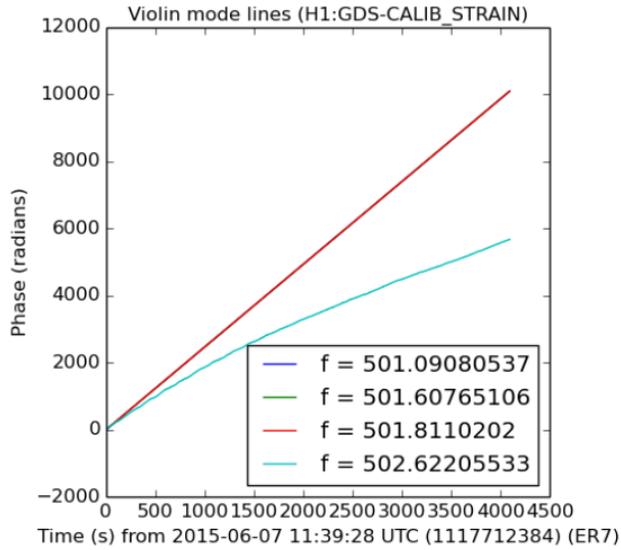


Figure 13: Heterodyne phase of the first several H1 violin mode lines.

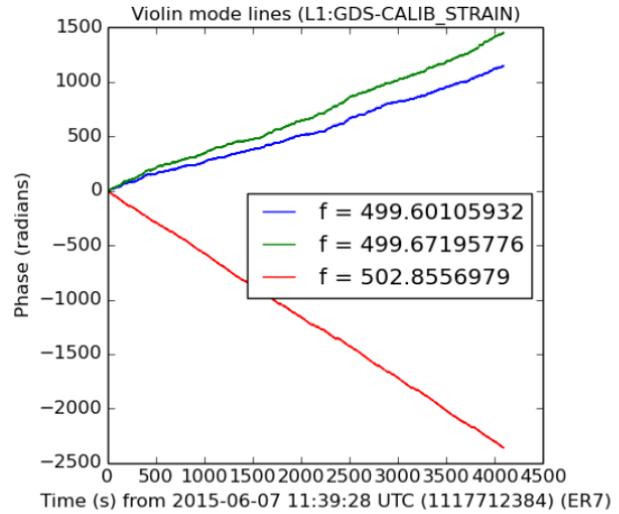


Figure 14: Heterodyne phase of the first several L1 violin mode lines.

4 Conclusions

We have found that heterodyning is an efficient and effective method for spectral line tracking. Our future goals are to understand precisely the movement of the lines, to test different heterodyne times, to track additional frequencies, and to calculate the error on our results. We intend to automate the process to constantly analyze new data minute by minute in quasi-real time. If a line drifts out of the heterodyne bandwidth, part of the automated process will be to compute a new frequency for the next execution. The results will be displayed in the control room, as well as sent to commissioners to help them improve the detectors, and to data analysts so that they can be more immune to noise effects in astrophysical searches.

5 Acknowledgments

I gratefully acknowledge the support of the United States National Science Foundation for the construction and operation of the LIGO Laboratory, as well as California Institute of Technology Student-Faculty Programs Office, the Research Experiences for Undergraduates Program of the National Science Foundation, and the LIGO Summer Undergraduate Research Program. I also acknowledge the use of the LIGO Open Science Center (<https://losc.ligo.org>), a service of LIGO Laboratory and the LIGO Scientific Collaboration, Python version 2.7, developed by the Python Software Foundation (available at <http://www.python.org>), and GWpy, created using Sphinx 1.2.2 by Duncan Macleod (2013). I especially thank Dr.'s Alan Weinstein and Jonah Kanner for all of their help and guidance.

References

- [1] "Science of LIGO." LIGO Livingston. 07 May 2015. <http://www.ligo-la.caltech.edu/LLO/overviewsci.htm>
- [2] A. Cho. Feature: Physicists gear up to catch a gravitational wave. *Science*, 2015.
- [3] Evans, M. et al. Observation of Parametric Instability in Advanced LIGO. *Phys. Rev. Lett.* (2015) 114, 161102.
- [4] Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9, 3, 90-95 (2007), IEEE COMPUTER SOC.
- [5] Kissel, J. H1 Calibrated Sensitivity Spectra Jun 9 2015 (Best for ER7). LIGO Document Control Center. 23 June 2015.
- [6] Macleod, D. GWpy. 2013. <https://gwpy.github.io/>
- [7] Macleod, D. Summary Pages. <https://ldas-jobs.ligo-la.caltech.edu/detchar/summary/day/20150803/>
- [8] Lyne, A. et al. 45 Years of Rotation of the Crab Pulsar. *MNRAS* (2015) 446, 857-864.
- [9] Stfan van der Walt, S. Chris Colbert and Gal Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, *Computing in Science & Engineering*, 13, 22-30 (2011).
- [10] Travis E. Oliphant. Python for Scientific Computing, *Computing in Science & Engineering*, 9, 10-20 (2007).

6 Appendix

Here we provide a mathematical explanation of the heterodyne technique as applied to our data.

First, we expand the data $h(t)$ into fourier components:

$$h(t) = \int_{-\infty}^{\infty} \tilde{h}(f)e^{i2\pi ft} df. \quad (1)$$

The data can separated into broadband noise and a signal:

$$\tilde{h}(f) = \tilde{n}(f) + \tilde{s}(f). \quad (2)$$

Thus Equation 1 can be rewritten as

$$h(t) = \int_{-\infty}^{\infty} \tilde{n}(f)e^{i2\pi ft} df + \int_{-\infty}^{\infty} \tilde{s}(f)e^{i2\pi ft} df. \quad (3)$$

Our heterodyne can be written as

$$H(g, t) = e^{i2\pi gt} \quad (4)$$

where we have chosen 1 as our amplitude.

When we heterodyne, we multiply $H(g, t)$ by $h(t)$ and integrate from 0 to T, and then divide by T.

$$\begin{aligned} Z &= \frac{1}{T} \int_0^T h(t)H(g, t)dt \\ &= \frac{1}{T} \int_0^T \left(\int_{-\infty}^{\infty} \tilde{h}(f)e^{i2\pi ft} df \right) H(g, t)dt \\ &= \frac{1}{T} \int_0^T \left(\int_{-\infty}^{\infty} \tilde{n}(f)e^{i2\pi ft} df \right) H(g, t)dt + \frac{1}{T} \int_0^T \left(\int_{-\infty}^{\infty} \tilde{s}(f)e^{i2\pi ft} df \right) H(g, t)dt \\ &= \frac{1}{T} \int_0^T \left(\int_{-\infty}^{\infty} \tilde{n}(f)e^{i2\pi ft} df \right) e^{i2\pi gt} dt + \frac{1}{T} \int_0^T \left(\int_{-\infty}^{\infty} \tilde{s}(f)e^{i2\pi ft} df \right) e^{i2\pi gt} dt \end{aligned}$$

The broadband noise should be evenly distributed at all frequencies above and below the heterodyne frequency, so it will average to zero, while the signal goes from a delta function in frequency to a peak with some width. Thus, heterodyning effectively suppresses the broadband noise at all frequencies except those near the heterodyne frequency.

We then want to solve the integral

$$Z = \frac{1}{T} \int_0^T \left(\int_{-\infty}^{\infty} \tilde{s}(f)e^{i2\pi ft} df \right) e^{i2\pi gt} dt. \quad (5)$$

Assuming that the fourier component of the signal is a pure sinusoid with amplitude A for simplicity, we have

$$Z = \frac{1}{T} \int_0^T A \sin(2\pi ft) e^{i2\pi gt} dt. \quad (6)$$

Taking advantage of Euler's formula, this integral can be rewritten as

$$Z = \frac{1}{T} \int_0^T \frac{A}{2i} (e^{2i\pi ft} - e^{-2i\pi ft}) e^{i2\pi gt} dt. \quad (7)$$

This can be expanded to get

$$Z = \frac{A}{2Ti} \int_0^T e^{i2\pi gt} e^{2i\pi ft} dt - \frac{A}{2Ti} \int_0^T e^{i2\pi gt} e^{-2i\pi ft} dt. \quad (8)$$

This can be further simplified so that

$$Z = \frac{A}{2Ti} \int_0^T e^{2i\pi t(g+f)} dt - \frac{A}{2Ti} \int_0^T e^{2i\pi t(g-f)} dt. \quad (9)$$

Then, because we can assume that oscillatory functions integrate to zero if T is much larger than a period, the sum terms disappear, and only the difference terms remain. Thus, we are left with

$$Z = \frac{-A}{2i} \int_0^T e^{2i\pi t(g-f)} dt. \quad (10)$$

Evaluating the integral, we get

$$Z = \frac{-A}{2i} \frac{e^{2i\pi t(g-f)}}{2i\pi(g-f)} \Big|_0^T. \quad (11)$$

Taking the limits, we are left with

$$Z = \frac{A(e^{2i\pi T(g-f)} - 1)}{4\pi(g-f)}. \quad (12)$$

This can be simplified further to

$$Z = \frac{2Ai \sin(\pi T(g-f)) e^{i\pi T(g-f)}}{4\pi(g-f)}. \quad (13)$$

To find the magnitude, we want

$$r = (a^2 + b^2)^{1/2}, \quad (14)$$

where

$$a = \text{Re} \left(\frac{2Ai \sin(\pi T(g-f)) e^{i\pi T(g-f)}}{4\pi(g-f)} \right) = \frac{-2A \sin^2(\pi T(g-f))}{4\pi(g-f)} \quad (15)$$

and

$$b = \text{Im} \left(\frac{2Ai \sin(\pi T(g-f)) e^{i\pi T(g-f)}}{4\pi(g-f)} \right) = \frac{2A \sin(\pi T(g-f)) \cos(\pi T(g-f))}{4\pi(g-f)}. \quad (16)$$

Thus

$$a^2 = \frac{4A^2 \sin^4(\pi T(g-f))}{16\pi^2(g-f)^2}, \quad (17)$$

and

$$b^2 = \frac{4A^2 \sin^2(\pi T(g-f)) \cos^2(\pi T(g-f))}{16\pi^2(g-f)^2}. \quad (18)$$

Therefore

$$r = \left(\frac{4A^2 \sin^4(\pi T(g-f)) + 4A^2 \sin^2(\pi T(g-f)) \cos^2(\pi T(g-f))}{16\pi^2(g-f)^2} \right)^{1/2}. \quad (19)$$

This can be simplified to

$$r = \left(\frac{A^2 \sin^2(\pi T(g-f)) (\sin^2(\pi T(g-f)) + \cos^2(\pi T(g-f)))}{4\pi^2(g-f)^2} \right)^{1/2} \quad (20)$$

or

$$r = \left(\frac{A^2 \sin^2(\pi T(g-f))}{4\pi^2(g-f)^2} \right)^{1/2} \quad (21)$$

which gives

$$r = \frac{A \sin(\pi T(g-f))}{2\pi(g-f)}. \quad (22)$$

Or, letting $g = f + \Delta f$, we have

$$r = r = \frac{A \sin(\pi T \Delta f)}{2\pi \Delta f}. \quad (23)$$

To find the phase, we want

$$\phi = \text{atan} \frac{b}{a}, \quad (24)$$

which equals

$$\phi = \text{atan} \left(\frac{2A \sin(\pi T(g-f)) \cos(\pi T(g-f))}{-2A \sin^2(\pi T(g-f))} \right) \quad (25)$$

or

$$\phi = \text{atan} (-\cot(\pi T \Delta f)). \quad (26)$$

Next, we want to know how the phase varies over time. Taking a time derivative, we get

$$\frac{d\phi}{dt} = \pi \Delta f. \quad (27)$$