Exploring Digital Sampling, Fourier Transforms, and both DSB and SSB mixers

Arthur Martirosyan Compas Professor Aaron Parsons

December 6, 2018

Abstract

This lab serves as an introduction to the undergraduate radio laboratory at the University of California Berkeley. Lab 1 is used as an introduction to the various tools and techniques we will employ during our observations including Nyquist sampling, heterodyne mixers, fourier analysis, and waveform reconstruction. Specifically we will focus on the sampling of a sine wave at different Nyquist frequencies, which will lead us to conclude that the Nyquist frequency is equal to one half of the sampling frequency. We will also gain an understanding of the limitations of our heterodyne mixer.

Introduction

We will sample signals at regular intervals with our computers, a process known as "discrete sampling", and our computers will change the voltages to digital numbers, a process known as "digitization". In the Lab professor Parsons discussed the process of inducing a potential within a wire to produce a signal which we can then process. We will become acquainted with aliasing and the basic law of sampling: the Nyquist criterion. Then we implement ugradio's Discrete Fourier Transform package to determine the frequency power spectrum of our time series. To determine the limitations of our setup, we need to understand leakage power and the frequency resolution when sampling a single sine wave. This includes analyzing the Fourier transform of a signal with pure noise, which is very important when dealing with radio astronomy. We will also discuss the role of the power spectrum. We will then discuss our frequency limitations and the implementation of mixing using the heterodyne technique. We also discuss the differences between real and ideal mixers, constructing a SSB and DSB mixer.

1 The Nyquist Criterion

1.1 Discretely-Sampled Signals

In discrete sampling, we first require that the signal be limited in bandwidth. That is, the highest frequency in its spectrum must be limited to an upper cutoff, which we call the bandwidth B; thus, the frequencies in the signal extend from 0 to B Hz. Then we must sample the signal periodically at a fast enough rate—specifically, we require ($\nu_{smpl} \geq 2B$ Hz). For our method we are using a Pico Technology PicoScope 2000 series which is a dual channel usb-powered oscilloscope, spectrum analyzer, and a function generator with a bandwidth of 10 MHz. Our analog signal comes from a Keithley Signal generator and we are verifying our signal using a Rigol oscilloscope. We are using the PicoSampler with UGRADIO to read our data into our into our local machines using the packages written by professor Parsons. The pico has a configurable sampling rate of (62.5/N)MHz and a max input of 20v. The pico then takes our analog input and converts it to a signed 16-bit integer digital signal. We are sampling at a rate of N = 5 which gives us a sampling frequency of 12.5MHz at 1V peak to peak. We then appropriately scale our output by the maximum value of that bit-space. Then we can convert our voltages to milivolts. As a sanity check we are using the Rigol scope to cross-reference our recorded datapoints. our analysis.

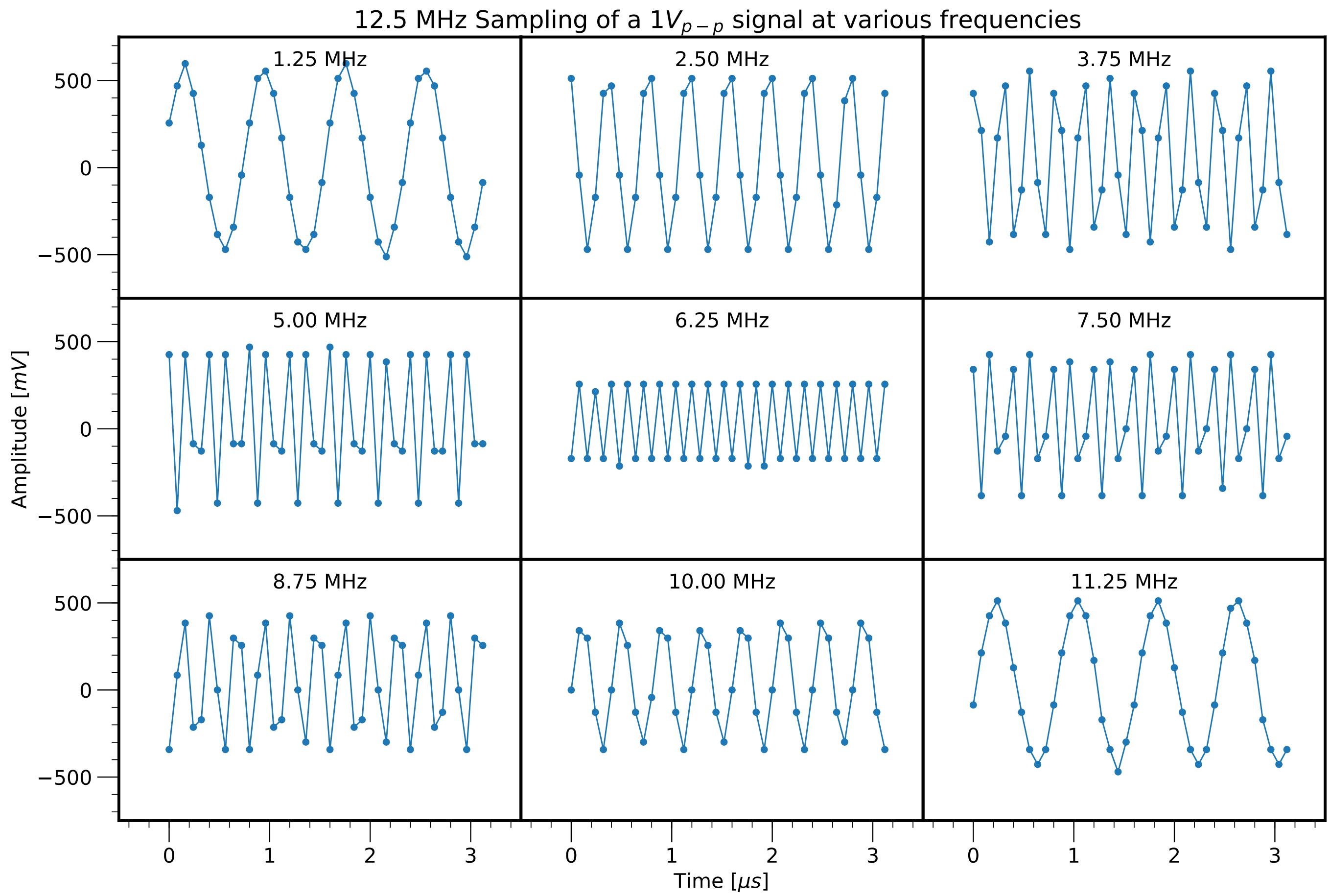

Figure 1: Aliasing in the Power Spectrum

1.2 Nyquist frequency in the Time Series

Figure 1 represents our recorded discrete samples at different frequencies, increasing by 1.25MHz. We can see our 6.25 MHz signal which is equal to our ν_{smpl}. All of the frequencies below by observation have the correct corresponding signal amplitude and peaks represented, but as we increase in frequency we begin to see odd and mysterious phenomena happening. Is this professor Parsons plan to sabotage our data? Or perhaps this could be aliasing we are seeing! We notice that higher frequencies don't look right and seem to be decreasing in frequency. In, fact is is aliasing that we are seeing. This tells us that if we are sampling a 6.25 MHZ signal, we at least need to be at double the frequency to resolve a proper spectra. This is the "Nyquist criterion". If the Nyquist criterion is violated, we have the problem of *aliasing*, which means that signals with frequencies $\nu > B$ *appear* as *lower* frequencies.

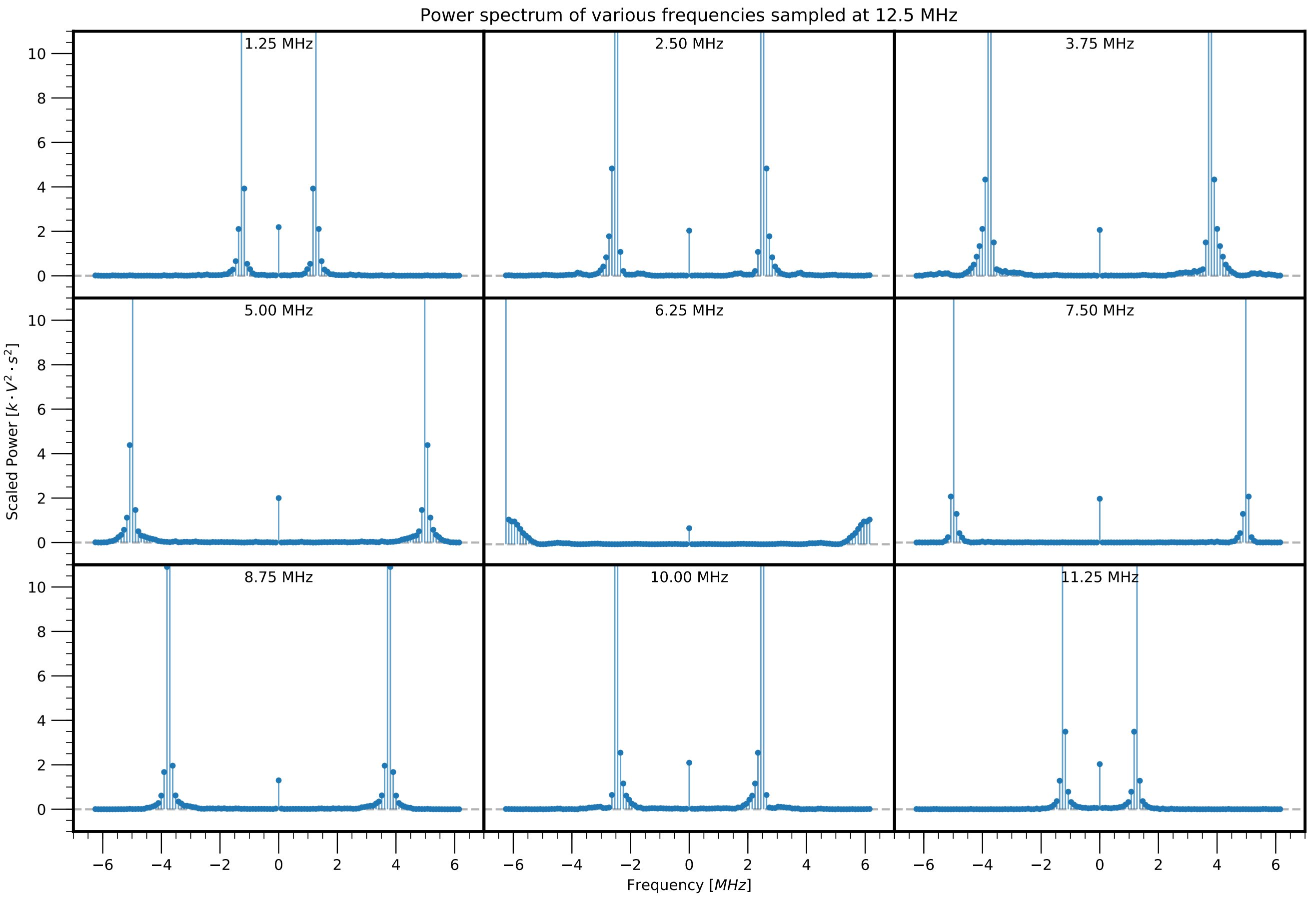

Figure 2: Aliasing in the Power Spectrum

1.3 Nyquist frequency in the Power Spectrum

We use the Fourier transform to convert from the time to the frequency domain, and *vice versa*. For continuously-sampled signals we use the Fourier integral, equation 1. For discretely-sampled signals we have to replace this by a summation. Figure 2 represents our discrete samples from figure 1 after we have preformed a fast fourier transform using the Numpy FFT function. Comparing figure 1 to 2, we see a similiar effect as before, where past a certain frequency, larger frequencies appear to be smaller. For example, the 1.25 MHz frequency seems to be identical to the 11.25 MHz frequency. We will come to find out how we can later leverage this effect in the lab.

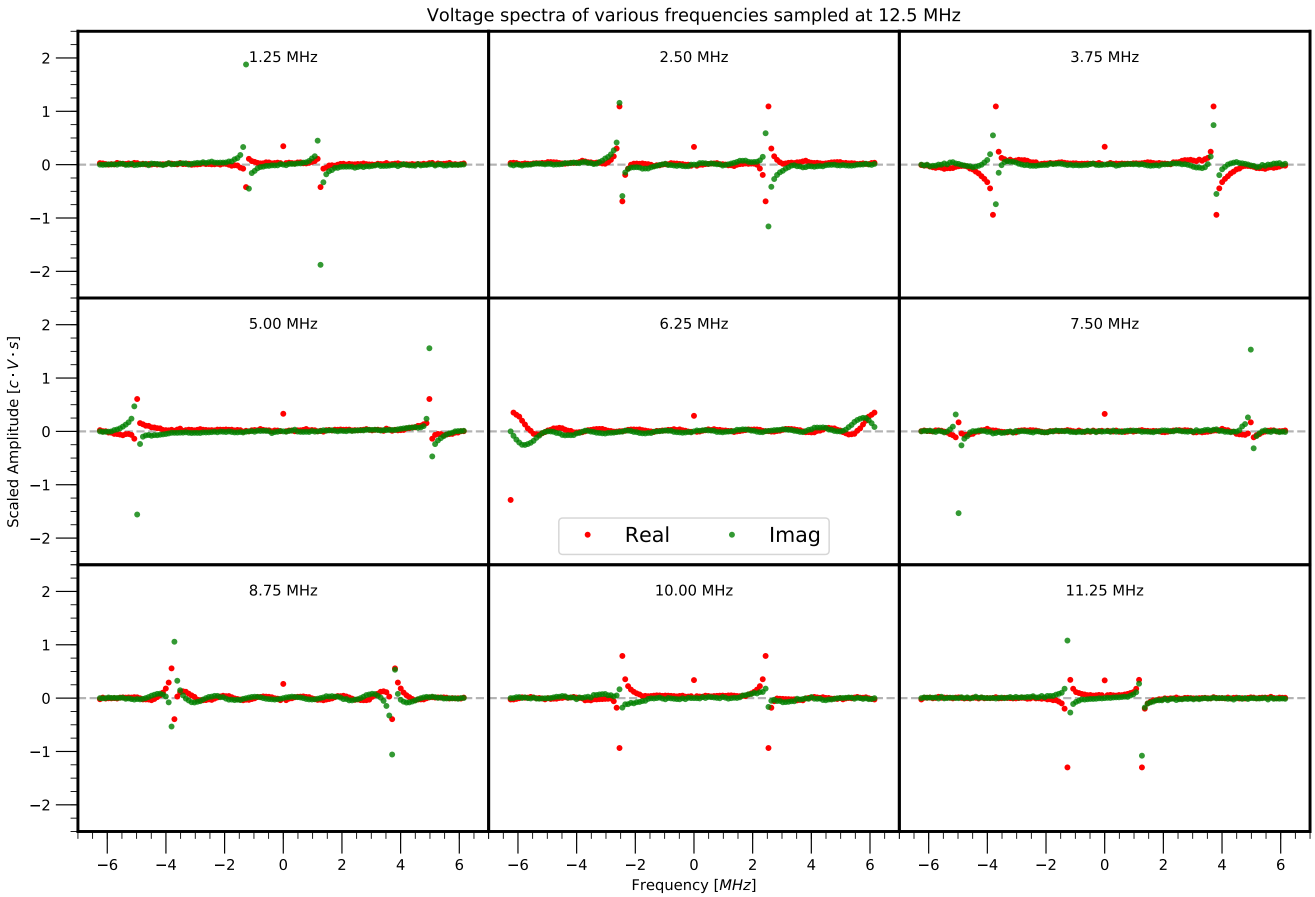

Figure 3: Aliasing in the Power Spectrum

1.4 Aliasing in Voltage Spectra

Figure 3 represents our Voltage time spectrum. Using UGRADIO we were able to separate the real and imaginary components of the Power Spectrum, and then overlay ed them in figure 3. When we are dealing with taking complex conjugates, the imaginary portion of the signal will be flipped. Looking at our plots, the real portions of the corresponding frequencies are reflected symmetrically. Meaning the real portion of our 1.25 MHz signal looks identical to our 11.25 MHz signal. However the imaginary portion of the signal will have its y axis inverted. Meaning the imaginary spectrum of 1.25 MHz will be equal to the inverted imaginary spectrum of 11.25 MHz. We do see some correctly inverted imaginary spectrum's however some of them do not look inverted. This may be from either not centering our frequency time data properly or this could also be from noise in our power spectra. Also the small wiggles at 0 arise from not taking exactly one period of the signal when computing the Fourier transform. That is just from shear luck as professor Parsons described in lecture.

2 Frequency Space

This is the *discrete Fourier transform*, or DFT [1], which allows us to switch from a time domain to a frequency domain. In discretely-samples signals we have to replace this by a summation.

$$E(\nu) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{+T} E(t) e^{[2\pi i]\nu t} dt \ . \tag{1}$$

2.1 Understanding the Voltage Spectra

The reason why the voltage spectra is complex is because we are multiplying our real time-series values by the complex exponent during the transform. This is why our plots display both an imaginary and real component. The complex exponential is defined as:

$$e^{[2\pi i]\pm\nu t} = \cos(2\pi\nu t) \pm i\sin(2\pi\nu t) \tag{2}$$

If you have ever taken a college level calculus course you will have in fact plotted complex values in the imaginary plane. The sign of our terms will alternate as ν changes. This is why we have both negative and positive frequency terms as well as negative magnitudes. Also when we take the convolution of our two terms, the sign of the sine term will flip, while our cosine term will stay the same. This is a display of Hermitian symmetry.

2.2 Understanding the Power Spectra

We are often interested in the *power spectrum*. Power goes as amplitude squared, so the power spectrum $P(\nu) \propto E(\nu)^2$. We can take the complex conjugate of the transform to obtain our power. This eliminates complex exponents, leaving real and positive components.

$$P(\nu) = E(\nu) \times [E(\nu)]^* \ , \tag{3}$$

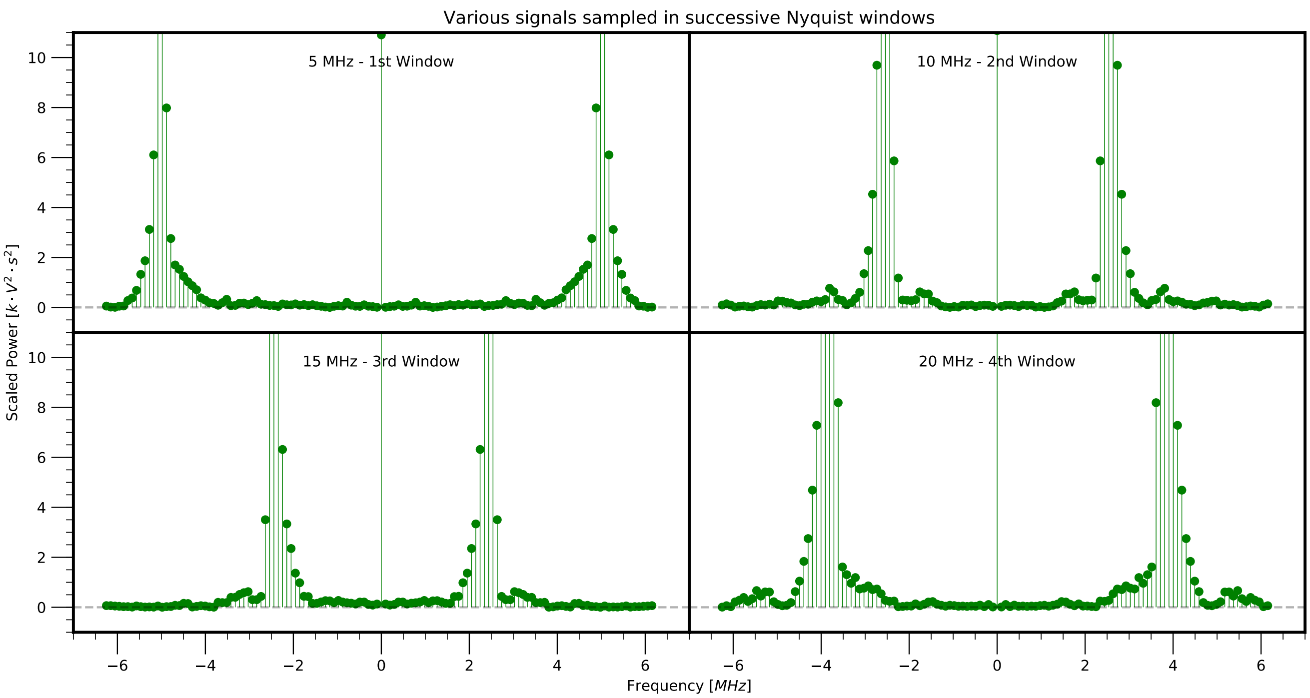

Figure 4: Aliasing in the Power Spectrum

‘

2.3 Nyquist Windows

We already know that when sampling at rate ν_{smpl} you cannot recover a spectrum wider in bandwidth than $f_N = \frac{\nu_{smpl}}{2}$. Our sampling frequency is 12.5 MHz and we sample signals from 5 - 20 MHz. we generated signals in regions that are above their Nyquist frequency. Meaning that when we look at the 20 MHz signal, we are actually looking at a 4 MHz reading. These are known as Nyquist windows. We computed their power spectra's and have plotted them in figure 4.

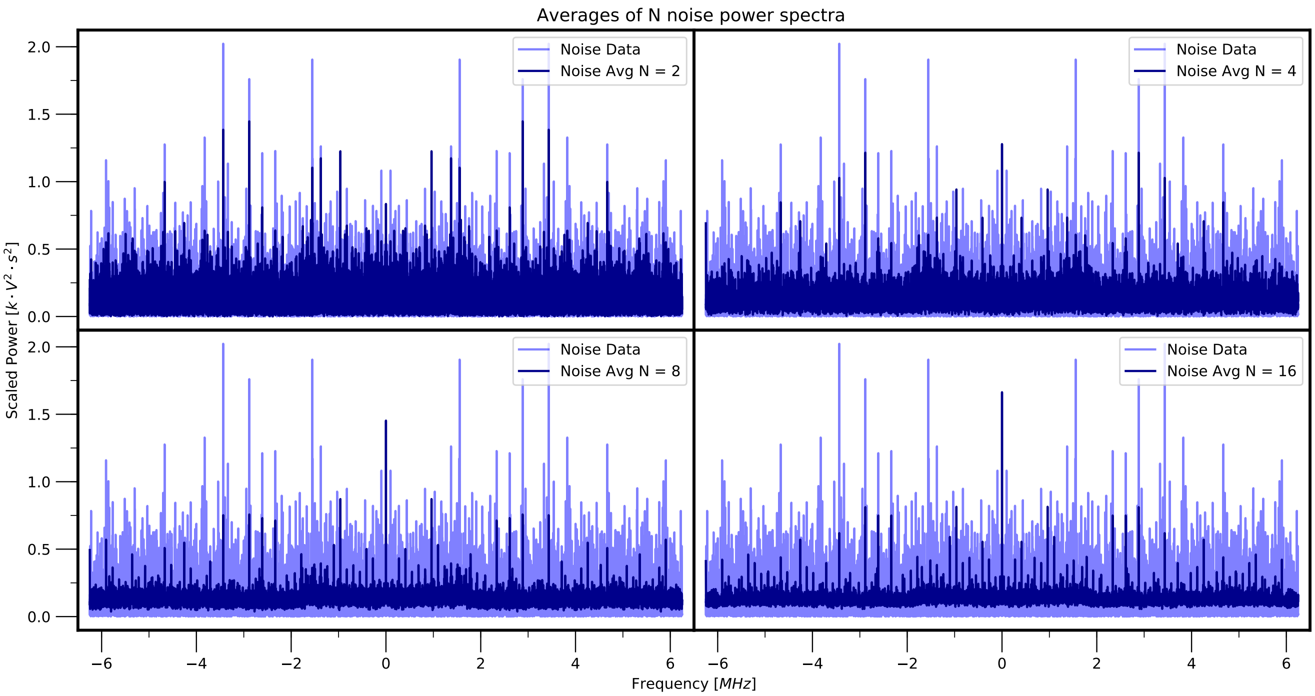

Figure 5: Aliasing in the Power Spectrum

2.4 FT's of Noise

Noise is a large topic in radio astronomy because It is something inherent in the process of collecting radio waves. Although there is a clever work around it, but It is important to always understand the role noise plays in your measurements and to be cognizant about it. Here we are sampling 16000 points of noise data collected from the Keithley function generator. Here we have not used the Micronetics 5250 noise generator or the Minicircuits SBP-21.4 6 Mhz band-pass filter. This is the reason why we are not seeing frequency peaks, or humps, in our power spectrum plots in figure 5. We are taking a total of N= 32 blocks of data, and as we increase in sample size, the less noise we get, and our SNR drastically improves. This shows us that we can leverage noise to not completely get rid of it but reduce it to a manageable level for proper SNR values.

3 In The Lab: Mixers

An analog mixer is a device that takes in two current-limited supplies and combines them to produce a third signal. Some mixers can be passive, while others are active and require power inputs. What mixers allow us to do is sample at frequencies well above a given Nyquist frequency so that our equipment at the lab can accurately read it. The mixer has two inputs, the RF, or high input for the frequency we are sampling and a LO (local oscillator) or low input to input a signal of a known frequency. In this lab we are using and discussing two mixers: the Double Sideband Mixer, and the Sideband-Separating Mixer. ‘

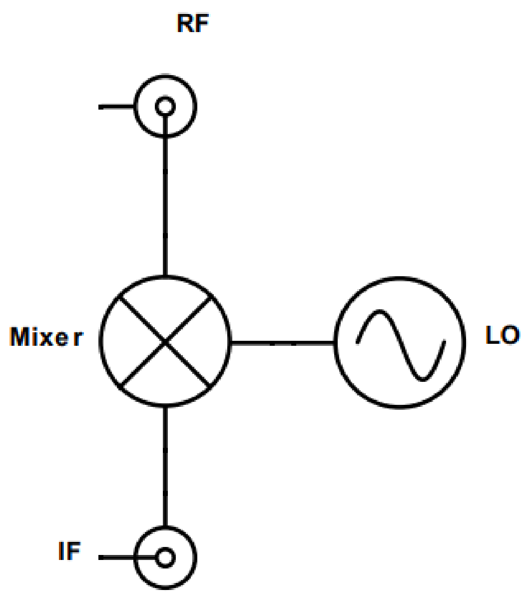

Figure 6: Double sideband mixer

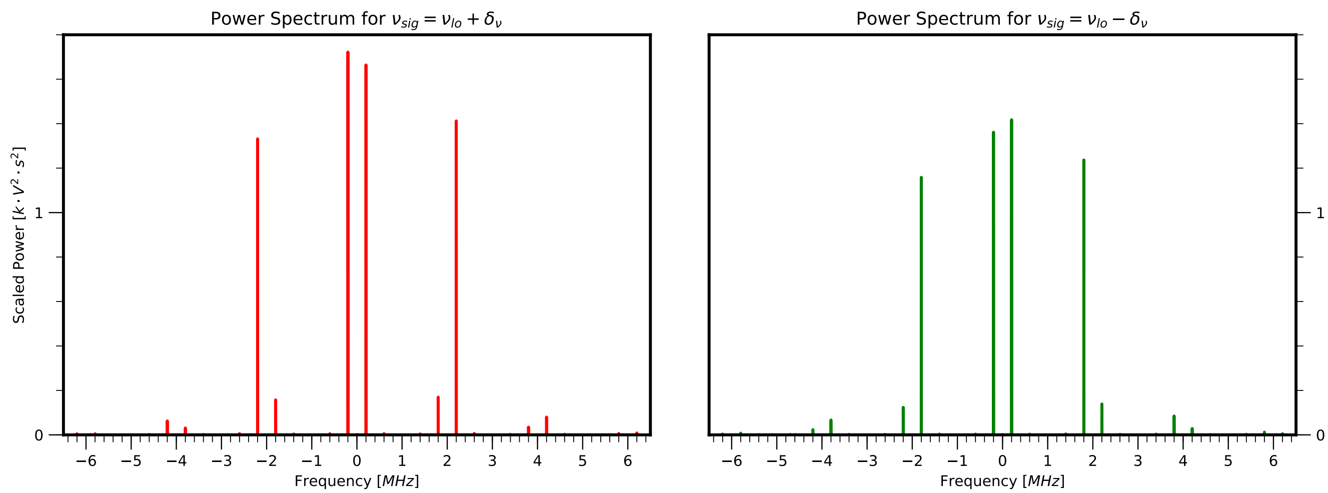

Figure 7: Double sideband mixer

3.1 The Double-Sideband Mixer

Here we are using a double side band mixer to sample two frequencies inputted to our single mixer. The mixers provided in class are the Mini-Circuits ZAD-1 mixers. Before the lab begun we went ahead and looked up the data sheet for the mixer online. We set up the mixer with both inputs, checking our data with the Rigol scope. We input two signals to our mixer, a ν_{sig} and a a ν_{low}. The two signals are in phase and have an amplitude of 500mV, but they differ in frequency by 0.20 MHz. We are sampling at 12.5 MHz and are using cables of equal length to prevent a phase shift. For both high and low cases we compute their power spectrum's as seen above in figure 7. Our signal values were 0.8 and 1.2 MHz while our ν_{low} was fixed at 1 MHz. We can clearly see the two middle peaks have a difference, and this is equal to the difference in frequencies of both signals. On the right side we see a combination of the two frequencies in the upper side band and the lower side band we see the sum of the negative frequencies.

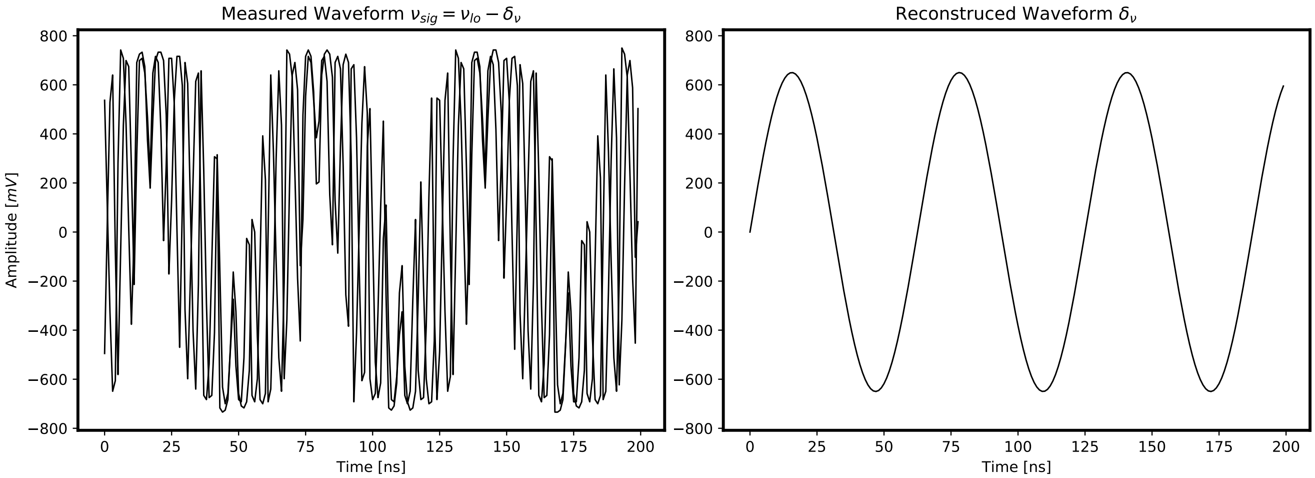

Figure 8: Double sideband mixer

3.1.1 Reconstructed Waveform

Using our power spectra's from above, our task was to recreate our $\delta\nu$. We did this by setting the higher frequency components to zero and then used the NUMPY IFFT function to compute the inverse Fourier transform. This gave us a reconstructed waveform to the right in figure 8. It is important to notice this waveform has a frequency equal to $\delta\nu$, not ν_{low}.

3.2 The Sideband-Separating Mixer

The second type of mixer we used is the Sideband-Separating mixer, also known as the Single-Sideband Mixer. This is using two mixers and it is used to remove the upper sideband signal from the previously calculated power spectrum's from above. It also should allow us to get a more accurate frequency readi

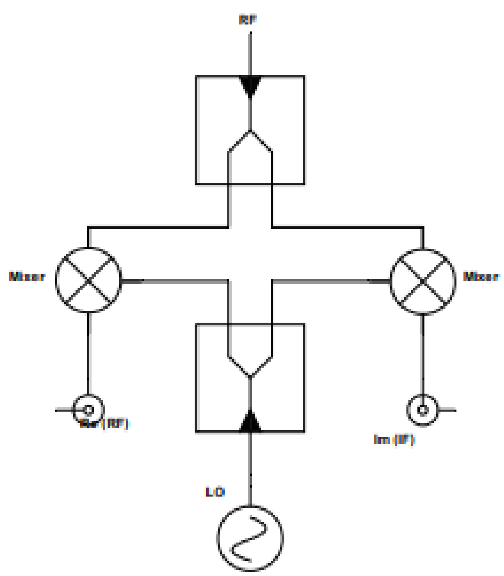

Figure 9: Double-sideband mixer

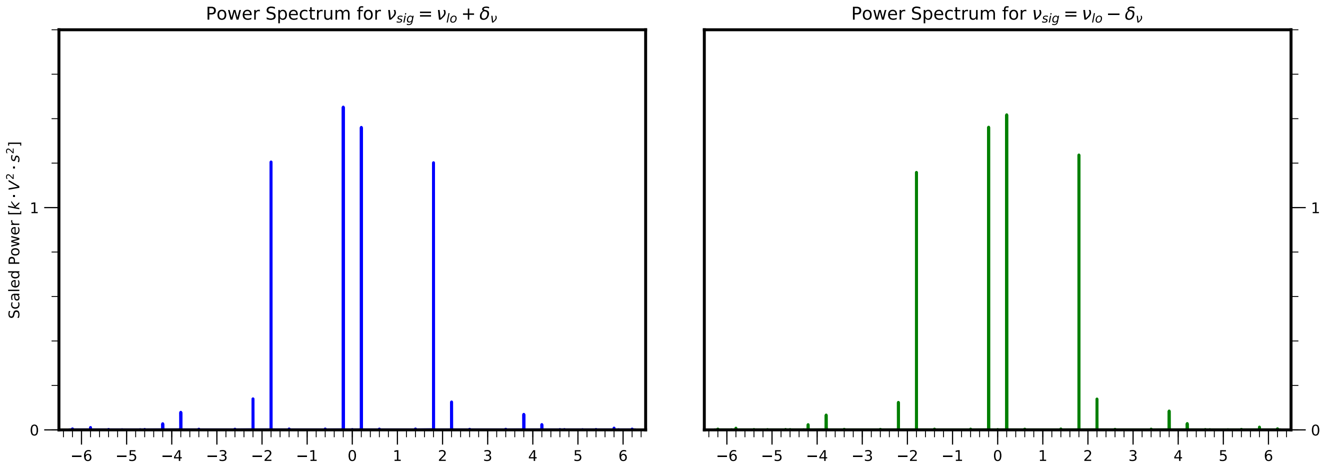

Figure 10: Double-sideband mixer

3.2.1 Sideband-Separating Mixer

After setting up our two ZAD mixers with a 90 degree phase shift using a longer cable for one input, we took our data just like we did above. By setting one input 90 degrees out of phase, we are attempting to generate a complex signal, allowing us to use our mixer as a true Sideband-Separating mixer. Our hypothesis was that we should see no upper sideband. Although in our plotting above we do still see some upper sideband and we attribute this to not achieving a proper 90 degree phase shift.

Conclusion

In this lab we dipped our toes in the fundamentals of radio astronomy ad delved into the different phenomena radio astronomers deal with in the current era. We showed how to properly sample and analyze signals, demonstrating proper use of the lab equipment, and knowledge of python programming. We demonstrated the ability to sample signals far out of specified range for our specified equipment. We also learned about the fundamentals of the Fourier transform and its implications in processing radio signals. It is one thing to painstakingly take a Fourier transform by hand but it is completely another to write a piece of code to do it, and fundamentally understand the function it is performing. It becomes much more intuitive and is something I wish I did before. our most troublesome aspect was the Sideband-Separating Mixer. Also we felt that Nyquist windows and ACF function were not properly explained. We feel that a review would be good to further cement the concepts.

Distribution of Effort

For Distribution, I focused on first off getting version control up and running with Github. This was difficult to manage with Jupyter notebooks because git doesn't recognize the file type and this can be troublesome when encountering merge-conflicts . After a couple of weeks I got this issue sorted (resolved using NBDIME). I handled the equipment setups and control, including all devices mentioned above. I also worked on plotting power spectrum's as well as making the functions for the graphs interactive. The mixer setups took a long time but professor Parsons was very helpful. Oscar worked on gathering data and inputting all functions for plots. Oscar primarily worked on the Nyquist windows, noise, and mixers. Matthew worked on gathering mixer data as well as the plots. He also wrote code to calculate power and voltage spectra's as well as the imaginary and real plots. Overall our workload was spread fairly evenly throughout the members and 80 percent of the time we were all here together working on the code. Overall I am very pleased with the performacnce of our team. Our appropriate code can be viewed under our corresponding branches: $http://github.com/arthurm101/radiolab/blob/Arthur/Lab$

References

[1] Exploring digital sampling, fourier transforms, and both dsb and ssb mixers. https://github.com/AaronParsons/ugradio/blob/master/lab$_mixers/allmixers.pdf$.