Astro 121: An Analysis of Digital Signal Processing Tools for Probing the Radio Universe

Neil Pichay
Partner: Elma Chuang Mia Birkelund
Professor: Aaron Parsons
University of California Berkeley

(Dated: July 13, 2024)

In field of radio astronomy, we present the foundations to digital processing, Fourier Transforms, and mixers. This lab applies a variety of skills and knowledge from debugging software, linear algebra, and using digital instruments. Various representation of data (power spectrum, convolutions, etc.) are presented while our methods for acquiring and analyzing our data are carefully detailed. Techniques like measuring frequency resolution, fitting noise patterns, and Fourier filtering add to the multi-dimensional lengths needed for characterizing the systems experimented in this lab. Our analysis provide us with guidance for understanding modern technology and serve as an important basis for probing the universe with radio.

I. INTRODUCTION

The goal of this lab is to understand digital sine wave Discrete Fourier Transforms (DFTs) and mixers. The methods used in acquiring data and the analysis of such data needs applies technical knowledge in both hardware and software. My lab report will detail the process of data acquisition and the various ways of analyzing that our data that test our understanding of the tools we use for radio astronomy.

II. INSTRUMENTS

The instruments used in this lab are the two wave generators (Keithley 3390, SRS DS345), a digital oscilloscope (Rigol DS1052E), a noise generator (NOD 5250), a Raspberry Pi, and a Nooelec SDR. While all of these instruments were crucial in our lab, I believe that the SDR is the most important instrument to understand because the SDR is what allows us to translate radio signals into data.

An SDR takes an input and processes it through various low pass and anti-aliasing filters until it eventually runs through an ADC and converts into a digital signal. While using the SDR was initially quite simple, we only started appreciating the intricacy of the SDR when understanding mixers.

III. BENEATH THE SINUSOIDAL WAVES

A. Digital Sine Waves and Fourier Transforms

Figure 1 shows a digital 300 kHz sinusoidal signal in time domain. To confirm the sinusoidal behavior, I over-plotted the signal with a fitted sine function of 300 kHz frequency and an arbitrary parameters (amplitude, phase, and DC offset). Aside from a few crests that weren't sampled, the sine function fit the digital signal well.

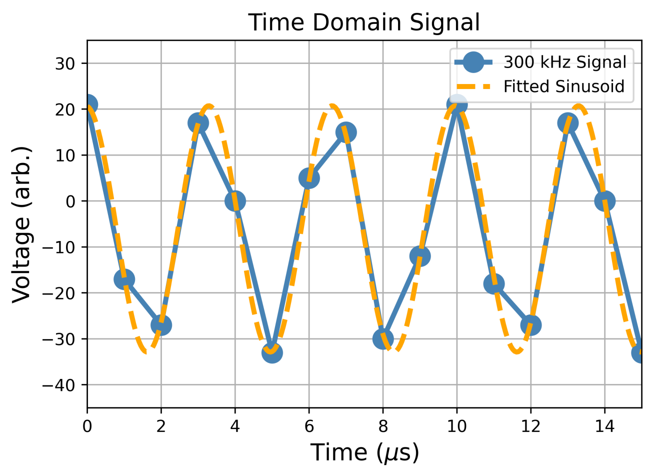

FIG. 1. Voltage spectra of 300 kHz sinusoidal signal fitted with a sine function. Given a computed amplitude, phase, and DC offset the fit is able to output a frequency of 300 kHz.

B. Voltage and Power Spectra (arb.)

A Fourier Transform (FT) is represented by the equation:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt \qquad (1)$$

where F(ω) is a frequency domain function, f(t) is a time domain function, $e^{i\omega t}$ is Euler's formula, and dt is an infinitesimally small time segment. A FT takes the inner product of a time domain signal with a complex sinusoidal to output a coefficients in the frequency domain (i.e. a frequency domain function).

In reality, we cannot compute FTs with dt because we are physically limited to finite time segments. To obtain the frequency domain function of a digital signal we compute a Discrete Fourier Transform (DFT).

Figure 2 shows the products of the 300 kHz signal DFT. These plots are known as voltage spectra in units of voltz/Hz vs. time. We can observe a clear resonance at 300 kHz originating from the input signal. There are real and imaginary components of the voltage spectra because the inner product inside of the FT is within the complex plane, meaning the output of the DFT (in frequency domain) will include real and imaginary frequencies.

Sine functions can be rewritten as:

$$sin(\omega t) = \frac{1}{2i}(e^{i\omega t} - e^{-i\omega t}) \tag{2}$$

Since our digital signal is well-fitted with a sine function, the positive and negative frequencies can be simply explained as phase relationship between the sinusoidal components of the wave.

You can see in 1 that the signal isn't totally symmetric around 0 Volts; this is the DC mean voltage that results in an apparent 0 Hz line in the real voltage spectra. The right panel in Figure 2 shows us correcting the DC mean voltage. We corrected all of the data presented in this paper.

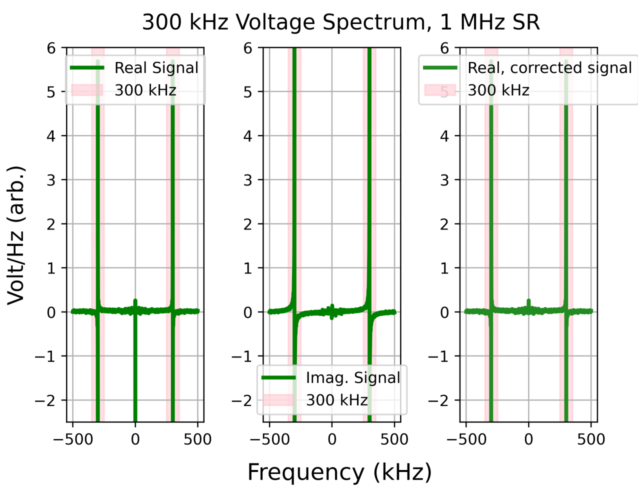

FIG. 2. Voltage spectra computed with discrete FTs of a 300 kHz signal. The left and middle panels show the real and imaginary components of the FT, while the right panel is the real FT corrected from its mean voltage.

Figure 3 is the power spectra of the signal, which is computed when we square the absolute value of voltage spectra. The logarithm of the power spectra allows to more clearly see what goes on in the DFT. A non-zero power at 0 Hz and ±300 kHz leaves a few implications: either the input isn't a pure 300 kHz signal or the power from 300 kHz is spilling into other frequencies.

We can confidently rule out the possibility of there being frequencies other than 300 kHz since our signal is best fitted with a 300 kHz sine wave. One related possibility is the existence of a higher-frequency signal that is aliased into the power spectrum, but we will address the topic of aliasing later into the paper. It could also be possible that the 300 kHz signal is producing harmonics that occur every N-th integer multiple of the signal frequency.

The primary reason for this non-zero power output is because of 300 kHz power spilling into other frequency bins. This occurs because each time segment sampled by the DFT has a finite amount of points it can sample from the signal frequency. Samples just outside of a time segment (i.e. bin) will spill over in the next time bin, resulting in a phenomena called spectral leakage. This is a consequence of sampling finite time bins that DFT's suffer from.

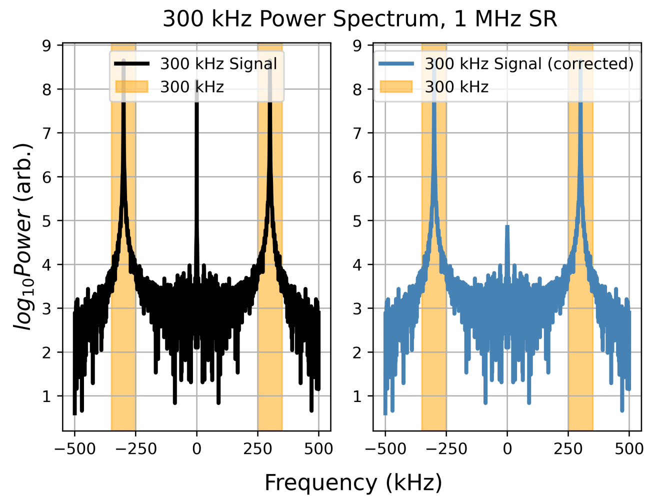

FIG. 3. Power spectrum of 300 kHz signal. The 0 Hz signal is partially removed by correcting the mean voltage. We observe spectral leakage in the non-zero power around away from the positive/negative signal resonance and 0 Hz.

C. Upper Limits on Frequency Resolution

Frequency resolution describes minimum frequency difference that two signals can be distinguishable from one another. To measure an upper limit on the frequency resolution, we combined the outputs of two signals: 400 kHz and 400 kHz + $\Delta\nu$. Figure 5 shows the values $\Delta\nu$ we inputted. While the middle panel shows a small bump at around 500 Hz, the minimum $\Delta\nu$ resulting in two signal distinguishable was measured around 900 Hz.

One very important variable in measuring this frequency resolution is the time resolution: the amount of time it took to sample our data. For a given sample rate (SR), we increased the time resolution by increasing the amount samples we took for 1 block of data. At a time resolution of 0.64 ms, the upper limit on our frequency resolution was about 3000 Hz. Increasing the time resolution resulted in better resolution. The time resolution

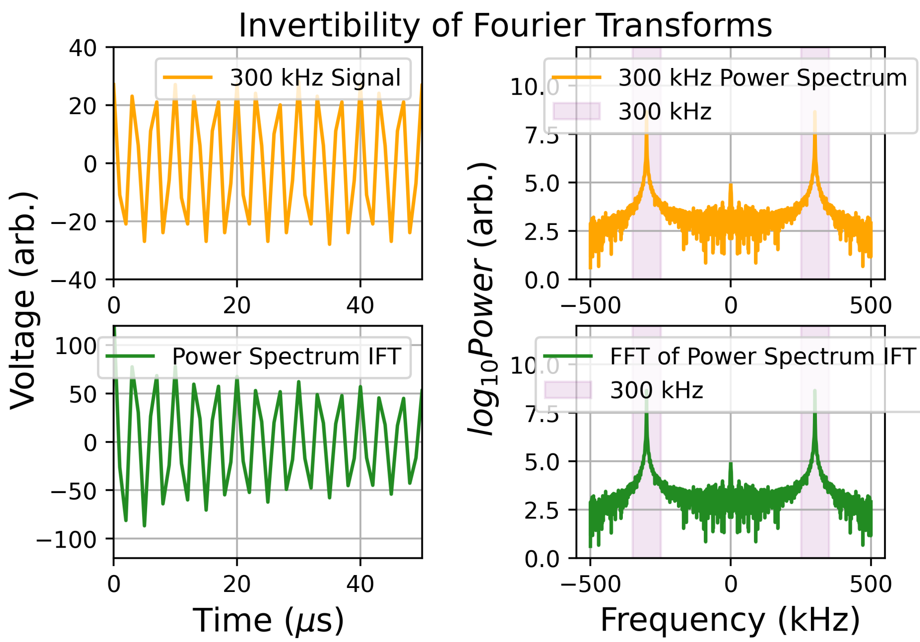

FIG. 4. Various representations of 300 kHz time domain signals and power spectra. The top row are plots of the original signal, while the plots on bottom row are produced by the inverse FTs of the original power spectrum.

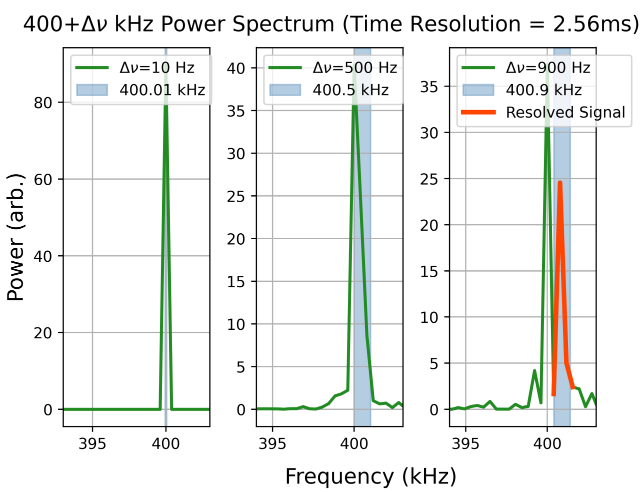

FIG. 5. Combined power spectra of 400 kHz and 400 + $\Delta\nu$ kHz signals. At a time resolution of 2.56 ms, an upper limit on the our frequency resolution is about 900 Hz.

is important for measuring frequency resolution because the DFT is fed more samples per bin; this increases the signal to noise ratio of the DFT and improves the resolution substantially. Hypothetically, we can keep placing better upper limits on the frequency resolution if we continue increasing our time resolution, but admittedly our eyes got tired of zooming into power spectra.

D. Representation Matters

Another operation we can act on our signal is an Inverse FT (IFT). As the name suggests, it does the inverse of a FT. An IFT is represented by the function:

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega \tag{3}$$

where the parameters are the same as the FT, but slightly inverted. Instead of inputting f(t) and outputting F(ω), we input a F(ω) and output a f(t). Another way of representing this is:

$$f(t) \to FT = F(\omega) \to IFT = f(t) \tag{4}$$

Figure 4 shows this exact result, where f(t) is a 300 kHz signal and F(ω) is the corresponding power spectrum that has been inverted to reproduce the original 300 kHz signal.

Alongside using IFTs to reproduce a power spectrum we have another tool called a convolution. A convolution of function g onto f is represented by the function:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{5}$$

where f(t) g(t) are time domain signals and dτ represent a small phase shift of g(t) relative to f(t). The output of a convolution is a third signal that is a combination of the input signals and represents the similarity of each

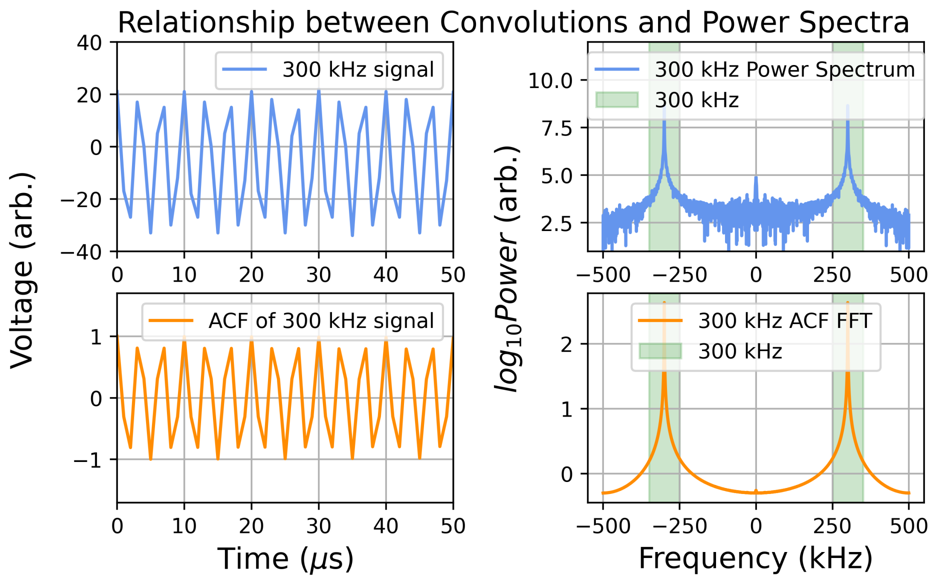

FIG. 6. An auto-correlation function of a 300 kHz signal it's FT compared to the original time domain signal and power spectrum. We observe that the FT of the auto-correlation function is similar to the power spectrum.

signal. In a way, a convolution is a filter in how it can get rid unwanted frequencies in both signals. The convolution theorem can also be used to explain spectra leakage because it is impossible to compare infinitesimally small segments of two functions together. The convolution of a time domain function onto itself is known as an auto-correlation function (ACF). Here is the result when the ACF is FT'd

$$FT[(f * g)(t)] = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)e^{i\omega\tau}d\tau = F(\omega)G(\omega) \tag{6}$$

The FT of an ACF converts time domain signals into frequency domain. While the convolution of two different signals has various applications in signal processing (e.g. Fourier Filtering), we can see something interesting when f(t) = g(t): this is the power spectrum of f(t)!

Figure 6 is used to visualize this exact result. First we take the ACF of a signal and then compare its FT to the power spectrum of the signal. I used a Python function *statsmodels.tsa.acf* - which takes in an array of my signal and a number of $d\tau$'s - and DFT'd the output of the ACF in the bottom right panel. We see that the original 300 kHz signal is preserved. It also appears that the DFT/FFT has less leakage and than the power spectrum. This proves that convolving in the time domain is the same as multiplying in the frequency domain and vice versa.

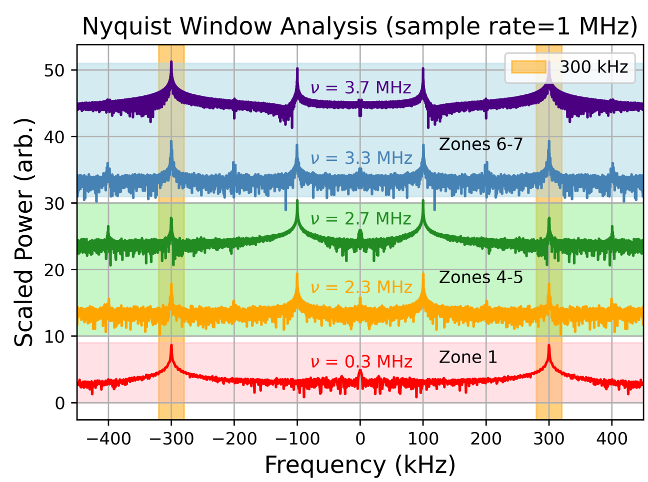

FIG. 7. Power spectra of various signal sampled at a rate of 1 Mhz. The labelled Nyquist Zones correspond to the higher frequency signals outside of the Nyquist criteria that become aliased.

E. Peeping into Nyquist Windows

As hinted in section B, our ability to observe (i.e. observe the corresponding resonance in the power spectrum) a signal strongly depends on the rate at which we are sampling our signal. For a given signal frequency, the Nyquist criteria tells us that we must sample at more than double of that frequency:

$$2f_{signal} < f_{sample} \text{ or } f_{signal} < \frac{f_{sample}}{2} \tag{7}$$

where f_{signal} is the frequency of the signal and f_{sample} is the SR. The idea behind the Nyquist criteria is that in order to capture the frequency of a signal you need to capture < 2 points per period. All of our presented work - that isn't meant to be aliased - is safely within the Nyquist criteria.

Signal frequencies that break the Nyquist criteria experience aliasing: an effect where signal frequencies aren't properly captured, leading to the misidentification of frequencies. For example, a 500 kHz signal needs a SR > 1 MHz in order to avoid being aliased. At exactly 1 MHz SR, the data will capture points separated precisely by one period, which would return a flat 0 Hz signal. At less than 1 MHz SR, the data will alias and fail to capture the 500 kHz resonance.

While it may seem that we should avoid aliasing at all cost, it's not totally bad thing; we can sometimes use aliasing to our advantage. Let's set our sampling rate to be f_{sample}; every $\frac{f_{sample}}{2}$ represents a new Nyquist window (zone). If $f_{sample} = 1$ MHz, for example, then a signal of 0.3 MHz in the first Nyquist window, 0.7 MHz is in the second window, 2.3 MHz is in the 4th window, etc.. Clearly any signal beyond the first Nyquist window will be aliased, but we can predict how that signal will be aliased.

Figure 7 shows us the power spectra of various signals all sampled at 1 MHz. An unaliased signal like 0.3 MHz is directly observed as 300 kHz, but if you notice so are all the other signals. 2.3 MHz "shows up" at 300 kHz because it is being folded into the first Nyquist window. Here is the path that 2.3 MHz takes:

$$\text{In MHz} : 2.3 \rightarrow 1.7 \rightarrow 1.3 \rightarrow 0.7 \rightarrow 0.3 \tag{8}$$

This is awesome because we can preserve information about signals even if they're aliased! However, there's a catch: I also specifically chose signals 2.7, 3.3, and 3.7 MHz to show how they also get folded into 300 kHz. Here is the general rule for aliased signals:

$$f_{\text{signal}} = \frac{N f_{\text{sample}}}{2} \pm \Delta f \tag{9}$$

$$\text{If } N \text{ is even: } f_{\text{signal}} \rightarrow \Delta f \tag{10}$$

$$\text{If } N \text{ is odd: } f_{\text{signal}} \rightarrow \frac{f_{\text{sample}}}{2} - \Delta f \tag{11}$$

In the field of signal processing, one goal is to be able to retain information about a signal while minimizing the amount of data points we take. This degeneracy means that we can only use aliasing to our advantage if we already know all the frequencies that exist in our signal.

F. Being Nosy with Noise

In radio astronomy, noise is in the instruments, noise is in the air, noise is everywhere. If we can't beat it, understanding it. To experiment with noise, we took several blocks of data (2048 samples per block) from a noise generator. Plotting it's time domain signal looked like an infant playing with a coloring book, so I thought it was more interesting to show noise in other representations.

Figure 8 are noise histograms with 12 bins at about 1 arb. V per bin. The left panel has two blocks of data while the right panel at 16 blocks. Since receivers themselves have Johnson Noise associated with thermal activity, both histograms are fitted with a normal distribution (i.e. Gaussian function). We can observe that as we increase the amount of samples, our histogram fits more accurately as a normal distribution. Understanding that noise acts as Gaussian has important applications in Gaussian smoothing (i.e. filtering). This is in agreement with the Central Limit Theorem, which states that a large enough sample size will be normally distributed.

A characteristic feature of noise is that it rings across all frequencies. Figure 9 is the power spectrum of noise (samples=16384) compared to the FFT of its ACF. Aside from the 0 Hz frequency caused by a DC voltage, the appearance of both plots and their FWHM are similar. This is in agreement with Figure 6.

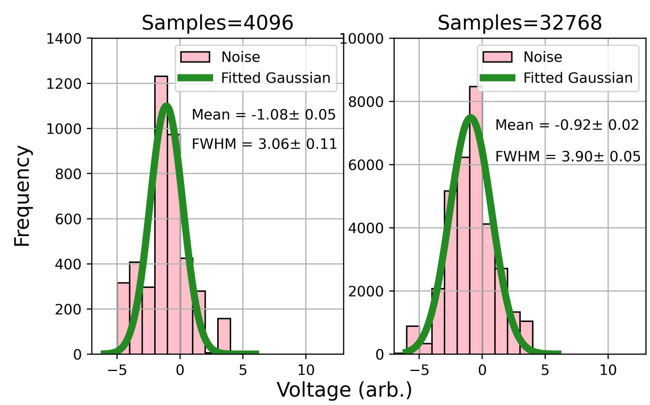

FIG. 8. Histograms of noise with different number of samples over plotted with fitted Gaussian function.

IV. MIXIN' IT UP

A. DSB Mixers

Mixing signals together means multiplying two signals to shift the input signal by certain frequency. This process is known as heterodyne mixing and mobilizes signals to and from different frequency regions.

To describe the inputs of mixers, we have a radio frequency (RF) and a local oscillator (LO). For this section,

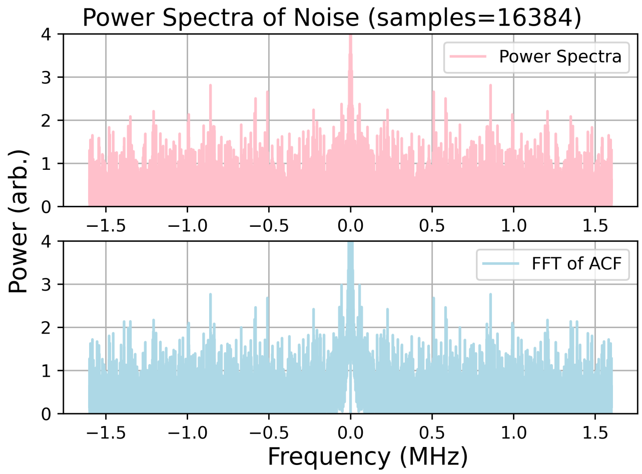

FIG. 9. Power spectrum of noise (samples = 16384) compared to the FT of the noise auto-correlation function. Aside from the 0 Hz signal, the noise is absent of any resonant frequencies.

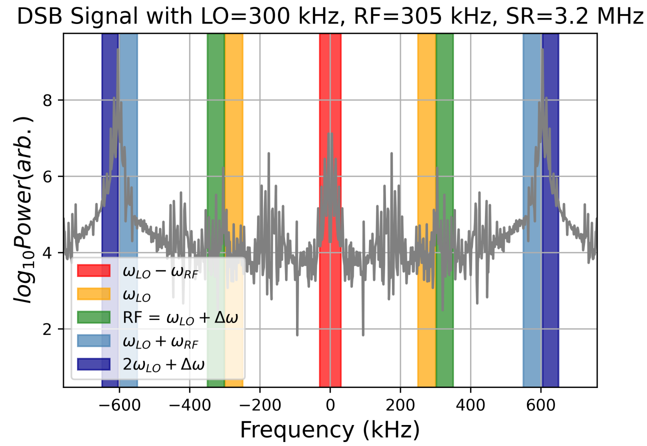

FIG. 10. Power spectrum of a double side band mixer with expected resonances at $\omega_{LO}, \omega_{RF}, \omega_{LO} - \omega_{RF}, 2\omega_{LO} + \Delta\omega$ and $\omega_{LO} + \omega_{RF}$.

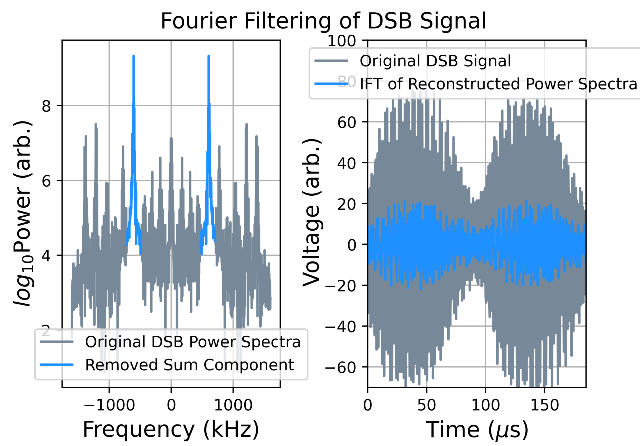

FIG. 11. Power spectrum of double side band with the real and imaginary sum frequency removed. The resulting time domain signal (in blue) resembles a beat frequency.

the LO corresponding to a sine function with frequency ω_{LO} and a RF of frequency $\omega_{RF} = \omega_{LO} + \Delta\omega$. The result of mixing the two signals together (after applying a low pass filter) is in the form:

$$\mathrm{RF}*\mathrm{LO} = sin((w_{LO}+\Delta\omega)t)sin(\omega_{LO}t) \approx cos(\Delta\omega) \quad (12)$$

Since $cos(\Delta\omega)$ can also be written as:

$$cos(\Delta\omega t) = \frac{1}{2}(e^{i\Delta\omega t} + e^{-i\Delta\omega t}) \quad (13)$$

the output of the mixers is $\pm$ $\Delta\omega$. Since a signal shows up a positive and negative frequency, this type of mixer is known as a double side band (DSB) mixer.

Figure 10 shows the DSB output the LO (300 kHz) and RF (305 kHz) mixed and sampled at 3.2 MHz. From the equation above, we can observe $\pm\Delta\omega$ in red, alongside $\omega_{LO}, \omega_{RF}, \omega_{LO}+\omega_{RF}$, and $2\omega_{LO}+\Delta\omega$ (derived from the expanded terms of $f(t)sin(\omega_0 t)$). Since positive and negative frequencies are indistinguishable from one another, the receiver redundantly uses more bandwidth and power to communicate the same signals. Later in the paper we will address how signals can be more efficiently communicated.

Figure 11 shows the power spectrum and time domain signal of a DSB mixer output as in Figure 10. We can see the characteristic beat frequency shape in grey on the right panel. When we zero out the real and imaginary components of the sum frequency $(\omega_{LO} + \omega_{RF})$ in the voltage spectrum and IFT, we get the new time domain signal in blue. It looks like an attenuated beat frequency. This process is known as Fourier Filtering, which I used to filter the signal in "secret_message.npz". Before filtering the secret signal, it sounded like "Aaron Parsons is an awesome professor." But after filtering high frequency (6000 Hz $< |f|$) components I heard "Fourier Transforms at the best." Fourier Transforms are indeed the best!

B. SSB Mixers

If the positive and negative frequencies of a DSB mixer relay the same information, it would be more efficient to only transmit a *single* frequency. This type of mixer is known as a single side band (SSB) mixer.

Here is the set up: we have a complex sinusoid with frequency w_{LO} as a LO, a RF with frequency $\omega_{RF} = \omega_{LO} + \Delta\omega$ and a low pass filter:

$$\mathrm{RF} * \mathrm{LO} = sin((\omega_{LO} + \Delta\omega)t)e^{-i\omega_{LO}t} \approx e^{i\Delta\omega t} \quad (14)$$

where $e^{i\omega_{LO}t}$ can be rewritten as two sinusoidal functions with a relative phase shift in the complex plane

$$e^{i\omega_{LO}t} = cos(\omega_{LO}t) + isin(\omega_{LO}t) \quad (15)$$

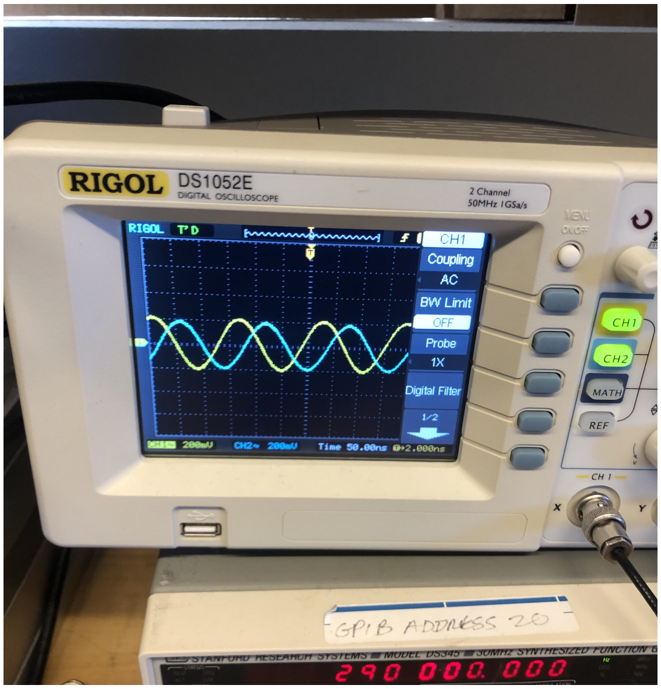

FIG. 12. An image of our attempt at producing two signals with relative phase shift. The oscilloscope gives us a result that is similar to plotting sin(x) over sin(x + 90°).

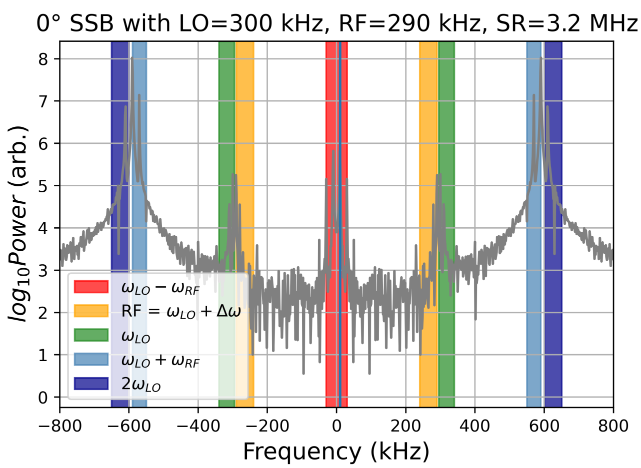

FIG. 13. Power spectrum of single side band mixer with identical local oscillator inputs. We observe that the resonance frequencies similarly correspond to the double side band mixer power spectrum in Figure 10.

There's a lot of algebra in between, but the idea is that when you mix a signal into the complex plane you can output a complex sinusoidal with frequency $\Delta\omega$. Without this relative phase shift, the SSB mixer looks very similar to the DSB mixer in Figure 13.

We had tried to produce two signals with a relative phase of 90° by splitting a signal where one of the outputs was fed to a really long wire. This attempt is displayed in Figure 12. We then realized that we couldn't

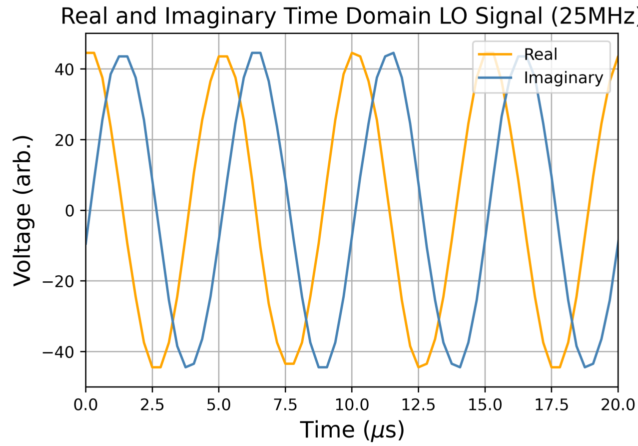

FIG. 14. Real and imaginary components of the internal LO at 25 MHz that are mixed with the external RF to produce a SSB.

properly combine the two outputs into the lab's quadrature splitter because of signal attenuation, so we tried another method.

To observe the output of a proper SSB, we relied on using the SDR's internal mixer. This works by generating a real LO and an imaginary phase shifted counterpart LO with a frequency ω_{LO} and mixing it with an RF of frequency $\omega_{LO} + \Delta\omega$. The LO now works as a functional complex sinusoid. This LO (25 MHz) is represented in Figure 14 and is mixed with an external RF (25.2 MHz).

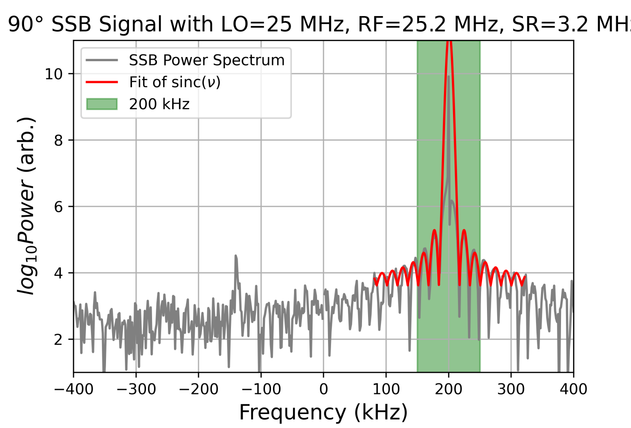

FIG. 15. Power spectrum of single side band mixer with 90° relative phase shift between the local oscillator inputs. The observed single resonance corresponding to $\Delta\nu$ is well-fitted with a sinc function due to a narrower bandwidth requirement and higher power efficiency than double side band mixers.

The product of this SSB mixer is happily displayed in Figure 15. The single resonance in the power spectrum corresponds to $\Delta\omega = 25.2$ MHz $-$ 25 MHz $=$ 200kHz, which is what we expected.

An "ideal" mixer ideally multiples two functions together to perfectly shift an input signal to another frequency. In reality we are constrained to internal diodes unable to sample at infinitesimally small time segments, causing power to leak into nearby frequency bins and harmonics. However, I thought it was impressive how distinguishable a sinc function can be seen in the SSB 200 kHz resonance, which why I fitted it with such sinc function.

Overall, SSB mixers are more ideal for communicating signals than DSB mixers because we aren't needing to transmit both positive and negative frequencies. The ability to transmit either one positive or negative frequency means a narrower bandwidth requirement and more effience power use.

V. CONCLUSIONS

My partners and I learned a lot about digitial signal processing after completing Lab 1. We initially got quite frustrated with both the hardware or software not working how we wanted them to, but it all paid off. In all, we learned about digital sampling, aliasing, FTs/IFT, convolution theorem, positive and negative frequencies, the complex plane, mixers, and Arctic Monkeys. I'm excited to apply our understanding in the next labs :D