EECE 301: Signals and Systems 11/11/17
Final Project: Guitar Tuner

Isaac Lawson-Hughes & Samuel Stone

Some really nice discussions and expressions but tempered by a few slip ups and /or misconceptions. Luckily the former out weigh the latter

## Introduction

The goal of this project is to use concepts of signals and systems to create the processing for a guitar tuner. We will be measuring the accuracy of our tuner using Eq. 1 which takes in a target frequency (f1) and estimate frequency (f2) to produce a measure of tuning called, "cents". To determine the direction in which f2 should change to approach f1, inequalities (1) and (2) are also shown below.

$$\frac{f_2}{f_1} = 2^{\frac{C}{1200}} \quad (1)$$

Eq. 1 Cents Tuning Estimate

$$f_2 > f_1, C > 0 \ (1)$$

$$f_2 < f_1, C < 0 \ (2)$$

To do this project we will be doing all of our processing in MATLAB which, most applicably, uses a Cooley-Tukey fft algorithm which boasts an O(N*log(N)) efficiency compared to the O(N^2) efficiency of the plain old DFT algorithm. Due to the nature of binary computing, it is recommended to choose the number of samples and zero-padding so that you process $2^x$ number of samples, so this is what we have done in the proceeding test cases.

$$O(N * \log(N)) \text{ vs. } O(N^2)$$

DFT vs. FFT Efficiency

## Part I. DFT of a Single Sinusoid and the Use of the Hamming Window

There are a few sources of error when considering the starting point of a continuous time signal going through to a DFT result. We will explore aliasing, smearing, and grid error as it relates to the inputs of DFT_1_Sine.m.

- Aliasing: The following test cases have sufficient samples and zero-padding to isolate the effects of aliasing.

Aliasing error happens when the frequency of a signal is more than half the sampling frequency of an ADC. This causes spectral replicas to overlap in the frequency domain after fft processing has been applied Below the Nyquist rate we expect to see a DFT sampling of the dirac-delta function at the frequency of the sine wave. This oversampled signal allows for space between the spectral replicants of the dirac-delta function while fully preserving the input signal.

- Oversampled signal (f1 < 2205 Hz)
  - We will see the sampled dirac function within the scope of what is sampled

>> DFT_1_Sine(1102.5, 2^15, 2^17)

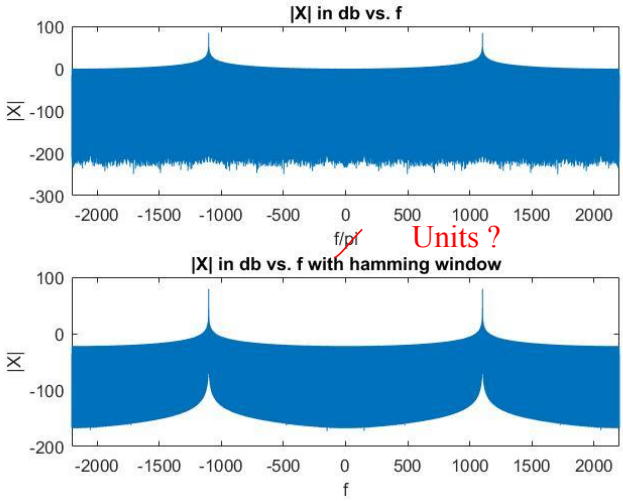


Figure 1. Oversampled Sinusoid

At the Nyquist rate the dirac-delta function, which theoretically should be double in height by its coincidence with its first spectral replicant, disappears. I am not sure why this happens yet.

- Critically sampled (f1 = 2205 Hz)
  - We will see the sampled dirac function on the edge of the plot

>> DFT_1_Sine(2205, 2^15, 2^17)

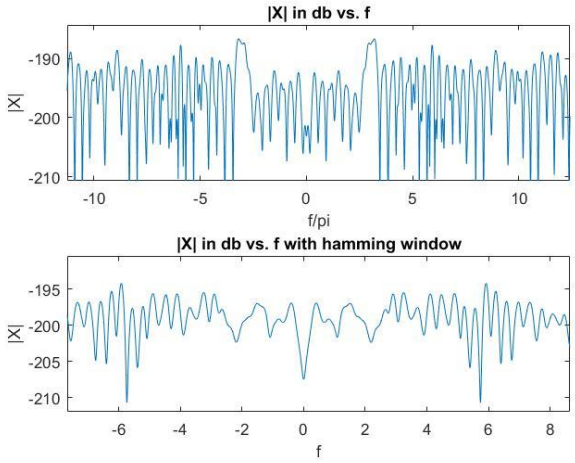


Figure 2. Critically Sampled Sinusoid

Above the Nyquist rate we will see aliasing error as the DFT's of higher frequency sinusoids creep into the Fs/2 window. Since we have a constant sampling rate of 4410 Hz

- Undersampled signal (f1 > 2205 Hz)
  - We will not see the peak

>> DFT_1_Sine(3307.5, 2^15, 2^17)

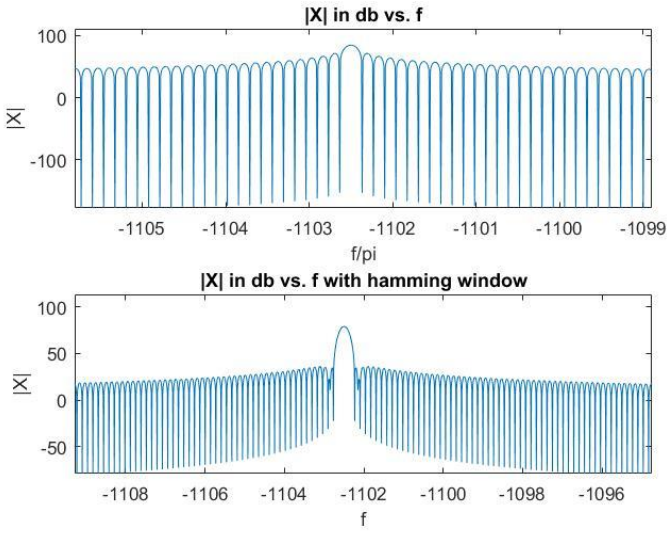


Figure 3. Undersampled Sinusoid

This happens because we have a fixed sampling rate of 4410 samples/second. The number of samples determines how long we sample at the given rate.

- Smearing: Ideally, a CT signal can be perfectly reassembled after digital conversion given that all meaningful information has been captured by sampling. This ideal case is known as DTFT_inf. This is a possibility in the case of a finite signal since the DFT can only compute finite-length signals. In most cases, including this one, the signal sampled is of infinite duration. When the DFT is applied to the finite samples taken of an infinite length signal, you get what is known as smearing error which appears to smooth the edges of the transformed signal which subsequently offers an imperfect representation of what's going on inside the signal. Since the rate of sampling is fixed in this application, we can take samples for a longer time (take more samples) to noticeably reduce the smearing.
-

>> DFT_1_Sine(1102.5, 2^16, 2^22)

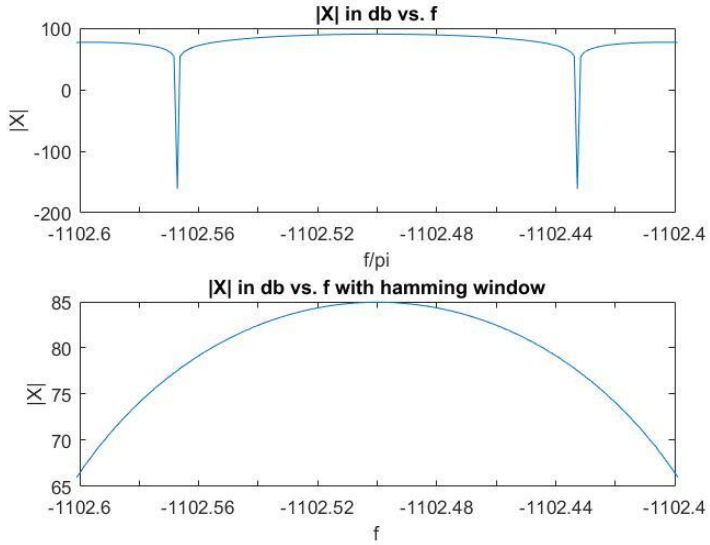


Figure 4. 2^16 Samples

>> DFT_1_Sine(1102.5, 2^18, 2^22)

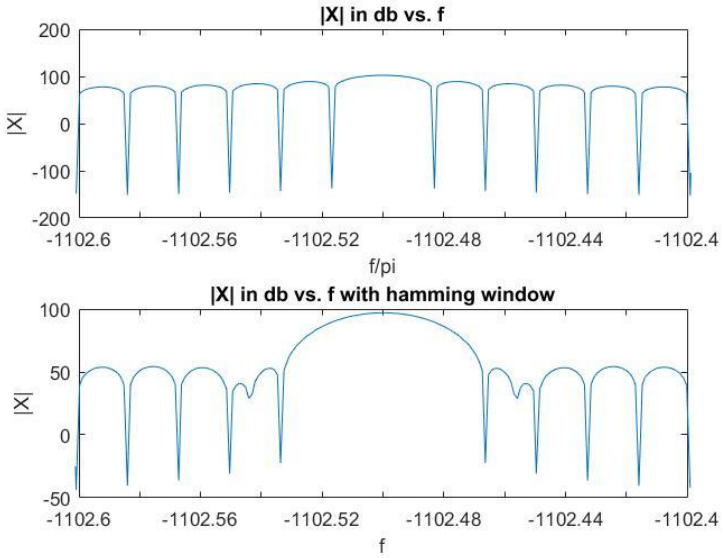


Figure 5. 2^18 Samples

- By element-wise application of a tapering window, one is able to control the effects of smearing by attenuating the samples before they go through DFT processing. Here is the window tool viewer for a Hamming of length 2^15.

```
>> wvtool(hamming(2^15))
```

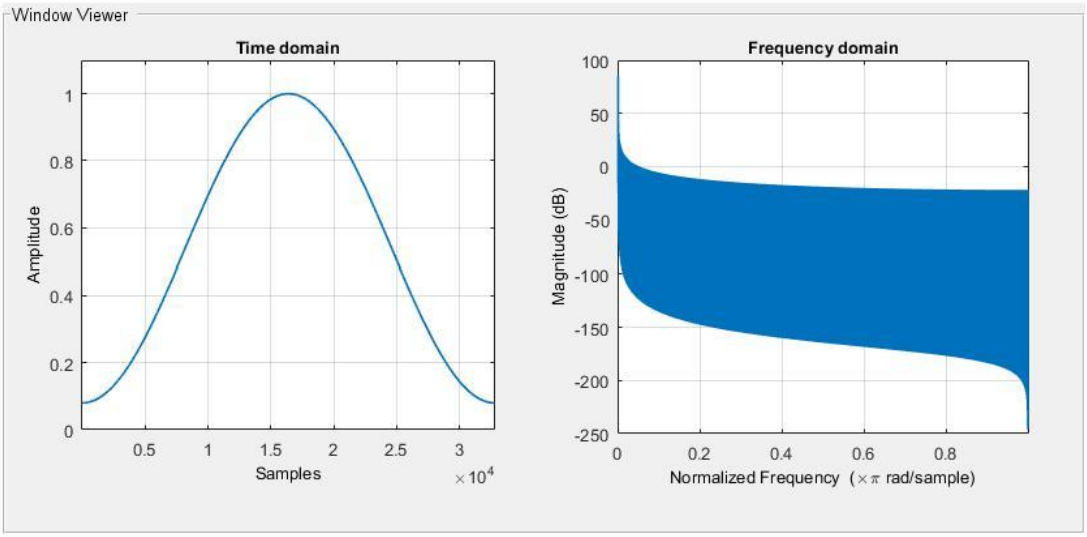


Figure 6. Hamming Window of length 2^15

  - Number of samples taken
  - Window choice
    - Hamming window is taken at the same number of points as is the signal

- Grid Error: When computing the DFT for a given number of points, one will obtain values that are exactly on the smeared DTFT curve. Although these points are exact, it is possible to obtain a better picture of the given smeared DTFT curve by computing a finer grid. This is done through a step

called zero padding which adds the number of samples that are zero to the array of signal samples. This does not add any additional information to the DTFT curve, but it allows the DFT to compute a finer grid which can resolve features to a higher degree.

In terms of mainlobe and sidelobe smearing, it is obvious between the non-windowed and windowed signal in both Figures 4 and 5 that the Hamming window reveals features swallowed by smearing error.

## Part II. DFT of Two Sinusoids

There are two main issues to discuss in this section: Mainlobe and Sidelobe attenuation. Without proper sampling length, the information to reveal any number of merged mainlobes or sidelobes would not be possible. Both smearing and grid error can affect whether the two resonant peaks can be resolved. In testing to see the relationship between resonant peak frequency difference and number samples taken, we will keep the distance between peaks constant, 0.05 for the mainlobe case and 0.2 for the sidelobe case. In order to make negligible any grid error in the test cases experimenting with different sizes of N, Nzp will remain constant, a few orders of magnitude above the highest value of N. This will ensure that the same number of points will be calculated on the smeared DTFT of whatever N gives us.

- Number of Samples
    - Mainlobe
    - Sidelobe
- The Hamming Window

>> DFT_2_Sine(1102.15, 1102.2, 1, 2^18, 2^20)

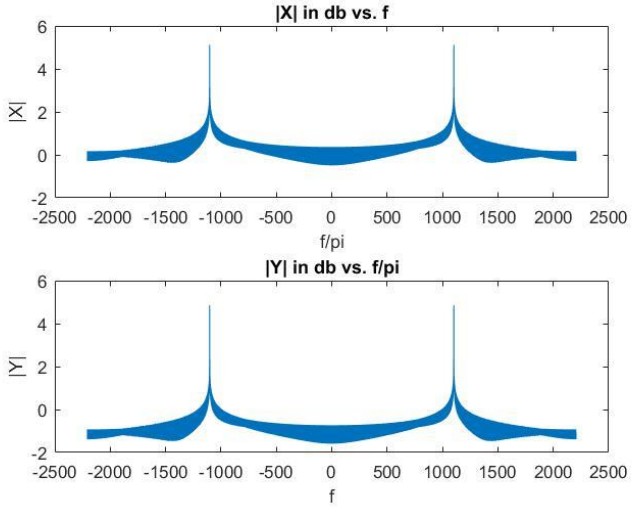


Figure 7. Full Axis Mainlobe

>> DFT_2_Sine(1102.15,1102.2,1,2^16,2^20)

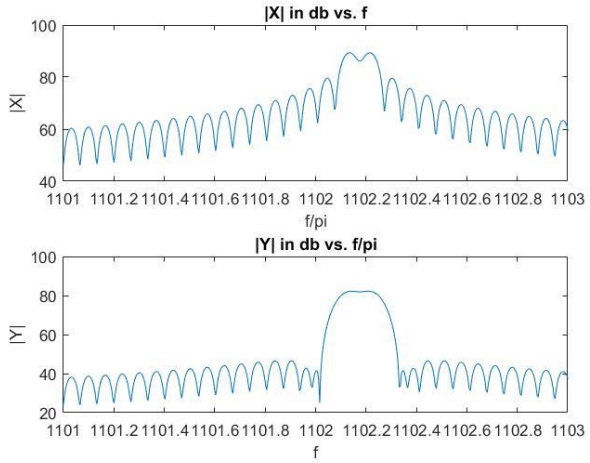


Figure 8. Mainlobe Smearing at 2^16 Samples

>> DFT_2_Sine(1102.15,1102.2,1,2^17,2^20)

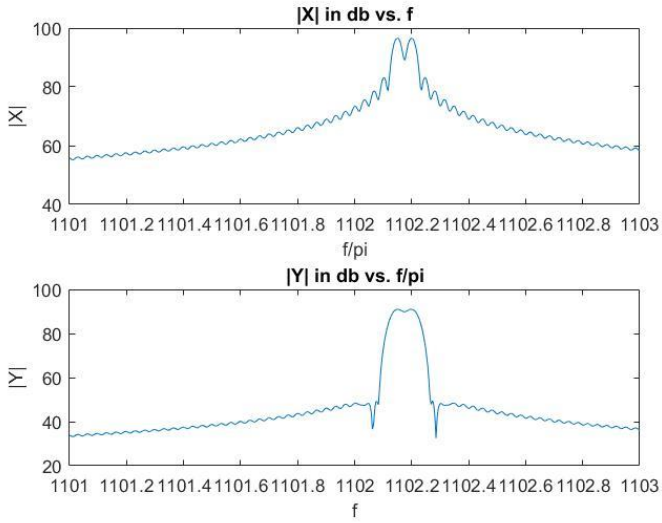


Figure 9. Mainlobe Smearing at 2^17 Samples

>> DFT_2_Sine(1102.15,1102.2,1,2^18,2^20)

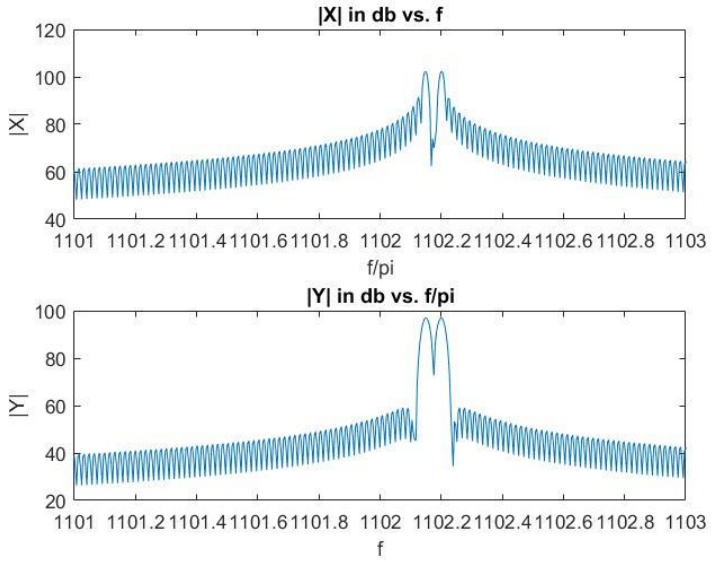


Figure 10. Mainlobe Smearing at 2^18 Samples

>> DFT_2_Sine(1102,1102.2,.01,2^16,2^20)

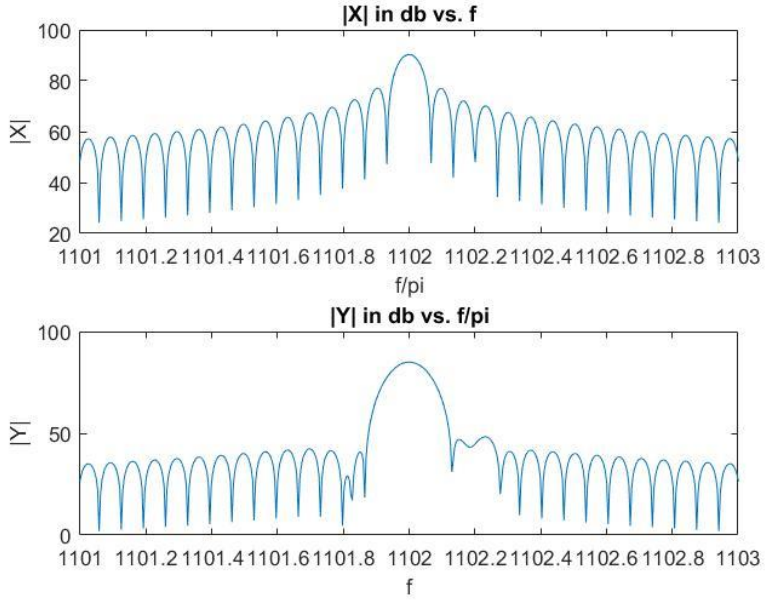


Figure 11. Sidelobe Smearing at 2^16 Samples

>> DFT_2_Sine(1102,1102.2,.01,2^17,2^20)

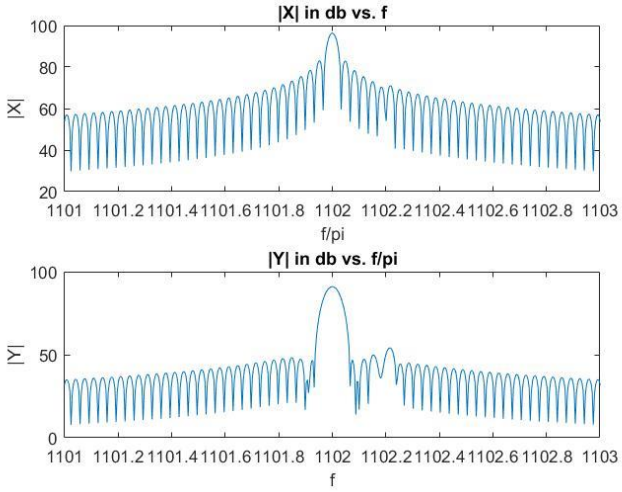


Figure 12. Sidelobe Smearing at 2^17 Samples

>> DFT_2_Sine(1102,1102.2,.01,2^18,2^20)

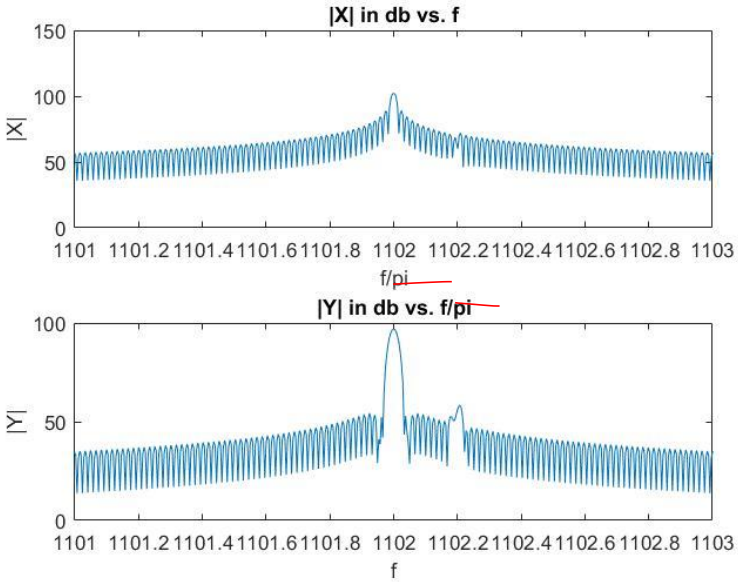


Figure 13. Sidelobe Smearing at 2^18 Samples

This is an example of two very closely spaced sinusoids on the full axis.

The Hamming window helps to attenuate noise on either side of the mainlobes. Though this is very useful when the two mainlobes are resolved (at longer sampling lengths), the non-windowed subplot resolves the two peaks at a shorter sampling length than the windowed subplot.

The Hamming window clearly makes sidelobe attenuation less prevalent at all sampling lengths tested.

## Part III. DFT of Individual Real Guitar Notes

The frequency domain signal appears to be periodic at first look. Further examination of the DFT data (Table 3) and equations (2) – (4) shows us a lot. Musical physics tells us that a vibrating string emits sound waves of frequencies that are integer multiples of its fundamental frequency. This is evident in the mathematical sequence going from the Fourier Series to the Generalized Fourier Transform of a periodic signal. A periodic signal, represented by the FS, when transformed, is a delta-dirac function at integer multiples of the fundamental angular frequency which is linearly related to the frequency in Hz. Finally, the collected data show very similar values when real processing has been applied.

$$x(t) = \sum_{k=-\infty}^{\infty} c_k * e^{jk\omega_0 t} \quad (2)$$

$$\mathcal{F}^{-1}\{x(t)\} = \sum_{k=-\infty}^{\infty} c_k * \mathcal{F}^{-1}\left\{e^{jk\omega_0 t}\right\} \quad (3)$$

$$X(\omega) = \sum_{k=-\infty}^{\infty} c_k * 2\pi\big(\delta(\omega - \omega_0 k)\big) \quad (4)$$

Going from the Fourier Series to the Generalized Fourier Transform.

The Hamming window helped to confirm the presence of some higher, attenuated resonances. In the Figures below each of the predicted peaks that are practically invisible without the Hamming window are marked in red. The Hamming window does this by reducing the effects of smearing which lowers the noise floor of the result. Also recorded below in Table 2 are expected values based on a measurement of the fundamental frequency of each processed signal. This reveals that each string is about two Hz lower than the target frequency for that string. Rearranging equation (1), we can calculate the tuning of each string.

$$1200 * log_2\left(\frac{f_2}{f_1}\right) = C \quad (1)$$

| Note | Cents |
|---|---|
| E | -47.1 |
| A | -31.8 |
| D | -27.7 |

A semitone is 100 cents which tells us that each string is less than half a semitone flat. (we know this because the cents are negative!)

| Note | Theoretical Resonant Frequencies (Hz) | | | | | |
|---|---|---|---|---|---|---|
| E (Low) | 82.41 | 164.82 | 247.23 | 329.64 | 412.05 | 494.46 |
| E (cont.) | 576.87 | 659.28 | 741.69 | 824.1 | 906.51 | 988.92 |
| A | 110 | 220 | 330 | 440 | 550 | 660 |
| A (cont.) | 770 | 880 | 990 | | | |
| D | 146.83 | 293.66 | 440.49 | 587.32 | 734.15 | 880.98 |
| D (cont.) | | | | | | |

Table 1. Expected Values Based on Theoretical $f_0$

| Note | Expected Resonant Frequencies (Hz) | | | | | |
|---|---|---|---|---|---|---|
| E (Low) | 80.2 | 160.4 | 240.6 | 320.8 | 401 | 481.2 |
| E (cont.) | 561.4 | 641.6 | 721.8 | 802 | 882.2 | 962.4 |
| A | 108 | 216 | 324 | 432 | 540 | 648 |
| A (cont.) | 756 | 864 | 972 | | | |
| D | 144.5 | 289 | 433.5 | 578 | 722.5 | 867 |
| D (cont.) | | | | | | |

Table 2. Expected Values Based on Measured $f_0$

| Note | Measured Resonant Frequencies (Hz) | | | | | |
|---|---|---|---|---|---|---|
| E (Low) | 80.2 | 160.6 | 241 | 321 | 402.6 | 483.5 |
| E (cont.) | 565 | 643 | 727.6 | 809.6 | 898.5 | 998.5 |
| A | 108 | 216.5 | 325 | 433.2 | 542 | 650.6 |
| A (cont.) | 759.5 | 867 | 978 | | | |
| D | 144.5 | 289.25 | 434 | 578.5 | 724 | 869 |
| D (cont.) | | | | | | |

Table 3. Actual Values

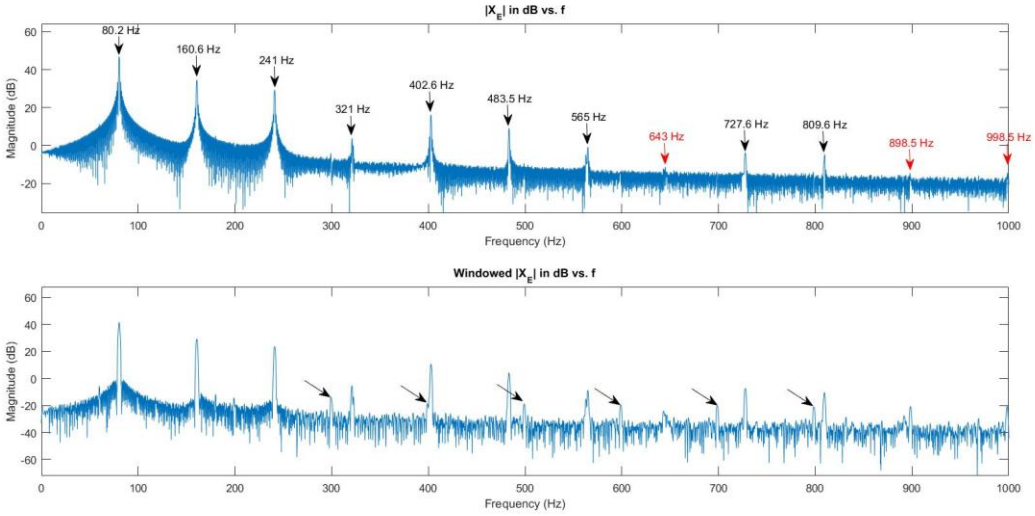


Figure 14. E String

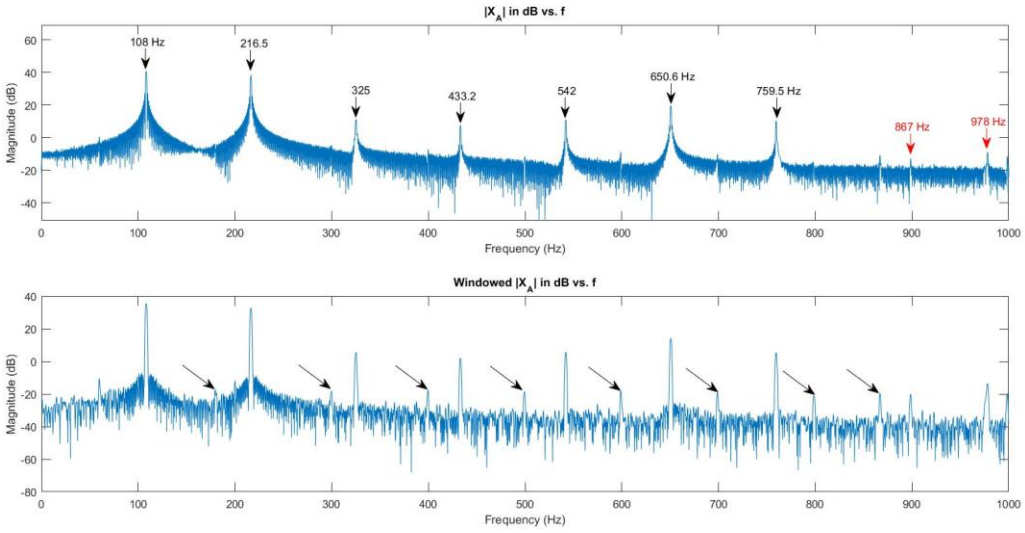


Figure 15. A String

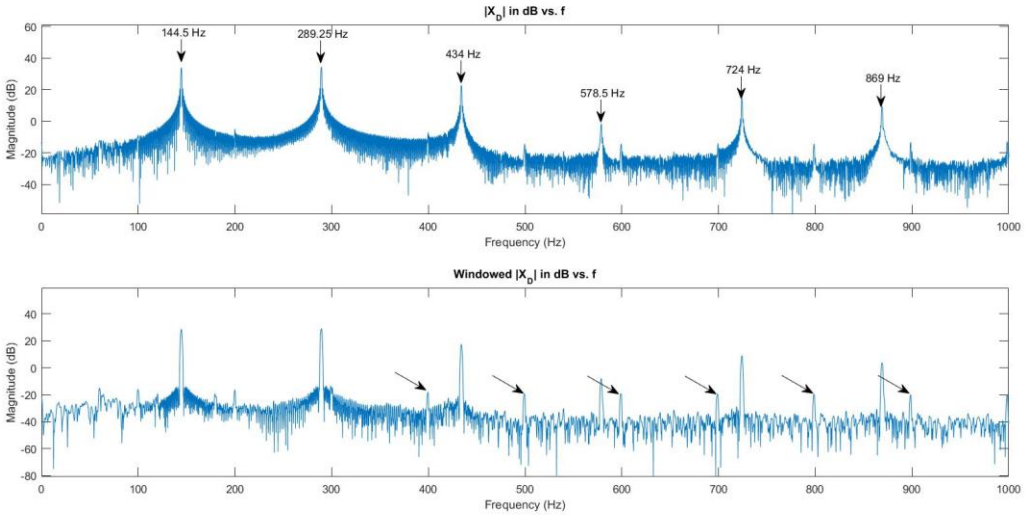


Figure 16. D String

Marked on each figure above in the Hamming window subplot by diagonal arrows is a periodic signal which occurs at approximately the same frequencies throughout each processing. The theory indicates that this is a continuous, single tone that is much weaker than the guitar being played. It is possible that the drone of a fan, HVAC system, etc. is present in the room where these recording were made.

## Part IV. DFT of Individual Synthetic Guitar Notes

This section of the project introduces the idea of using a synthetically generated note to test our system. With the Amp-Phase coefficients given, all we have to do is plug in a frequency we would like to test and generate it.

$$x(t) = A_0 + \sum_{k=1}^{\infty} A_k * \cos(k\omega_0 t + \phi_k) \quad (5)$$

Eq. 5. Amp-Phase FS

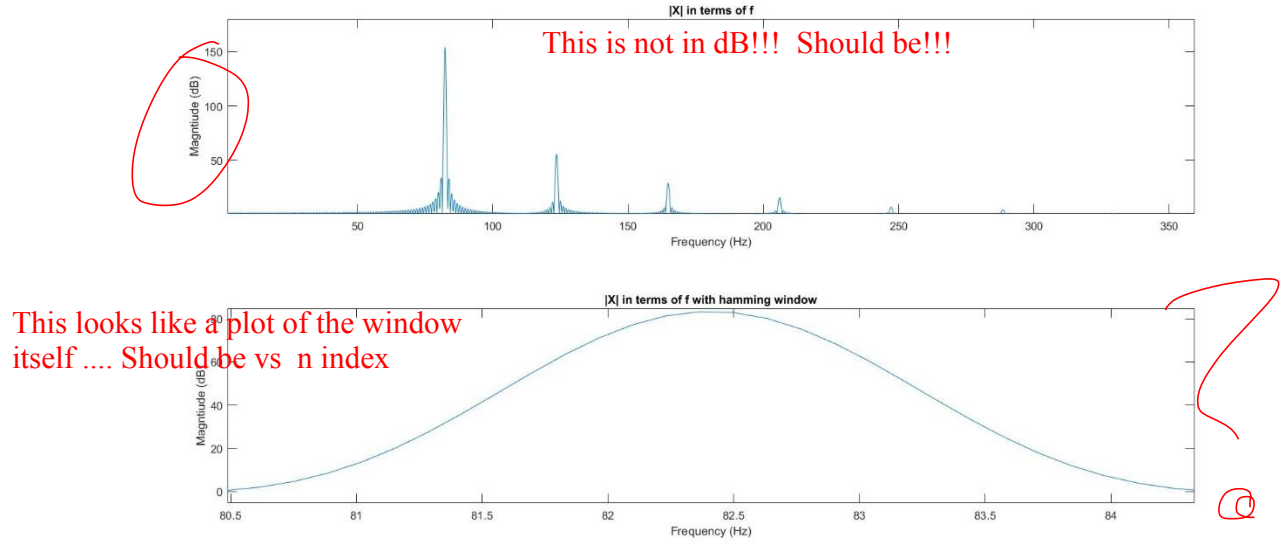


Figure 17. Synthetic E (Low)

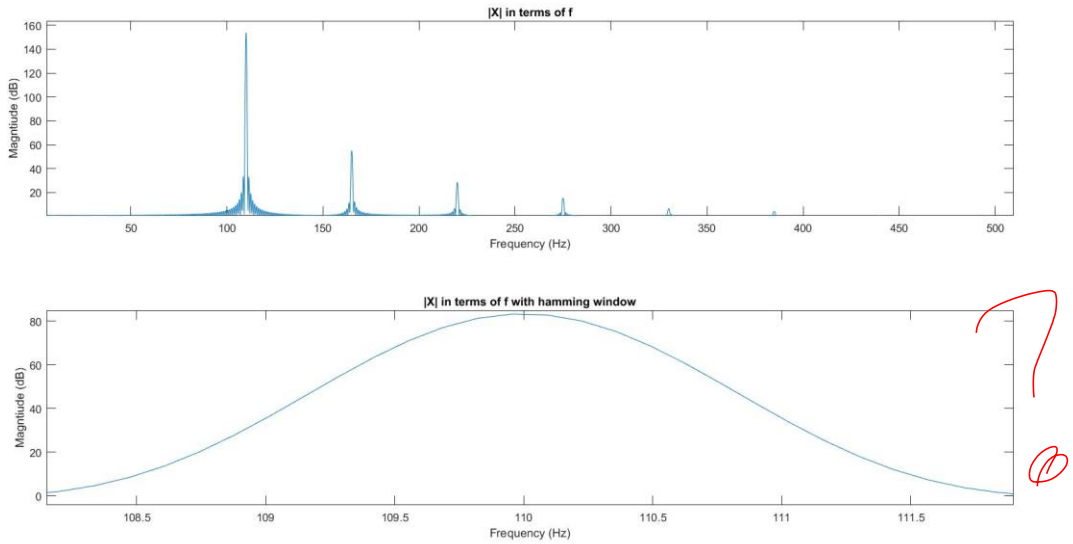


Figure 18. Synthetic A

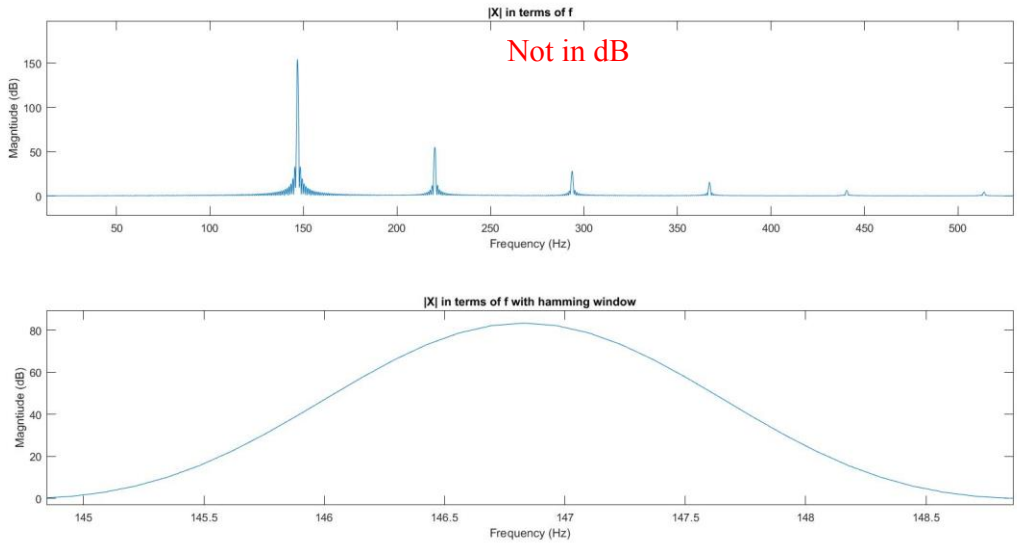


Figure 19. Synthetic D

The largest frequency that can be properly processed with Fs = 4410 is Fs/2 for a single sinusoid. In this part we are generating a musical note using a Fourier Series. The Amp-Phase model (5) uses the ten pre-determined significant $A_k$ values to determine the amplitude of each resonant peak. It is apparent that each of the resonant frequencies must stay intact to form a proper synthetic note. We can determine that the highest frequency of any simulated note will be $10 * f_0$ therefore restricting the limit of this model to 220.5 Hz this eliminates any chance of aliasing error.

# Part V. DFT Based Tuner

## Part V-A. Detection Filter Design

The design of Tuner_FIRs.m heavily relies on the equations presented in NS-28.

$$\text{Passband Ripple} = 20 * log_{10}(1 + \delta_p) \quad (6)$$

$$\text{Stopband Attenuation} = -20 * log_{10}(\delta_s) \quad (7)$$

By solving these equations for $\delta_p$ and $\delta_s$, we were able to properly plug in values to the firpmord MATLAB command. Finite Impulse Response filters are known for their inherent stability due to the lack of feedback terms, but this produces a tradeoff between accuracy and computational power consumption. In this case we must use less than 1000 coefficients which is obtained by widening the transition band specification. This causes anomalies to occur at the edges of the filter. For example, between the Hz values of 68-70 fall in the transition band dead zone as shown in the figure below. This produces false detection that the fundamental frequency is in the 110 Hz band when it is clearly in the 82.41 Hz band. This happens because the fundamental frequency peak is attenuated lower than the next highest resonant peak which increases the average power of the next highest band. This matches what is seen in the results and makes sense since the false detection occurs when the fundamental peak is attenuated below -10 dB. Similar effects happen at the edges of each filter. Below is a table of the false detection zones due to transition band attenuation. To correct this, one would require a more accurate filter, thus more computational power in the case of an FIR

filter or more complex design in the case of IIR filters. The transition band width was tuned to bring the number of coefficients in the filter just below 1000.

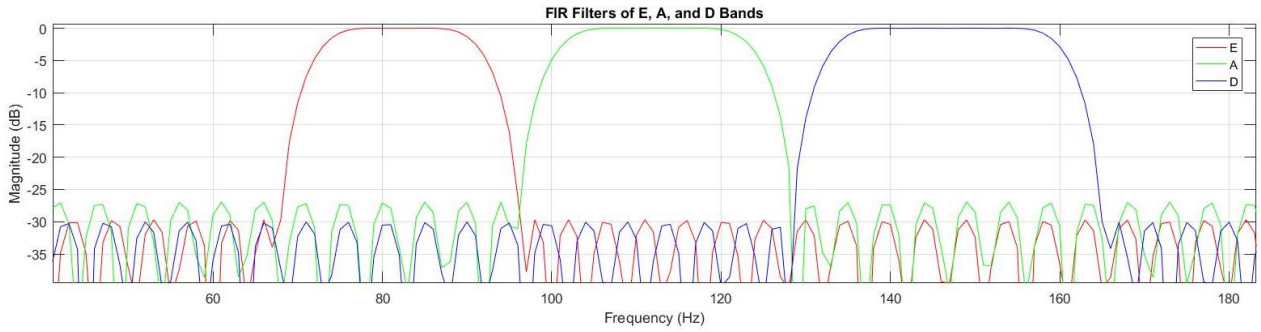


Figure 20. FIR Filters

| Transition | Frequency In (Hz) | Correct Band | Predicted Band |
|---|---|---|---|
| Low-E | 68 - 70 | E | A |
| E-A | 94 - 98 | E - A | D |
| A-D | 126 - 131 | A - D | Far Too High |
| D-High | 162 -165 | D | Far Too High |

Table . FIR Filter Dead Zones

## Part V-B. Detection Filter Processing

In this section the FIR filters are applied to the input signal which produces an output signal which has a higher average power for the band which contains the fundamental frequency of the input. Average power is calculated using Parseval's Theorem of the discrete-time variety shown in the equation below.

$$E = \sum_{-\infty}^{\infty} x^2[n] \quad (8)$$

Equation . Parseval's Theorem (Energy)

By putting the output from each filter into this equation (with proper bounds) allows us to simply pick the band in which the fundamental lies. The figures below illustrate this by comparing the the input signal with each filter output and then showing the relative power of each band in that case. Since the bar graphs of power are in dB the least negative dB bar shows the detected band. The previously mentioned "false detection zones" will be discussed in part E.

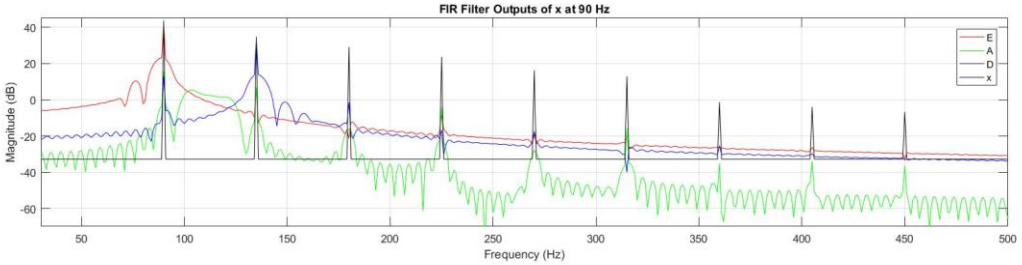


Figure 21. E-Band Detection

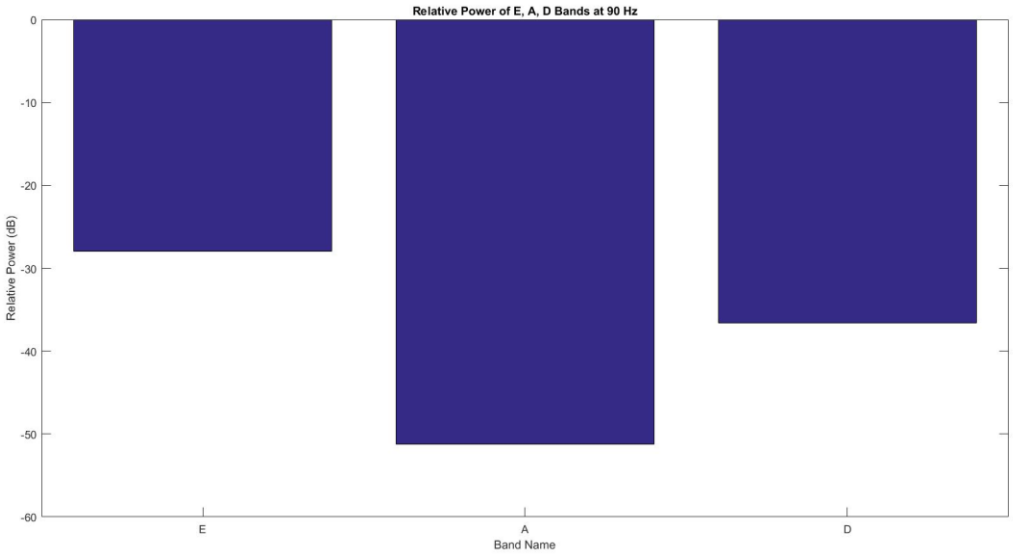


Figure 22. E-Band Detection via Power

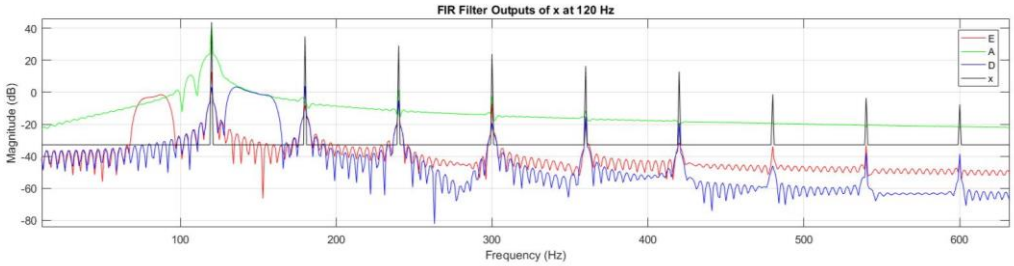


Figure 23. A-Band Detection

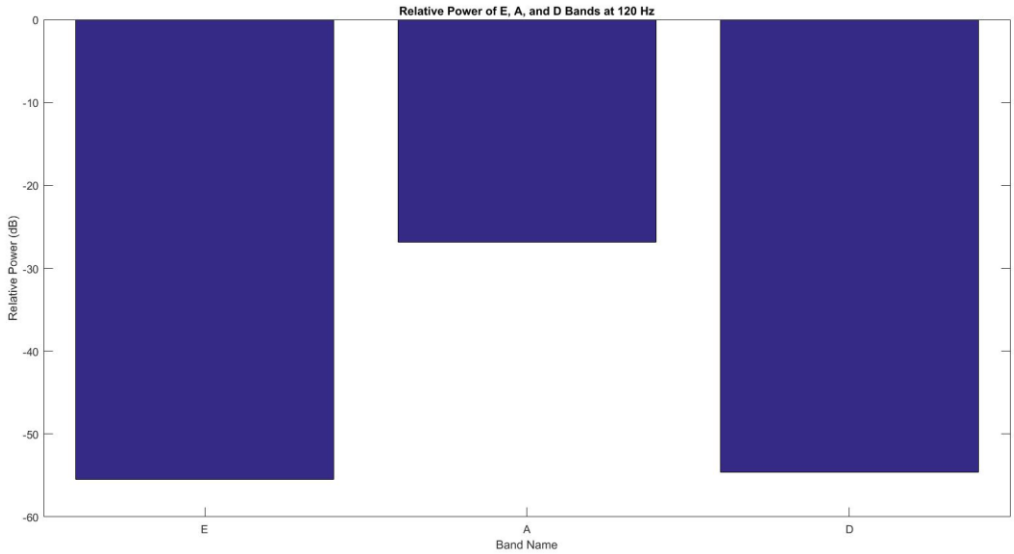


Figure 24. A-Band Detection via Power

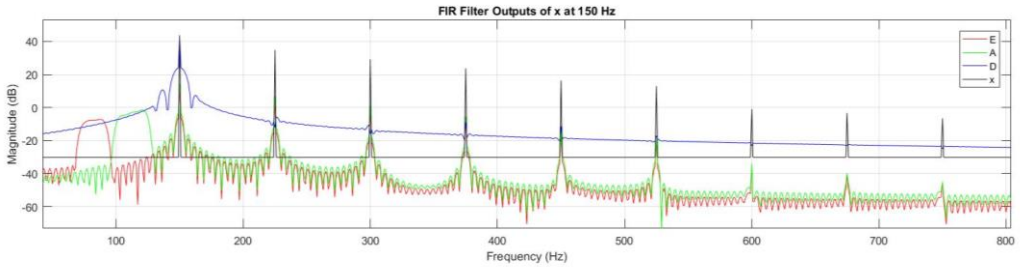


Figure 25. D-Band Detection

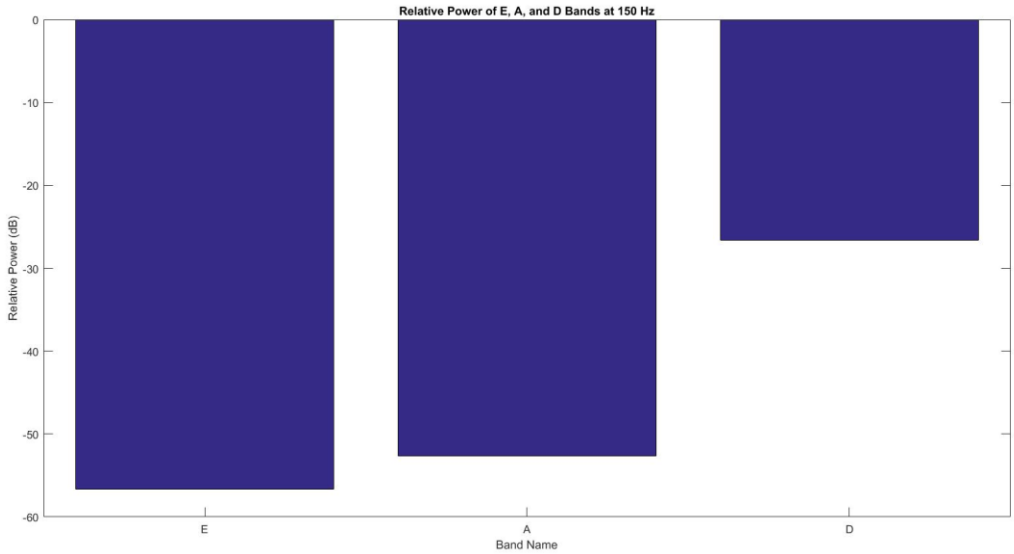


Figure 26. D-Band Detection via Power

## Part V-C. DFT Processing

This section mirrors many previous sections in terms of basic DFT processing. Due to the "false detection" described in part A, the selected band is already incorrect, therefore we deemed the adjustment to "cover" the band unnecessary. The figures below show the implementation of the tuner on E,A and D which shows the f0_est and cent approximation.

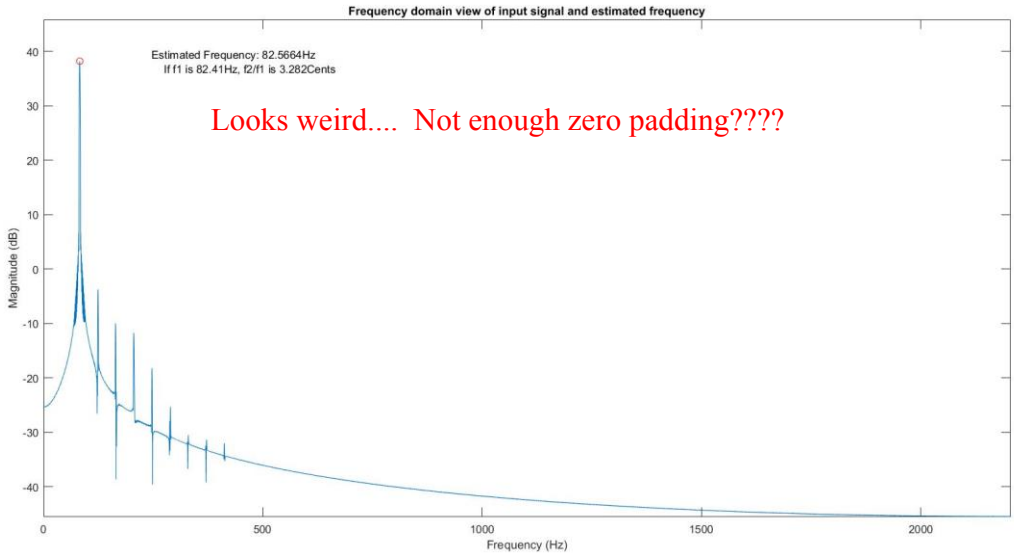


Figure 27. E-Note within 5 Cents

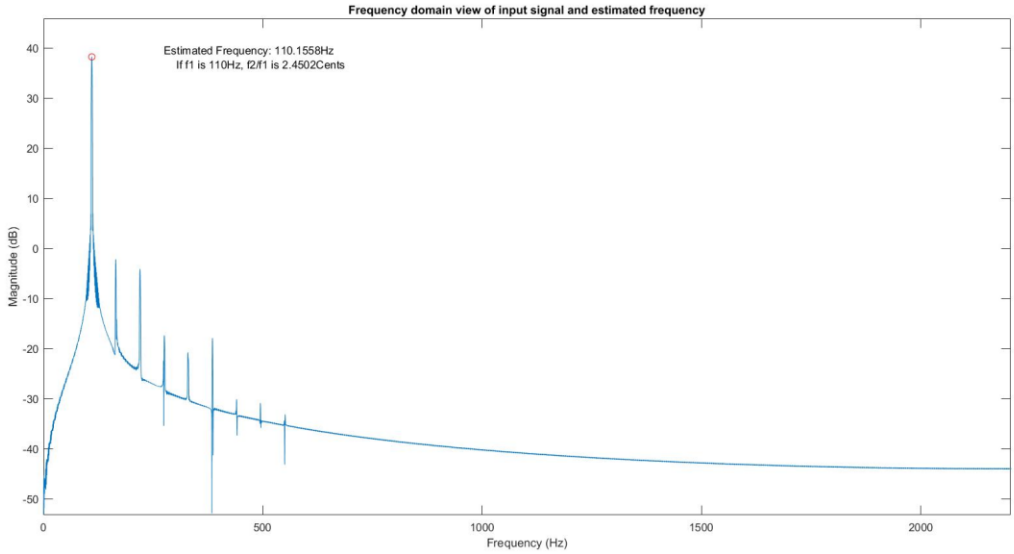


Figure 28. A-Note within 5 Cents

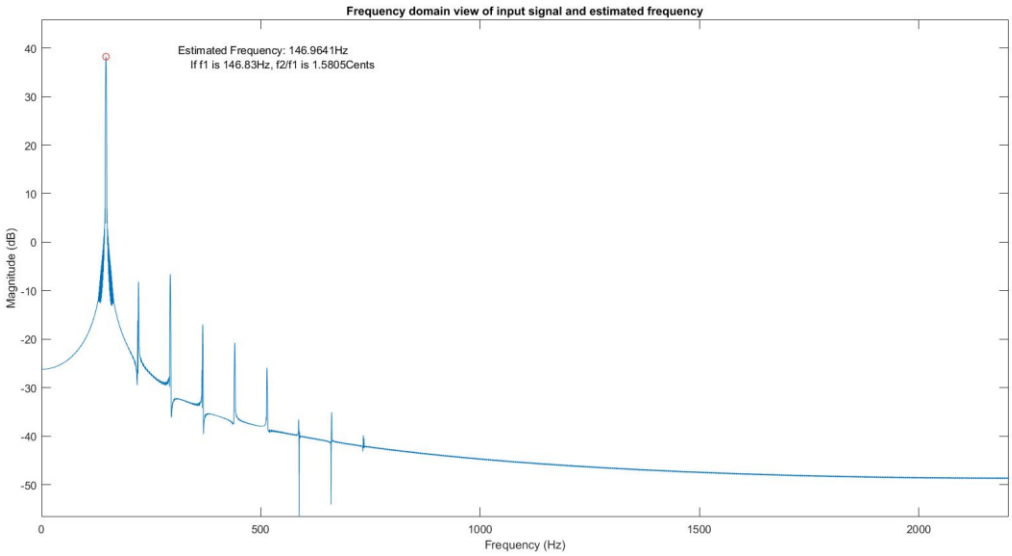


Figure 29. D-Note within 5 Cents

## Part V-D. Detecting Out-of-Range Notes

To detect notes outside of the 68 – 165 Hz specification in this section, we implemented one low pass and high filter below and above the range, respectively. These were specified similarly to the E, A, and D FIR filters in that they have less than 1000 coefficients and have 1 dB ripple and 60 dB stopband attenuation. Also, like Part B, the LP and HP filtered x signal was then summed using Parseval's Theorem and compared to the powers of the other bands. If a low or high detection was made no fundamental estimate resulted. For the sake of simplicity, the filtered signals plot excludes the curve for E, A, and D. Their relative powers can be seen in the corresponding bar charts.

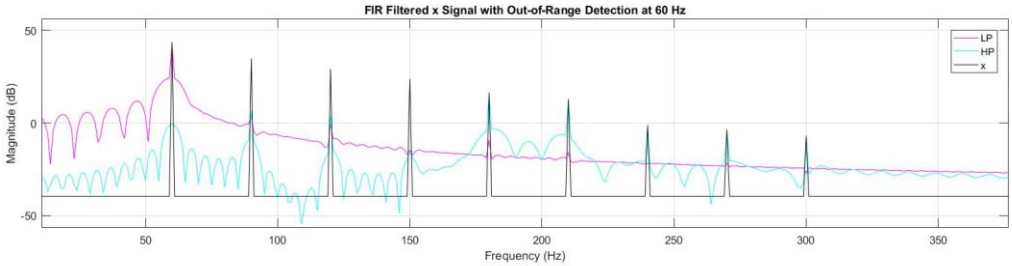


Figure 30. LP Band Detection

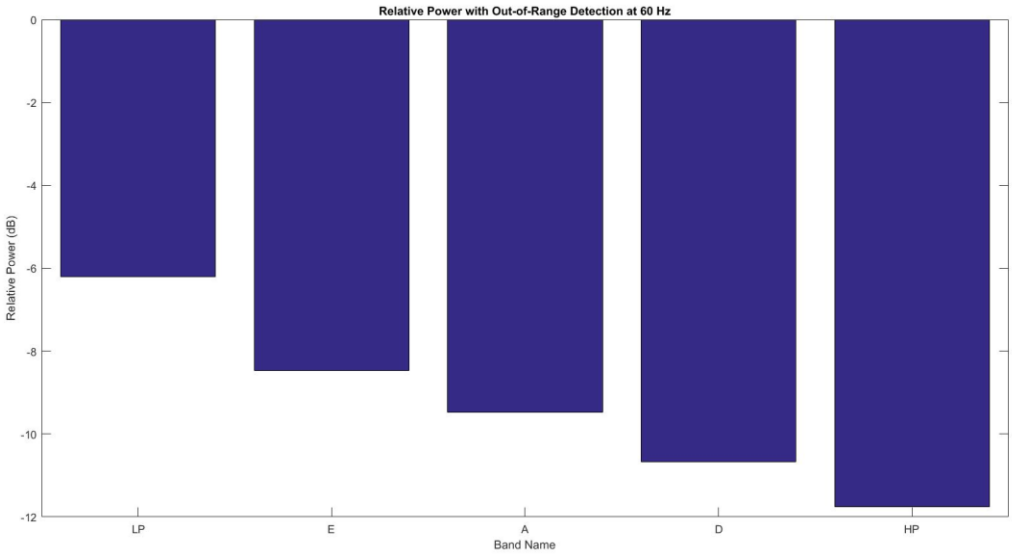


Figure 31. LP Band Detection via Power

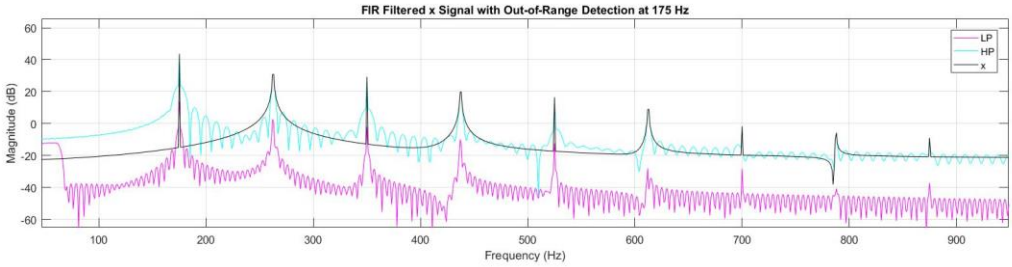


Figure 32. HP Band Detection

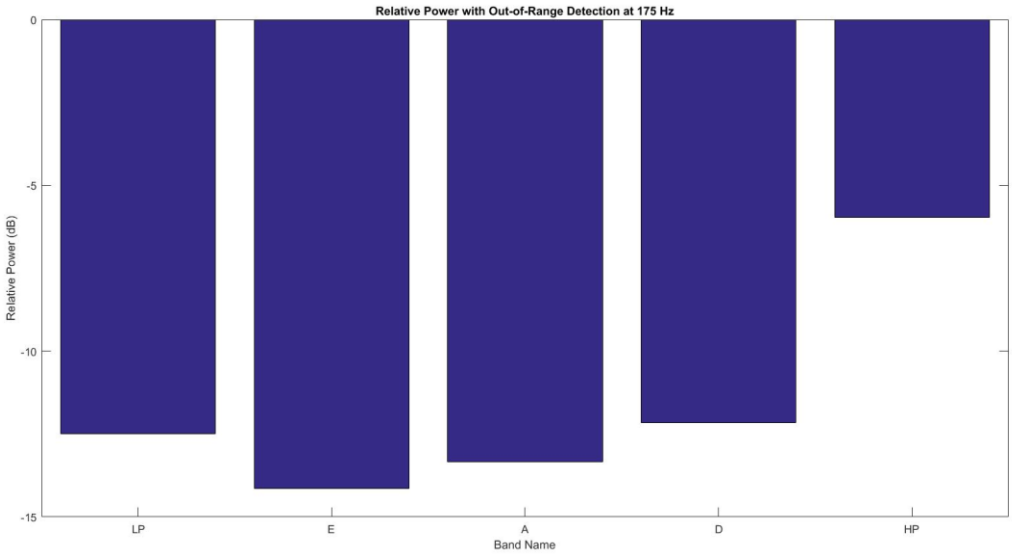


Figure 33. HP Band Detection via Power

## Part V-E. Testing

The "false detection" in FIR filter transition bands can be more clearly demonstrated now that power detection has been discussed. As is shown in Table above, there are certain fundamentals that are attenuated enough to allow the highest average power shift to another band thus causing an incorrect estimation of the fundamental frequency. This instead estimates the fundamental frequency of the second resonant peak of a given signal, so theoretically we could program our tuner to take the dead zone fundamental estimates, divide them by two and calculate the cent accuracy based on this value with the correct target frequency.

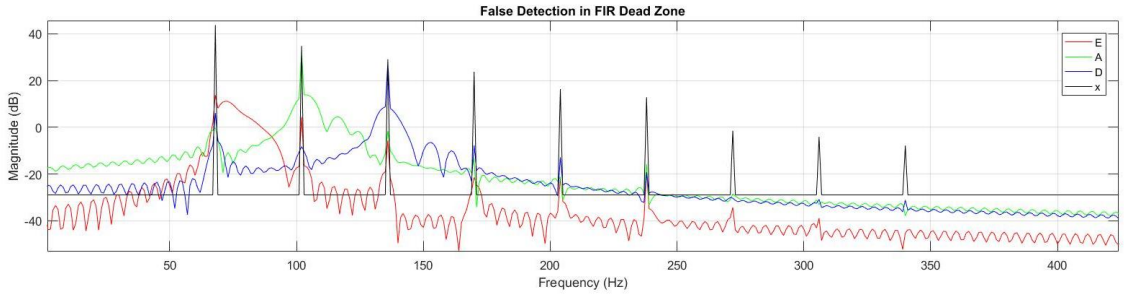


Figure 34. 68 Hz False Detection Plot

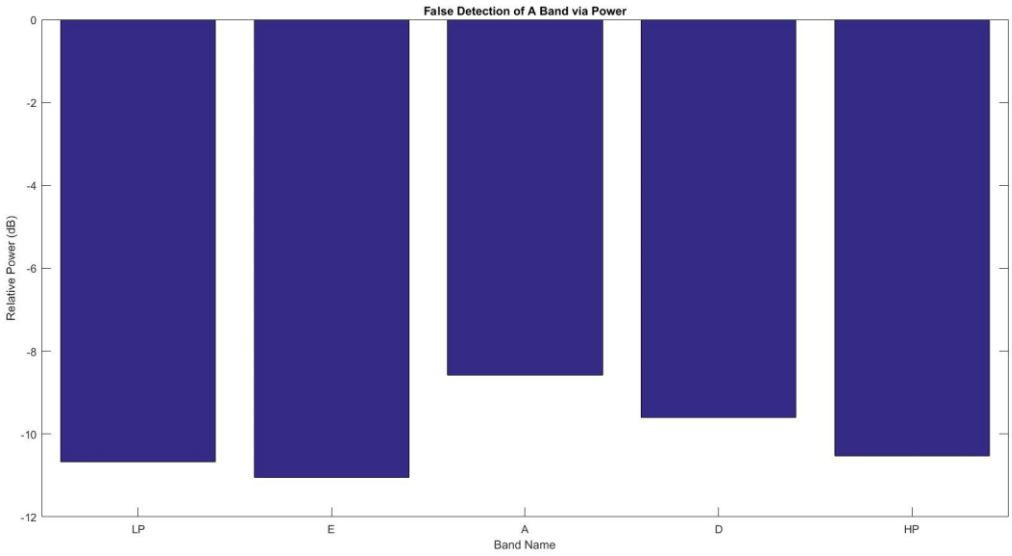


Figure 35. False Detection Relative Power

This clearly shows that the A band has a greater relative power than any other band which is explained by the attenuation of the fundamental resonant peak.

To gauge whether the Hamming window is essential in this application, we tested it without the Hamming window. The two yielded similar results with a small deviation in the cent estimation. This is bolstered by our theory and previous application of the Hamming window. The Hamming window is best used to reduce aliasing in a signal and therefore will not influence the power of a signal or where it lies on the frequency axis. When finding the maximum amplitude of a resonant peak, the unwindowed version causes the estimation to shift because of sidelobe and mainlobe attenuation which is seen in Part II of this project.

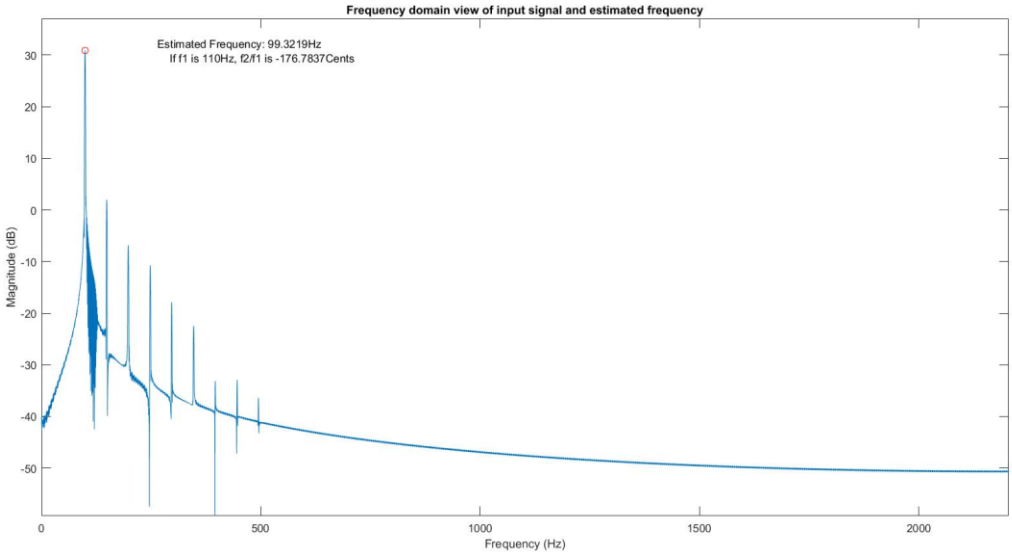


Figure 36. Hamming Window Applied

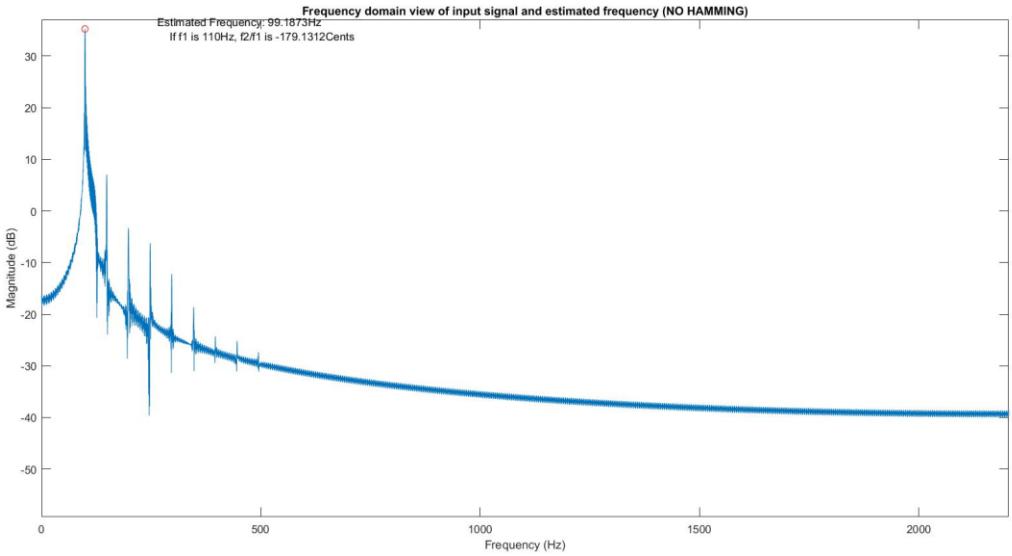


Figure 37. No Hamming Window Applied

## Conclusion

There were many specifications given in this project to describe the guitar tuner. By meeting some of these constraints, others were impossible to also achieve. One example of this was the specification of bandpass values for the three, overarching FIR filters. While most of these frequencies could be properly processed, there was a conflict with the limit of FIR coefficients. Having to limit the coefficients to under 1000, this could be thought of as a constraint on the computational power of whatever theoretical device our tuner is operating on. The transition band was the limiting factor in achieving this specification which significantly attenuated fundamentals that fell in FIR "dead zones". This was not what I was expecting and if this were an assignment for an employer, it would be very important to communicate the forced tradeoff between these two factors. By recognizing this flaw in the constraints of the tuner, I was able to think of a static way to correct for the problem without having to use a different type of filter or more computational power. This is obviously an example of what kind of thinking engineers must employ in their day to day jobs.

Another important theme of this project was the question of the Hamming filter. When trying to detect tiny discrepancies lost in mainlobe and sidelobe attenuation, it was important to remove the hamming filter to see what was happening. On the other hand, using the Hamming window drastically lowered the sound floor of a given signal. This forced us to consider the key features needed in the scenario given. This is another example of how an engineer must think by not letting one result say whether a window is good or bad, but can be useful or not depending on the situation. This kind of fluid thinking keeps us from making assumptions that will cause problems in the future.

The most important lesson of this project (in my opinion) was the chunking of information. Each part was designed to help digest information which would later prove crucial to understanding the core function of the tuner. This goes to show that it is important to study the different concepts of a complex system of ideas before mixing their applications together. In this way it is easier to think about the effects of each part individually. Specifically, this was useful when thinking about why zero-padding would produce a better cents estimation.