ECE 351 - Section 51

April 26 2022

Lab 12 - Final Project: Filter Design

Submitted By :
Jackie Lee

Contents

1 Introduction

In this lab we utilize our experience with Python and ECE 350 concepts this semester to design a filter that removes unwanted noise of a given input signal. We are given the range of frequencies where the information is located and are expected to design a filter and verify its capabilities through Python.

2 Equations

Transfer Function of Series RLC Bandpass Filter

$$H(s) = \frac{s\frac{R}{L}}{s^2 + s\frac{R}{L} + \frac{1}{LC}}$$

Bandwidth

$$BW = \frac{R}{L} = rad/s$$

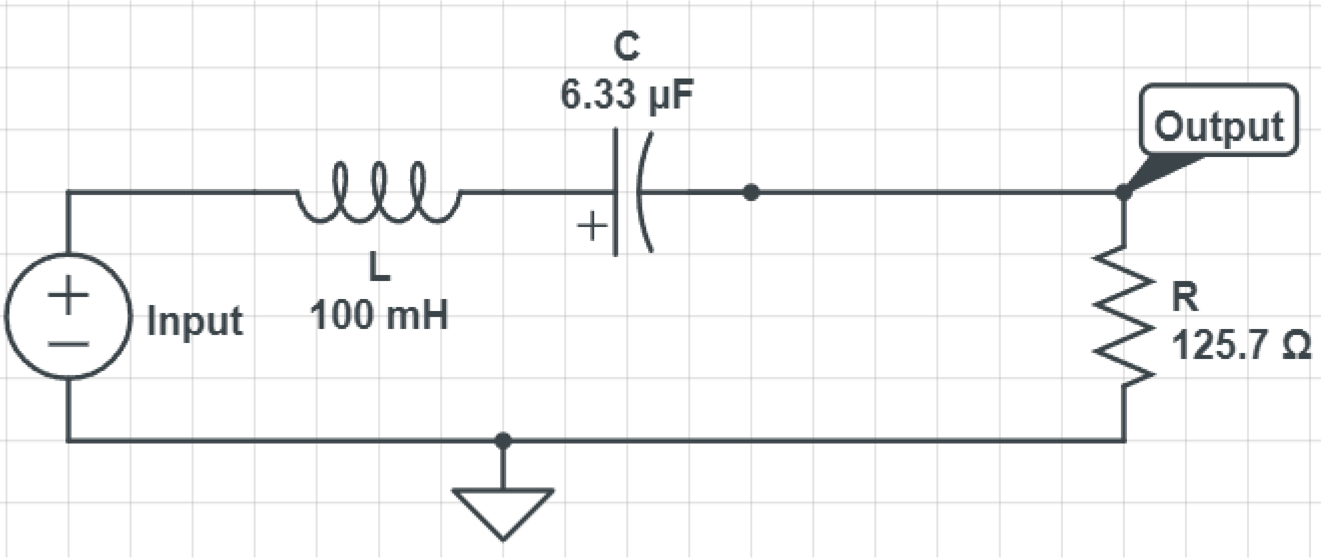
Center Frequency

$$w_o^2 = \frac{1}{LC} = rad/s$$

3 Methodology

To get started, the input signal was read using the provided code. From this, the noise is preventing the position measurement information from being readable so it needs a filter. A band-pass filter is necessary because there is an upper limit and lower limit frequency therefore only using a low-pass or a high-pass filter won't suffice.

I needed a refresher on band-pass filters so I went with a simple series RLC band-pass filter. I chose 1000 mH for the inductor because when I researched standard values of components, inductors were the most expensive so I went with its standard first and solved for R and C. I was able to solve for R using the bandwidth equation. This left only C which I used the center frequency equation, however when I initially went with 1900 Hz as the center frequency it did not lead to a cleaner output signal but using 200 Hz which happened to be the bandwidth, lead to the better signal. This lead me to believe either the equation was wrong or I failed to recall what the center frequency should be. My final circuit resulted in:



By using the bilinear and lfilter functions with the transfer function of the filter, an output signal can be plotted.

```
#%% Code that implemented the transfer function of the filter and
   generated the output signal from the filter
#Transfer Function in Symbolic Form#
num = [R/L, 0]
den = [1, R/L, 1/(L*C)]

#Filtered output#
z, p = sig.bilinear(num, den, 100000) #Up to 100k

outputSignal = sig.lfilter(z, p, sensor_sig)
```

I carried over the fast Fourier transform function from an earlier lab so that it can be used with the output signal and be plotted.

```
#%% Code that implemented fast Fourier transfrom
def fftfunc(x, fs):
    N = len(x)
    X_fft = scipy.fftpack.fft(x)
    X_fft_shifted = scipy.fftpack.fftshift(X_fft)

    freq = np.arange(-N/2, N/2)*fs/N

    X_mag = np.abs(X_fft_shifted)/N
    X_phi = np.angle(X_fft_shifted)
    return freq, X_mag, X_phi
```

Bode plots were implemented using the bode function to generate values in dB and semilogX to set the x axis.

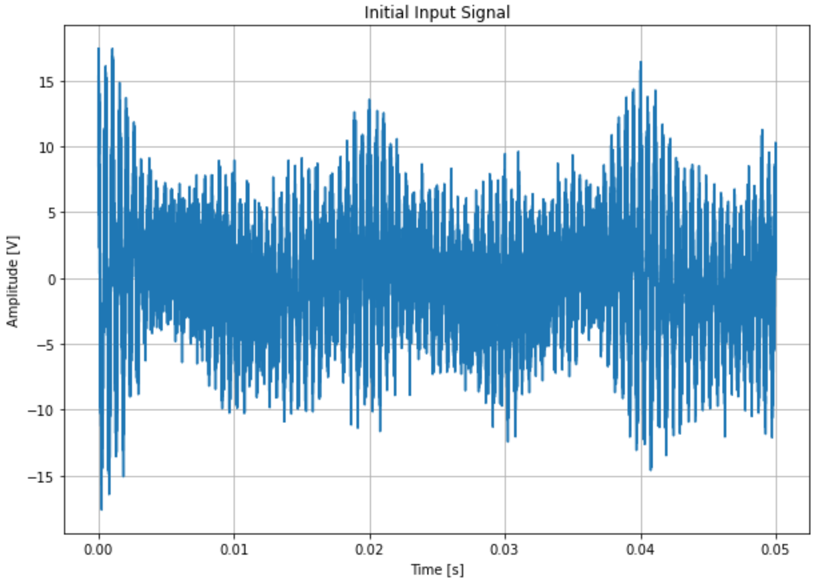
```python
#%% Code that demonstrates using the bode function and plotting a bode plot
sys = sig.TransferFunction(num,den) #Transfer Function implemented to be used with bode
w2, mag2, phase2 = sig.bode(sys, hz) #Generate the corresponding values for the Bode Plot

fig, ax = plt.subplots(2)
fig.suptitle("Bode Plot of the Whole Signal in Hz")
plt.tight_layout()
ax[0].semilogx(hz, mag2)
ax[0].set_ylabel("Decibel (dB)")
ax[0].set_title("magnitude")
ax[0].grid()
```

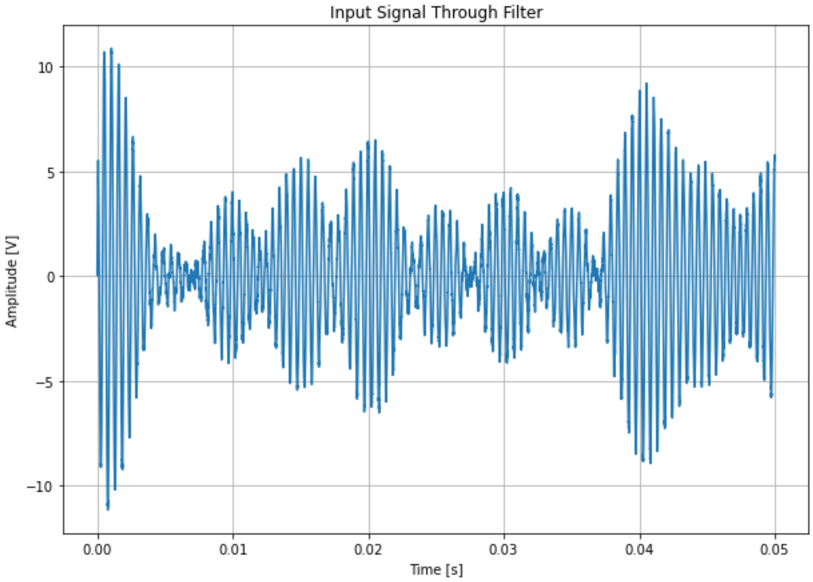
I was able to utilize the reference code for stem plotting in the handout to plot the fast Fourier transforms of the output signal. Up to this point I was able to generate the corresponding bode plots and filtered output signals. Lastly, I analyzed these plots to make sure that it fit the expectations of the filter based on the handout and what I expected a clean signal to look like.

4 Results

Input Signal:

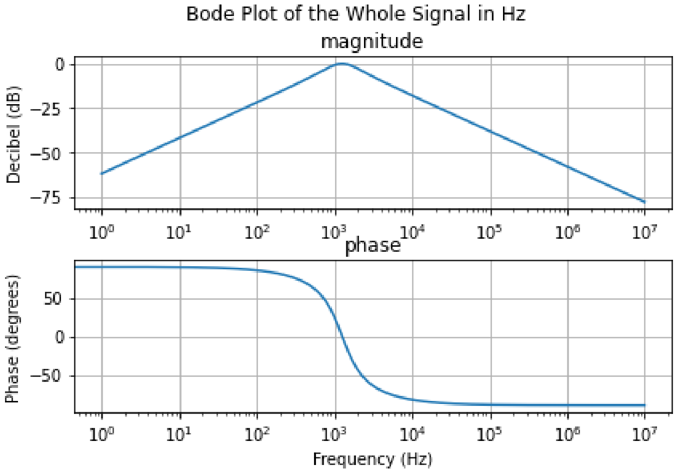


Filtered Input Signal:

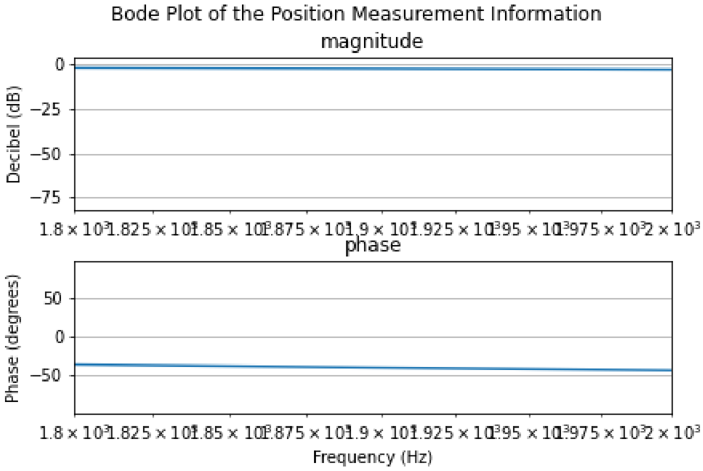


Between the two, the new plot is more readable and we can tell from the shape of both plots that they share the same function but now it removes the excess noise from the first plot which was indicated to be the switching amplifier and the Vibration noise.

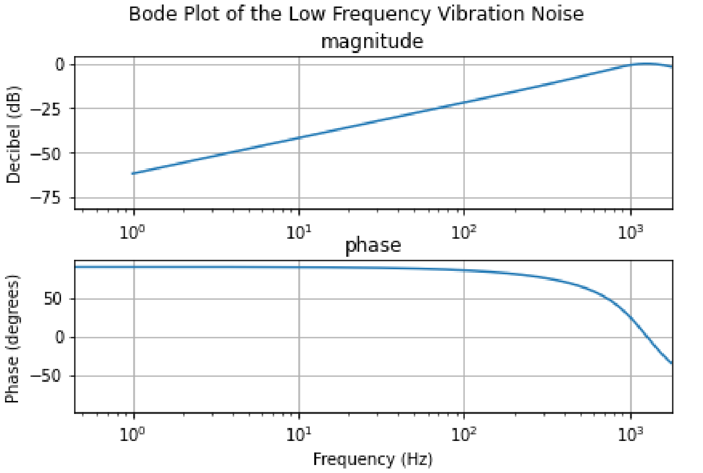
Bode Plots:



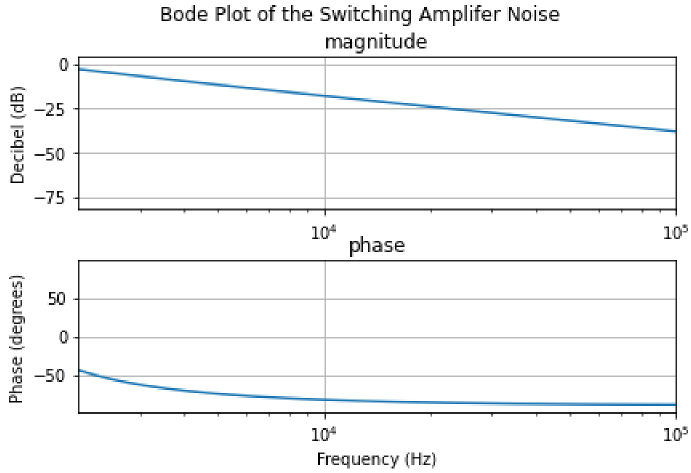
This plot shows the attenuation of the signals through all the frequencies. It matches expectation in that at the peak around 2000 Hz that it's dB is around 0 whereas the other values are attenuated.



Here we show the specifically the attenuation for the information that would like to be unchanged which is supported that from 1800 to 2000 Hz, the dB is around 0.

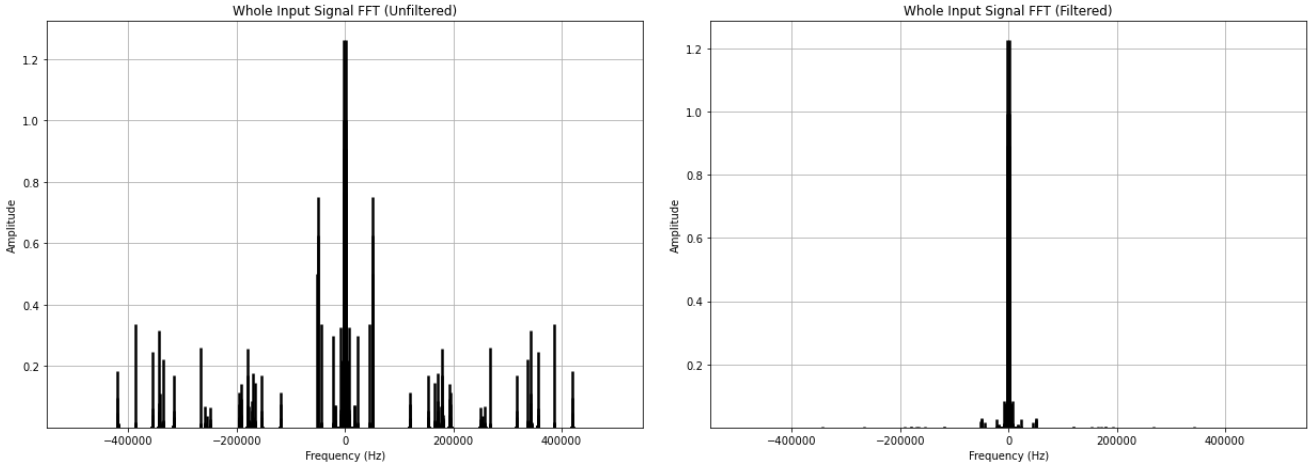


In this plot, we show that the low frequency vibration noise gets cancelled out by the filter since the dB gets lower as the frequency decreases past 1800 Hz.

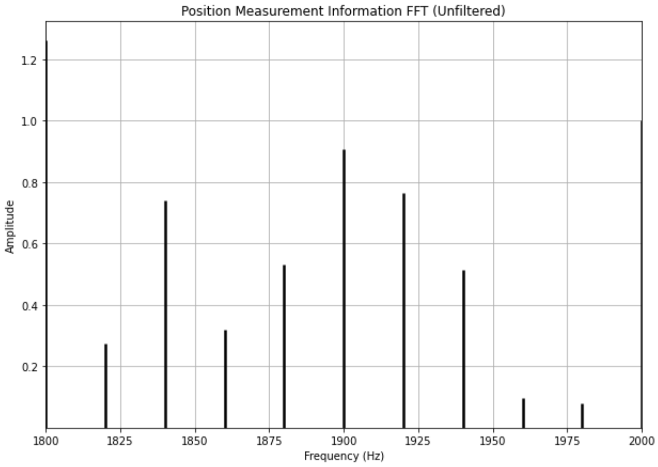


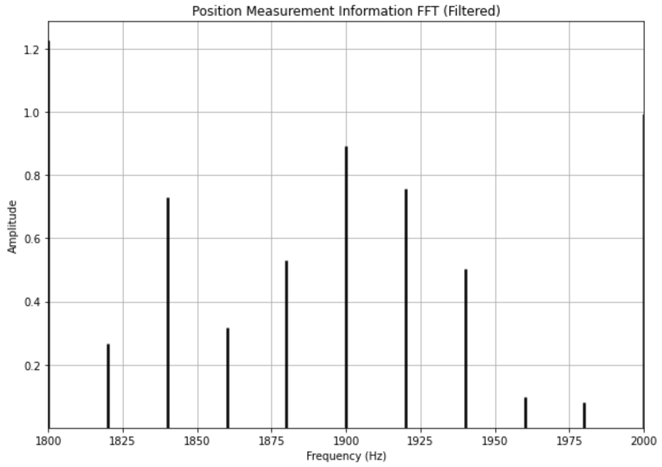
Similar to the last plot, this shows that for increasing frequencies past 2000 Hz, the dB decreases which will effectively remove the switch amplifier noise from the input signal.

FFT Plots:

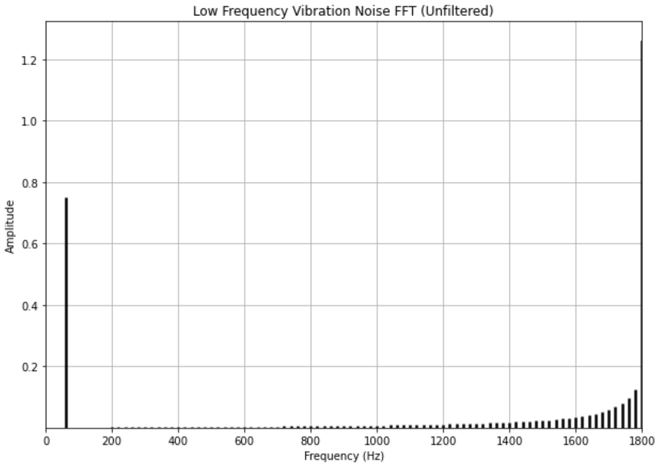


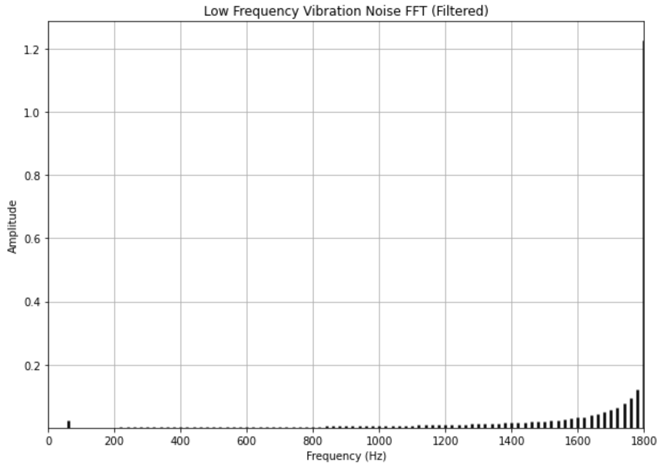
Similar to its respective Bode plot, the filtered plot shows that the small range of 200 Hz of the information is wanted to be kept is non attenuated compared to its noise counterparts where they are effectively filtered out. This is further supported by looking at its unfiltered plot where values outside of 1800 t0 2000 Hz range show noticeable values with a peak magnitude around .9.



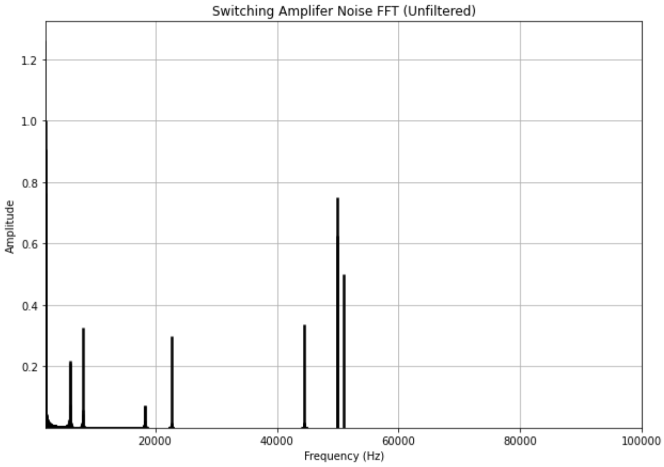


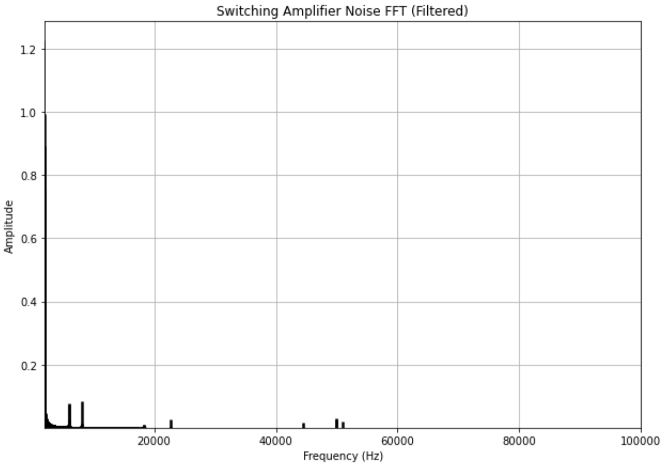
This shows that the information that was wanted from 1800 to 2000 Hz is unchanged when the input signal goes through the filter as the difference between the two plots is unnoticeable.





When comparing the two there is a noticeable low frequency noise spike below 200 Hz in the unfiltered plot where it has been removed in the filtered plot. This leads to the conclusion that the indicated low frequency noise is around 70 Hz with a magnitude of about .75 V.





Similar to the low frequency noise from the last set of plots, we can tell that the Switching Amplifier Noise has noticeable spikes above .5V around 50000 Hz, these along with the smaller ones are attenuated in the filtered plot. This leads to the conclusion that the magnitude and frequency of the Switching Amplifier Noise specified in the handout is .5V at 50000 Hz.

5 Error Analysis

I'd say the biggest problem that I had for this one was that the center frequency equation I used to find C didn't seem to go as expected. I also got confused between rad/s and Hz with the equations but since I was able to plot the signals and verify its attenuations around the specified frequencies, I was able to find a reasonable output due to the filter.

6 Questions

1. I'd say I accomplished what I wanted in this course, I first time experience with Python and feel more comfortable with it. I can't say all of this is transferable to CS courses with Python since it did involve ECE 350 concepts but it made me comfortable with the language. Plus as an added bonus I was able to create Latex Documents and organize my work through Github.

7 Conclusion

By the end of the lab I was able to utilize Python and my knowledge from my ECE courses to design a filter that would filter out unwanted noise from a signal. In this case it was the low frequency noise and the switching amplifier noise. This would result in only passing in the Position Measurement Information from the signal between 1800-2000 Hz. This experience and verification method could be implemented in different scenarios involving different frequencies.

Bibliography

Dennis M. Sullivan, Signals and Systems for Electrical Engineers I, 2018