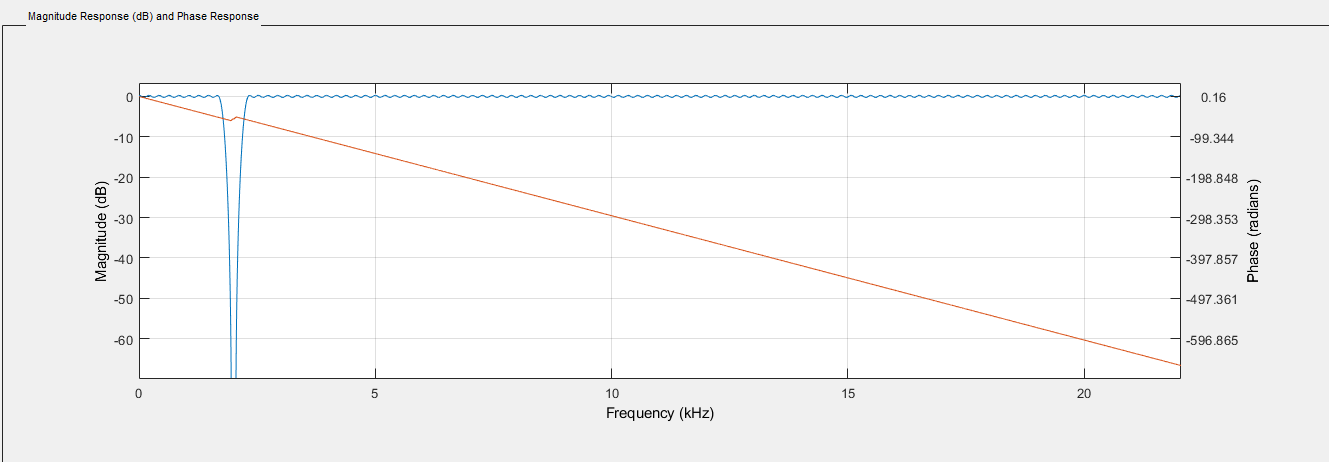
**1. Description**

In this portion of the project we are filtering out the sine noise that was introduced in Phase 1. MATLAB is used to both design the filter header files as well as find the dominant spectral component. The filter is to have a minimum attenuation of 60 db. A c++ program is to be written that will filter out the sine noise sample by sample. The total execution time needs to be measured so that I can observe the tradeoff of using different quality filters. The program expects a wav file with a sampling frequency of either 44100 Hz or 22050 Hz. The program runs much faster with 22050 Hz .wav file because the filter does not have as many coefficients, but the audio quality isn’t as good. I used a 44100 Hz audio file, but the 22050Hz file also works fine.

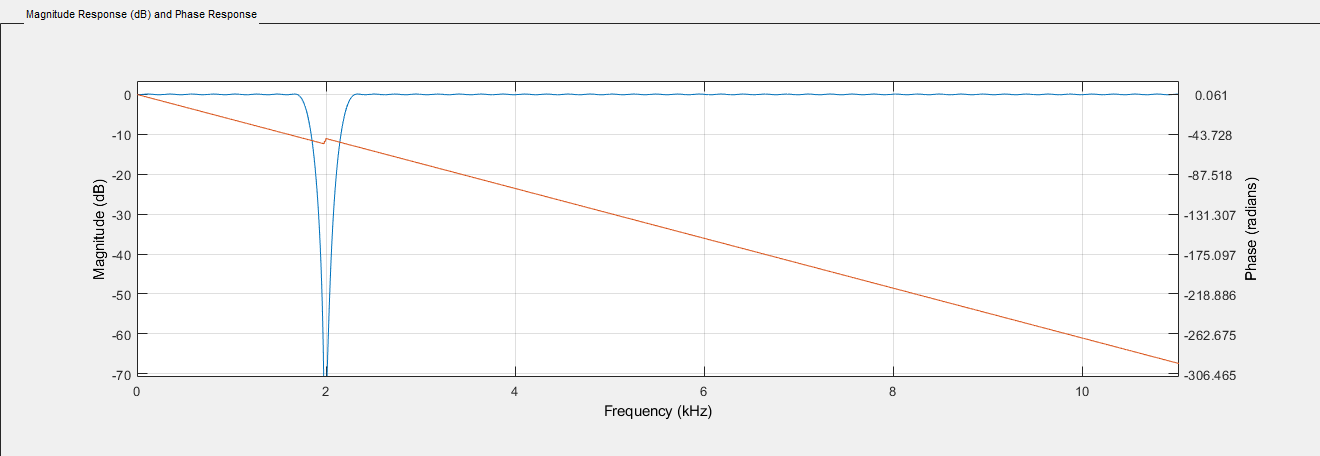
**2. Solution (MATLAB)**

I designed a bandstop FIR filter for 44100 Hz and 22050 Hz. I chose this type of filter because I knew that the sine noise occurred at 2000 Hz, and I want to leave the 2500 wave alone. I made my FPass1 1700 Hz and my FStop1 1953 Hz. My Fstop2 is 2050 and my Fstop2 is 2300 Hz. This will cut off the 2000 Hz wave but leave the 2500 Hz wave alone.

The magnitude and phase response of the 44100 Hz filter is shown below.



The magnitude and Phase response of the 22050 Hz filter is shown below.

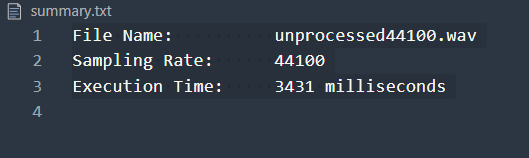


These filters were exported as c header files. They were then used in the c++ portion of the project.

**3. Solution (c++)**

The project was compiled using g++. The two filters are present as header files. Another file called tmwtypes.h is included as well. The wave file is read sample by sample just like in the first part of the project. Two arrays are created. They are the same size of the filter arrays. I then used the convolution formula on each sample. The sample is then written to a new .wav file.

The summary output of the program is below.

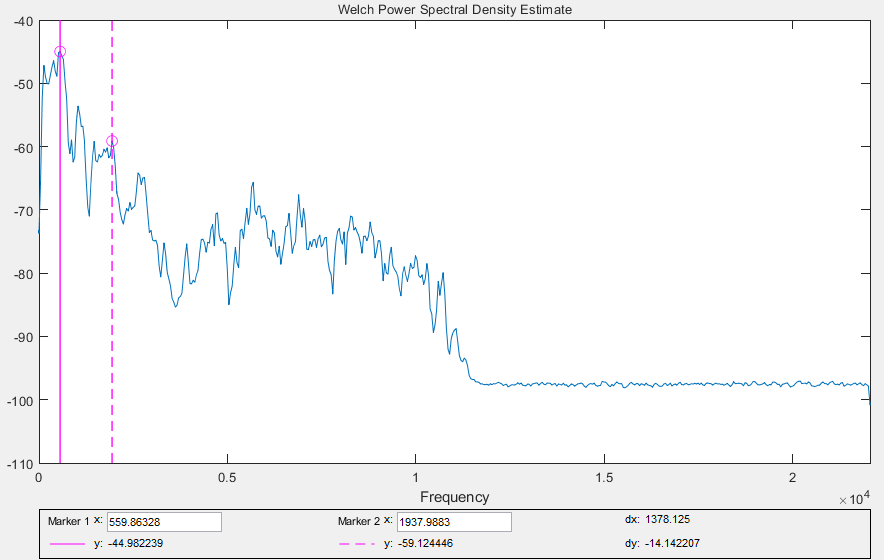


The program ran in a shorter time than the .wav file therefore it can run in real time.

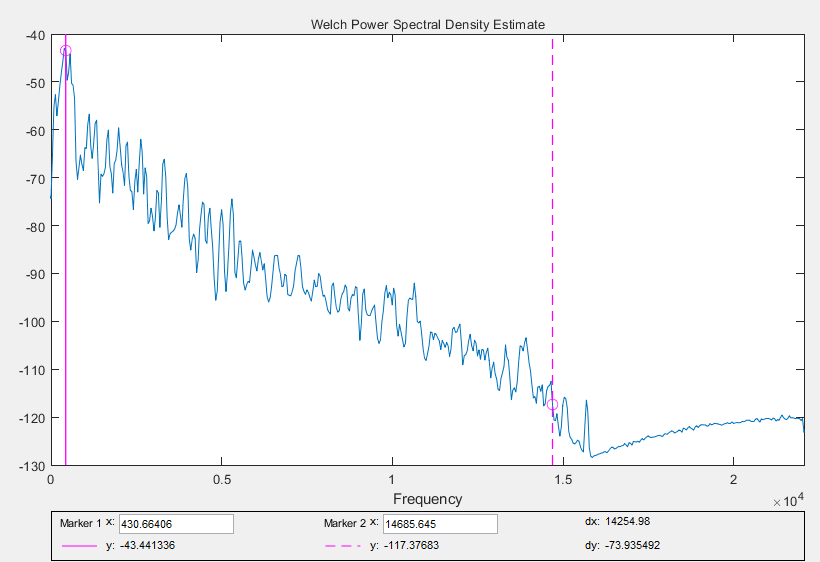
**4. Spectrum Analysis**

The spectrum of the unmodified file is below

Less than 10 seconds

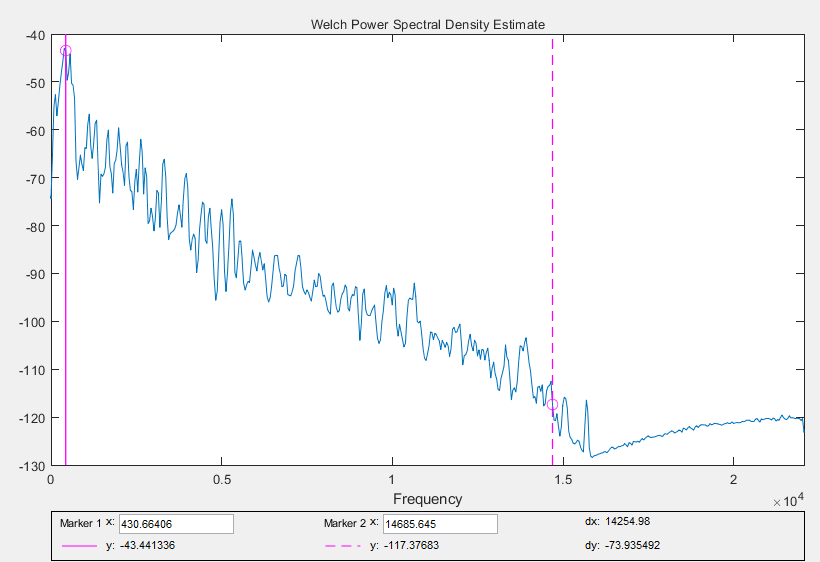


Greater Than 10 Seconds

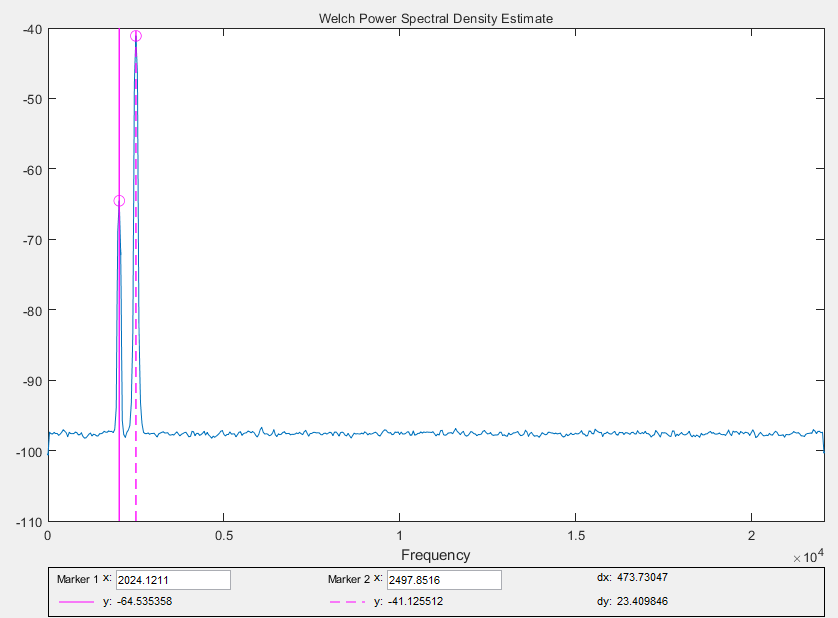


The spectrum of the file that was modified in phase 1 is shown below.

Greater Than 10 Seconds



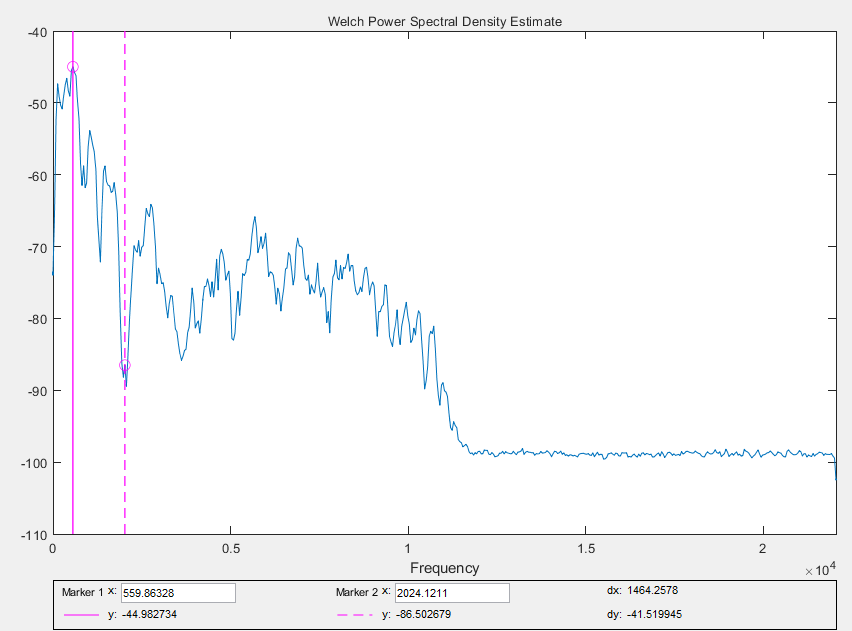
Greater Than 10 Seconds



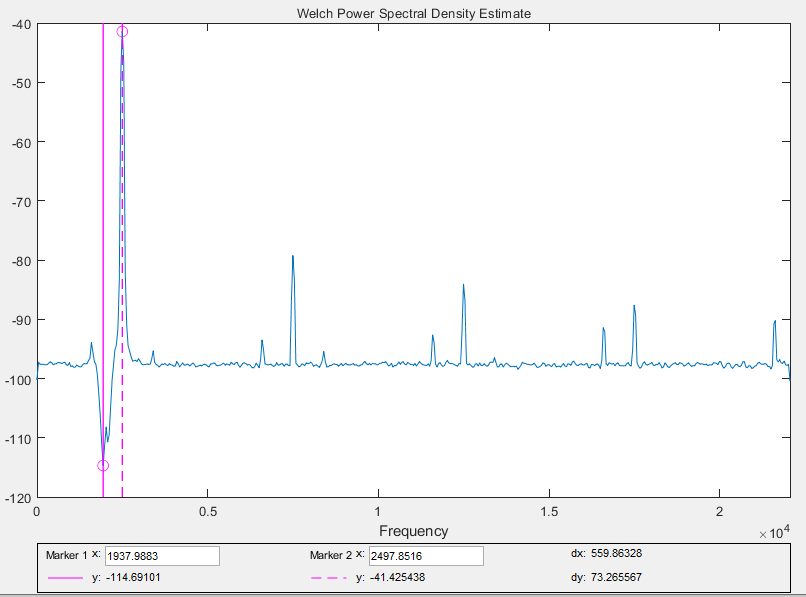
You can clearly see the 2000 Hz wave and the 2500 Hz wave. Everything else has been overwritten.

Below is the spectrum of the processed wav file.

Less Than 10 Seconds



Greater Than 10 Seconds



The 2000 Hz wave is clearly eliminated, but the 2500 Hz signal is left intact.

**5. Conclusion**

All in all, this project taught me the structure of a .wav file as well as real time data processing. We added random sine noise of different frequencies to a .wav file and filtered one of the sine waves out while leaving the other alone. I designed a bandstop filter to remove a small range of frequencies and used the convolution theorem to implement it.