Guitar Pedal Filter Project Report

1.0 Introduction

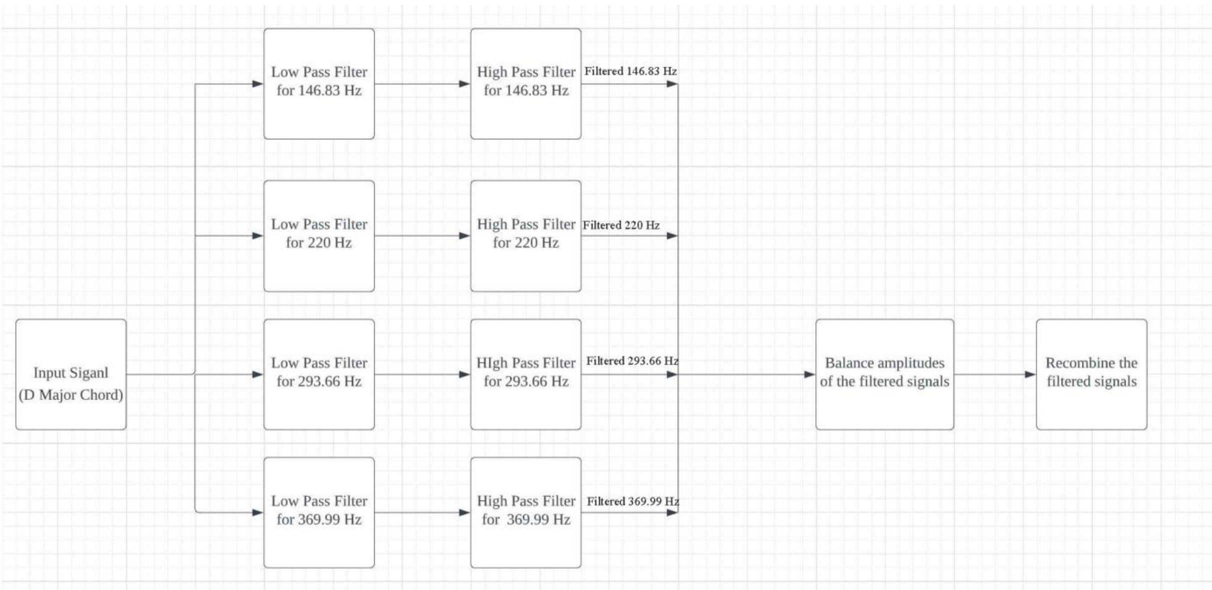
Guitar pedals are used to modify (i.e., filter) the sound and frequencies of an electric or acoustic guitar. They are typically located on the ground, hence the name 'guitar pedal'. The signal is picked up by either a microphone of an electromagnetic pickup, then put through the guitar pedal(s), and then into the amplifier/speaker. These guitar pedals can be analog or digital. Digital guitar pedals can be designed using MATLAB and digital signal processing filters.

There is a common guitar pedal called compression pedal. This pedal compresses the volume (i.e., amplitude) of different pitches, i.e., making low volumes pitches higher volume and making high volume pitches a lower volume. This evens out the sound (with all amplitudes being close to equal) and is a popular effect for the guitar as many strings are played at once.

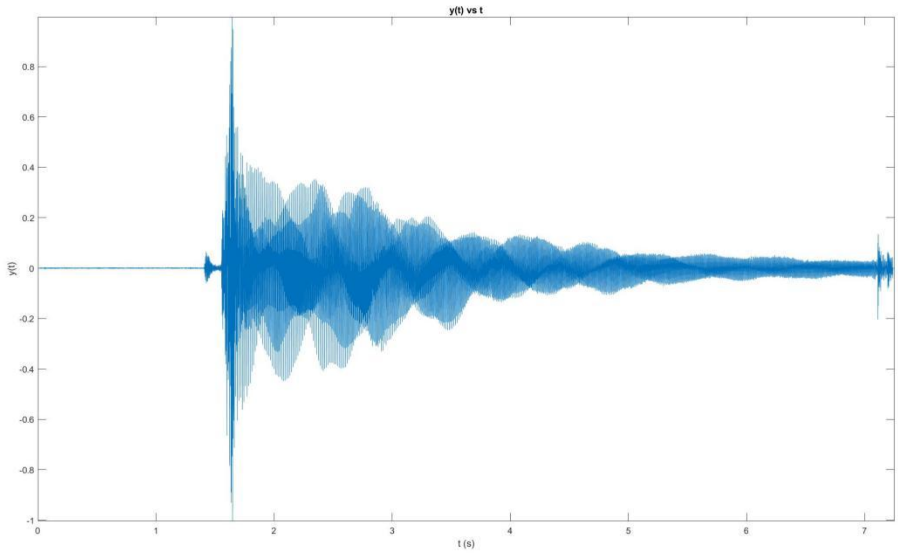
The project uses MATLAB to design the digital signal processing filters for a digital guitar compressor pedal. It isolates four desired signals of D Major Chord and recombines them with scaled amplitudes.

2.0 Discussion

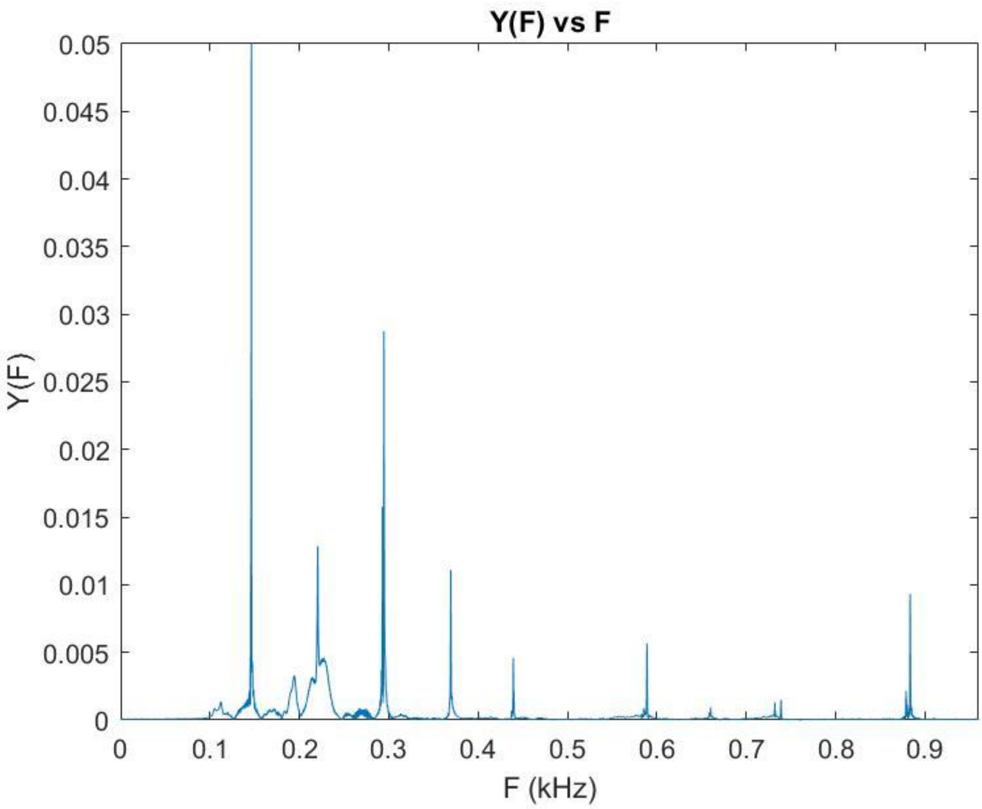
2.1 Flow Chart



2.2 Original Audio Signal Analysis
The following graph is the original audio signal in the time domain.



The following graph is the original signal in the frequency domain.



2.3 Signal Filter Analysis

2.3.1 Justification of filter parameters
As described in the flow chart, the bandpass filter for each signal consists of a low pass filter and a high pass filter. The MATLAB function Cheby1 is implemented to

design a filter. There are three input parameters for Cheby 1 function including n, Rp, Wp.

2.3.1.1 Order N

N is the order of the filter. When n is relatively small (e.g n = 3), the filtered signal could include some signals at other frequencies besides the desired frequency. It means that there are some errors when n is a small value. While increasing the value of order n, there are fewer signals at other frequencies appearing in the frequency domain. Finally, when n equals 7, there are almost no other frequencies besides the desired one. It means that there are fewest errors when n is 7.

2.3.1.2 Ripple Rp

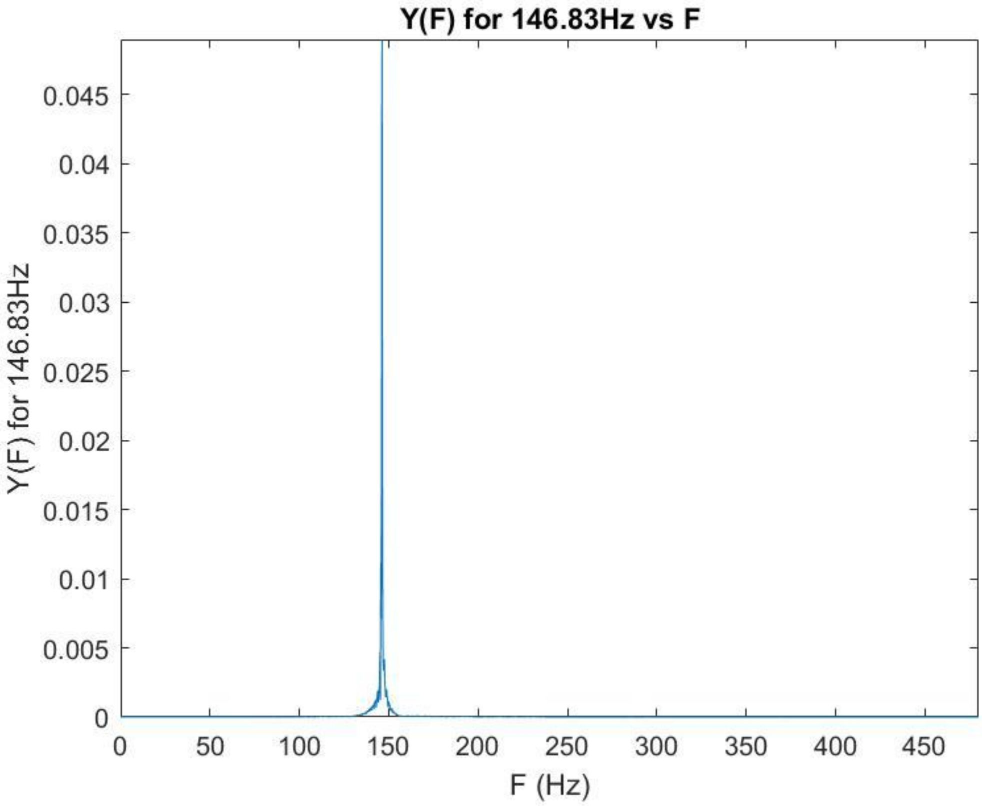
Rp is the peak-to-peak bandpass ripple. When Rp is relatively large (e.g Rp = 9), the filtered signal could include some signals at other frequencies besides the desired frequency. It means that there are some errors when Rp is a large value. While decreasing the value of ripple Rp, there are fewer signals at other frequencies appearing in the frequency domain. Finally, when Rp equals 2, there are almost no other frequencies besides the desired one. It means that there are fewest errors when Rp is 2.

2.3.1.3 Passband Edge Frequency Wp

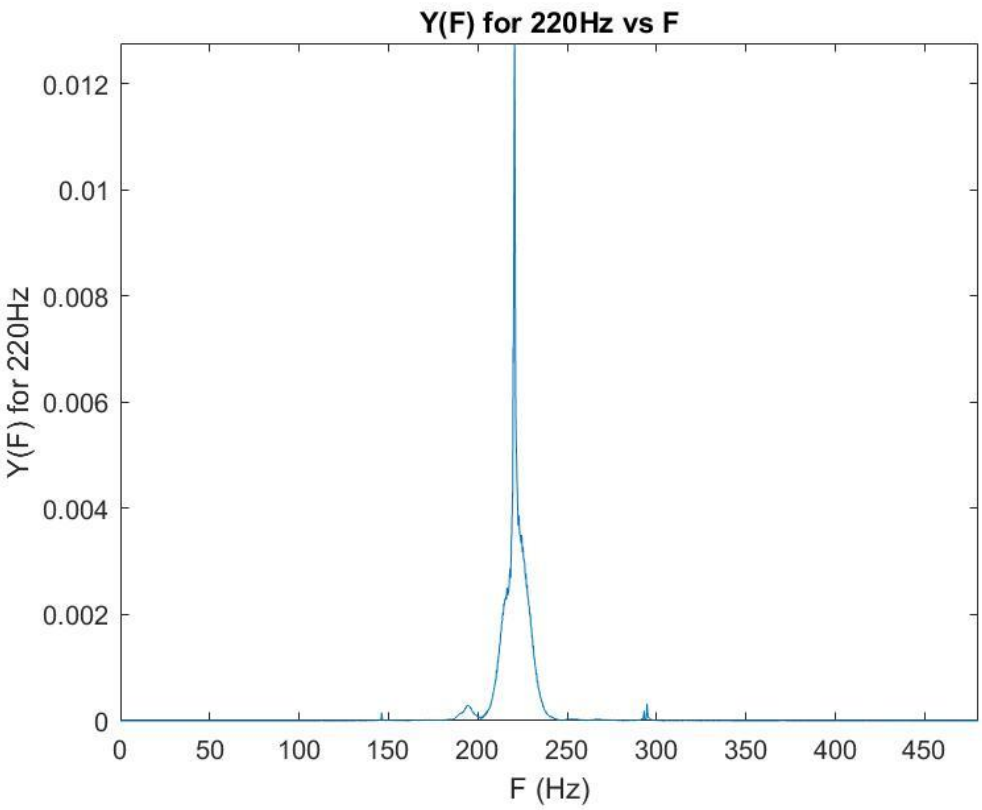
Wp is the normalized cut-off frequency. When Wp is applied to a low-pass filter, the filter can allow the signals under Wp to pass and filter the signals over Wp. When Wp is applied to a high-pass filter, the filter can allow the signals over Wp to pass and filter the signals over Wp. For instance, it is expected to design a series of filters to filter the signal at 146.83 Hz. In the low pass filter, the cut-off frequency is 146.83 + 5 = 151.83 Hz. After normalization, the Wp of the low-pass filter is 0.0063 Hz(151.83/(1/2*sampling frequency)). In the high pass filter, the cut-off frequency is 146.83 - 5 = 139.83 Hz. After normalization, the Wp of the high-pass filter is 0.0059 Hz(139.83/(1/2*sampling frequency)). To sum up, the cut-off frequency of a low-pass filter is a bit higher than the desired filtered frequency. The cut-off frequency of a high-pass filter is a bit lower than the desired filtered frequency.

2.3.2 Frequency Response of Bandpass Filters

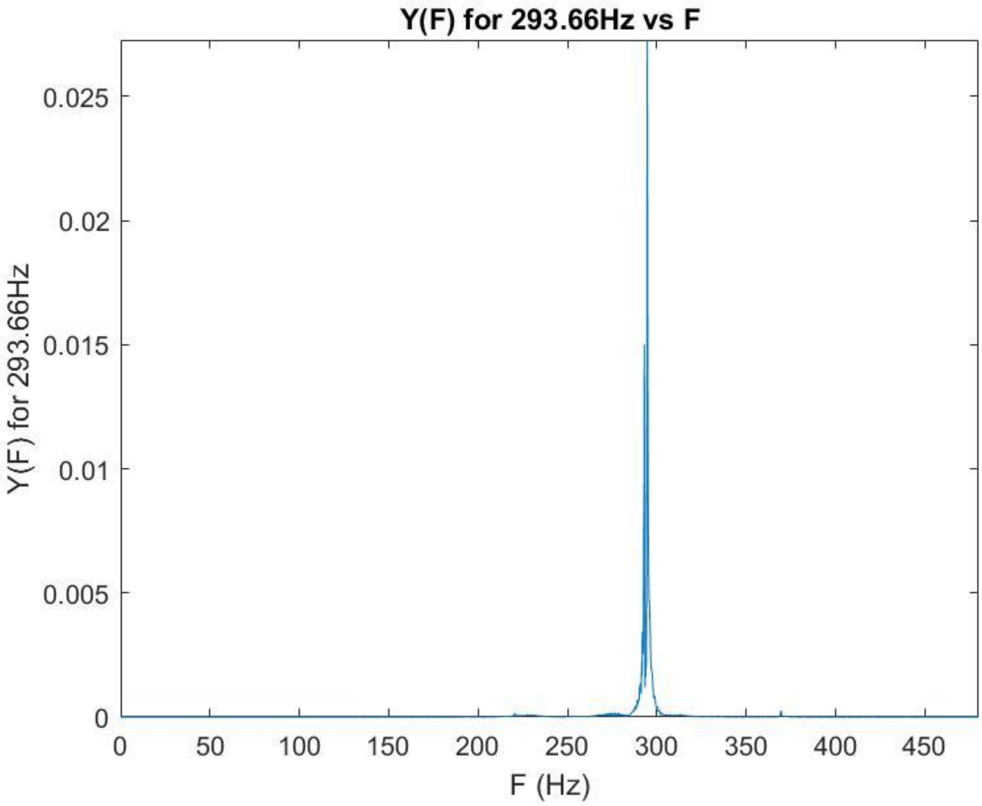
The following graph is the frequency response of the bandpass filter for 146.83 Hz (D3).



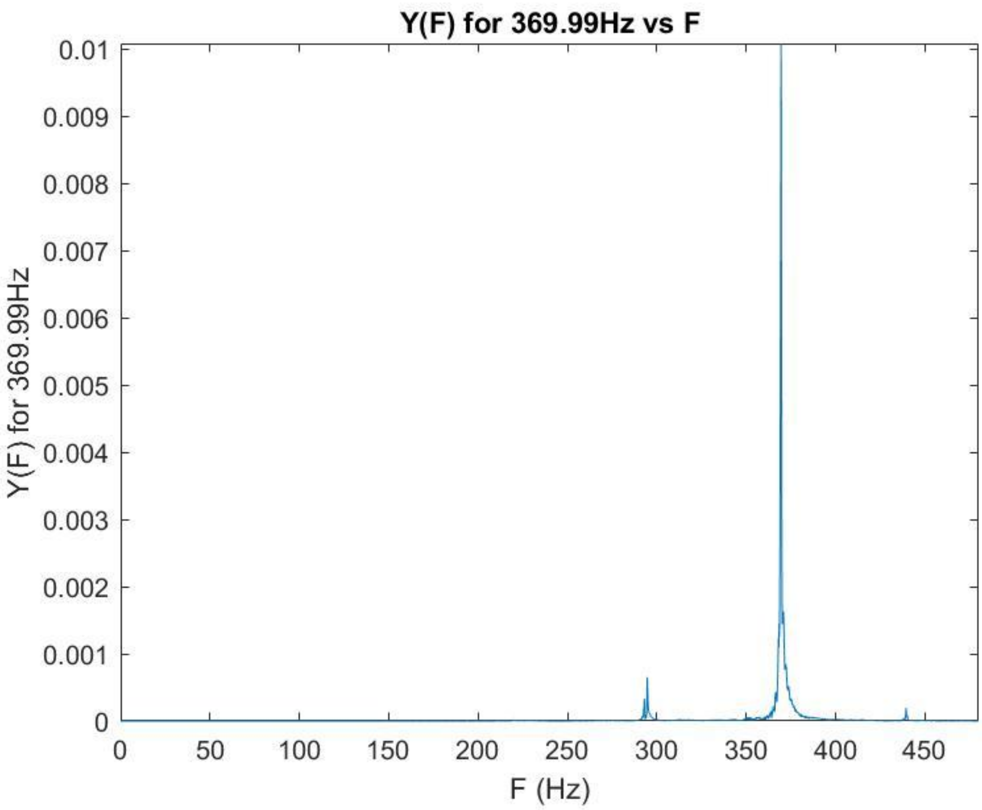
The following graph is the frequency response of the bandpass filter for 220 Hz (A3).



The following graph is the frequency response of the bandpass filter for 293.66 Hz (D4).



The following graph is the frequency response of the bandpass filter for 369.99 Hz (F#4).



2.3.3 Amplitudes

The following graph shows the amplitudes of each pitch.

```
Command Window

>> amplitude_D3

amplitude_D3 =

    0.0490

>> amplitude_A3

amplitude_A3 =

    0.0128

>> amplitude_D4

amplitude_D4 =

    0.0273

>> amplitude_F4

amplitude_F4 =

    0.0101
```

The amplitude of 146.83 Hz (D3) is 0.0490;

The amplitude of 220.00 Hz (A3) is 0.0128;

The amplitude of 293.66 Hz (D4) is 0.0273;

The amplitude of 369.99 Hz (F#4) is 0.0101.

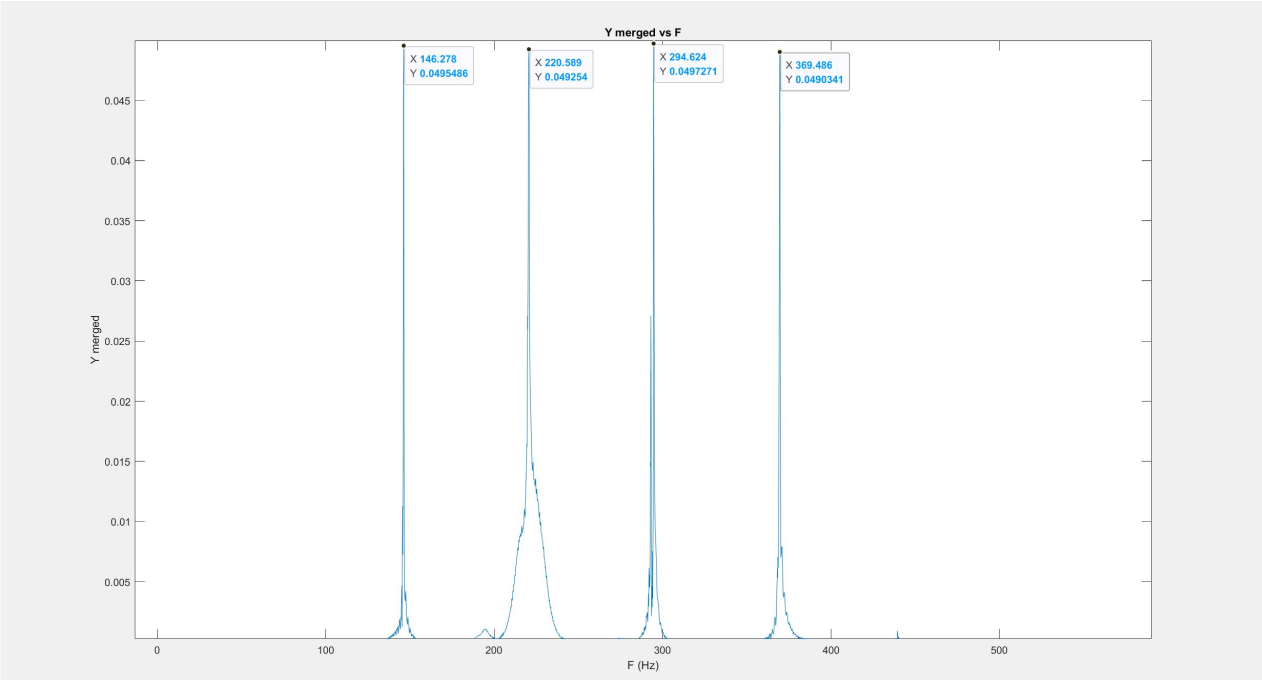
2.4 Compression

2.4.1 Scaling

After filtering, the design collects filtered signals and balances their amplitudes. It scales four pitches' amplitudes together to the greatest among them, which is the amplitude of 146.83 Hz (D3) (i.e. 0.0490).

2.4.2 Recombination

Four scaled pitches with the same amplitude are recombined to form a D major chord again.



After recombination, the amplitude of 146.83 Hz (D3) is 0.0495. The amplitude of 220.00 Hz (A3) is 0.0493. The amplitude of 293.66 Hz (D4) is 0.0497. The amplitude of 369.99 Hz (F#4) is 0.0490. All errors between amplitudes of signals are within 10% of each other, which meets the project requirements.

3.0 Conclusion

In conclusion, the project completes a digital guitar compressor pedal design. It isolates four desired signals and recombines them with scaled amplitudes. The errors between amplitudes of signals are smaller than 2% of each other. It means that the final results are within the specifications for compression.