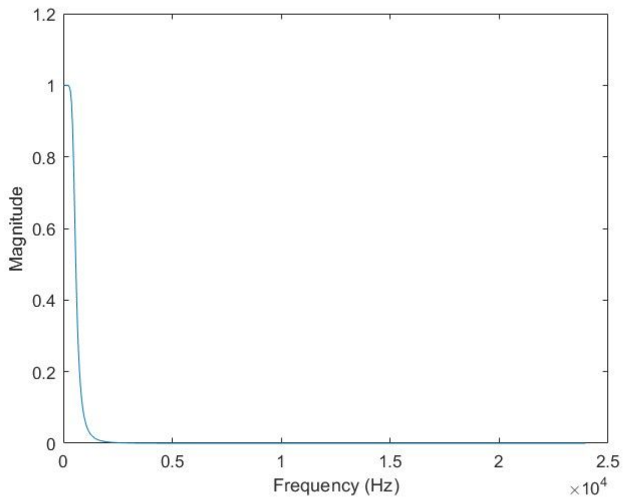
CMPE 362 - HW 3

Muhammet Ali Topcu - 2020400147

Firstly, I declared each filter's frequency interval in variables fc, fh, fpass and sampling frequency in fs. The filters' order is 4 and length of the filters are 4 + 1. Each of the filters are FIR filters.

1. Lowpass FIR Filter



To separate the sound of the drum kick, we need a lowpass filter. So, I used MATLAB's butter filter which takes order, normalized cutoff frequency and filter type and returns the function coefficients. Butter filter's formula is as given:

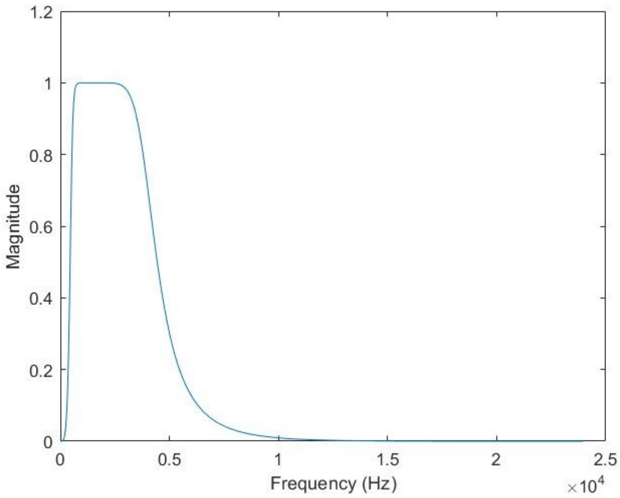
$$H(z) = \frac{B(z)}{A(z)} = \frac{\mathrm{b}(1) + \mathrm{b}(2)\, z^{-1} + \cdots + \mathrm{b}(\mathrm{n}{+}1)\, z^{-n}}{\mathrm{a}(1) + \mathrm{a}(2)\, z^{-1} + \cdots + \mathrm{a}(\mathrm{n}{+}1)\, z^{-n}}.$$

Then, with MATLAB's freqz function, I get frequency response of the filter to plot with plot function of MATLAB.

Lastly, I applied the filter function with parameters function coefficients of the filter and sampled data of the original audio. I applied the filter twice so that it separates better.

As it could be seen from the graph of the frequency, Lowpass filter separates sounds with lower frequencies and attenuates sounds with higher frequencies. So, in the output kick.wav file, we mostly hear drum kick sound.

2. Bandpass FIR Filter



To separate the sound of the piano chords, we need a lowpass filter. So, I used MATLAB's butter filter which takes order, normalized cutoff frequency and filter type and returns the function coefficients. Butter filter's formula is as given:

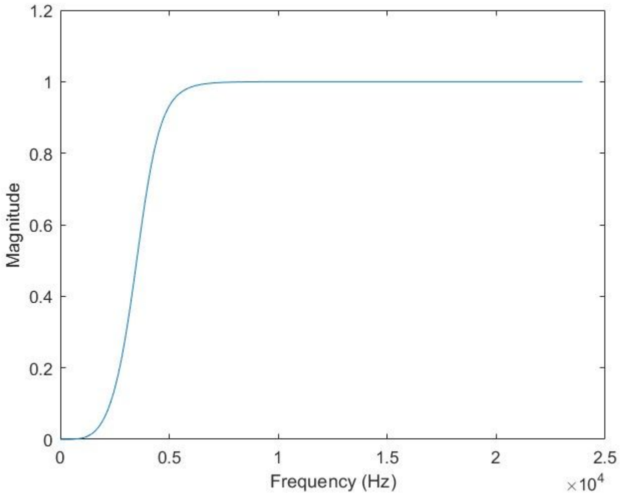
$$H(z) = \frac{B(z)}{A(z)} = \frac{\mathrm{b}(1) + \mathrm{b}(2)\, z^{-1} + \cdots + \mathrm{b}(\mathrm{n}{+}1)\, z^{-n}}{\mathrm{a}(1) + \mathrm{a}(2)\, z^{-1} + \cdots + \mathrm{a}(\mathrm{n}{+}1)\, z^{-n}}.$$

Then, with MATLAB's freqz function, I get frequency response of the filter to plot with plot function of MATLAB.

Lastly, I applied the filter function with parameters function coefficients of the filter and sampled data of the original audio. I applied the filter twice so that it separates better.

As it could be seen from the graph of the frequency, Bandpass filter separates sounds with frequencies in the given interval and attenuates sounds with other frequencies. So, in the output piano.wav file, we mostly hear piano chords sound.

3. Highpass FIR Filter



To separate the sound of the cymbals, we need a highpass filter. So, I used MATLAB's butter filter which takes order, normalized cutoff frequency and filter type and returns the function coefficients. Butter filter's formula is as given:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\mathrm{b}(1) + \mathrm{b}(2)\, z^{-1} + \cdots + \mathrm{b}(\mathrm{n}{+}1)\, z^{-n}}{\mathrm{a}(1) + \mathrm{a}(2)\, z^{-1} + \cdots + \mathrm{a}(\mathrm{n}{+}1)\, z^{-n}}.$$

Then, with MATLAB's freqz function, I get frequency response of the filter to plot with plot function of MATLAB.

Lastly, I applied the filter function with parameters function coefficients of the filter and sampled data of the original audio. I applied the filter twice so that it separates better.

As it could be seen from the graph of the frequency, Highpass filter separates sounds with higher frequencies and attenuates sounds with lower frequencies. So, in the output cymbal.wav file, we mostly hear piano chords sound.

Lastly, waveform plot of the audio.wav file is as given: