Filtering noise in a Audio file

Problem description:- You are given a music track that has been corrupted by the presence of an unwanted instrumental beats, disrupting the intended harmony. Your task is to analyze the frequency characteristics of the input, then design and implement an appropriate filtering technique using Python to suppress the interfering component. The goal is to recover the original song as faithfully as possible and export the cleaned version. Use various analysis tools like Bode graphs,power spectral density, pole-zero plots of your system design and isolate frequency regions, and design filters accordingly to remove unwanted solo instrumental music.

Solution approach:- To filter noise from an audio file, I first converted the audio signal into the frequency domain. This transformation was performed using the Fourier Transform, which revealed the frequency components present in the signal. To make the spectral analysis more intuitive, I applied a shift to center the zero frequency in the transformed data.

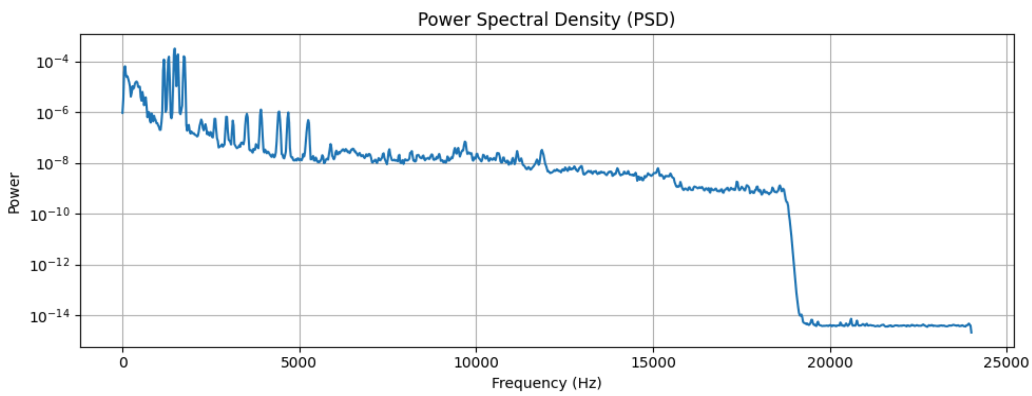
Once the audio was in the frequency domain, I computed and visualized both the Power Spectral Density (PSD) and the spectrogram. The PSD helped me identify dominant frequency spikes that likely corresponded to the interfering instrumental sounds. I also used an interactive spectrogram to correlate these frequency spikes with specific time segments in the track, providing a clearer understanding of when and where the noise occurred.

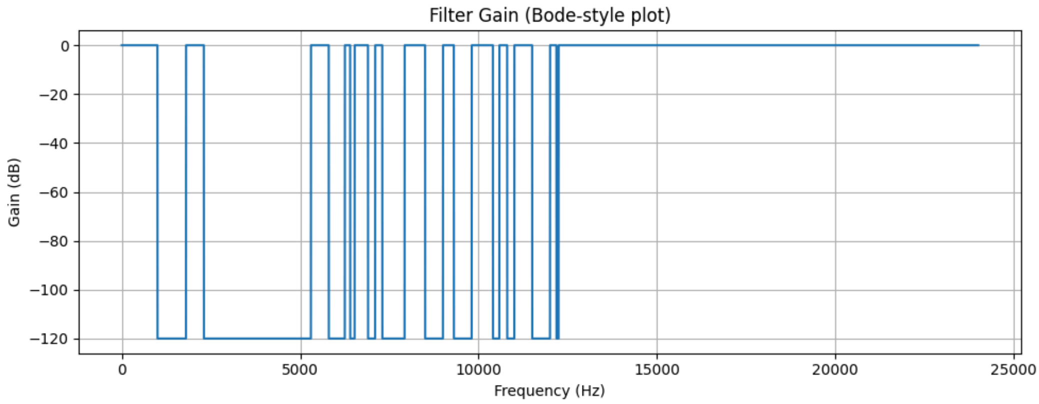
After analyzing the PSD and spectrogram, I isolated the exact frequency bands responsible for the noise. I then designed a custom band-stop filter to attenuate only those specific ranges, while preserving the rest of the musical content. This filter was implemented in the frequency domain using a binary gain mask. Following the filtering process, I applied the inverse Fourier Transform to reconstruct the audio signal in the time domain.

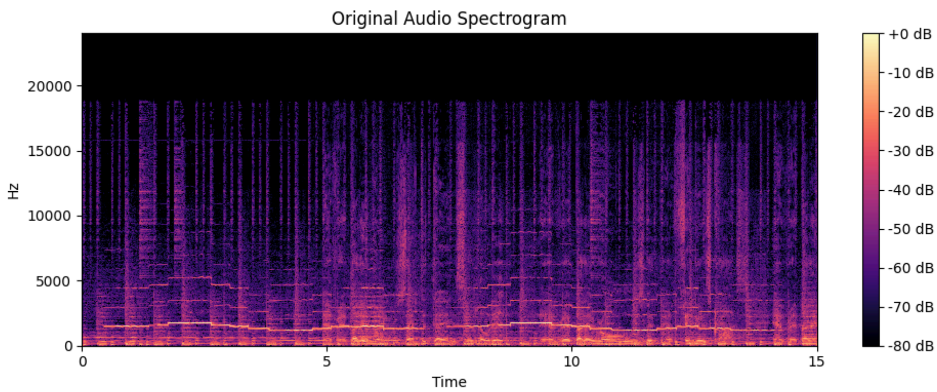
Finally, I normalized and exported the resulting audio as a cleaned version of the original track. To validate the effectiveness of my filter design, I generated Bode plots which helped me understand the behavior of the filtering system. This approach allowed me to significantly reduce the intrusive elements while retaining the integrity of the intended musical content.

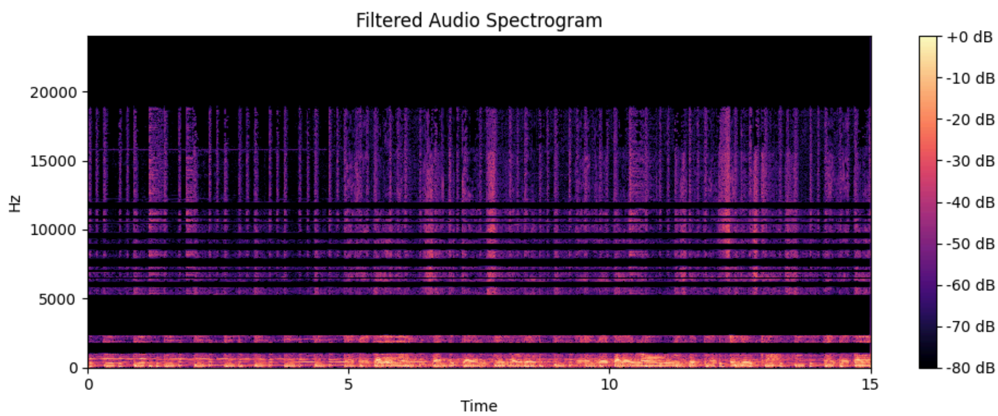
The transfer function of the filter
:- H(f) = 1 if(f is not in the noise frequency range) else 0.

Design choices:- I applied spectral subtraction in the frequency domain using the Fast Fourier Transform (FFT) to suppress an unwanted solo instrument. To estimate the noise profile, I used a segment of the audio that contained only the target instrument (specifically, the first one second). I then transformed the full audio using FFT and computed its magnitude spectrum. Next, I subtracted the estimated noise spectrum from the full spectrum on a bin-by-bin basis, which helped reduce the energy contribution of the unwanted source. To maintain physical validity, I clamped any negative values to zero. Finally, I reconstructed the cleaned signal by combining the modified magnitudes with the original phase information and applied the inverse FFT. This method provided a more targeted and adaptive approach compared to traditional band filtering, as it specifically suppressed the spectral signature of the noise without affecting the broader frequency content.


Power Spectral Density (PSD)
Power
Frequency (Hz)


Filter Gain (Bode-style plot)
Gain (dB)
Frequency (Hz)


Original Audio Spectrogram
Hz
Time


Filtered Audio Spectrogram
Hz
Time