REPORT ON Frequency de-mixer: 'Unwanted Solo'

Bhavit

July 2025

1 Introduction

In modern music production and audio engineering, removing unwanted instrumental sounds while preserving the clarity and quality of the intended recording is crucial. This experiment focuses on removing the unwanted instrumental frequencies from a provided music track while maintaining the overall quality of the song.

2 Objective

The objective of this experiment is to analyze the frequency characteristics of a music track corrupted by unwanted frequencies and to design and implement appropriate filtering techniques to suppress these interfering components. The goal is to maximize the audio's clarity while preserving its quality using systematic frequency-domain analysis tools including Bode plots, pole-zero plots, power spectral density (PSD), and spectrogram visualizations. The implementation ensures that specific interfering frequencies are attenuated while retaining the intended musical elements, resulting in a cleaned and faithful restoration of the original song for analysis and listening.

3 Theory

An audio signal is composed of a mixture of various frequencies that together form music, speech, and background sounds. To remove unwanted noises we used frequency analysis techniques to identify these interfering components and applied carefully designed filters to remove them while preserving the rest of the audio.

3.1 Fourier Transform (FT)

The Fourier Transform helps convert an audio signal from the time domain to the frequency domain. This allows us to see which frequencies are present in

the audio, making it easier to identify unwanted tones or noise for removal.

3.2 Short-Time Fourier Transform (STFT)

The STFT extends the Fourier Transform by showing how the frequency content of a signal changes over time. It is used to create spectrograms that visually represent which frequencies are active at different moments in the audio.

3.3 Power Spectral Density (PSD)

PSD analysis shows how the power of a signal is distributed across different frequencies. Using Welch's method, we can identify which frequency bands carry the most energy and locate the unwanted interfering frequencies precisely.

3.4 Spectrogram

A spectrogram is a visual tool that displays the intensity of different frequencies over time. It helps us see where unwanted frequencies like piccolo tones appear in the audio, guiding us in selecting which frequencies to filter out.

3.5 Butterworth Filters

Butterworth filters are used to remove specific frequencies from the audio smoothly without significantly affecting nearby frequencies. In this experiment, we used band-stop Butterworth filters to remove piccolo frequencies.

3.6 Bode Plots

Bode plots show how a filter affects different frequencies by displaying its magnitude and phase response. This helps us understand how much attenuation is applied to unwanted frequencies and ensures that the filter behaves as intended.

3.7 Pole-Zero Plots

Pole-zero plots visualize the locations of poles and zeros of a filter, helping us check the filter's stability and how it shapes the frequency response. A stable filter ensures effective frequency removal without introducing artifacts into the audio.

3.8 Filter Selection

Filter selection involves deciding which type of filter, cutoff frequencies, and filter order to use based on the analysis of the audio. By carefully selecting these parameters, we can effectively remove unwanted frequencies while preserving the desired quality of the music.

3.9 Welch's Method

Welch's Method is a technique used to estimate the Power Spectral Density (PSD) of a signal with reduced variance compared to a single periodogram estimate. It involves dividing the signal into overlapping segments, applying a window function to each segment, computing the periodogram for each, and then averaging these periodograms.

This method helps in identifying the frequency components present in a signal with greater stability and clarity. In this practical, Welch's Method was used to analyze the audio track and accurately locate unwanted piccolo frequencies that required removal using filtering techniques.

4 Implementation

4.1 Audio Loading and Inspection

The provided music track was loaded using Python libraries, and its waveform was inspected to understand its structure and check for visible abnormalities in the time domain.[FIGURE.1]

4.2 Signal Specifications

The input audio signal used for analysis had the following properties:

- **Sample Rate:** 48,000 Hz
- **Duration:** 15.0 seconds
- **Number of Samples:** 720,001 samples

These parameters define the frequency analysis resolution, Nyquist frequency, and time context used throughout the practical.

4.3 Frequency Analysis

Power Spectral Density (PSD) analysis and spectrogram visualizations were used to identify the frequency components of the audio and locate the unwanted piccolo frequencies requiring removal.[FIGURE.7]

4.4 Filter Design

Butterworth band-stop filters was used to target and remove the identified piccolo frequencies precisely without affecting the rest of the audio. We targeted multiple frequencies (e.g., 1200 Hz, 1500 Hz, 3100 Hz, etc.) with different bandwiths (e.g., 500 Hz, 250 Hz, 100 Hz), selected manually to remove unwanted frequencies.

4.5 Verification Using Bode and Pole-Zero Plots

Bode plots were generated to visualize the magnitude and phase response of the filters, ensuring effective attenuation at the target frequencies[FIGURE.2]. Pole-zero plots were used to verify the stability and effectiveness of the filters before application[FIGURE.3].

4.6 Filtering and Evaluation

The designed filters were applied to the audio, and the filtered output was evaluated using spectrograms and subjective listening tests to confirm that the unwanted frequencies were effectively removed while maintaining the overall quality and clarity of the track.[FIGURE.4]

5 Analyzation

This section explains in detail the concepts, methods, filter design, and library tools used systematically during the practical.

5.1 Approach to Frequency De-Mixing

Audio signals often contain a mix of desired musical content and unwanted noise or interference (such as piccolo frequencies in this experiment). The core idea of frequency de-mixing is to identify these interfering components and attenuate them using frequency-domain analysis and filtering while preserving the clarity and quality of the desired audio.

5.2 Methods Used and Their Purpose

- **Fourier Transform (FT):** Used to convert the time-domain audio into the frequency domain, enabling us to see the distribution of frequency components.
- **Power Spectral Density (PSD):** Provided insights into the power distribution across frequencies, helping to identify the prominent unwanted frequency bands that needed filtering.
- **Short-Time Fourier Transform (STFT) and Spectrogram:** Allowed us to visualize how frequencies varied over time, helping us precisely locate and monitor interfering piccolo frequencies.
- **Frequency Response Comparison:** By plotting the frequency response before and after filtering, we could verify the effectiveness of our filtering process.

5.3 Libraries Used and Their Roles

- **NumPy:** Used for efficient numerical operations, including FFT computations and array manipulations.
- **SciPy:** Provided advanced signal processing capabilities, including filter design (Butterworth filters), filter application (`sosfiltfilt`), and frequency response computation (`sosfreqz`).
- **Librosa:** Utilized for audio file loading, STFT computation, and spectrogram visualization, providing tools to analyze audio signals effectively.
- **Matplotlib:** Used to visualize waveforms, PSD, spectrograms, frequency responses, Bode plots, and pole-zero plots, aiding in the interpretation of our processing steps.

5.4 Filter Design and Implementation in Detail

The **Butterworth band-stop filter** is a type of signal processing filter designed to attenuate a specific band of frequencies while allowing frequencies outside this band to pass with minimal distortion.

The Butterworth filter defined by:

$$|H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}$$

where n is the filter order and ω_c is the cutoff frequency.

For digital filter design, we normalized the cutoff frequencies using:

$$\omega_n = \frac{f_c}{f_s/2}$$

where f_c is the cutoff frequency and f_s is the sampling frequency.

To implement this in our practical, we used the `scipy.signal` Python library, specifically the `butter` function to design the band-stop Butterworth filters and the `sosfiltfilt` function to apply zero-phase filtering, ensuring there was no phase distortion in the filtered audio.

The Butterworth filter was chosen for its flat frequency response in the passband and a reasonably sharp roll-off, which ensured that:

- Unwanted frequencies were attenuated effectively.
- No additional ripples were introduced in the passband.
- The musical integrity of the track was maintained.

We designed the filters using the `butter` function in `scipy.signal`, specifying:

- **Order:** Chosen based on the desired sharpness of attenuation.

- **Cutoff frequencies:** Centered around the identified unwanted frequencies with an appropriate bandwidth for effective removal.
- **Filter type:** `bandstop` for narrow-band removal and `lowpass` for managing high-frequency noise.

The filters were applied using `sosfiltfilt` to achieve zero-phase filtering, ensuring no phase distortion in the audio signal.

5.5 Verification Using Plots

- **Bode Plots:** Generated to visualize the magnitude and phase response of the designed filters to ensure accurate attenuation at target frequencies.
- **Pole-Zero Plots:** Used to verify filter stability and effectiveness, ensuring poles remained inside the unit circle to maintain stability.
- **Frequency Response Comparison:** Plotted before and after filtering to confirm that only the unwanted frequency bands were attenuated while the rest of the audio spectrum remained unaffected.[FIGURE **??**]

5.6 Step-Wise Application and Monitoring

1. Loaded and inspected the audio to understand its structure.
2. Analyzed the frequency characteristics using PSD and spectrograms.
3. Designed band-stop filters for identified piccolo frequencies.
4. Verified the filtering process using Bode plots and pole-zero plots.
5. Reviewed filtered audio using spectrograms and subjective listening to ensure clarity and fidelity were preserved.

This structured and methodical approach allowed us to effectively remove unwanted piccolo frequencies while maintaining the clarity, integrity, and quality of the music track.

6 Observations

During the experiment, the following observations were made systematically:

- **Time-Domain Inspection:** The original audio waveform displayed a mix of high and low amplitude regions, indicating a rich frequency composition. No clipping or major distortions were visible initially. [FIGURE.1]
- **PSD Analysis:** Using Welch's Method, strong frequency bands corresponding to unwanted frequencies were identified in the range of approximately 1000–5000 Hz, guiding the precise selection of filter bands.[FIGURE.7]

- **Spectrogram Analysis:** Spectrograms of the original audio revealed bright horizontal bands indicating persistent piccolo frequencies[FIGURE.4]. Post-filtering spectrograms confirmed the attenuation of these bands while retaining the overall structure of the music.[FIGURE.5]
- **Bode and Pole-Zero Plots:** Bode plots verified that the designed filters provided effective attenuation at the target frequencies with minimal impact on the surrounding bands[FIGURE.2]. Pole-zero plots confirmed filter stability throughout the process.[FIGURE.3]
- **Frequency Response Comparison:** Frequency response plots before and after filtering showed clear dips at the filtered frequencies while maintaining the energy distribution in non-targeted frequency bands, confirming effective selective filtering.[FIGURE **??**]
- **Spectrograms:** spectrograms shows unwanted frequencies approximate the 0–5000 Hz range provided clearer insight into piccolo frequency attenuation, validating the effectiveness of the applied filtering process.[FIGURE.4]
- **Subjective Listening:** After filtering, the audio was audibly cleaner, with noticeable reduction in the sharp piccolo tones while preserving the integrity, brightness, and clarity of the music.

Figures:

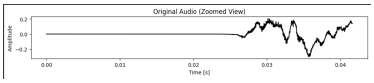

Figure 1: Time-domain waveform of the original audio

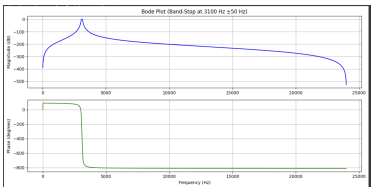

Figure 2: Bode plot of the band-stop filter designed at 3100 Hz

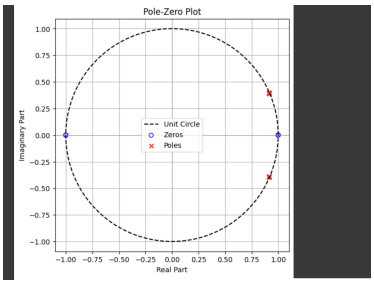

Figure 3: Pole-zero plot with unit circle for the band-stop filter at 3100 Hz.

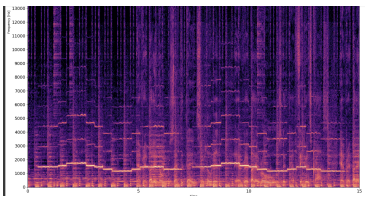

Figure 4: Spectrogram of the original audio showing persistent piccolo frequencies.

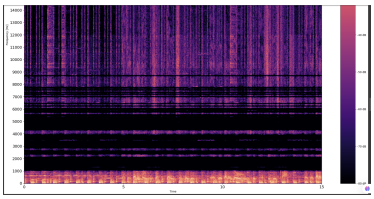

Figure 5: Spectrogram after filtering showing effective attenuation of piccolo frequencies.

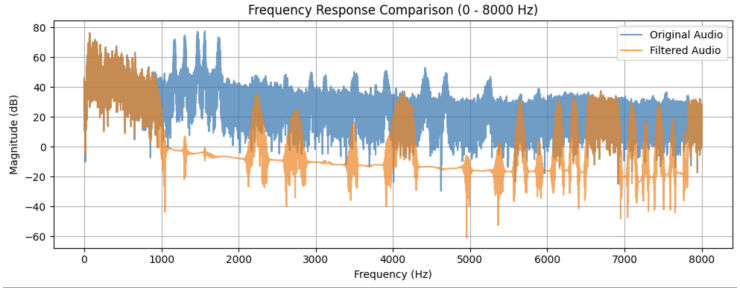

Figure 6: Frequency response comparison before and after filtering.

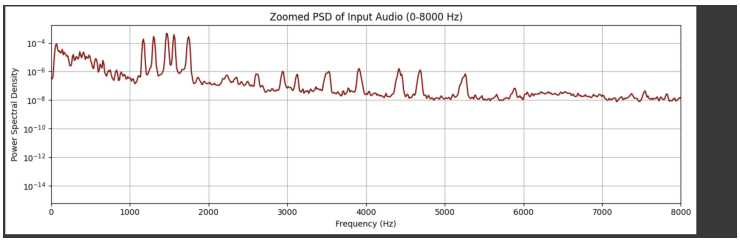

Figure 7: Zoomed Power Spectral Density (PSD) showing filtered frequency bands.

7 WHAT IF:

I MENTIONED ONLY BAND FILTER IN MY NOTEBOOK , THIS ARE ALTERNATIVE METHODS THAT WERE USED ,WHAT IF :

7.1 1. Low-Pass Filter (Above 1000 Hz)

what if we had applied a low-pass filter with a cutoff frequency around 1000 Hz to remove most high-frequency components, including the piccolo tones. This would effectively attenuate unwanted frequencies above the cutoff. However, it would also remove higher harmonics, instrumental details, and the brightness of the music, resulting in a dull and muffled sound. While effective in suppressing high-frequency noise, it would significantly compromise the quality and fidelity of the music, making it unsuitable for scenarios where preserving musical richness is essential.

7.2 2. High-Order Band-Stop Filters

what if we used higher-order band-stop filters, such as 6th or 8th-order Butterworth or Chebyshev filters, instead of the 4th-order filters we used. Higher-order filters provide sharper cutoffs around the targeted frequencies, allowing for more precise removal of unwanted tones while preserving nearby frequencies. However, this comes with the trade-off of potential ringing artifacts and increased phase distortion, particularly near the cutoff regions, if not carefully designed. The implementation complexity also increases, requiring careful stability and frequency response checks using Bode and pole-zero plots.

7.3 3. FIR Filters (Finite Impulse Response)

what if we used Finite Impulse Response (FIR) filters could also be considered as an alternative for filtering out unwanted frequencies. FIR filters are advantageous because they exhibit a linear phase response, ensuring that the phase relationships in the audio signal are preserved, which is beneficial for maintaining audio quality. However, to achieve sharp transitions comparable to IIR

filters, FIR filters require a much higher order, leading to increased computational cost and longer processing times. For practical real-time filtering, this may not be efficient, although it provides a clean and stable filtering option for offline processing.

These alternative methods highlight the various possibilities for frequency de-mixing and interference removal. However, for this experiment, using multi-frequency band-stop Butterworth filters provided an optimal balance between effective frequency removal and preservation of the audio quality, aligning well with the goals and practical constraints of the task.

8 Conclusion

In this experiment, we successfully applied systematic frequency de-mixing techniques to remove indentified unwanted piccolo frequencies from a music track while preserving overall quality. Using tools such as Power Spectral Density analysis, spectrograms, and frequency response comparisons, we precisely identified and verified the interfering frequency bands.

By designing and applying multi-band band-stop Butterworth filters with carefully chosen parameters we removed the frequencies, Additional verification through Bode and pole-zero plots confirmed the accuracy and stability of our filters.

Subjective listening tests and post-filtering spectrograms confirmed the improvement in audio clarity while maintaining musical integrity.