Multiband Compression

Khandaker Hossain, Jephtey Adolphe, Mazen Abdalla

What is Multiband Compression?

Multiband compression is a type of audio compression that rather than compressing an entire audio signal, it divides the signal into multiple bands based on frequency ranges. This allows each individual band to have their own compression settings with any modifications to those settings remaining independent of the other bands' settings. This gives more control over how the different sounds and frequency ranges in the signal will be compressed.

Motive

When listening to raw, unedited audio oftentimes there will be unbalanced sounds such as instruments or voices in it. Sometimes balancing those sounds in an audio signal doesn’t require the modification of the signal in its entirety, rather altering a specific frequency range would suffice. Multiband compression provides an extremely useful tool for handling those situations which come up very often in many different scenarios.

Summary of Technical Work

Multiple lines of code were required to get the ideal results. We started by concocting codes, a different algorithm for the lowpass, bandpass and high pass respectively.

After these codes were completed, we focused on constructing codes for treble, bass and mid functions.

Finally they had to be implemented with a GUI function

Accomplishments

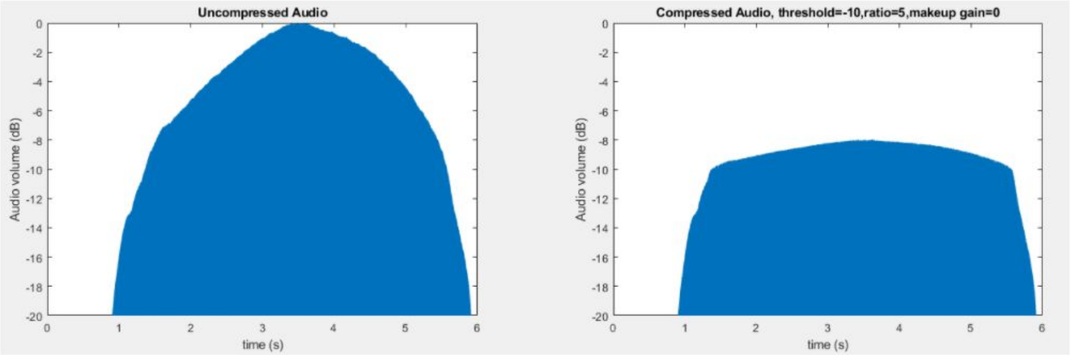


Figure 5: Compression being applied to a 440 Hz sine wave of increasing amplitude.

The top row of figure 6 below shows two frequency ranges, or bands, of the sine wave of figure 5. The bottom row demonstrates the modification of an individual band while leaving the other band untouched.

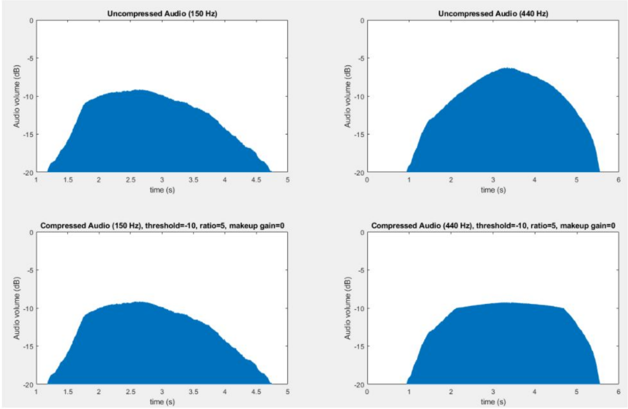
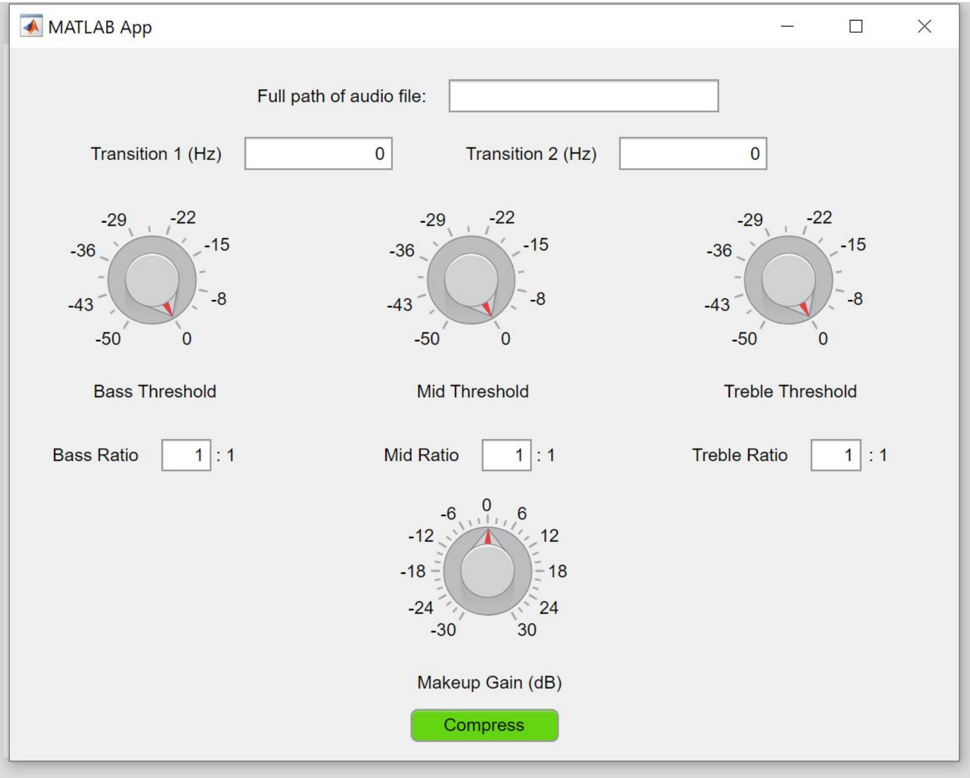


Figure 6: The top two plots are the 150 Hz component and 440 Hz components of the input signal (displayed on separate plots for clarity), and the bottom two plots are those same frequency components after going through the dynamic compressor (settings: transition 1 = 250 Hz, transition 2 = 1000 Hz, all thresholds = -10 dB, all ratios = 5, makeup gain = 0). The louder 440 Hz component was reduced in volume, making the 150 Hz signal easier to hear.

In figure 5 we see the result of passing a pure sine wave, which is on the left, at 440 Hz into the compressor. In this case we used a threshold of -10 dB and a ratio of 5. The plot on the right hand side of figure 5 shows the compressor successfully turning on at -10 dB, decreasing the amount that the signal rises by.

Accomplishments cont'd



Graphical user interface for our multiband compressor which contains the parameters required to modify an audio file.

Division of Labor

Jephtey - Designed the graphical user interface of the multiband compressor with the required parameters such as the individual band ratios and thresholds, makeup gain, and transition frequencies and connected it to our main code. Helped with proposal, report, and slides.

Khandaker - Designed the codes for the low pass, band pass, and high pass filters with windows. These codes were the prerequisite for the treble, bass and mid functions. Also contributed to the proposal, code and slides.

Mazen - Designed the main dynamic compression function that is integrated in the GUI code. This code handles the filtering, processing, and exporting of the compressed audio. Contributed to the proposal, report, video demonstration, and README file.

Lessons Learned

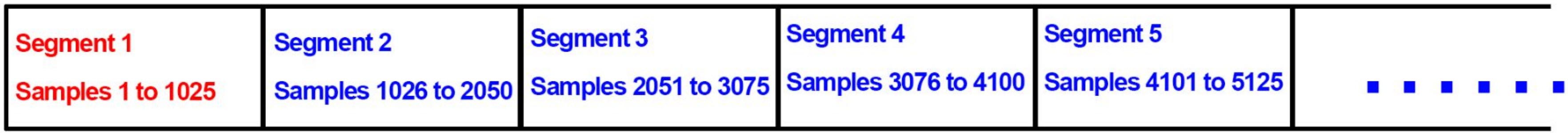
- We learned the basics of the different types of audio compression and learned the importance of them in different applications.
- An actual application of how LPF can cut higher frequencies whilst HPF cut lower frequencies.

Challenges

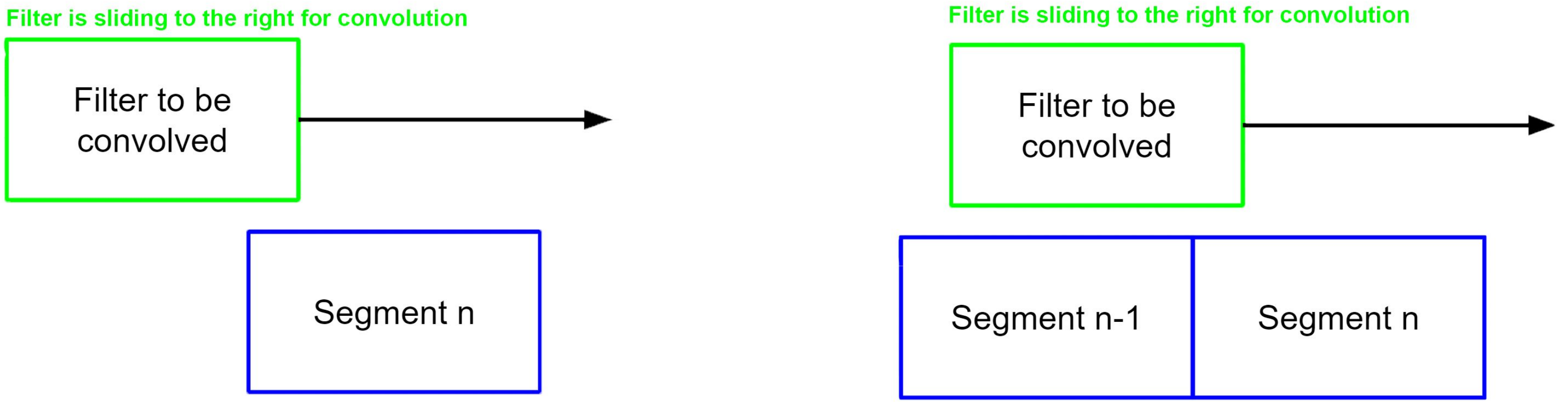
- Trying to split and modify the bands of an audio signal in real-time was difficult whether it was editing it as the audio file was playing or of it was providing sound through a microphone.
- Developing a thorough understanding of of the intricacies of audio compression in general in a short time period.

Code Challenges

- The sequence needed to be split into smaller pieces



- We needed to convolve multiple segments to reduce transients



Video Demo

We also have a video demonstration of our program running

Follow Up Questions

- What measures could be taken to further refine a multiband compressor?
- Are there any scenarios where using multiband compression is discouraged?
- Is parallel compression the better alternative due to multiband compression resulting in unwanted noise and ringing

Conclusion

Based on the results obtained from our implementation, our compressor archives dynamic compression by passing the input through a low pass, band pass, and high pass filter, analyzing a small segment of the audio signal, and applying a calculated amount of attenuation to each segment. The GUI makes a smooth connection with the interface without requiring extra code.