B EE 507 A

Final Project: Vocoder Design and Implementation

Gwen Montague and Olivia Killingsworth

June 10, 2022

Abstract:

In this final project we implement an N channel voice vocoder using band-pass filtering via a For Loop in MATLAB. After testing multiple audio signals and N-values, we found the ideal N value depends on the purpose of the vocoder. Additionally, we found that vocoders are not likely to be useful for filtering musical signals.

Frequency Responses of all Bandpass and Low-Pass Filters

Audio signal 011 used

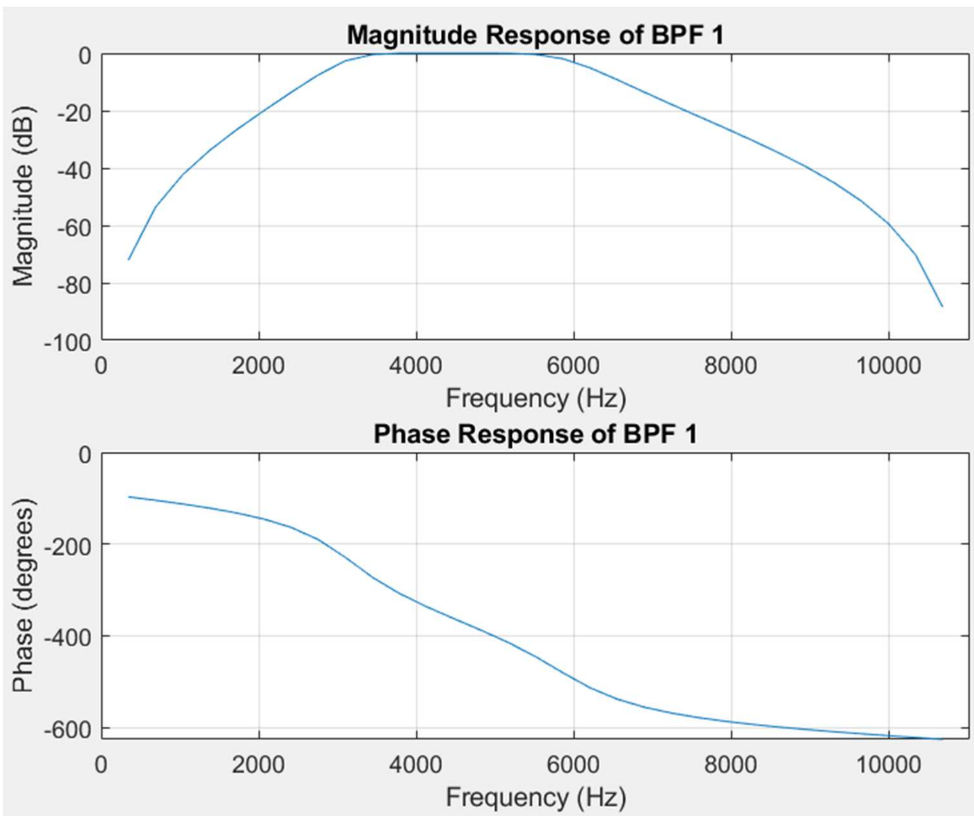

Figure 1.1: Frequency Response of initial 1st BPF (name: "BPFilter_signal")

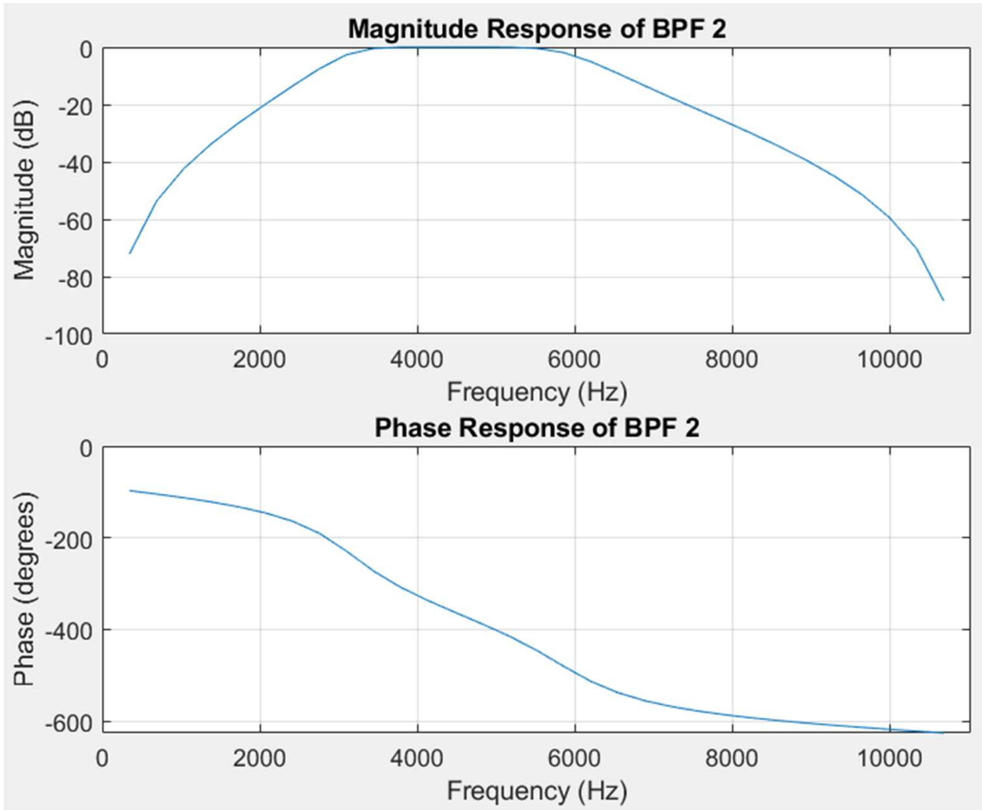

Figure 1.2: Frequency Response of 2nd BPF applied after envelope (name: "BPFcs")

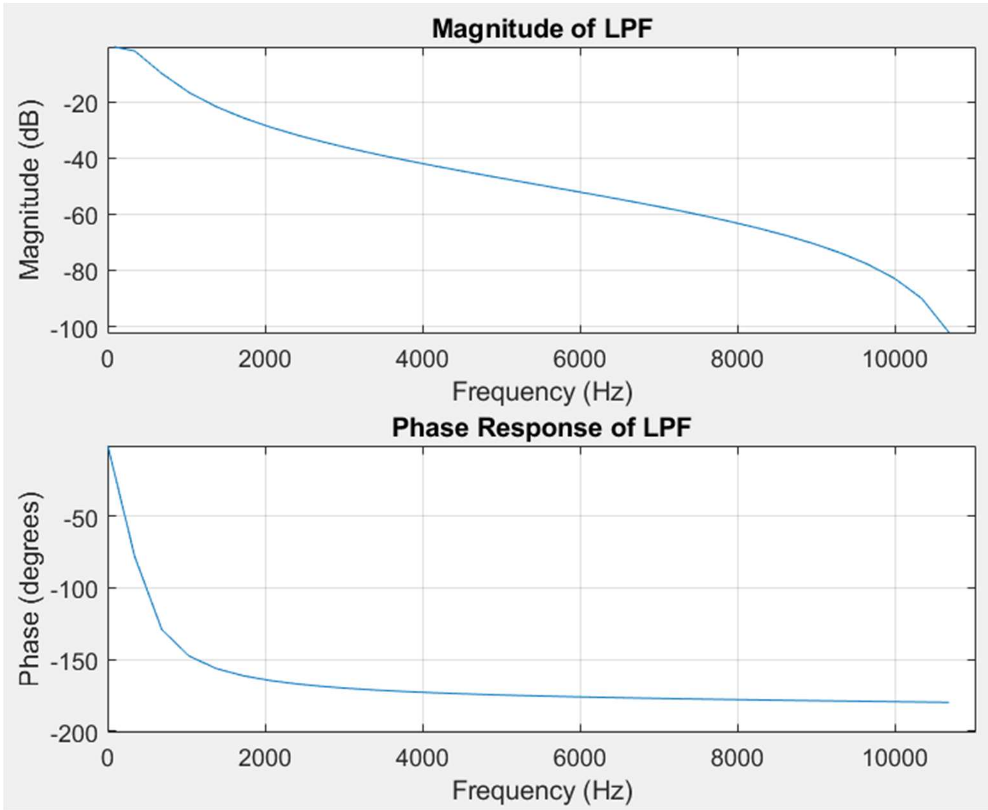

Figure 1.2: Frequency Response of LPF (name: "LPFilter_signal")

Envelopes at N =1 through N = 4

Audio signal 011 used

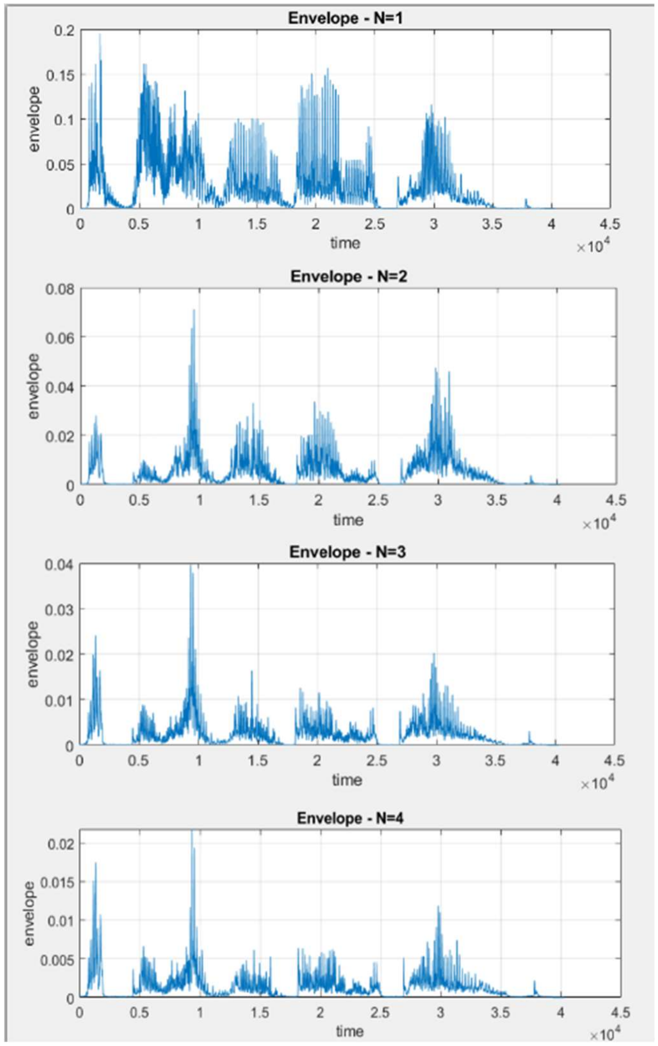

Figure 2.1: Envelopes at N=1, N=2, N=3, & N=4

Modulated Signals

Audio signal 011 used

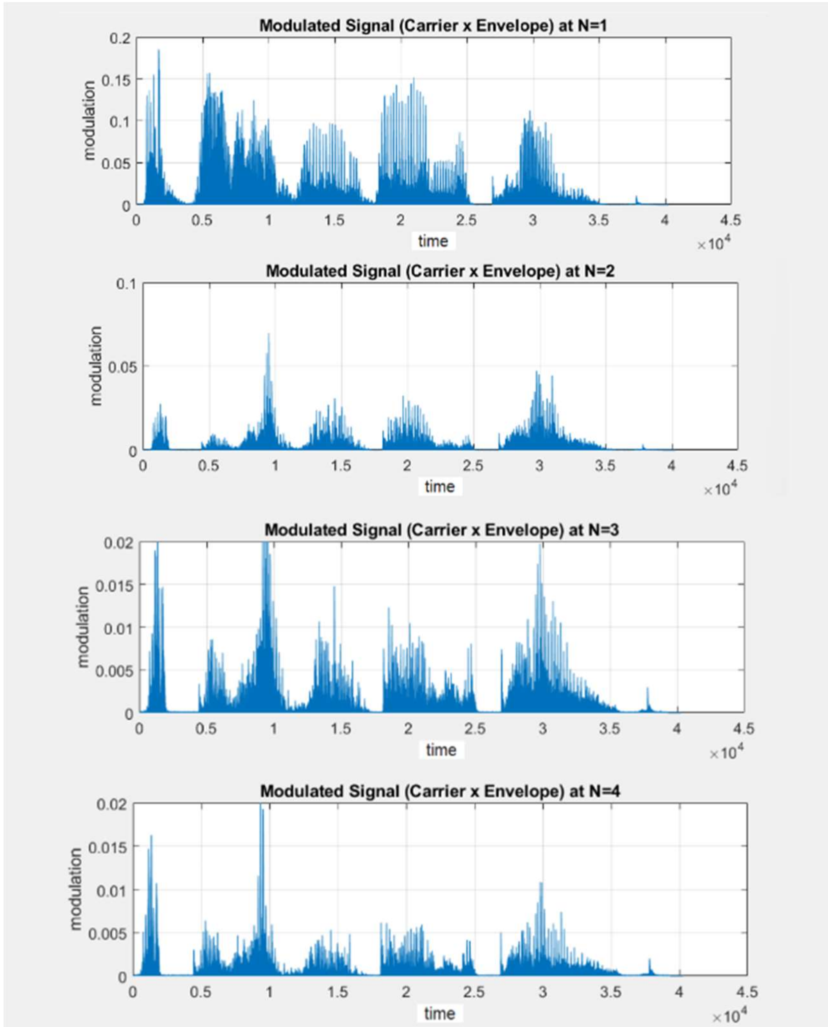

Figure 3.1: Modulated Signal (Carrier x Envelope) at N=1, 2, 3, & 4

Original vs. Vocoded Sound

Audio signal 011 used

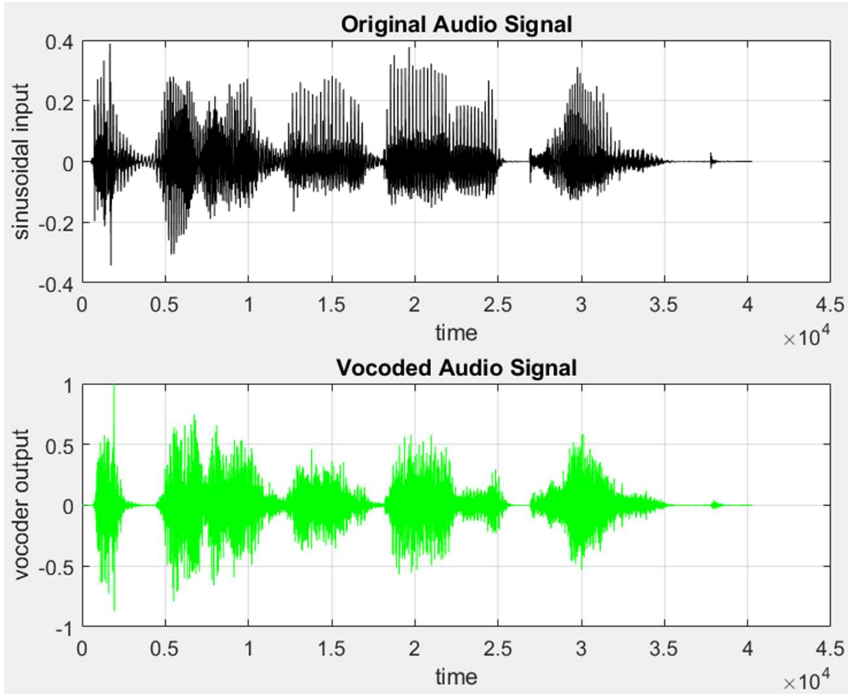

Figure 4.1: Original Audio v Vocoded Audio at N=8

Spectrogram

Audio signal 011 used

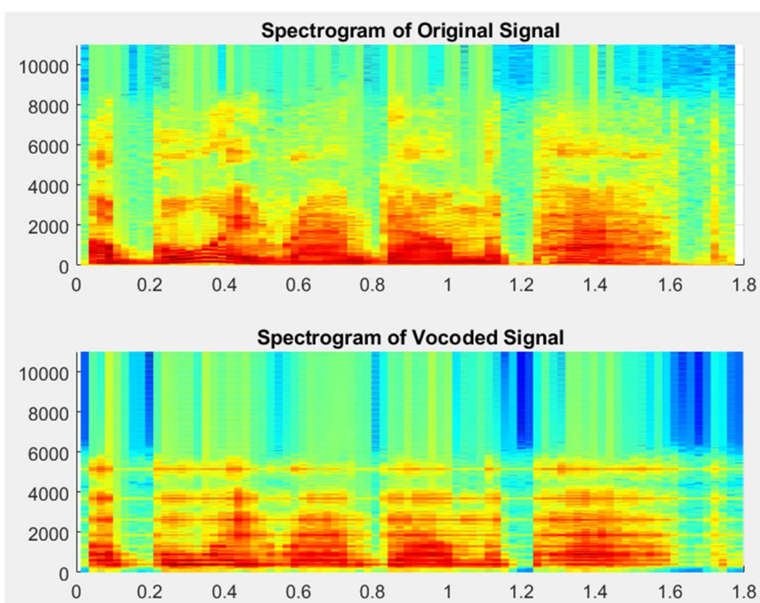

Figure 5.1: Spectrogram of 011 at N=8

Musical Signals: Original vs. Vocoded

Classical piano audio file used

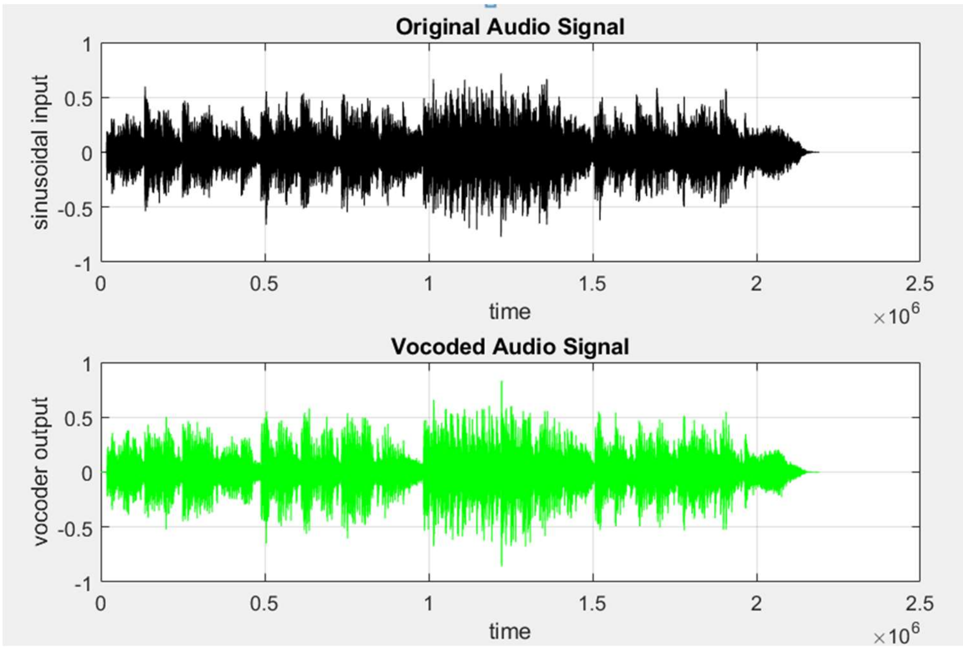

Figure 6.1: Classical Piano at N=20

Discussion

For our final project we implemented an N channel voice vocoder on multiple audio signals. To do this we used a For Loop to implement a band-pass filter, an envelope, modulation, and then band-limiting for N filters. After implementing the code to perform this filtering (see separate script files), we graphically and auditorily compared the original audio signal to the vocoded audio signal.

The original sound was warm and full of varying inflections, while the vocoded sound was robotic and more monotone sounding. By filtering out frequencies below 300 and above 6000 Hz and dividing the sound into N bands, the inflection and warmth of the original audio was eliminated.

We also uploaded various audio files and adjusted the N value on each of them. For speech files that were approximately 5 to 6 words in length, we discovered that for N<6 the sound was not consistently decipherable to people. While the designers (Gwen and Olivia) could understand the sound around N=4, people who were not familiar with the sound files or people with poor hearing could not understand the sounds at N<6.

At N=8, 10, and 12 people could easily understand vocoded sentences that were approximately 5 to 6 words long. At N=20, the vocoded sound sounded similar to the original sound to the point where someone might be able to recognize the voice that is speaking. However, by N>80 the vocoded signal was understandable, yet distorted. At these higher N-values the vocoded sound was more electronic than classically robotic.

We also compared 2 musical signals. One was an electronic music sound clip, and the other was a classical piano sound clip. We determined the vocoder distorts musical signals so much that the music is completely unrecognizable. The classical piano piece was so distorted it no longer sounded like a piano at all – even at 'better' N values of 8 to 20. This is due to the pitch at which classical piano music needs to be played and auditorily received at.

In summary, we determined that speech signals are best understood at N>6 but N<80, and sound most like the original around N=20. We also determined that the distortion is different at extremely low versus extremely high N-values. Therefore, the ideal N depends on the purpose of the vocoder. For example, if someone is trying to disguise their voice N=20 would be the worst value, and N=7 or 8 would be much better. Lastly, we determined classical music is painful to listen to when filtered through a vocoder.