CS591 Final Project

May 11th

JINRAN TIAN

WENYU WANG

Instructor: Wayne Snyder

Final Project Report

We did the third topic list on the sheet for our final project.

Digital Implementation of Guitar Pedals. Guitar pedals provide a bewildering variety of effects (as filters) for modifying the basic timbre of the electric guitar. For this project, I would like someone to pick an interesting set of effects (e.g., frequency modulation, flanging, distortion, reverb, etc.) and investigate how to implement these with a Python program.

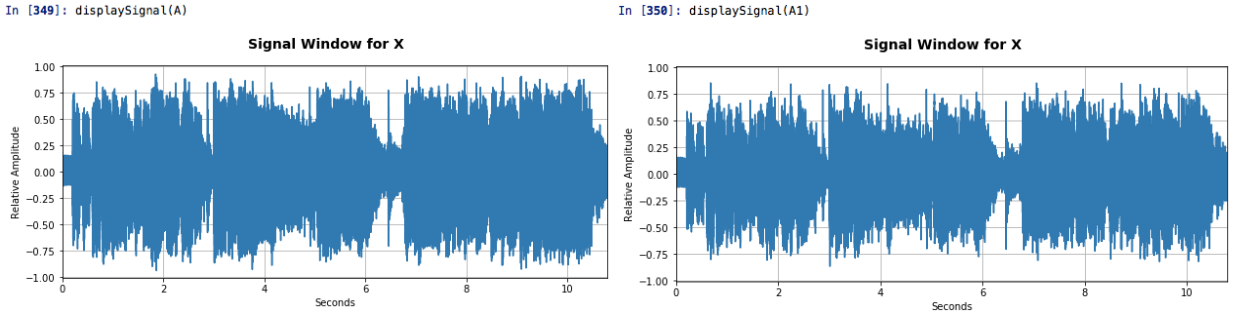
To begin, we did some research on the topic. As we learned for the first time, a guitar pedal is a sound modification tool that support electric guitar that adds special effect to the sound it produces. This means we are to use the knowledge learned from classes, mainly digital signal processing to perform this. An example from the class material will be the ASR envelope that we simply apply to the signals to make changes. In this project, we developed more technics to make different effects to the signals.

Distortion

We learned that audio distortion refers to any kind of deformation of an output waveform compared to its input, usually clipping, harmonic distortion, or intermodulation distortion caused by non-linear behavior of electronic components and power supply limitations. And before we tried to implemented it. We listened to many distortion samples and found that most distorted sound have a little echo in the background and most rock and jazz music has this property according to our knowledge. So we first weaken all the peaks in the signal by slightly

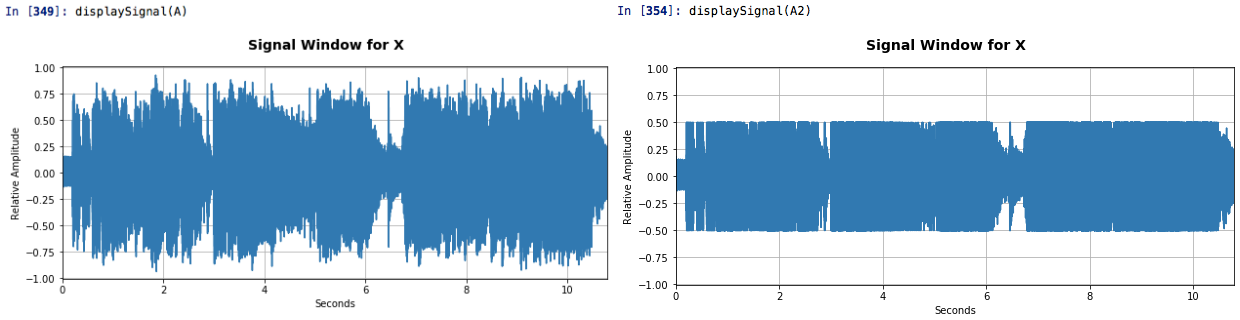
"chopping" off the peaks. We used "BluesGuitar.wav" and get "BGsimpleDistortion.wav" and the following graph comparison.

(before) (after)



The difference is not obvious in such scale unless you scan along the 0.75 and - 0.75 for the amplitude. Although the peaks are weakened, the sound is not satisfying since it's not how distortion sounds like and has noise in the back. Then we used clipping method which clips all amplitudes that's off a certain threshold. This time we get "BGclipDistortion.wav".

(before) (after)

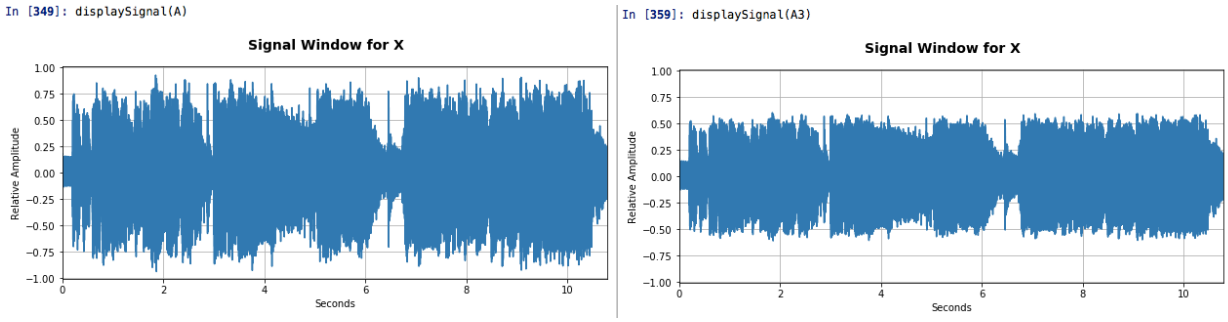


There's no noise this time, but the distortion effect is still not great. At last we found a better method for distortion. As we wrote in the previous report. The DAFX book by Udo Zölzer about the distortion effect introduced the following function:

$$f(x) = \frac{x}{|x|}\left(1 - e^{-x^2/|x|}\right)$$

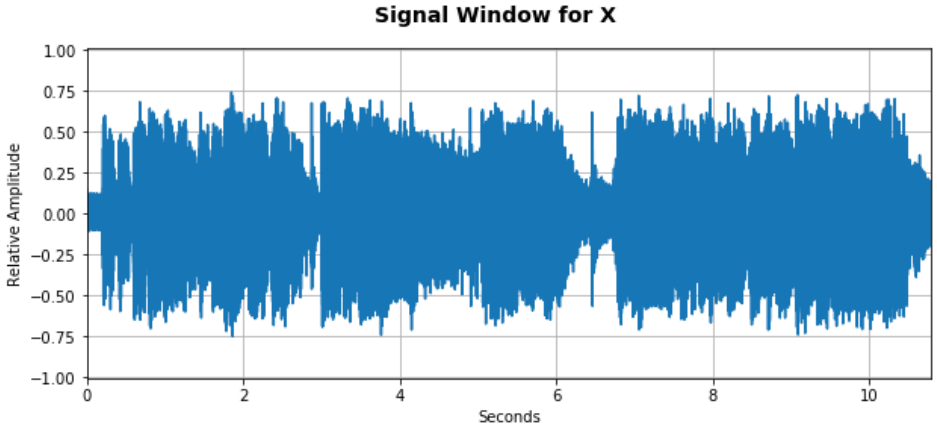
This time we get "BGdistortion.wav".

(before) (after)



Chorus

The second effect we work on is chorus. Chorus means make the voice free and natural by repeating. This effect will make a voice with just one source sounds like coming from several different places, which could make the voice more expressive. The key is “repeat”. Our basic idea is to copy the original file, then make it a little bit weak by decreasing its amplitude, and let it follow behind the original piece. As we already knew that in one file we can just show one signal, which means one piece of voice, performing two different signal in one file is not allowed. So here we get the idea to add a new thread, and let it run the new piece of voice. Also, we apply time.sleep() function to let the thread with new voice start a little bit late than the original voice, for example 0.2 second, which can achieve our aim of “repeat”.



The output signal of the chorus code of “BluseGuitar_chorus.wav”