Exam, Spring 2022

DSP Lab – ECE-GY 6183

1. Sign and submit the attached 'oath' along with your solutions.
2. You may use the course resources (lecture videos, demo programs, textbook, etc) and your own prior work for this course.
3. You may ask me (Ivan Selesnick) if you have questions about the exam. But you may not discuss the exam with anyone else. You may not get help from anyone. You may not ask anyone for assistance. Your submitted work must be 100% your own effort.
4. You may consult the web for information (e.g., python documentation) but not for how to solve the problems below.
5. A violation may lead to failing the course. If in doubt, ask Ivan Selesnick by email at selesi@nyu.edu

For each question, ensure that your submitted program files run on their own (without additional files needed) so that we can run your file. Also, write a short explanation for how you approached each question.

Question 1: Chorus effect with GUI

Using Python and TKinter, implement in real-time the multivoice chorus effect with a graphical user interface (GUI). Use two voices, as shown in Fig. 2.17 on page 53 of the text book *Audio Effects: Theory, Implementation and Application*. The relevant pages are provided with this exam. Your GUI should have sliders for the user to vary the gain, LFO frequency, and LFO sweep depth of the two voices. Your program should use a circular buffer and interpolation. The user should be able to hear the effect on the signal as the sliders are moved with negligible latency. Audible artifacts when sliders are moved should be avoided. The GUI should run until a 'quit' button is pressed. The output audio signal should be saved to a wave file. Your program should read the wave file `audio_Q1.wav`. Submit your program files and a wave file of the output signal. Provide a short video (screen recording) demonstrating your program.

Question 2: Tonal noise suppression

Using Python, implement a real-time filter to remove additive tonal noise from a signal where the frequency of the tonal noise varies with time. The input signal should be read and processed one block at a time. For each block, the tonal noise can be identified as the largest peak of the Fourier transform. Audible artifacts should be avoided as much as possible. Your program should save the output signal to a wave file. Run your program on audio signal in the file `audio_Q2.wav`. Submit the wave file of the output signal.

In addition, create your own example of a test audio signal with time-varying tonal noise. Run your program on your own example as well as on the provided audio signal. Submit your input and output wave files.

Submit your program files and a short video (screen recording) demonstrating your program.