EECS 3602 Lab 2 : Signals and Sound Waves

Submission details: Write your responses to the following questions and submit them electronically as a lab report, along with any code that you write. If your responses are handwritten, scan them for electronic submission. Submission is via Moodle. Due date: October 15, 2015.

Grading details: 70% of your lab grade is for correctly completing the lab requirements; 20% is for clear writing and good presentation, including readable and well-documented code; 10% is for extra work or analysis that expands on or goes beyond the lab requirements.

Introduction. For this lab, you will need a computer with a working sound system, as well as stereo headphones. These instructions are written for MATLAB, but related packages exist for Python; you may use Python if you wish.

MATLAB includes functions that allow you to read and write sound files in various formats. A simple such format is WAV, in which audio is stored as pulse code modulation (PCM) data. In this format, the amplitude of each sample is encoded in binary and stored in the WAV file; thus, sample amplitudes can be directly read from and written to a WAV file.

In this lab, we will use MATLAB's `wavwrite()` command to write WAV files. Using `wavwrite(z,Fs,N,filename)`, you can specify four parameters:

- `z`: A vector of sample data. If z contains two columns, then the resulting WAV is stereo, with the first column on the left channel and the second column on the right channel; if z contains one column, then the resulting WAV is mono.
- `Fs`: The sampling frequency in hertz.
- `N`: The number of bits per sample: 8, 16, or 24.
- `filename`: A string containing the file name for the WAV file (e.g., 'foo.wav')

Part 1: Basic sound generation. The keys of a piano generate musical notes at given frequencies. For a list of these frequencies, see here:
`https://en.wikipedia.org/wiki/Piano_key_frequencies`

1. Write a function that generates a mono sine wave at the frequency of "middle C" (key number 40), lasting 3 seconds, with a sampling frequency of 8000 Hz and using 16 bits per sample. The WAV file should be saved in: `middleC.wav`

2. Write a function that generates a mono sine wave over the C major scale (key numbers: 40, 42, 44, 45, 47, 49, 51, 52), each note lasting 0.5 seconds, with a sampling frequency of 8000 Hz and using 16 bits per sample. The WAV file should be saved in: `Cscale.wav`

3. For the report: Explain how you wrote the above functions, and discuss the sounds generated. What do you observe about the notes/sounds from keys 40 and 52 and their frequencies? Can you spot similar patterns for other keys?

Part 2: Harmonics.

1. For the report: The documentation for `wavwrite()` states: *"Note that for floating point data ... amplitude values are clipped to the range* $-1.0 <= Y < 1.0$*"*. What does this mean? Generate a "middle C" signal (like in Part 1) that exceeds these limits. What does it sound like compared to a signal that does not exceed them? Draw a diagram (or generate a plot in MATLAB) to illustrate what is happening.

2. The *harmonics* of a note occur at integer multiples of the base frequency. That is, if a note is generated with frequency f (as in Part 1), then the harmonics would have frequencies ℓf for integers $\ell > 1$. Write a function that takes, as argument, a base frequency f and a vector of amplitudes $\vec{a} = [a_1, a_2, \ldots, a_n]$, and writes a WAV file with signal $x(t) = \sum_{\ell=1}^{n} a_\ell \sin(2\pi \ell f t)$. The WAV file should be saved in: `harmonic.wav`

3. For the report: Try your function with base frequency at middle C, and various amplitudes. What is the difference in the sound? Does it sound like middle C, or a different note? What seems to govern the note that you hear?

4. For the report: How is this related to the Fourier series? Can you generalize the function you wrote in question 2 to an arbitrary periodic function?

Part 3. Stereo.

1. If you provide `wavwrite()` with a two-column signal, it generates a stereo WAV file: the first column plays on the left channel, and the second column plays on the right channel. Write a function that plays middle C for 3 seconds as a stereo signal, fading from left to right (i.e., the middle C note starts all on the left side, with silence on the right; smoothly fading from left to right so that after 1.5 seconds the two sides are equal, and after three seconds it is all on the right side, with silence on the left.)