Wave File Exercises (related to demo 02)

DSP Lab (ECE 4163 / ECE 6183)

Fall 2018

1 wav files

We will use the Python wave module a lot in this course.

https://docs.python.org/3/library/wave.html

Read and experiment with the wave module and how to use it to work with wave (.wav) files.

To read basic information from the header of a wave file, we can use functions from the wave module as follows. We can open the wav file.

```
wf = wave.open( 'cat01.wav', 'rb')
```

Then we can use the commands

```
# read the properties of the wav file
num_channel = wf.getnchannels()         # number of channels
fs = wf.getframerate()                  # sampling rate
length_signal = wf.getnframes()         # signal length
width = wf.getsampwidth()               # byte per frame
```

to read

1. the number of channels,
2. the sampling rate (frames per second),
3. signal length and width (the many bytes per sample).

2 Python pack function

The available data formats are listed in the Python documentation under Format Characters.

https://docs.python.org/3/library/struct.html

Experiment with data formats 'B', 'h' and 'i'. Do you see how numbers are stored in a binary file for each format?

3 Assignments

These exercises are related to the demo files for Lecture 1.

Do all parts. Submit only the three indicated exercises.

1. Record a wav file of your own voice with one channel (mono) with a sampling rate of 16 kHz and 16-bits per sample. (You may use `Audacity` or some other audio software.)

2. Write a Python script using the `wav` module to read and print basic information about your `wav` file. See the demo file `read_wavefile_02.py`. Verify that the provided information matches the intended properties of the wave file. For your 16-bit wav file, what is the value of `width` returned by `getsampwidth()`? Submit your recorded `wav` file, Python code, and written comments. SUBMIT

3. Record `wav` files of your voice with identical settings, but use 8-bit and 32-bit formats. For these files, what values are returned by `getsampwidth()`?

4. The program `write_sin_02.py` generates a wave file with 32 bits per sample. Use MATLAB to read `sin_02_mono.wav` and determine the quantization size. What is the quantization size? How many quantization levels are there?

5. Use Python to generate a `wav` file of a sine wave at 8 bits per sample. Read your 8 bit/sample `wav` file into MATLAB and plot the signal to verify that it is a sine wave (zoom in if necessary to show the waveform). Verify that the quantization step size is as expected and verify its spectrum. Is there any noticeable effect of a lower number of bits/sample on the sound quality (keeping the same number of samples/second)? Submit your Python code, `wav` file, and written comments. SUBMIT

6. Use Python to generate sinusoids of lower frequency, like 50 Hz and 25 Hz. Listen to the `wav` file.

7. Use higher sampling rates, like 16K, 32K, and 44.1K samples/second.

8. In `write_sin_01.py`, can you set the number of channels to be more than 2? Use Python to generate a `wav` file with more than two channels, with different waveforms for each channel. Read the `wav` file into MATLAB and plot the individual channels (zoom in if necessary to show the waveforms). Submit your Python code, MATLAB code, comments, and MATLAB plot saved as a pdf file. SUBMIT

4 Submission instructions

Assignments should be submitted via the "Assignments" section of the course NYU Classes page. The submitted materials for each exercise should include the code and written comments for the exercise. Include all files needed to run and verify your work. Your written comments should be provided as pdf files or text files. The system does not allow you to modify your submission after you upload it. Therefore, before uploading your work, please verify that each submission includes the complete materials and that your programs run.

5 Note

To save an 8 bit wave file using Audacity:

1. Use 'export audio' in 'File' option
2. Save type as 'other uncompressed files'
3. Select WAV(Microsoft) as Header
4. Select 'Signed 32-bit PCM' as Encoding
5. There are 4 PCM options in 'Encoding'. They are 'Signed 16-bit PCM', 'Signed 24-bit PCM', 'Signed 32-bit PCM', and 'Unsigned 8-bit PCM'.