Lab 2: Wave Files and Pack

DSP Lab (EE 4163 / EL 6183)

Last Edit: Thursday 15th September, 2016 19:49

1 `wav` files

We will use the wave module a lot in this course:

- `wave` module: https://docs.python.org/2/library/wave.html

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

To read basic information from the header of a wave file using the `wave` module, we can use the following functions.

After opening the `wav` file using

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

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

from the file header.

2 Python pack function

The available data formats are listed in the Python documentation: Section 7.3.2.2. Format Characters
https://docs.python.org/2/library/struct.html

Experiment with data formats: 'B', 'h' and 'i'.

Do you see how the numbers are stored in the binary file for each format?

3 Assignments

These exercises are related 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 `read_wav_example_01.py` of the lecture 1 demo files.)
   Verify this 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 and Python code. * * *

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

4. The program `make_sin02.py` generates a wave file with 32 bits per sample. Use MATLAB to read `sin02_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.
   Verify that the quantization step size is as expected and verify its spectrum.
   Is there any noticeable effect of lower number of bits/second ?

   * * * Submit your Python code and `wav` file and written comments. * * *

6. Use Python to generate a sinusoid of lower frequency, like 50 Hz. Listen to the `wav` file.

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

8. In `make_sin01.py`, what happens if 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 this `wav` file into MATLAB and plot the individual channels (zoom in if necessary to show the waveforms).

   * * * Submit your Python and MATLAB code and MATLAB plot saved as a pdf file. * * *