Lab 4: Filtering a wave file

DSP Lab (EE 4163 / EL 6183)

Last Edit: Thursday 29th September, 2016 17:49

1 Play a wave file using PyAudio

The demo python program `play_wav_mono.py` shows how to read and play a wave file using PyAudio. It is assumed the wave file is mono (single channel). The program `play_wav_stereo` plays a stereo (two channel) wave file.

Note that, in this example, we use a `while` loop to write a `string` to the `stream`, and inside the loop, we use `unpack()` and `pack()` methods to translate between a binary string and a list of integers. Note that in the `while` loop, we use a gain variable to amplify the signal and the function `clip16()` to keep the value within the range of a signed 16-bit integer to avoid possible overflow run-time errors. Correspondingly the letter `h` (`hh` for stereo) are used in the `unpack()` and `pack()` methods to set the encoding format.

Documentation for the `wave` module is at:

https://docs.python.org/2/library/wave.html

1.1 Exercises

1. **Single program for mono and stereo.** Write a single Python program for playing both mono and stereo wave files. The program should determine the number of channels by reading the wave file information.

   (a) Verify that your program can play both mono and stereo wave files encoded with 16-bits per sample.

   (b) Modify your program in (a) so that it can be used at the command line like SUBMIT

   ```
   >> python mywave_play.py filename.wav
   ```

   For this, you will need to import the `sys` module. For example, if the file `demo_sys.py` is:

   ```python
   # demo_sys.py
   import sys
   print 'Argument 0 is', sys.argv[0]
   if len(sys.argv) > 1:
       print 'Argument 1 is', sys.argv[1]
   ```

   then at the terminal command line we get:

   ```
   >> python demo_sys.py
   ```

```
Argument 0 is demo_sys.py

>> python demo_sys.py aaa.wav
Argument 0 is demo_sys.py
Argument 1 is aaa.wav
```

2. Write a Python program that can play wave files with 8, 16, or 32 bits per sample.

   (a) Note that `h` is used to set the encoding scheme to signed 16-bit integer in `unpack()` and `pack()` methods. What letters should be used for (1) signed 32-bit integer, and (2) unsigned 8-bit integer?

   (b) What are the ranges of signed 32-bit integer and unsigned 8-bit integer? Write your own clipping functions to avoid run-time overflow errors while playing wave files of signed 32-bit integer and unsigned 8-bit integer.

   (c) Modify the example Python program so that it can play a mono wave files encoded using any of: unsigned 8-bit integer, signed 16-bit integer, or signed 32-bit integer. The PyAudio function `pyaudio.get_format_from_width` maybe useful here. SUBMIT

2 Filtering a wave file

Two demo programs are available. See the Matlab demo program `make_and_test_filter.m` and the Python program `filter_play_wav_mono.py`.

(a) Modify the python program to write the audio output to a wave file. Then verify that the wave file is the same as the wave file produced by the Matlab program (e.g., read both wave files into Matlab and plot the error between them).

(b) Modify python program so that it plays a stereo: The left channel should play the original signal. The right channel should play the output of the bandpass filter. SUBMIT

(c) Instead of a bandpass filter, use a bandstop filter. Modify the Matlab program so that it designs a bandstop filter and applies it to the speech signal. Then modify the Python program so that it implements the same bandstop filter. SUBMIT

(d) In the Python program, the fourth-order difference equation is implemented in 'direct form'. Another implementation is called the 'canonical form' which requires fewer delay elements. This form is illustrated in the block diagram in Fig. 7.2.4 on page 274 of the text book 'Introduction to Signal Processing' by Orfanidis SUBMIT

   http://www.ece.rutgers.edu/~orfanidi/intro2sp/orfanidis-i2sp.pdf

   The software implementation of the canonical form is shown in Equation 7.2.5 on the same page.

   Modify the Python program so that it implements the difference equation using the canonical form. Instead of 8 'delays', this form should have just 4 'delays'. Verify that the output produced by this implementation of the difference equation is the same as the direct form implementation.