Demo 6 Exercises: Filter a wave file

DSP Lab (ECE 4163 / ECE 6183)

2019

Demo files

`wave_filter_matlab.m`
`wave_filter_python.py`
`author.wav`

In the Matlab demo program, we read a whole audio signal from a wave file, and then we filter it with a band-pass filter. In the Python demo program, we read an audio signal from a wave file one signal value at a time, we filter the audio signal as we read it, and we write it to the output audio stream one signal value at a time.

Exercises

1. Modify the Python demo 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 demo program. (To verify they are the same, you can read both wave files into Matlab and plot the error between them).

2. Modify the Python demo program to play a stereo output signal from the mono-channel wave file `author.wav`. The left channel should be the original signal. The right channel should be the output of the bandpass filter.

3. Instead of a bandpass filter, use a bandstop filter. Modify the Matlab demo program to design a bandstop filter and apply it to the speech signal. Then modify the Python demo program to implement the same bandstop filter.

4. The Python demo program implements the fourth-order difference equation with 8 variables to store past values (i.e., 8 delay units). This is the the *direct form* implementation. But a fourth-order difference equation can be implemented using just 4 variables to store past values (i.e., 4 delay units). The *canonical form* can be used for this purpose. See 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 demo program to implement the difference equation using the canonical form. Instead of 8 delay variables (`y1, y2, y3, y4, x1, x2, x3, x4`) your new program should have just 4 delay variables. Verify the output produced by this implementation is the same as the output produced by the demo program.