Demo 14 Exercises: Block filtering

DSP Lab (ECE 4163 / ECE 6183)

2019

Demo files

`demo_filter.py`
`demo_filter_blocks.py`
`demo_filter_blocks_corrected.py`
`demo_filter_blocks_mtlb.m`
`myfunctions.py`
`author.wav`

In previous demos we used the Matlab function `filter` to implement a difference equation. In Python, a similar function called `lfilter` is available in the SciPy library for scientific computing. (Here `lfilter` means *linear* filter.)

`http://docs.scipy.org/doc/scipy/reference/signal.html`

To avoid transient artifacts at the start of each block, we specify the initial states `zi` in the `lfilter` function as the final states `zf` from the previous block.

Exercises

1. The demo programs take the input audio signal from a wave file and apply a bandpass filter. In this exercise, modify the demo program `demo_filter_blocks_corrected.py` to take the input audio signal from the microphone.

2. Like the previous exercise. Also plot the input and output signals in real-time in a figure window (use different colors for the input and output signals). SUBMIT

3. The Matlab function `butter` gives the coefficients of a digital Butterworth filter. For example, a band-pass filter with a pass-band from 500 Hz to 1000 Hz can be obtained in Matlab using:

   ```
   [b, a] = butter(2, [500 1000]*2/Fs)
   ```

   What is the order of this filter?

   In Python, there is also a function `butter` in the SciPy library `scipy.signal`. Verify that the Python function gives the same coefficients as the Matlab function.

4. **Filtering.** Write a Python program that applies a bandpass filter to the microphone input signal. The bandpass filter should have a passband from 500 Hz to 1000 Hz. The program should plot the live frequency spectra (Fourier transform) of both the input and output signals (use two different colors). The program should also play the output to the speaker/headphones. How is the spectrum of the output signal related to the spectrum of the input signal?