Demo 13: Blocking and filtering
Exercises

DSP Lab (EE 4163 / EL 6183)

Fall 2017

1 Demo files

`filter_wav_file.py`
`filter_wav_file_blocking.py`
`filter_wav_file_blocking_fixed.py`
`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.

2 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 `filter_wav_file_blocking_fixed.py` to take the input audio signal from the microphone.

2. Same as the previous exercises, but also plot the input and output signals in real-time in a figure window (use two different colors for the two 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.