CSE 3313 – Homework #5

Assigned: Wednesday, February 27, 2019
Due: Wednesday, March 6, 2019 at the end of class

Assignment:

1. (10 points) Find the impulse response of the system that can replace the series of systems with impulse responses $h_1[n] = \{1, 2, 3\}$ and $h_2[n] = \{-1, 0, 4, -2\}$.

2. (10 points) If the two systems from the previous problem had been in parallel, what single system could replace them? Give the impulse response of this system.

3. (10 points) Find `xcorr(A, B)` where

$$A = \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & 2 & 3 \\ -1 & -2 & -3 \end{bmatrix}$$

   Remember that in this case matrix A is the fixed matrix and matrix B passes over it.

4. (35 points) Find the DFT or inverse DFT as directed. Answers should be in Cartesian coordinates.

   (a) Find the DFT of `x[n]` $= \{1, -1, 1, -1\}$.

   (b) Find the DFT of `x[n]` $= \{1, -1, 1\}$. *Ask yourself why this problem and the previous problem have different answers. Also think about constructing the signal in the previous problem from the frequencies you found and what the consequences are of trying to identify the frequencies present without using all of the previously generated values.*

   (c) Find the inverse DFT of `X[k]` $= \{6, 0, -2, 0\}$.

   (d) Find the DFT of `x[n]` $= \{2, 0, -2, 0\}$.

   (e) Find the DFT of `x[n]` $= \{2, 0, -2\}$. *Ask yourself why this problem and the previous problem have different answers.*

   (f) Find the inverse DFT of `X[k]` $= \{2, -1 - j, 8, -1 + j\}$.

5. (35 points) **Application Area: Noise Removal with lowpass FIR Filters**
   Purpose: Learn how to apply a lowpass filter to music to remove noise with high frequencies. Also learn how to apply a window to filter coefficients to change the frequency response of the filter.

   (a) Your source code will be called `lowpassNoiseRemoval.py`.

   (b) On Blackboard is a file, `P_9_2.wav`, that contains a piece of music that has been corrupted with noise. This file is from [SB96].

   (c) To remove the noise, you will use the lowpass filter from hw04 with these specifications:

      - a cut-off frequency of $f_c = 7500$ Hz

- the filter length $L = 101$ and $M =$ filter length - 1.

which is based on the tutorial at [Gre15].

(d) To make this slightly more complex than the last time, instead of using a rectangular window (the default if you do nothing), you will apply a Hamming window as described in the same tutorial. Be careful that you apply the Hamming window weights to the low-pass filter coefficients; don't try to just use the Hamming window weights as the filter coefficients. **Do not use NumPy's built-in functions that create the filter coefficients or the Hamming window .**

(e) After producing the L Hamming window values, apply them to the filter coefficients by performing an *element-wise* multiplication between $h[n]$ and $w[n]$.

(f) We also want to look at the frequency response of our filter before and after applying the window. To do this, use

```
x, y = freqz(filter_coefficients, 1)
```

after importing the library with

```
from scipy.signal import freqz
```

The y values may be complex, so to plot them you will need to use `plot(x, abs(y) )`. **Put both frequency response plots on the same figure** to make it easier to see the effect. The two plots should overlay each other, similar to Figure 1.

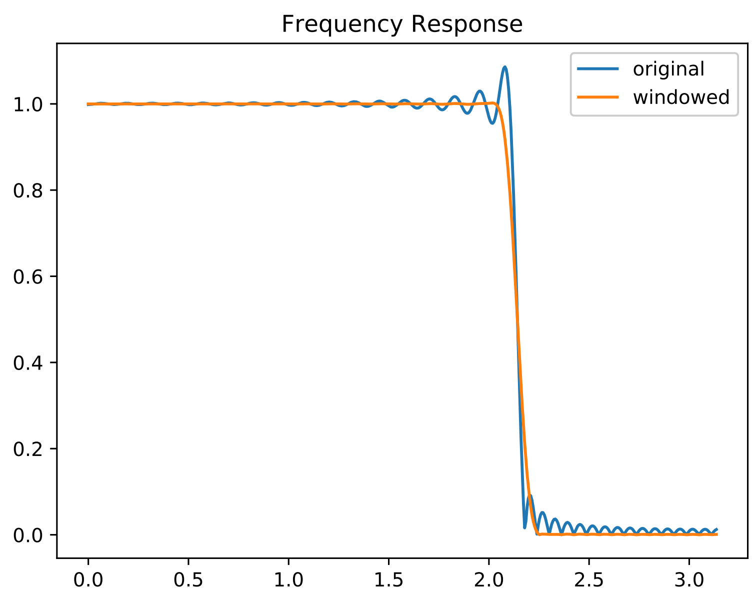

Figure 1: Frequency response with and without windowing

(g) Here is the overall process:

i. read the signal, i.e., the music, from the file
ii. produce the filter coefficients, $h[n]$
iii. plot the frequency response of $h[n]$
iv. produce the Hamming window coefficients, $w[n]$

v. apply the window to the filter coefficients to produce $\widehat{h}[n]$

vi. plot the frequency response of $\widehat{h}[n]$

vii. convolve the signal with the new filter coefficients, $\widehat{h}[n]$

viii. save the output as `cleanMusic.wav`

(h) Don't hard-code your logic to this particular data.

General requirements about the Python problems:

a) As a comment in your source code, include your name.

b) The Python program should do the work. Don't perform the calculations and then hard-code the values in the code or look at the data and hard-code to this data unless instructed to do so.

c) The program should not prompt the user for values, read from files unless instructed to do so, or print things not specified to be printed in the requirements.

To submit the Python portion, do the following:

a) Create a directory using your net ID in lowercase characters plus the specific homework. For example, if your net ID is `abc1234` and the homework is `hw04`, then the directory should be named `abc1234-hw04`.

b) Place your .py files in this directory.

c) Zip the directory, not just the files within the directory. You must use the zip format and the name of the file (using the example above) will be `abc1234-hw04.zip`.

d) Upload the zip'd file to Blackboard.

References

[Gre15] Andrew Greensted. FIR Filters by Windowing. http://www.labbookpages.co.uk/audio/firWindowing.html, accessed March 15, 2015.

[SB96] Virginia Stonick and Kevin Bradley. *Labs for Signals and Systems Using MATLAB*. PWS Publishing Company, 1996.