CSE 3313 – Homework #3

Assigned: Wednesday, February 6, 2019
Due: Wednesday, February 13, 2019 at the end of class

Note the following about the homework:

1. You must show your work to receive credit.
2. For the hand-written problems, submit a hard-copy.
3. For the problems requiring Python, upload your source code to Blackboard. See instructions at the end of the document.

Assignment: Work these by hand

1. For each of the following systems, prove algebraically if it is linear or nonlinear.

   (a) `y[n] = log( x[n] )`

   (b) `y[n] = x[n] + 4x[n-1]`

   (c) `y[n] = (x[n])`^{2}

   (d) `y[n] = |x[n]|`

   Be sure to state your final conclusion.

2. Evaluate the moving average filter

$$y[n] = \frac{1}{L}\sum_{k=0}^{L-1} x[n-k]$$

   for the *unit-step* input signal

$$u[n] = \begin{cases} 0 & \text{for } n < 0 \\ 1 & \text{for } n \geq 0 \end{cases}$$

   (a) Plot $u[n]$ as a stem plot before finding $y[n]$.

   (b) Find the numerical values of $y[n]$ over the range $-5 \leq n \leq 10$ with $L = 5$.

   (c) Plot $y[n]$ as a stem plot for the values that you just found.

3. An LTI system can be described by the difference equation

$$y[n] = 2x[n] - 3x[n-1] + x[n-2]$$

   (a) Represent the difference equation as a block diagram in direct form.

   (b) Represent the difference equation as a block diagram in transposed form.

Applications

4. **Application Area: Communications**
Purpose: Learn how to process digitized data and to use a matched filter to identify the correct value in the presense of noise.

In digital communications, a bit can be represented by a pulse of a specific duration. In theory, a receiver that knows what a 0 bit and a 1 bit looks like can read the message it received by matching each portion of the signal with the appropriate pulse. The problem in practice is that the signal can become corrupted by noise, making identification of the bits difficult. One way to look at a portion of a corrupted signal to determine the pulse that it likely represents is with a **matched filter**.

Imagine that we have two vectors $\vec{v}$ and $\vec{w}$. The angle θ between real vectors $\vec{v}$ and $\vec{w}$ can be found using

$$\cos\theta = \frac{\vec{v}\cdot\vec{w}}{||\vec{v}||_2||\vec{w}||_2}$$

When the vectors are orthogonal, $\cos\theta = 0$, and when the vectors are collinear, $|\cos\theta| = 1$. This relationship can be used to produce the **Cauchy-Schwarz inequality**:

$$0 \leq \left|\frac{\vec{v}\cdot\vec{w}}{||\vec{v}||_2||\vec{w}||_2}\right| \leq 1$$

This tells us that the dot product of two vectors measures the similarity of the two vectors. The more similar the vectors, the closer the dot product gets to the product of the 2-norms of each vector. For more information, see [Bar14]. For an example of how matched filters can be used to produce a photomosaic, see
`http://www.clear.rice.edu/elec301/Projects02/photoMosaic/301.htm`.

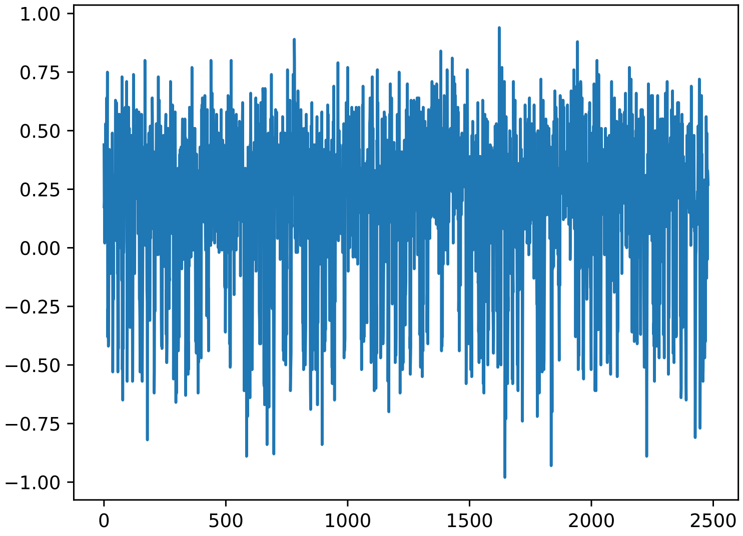
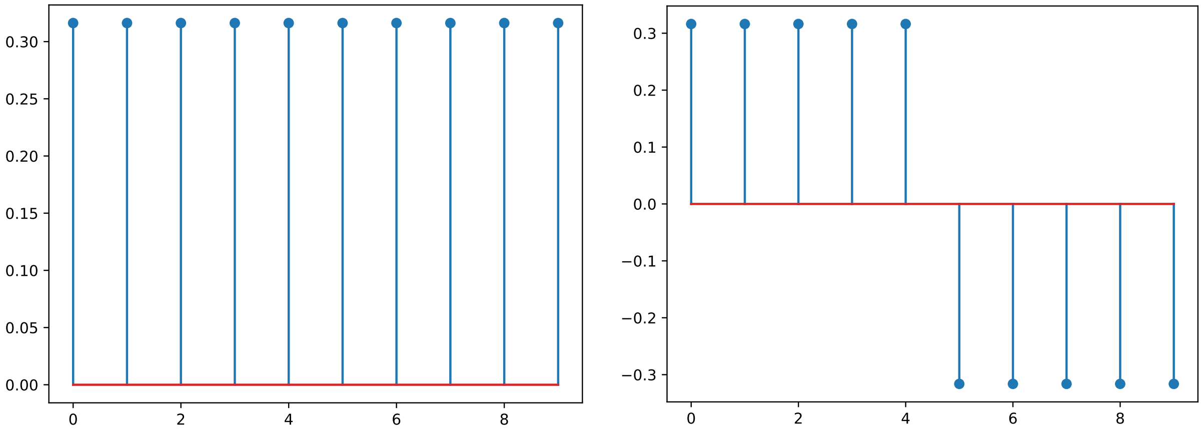


Figure 1: Digital Message

On the course website is a file, `data-communications.csv`, that contains a sequence of pulses that have been corrupted with noise; the signal is shown in Figure 1. The original, uncorrupted pulses can be represented in Python by

```
pulse0 = np.ones( 10 )
pulse0 = pulse0/np.linalg.norm(pulse0)
pulse1 = np.append( np.ones( 5 ), -1*np.ones( 5 ) )
pulse1 = pulse1/np.linalg.norm(pulse1)
```

as shown in Figure 2.



(a) binary 0

(b) binary 1

Figure 2: Pulses representing bits

A pulse represents a single bit. Each set of 8 bits in the message represents an ASCII character.

Write a Python program to read `data-communications.csv` and print to the screen the encoded message. This should be a standard line of text, for example,

```
test message
```

instead of

```
t
e
s
t

m
e
s
s
a
g
e
```

Here's the process:

(a) for each set of 10 numbers in the encoded message, determine which of the original uncorrupted pulses it is closest to; classify it as a '0' bit or a '1' bit

(b) for each set of 8 bits, determine its ASCII value

(c) translate each ASCII value to the corresponding character

Note that you should be able to read the message; if you get something with weird characters or mispellings, then you did something wrong.

The filename of your submission should be `hw03.py`.

General requirements about the Python problems:

a) Your code should run using Python 3.6 or greater and should not need libraries not provided in the Anaconda distribution unless instructed to do so.

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

c) 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.

d) The program should not prompt the user for values.

To submit the Python portion, do the following:

a) Create a directory using your net ID in lowercase characters. This should be something of the form `abc1234`.

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, assuming your net ID is `abc1234`, will be `abc1234.zip`.

d) Upload the zip'd file to Blackboard.

Review Questions

These are not for credit, but instead are intended to test your understanding of the concepts. You should be able to answer these without simply regurgitating equations.

1. What does it mean for a system to be time-invariant?
2. What does it mean for a system to be linear?

References

[Bar14] Richard Baraniuk. Discrete signals and systems, Cauchy-Schwarz inequality. https://www.youtube.com/watch?v=GqDIHmZj8dc, retrieved: December 23, 2014.