**ENGR 2013: Engineering Analysis and Applications**

**Laboratory No. 5: How Do You Find a Signal in Noise?**

**Goal:** The goal of this lab is to introduce you to a very important computation in signal processing – autocorrelation. This is used in a wide range of engineering and statistics applications. We are also going to build on what you learned in lab no.4.

**Preliminary Work:** In this lab we are going to explore the following equation:

.

Write a simple Python program that generates values of for the range [0, 10,000]. Set and , and . These are very common values in digital telephony. For , use the numpy function random to generate random values:

np.random.normal(0,1,1) .

This generates what we refer to as zero mean Gaussian white noise. Your program should just print to the terminal. To capture this data in a file, redirect the program output to a file using this command:

python gen\_signal.py > signal.dat

Your data should look something like this:

0.462001

2.116416

1.787632

…

In this lab, we are going to explore three fundamental mathematical/signal processing concepts. First, we introduce the autocorrelation function, which for this lab can be defined as:

,

where is the total number of samples used in the signal, which we refer to as the window size, and is the number of lags in your autocorrelation function. This function can be computing as a dot product between the vector and a version of that is shifted by samples.

Next, we introduce the concept of an [autocorrelation](https://www.geeksforgeeks.org/autocorrelation/) matrix (a special case of a [covariance matrix](https://www.geeksforgeeks.org/covariance-matrix/)):

Note this is a symmetric matrix whose values are simply a function of the distance from the diagonal (or more precisely related to the difference of the row and column indices). Later in the course we will see this is a form of what is called a [Toeplitz matrix](https://www.geeksforgeeks.org/find-if-given-matrix-is-toeplitz-or-not/) and because of that, it has special properties.

Finally, we are going to explore the condition number of a matrix. You can learn about this [here](https://www.geeksforgeeks.org/compute-the-condition-number-of-a-given-matrix-using-numpy/). The [condition number](https://en.wikipedia.org/wiki/Condition_number) is a simple calculation in Python that gives you insight into whether a matrix is invertible (low is good). A good discussion of this can be found [here](https://math.stackexchange.com/questions/261295/to-invert-a-matrix-condition-number-should-be-less-than-what#:~:text=Mathematically%2C%20if%20the%20condition%20number,can%20be%20just%20as%20bad.).

**Tasks:**

1. Set = 100, = 0, and =1. This generates a single sinewave a Hz which is sampled at  Hz. Compute and plot the autocorrelation function for . Explain what integer value, or lag, this function peaks at and how that relates to the frequency of the sinewave and the sampling frequency.
2. Using the values in (1), plot the maximum value of the autocorrelation function as a function of for . Plot this on a log-linear scale ( vs. ).
3. Set = 100, = 100, while keeping =1. Plot the autocorrelation function. Explain what you observe, namely how the peaks relate to the frequencies of the sinewaves.
4. Construct the autocorrelation matrix, , for , , while keeping . Create a table that shows the rank of the matrix as a function of for the range . What do you observe? Repeat this for , and . Does anything change?
5. Construct a table that shows the condition value of the autocorrelation matrix for for = (keep while you iterate over the values ). Explain what you observe.

For these tasks, place your code in a subdirectory t01, t02, …, t05. Also place a pdf file containing your plots (t01.pdf, t02,pdf, …) in the corresponding directory.

The TAs will ask you to explain your results, and you will be judged on your ability to explain the reasons why you are observing the outputs you generated.

**Summary:**

The sinewave signal you created has a dimensionality of (each sinewave requires two parameters). We also often say it has degrees of freedom. When you add noise, it makes the signal to appear more complex than it is. The degree to which the noise confuses the process depends on the signal to noise ratio – the power of the sinewaves divided by the power of the noise. In your statistics class, you will learn how autocorrelation and covariance functions can be used to remove noise from a signal and allow you to uncover its hidden structure.

Would this be useful to you in say, a place like a casino? ☺