[Robert Irwin]

ECE 3522: Stochastics

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 1912

# Problem Statement

In this experiment we are interested in correlation and covariance. The first part of the experiment called for a vector that contained 240 consecutive samples of an audio signal sampled at 8kHz. The 240 samples were to be taken starting at time, t = .9 seconds. Then we were to define a second vector y that contained the same number of samples. This vector however would start one sample later than the vector x. We were to compute the correlation of the vectors x and y. Then we were to redefine y starting at the sample one after the previous starting point. This was done 512 times. Each time we computed the correlation between x and y.

The next part of the assignment called for us to compute the covariance of a chunk of the signal. Again we used 240 samples. The following formula is used to calculate the covariance.

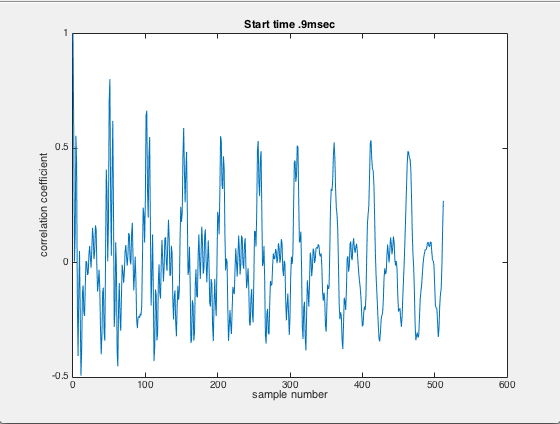


*Equation 1: Covariance*

In our case, i and j were defined on the interval [0,15] and N is 240. This results in a 16x16 covariance matrix. This matrix was computed for starting times of 1.1 seconds and 3 seconds. Then we were to figure out how and why the matrices were different.

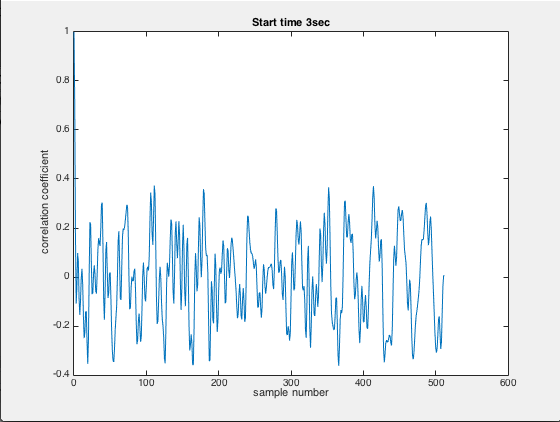
# Approach and Results

For the first part of the experiment we had to pick out 240 samples starting at .9seconds. Multiplying the sampling frequency by the desired starting point gave us the proper sample. Then we implemented a for loop that stored the next 239 samples into a new variable. This resulted in the desired x vector. A similar approach was used to shift the window of samples. An index was defined to range [1, 512]. Then the starting point was shifted by k and the next 239 samples were stored in a column vector. At the end of this process, a 240x512 matrix was the result. Each of the 512 columns held the signal samples corresponding to the desired k value. Then we computed the correlation between x and each column of the 240x512 matrix. To compute the correlation the Matlab macro corrcoef was implemented. For two inputs, this macro outputs a 2x2 matrix, with ones on the main diagonal and the correlation coefficient everywhere else. These values were stored in a variable that we would later use to plot the correlation. The process was repeated for a starting time of 3 seconds. The two resulting plots are shown below.



*Figure 1: Correlation Coefficient Start Time = .9 seconds*

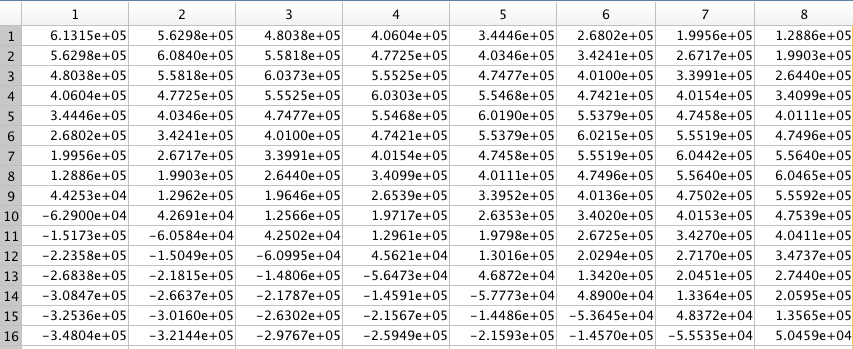
From the plot we can see that the correlation coefficients oscillate around 0. We also notice that the correlation is 1 for k = 0. This is the result we expect as at k = 0 we compute the correlation between the same vector. We also know that the audio signal at this time is predominantly sinusoidal which explains why the correlation oscillates around 0. The zero values indicate when the vectors are completely out of phase with each other.



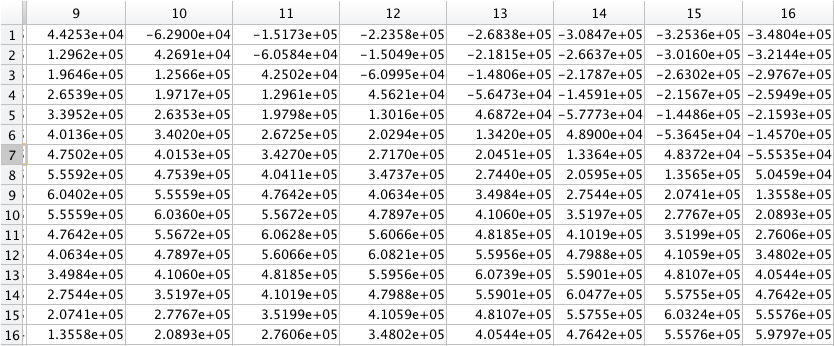
*Figure 2: Correlation Coefficient Start Time = 3 seconds*

This part of the signal contains only noise. This is the reason the magnitude of the correlation coefficients are lower than that in the figure above. Again, we see that when k = 0, the correlation coefficient is 1, which means they are perfectly correlated.

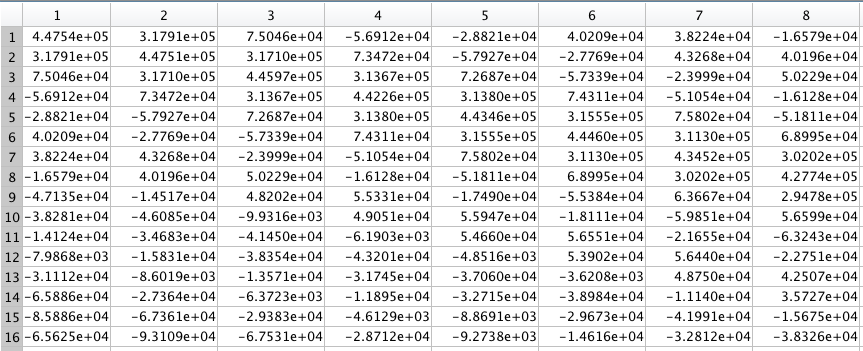
In order to calculate the covariance matrix, we had to account for the indexing issues that arose. For all negative indices, we had to take samples before our starting point corresponding to the magnitude of the index. Once this was completed, we simply implement equation 1. This produces a 16x16 covariance matrix. The covariance matrices for start times of 1.1 seconds and 3 seconds are shown below.



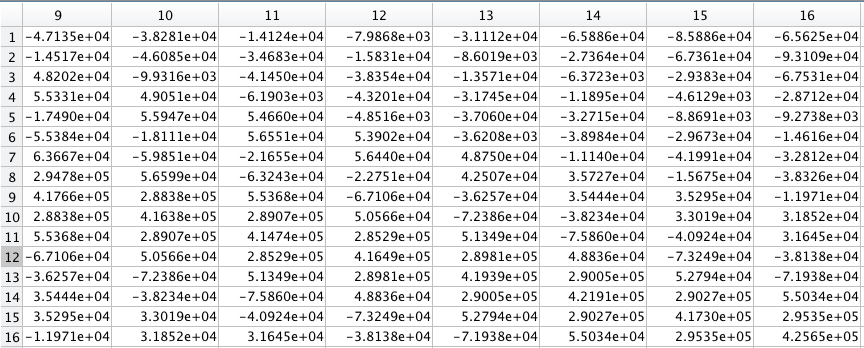
*Figure 3: Covariance Matrix Start Time 1.1 seconds*

**

*Figure 4: Covariance Matrix Start Time 1.1 Seconds*

**

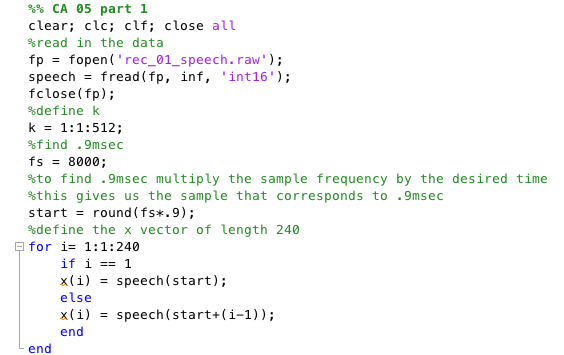
*Figure 5: Covariance Matrix Start Time 3 Seconds*

**

*Figure 6: Covariance Matrix Start Time 3 Seconds*

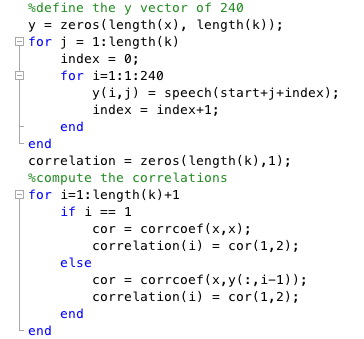
In comparing the two matrices, we notice the magnitude of the covariance is much greater when the start time is 1.1 seconds. This is because during this time interval, there is speech present in the signal. The magnitude of the covariance measures the strength of the linear relation. Because a speech signal is relatively normal, we expect the covariance of this part of the signal to be greater than the covariance of the portion of the signal that is strictly noise.

# MATLAB Code



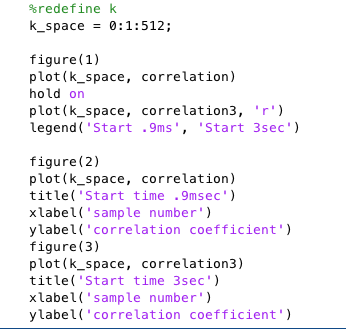
*Figure 7: Defining the X Vector*

In this code we are defining our variables according to the specifications in the assignment. The variable start is an index variable. This variable is set equal to the sampling frequency multiplied by the desired starting time. This results in the desired sample number. Then we simply define a new vector starting from this point that includes the next 239 points.

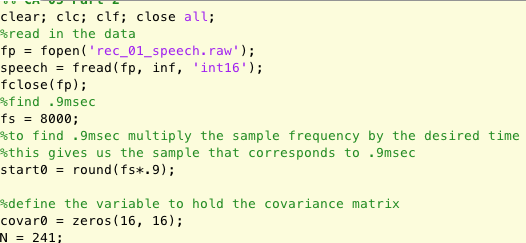
**

*Figure 8: Defining the Y Vectors and Computing the Correlation Coefficients*

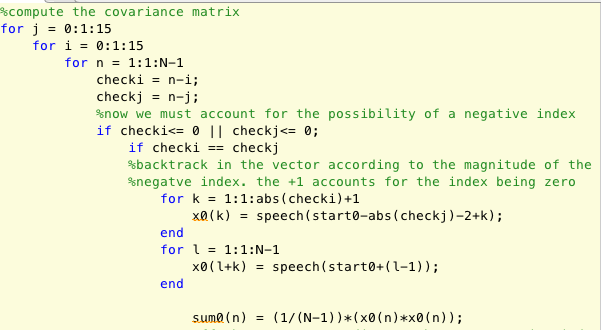
In this piece of code we define the vector y and then begin computing the correlation coefficients. To do this, we use the Matlab macro corrcoef. This outputs a matrix with the correlation coefficient in the off-diagonal position of the matrix.

**

*Figure 9: Plotting the Correlation Coefficients*

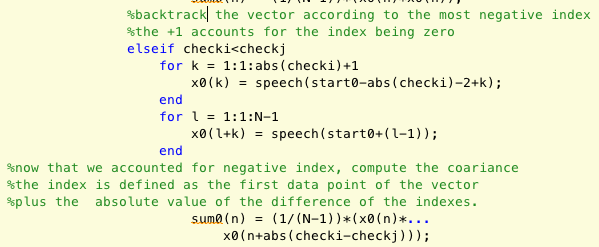
**

*Figure 10:Defining Variables*

**

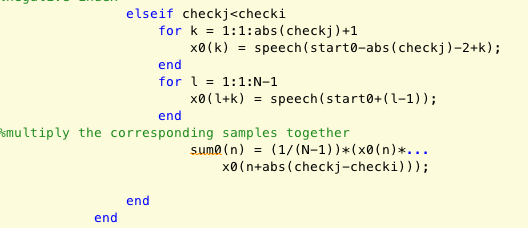
*Figure 11: Begin Computing the Covariance Matrix*

In the code above, we are checking for negative indices. If the index is negative, we have to go backwards in the signal corresponding to the magnitude of the negative index. This shows the case that n-i=n-j. The resulting product is then stored in a variable that will be summed later. This summation gives us the covariance for a specific i and j combination.

**

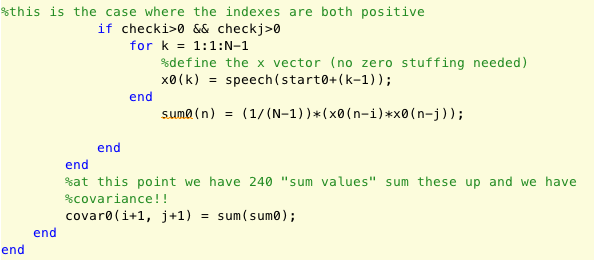
*Figure 12: Computing the Covariance Matrix*

The process is the same as above. However this considers the case that n-i<n-j.

**

*Figure 13: Computing the Covariance Matrix*

This is the case that n-j<n-i.

**

*Figure 14: Computing the Covariance Matrix*

This shows the case that both indices are positive. At this point in the code, all possible n’s have been looped through and we sum up the individual products calculated previously. Once all i’s and j’s have been looped through, we have our 16x16 covariance matrix.

This process is repeated for the start time of 1.1 seconds and 3 seconds. However start is defined as fs\*1.1 and fs\*3 respectively.

# Conclusions

In this assignment it was shown that the correlation of a speech signal is much higher than the correlation of a signal that is strictly noise. Because we know that noise is random, we expected this result. We also notice that, like the audio signal itself, the correlation coefficients that result in shifting one vector by a single sample also oscillate around 0.

A couple of interesting observations can be made from the covariance matrices. The first one is that the covariance magnitudes are higher for the part of the signal that contains speech. We also notice that for all values of i=j, the covariance is almost identical. This is because covariance measures the strength of the linear relation between two random variables. If i=j, the two vectors are identical, which would result in almost the same covariance for every calculation.

We can also notice that the value of the covariance for i=j is the largest magnitude of the covariance, and as i stays constant while j increases, or vice-versa, the covariance decreases. This makes sense as the vectors are becoming more distinct with each iteration of i and j.