Emilie Doyle

ECE 3522: Stochastic Process in Signals and Systems

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

# Problem Statement

The subject of this computer assignment is the calculation of correlation and covariance within sets of data. The given data set is the audio signal sampled at 8 kHz that was given out at the beginning of the semester for the purpose of these assignments. The first task of the assignment is to essentially create our own autocorrelation function. This was accomplished by calculating the correlation between two segments of the same overall set of data. The first sample was the first set of 240 data points starting at 0.9 seconds in the signal. The second sample was also 240 data points from the audio signal but it was offset by a value ranging from 0 to 512. All of these correlation values were then plotted together to view the correlation of the sections all together. This process was then completed identically for time starting at 3.0 seconds. The second task was to calculate the covariance matrix between similar sets of data from the audio signal. It was first started at 0.9 seconds, but it was also run starting at 1.1 seconds, as well as 3.0 seconds. All of these were compared for potential differences.

# Approach and Results.

To get the data from the audio signal into MATLAB, I followed the same process that I have for all the previous computer assignments involving this audio signal. I used the fopen and fread commands to get the data into an array D and then used the data right from there. For the first task I had to find what data point in D correlated to 0.9 seconds, given that 0.03 seconds equaled 240 points. Using that relationship, I calculated 0.9 seconds to be at data point 7200 and 3.0 seconds to be at data point 24000. I put those in an array (I) to reuse the same code for them. I also made an array (J) to hold the seconds values, which will be used later in labeling plots. The array, x , that would contain the 240 points was made using linspace command. I also used it to make arrays n and j for gathering and storing data, as well as plotting. I made one big for loop to loop through the values of I, and then in that loop I have more loops for various things. The first loop in I is for obtaining those original 240 data points and storing them in the array x. The next loop goes through the data and offsets it by the value k, and it stores the result in a giant, master array that contains all of the shifted arrays for all values of k. Lastly, I calculated the correlation between the original array x, and each of the offset values in the array y, and these values were stored in the array n. I then subplotted these correlation arrays against a linearly spaced array, j, so that the correlation from both t= 0.9 seconds and t= 3.0 seconds could be observed.

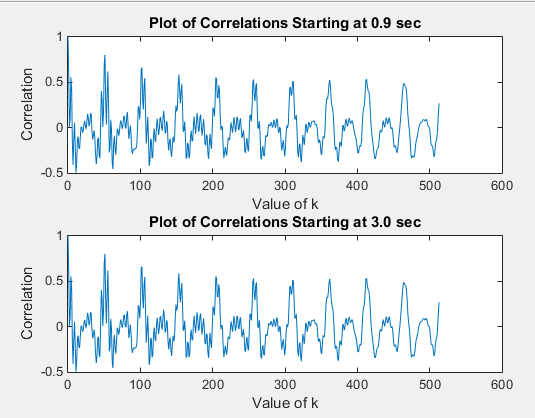


Figure : Part 1 Plots of Correlations

For the second task, I set up an array I that had the data point values that correspond to the times from which we were to start from. I also included an array J with the actual time values that match up, just to keep track of them. I created an array X that contained 240 doubles in order to store the data from D. I set N at 240, which was the number of samples to be collected and put into X. I also made a 15 by 15 array x to put the covariance matrix in. I made a loop to go through the values of D and store the correct range in the array X. Next I looped through the array x, going from values of 1 to 16 for both row and column. I then put another inner loop to cycle through the 240 data points and to calculate the covariance of the points based on the given formula. These values were then stored in the appropriate spot in the array x and could be compared at a later date. Note that for the sake of clarity, it would be best to cycle through only 1 possible time value in a running of the program. This could be controlled in the for loop controlled by variable m- just by selecting which value of I you wish.

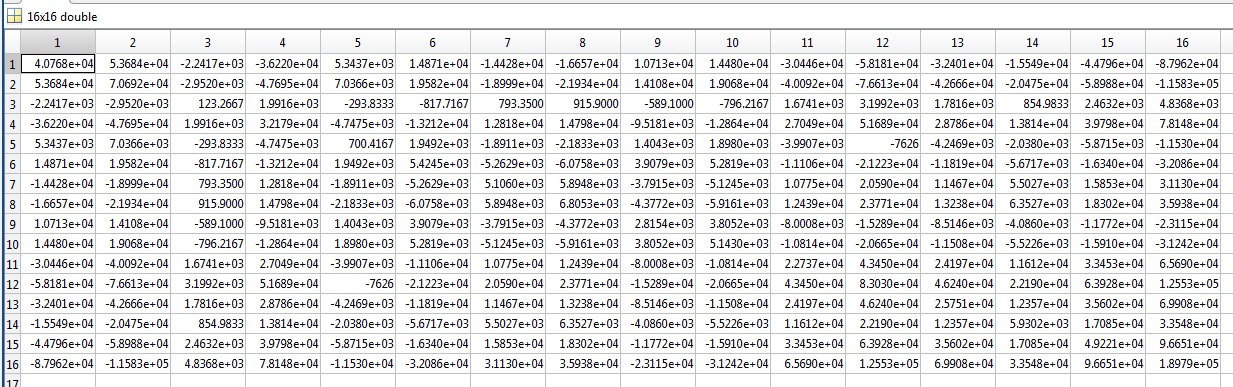


Figure : Covariance Matrix Starting at 0.9 seconds

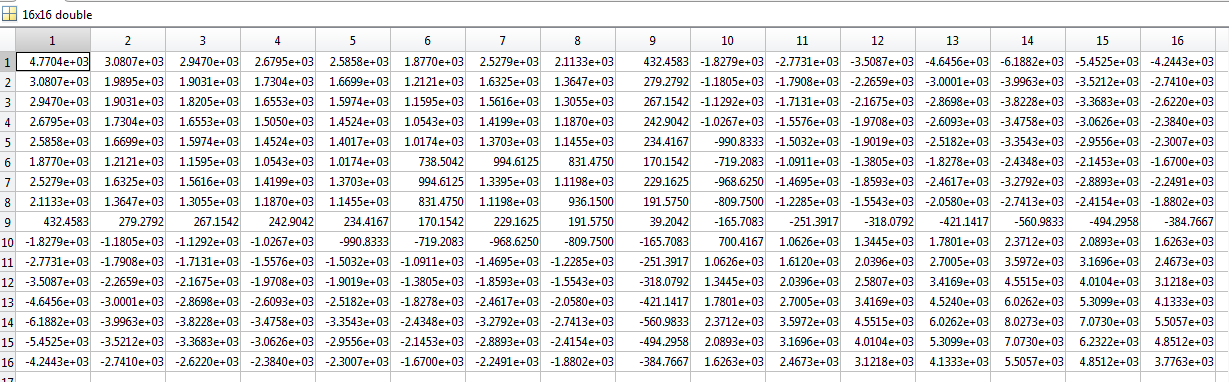


Figure : Covariance Matrix Starting at 1.1 seconds

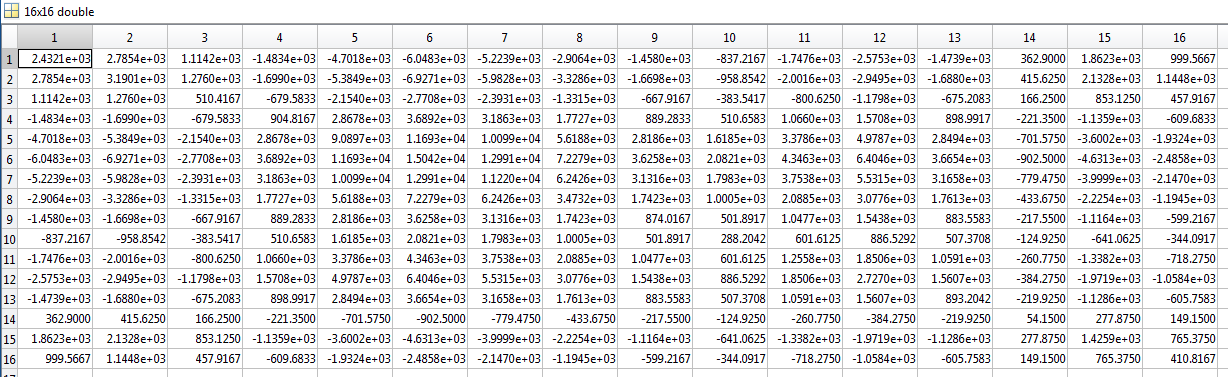


Figure : Covariance Matrix Starting at 3.0 seconds

# MATLAB Code

%StochCA5

%Read in Audio Signal

FID= fopen('rec\_01\_speech.raw', 'r');

if FID<0

error('Failed to Open File');

end

D=fread(FID, inf, 'int16');

fclose(FID);

%1

I=[7200 24000];

J=[ 0.9 3];

x=linspace(1, 240, 240);

n=linspace(1,512,512);

j=linspace(0,513,513);

%240= 0.03 seconds, want to start at 0.9 seconds, which is 30 times as

%much, so start at 7200

for m=1:length(I)

for i= (I(m)+1): (I(m)+240)

x(i-I(m))= D(i-1);

end

%Get all values for y for all values of k

%y=zeros(240,513);

for k=1:513

for i= (I(m)+1): 7441

y(i-I(m),k)= D(i-1+(k-1)); %First row of y is D(7200) then D(7201) which is k=0...

%second row starts with D(7201) so k=1, and so on. k is (row - 1)

end

n(k)= corr( y(:,k),y(:,1));

end

subplot(2, 1, m)

plot(j, n)

str=sprintf('Plot of Correlations Starting at %.1f sec', J(m));

title(str);

xlabel('Value of k');

ylabel('Correlation');

end

%2

I=[7200 8800 24000];

J=[ 0.9 1.1 3];

X=linspace(1,240,240);

x=zeros(16,16);

N=240;

for m=1:1%length(I)

for l=I(m)+1: (I(m)+240)

X(l-I(m))=D(l-1);

end

end

for i=1:16

for j=1:16

for n=1:N

x(i,j)=(1/N)\*(sum((X(N-i))\*(X(N-j))));

end

end

end

# Conclusions

The graphs for the first task show that there is a positive correlation between various sets of the given data, and for others there is a negative correlation between the various sets of data. These sections of high correlation appear to happen almost periodically, and are present in both that starting at 0.9 seconds, as well as starting at 3.0 seconds. The implication of this is that we can use the correlation between past sets of data and use it to model future values, if we can predict whether or not the correlation will be positive or negative. The data appears to smooth out considerably in the later portion of the correlation array, so that may indicate that this method may be helpful in long term analysis of values or stocks. On the other hand, it may mean that data is getting smoothed over, and it is not giving us an accurate showing.

For the second task, the result was three matrices, 15 by 15 each. The first one corresponds to the covariance matrix with data points starting at 0.9 seconds. The second one corresponds to the covariance matrix with data points starting at 1.1 seconds, and the last matrix corresponds to the covariance matrix with data points starting at 3.0 seconds. One obvious account for the differences between the 1.1 starting matrix and the 3.0 starting one is the difference in the data. An audio signal would most likely naturally vary between a range, due to the consistency in the person’s speaking voice/ whatever the audio signal is. At the same time, if it is a person speaking, and they are speaking, say a sentence or just anything that is not the same letter over and over again, the content of the signal would vary based on the letter, pitch, tone and frequency it is spoken at. One might expect to see that the two matrices would converge if the data sample points were closer together in time.