Edward Moore

ECE 3512: Stochastic Processes in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19121

# Problem Statement

This assignment provides an introduction to covariance and correlation calculations. Covariance and correlation of an audio signal are calculated over multiple ranges and compared. The goal of this is to provide students with some insight and intuition as to how these calculations work and what they are representative of.

# Approach and Results

To plot statistical correlation, the correlation coefficient is to be calculated over a range of values. To do so, two vectors are set over a portion of the data, one of them shifted one data point with each iteration of the coefficient calculation. This is what’s known as autocorrelation. For the given audio signal, autocorrelation is to be plotted starting at 0.9 and 3.0 seconds and calculated for 512 samples.

The original audio signal is shown below in Figure 1 with the two autocorrelations plotted in Figures 2 and 3. Both of the outputs oscillate around 0. This is expected since the original signal also oscillates around 0. Due to this, the calculated correlation should oscillate between -1 and 1 evenly since the correlation calculations flip between being directly and inversely related. All relevant code can be seen in the MATLAB Code section.



Figure 1



Figure 2



Figure 3

Next, a 16x16 covariance matrix is calculated over three time ranges and compared. The matrix values were derived from Equation 1. Values i and j were two vectors containing integers from 0 to 15. The covariance was calculated using 240 samples. Ranges to be calculated over start at 0.9, 1.1 and 3.0 seconds. Code used is presented in the MATLAB Code section. The result of the first matrix calculations is shown below in Figure 4.



Equation 1



Figure 4

To ensure correct calculations, values were plugged into Equation 1 and calculated to ensure the matrix was filled with correct values. When starting at sample 7200 (0.9 seconds), at i = 5 and j = 5 the equation yielded 0.0104, equivalent to the value in the above matrix, meaning the values filled in as expected. This result is pictured below in Figure 4. The other two matrices were then filled to be compared, these are shown below in Figures 6 and 7.

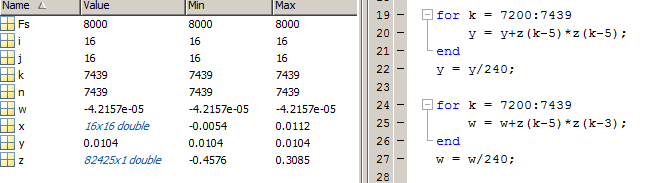


Figure 5



Figure 6



Figure 7

It appears that just as with the autocorrelation calculations, the covariance oscillates around 0. This is, again, due to the fact that the original signal is periodic.

# MATLAB Code

Code for autocovariance starting at 0.9 seconds:

[z, Fs] = audioread('rec\_01\_speech.wav');

figure(1);

plot(z);

title('Audio Signal')

xlabel('time')

ylabel('amplitude')

covmat = zeros(512, 1);

x = zeros(240, 1);

y = zeros(240, 1);

for b = 1:512 %want 512 covariance measurements

for a = 7200:7440 %to start at .9 sec 8000Hz\*.9 = 7200

x(a-7199) = z(a); %x is 240 audio signal samples from sample 7200

y(a-7199) = z(a+b); %y is 240 audio signal samples like x but shifted one right with each iteration

end

cov = corrcoef(x,y);

covmat(b) = cov(1,2);

end

figure(2);

plot(covmat);

title('Autocorrelation of Audio Signal');

xlabel('Correlation Caculation Number');

ylabel('Correlation Coefficient');

Code for covariance matrix starting at 0.9 seconds:

[z, Fs] = audioread('rec\_01\_speech.wav');

y = 0;

w = 0;

x = zeros(16,16);

for i = 1:16

for j = 1:16

x(i,j) = 0;

for n = 7200:7439

x(i,j) = x(i,j)+z(n-i)\*z(n-j);

end

x(i,j) = x(i,j)/240;

end

end

# Conclusions

The results of the calculations performed here could have easily been predicted. This is in thanks to the fact that the original signal is periodic (oscillates). From an oscillating signal, calculating the autocorrelation and covariance matrix should produce oscillating results as are seen here. In comparison, if the signal were a pure sine wave the autocorrelation would oscillate evenly between -1 and 1.