Computer Assignment (CA) No. 5: Covariance and Correlation

Tyler Berezowsky

February 25, 2015

1 Problem Statement

The goal of this assignment is to introduce covariance and correlation calculations. For this task, only the audio signal was used. The tasks to be accomplished are:

1. Define a vector, x, of length 240 (30 msec) that contains the 240 samples of the signal starting at $t = 0.9$ secs. Define a second vector, y, which also represents 240 samples, but consists of samples shifted by k samples (e.g., k = 1 implies y starts one sample later than x). Plot the statistical correlation between x and y for k = 0, 1, ..., 512. Can you explain what you observe? (Hint: think about what would happen if the signal were a sinewave.)

   Repeat this for t = 3.0 secs. Compare the two functions and relate them to properties of the audio signal.

   The function you are plotting is known as the autocorrelation function. It is a minimum phase version of the actual signal. You can learn more about that in a course on digital signal processing.

2. Again, start at $t = 0.9$ secs. Take the first 16 samples as a vector: $x = [x_1, x_2, x_3, ..., x_{16}]$. Compute the covariance matrix using 240 samples. Each element in the matrix is governed by the equation:

$$x[i,j] = \frac{1}{N}\sum_{n=0}^{N-1} x[n-i]x[n-j] \tag{1.1}$$

   where i is defined over the range [0, 15] and j is defined over the range [0, 15]. Do this for t = 1.1 secs and t = 3.0 secs. Compare the two matrices and explain why they are different.

2 Approach and Results

2.1 Correlation

The correlation coefficient is defined by equation 2.1. It consists the covariance of two random variables divided by the standard deviation of each variable multiplied. This reduces the range of covariance to -1 and 1.

$$\rho_{XY} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sqrt{Var(X)Var(Y)}} \tag{2.1}$$

The correlation coefficient provides a metric to determine how linearly related two random variables are. If the coefficient is negative 1, they are inversely related. If the coefficient is a positive one, they are directly related. A coefficient of zero states that the variables are not linearly correlated. Note that this does not mean they are not correlated at a higher power.

For task 1, the correlation of a vector, x ,from the provided audio signal was taken against itself, y, but shifted by k bits. ($k = 512$) The starting time of x was equal to $x_1 = 0.9$, $x_2 = 1.1$, and $x_3 = 3.0$ seconds. The resulting correlation coefficients were plotted against their respective k value. The plots for offsets x_1, x_2, and x_3 can be seen in figure 2.1.

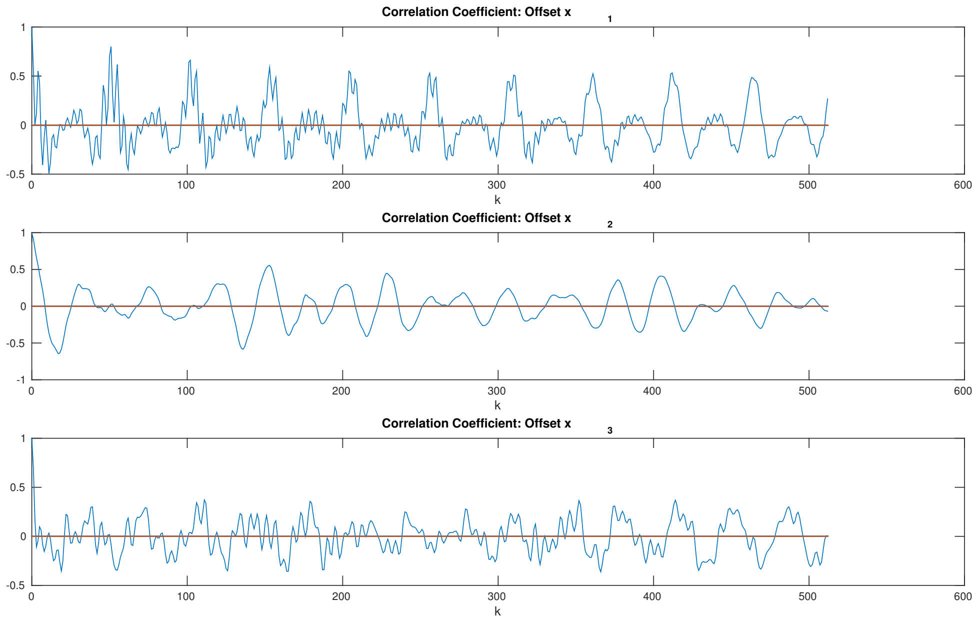

Figure 2.1: Correlation of audio signal vector x starting at offset x_i with itself shifted by k bits. Offset $x_1 = 0.9s$, $x_2 = 1.1s$, and $x_3 = 3.0s$

For every offset value, the correlation coefficient begins with 1 as k starts at 0. This compares the x vector with the un-shifted version of itself, therefore the correlation is 1. Offset x_1 decays after $k = 0$ with periodic pulses. Offset x_2 also decays after $k = 0$ with random peaks. Offset x_3 decays after $k = 0$ substantially faster than x_1 and x_2 and remains under 0.5 to -0.5 for the duration of the rest of the signal.

2.2 Covariance Matrix

The covariance matrices for offsets $x_0 = 0.9$, $x_1 = 1.1$, and $x_2 = 3.0$ seconds are respectively displayed below. The covariance matrix computes the covariance between the ith and jth elements of the applied vector. Therefore, when $i = j$ the same elements in the vector are being compared and should represent the maximum value of covariance between elements in the matrix. Due to distribution of the elements, all covariance matrices are also symmetrical along the diagonal. This is represented in the matrices generated.

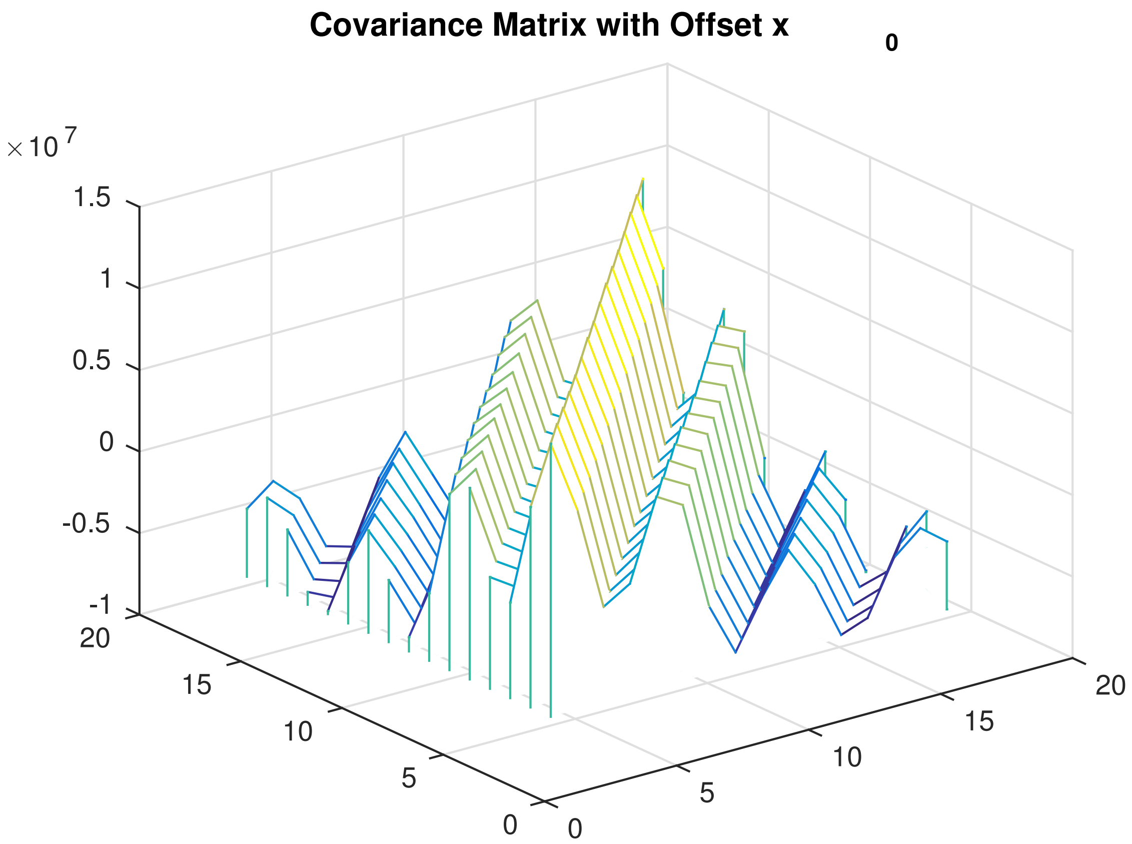

Figure 2.2: Covariance matrix with offset $x_0 = 0.9$ seconds.

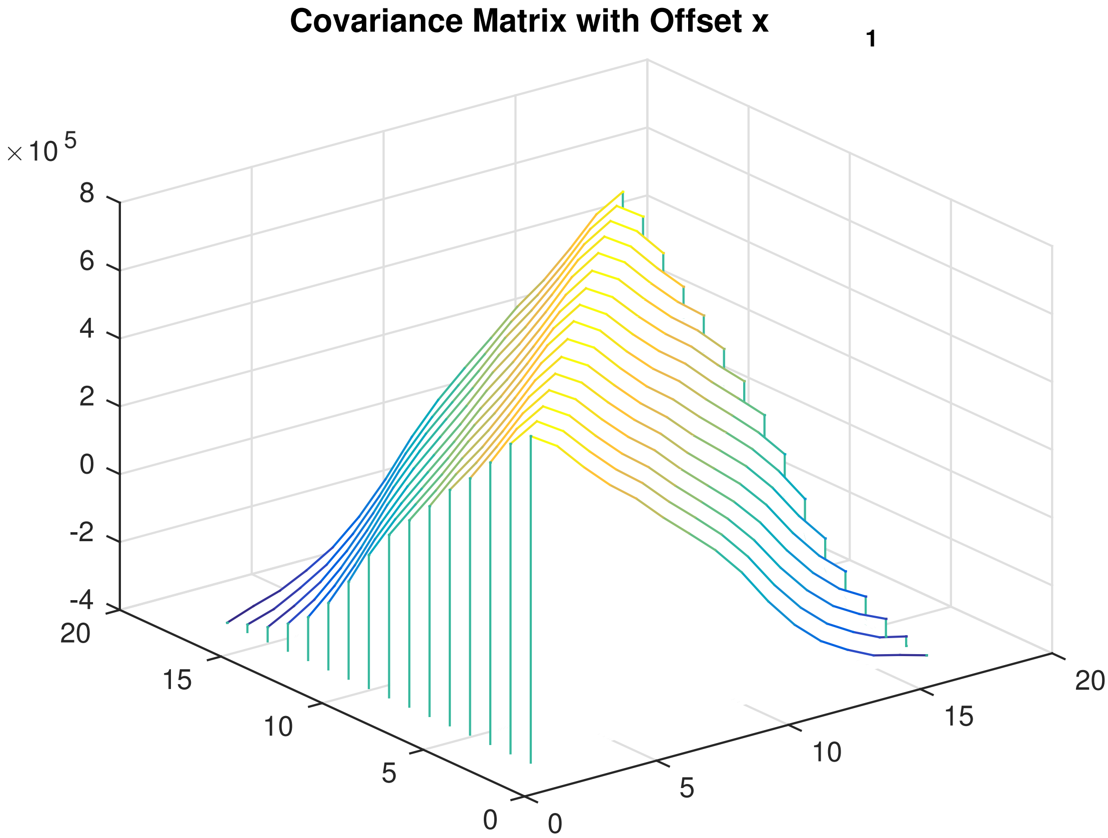

Figure 2.3: Covariance matrix with offset $x_1 = 1.1$ seconds.

9.

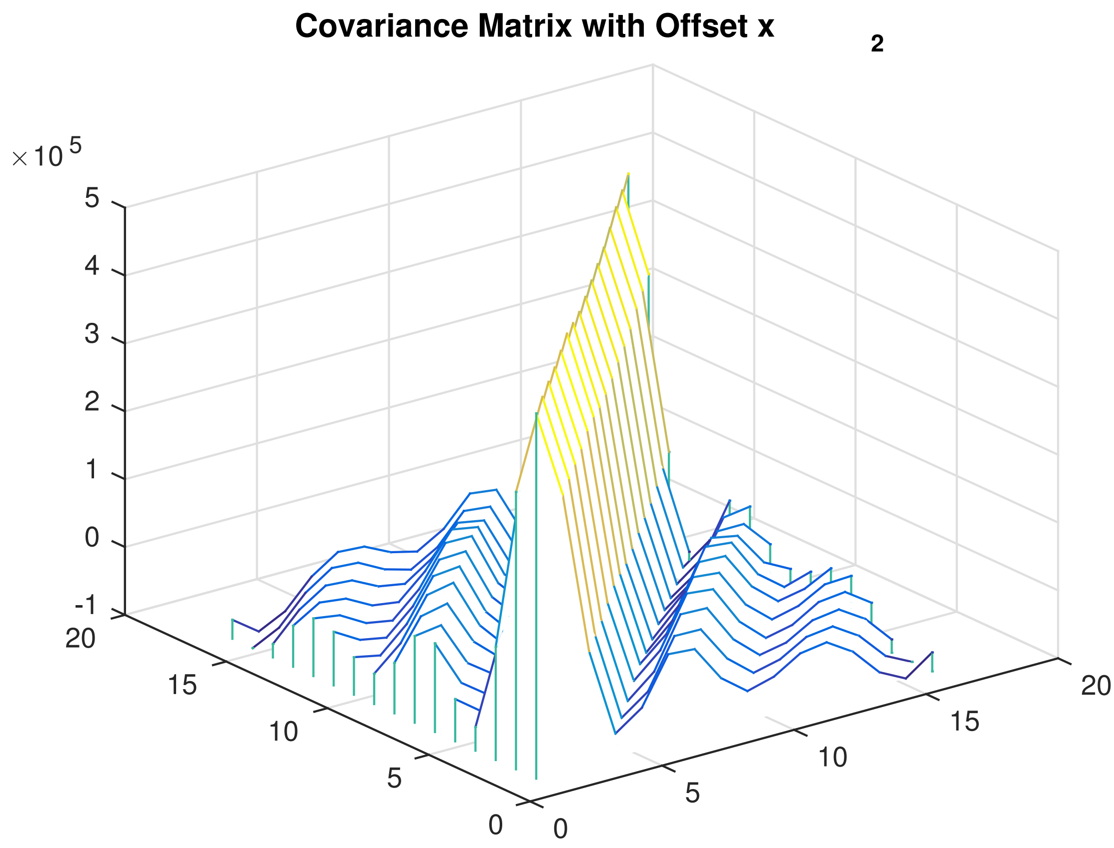

Figure 2.4: Covariance matrix with offset $x_2 = 3.0$ seconds.

2.3 Spectrum Analysis

To further understand the distribution of covariance in the matrices, the Fourier transform of the audio vectors was taken. This provided a glimpse into the audio spectrum of each vector. The transforms are displayed in figure 2.5 below.

The frequency spectrum for x with offset x_0 contains large peaks between 0 to 500 Hz and 1200 to 2000 Hz. The frequency spectrum for x with offset x_1 contains most of its power under 500 Hz. The frequency spectrum for x with offset x_2 contains peaks from 0 to 2000 Hz.

Attempting to bridge the gap between the audio spectrum and the covariance matrices, I'll start with the data generated by the 3.0 second offset. The audio spectrum contains various peaks from 0 to 2000Hz. The correlation matrix displays a peak along the diagonal, as to be expected, but then drops within 5 samples to near zero. This behavior is different than the other two offsets which contain much slower decaying or oscillating decay. This suggests the signal is highly uncorrelated which is also possibly represented by the amount of energy spread across the spectrum seen in the FFT. The covariance of the other two offsets, x_0 and x_1, both drop gradually to zero. x_0 simply appears to do it in an oscillatory manner. The covariance should drop over time with any random audio signal, but it does not explain the oscillatory behavior for offset x_0 and why x_1 lacks it.

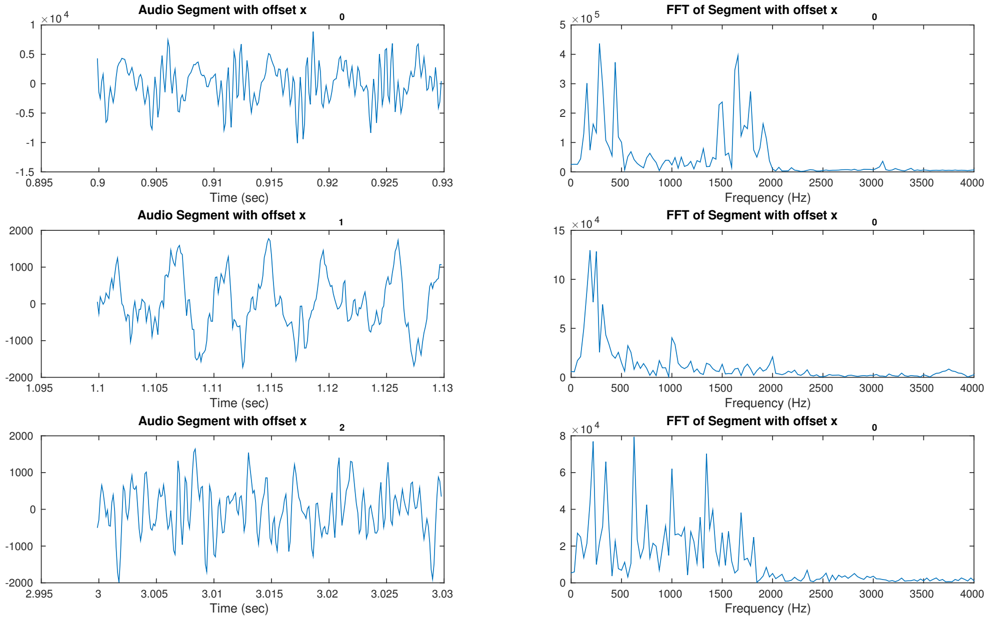

Figure 2.5

3 MATLAB Code

Listing 1: 'MATLAB solution for CA: 05.'

```
%% Computer Assignment 5

% Import the .raw file into an array

filename = 'rec_01_speech.raw';
file = fopen(filename, 'r');
audio = fread(file, inf,'short');

%audio = audio(1:10000); % cut down audio!

sampleRate = 8e3
time = (0:length(audio)-1)*1/(sampleRate);

%% Task 1
% \rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X\sigma_Y} =
% E[(X-\mu_x)(Y-\mu_Y)]}{\sigma_X\sigma_Y}

clf;

h = [floor(sampleRate*0.9) floor(sampleRate*1.1) floor(sampleRate*3.0)];

figure();
title('Autocorrelation Function');
for i=1:length(h);
    x = audio(h(i):h(i)+239);
    corr_xy = zeros(512+1);
    for k = 0:512
        y = audio(h(i)+k:h(i)+239+k);
        corr_xy(k+1) = mean( (x - mean(x)).*(y - mean(y))) / (std(x)*std(y));
    end
```

```
    k = 0:512;
    subplot(3,1,i)
    plot(k, corr_xy);
    titleSTR = sprintf('Correlation Coefficient: Offset x_%d', i)
    title(titleSTR);
    xlabel('k')
end

%% Task 2

N = 240;
x_0 = floor(sampleRate*0.9); % 0.9 second delay
x_1 = floor(sampleRate*1.1); % 1.1 second delay
x_2 = floor(sampleRate*3.0); % 3.0 second delay


corr_matrix_x0 = zeros(16, 16);
corr_matrix_x1 = zeros(16, 16);
corr_matrix_x2 = zeros(16, 16);


% Due to lazyness, I'm copying and pasting code for each offset instead of
% generating a for loop to increment through and array of offset values.

% index matrix and insert blah
for i=0:15
    for j=0:15
        sum = 0;
        for n = 0:N-1
            sum = sum + audio(n-i+x_0)*audio(n-j+x_0);
        end
        corr_matrix_x0(i+1,j+1) = (1/N)*sum;
    end
end

for i=0:15
    for j=0:15
        sum = 0;
        for n = 0:N-1
            sum = sum + audio(n-i+x_1)*audio(n-j+x_1);
        end
        corr_matrix_x1(i+1,j+1) = (1/N)*sum;
    end
end

for i=0:15
    for j=0:15
        sum = 0;
        for n = 0:N-1
            sum = sum + audio(n-i+x_2)*audio(n-j+x_2);
        end
        corr_matrix_x2(i+1,j+1) = (1/N)*sum;
    end
end

%% Generate Plots
% I have no clue at what I'm looking at so why not plot it in 3D. 3D is
% cool right.
figure()
waterfall(corr_matrix_x0);
title('Covariance Matrix with Offset x_0');
figure()
waterfall(corr_matrix_x1);
```

```
title('Covariance Matrix with Offset x_1');
figure()
waterfall(corr_matrix_x2);
title('Covariance Matrix with Offset x_2')

%% Plot Audio Signal Segments

NFFT = 2^nextpow2(240);
f = sampleRate/2*linspace(0,1,NFFT/2+1);


figure()
subplot(3,2,1);
    plot(time(x_0: x_0 + 239), audio(x_0: x_0 + 239));
    xlabel('Time (sec)')
    title('Audio Segment with offset x_0')
subplot(3,2,2);
    %histogram(audio(x_0: x_0 + 239), 240);
    sigFFT = abs(fft(audio(x_0: x_0 + 239)));
    plot(f,2*abs(sigFFT(1:NFFT/2+1)));
    xlabel('Frequency (Hz)')
    title('FFT of Segment with offset x_0')

subplot(3,2,3);
    plot(time(x_1: x_1 + 239), audio(x_1: x_1 + 239));
    xlabel('Time (sec)')
    title('Audio Segment with offset x_1')
subplot(3,2,4)
    %histogram(audio(x_1: x_1 + 239), 240);
    sigFFT = abs(fft(audio(x_1: x_1 + 239)));
    plot(f,2*abs(sigFFT(1:NFFT/2+1)))
    title('FFT of Segment with offset x_0')
    xlabel('Frequency (Hz)')
subplot(3,2,5)
    plot(time(x_2: x_2 + 239), audio(x_2: x_2 + 239));
    xlabel('Time (sec)')
    title('Audio Segment with offset x_2')
subplot(3,2,6)
    %histogram(audio(x_2: x_2 + 239), 240);
    sigFFT = abs(fft(audio(x_2: x_2 + 239)));
    plot(f,2*abs(sigFFT(1:NFFT/2+1)))
    xlabel('Frequency (Hz)')
    title('FFT of Segment with offset x_0')
```

4 Conclusions

Correlation and covariance are interesting metrics which are utilized to track how effectively two random variables are linearly related. With some ingenuity, the second task illustrates how the covariance between each element of a vector can be calculated and from that properties of the signal could be possibly extrapolated such as the amount of noise. Despite this tiny revelation, it presents more questions than answers. Why is the first and second offset different? Why is the first offset oscillating? How exactly does the covariance matrix calculation work?