Marc Jurchak

ECE 3512: Signals – Continuous and Discrete

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

# Problem Statement

We have three problems. The first is to load two data files into MATLAB. The first file is Google’s daily stock prices since its inception, and the second data file is a sound file sampled at 8kHz. The second problem is to compute global statistics on these signals. We will compute minimum value, maximum value, mean, median and variance. In the third problem we loop over the two data files and compute the average and variance between various window and frame sizes. We loop over all combinations of window sizes 1, 7, 14, 30 and frame sizes 7, 30 for the Google stocks. For the sound signal we use window sizes 40, 80 and 160 and frame sizes 160 and 240.

# Approach and Results

In the first problem, we loaded the Google stock prices with the “xlsread” function, and loaded the speech data with “fopen” and “fread” functions. For the Google stock prices, we plot only the closing stock price. Figure 1 is the Google stock price plot, and figure 2 is the speech signal.



Figure 1: Closing stock price plot

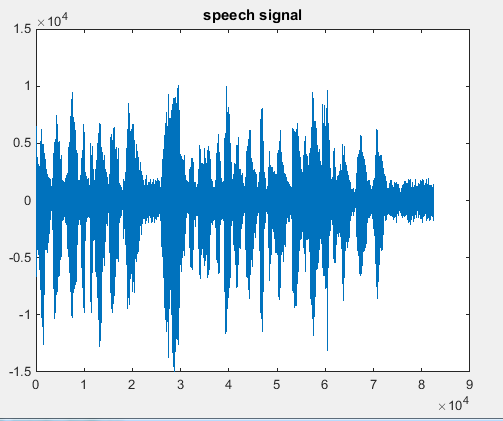


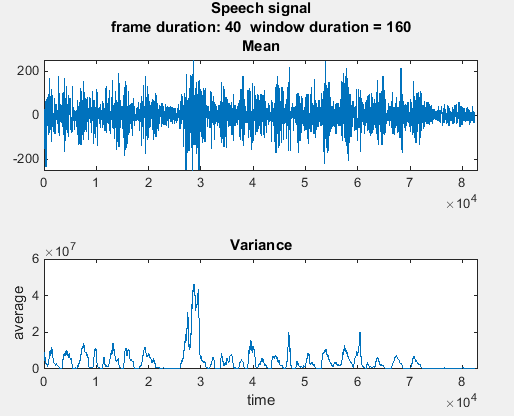
Figure 2: speech signal plot

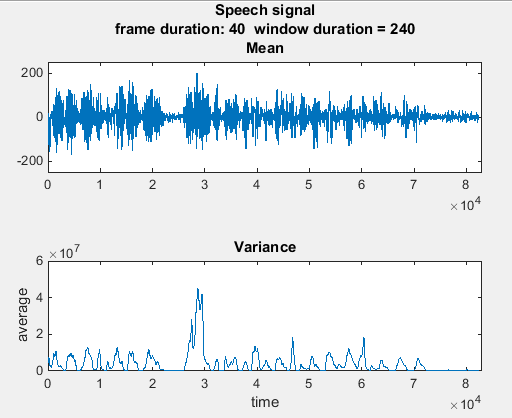
For the second problem, we compute the minimum value, maximum value, mean, median and variance of each statistic. Table 1 displays these statistics.

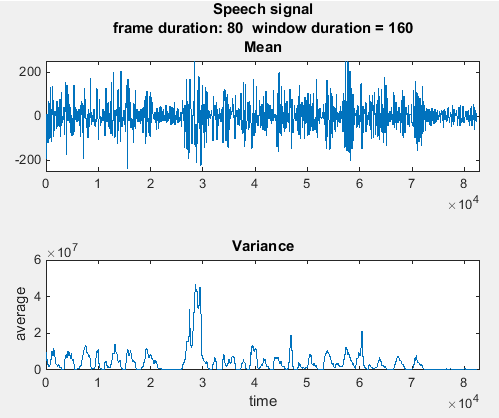
|  |  |  |
| --- | --- | --- |
|  | closing stock price | sound signal |
| minimum value | 49.95 | -14993 |
| maximum value | 609.47 | 10104 |
| mean | 286.7374 | -0.3891 |
| median | 264.825 | 83 |
| variance | 1.62E+04 | 4.14E+06 |

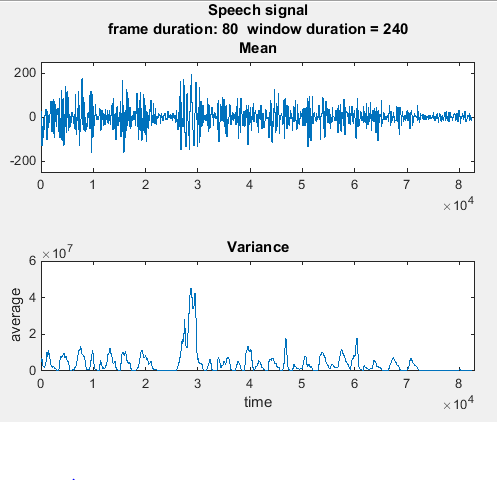
Table1: Statistics of the closing Google stock prices, and the speech signal.

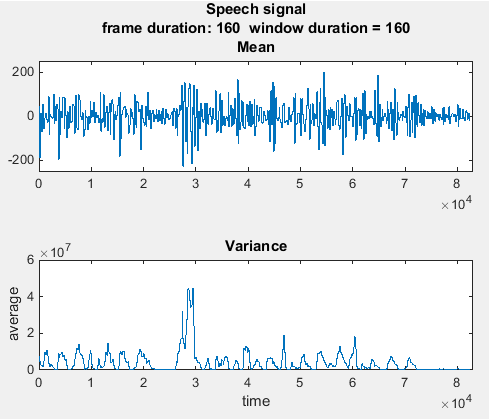
In the third problem we plot the Google stock prices with window sizes 1, 7, 14, 30 and frame sizes 7, 30. We also plot the sound signal with window sizes 40, 80 and 160 and frame sizes 160 and 240. These combinations are shown below, with a single frame duration and two window durations at each page.

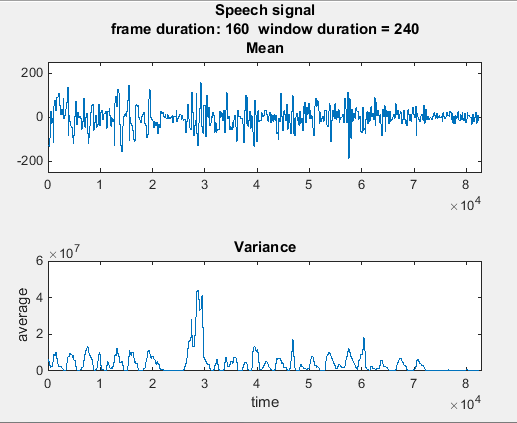


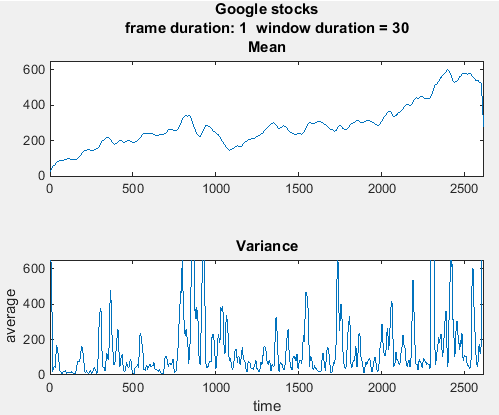
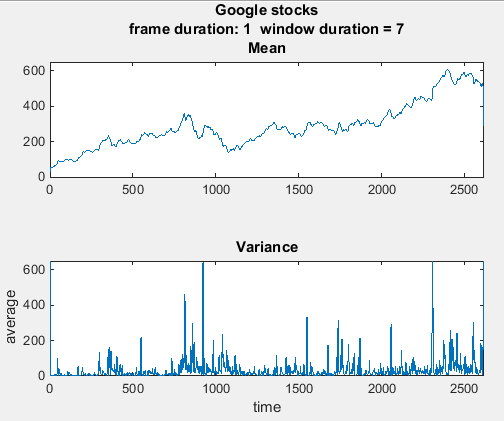


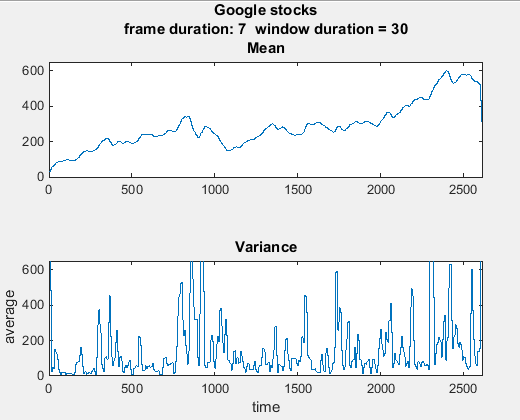
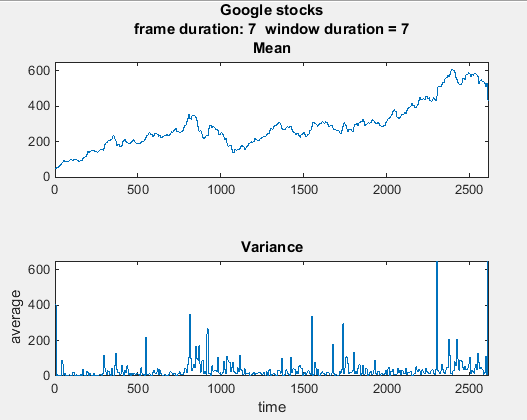


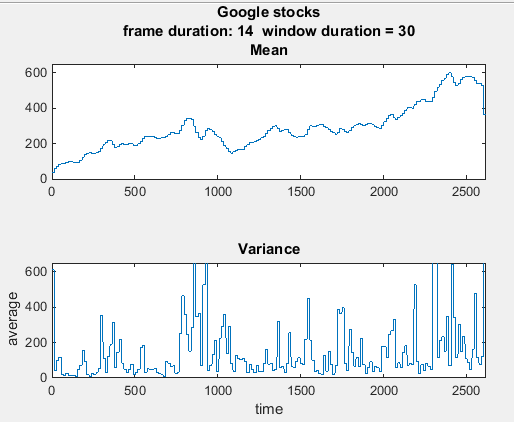
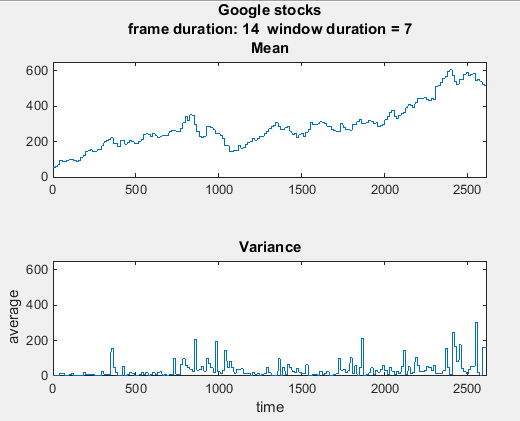


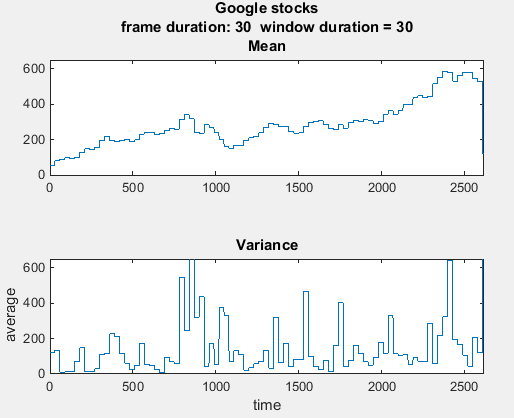
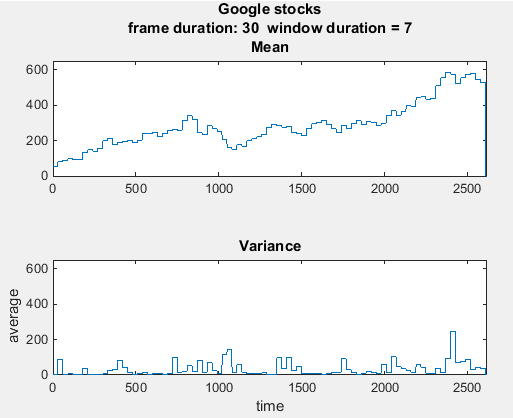












# MATLAB Code

This program plots Mean and Variance plots with specified window and frame technique. The window and frame vectors are produced using a slightly modified version of Christian Ward’s code.

function ca\_01\_part2\_v03

clc;

M\_stock = [ 1, 7, 14, 30]; % frame size of Google stock

N\_stock = [ 7, 30]; % window size of Google stock

M\_speech = [40, 80, 160]; % frame size of speech signal

N\_speech = [160, 240]; % window size of speech signal

test\_signal = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ...

19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 1 1]; % test signal

% read in google stock prices

google\_stock = xlsread('google\_v00.xlsx');

% take only the fourth column, make it a row

google\_stock = google\_stock(:,4)';

% read in sound signal

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

speech = fread(fp,inf,'int16');

fclose(fp);

% plot google stock and sound signal's average and variance values with

% specified frame and window sizes.

[google\_averages, google\_variances] = plot\_avg\_and\_var(google\_stock, M\_stock, N\_stock, 0, 2618, 0, 650, 0, 2618, 0, 650, 'Google stocks');

[sound\_averages, sound\_variances] = plot\_avg\_and\_var(speech, M\_speech, N\_speech, 0, 8.3e4, -250, 250, 0, 8.3e4, 0, 6e7, 'Speech signal');

figure(102)

plot(sound\_averages{1,1});

hold on;

plot(sound\_averages{2,1});

plot(speech);

title('frame 1');

figure(103)

plot(sound\_variances{1,1});

hold on;

plot(sound\_variances{2,1});

plot(speech);

title('frame 1');

end

% arguments to this function.

%

% signal: vector containing signal data

% frames: array of frame sizes to be plotted

% windows: array of window sizes to be plotted

% mean\_left, mean\_right, mean\_down, mean\_up: axis([]) values for mean plot

% variance\_left, variance\_right, variance\_down, variance\_up: axis([]) values for variance plot

% name: name of plot

% returned values

%

% averages: cell array containing vector of mean plots

% variances: cell array containing vector of variance plots

function [averages,variances] = plot\_avg\_and\_var(signal, frames, windows, mean\_left, mean\_right, mean\_down, mean\_up, variance\_left, variance\_right, variance\_down, variance\_up, name)

% these are all plot data output from the function.

averages = cell(length(windows),length(frames));

variances = cell(length(windows), length(frames));

for n = 1:length(windows)

for m = 1:length(frames)

[average, variance] = compute\_avg\_and\_var\_v01(signal, frames(m), windows(n));

averages{n,m} = average;

variances{n,m} = variance;

h1 = figure('name', 'average plot', 'numbertitle', 'off');

figure(h1);

str = sprintf('%s\nframe duration: %d window duration = %d\nMean', name, frames(m), windows(n));

subplot(2,1,1);

plot(average);

title(str);

axis([mean\_left mean\_right mean\_down mean\_up]);

subplot(2,1,2);

plot(variance);

title('Variance');

axis([variance\_left variance\_right variance\_down variance\_up]);

xlabel('time');

ylabel('average');

end % end "m" for loop

end % end "n" for loop

end % end "plot\_avg\_and\_var" function

# Conclusions

We can see right away from the plots above, especially in the Google stock prices plots, that the variance is far greater for larger window sizes. This is because since more points are included in the calculation, there is a greater chance that points will stray from the mean. We also see that as the frame sizes increase, we get blockier representations of the signals. This allows for a more general view of the signal.

The window and framing plots of the signals show us that the definitions of average or variance depend on what range of data you are averaging. For example, figure 3 demonstrates the difference between different window sizes with the same frame size.

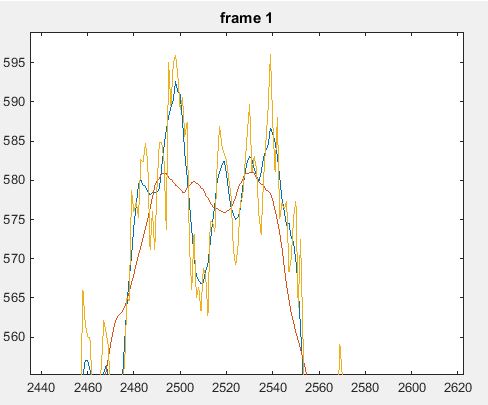


Figure 3: Google stock prices, all at 1 frame. The Yellow line is the original signal, the blue line is taken with a window of 7, and the red one is taken at a window of 30.

We see that since a larger window size computes the average over a broader range, so it is a smoother signal taking into account the peaks and valleys of the original signal. The smaller window size does not account for both peaks and valleys of the signal, so it matches the original signal far more. We see that if we wish to invest in Google stock over a long period of time, the larger window size is more valuable to us since it maps long term behavior of the signal. If we are interested in the short term stock price, for example if we wish to sell our stock as soon as possible, the smaller window size will be more useful since it will track the stock’s behavior in the short term.

These plots give us far more information about the average value of the signal than the information in table 1. We can say that the average in table 1 is a one frame, one window plot taking all data points into account. This gives far less information than fine tuning a frame and window size that fits our needs.