Tyler Castelli

ECE 3512: Signals – Continuous and Discrete

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

# Problem Statement

The purpose of this assignment was to model the data of the Google stock data and the audio signal in different ways. For the Google stock data, we were asked to calculate the mean using the window-frame analysis used in the last assignment, and use a window size of 7 and a frame size of 1. We were then asked to calculate a linear regression line. Finally, we plotted the mean, original signal, and the linear regression line onto one plot. For the audio signal, we were required to take the signal and plot a histogram of the data. The histogram has a bin width of 10 and extends from -32767 to 32767. The data in each bin was to be normalized by the total number of samples in the signal.

# Approach and Results

For part of this assignment, I took my code from the previous assignment and modified it to conform to the requirements of this assignment. The loops were taken out as they were not needed, and the plots were altered so only one plot was generated. I then looked up how to calculate a linear regression line in MATLAB, and found that I could use the polyfit() and polyval() functions to do so. I implemented these functions and came up with the following plot:

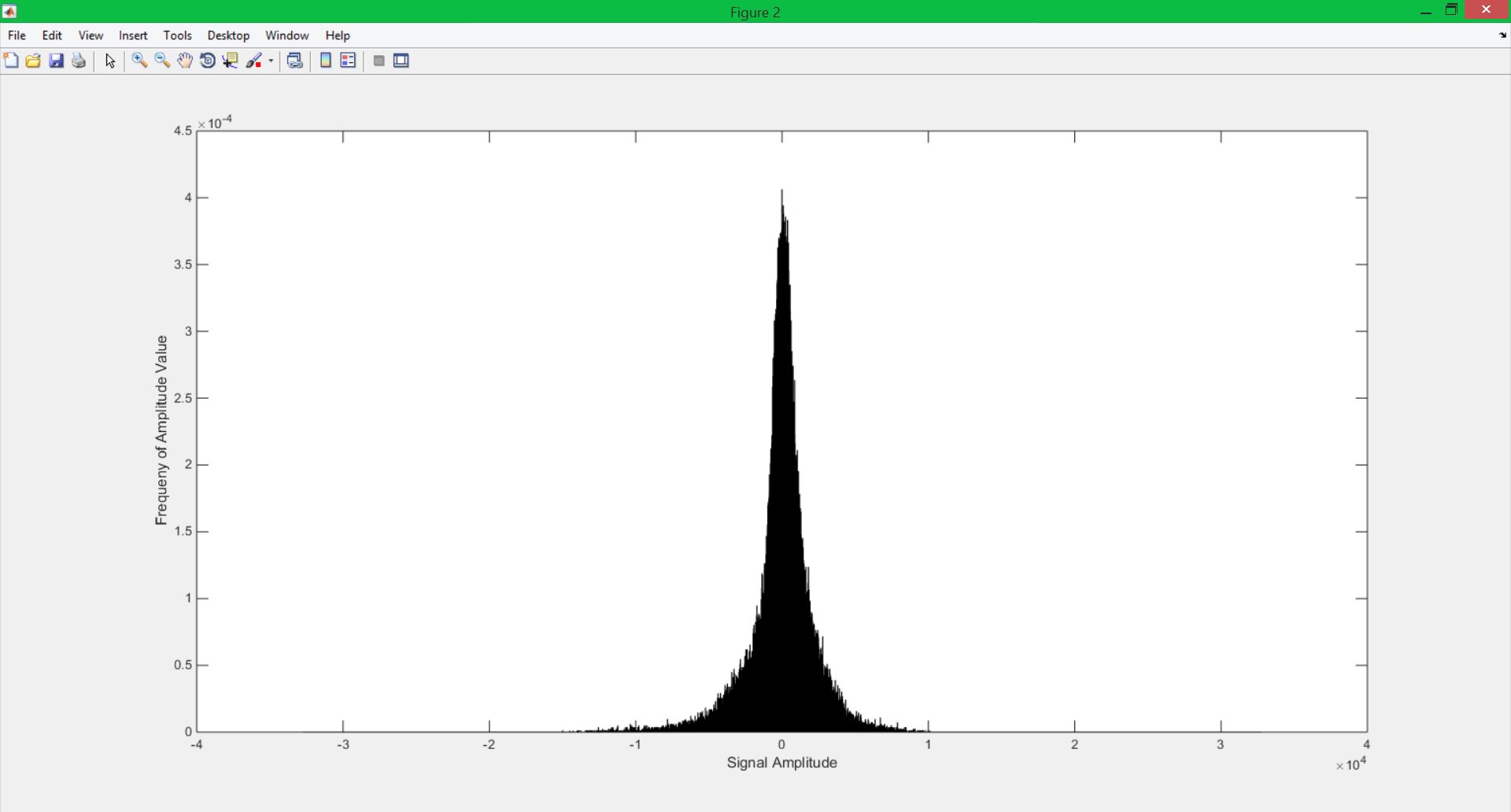
Figure 1



Because the mean value is calculated using a frame size of 1, the mean plot is very similar to the original data. If we alter the frame and/or window size, we will notice a more staircase look of the mean plot and will notice the difference between the mean and the original data, as shown in the previous assignment. By simple inspection, one can see that, despite several drops in the stock values, the data generally increases as time goes on. This is reflected when the linear regression line is plotted on top of the data. The linear regression line is a first order linear polynomial that represents the trend of the data. The slope and y-intercept of the regression line is calculated using the polyfit() and polyval() functions of MATLAB. These values are then used to generate the regression line plot.

For part 2 of this assignment, I used MATLAB’s histogram function and some of its features to get the plot to conform to the requirements of this assignment. The approach was rather simpl: I created a histogram object of the data and altered some of its local data such as the bin width, range, and normalization. This resulted in the histogram shown below in figure 2:

Figure 2



This plot shows that a majority of the signal amplitude occurs between approximately -50,000 and 50,000. These results as well as the results from part 1 will be discussed in the conclusion section.

# MATLAB Code

%Part 1: statistical analysis of Google stock prices

function castelli\_tyler\_ca02\_pt1

%close all open sessions, clear variables and clear console

clear all; clc; close all;

%open excel file and read it in. Take only the closeing values

%and place them into a vetor to use

google\_data = xlsread('google\_v00');

data\_set = google\_data(:,4);

%Vectors for the window (N) and frame (M) sizes

N = 7;

M = 1;

%Vectors to store mean and values and get the x-values for the linear

%regression line

mean\_val = zeros(1, 1, length(data\_set));

lr\_x = (0:length(data\_set)-1)';

% set up a plotting window and label it

%

mean\_str = sprintf('Mean of Google Stock prices 2004-2014');

h1 = figure('name',mean\_str,'numbertitle','off');

% call functions function to compute the mean vector

%

mean\_val(1,1,:) = get\_Mean(N,data\_set,M);

%Plot mean vector and label it

figure(h1);

str = sprintf('Frame Duration = %d Window Duration = %d', M, N);

hold on

%subplot(2,1,1);

plot(squeeze(mean\_val(1,1,:)));

title(str);

xlabel('Time: Day');

ylabel('Values');

% plot the signal on top of the mean plot

plot(data\_set);

%calculate the linear regression line.

%overlay plot on the previous two plots

linreg = polyfit(lr\_x,data\_set,1);

yfit = polyval(linreg,lr\_x);

hold on;

plot(lr\_x,yfit);

legend('Mean','Stock Value','Linear Regression line');

end

% function: get\_Mean

%

% arguments:

% data: the input signal (input)

% frame\_sz: the frame size in samples (input)

% win\_sz: the window size in samples (input)

%

% return:

% mean\_v: a vector of mean values (output)

function mean\_v = get\_Mean(win\_sz,data,frame\_sz)

%create local variables to use in function call

sig\_buf = zeros(1,win\_sz);

samples = length(data);

frames = 1+ceil(samples/frame\_sz);

mean\_v = zeros(samples,1);

%loop over entire signal

for i = 1:frames

%find center, upper boound and lower bound pointers

%these determine how to move through the signal

center = ceil((i-1)\*frame\_sz+(frame\_sz/2));

l\_bound = ceil(center-(win\_sz/2));

u\_bound = l\_bound+win\_sz-1;

%if the lower or upper bound is outside the vector, zero stuff the

%signal buffer to make sure enough samples are added

if (l\_bound < 0 || u\_bound > samples)

sig\_buf = zeros(1,win\_sz);

end

%transfer data into the buffer

for j = 1:win\_sz

index = l\_bound + (j-1);

if ((index > 0) && (index <= samples))

sig\_buf(j) = data(index);

end

end

%find the mean of the window of data

avg = mean(sig\_buf);

%put the mean vale into the mean\_v output vector.

%return the value for the mean value vector

for k = 1:frame\_sz

index = ceil((center+(k-1)-(frame\_sz/2)));

if((index > 0 ) && (index <= samples))

mean\_v(index) = avg;

end

end

end

end

**This is the MATLAB code for part 1. Here I took my assignment 1 code and altered it for this assignment. This code finds the mean value using a frame size of 1 and a window size of 7 and plotted using the first 1000 points of the signal. The original signal is then overlaid on top of the mean plot. The slope and y-intercept of the linear regression line is then found using the polyfit functions. Once those values are found, a vector to contain the y-values of the regression line is created using the polyval function. The regression line is then plotted onto the same plot as the mean and original signal plots.**

%Part 2: Histogram analysis of speech signal

clear all; close all; clc;

%Open file

f1 = fopen('rec\_01\_speech.raw','r');

sig = fread(f1,inf,'int16');

fclose(f1);

%Plot histogram and alter specifications

figure(2);

h = histogram(sig,'BinWidth',10); %change widht of the bins

h.BinLimits = [-32767 32767]; %Alter limits of historam

h.Normalization = 'pdf'; %Normalize data by total number of samples

ylabel('Frequeny of Amplitude Value');

xlabel('Signal Amplitude');

[x,y] = ginput(2);

**This is the code for part 2 of the assignment. Here I read in the data and created a histogram object of the signal. I then modified the properties of the histogram using static methods of the histogram object. The ginput function is used to find where the majority of the data is held in the histogram.**

# Conclusions

Through this assignment, I found that the linear regression can be used to estimate and predict future values of a random variable. Since the regression line shows the general trend of the data, it can be useful for predicting future values of such things as stock prices, as clearly depicted in this assignment. By analyzing the window size from which the data is taken, one can estimate how the stock price will vary given a similar time frame. Using the full length of the data, one can predict that Google stock will continue to rise over the years, and that investing in them will most likely be a wise long term investment. However by paying attention to the short term (depending on which section of values you look at), investing in Google could yield problems for stock holders. For the audio signal, the histogram shows that a majority of the signals amplitude is between approximately -50,000 and 50,000. This makes sense as the people *generally* speak in a somewhat constant volume and tone when talking. Therefore, the amplitude will oscillate between the same values for the most part, with some outliers. The bin width and normalization give a better idea of what is going on in the signal and produce a plot that shows the mean values of the signal based on each bin.