Anton LeKang

ECE 3522: Stochastic Processes in Signals and Systems

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

CA9

# Problem Statement

The objective of this computer assignment is to enhance the knowledge of signals with the use of white noise, and to observe how the variance plays a role in the signal process. This assignment will look at sinusoidal signal with the addition of a Gaussian white noise. The noise of the signal will be influenced by the noise ratio, which is in decibels. Furthermore a signal can be looked at in comparison from an autocorrelation, and the noise will change based on the input of the signal to noise ratio (SNR). Sometime the only way to make a signal less noisy is to add a filter to the signal. Filters can be added to a signal, and the output of the signal will also change based on the noise ratio. This computer assignment will use the sine wave, and Gaussian white noise function in MATLAB to observe the effects on the total signal.

# Approach and Results

Part 1: Create a function to for a sinewave plus Gaussian and look at the power of the signal

In the first part of this computer assignment the power of a signal with some Gaussian white noise is looked at. The figure below shows the values that were inputted into the sine wave. First the sine wave has a frequency 10,000Hz, last 3 seconds long, has a sample frequency of 8000, and the sample ratio is 10. When this wave is looked at and computed the power came out to be .599.

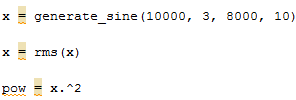
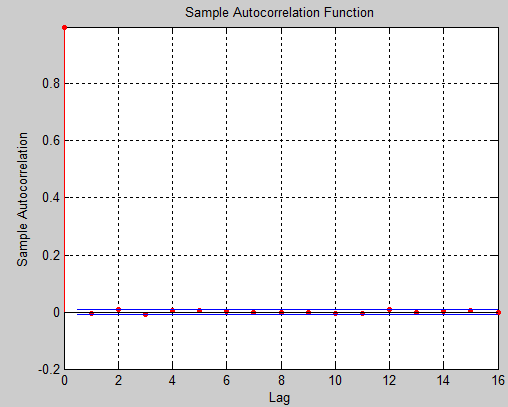
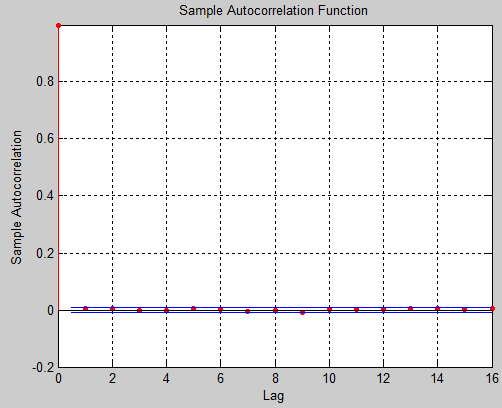


Figure 1 – Values of sinusoidal signal

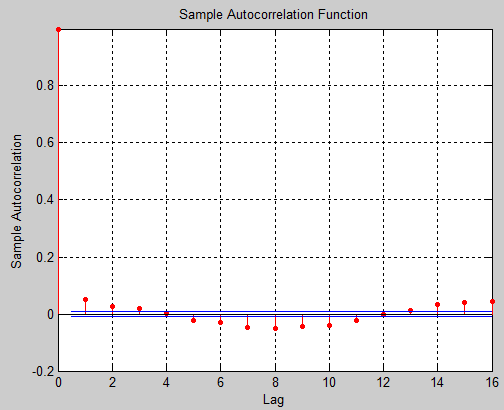
Part 2: First 16 Lags of the autocorrelation of the signal

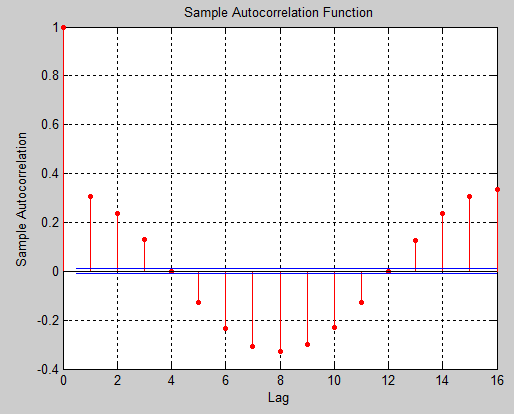
This next part of the computer assignment looked at the first 16 lags of the autocorrelation for the signal used in part 1. Also the input of the signal ratio will range from [-30 dB to 30dB] in increments of 10 dB. There will be 7 graphs for each of the dB values.

Figure 2 – First 16 Lags of signal at -30dB

Figure 3 – First 16 Lags of signal at -20dB

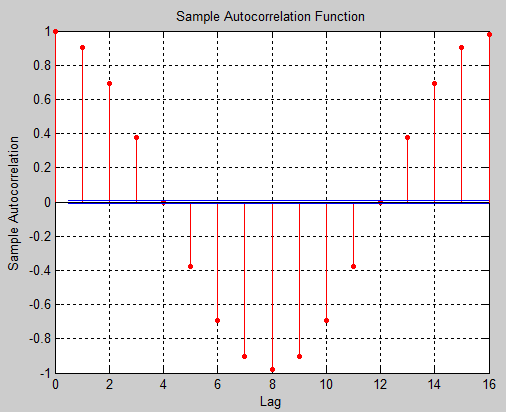
These two figures above show the autocorrelation and lag for the first two noise ratios of the signal. Figure 2 represents with the noise is set to 30dB, and figure 3 represents the noise when it is at 20dB.

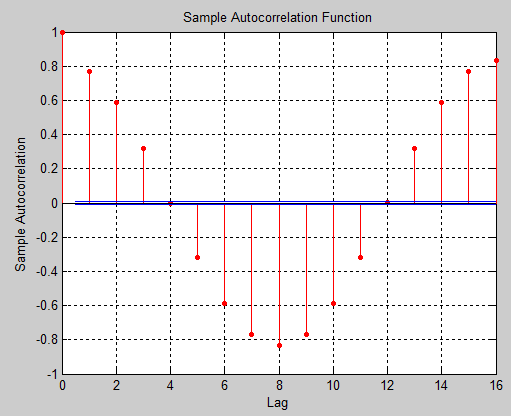
Figure 4 – First 16 Lags of signal at -10dB

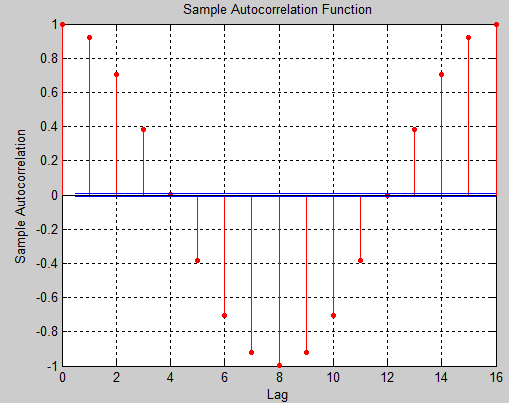
Figure 5 – First 16 Lags of signal at 0dB

These next two figures show the autocorrelation and lag for the next two noise ratio values. The figure on the left (figure 4) represents when the ratio is set to -10dB, and the figure on the right (figure 5) represents when the ratio is set to 0dB. Notice how the signal is starting to get some values of the autocorrelation.

These next two figures show the increase in noise ratio of the signal with the white Gaussian noise. The figure on the left is when the ratio is set to 10dB, and the figure on the right is when the ratio is set to 20dB. The correlation values keep on increasing in magnitude when the dB ratio increase.

Figure 7 – First 16 Lags of signal at 20dB

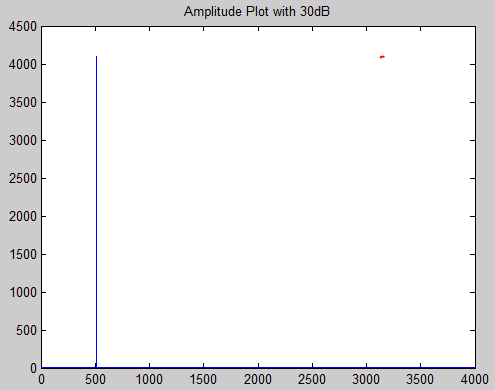
Figure 6 – First 16 Lags of signal at 10dB

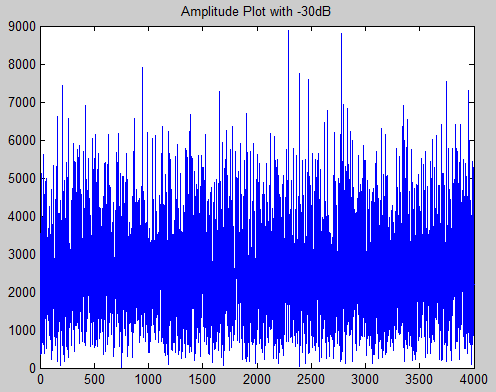
Figure 8 – First 16 Lags of signal at 30dB

This is the last plot of the autocorrelation with the ratio of the noise at 30dB. As seen in the figure, the values of the autocorrelation are a lot high than when it start at -30dB. From looking at all of these plots it seems like the largest gap in autocorrelation happens when the ratio is increased from -10dB, to 0dB. Figure 5 shows that plot, and that is where the starts to measurably impact the additive Gaussian noise.

Part 3: Comparing -30dB and 30dB with 8192 points of a FFT

This part of the computer assignment looks at the magnitude of the signal after computing the FFT of it. The signals looked at will have -30dB as the ratio, and +30dB.

Figure 10 – FFT with noise ratio at +30dB

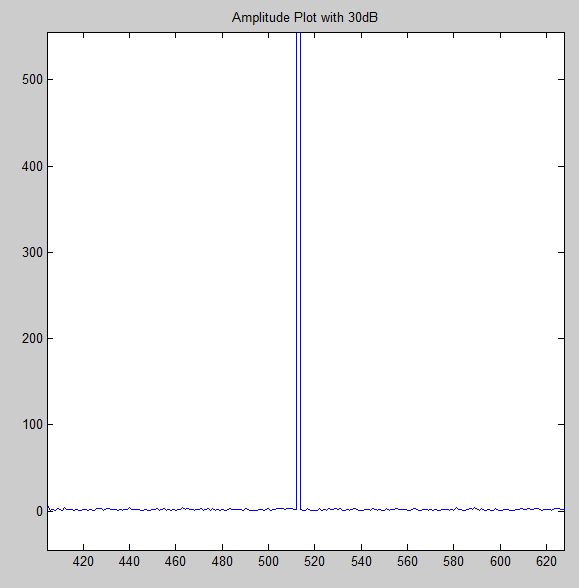
Figure 9 – FFT with noise ratio at -30dB

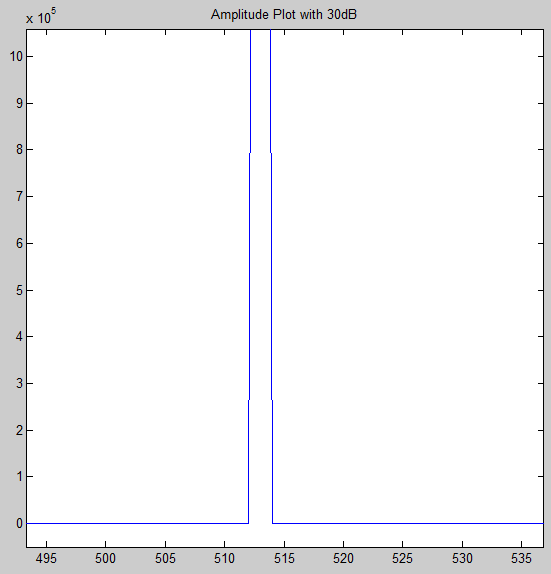
These two figures show magnitude spectrum by computing the FFT of the signals. The figure on the left is when the ratio of the noise is set to -30dB, the figure on the right is when the noise ratio is set to +30dB. The figure on the right is a much better plot of the signal, and even has the peek at 500Hz. That is because the signal is a 500Hz sine wave, so from computing the FFT, it should be at 500. The figure on the left is extremely noisy, this would not be a good representation of the signal

Part 3: Applying a filter to the signal

The next activity looks at the signal with the noise ratio set at 30dB, but then it is passed through a filter. There will be two plots looked at, first the plot of the magnitude spectrum of the Fourier Transform, and then the square of the magnitude sprectrum.

As seen from these two figure the output of the signals are different. First off the figure on the left (11) is not as smooth as the figure on the left. Also this figure is substantially smaller than the figure on the right. Figure 12 has a larger frequency response set the output of the Gaussian white noise is doubled.

Figure 11 – Filter of magnitude at 30dB

Figure 12 – Filter of magnitude squared at 30dB

# MATLAB Code

Part 1:

function CA\_9\_1

x = generate\_sine(10000, 3, 8000, 10)

x = rms(x)

pow = x.^2

end

function sig = generate\_sine(frequency\_in\_Hz, duration\_in\_secs, sample\_frequency\_in\_Hz, snr\_db)

t = 0:1/sample\_frequency\_in\_Hz:duration\_in\_secs;

%creating the sine wave

y = sin(2\*pi\*frequency\_in\_Hz\*t);

sig = awgn(y,snr\_db)

end

Part 2:

function CA\_9\_2

clear all

clc

i = 1;

for SNRs = -30:10:30

x = generate\_sine(500, 5, 8000, SNRs)

%computes the autocorrelation of the lags

figure(i)

autocorr(x,16)

i = i + 1;

end

end

function sig = generate\_sine(frequency\_in\_Hz, duration\_in\_secs, sample\_frequency\_in\_Hz, snr\_db)

t = 0:1/sample\_frequency\_in\_Hz:duration\_in\_secs;

%creating the sine wave

y = sin(2\*pi\*(frequency\_in\_Hz)\*t);

sig = awgn(y,snr\_db)

end

Part 3:

function CA\_9\_3

clear all

clc

x1 = generate\_sine(500, 5, 8000, -30);

x2 = generate\_sine(500, 5, 8000, 30);

length = 8192

tran1 = fft(x1, length);

tran1 = tran1(1:length/2-96);

tran2 = fft(x2, length);

tran2 = tran2(1:length/2-96);

frequency = 1:4000;

figure(1);

plot(frequency, abs(tran1))

title('Amplitude Plot with -30dB')

figure(2);

plot(frequency, abs(tran2))

title('Amplitude Plot with 30dB')

end

function sig = generate\_sine(frequency\_in\_Hz, duration\_in\_secs, sample\_frequency\_in\_Hz, snr\_db)

t = 0:1/sample\_frequency\_in\_Hz:duration\_in\_secs;

%creating the sine wave

y = sin(2\*pi\*(frequency\_in\_Hz)\*t);

sig = awgn(y,snr\_db);

end

Part 4:

function CA\_9\_5

x1 = generate\_sine(500, 5, 8000, 30);

b = 1;

a = [1 0.5];

y = filter(b, a, x1)

length = 8192

tran1 = fft(y, length);

tran1 = tran1(1:length/2-96);

tran2 = fft(y, length);

tran2 = tran2(1:length/2-96).^2;

frequency = 1:4000;

figure(1);

plot(frequency, abs(tran1))

title('Amplitude Plot with 30dB')

figure(2);

plot(frequency, abs(tran2))

title('Amplitude Plot with 30dB')

end

function sig = generate\_sine(frequency\_in\_Hz, duration\_in\_secs, sample\_frequency\_in\_Hz, snr\_db)

t = 0:1/sample\_frequency\_in\_Hz:duration\_in\_secs;

%creating the sine wave

y = sin(2\*pi\*(frequency\_in\_Hz)\*t);

sig = awgn(y,snr\_db);

end

These are the four scripts that were used for this computer assignment. In the first part that is mainly the creation of the sine function with the added white noise. The function is then copied and used in all of the other sections. In the second part the autocorrelation is added into the figure and plotted, this is seen from the command auttocorr(x, 16). It takes the signal x, and computes the first 16 lags of it. In the third part of this activity it is asked to plot the FFT at a length of 8192. So that is the length into the FFT function and x1/x2 is the inputted signal. The x1 represents when the ratio value is set to -30dB, and the x2 represents when the ratio is set to +30dB. The fourth scripts shows the function with a filter added to the signal. This filter is represented by b and a, and is the z transform of the y[n] filter. The FFT is then taken and then plotted. The only difference between the plots is that the second plot is the squared magnitude.

# Conclusions

This computer assignment was very valuable in expressing the importance of the noise ratio of the output of the signal. Most signals are not perfect and have noise, so this implements a real life example. In the first part of this computer assignment the power is found after adding the noise to the signal. This value seem to stay within the boundaries of .4 and 1. In the second part of this computer assignment the first 16 lags were looked at of the noisy signal. As the decibel value got larger and more positive the autocorrelation magnitude values were getting larger. The last two parts of this assignment dealt with looking at the FFT of the signal. When the ratio was -30dB the FFT did not produce a good output and the data wave very noisy. But when +30dB was passed through the FFT, there was a peak at 500 which represented the 500 Hz sine wave. In the last part of this assignment a filter was added to the signal. When the filter was applied to the FFT it had a spike at 500, but the baseline was still a little noisy. When the square of the magnitude was taken of the signal it not only got substantially larger, but the plot got smoother. The squared magnitude has a much larger frequency response since it was more linear. This computer was very informative, and showed good insight on correlation with signals that had Gaussian white noise added to it. It was also very valuable in showing the importance of the noise ratio, and how it is affected based on computing the FFT, and applying a filter to the signal.