Marc Jurchak

ECE 3522: Stochastic Processes in Signals and Systems

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

# Problem Statement

We will compute the autocorrelation function for six signals, as well as the power spectral density (PSD) of these signals. Since the PSD is the FFT of the autocorrelation, we will show plots autocorrelation and the FFT of the autocorrelation.

# Approach and Results

We begin by writing a function that implements the autocorrelation function, defined as:

Where

We will define a signal and values of N and M to loop over to get the autocorrelation function. Next we will take the FFT to see the PSD. There are six signals we will analyze.

First signal: Gaussian white noise with N = 100 and M = 20.

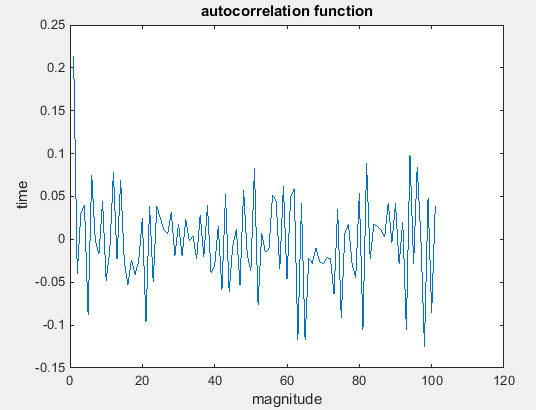


Figure : autocorrelation function of white gaussian noise

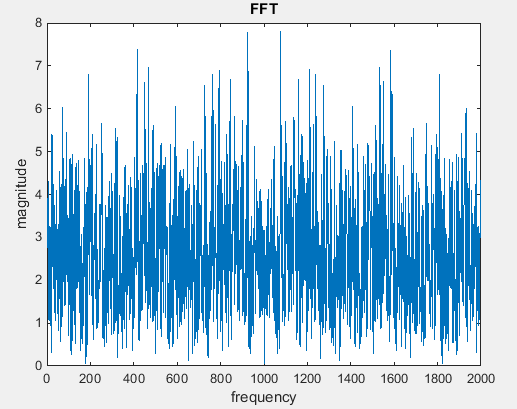


Figure : PSD of white gaussian noise

Signal 2 is an impulse function with N = 100, M = 20.

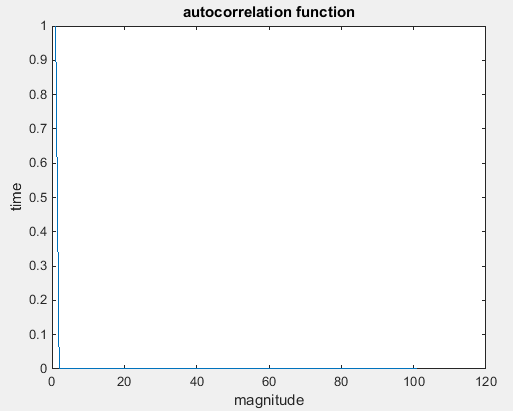


Figure : Autocorrelation function for impulse

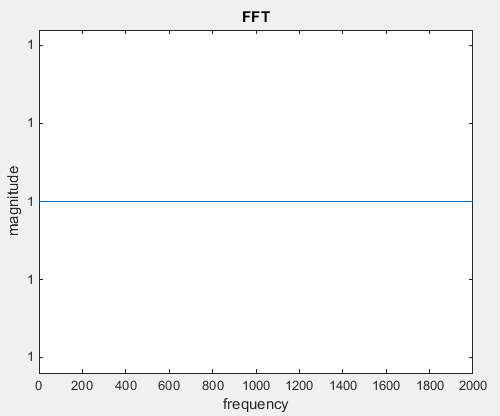


Figure : PSD of impulse function

The third signal is a periodic impulse train with a period of 20 samples with N = 200, M = 60.

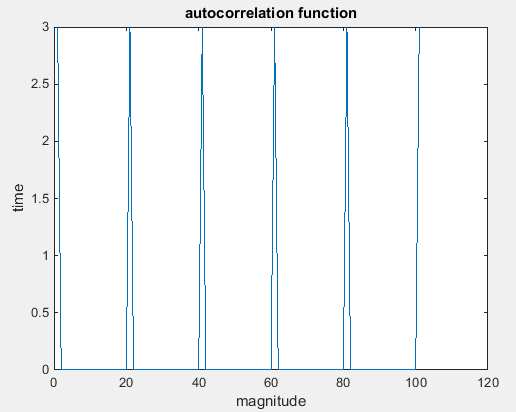


Figure : autocorrelation function of an impulse train

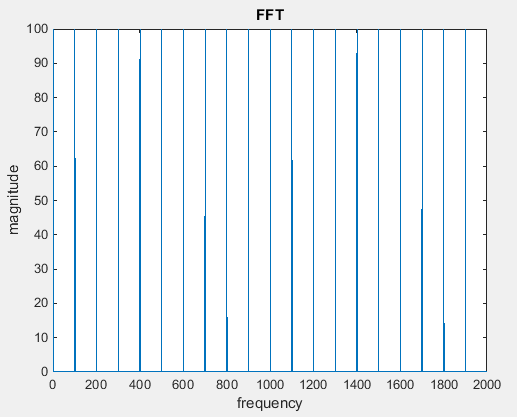


Figure : PSD of impulse train

The fourth signal is a sinewave with a period of 20 samples with N = 200 and M = 60.

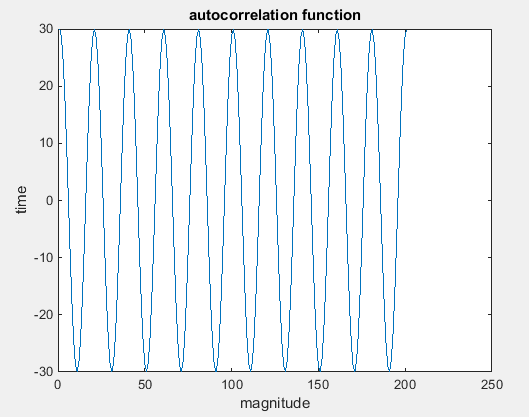


Figure : Autocorrelation of sinewave

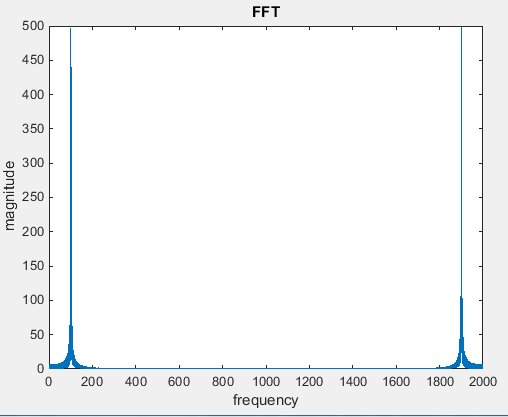


Figure : PSD of sinewave

The fifth signal is a sinewave with varying values of N: 14, 17, 20, 23, 26.

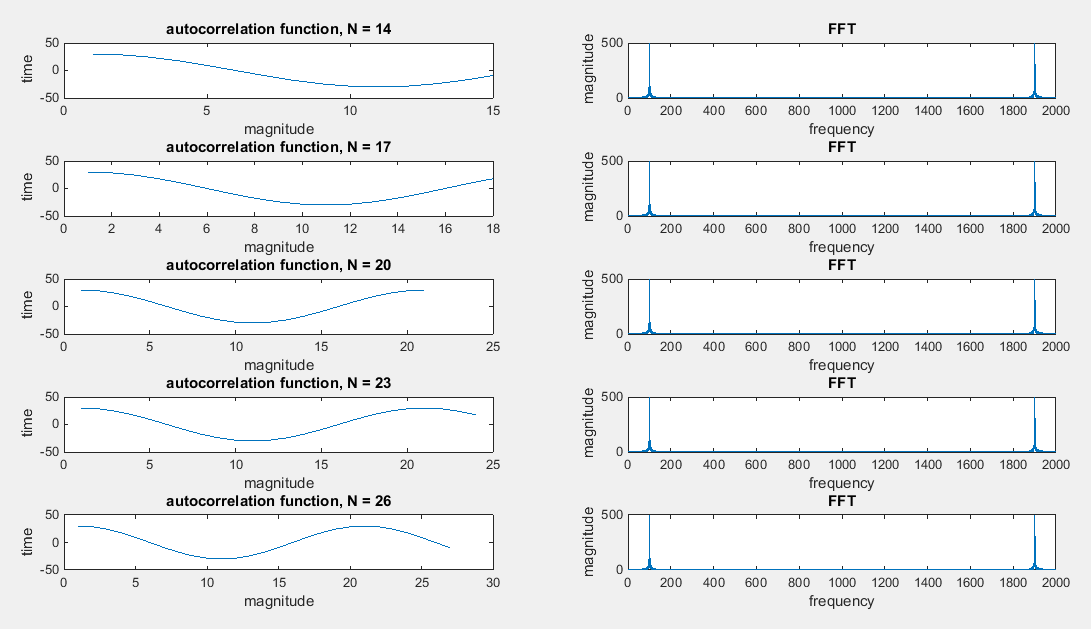


Figure : Autocorrelation and PSD of sine wave with various N

The sixth signal is the sum of a sine wave and noise with a signal to noise ratio (SNR) of 10 dB with N = 200 and M = 60.

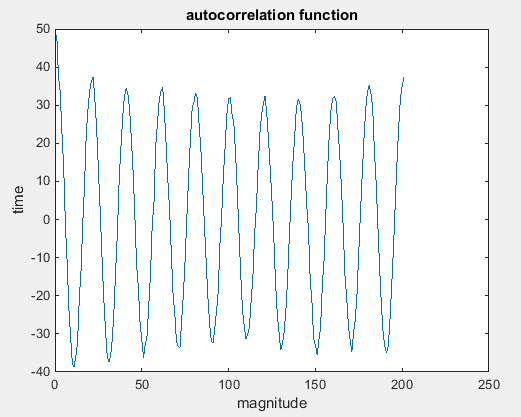


Figure : Autocorrelation of sine wave with noise

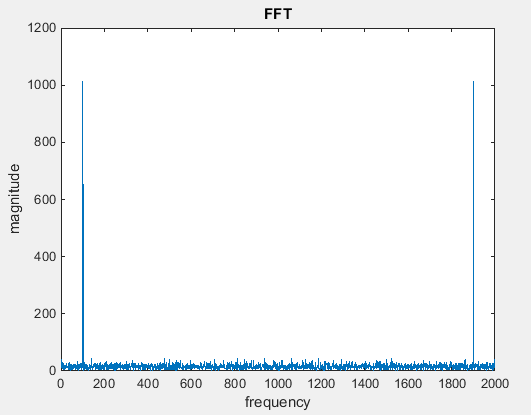


Figure : PSD of sine wave with noise

# MATLAB Code

**Autocorrelation Function**

function R = auto\_correlation(signal, N, M)

R = zeros(1, M+1);

tau = 0:1:M;

sum = 0;

start = floor(length(signal)/2);

for i = 1:(M+1)

for j = 1:N

n = start + j - 1;

sum = sum + signal(n)\*signal(n - tau(i));

end

R(i) = sum;

sum = 0;

end

end

**Signal to Noise ratio function**

function [sig\_and\_noise,snr,t] = sine\_and\_noise (freq\_Hz, dur\_sec, sample\_freq\_Hz, snr\_db )

% freq\_Hz = .5;

% dur\_sec = 2;

% sample\_freq\_Hz = 100;

% snr\_db = -30;

% time vector

t = linspace(0,dur\_sec,dur\_sec \* sample\_freq\_Hz);

% generate sine wave

sine\_wave = sin(freq\_Hz\*2\*pi\*t);

% after calculating SNR after using the function, we found

% that dividing the desired SNR by 2 yields the correct result

snr\_db\_adjusted = snr\_db / 2;

% Thus function automaticcaly adds Gaussian noise to the function

sig\_and\_noise = awgn(sine\_wave,snr\_db\_adjusted, 'measured');

% separate the noise from the signal

noise = sig\_and\_noise - sine\_wave;

% noise energy is the variance

noise\_energy = var(noise);

% sine wave energy without noise

signal\_energy = var(sine\_wave);

% calculate SNR to see if it matches the input

snr = 20\*log(signal\_energy/noise\_energy);

end

**Main Code (commented lines for signal 5)**

clear;

clc;

t = linspace(0,2\*pi,1000);

signal1 = wgn(1,1000,-20);

signal2 = [zeros(1,200),1,zeros(1,200)];

signal3 = [zeros(1,19),1];

signal3 = repmat(signal3,1,100);

signal4 = sin(50\*t);

signal\_5\_N = [14 17 20 23 26];

signal6 = sine\_and\_noise(50,2,1000,10);

signal = signal6;

N = 200;

M = 60;

%z = [1 3 5 7 9];

%y = [2 4 6 8 10];

%for i = 1:length(signal\_5\_N)

%N = signal\_5\_N(i);

%name = sprintf('autocorrelation function, N = %d',N)

auto = auto\_correlation(signal,M,N);

figure(1)

%subplot(5,2,z(i))

plot(auto)

%title(name)

title('autocorrelation function')

ylabel('time')

xlabel('magnitude')

figure(2)

%subplot(5,2,y(i))

x = fft(signal,length(t)\*2);

plot(abs(x))

title('FFT')

ylabel('magnitude')

xlabel('frequency')

%end

# Conclusions

Since we know that the Fourier transform of the autocorrelation function is the PSD, we can easily compute the PSD for a variety of signals. The PSD is important because it gives us the average power of a signal over a specific bandwidth. We can see this from figures 1 and 2, where the autocorrelation and PSD of Gaussian white noise is mostly constant. This makes sense since Gaussian white noise is usually specified in dB, i.e. a constant value of power. Figures 3 and 4 show the autocorrelation and PSD of an impulse. The autocorrelation of an impulse is another impulse, and the FFT of an impulse is a constant, as expected. The impulse also has constant power at all frequencies like the white noise. Figures 5 and 6 show the autocorrelation and PSD of an impulse train. The autocorrelation of an impulse train is another impulse train, and the FFT of an impulse train is another impulse train. We see that the power is reserved to only certain frequencies. Next figures 7 and 8 show a similar operation of a sinusoid. As we expect the autocorrelation of a sinusoid is another sinusoid, and the FFT of a sinusoid is an impulse at its frequency. Thus the sinusoid only contains power at its frequency. Figure 9 shows that varying N does not have an effect on the PSD of the plot, since the sinusoid is carrying power at its frequency, regardless of how much of the signal is computed. Figures 10 and 11 show how the signal plus noise show a combination of each signal’s respective PSD: The noise contributes to the constant noise at the bottom of figure 11, and the impulses are from the frequency of the sine wave.