Claire Durand

ECE 3522: Stochastic Processes in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19122

# Problem Statement

This assignment involved plotting the autocorrelation functions and power spectral densities of various signals and comparing them. The signals included were Gaussian white noise, an impulse function, a periodic impulse train, a sinewave, and a sinewave corrupted with Gaussian white noise.

# Approach and Results

Each signal was created in MATLAB either from scratch, like the impulse and the periodic impulse train which were implemented by assigning large values at strategic indices of an otherwise 0 matrix, or using MATLAB’s defined functions such as *wgn()* for white Gaussian noise and *sin()* for a sine wave. The native function *autocorr()* was used with the given number of lags, *M*, in order to find the autocorrelation function for each signal. The fast Fourier Transform function, *fft()*, was then used to find the power spectral density since it is known that the PSD is simply the Fourier Transform of the autocorrelation function.

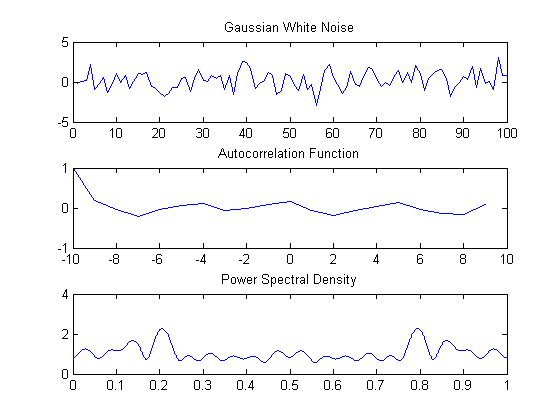


Figure 1 - Gaussian White Noise

The autocorrelation of the white Gaussian noise shown in Figure 1 hovers around 0. This is to be expected since by virtue of being noise, there should be no periodic component, and therefore no strong correlation between any two sections of the signal. However, the Power Spectral Density shows us that there still is energy in the signal which is also to be expected since the original signal is not static.

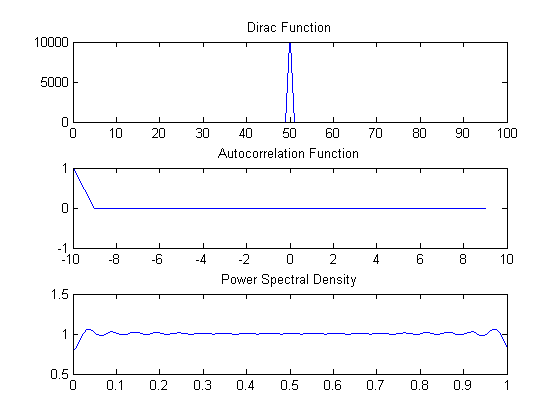


Figure 2 - Impulse Function

The autocorrelation for the impulse, shown in Figure 2, is only non-zero when there is no time shift (the x-axis is a bit off). The rest of the time, the autocorrelation is 0. This is expected since 0 is the only point at which the signal is active, therefore there cannot be any periodic component any other time. However, the PSD should look like non-zero constant as can be seen in the third subplot. This makes sense because a signal that is narrow in the time domain is wide in the frequency domain, and by taking the Fourier Transform, we have indeed transitioned into the frequency domain.

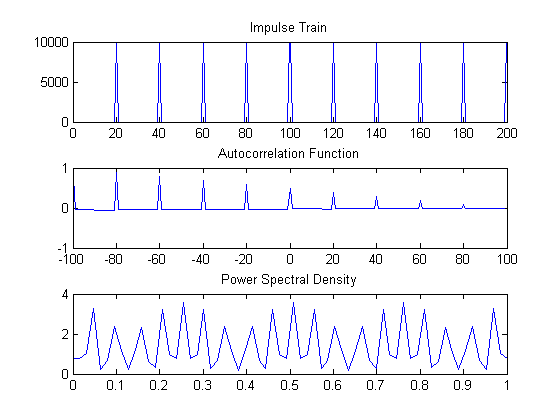


Figure 3 - Periodic Impulse Train

The autocorrelation of the periodic impulse train shown in Figure 3 is actually a little bit off. The periodic peaks in the autocorrelation should continue indefinitely with the original signal, since, at every time shift that corresponds to a period, the two signals should line up perfectly. However, in this case, the signal used to represent the impulse train in MATLAB was finite, and therefore, the autocorrelation function “ran out of signal” to compare to the original, non-time-shifted signal. Additionally, we remember that because we have a discrete signal in the time domain, we should end up with a continuous PSD since it is the Fourier transform.

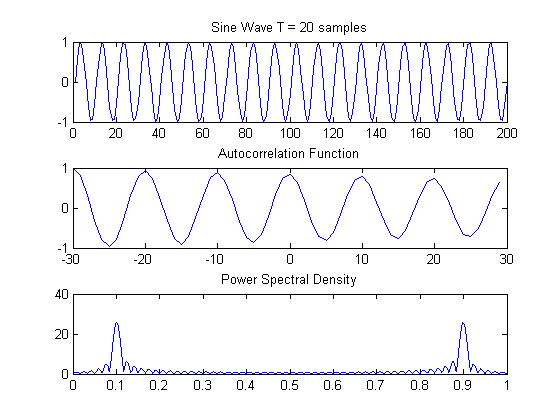


Figure 4 - Sine Wave

In Figure 4 we see our old friend the sine wave. As expected, its autocorrelation function is also a sine wave, since with the phase change the autocorrelation decreases until it reaches its lowest value at a phase shift of 180ᵒ, where the two signals are completely opposite, then starts increasing again until the signals are perfectly lined up at 360ᵒ, yielding a maximum autocorrelation of 1. The two peaks seen in the PSD plot are expected since the Fourier Transform of a sine wave is two peaks at the (positive and negative equivalent) natural frequency of the wave.

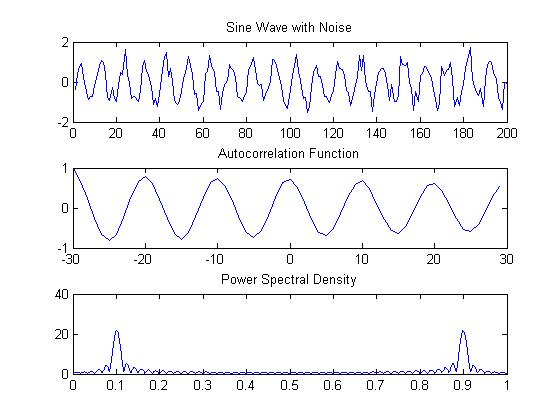


Figure 5 - Sine Wave with Gaussian White Noise

Even when a sine wave is corrupted with white Gaussian noise as shown in Figure 5, the autocorrelation function is still capable of looking past the white noise’s aperiodic components in order to pull out the sinusoidal autocorrelation of the sine wave. Once, again, the PSD is as expected from the Fourier Transform of a sine wave (note that the axis are a bit off).

# MATLAB Code

## Gaussian White Noise

N1 = 100; M1 = 20;  
GaussWN = wgn(1,N1,0);  
  
R1 = autocorr(GaussWN, M1-1);  
  
x1 = linspace(-(M1/2),((M1/2)-1), M1);  
  
NFFT1 = 2^nextpow2(N1);  
PSD1 = abs(fft(R1,8192)); % power of 2 as samples --> 8192  
f1 = linspace(0,1,8192); %  
  
figure(1)  
subplot(3,1,1)  
plot(GaussWN)  
title('Gaussian White Noise')  
subplot(3,1,2)  
plot(x1,R1)  
title('Autocorrelation Function')  
subplot(3,1,3)  
plot(f1,PSD1)  
title('Power Spectral Density')

## Make impulse function

N2 = 100; M2 = 20;  
dirac = zeros(1,N2);  
for i=1:N2  
 if(i==(N2/2))  
 dirac(i) = 10000;  
 end  
end  
  
R2 = autocorr(dirac, M2-1);  
x2 = linspace(-(M2/2),((M2/2)-1), M2);  
  
NFFT2 = 2^nextpow2(N2);  
PSD2 = abs(fft(R2,NFFT2)); % use next largest power of 2  
f2 = linspace(0,1,NFFT2); %  
  
figure(2)  
subplot(3,1,1)  
plot(dirac)  
title('Dirac Function')  
subplot(3,1,2)  
plot(x2,R2)  
title('Autocorrelation Function')  
subplot(3,1,3)  
plot(f2,PSD2)  
title ('Power Spectral Density')

## Make impulse train

M3 = 200; N3 = 60;  
train = zeros(1,M3);  
for i=1:M3  
 if((mod(i,20)==0))  
 train(i) = 10000;  
 end  
end  
  
x3 = linspace(-(M3/2),((M3/2)-1), M3);  
R3 = autocorr(train, M3-1);  
  
NFFT3 = 2^nextpow2(N3);  
PSD3 = abs(fft(R3,NFFT3));  
f3 = linspace(0,1,NFFT3); %  
  
figure(3)  
subplot(3,1,1)  
plot(train)  
title 'Impulse Train'  
subplot(3,1,2)  
plot(x3,R3)  
title 'Autocorrelation Function'  
subplot(3,1,3)  
plot(f3,PSD3)  
title 'Power Spectral Density'

## Sine wave

N4 = 200; M4 = 60;  
%fs = 20000;  
f\_sine = 20; % frequency of input sine wave  
  
t = linspace(0,1,200);  
sine = sin(2\*pi\*f\_sine\*t); % change this guy to change frequencies  
  
x4 = linspace(-(M4/2),((M4/2)-1), M4);  
R4 = autocorr(sine, M4-1);  
  
NFFT4 = 2^nextpow2(N4);  
PSD4 = abs(fft(R4,4096));  
f4 = linspace(0,1,4096); % out front : Fs/2\*  
  
figure(4)  
subplot(3,1,1)  
plot(sine)  
title 'Sine Wave T = 20 samples'  
subplot(3,1,2)  
plot(x4,R4)  
title 'Autocorrelation Function'  
subplot(3,1,3)  
plot(f4,PSD4)  
title 'Power Spectral Density'

%% Noisy Sine Wave

N4 = 200; M4 = 60;

f\_sine = 20; % frequency of input sine wave

t = linspace(0,1,200);

sine = sin(2\*pi\*f\_sine\*t); % change this guy to change frequencies

GaussWN = wgn(1,N4,0);

for i=1:N4

noisy\_sine(i) = sine(i) + (GaussWN(i))\*(1/3);

end

x4 = linspace(-(M4/2),((M4/2)-1), M4);

R5 = autocorr(noisy\_sine, M4-1);

PSD5 = abs(fft(R5,4096));

f5 = linspace(0,1,4096); % out front : Fs/2\*

figure(6)

subplot(3,1,1)

plot(noisy\_sine)

title 'Sine Wave with Noise'

subplot(3,1,2)

plot(x4,R5)

title 'Autocorrelation Function'

subplot(3,1,3)

plot(f5,PSD5)

title 'Power Spectral Density'

[*Published with MATLAB® R2014a*](http://www.mathworks.com/products/matlab)

# Conclusions

The Power Spectral Density is an easy tool to use since it relates to the autocorrelation function as its Fourier Transform. Using knowledge attained in ECE 3512, its form can be anticipated just by looking at the autocorrelation function. The shape of an autocorrelation function is also pretty intuitive when considering the original signal. These two manipulations can be powerful tools in that they can easily pick out the periodic components of a signal that are not obvious to the naked eye, such as in a sine wave heavily corrupted with Gaussian white noise.