Edward Moore
ECE 3512: Stochastic Processes in Signals and Systems
Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19121

I. PROBLEM STATEMENT

The assignment explores the effect of white noise being introduced into a sine wave signal and how it effects power, autocorrelation and power spectral density. This is done by creating a function to generate a sinewave with a signal to noise ratio that is input in dB. From here, various calculations can be performed to obtain the information we require. It would prove effective to make as many separate functions as possible to perform these calculations for improved code readability and future modification/ design reuse.

II. APPROACH AND RESULTS

A function was first created to take in parameters for frequency, time, sampling frequency as well as signal to noise ratio. This function generated a sine wave with some ratio of signal to noise based on the input ratio value. Frequency spectrum outputs of this function are shown below in figure 1 where the inputs were frequency of 500Hz, sampling frequency of 8000Hz, time of 0.05sec and signal to noise ratios of -30dB and 30dB. The difference is apparent, the higher the signal to noise ratio, the more bandlimited the output frequency spectrum is. This is illustrated in figure 1.

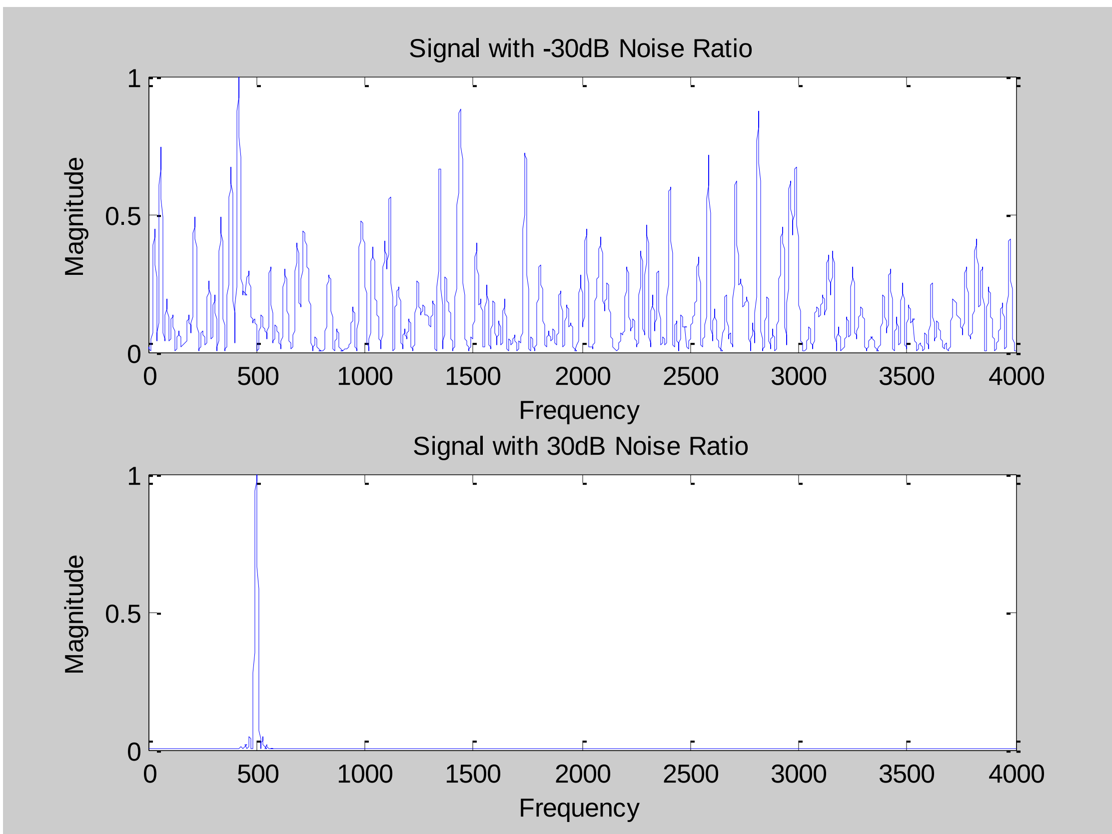


Figure 1

Another function was then created to take in a signal and number of lags over which the autocorrelation of the signal was to be calculated. This was used to plot the autocorrelation of the signal to itself over the range of -30 to 30dB for signal to noise ratio in steps of 10dB. Simultaneously, the FFT was calculated using a user defined myFFT function that takes input parameters including signal, nfft, and sampling frequency. The output of these two functions are shown in figures 2 and 3.

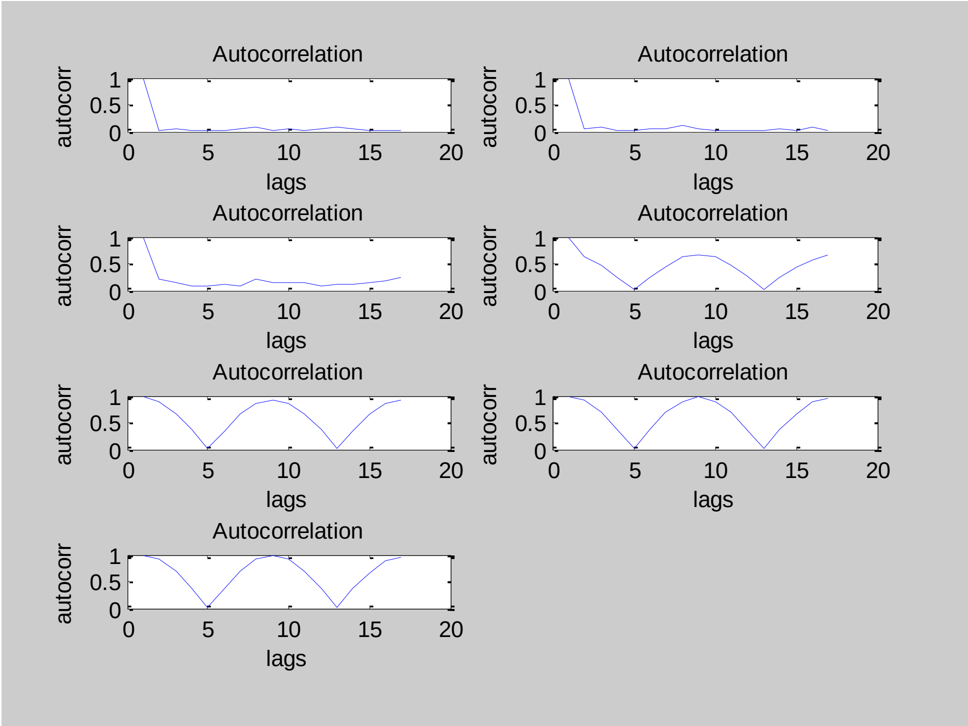


Figure 2

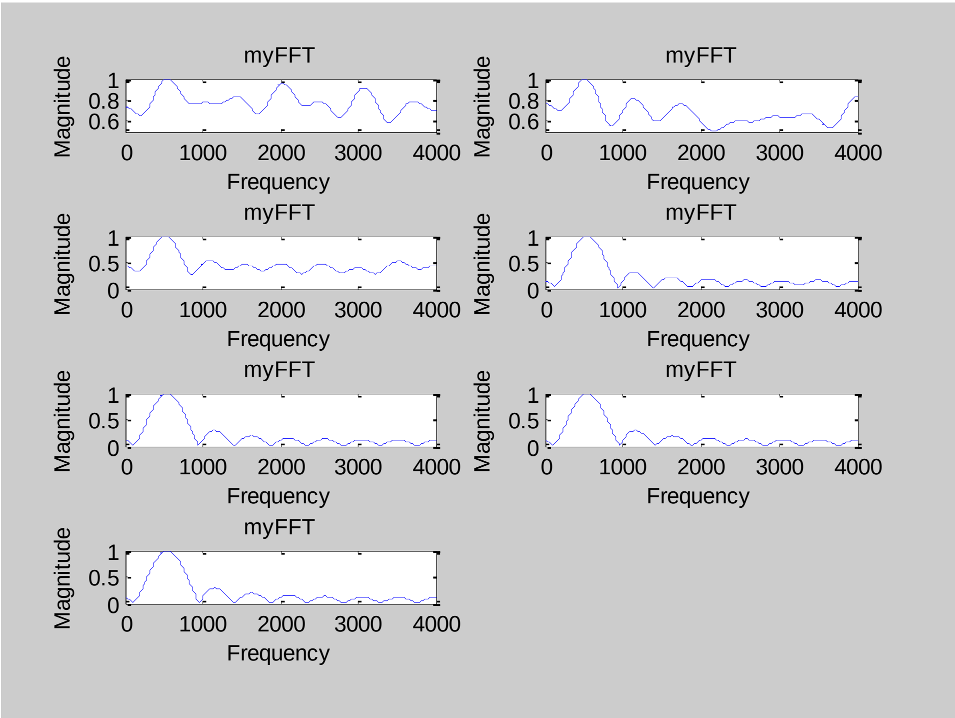
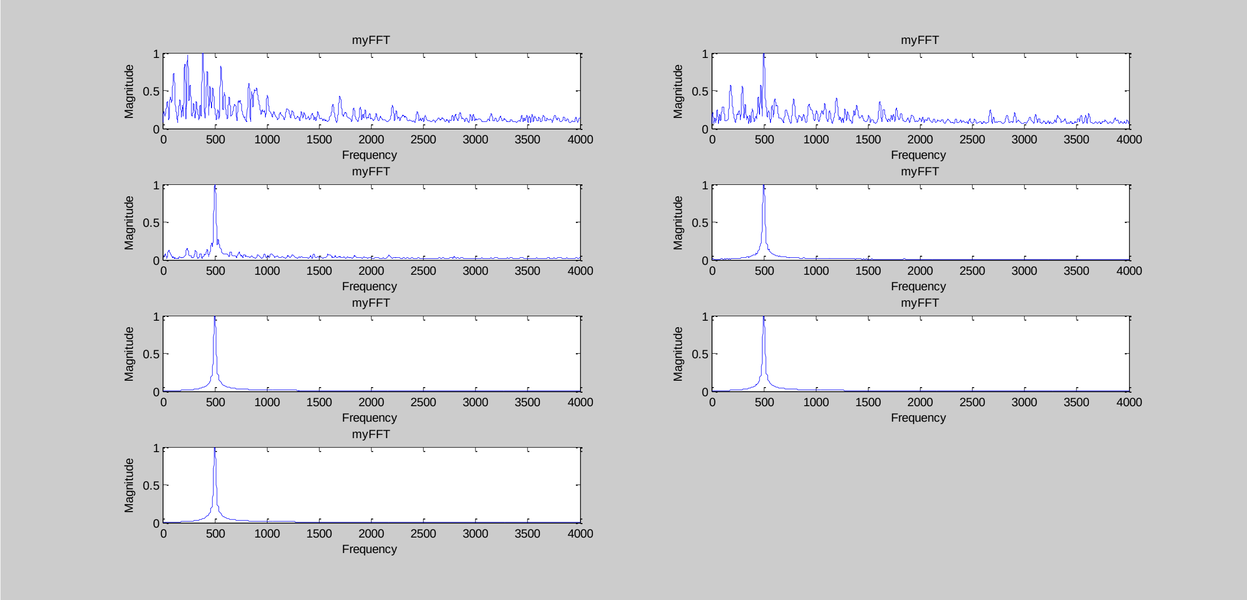


Figure 3

Finally, the frequency spectrum of the autocorrelation of a digitally filtered sine wave with noise was plotted. This was done using the functions previously described. The digital filter was coded according to the following equation: $y[n] = 0.5y[n-1] + x[n]$. The result of this calculation being plotted over the same interval as was previously implemented is shown below in figure 4.



III. MATLAB CODE

Main:

```
f = 500;
t = .05;
fs = 8000;
nfft = 8192;
snr_db = -40;
for k = 1:7
    x = generate_sine(f, t, fs, snr_db+k*10);     %+k*10
    y = x;
    for i = 2:length(x)    %loop used for digital feedback filter
        y(i) = .5*y(i-1) + x(i);
    end
    yfft = fft(x, nfft);
    yfft = yfft.^2;
    array = linspace(0, 1, nfft/2+1);
    f = fs/2*array;
    norm_f = abs(yfft(1:nfft/2+1))/(max(abs(yfft(1:nfft/2+1))));

    figure(1)
    subplot(4, 2, k)
    plot(f, norm_f);
    title('Signal with Noise')
    xlabel('Frequency')
    ylabel('Magnitude')
```

```
    figure(2)
    subplot(4, 2, k)
    lags = length(y) - 1;
    R = myautocorr(y, 16);
    R = R/max(R);

    figure(3)
    subplot(4, 2, k)
    S = myFFT(R, nfft, fs); %since fft of autocorr is pwr spec density
end
```

Generate Sine with Noise:

```
function [sig] = generate_sine(f, t, fs, snr_db)
    samp_period = 1/fs;
    t_array = [0:samp_period:t];
    sine_wave = sin(2*pi*f*t_array);
    pwr_sin = 1/2;
    pwr_noise = pwr_sin/db2mag(snr_db);

    noise1 = rand(length(sine_wave),1);
    noise2 = rand(length(sine_wave), 1);
    noise12 = sqrt(-2*log(noise1)).*cos(2*pi*noise2);
    noise = pwr_noise*noise12;

    full_signal = sine_wave' + noise;
    sig = full_signal/max(full_signal);
```

myFFT:

```
function [freq] = myFFT(x, nfft, fs)
    y = fft(x, nfft);
    array = linspace(0, 1, nfft/2+1);
    freq = fs/2 * array;
    nf = abs(y(1:nfft/2+1))/max(abs(y(1:nfft/2+1)));
    plot(freq, nf);
    title('myFFT')
    xlabel('Frequency')
    ylabel('Magnitude')
```

myautocorr:

```
function [acorr] = myautocorr(signal, lags)
    acorr = autocorr(signal, lags);
    plot(abs(acorr))
    title('Autocorrelation')
    xlabel('lags')
    ylabel('autocorr')
```

IV. CONCLUSIONS

A signal to noise ratio of -30dB makes it almost impossible to determine what the frequency of the underlying sine wave was. However, by approximately -10dB it was obvious what the frequency of the sine wave was. As expected, when computing the autocorrelation of high noise to signal ratio waves, the plot of autocorrelation showed how uncorrelated the signal was with itself. Conversely, as noise was removed the signal showed a much stronger autocorrelation. The absolute value of the autocorrelation was used when plotting. The peaks represent when the autocorrelation calculation used two samples completely in phase or 180˚ out of phase. Valleys of the autocorrelation, on the other hand, represent when the calculation used two samples either 90˚ or 270˚ out of phase. Interestingly, the FFT of the digitally filtered plot didn’t show the harmonics of the system nearly as smoothly and appeared much noisier.