Simon Honigmann

Sensor Orientation

September 30, 2018

Lab 1: Stochastic Processes

Part A: Noise Fabrication

The first step of this laboratory exercise was to create pseudo-random, computer-generated, white noise sequences. Three different seeds were used to generate different random sequences, each 200,000 values long. Integrating these sequences yielded 3 different random walks. Additionally, Gauss-Markov processes were defined for each seed, with correlation times of 2000 and 500 steps, respectively. All 4 types of noise are shown in Figure 1 below.

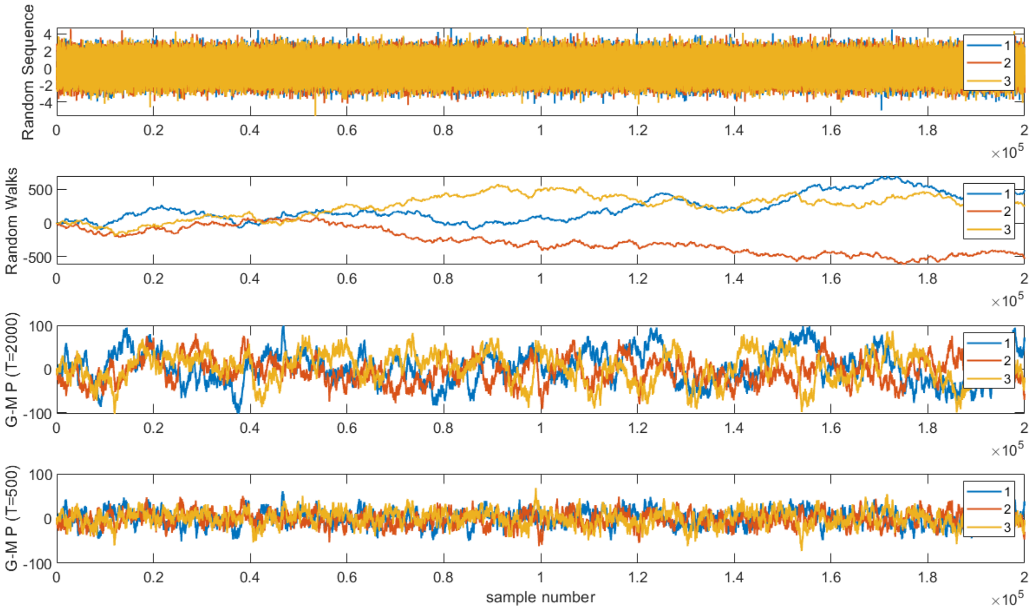


Figure 1: visualization of different noise types, demonstrating from top to bottom: white noise, random walk, and first order Gauss-Markov processes with correlation times of 2000 and 500 steps.

Part B: Noise Characterization

For each of the 4 types of sequence above, the biased autocorrelation function and power spectral density were computed and graphed using Matlab. The results are shown in Figures 2 and 3 on the following page. Finally, the data was exported into the Generalized Method of Wavelet Moments (GMWM) software to compare the generated processes with estimates based on known models. These graphs are subsequently shown in Figures 4-7 and are discussed in the questions section of this report.

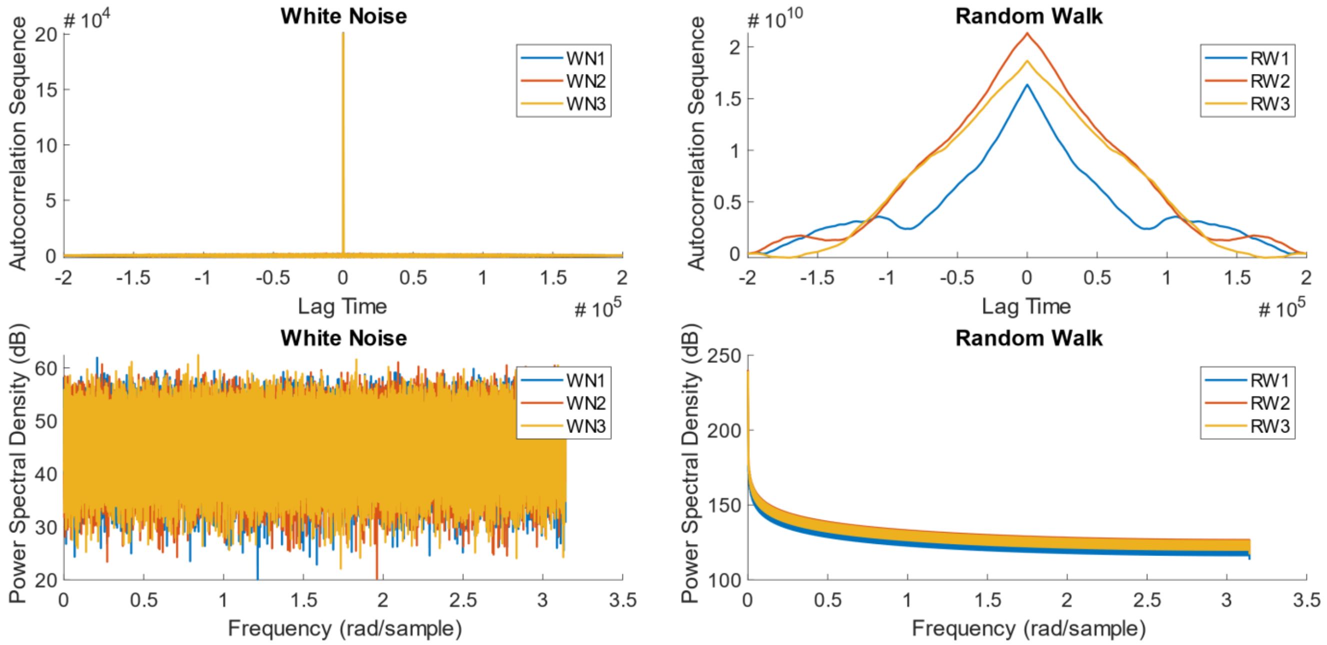


Figure 2: autocorrelation function and power-spectral-density shown for white noise (left) and random walk (right).

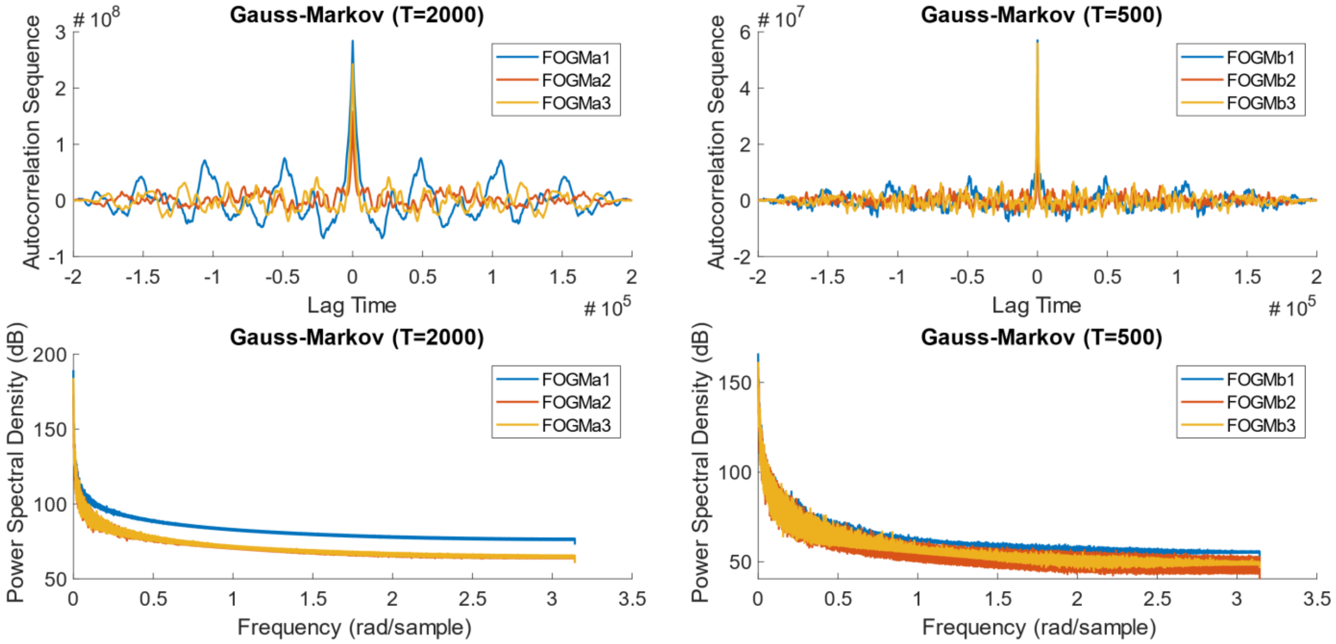
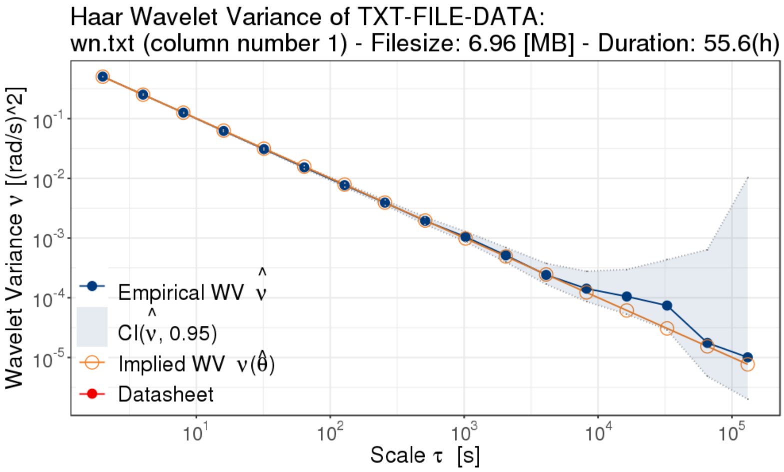
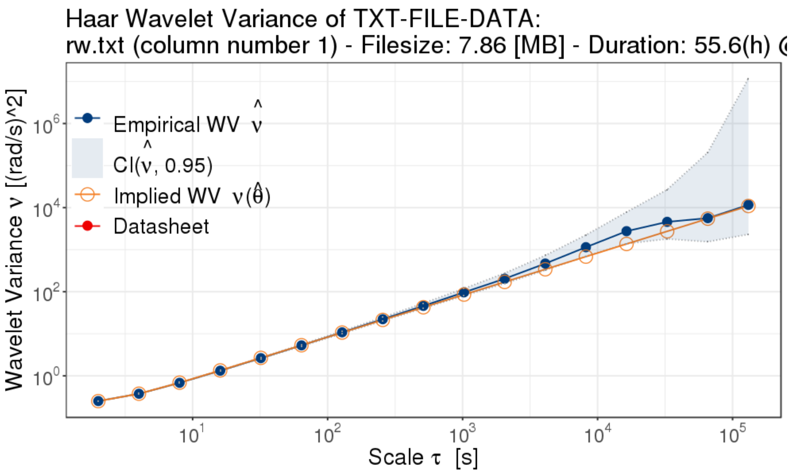


Figure 3: autocorrelation function and power-spectral-density shown for first order Gauss-Markov processes for correlation times of 2000 (left) and 500 steps (right).



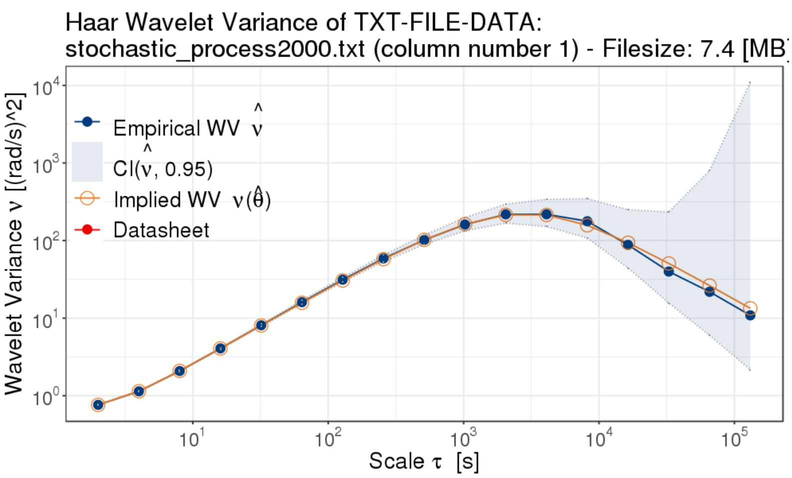

```
Error: An error has occurred. Check your logs or contact the a
pp author for clarification.
```

Figure 4: GMWM noise characterization and fit estimate for white noise




```
Error: An error has occurred. Check your logs or contact the a
pp author for clarification.
```

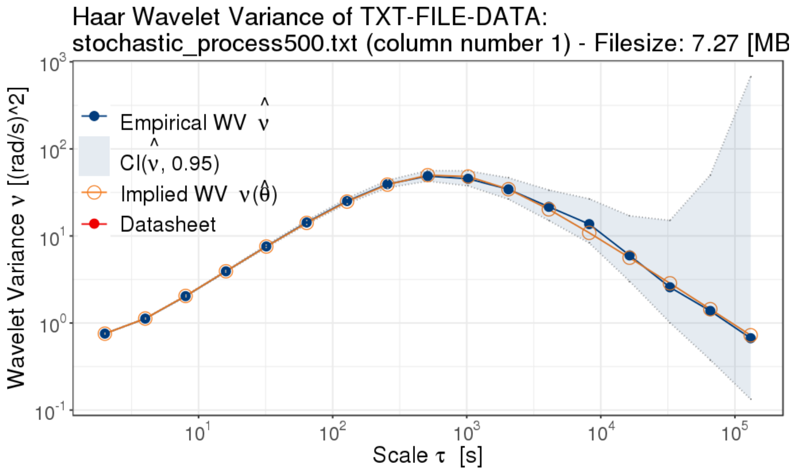
Figure 5: GMWM noise characterization and fit estimate for a random walk sequence. The parameter summary was unavailable due to a consistent GMWM error.




```
Objective Function:  0.112379
Seed Number:  1982

                Estimates       CI Low       CI High
BETA         2.026808e-07  2.234609e-07 1.819006e-07
SIGMA2_GM    5.477694e-10 -1.794304e+00 2.204264e+00
BETA         1.307752e-03  1.434940e-03 1.180581e-03
SIGMA2_GM    1.167217e+03  1.058655e+03 1.299150e+03
```

Figure 6: GMWM noise characterization and fit estimate for a 1st order Gauss-Markov Process with a correlation time of 2000 samples




```
Objective Function:  0.1229229
Seed Number:  1982

               Estimates        CI Low      CI High
BETA       5.636618e-03   0.005985622 5.287735e-03
SIGMA2_GM 2.682403e+02 241.757306984 2.982038e+02
BETA       4.151795e-02   0.041517946 4.151795e-02
SIGMA2_GM 2.786638e-15  -1.655072817 1.655073e+00
```

Figure 7: GMWM noise characterization and fit estimate for a 1st order Gauss-Markov Process with a correlation time of 500 samples

Part 3: Questions

1. Does the shape of the empirically determined autocorrelation function correspond well to the theoretical ones in all cases?

The empirically determined autocorrelation function and power-spectral-density function shapes correspond well to the theoretical shapes in all cases. For instance, the empirically determined plots for white noise match nearly perfectly with the theoretical case. As anticipated, the autocorrelation function revealed an impulse function at τ=0 and zero correlation at all other times. Unlike the theoretical case, there also appear to be impulse values at both extreme ends of the data set, but this is do to the finite nature of the sequence. The random walk function also appeared to match expectations with a prominent *mountain-like* shape. The random walk is equivalent to a Gauss-Markov process with infinite correlation time. Indeed, we see that the Gauss-Markov process with a correlation time of 2000 approaches the random walk ACF response, whereas the Gauss-Markov process with a correlation time of only 500 is closer to the white-noise autocorrelation plot.

2. How do the empirically determined values of standard deviation (from all realizations) and correlation length (derived from the plot) deviate from the true values (i.e. those in simulation)?

Note: I have no idea what is desired by this question. The lab document made no prior mention to standard deviation, and my lack of background in signal processing is leaving me a bit perplexed. As a lowly mechanical engineer by trade, I will make a big assumption and guess that it is as simple as computing the standard deviation of each process. The table below includes these values.

Table 1: Noise Signal Standard Deviation

	Process Standard Deviation			
Process	***Seed 1***	***Seed 2***	***Seed 3***	***Average***
White Noise	*1.00*	*1.00*	*1.00*	*1.00*
Random Walk	*188.9*	*186.8*	*171.2*	*182.3*
GM (T=2000)	*37.5*	*27.7*	*34.7*	*33.3*
GM (T=500)	*16.8*	*15.4*	*16.7*	*16.3*

The Scientist and Engineer's Guide to Digital Signal Processing (DSPGuide.com) mentions comparing the standard deviation to the peak-peak signal value. These values are included in the table below.

Table 2: Peak-Peak Signal as a Function of Standard Deviation

	Process Peak-Peak Magnitude as a Function of Std. Dev. $V_{pp} = n\sigma$			
Process	***Seed 1*** (σ)	***Seed 2*** (σ)	***Seed 3*** (σ)	***Average*** (σ)
White Noise	*9.6*	*8.9*	*10.4*	*9.6*
Random Walk	*4.3*	*3.8*	*4.5*	*4.2*
GM (T=2000)	*5.4*	*6.3*	*5.5*	*5.7*
GM (T=500)	*6.7*	*7.3*	*8.5*	*7.5*

According to DSPGuide.com, white noise typically has a peak-peak magnitude that is 6-8 standard deviations. This is close to the Matlab simulated result of 9.6 standard deviations.

One final attempt at answering this question.

Table 3: Standard Deviation Calculation from Theoretical Equations

		Standard Deviation			
Process	***Equation(s)***	***Seed 1***	***Seed 2***	***Seed 3***	***Average***
White Noise	$\Phi_{xx}(\omega) = \sigma^2$ $\sigma = \sqrt{\Phi_{xx}(\omega)}$	7.87	7.81	7.90	7.86
Random Walk	$\phi_{xx}(\omega) = \sigma^2 e^{-\beta\lvert\tau\rvert}$ $\phi_{xx}(0) = \sigma^2$ $\sigma = \sqrt{\phi_{xx}(0)}$	10.11	10.16	10.13	10.13
GM (T=2000)		9.19	9.05	9.16	9.14
GM (T=500)		8.81	8.76	8.80	8.79

Similarly, for the non-white-noise processes, the correlation time can be calculated using the equation:

		Calculated Correlation Time			
Process	***Equation(s)***	***Seed 1***	***Seed 2***	***Seed 3***	***Average***
Random Walk	$\Phi_{xx}(0) = \frac{2\sigma^2}{\beta}$ $= 2\sigma^2 t_{correlation}$ $t_{correlation} = \frac{\Phi_{xx}(0)}{2\sigma^2}$	1.16	1.16	1.17	1.16
GM (T=2000)		1.12	1.09	1.09	1.10
GM (T=500)		1.07	1.03	1.04	1.05

Neither the standard deviation calculations nor the correlation times appear to match expectations. The manually input correlation times of 2000 and 500 are much greater than the calculated times of 1.1 and 1.05. The standard deviation calculated directly from the data for each type of process is also off by nearly an order of magnitude for most of the process types.

3. Which computed characteristics identified best the underlying process?

The online Haar wavelet variance noise characterization tool appeared to be the most reliable in identifying the underlying process of the seemingly random noise sequences. From the characteristics computed in Matlab, the shape of the autocorrelation plots was the most immediately recognizable for me. Numerically, perhaps standard deviation would be the next best, though corrections to the calculation methodology would probably be required.

4. How do the real values compare to those estimated by GMWM?

The processes graphs estimated by GMWM come remarkably close to the real values generated by Matlab. However, the returned values of correlation time and variance are quite different than the real values. Assuming the correlation time constant associated with the larger standard deviation is the primary correlation time to be compared: for a real correlation time of 2000 samples, GMWM estimated a correlation time of $\frac{1}{\beta} = \frac{1}{0.0013077}$ =765 samples. Similarly, for a real correlation time of 500 samples, GMWM estimated a correlation time of $\frac{1}{\beta} = \frac{1}{0.0056366}$ =177 samples. Both of these values are within an order of magnitude, and perhaps go to indicate the level of precision to be expected with this sort of signal processing. The standard deviations on the other hand, match very closely. For T=2000, the standard deviation estimated by GMWM was 34.2, compared to an actual standard deviation computed by Matlab of 33.3. Similarly, for T=500, GMWM yielded a standard deviation of 16.4 compared to an actual value of 16.3. These values indicate that standard deviation is an excellent metric for characterizing noise.

Appendix A: Matlab Code

Main Function

```
%%%% Lab 1
%1) Generate 3 random sequences (white noise) -
calculations in Random Walk
%function
%2) Generate random walks using sequences
%sequences are denoted s, walks are denoted rw
[s1,rw1] = randomWalk(200000,1);
[s2,rw2] = randomWalk(200000,2);
[s3,rw3] = randomWalk(200000,3);
%for plotting
x = 1:200000;

%3) Generate 1st order guass-markov process for
two correlation times
dt = 1;

%a) tau = 2000
t=2000;
fogm1_1 = firstOrderGaussMarkov(t,dt,s1);
fogm2_1 = firstOrderGaussMarkov(t,dt,s2);
fogm3_1 = firstOrderGaussMarkov(t,dt,s3);

%b) tau = 500
t=500;
fogm1_2 = firstOrderGaussMarkov(t,dt,s1);
fogm2_2 = firstOrderGaussMarkov(t,dt,s2);
fogm3_2 = firstOrderGaussMarkov(t,dt,s3);

S = [s1';s2';s3'];
RW = [rw1';rw2';rw3'];
FOGMa = [fogm1_1';fogm2_1';fogm3_1'];
FOGMb = [fogm1_2';fogm2_2';fogm3_2'];

%Save data in a text file
fileID = fopen('stochastic_process2000.txt','w');
fprintf(fileID,'%8.8f, %8.8f,
%8.8f\n',[fogm1_1,fogm2_1,fogm3_1]);
fclose(fileID);

fileID = fopen('stochastic_process500.txt','w');
fprintf(fileID,'%8.8f, %8.8f,
%8.8f\n',[fogm1_2,fogm2_2,fogm3_2]);
fclose(fileID);

fileID = fopen('rw.txt','w');
fprintf(fileID,'%8.8f, %8.8f, %8.8f\n',RW);
fclose(fileID);

fileID = fopen('wn.txt','w');
fprintf(fileID,'%8.8f, %8.8f, %8.8f\n',S);
fclose(fileID);

figure();
title('Original Signals');
hold on;
subplot(4,1,1);
plot(x,s1,x,s2,x,s3);
legend('1','2','3');
ylabel('Random Sequence');
subplot(4,1,2);
plot(x,rw1,x,rw2,x,rw3);
legend('1','2','3');
ylabel('Random Walks');
subplot(4,1,3);
plot(x,fogm1_1,x,fogm2_1,x,fogm3_1);
ylabel('G-M P (T=2000)');
legend('1','2','3');
subplot(4,1,4);
plot(x,fogm1_2,x,fogm2_2,x,fogm3_2);
ylabel('G-M P (T=500)');
legend('1','2','3');
xlabel('sample number');

%4) compute the noise characteristics
for i=1:3
%a) autocorrelation function
   C_S(:,i) = xcorr(S(i,:),'unbiased');
   C_RW(:,i) = xcorr(RW(i,:),'unbiased');
   C_FOGMa(:,i) = xcorr(FOGMa(i,:),'unbiased');
   C_FOGMb(:,i) = xcorr(FOGMb(i,:),'unbiased');

%b) power spectral density
   [H_S(:,i),w1] = pwelch(C_S(:,i));
   [H_RW(:,i),w2] = pwelch(C_RW(:,i));
   [H_FOGMa(:,i),w3] = pwelch(C_FOGMa(:,i));
   [H_FOGMb(:,i),w4] = pwelch(C_FOGMb(:,i));

end

%plotting values from Part B:
figure();
title('Noise Characteristics');

x = 1:size(C_S,1);
```

```
for i=1:3
   subplot(2,4,1);
   hold on;
   plot(x,C_S(:,i));
end
title('White Noise');
legend('WN1','WN2','WN3');
ylabel('Autocorrelation Function');

for i=1:3
   subplot(2,4,2);
   hold on;
   plot(x,C_RW(:,i));
end
title('Random Walk');
legend('RW1','RW2','RW3');

for i=1:3
   subplot(2,4,3);
   hold on;
   plot(x,C_FOGMa(:,i));
end
title('Gauss-Markov (T=2000)');
legend('FOGMa1','FOGMa2','FOGMa3');

for i=1:3
   subplot(2,4,4);
   hold on;
   plot(x,C_FOGMb(:,i));
end
title('Gauss-Markov (T=500)');
legend('FOGMb1','FOGMb2','FOGMb3');

x=w1;
for i=1:3
   subplot(2,4,5);
   hold on;
   plot(x,10*log10(H_S(:,i)));
end
title('White Noise');
legend('WN1','WN2','WN3');
ylabel('Power Spectral Density');

for i=1:3
   subplot(2,4,6);
   hold on;
   plot(x,10*log10(H_RW(:,i)));
end
title('Random Walk');
legend('RW1','RW2','RW3');

for i=1:3
   subplot(2,4,7);
   hold on;
   plot(x,10*log10(H_FOGMa(:,i)));
end
title('Gauss-Markov (T=2000)');
legend('FOGMa1','FOGMa2','FOGMa3');

for i=1:3
   subplot(2,4,8);
   hold on;
   plot(x,10*log10(H_FOGMb(:,i)));
end
title('Gauss-Markov (T=500)');
legend('FOGMb1','FOGMb2','FOGMb3');
```

White Noise and Random Walk Generator

```
function [s,rw] = randomWalk(n,seed)
% computes a white noise and random walk
for a given sample size and seed value
    rng(seed);
    s = randn(n,1);
    rw = s;

    for i = 2:n
        rw(i) = rw(i)+rw(i-1);
    end
end
```

Gauss-Markov Process Generator

```
function [fogm] =
firstOrderGaussMarkov(t_correlation,
t_sample, s)
% computes the first order guass-markov
process given:
% correlation time, sample time, and
random sequence
Z = zeros(length(s),1);
Z(1) = s(1);
Beta = 1/t_correlation;
for i=2:length(s)
    Z(i) = exp(-Beta*t_sample)*Z(i-
1)+s(i);
end
fogm = Z;
end
```