Simon Honigmann

Sensor Orientation

October 12, 2018

Lab 2: Analysis and Modeling of Observed Stochastic Processes

Part A: Data Collection and Plotting

In the first session of this laboratory exercise, data was collected for 6 different IMUs across all areas of the quality and cost spectrum, ranging from Navigation grade to low-cost consumer grade. Approximately 20 minutes of static environment data was collected for each sensor to be processed and analyzed. The data was further subdivided into measurement type (gyroscope or accelerometer) and axis (x, y, or z).

I was assigned to group 7 which was tasked with analyzing the y and z axis data for the AIRINS iXSea navigation grade fibreoptic gyroscope and Intersense NavChip low-cost MEMS gyroscope. For each selected data series, plots were made to show the raw data, the signal autocorrelation, the signal's power spectral density, the Allan Variance, and the Wavelet Variance. These are included on the following pages. Table 1 below includes the average and standard deviations of the signals for each sensor and axis.

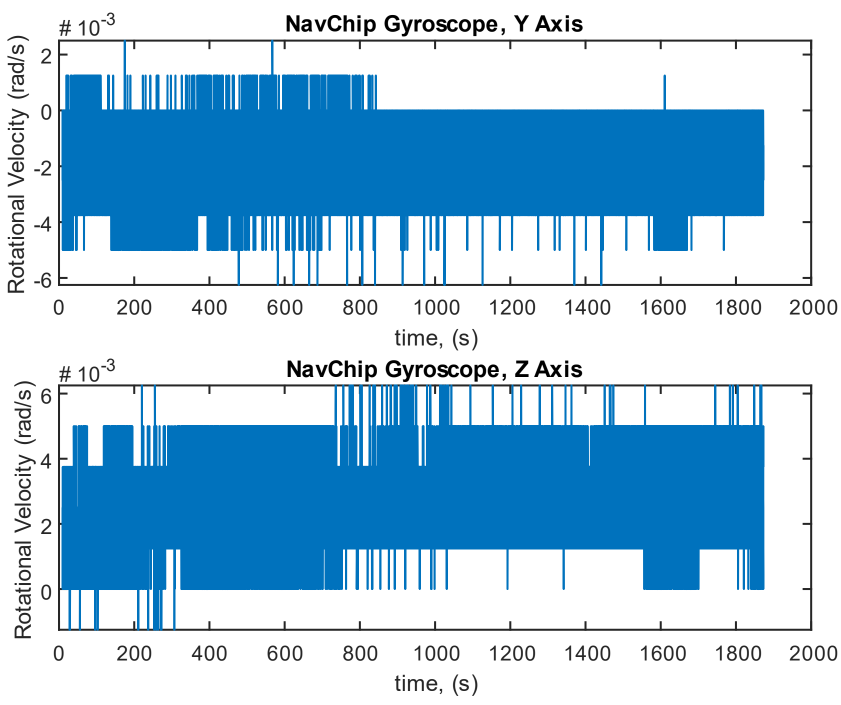


Figure 1: NavChip Gyroscope Y and Z Axis Data for a Stationary Environment

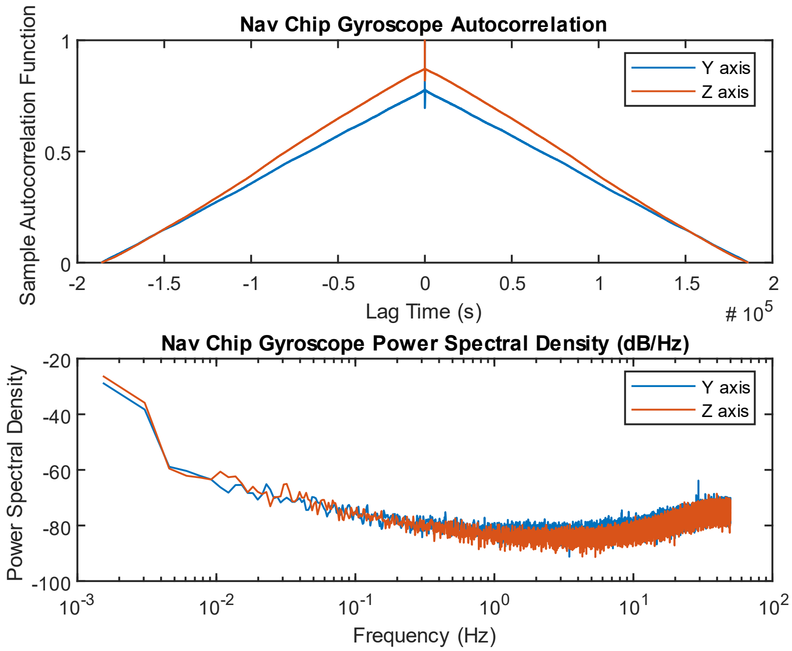
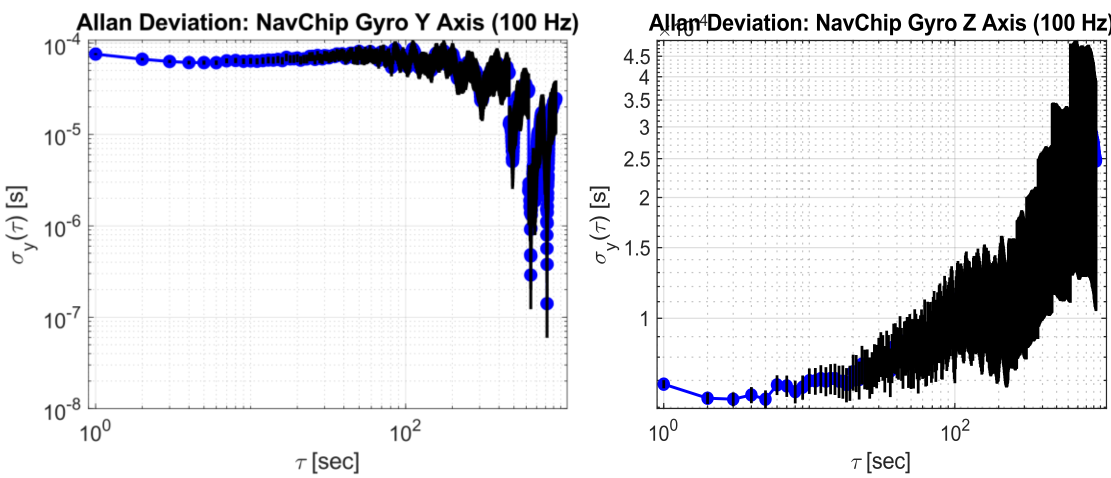


Figure 2: Autocorrelation Function and Power Spectral Density for the NavChip Gyroscope

a)



b)

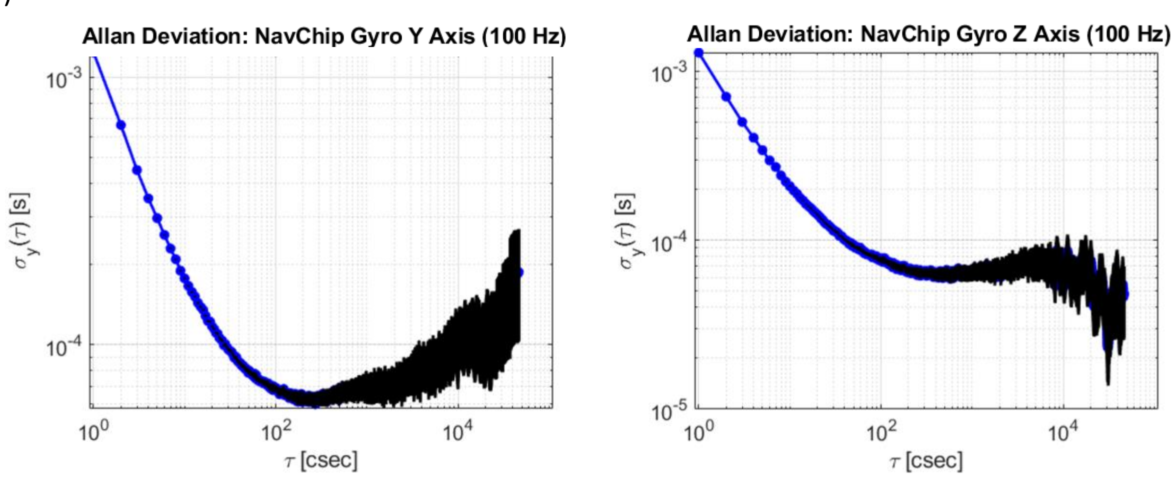
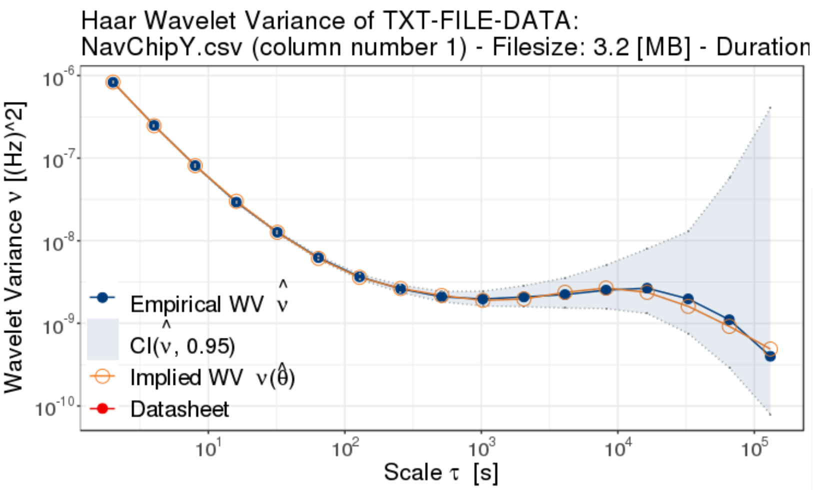


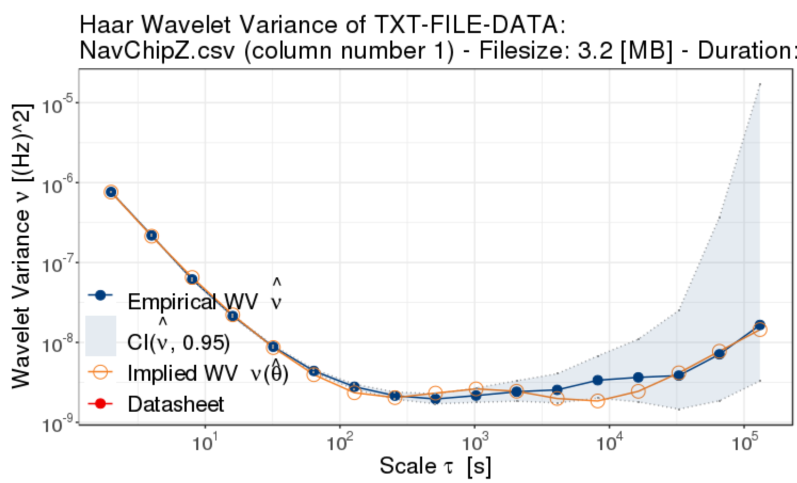
Figure 3: NavChip Gyroscope Allan Deviation Plot for the X and Y Axis with (a) Modified Function Sampling Frequency and with (b) UnModified Sampling Frequency and Larger Plot Domain, labels manually changed. Note x axis unit change.




```
Objective Function:  0.5445247
Seed Number:  1982

            Estimates       CI Low       CI High
BETA      3.985194e-02 3.985194e-02 3.985194e-02
SIGMA2_GM 1.323445e-08 1.128931e-08 1.517958e-08
BETA      1.123650e+00 1.123650e+00 1.123650e+00
SIGMA2_GM 6.349023e-09 5.742944e-09 6.955103e-09
WN        3.080514e-07 3.041819e-07 3.119209e-07
QN        4.519083e-07 4.480728e-07 4.557438e-07
```

Figure 4: Haar Wavelet Variance of the NavChip Gyroscope Y Axis




```
Objective Function:  4.119528
Seed Number:  1982

            Estimates       CI Low       CI High
BETA      1.455182e-03 1.455182e-03 1.455182e-03
SIGMA2_GM 1.381061e-08 1.270239e-08 1.491882e-08
BETA      3.807681e+00 3.807681e+00 3.807681e+00
SIGMA2_GM 3.079805e-07 3.045229e-07 3.114381e-07
QN        4.520769e-07 4.483507e-07 4.558031e-07
```

Figure 5: Haar Wavelet Variance of the NavChip Gyroscope Z Axis

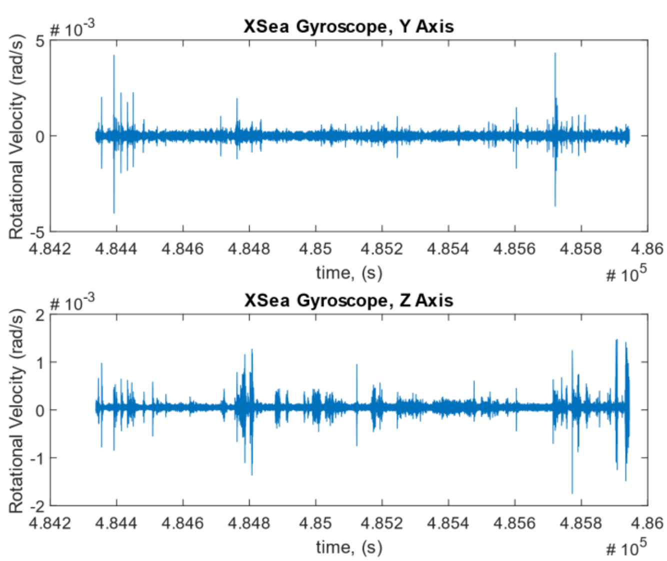


Figure 6: IXSea Gyroscope Y and Z Axis Data for a Stationary Environment

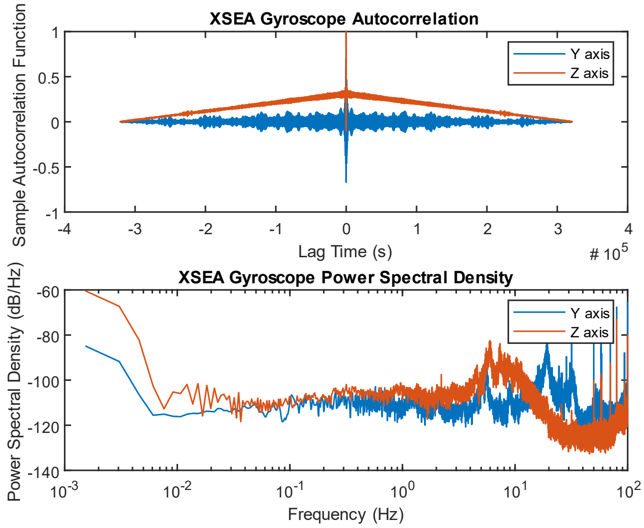


Figure 7: Autocorrelation Function and Power Spectral Density for the IXSEA Gyroscope

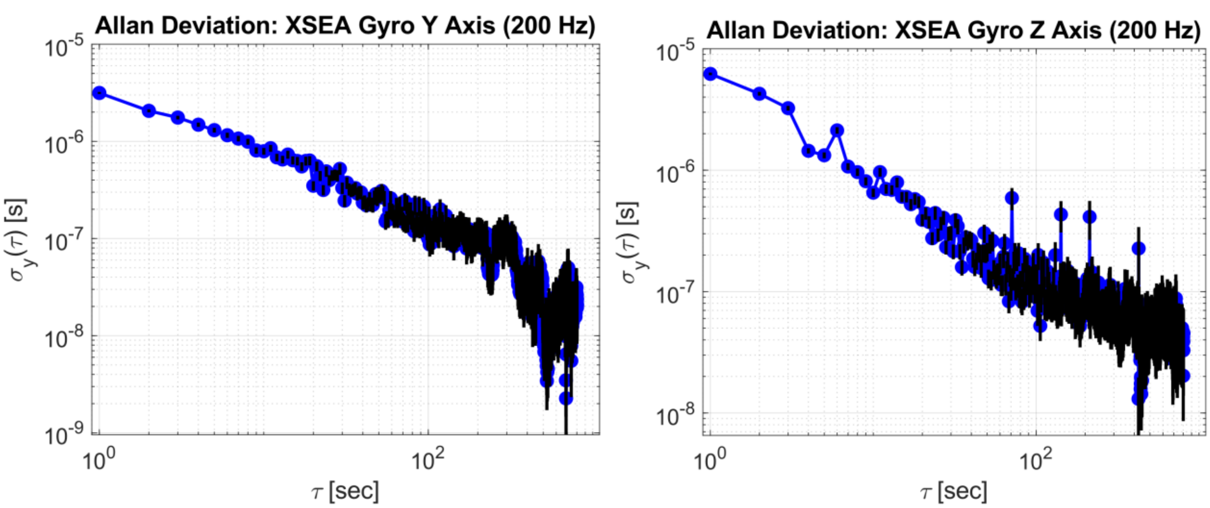
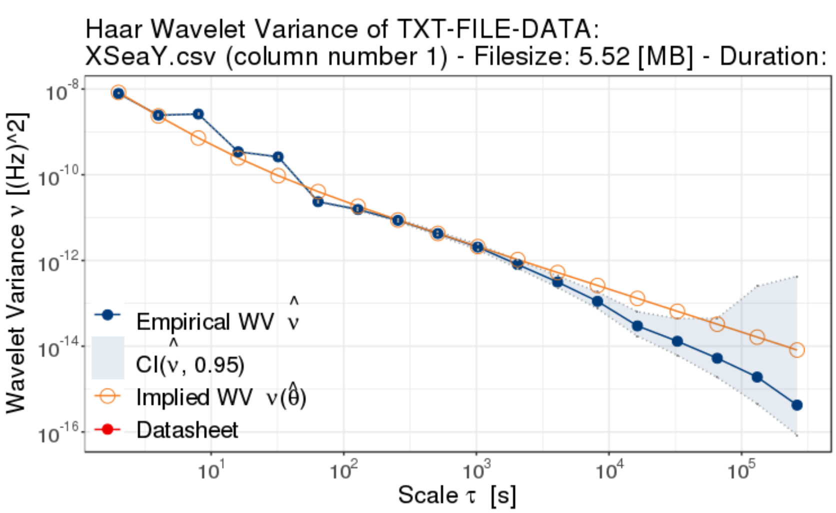


Figure 8: X and Y Axis Allan Deviation for the IXSEA Gyroscope




```
Objective Function:   989.8248
Seed Number:   1982

      Estimates        CI Low       CI High
WN 2.134794e-09 2.106820e-09 2.162768e-09
QN 4.858064e-09 4.830358e-09 4.885770e-09
```

Figure 9: Haar Wavelet Variance of the IXSEA Gyroscope Y Axis

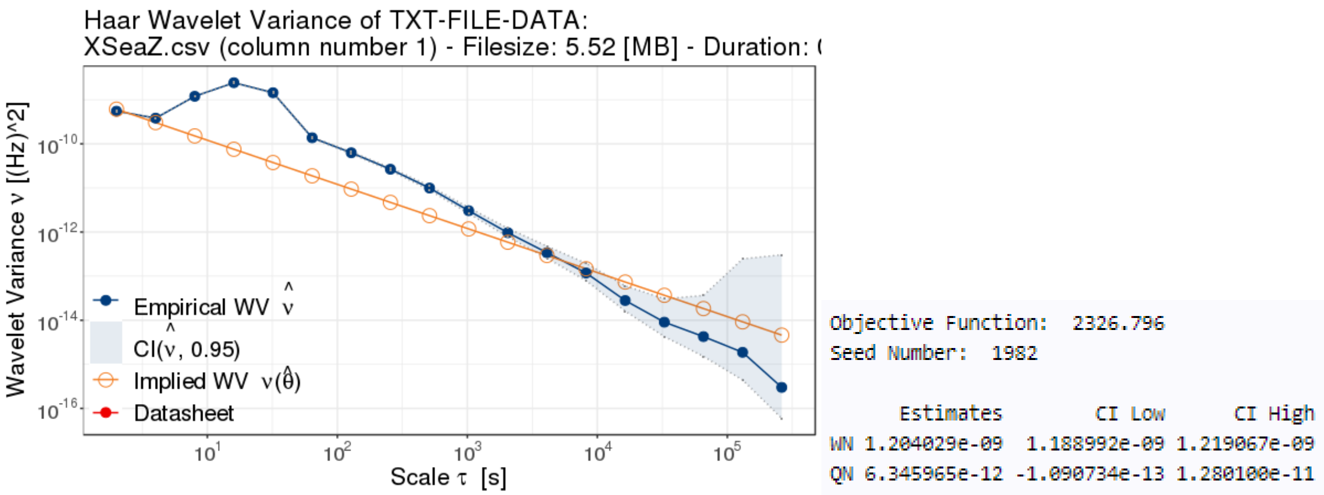


Figure 10: Haar Wavelet Variance of the IXSEA Gyroscope Z Axis

Part B:

Each signal was analyzed, similarly to in Lab 1, using the plots shown in Part A, as well as the characteristic values shown below in Table 1.

Table 1: Sensor Noise Characteristics

Sensor	Y Axis Mean (Std Dev.) rad/s	Z Axis Mean (Std Dev.) rad/s	Average Mean (Std Dev.) rad/s
Intersense NavChip	-2.1e-3 (1.1e-3)	2.7e-3 (1.1e-3)	3.2e-4 (1.1e-3)
AIRINS iXSea	3.15e-6 (1.17e-4)	5.30e-5 (7.91e-5)	2.49e-5 (9.79e-4)

While signal mean was not directly requested, it is worth noting that the XSEA navigation grade sensor has a signal mean which is more than an order of magnitude smaller than the standard deviation. While the mean ignores many potential dynamics of the process, it is interesting to theorise that this suggests the absence of a constant bias or integrated noise (e.g. random walk). Conversely, with the low-cost NavChip sensor, the mean value is an order of magnitude larger than the standard deviation which could potentially indicate a constant bias or the presence of correlated noise. As anticipated, the standard deviation of the navigation grade sensor is an order of magnitude smaller than that of the low-cost sensor; a fact which serves to justify the several orders of magnitude in cost difference between the two.

The following discussion questions will go into more detail on the existence and behaviour of the different process types which each signal is comprised of. Each sensor will be considered separately.

Intersense NavChip

I. Modeling: How do you suggest to model the observed process and why? Decide on the process type and strength (e.g. amplitude, correlation time, etc.)

Based on the highly linear Autocorrelation plot for both axes, it is clear that the noise of the NavChip gyroscope has a linear dependence between the current value of the series and the lagged values of the series. The plot does decay to zero, indicating that the series is stationary. The general shape is reminiscent of the Random Walk autocorrelation seen in Lab 1, however the Power Spectral Density plot indicates the process is likely not a Random Walk. Looking at the signal's power spectral density, it appears to be composed of two main components: white noise at frequencies between 0.1 and 50 Hz, characterized by zero slope, and flicker noise for frequencies below 0.1 Hz.

The Allan Deviation presents different large average-time characteristics for the Y and Z axes. Both exhibit a a -0.5 slope for frequencies below 1s and then a zero-slope behaviour associated with bias instability until about 100s. However, as the sampling time increases beyond 100 seconds, the characteristics diverge. For the Y axis, the Allan Deviation function transitions to a slope of approximately -1, indicating quantization noise. The Z axis on the other hand, transitions to a slope of approximately +0.5 indicating a rate random walk. However, in both cases the white noise and bias instabilities appear to dominate the response.

Based on these observations, a model for the noise was first made by combining white noise and flicker noise or bias instability. White noise variance was set using the total system variance as determined by Matlab's std() function.

To create a flicker noise process, a custom function was created based on equation 5.40 in the course notes. The function requires a time vector, a white noise process, and a bias instability period. The bias instability period was selected to correspond with the end of the bias instability region in the Allan Variance plot. This was approximately 10,000 csec or 100 s. Again, to scale the process, it was multiplied by the signal's standard deviation.

These results are shown in Table 2 below.

Table 2: Manually Selected Models and Parameters for the NavChip Gyroscope

Axis	**Iteration**	**Process**	**Parameters**	
			Variance	**Bias Instability Period, T_{BI} (s)**
Y	1	White Noise	1.23e-06	N/A
		Bias Instability	1.23e-06	100
	2	White Noise	TBD[1]	N/A
		Bias Instability	TBD	TBD
Z	1	White Noise	1.17e-06	N/A
		Bias Instability	1.17e-06	100
	2	White Noise	TBD	N/A
		Bias Instability	TBD	TBD

[1] Unfortunately, due to time constraints and due to difficulties generating a bias instability signal with the appropriate behaviour, a second iteration was not performed. This is discussed further below.

Table 3: Models and Parameters Determined using GMWM for the NavChip Gyroscope

Axis	Process	Parameters	
		Variance	Correlation Time (s)
Y	White Noise	3.08e-7	N/A
	Quantization Noise	4.52e-7	N/A
	First Order Gauss-Markov (#1)	1.32e-8	25.1
	First Order Gauss-Markov (#2)	6.35e-9	890e-3
Z	Quantization Noise	4.52e-7	N/A
	First Order Gauss-Markov (#1)	1.38e-8	687
	First Order Gauss-Markov (#2)	3.08e-7	263e-3

II. Verification: Simulate synthetic observations by applying the suggested model from Question I. Calculate again characteristic function from the simulated data and compare it to the characteristic obtained from the real data in Step 4. Answer the following question: Does the characteristic function obtained from real data correspond to the synthetic one? If not, what do you suggest to improve?

The original model, composed of bias instability and white noise did come reasonably close to representing the actual data, however there was room for improvement. The Autocorrelation and PSD plots, shown below, matched the originals quite well.

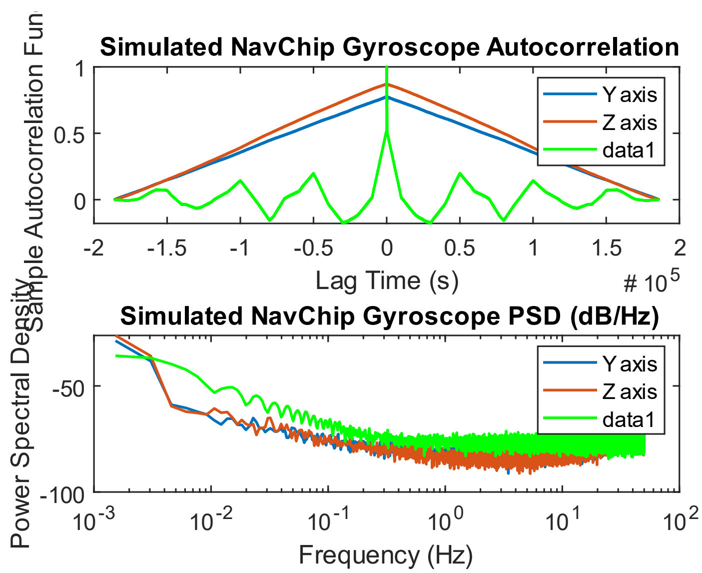


Figure 11: NavChip Z Axis Simulated Data vs Real Data – Autocorrelation and PSD

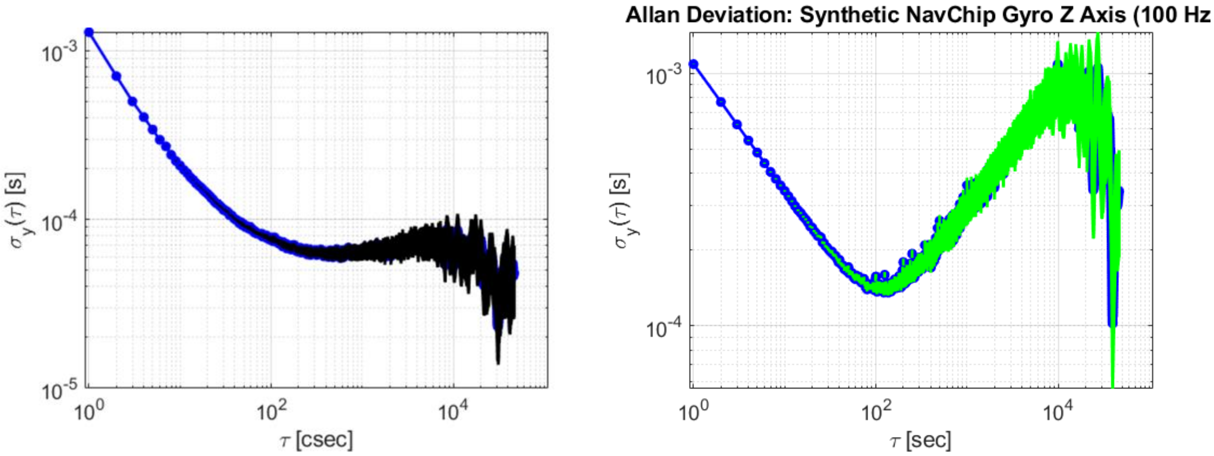


Figure 12: NavChip Z Axis – Real (left) vs Simulated (right) Allen Variance

However, the Allan variance plot indicated a +0.5 slope where a 0 slope was desired. This indicated a problem with the Bias Instability function I had made. Instead, it behaved like a random walk. I checked and double checked my function and how it was used and could not find the issue… Below I have included the generated bias instability function (prior to modifying the variance of the input noise) as a sanity check.

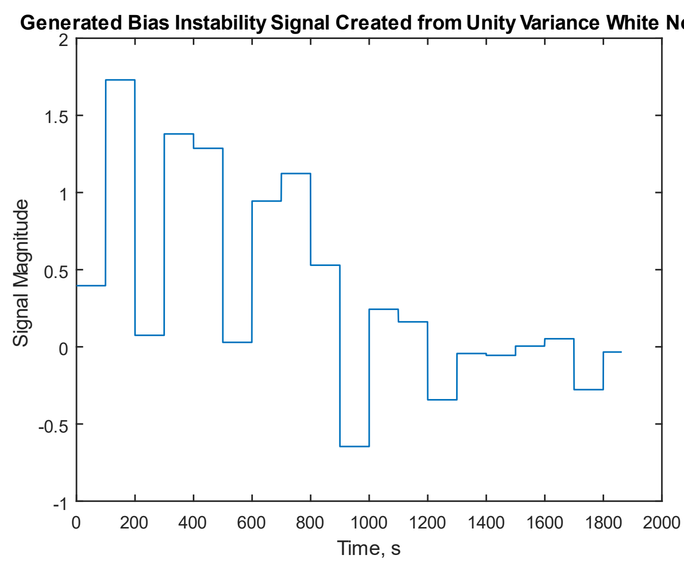


Figure 13: Generated Bias Instability Function

Hopefully just a windowing issue? At this point, I am out of time and out of knowledge, so I will do what I might do in industry and say "it's good enough, time to divert resources elsewhere". My suggestion for improvement would be to find a standard process that actually has a 0 slope in the Allan Variance plot, and to add that around a frequency of 10mHz. A Gauss-Markov process would likely be suitable, as demonstrated by the GMWM function fitting. The variance's used to scale each component process could also be tweaked to better approach the real signal.

AIRINS iXSea

I. Modeling: How do you suggest to model the observed process and why? Decide on the process type and strength (e.g. amplitude, correlation time, etc.)

The autocorrelation function for the XSEA Gyroscope is particularly interesting because the Y and Z axes demonstrate different processes. The Y axis predominantly shows white noise, with relatively weak periodic correlations. The Z axis however, indicates a noticeable correlation demonstrated by the decreasing linear trend seen in the plot, again reminiscent of a random walk or Gauss-Markov process. The Z axis does not appear to have a significant periodic component.

The power spectral density plot shows that the majority of the spectrum, from approximately 10 mHz to 5 Hz appears to be white noise with zero slope. At very low frequencies, both signals reach a peak. The measurements at this range are limited by the duration of our sample, but it is very possible that they would continue to increase. One possible explanation for very low frequency content is that the sensor is picking up on planetary rotations, however it would be difficult to comment with any certainty without a longer data series. At higher frequencies (f>5 Hz) both signals have local peaks. The Allan Variance plots for the XSEA Gyroscope are nearly ideal representations of a white noise process. Both plots are highly linear and have a slope of about -1.

Based on these observations, a white noise model was used to represent the characteristics. The standard deviation returned by Matlab's std() function was used to set the variance. The parameters for this model are shown below in Table 4.

Table4: Manually Selected Models and Parameters for the XSea Gyroscope

Axis	**Process**	**Variance**
Y	White Noise	1.37e-8
Z	White Noise	6.26e-9

Again, the model was compared to that determined using the GMWM application (parameters in the table below).

Table 5: Models and Parameters Determined using GMWM for the XSea Gyroscope

Axis	**Process**	**Variance**
Y	White Noise	2.13e-9
	Quantization Noise	4.86e-9
Z	White Noise	1.20e-9
	Quantization Noise	6.35e-12

The wavelet analysis indicated the need, in both axes, for a process in addition to the white noise previously identified. A combination of quantization noise and white noise resulted in the best fitting models. On the Y axis, the white noise and quantization noise are of similar magnitudes. However, with the Z axis, the quantization noise is several orders of magnitude smaller than the white noise component. It is worth noting, however, that even generating a model incorporating every type of noise and multiple Gauss-Markov processes could not result in suitable fit at lower values of τ.

II. Verification: Simulate synthetic observations by applying the suggested model from Question I. Calculate again characteristic function from the simulated data and compare it to the characteristic obtained from the real data in Step 4. Answer the following question: Does the characteristic function obtained from real data correspond to the synthetic one? If not, what do you suggest to improve?

Synthetic data was simulated using Matlab based on the process specified in Table 4. The synthetic process was then compared to the actual process using the Autocorrelation Function, PSD, and Allan Variance plots. While the magnitude varied slightly, and the lower strength frequency content seen in the original PSD plot was not present, the white noise model generally represented the actual data very well. The simulated data was reprocessed using the variance returned by the GMWM and the result was a much closer match. Both iterations of simulated data are compared to the actual data in the following figures.

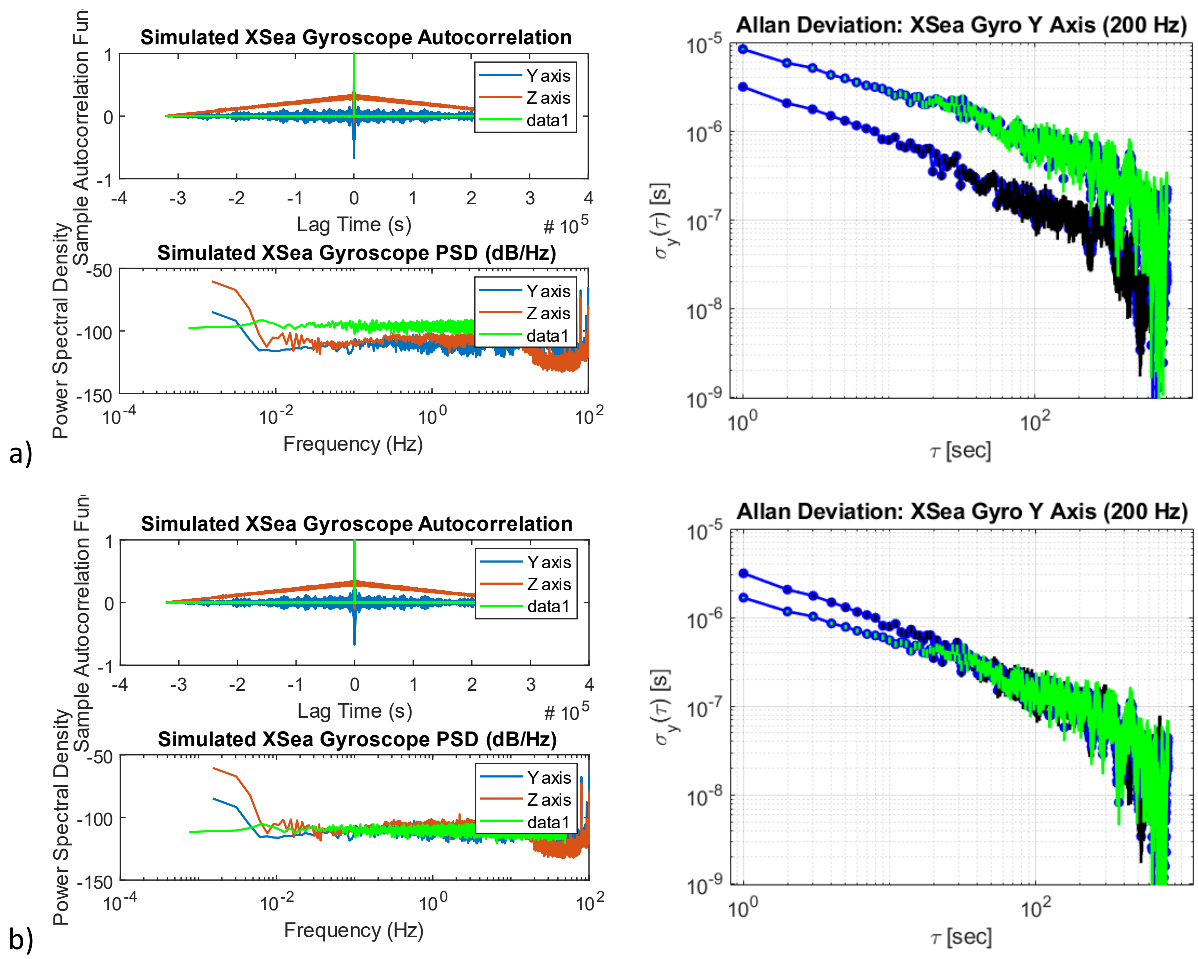


Figure 14: Synthetic Noise Characteristics Compared to Measured Noise Characteristics, using Real Signal Standard Deviation (a), and 1/5 Standard Deviation (b).

Appendix A: MATLAB Code

```
%% Simon Honigmann
% Sensor Orientation Lab 2
% Group: 7
% Sensors: AIRINS/XSEA (Mechanical) & NavChip/Intersense (MEMS)
% Sensor Analyzed: Gyroscope
% Axes: Y & Z

%% Collect Data
clc;
load Sensor1and4Data; %saved relevant data as a mat file to not have to manually select
every time

%extract and store sampling frequencies
fs_NC = 1/(dataNavChip(2,1)-dataNavChip(1,1));
fs_XS = 1/(dataXSEA(2,1)-dataXSEA(1,1));

%% Compute Characteristic Parameters

stdDevs = [std(dataNavChip(922:187222,3)),
std(dataNavChip(922:187222,4));std(dataXSEA(:,3)),std(dataXSEA(:,4))];
stdDevs = [stdDevs,mean(stdDevs,2)];

means = [mean(dataNavChip(922:187222,3)),
mean(dataNavChip(922:187222,4));mean(dataXSEA(:,3)),mean(dataXSEA(:,4))];
means = [means,mean(means,2)];

%% Plot Raw Data
if 0
figure(1);
subplot(2,1,1);
plot(dataNavChip(922:187222,1),dataNavChip(922:187222,3));
title('NavChip Gyroscope, Y Axis');
xlabel('time, (s)');
ylabel('Rotational Velocity (rad/s)');

subplot(2,1,2);
plot(dataNavChip(922:187222,1),dataNavChip(922:187222,4));
title('NavChip Gyroscope, Z Axis');
xlabel('time, (s)');
ylabel('Rotational Velocity (rad/s)');

figure(2);
subplot(2,1,1);
plot(dataXSEA(:,1),dataXSEA(:,3));
title('XSea Gyroscope, Y Axis');
xlabel('time, (s)');
ylabel('Rotational Velocity (rad/s)');

subplot(2,1,2);
plot(dataXSEA(:,1),dataXSEA(:,4));
title('XSea Gyroscope, Z Axis');
xlabel('time, (s)');
ylabel('Rotational Velocity (rad/s)');
end
%% Autocorrelation
[C_NCy,L_NCy] = xcorr(dataNavChip(922:187222,3),'coeff');
[C_NCz,L_NCz] = xcorr(dataNavChip(922:187222,4),'coeff');
```

```
[C_XSy,L_XSy] = xcorr(dataXSEA(:,3),'coeff');
[C_XSz,L_XSz] = xcorr(dataXSEA(:,4),'coeff');

figure(3);
subplot(2,1,1);
plot(L_NCy,(C_NCy));
hold on;
plot(L_NCz,(C_NCz));
title('Nav Chip Gyroscope Autocorrelation');
legend('Y axis', 'Z axis');
ylabel('Sample Autocorrelation Function');
xlabel('Lag Time (s)');

figure(4);
subplot(2,1,1);
plot(L_XSy,(C_XSy));
hold on;
plot(L_XSz,(C_XSz));
title('XSEA Gyroscope Autocorrelation');
legend('Y axis', 'Z axis');
ylabel('Sample Autocorrelation Function');
xlabel('Lag Time (s)');


% Power Spectral Density
[H_NCy,f_NCy] = pwelch(dataNavChip(922:187222,3),[],[],[],fs_NC);
[H_NCz,f_NCz] = pwelch(dataNavChip(922:187222,4),[],[],[],fs_NC);
[H_XSy,f_XSy] = pwelch(dataXSEA(:,3),[],[],[],fs_XS);
[H_XSz,f_XSz] = pwelch(dataXSEA(:,4),[],[],[],fs_XS);

figure(3);
subplot(2,1,2);
x=f_NCy;
y=10*log10(H_NCy);
x = [-fliplr(x')';x];
y = [fliplr(y')';y];
plot(x,y);
hold on;
x=f_NCz;
y=10*log10(H_NCz);
x = [-fliplr(x')';x];
y = [fliplr(y')';y];
plot(x,y);title('Nav Chip Gyroscope Power Spectral Density (dB/Hz)');
legend('Y axis', 'Z axis');
ylabel('Power Spectral Density');
xlabel('Frequency (Hz)');
set(gca,'XScale','log');

figure(4);
subplot(2,1,2);
x=f_XSy;
y=10*log10(H_XSy);
x = [-fliplr(x')';x];
y = [fliplr(y')';y];
plot(x,y);
hold on;
x=f_XSz;
y=10*log10(H_XSz);
x = [-fliplr(x')';x];
y = [fliplr(y')';y];
```

```
plot(x,y);
title('XSEA Gyroscope Power Spectral Density');
legend('Y axis', 'Z axis');
ylabel('Power Spectral Density (dB/Hz)');
xlabel('Frequency (Hz)');
set(gca,'XScale','log');

%% Allan Variance
if 0
    av_NCy = allandev(dataNavChip(922:187222,3),'NavChip Gyro Y Axis',5,'k',100);
    av_NCz = allandev(dataNavChip(922:187222,4),'NavChip Gyro Z Axis',6,'k',100);
    av_XSy = allandev(dataXSEA(:,3),'XSEA Gyro Y Axis',7,'k',200);
    av_XSz = allandev(dataXSEA(:,4),'XSEA Gyro Z Axis',8,'k',200);
end
%% Synthetic Noise for z axis of NavChip and Y axis of XSea
rng(1); %seed for repeatibility

%NavChip
std_z = stdDevs(1,2); %depends if I want to use std dev calculated for
%signal, or variance
wn = randn(187222-922+1,4); %white noise process
[~,rw] = randomWalk(size(wn,1),12345);

%Bias Instability
t = dataNavChip(922:187222,1)-dataNavChip(922,1);
Tbi = 100;
bi = biasInstability(t,wn(:,4),Tbi);

%Linear Combination of Processes Exhibited
NCNoiseZ = (bi(1:186300)+wn(1:186300,1))*std_z+QN;
%NCNoiseZ = QN+bi(1:186300)*std_z; %desperately trying to debug
% WHY DOES BIAS INSTABILITY HAVE A NON ZERO SLOPE IN ALLAN VARIANCE??? ☹

%PLOTTING THINGS
%Autocorrelation
[C,L] = xcorr(NCNoiseZ,'coeff');
figure(3);
subplot(2,1,1);
plot(L,(C),'g');
hold on;
title('Simulated NavChip Gyroscope Autocorrelation');
ylabel('Sample Autocorrelation Function');
xlabel('Lag Time (s)');

% Power Spectral Density
[H,fp] = pwelch(NCNoiseZ,[],[],[],f);

subplot(2,1,2);
x=fp;
y=10*log10(H);
x = [-fliplr(x')';x];
y = [fliplr(y')';y];
plot(x,y,'g');
hold on;
title('Simulated NavChip Gyroscope PSD (dB/Hz)');
ylabel('Power Spectral Density');
xlabel('Frequency (Hz)');
set(gca,'XScale','log');
```

```
%allan variance
avNC_synth = allandev(NCNoiseZ,'NavChip Gyro Z Axis',1,'g',100);
%av_NCz = allandev(dataNavChip(922:187222,4),'NavChip Gyro Z Axis',9,'k',100);

%% XSea Gyro Synthetic:
if 0 %stop from graphing when not working on this section
    std_y = stdDevs(2,1);
    wn_y = std_y*randn(size(dataXSEA,1),1);

    %Autocorrelation
    [C_NCy,L_NCy] = xcorr(wn_y,'coeff');
    figure(4);
    subplot(2,1,1);
    plot(L_NCy,(C_NCy),'g');
    hold on;
    title('Simulated XSea Gyroscope Autocorrelation');
    ylabel('Sample Autocorrelation Function');
    xlabel('Lag Time (s)');

    % Power Spectral Density
    [H_NCy,f_NCy] = pwelch(wn_y,[],[],[],fs_NC);

    subplot(2,1,2);
    x=f_NCy;
    y=10*log10(H_NCy);
    x = [-fliplr(x')';x];
    y = [fliplr(y')';y];
    plot(x,y,'g');
    hold on;
    title('Simulated XSea Gyroscope PSD (dB/Hz)');
    ylabel('Power Spectral Density');
    xlabel('Frequency (Hz)');
    set(gca,'XScale','log');

    %allan variance
    av_XSy = allandev(dataXSEA(:,3),'XSea Gyro Y Axis',10,'k',200);
    av_XSy_synth = allandev(wn_y,'XSea Gyro Y Axis',10,'g',200);
end
```

Appendix B: Bias Instability Function

```
function BI = biasInstability(t,W,Tbi)

    X = t;
    X(1) = W(1);

    for k=1:length(t)-1

        if(mod(t(k+1),Tbi) == 0)
            X(k+1) = W(k+1);
        else
            X(k+1) = X(k);
        end
    end

    BI=X;
end
```