Northeastern University
360 Huntington Ave, Boston, MA 02115

Report on
IMU Noise Characterization with Allan Variance

LAB_3 - EECE 5554 Robotics Sensing and Navigation

Submitted by:
Pavan Rathnakar Shetty

Submitted to
Hanumant Singh
Professor, Electrical and Computer Engineering

Detailed Analysis: Graphical plotting and Analysis of the CSV file was done using Jupyter Notebook by using Python and libraries like Matplotlib, CSV, Numpy, Statistics, and Pandas as described in the Imu Analysis.ipynb file.

IMU Noise Characterization with Allan Variance was done using MATLAB.

Roll in Degree MEAN: 3.1172 STANDARD DEVIATION: 0.0299	Pitch in Degree MEAN: -0.0296 STANDARD DEVIATION: 0.1215	Yaw in Degree MEAN: 12.3632 STANDARD DEVIATION: 0.2312
Gyro x in rad/s MEAN: 1.7954 STANDARD DEVIATION: 0.0011 Angle of Random Walk in degree/√hour N = 8.9345 Rate Random Walk in degree/√hour K = 0.3961 Bias Instability in degree/hour B = 412.529612495	Gyro y in rad/s MEAN: 5.9410 STANDARD DEVIATION: 0.0009 Angle of Random Walk in degree/√hour N = 1.3379 Rate Random Walk in degree/√hour K = 2.9078e-04 Bias Instability in degree/hour B = 1.1733579768	Gyro z in rad/s MEAN: -4.0518 STANDARD DEVIATION: 0.0006 Angle of Random Walk in degree/√hour N = 3.70543 Rate Random Walk in degree/√hour K = 4.973e-03 Bias Instability in degree/hour B = 1.11857404428
Accel x in m/s^2 MEAN: -0.0053 STANDARD DEVIATION: 0.0240	Accel y in m/s^2 MEAN: -0.5239 STANDARD DEVIATION: 0.0127	Accel z in m/s^2 MEAN: -9.6234 STANDARD DEVIATION: 0.0185

Angle of Random Walk in degree/√hour N = 6.8727 Rate Random Walk in degree/√hour K = 0.5656 Bias Instability in degree/hour B = 154.8306141617	Angle of Random Walk in degree/√hour N = 6.1854 Rate Random Walk in degree/√hour K = 2.3545 Bias Instability in degree/hour B = 226.8912868724	Angle of Random Walk in degree/√hour N = 8.9345 Rate Random Walk in degree/√hour K = 0.3961 Bias Instability in degree/hour B = 412.5296125
Mag x in Gauss MEAN: 0.1828 STANDARD DEVIATION: 0.00216	Mag y in Gauss MEAN: 0.0267 STANDARD DEVIATION: 0.0054	Mag z in Gauss MEAN: 0.4384 STANDARD DEVIATION: 0.0075

Since IMU with gyros and accelerometer are arguably the most important sensors of a control system, proper sensor modeling is important to achieve accurate vehicle simulation. In order to model the sensor, we need to characterize its noise.

If a noisy output signal from a sensor is integrated, for example integrating an angular rate signal to determine an angle, the integration will drift over time due to the noise. This drift is called random walk, as it will appear that the integration is taking random steps from one sample to the next. The in-run bias stability or the bias instability is a measure of how the bias will drift during operation over time at a constant temperature. This parameter also represents the best possible accuracy with which a sensor's bias can be estimated. Due to this, in-run bias stability is generally the most critical specification as it gives a floor to how accurate a bias can be measured.

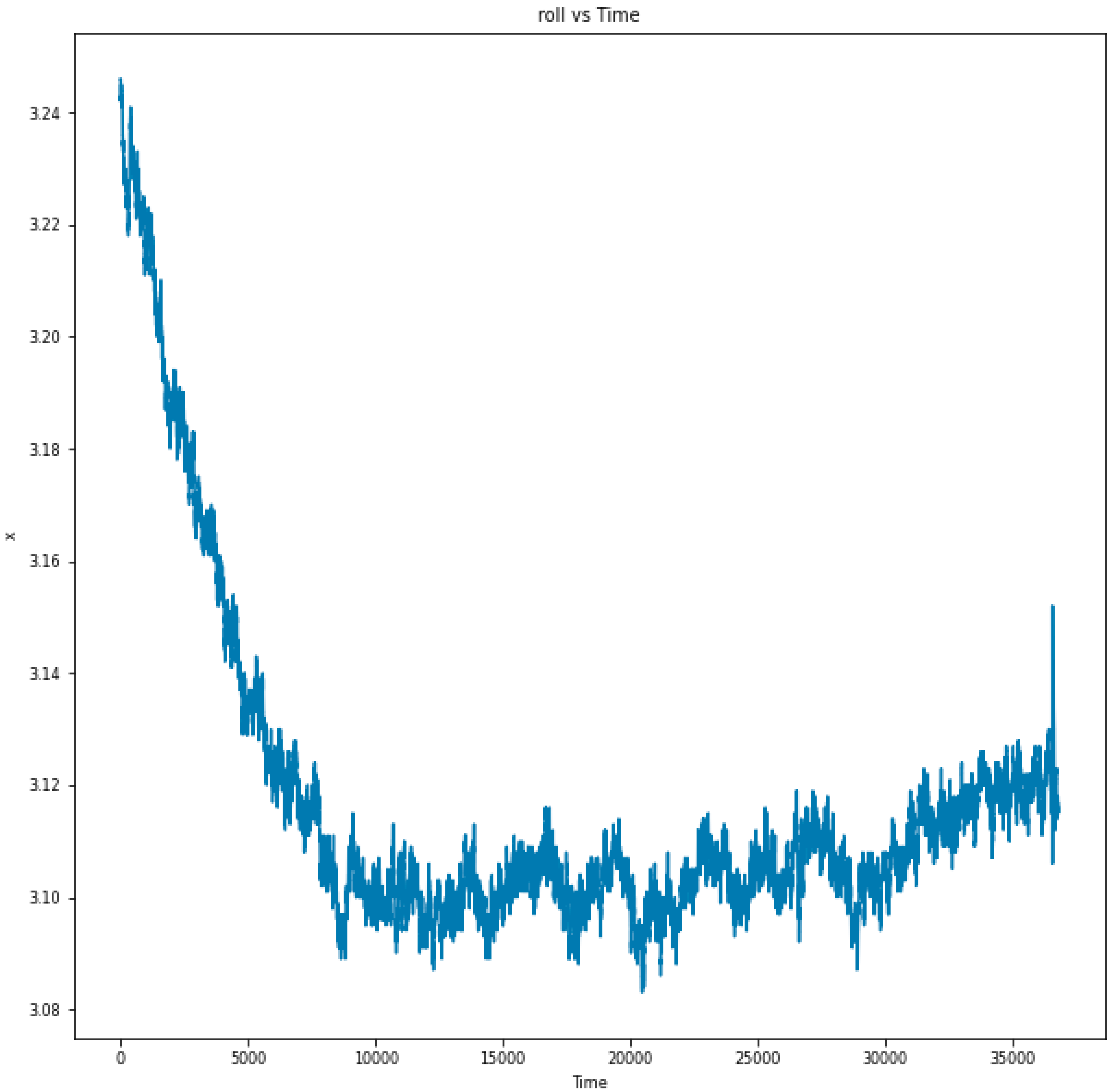
Every sensor will have measurement biases. Gyro bias will cause the angle to drift over time in time-integrated data. When the IMU is stationary the gyro measurement will have a non-zero reading which is called turn-on bias. In reality the gyro bias change over time. A high value of B(Bias instability) means that the sensor is less stable and change at a quicker rate.
Temperature is also another important factor for the instability of a sensor.

Fancy Ring Laser Gyro, Fibre Optic Gyro have low Biases and Random Walk values while MEMS Gyros have High B, N and K values.

Example of Random Walk and Bias Instability for a FXAS21002 X- Gyro, as shown above, has a pretty high value suggesting the sensor drifts over time and is quite unstable. This could be because of noisy data or unstable sensors.

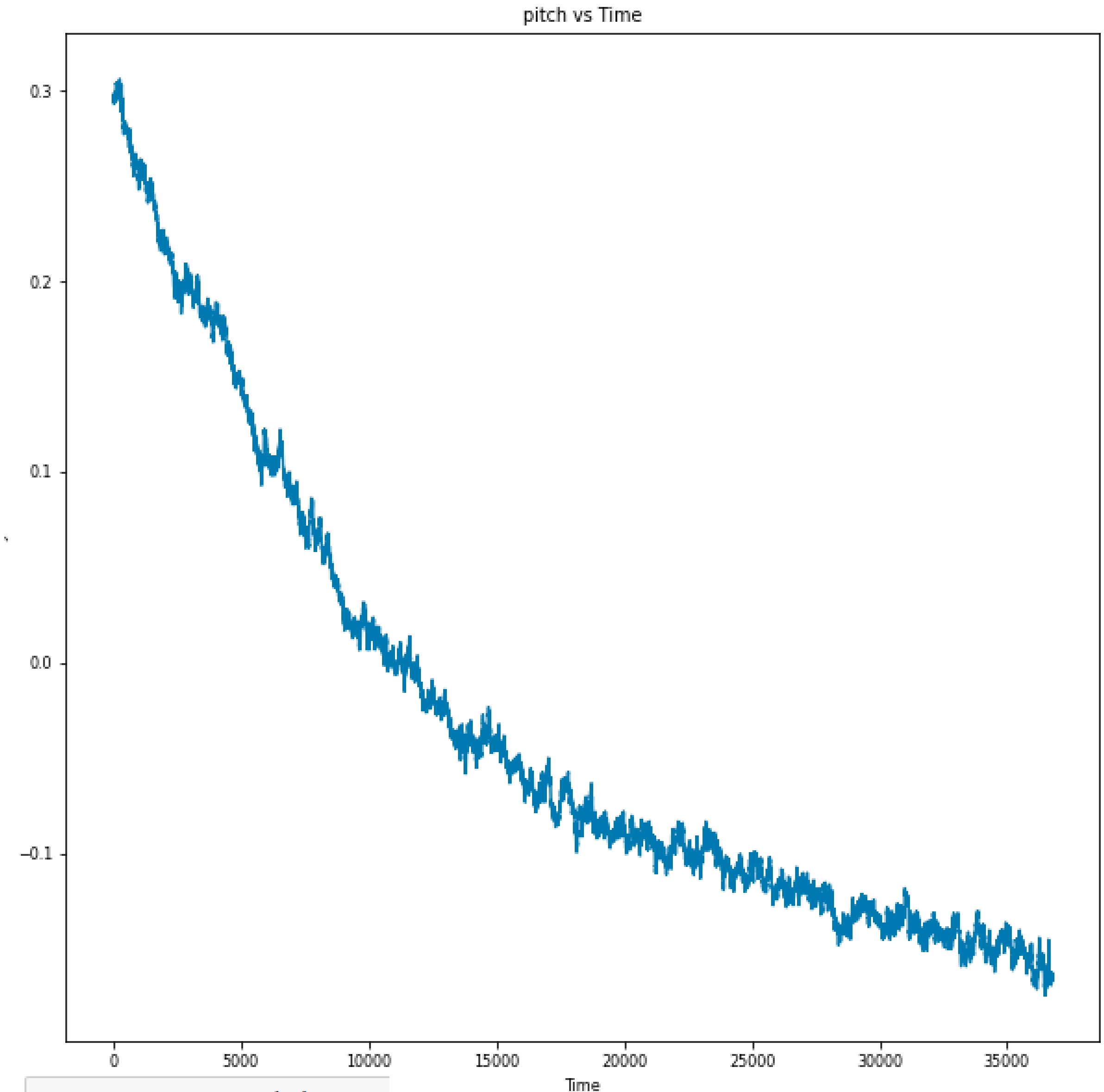
In our Allan Variance plots, we can understand that our sensor has Gaussian white noise which is very important because most sensor fusion algorithms, such as Kalman Filters, assume state measurements have Guassian white noise.
I recorded IMU data for over 5 hours at 40Hz at a place where there was limited vibration from all the places I could (basement).

After the Graphical analysis and Allan Variance analysis for 5-hour data, the values of B, K, and N with the Data Sheet of VN100 IMU. It is closer to consumer standards because of its high value of B, K, and N. Although some values match up to Industrial and Tactical standards, overall the values are comparable to Consumer Standards.
Graphs and error metrics for the collected data is shown below:



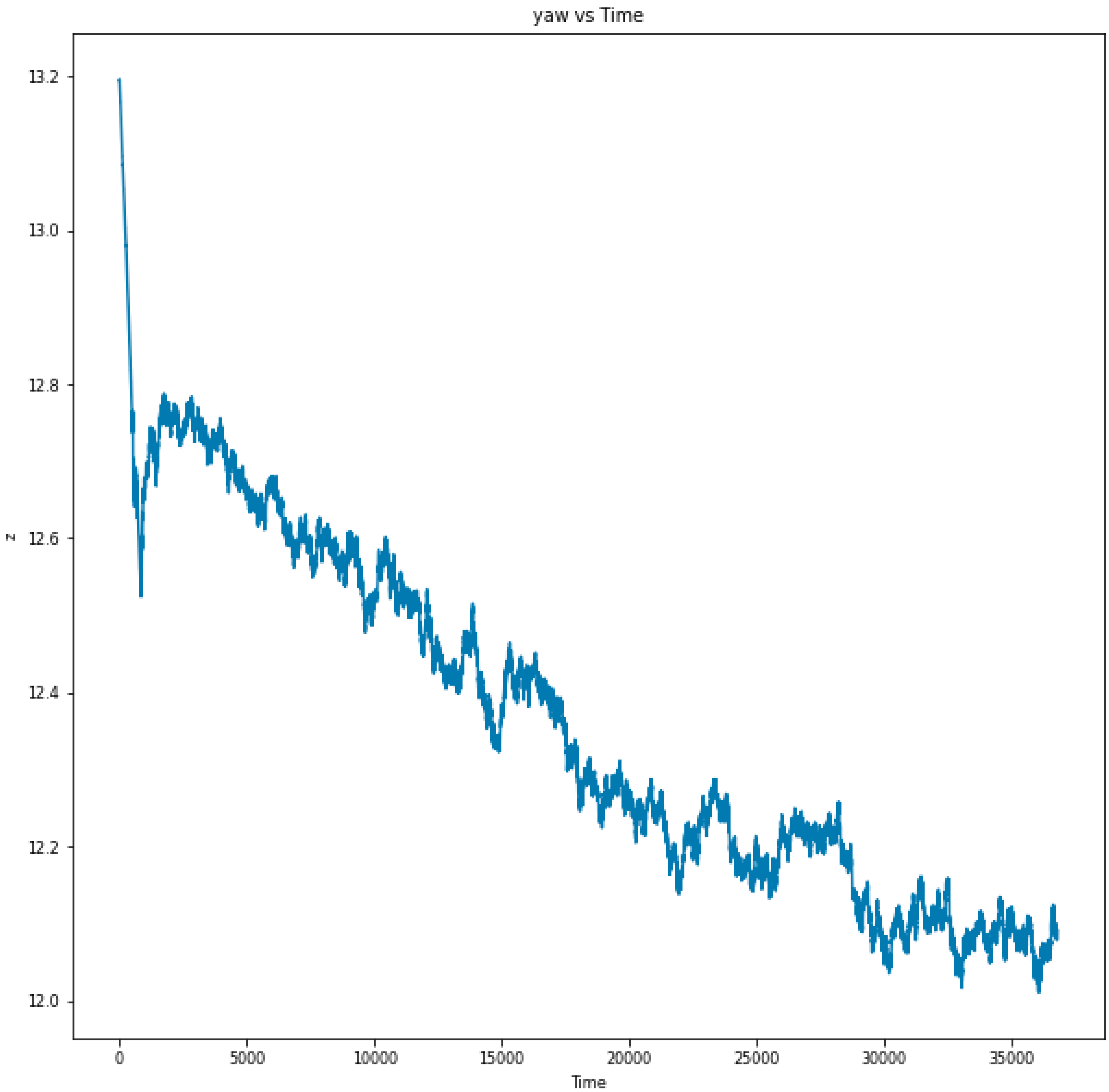

```
mn = numpy.mean(x)
print(mn)
sd = numpy.std(x)
print(sd)
```

```
3.1172670454545455
0.029914684860169842
```



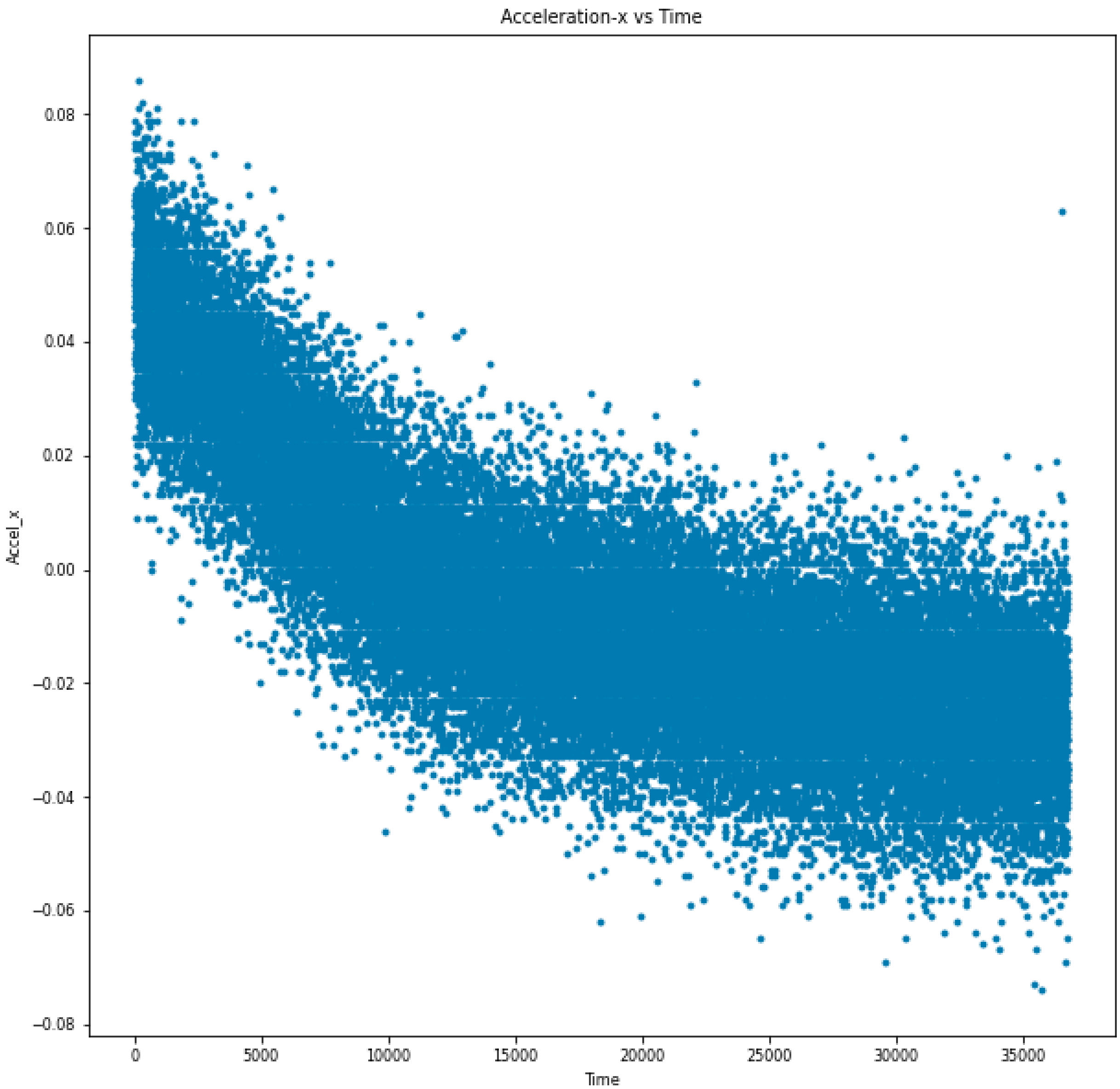

```
: mn = numpy.mean(y)
  print(mn)
  sd = numpy.std(y)
  print(sd)
```

```
-0.029666893214441062
0.12153004976969564
```



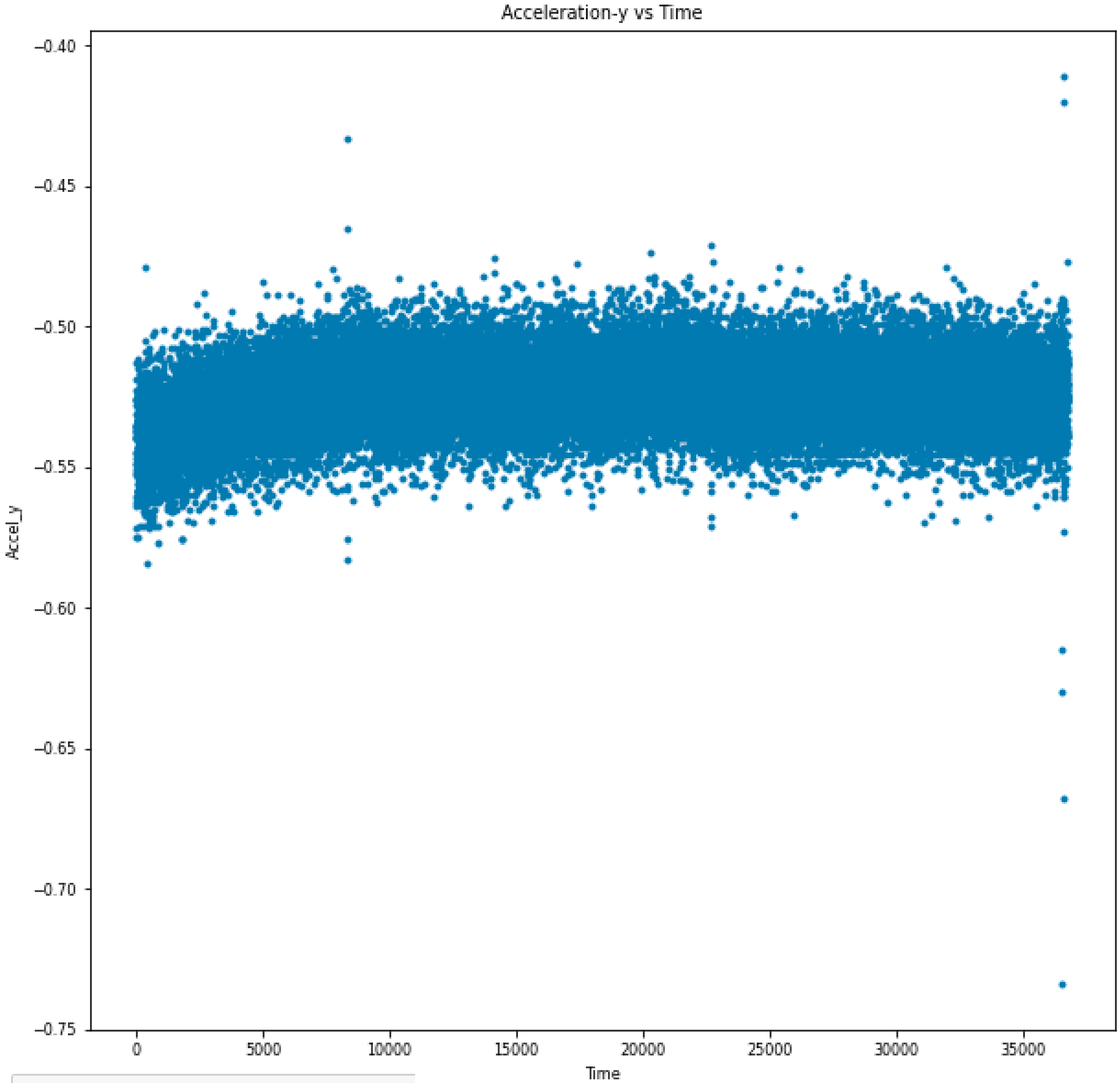

```
mn = numpy.mean(z)
print(mn)
sd = numpy.std(z)
print(sd)
```

```
12.363224282296649
0.23129851910217825
```



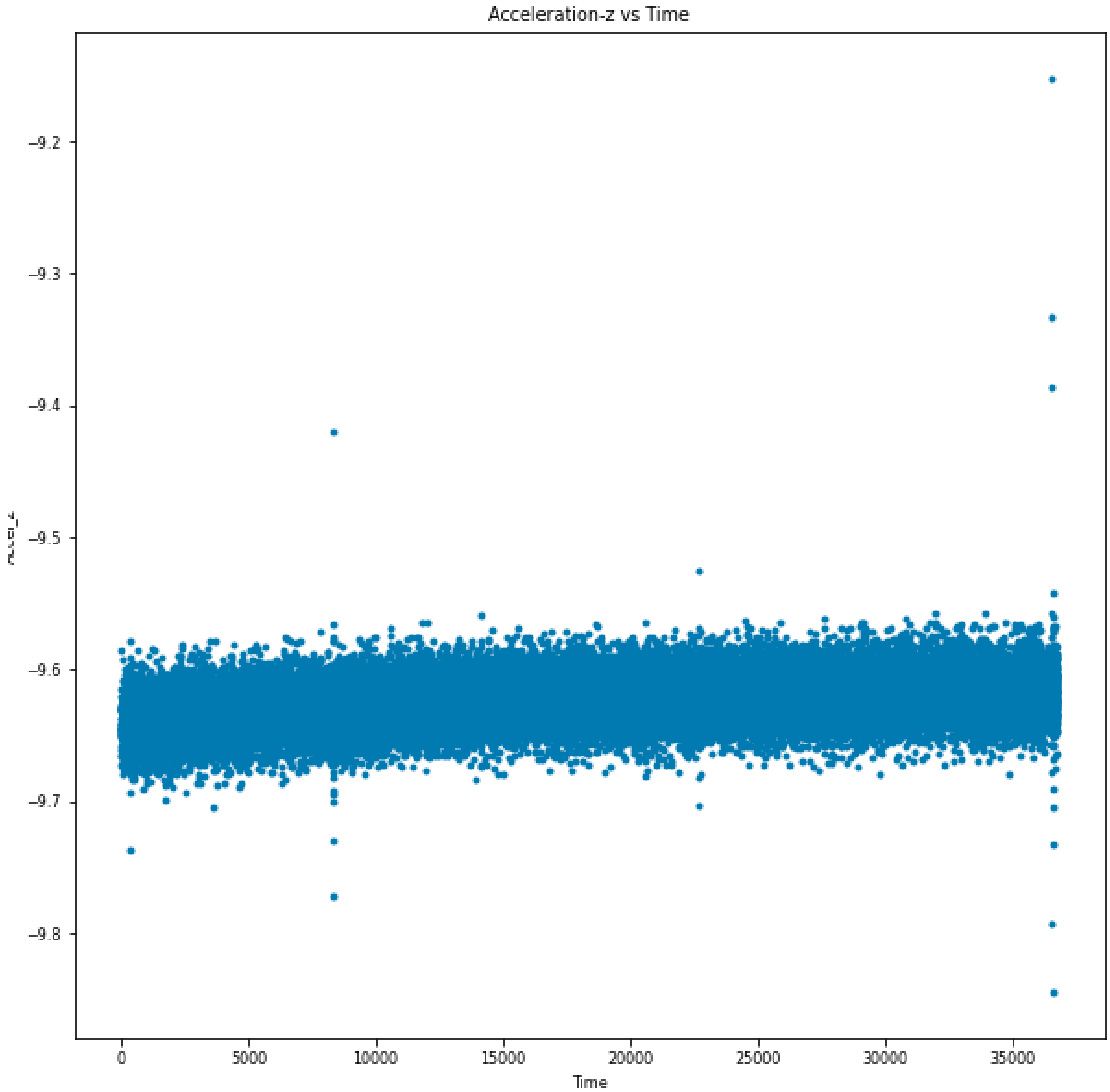

```
mn = numpy.mean(accelx)
print(mn)
sd = numpy.std(accelx)
print(sd)
```

```
-0.0053912570682905624
0.02402497014844733
```



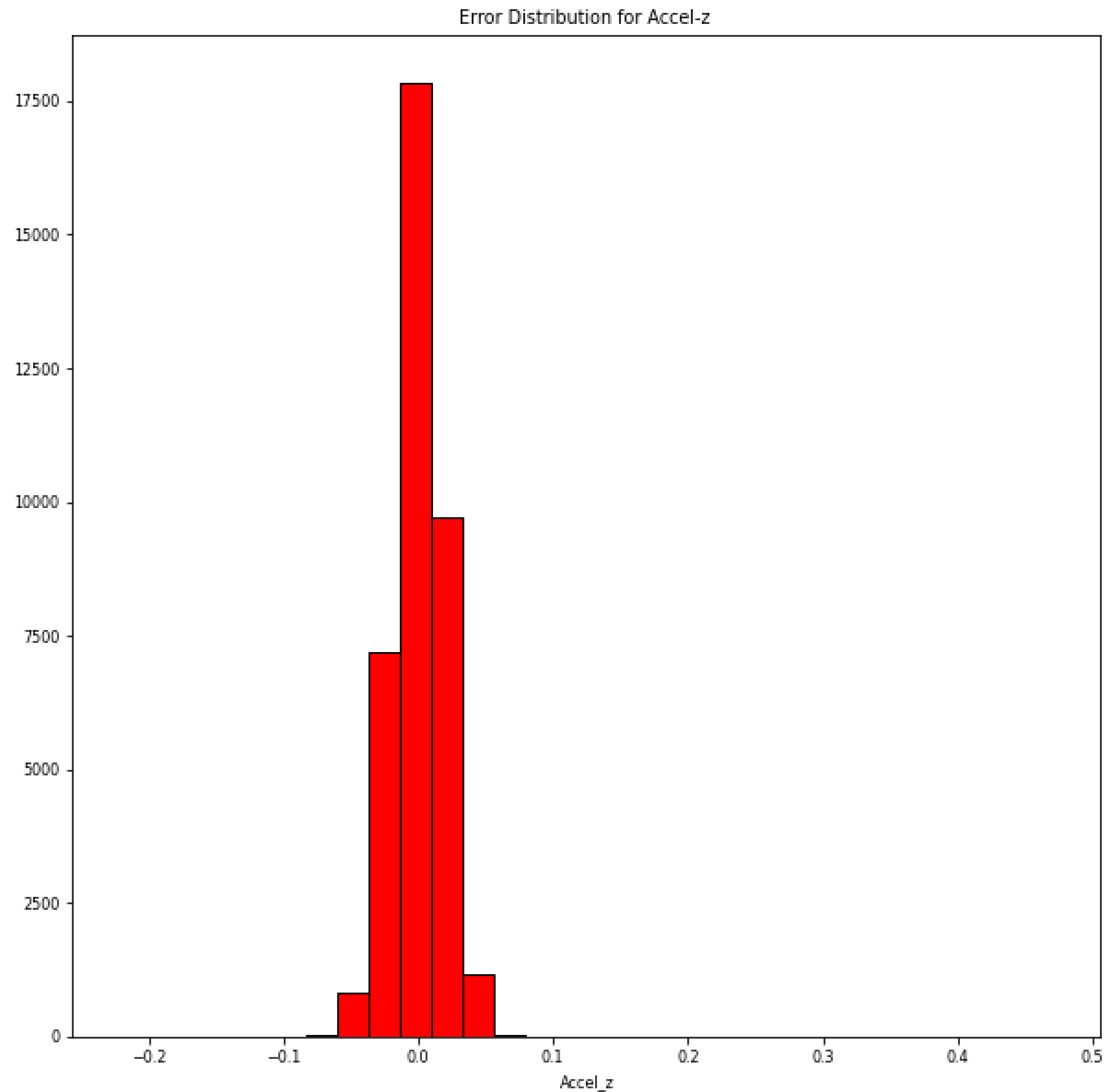

```
mn = numpy.mean(accely)
print(mn)
sd = numpy.std(accely)
print(sd)
```

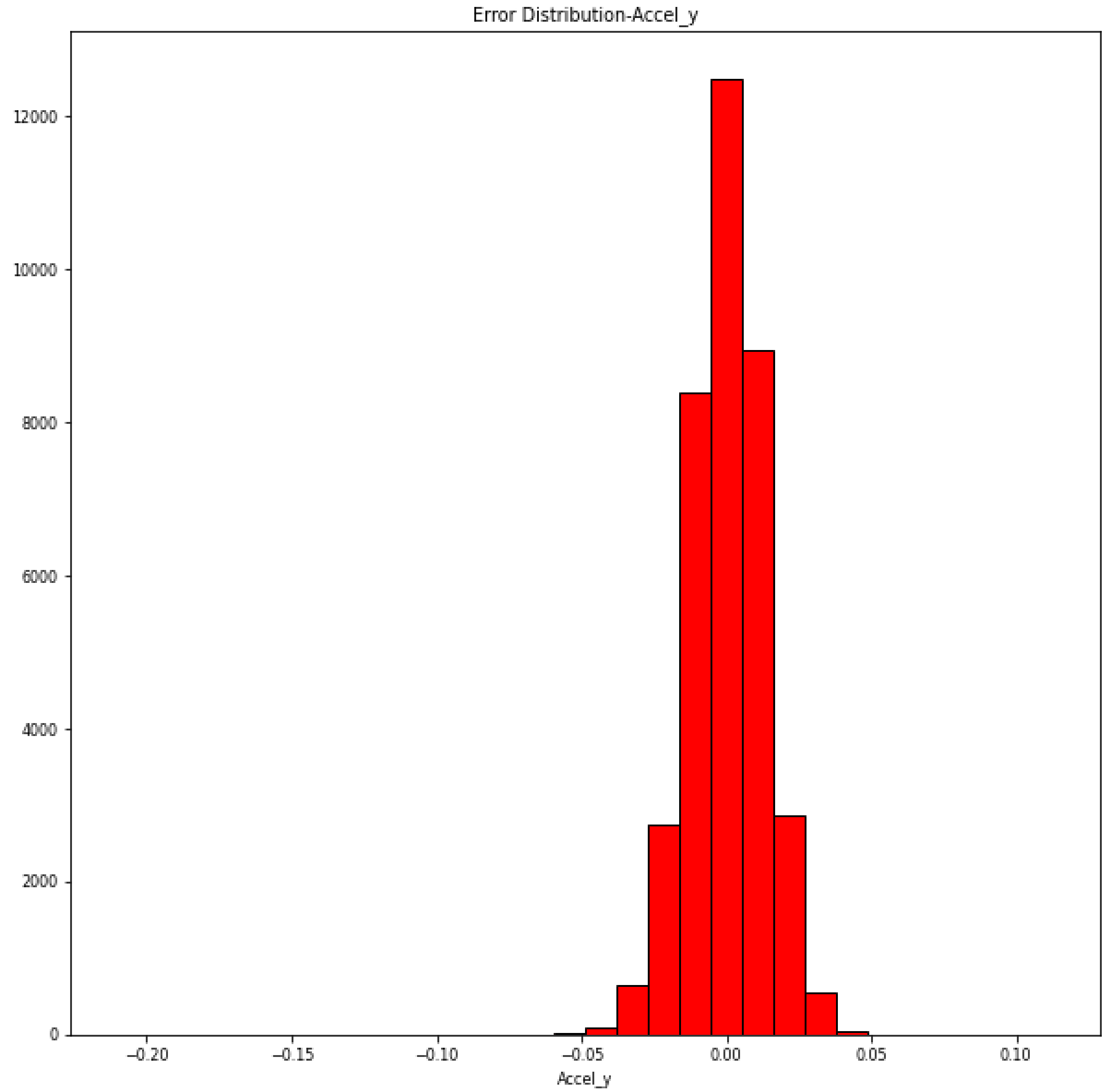
```
-0.5239382340147891
0.012759572892717352
```



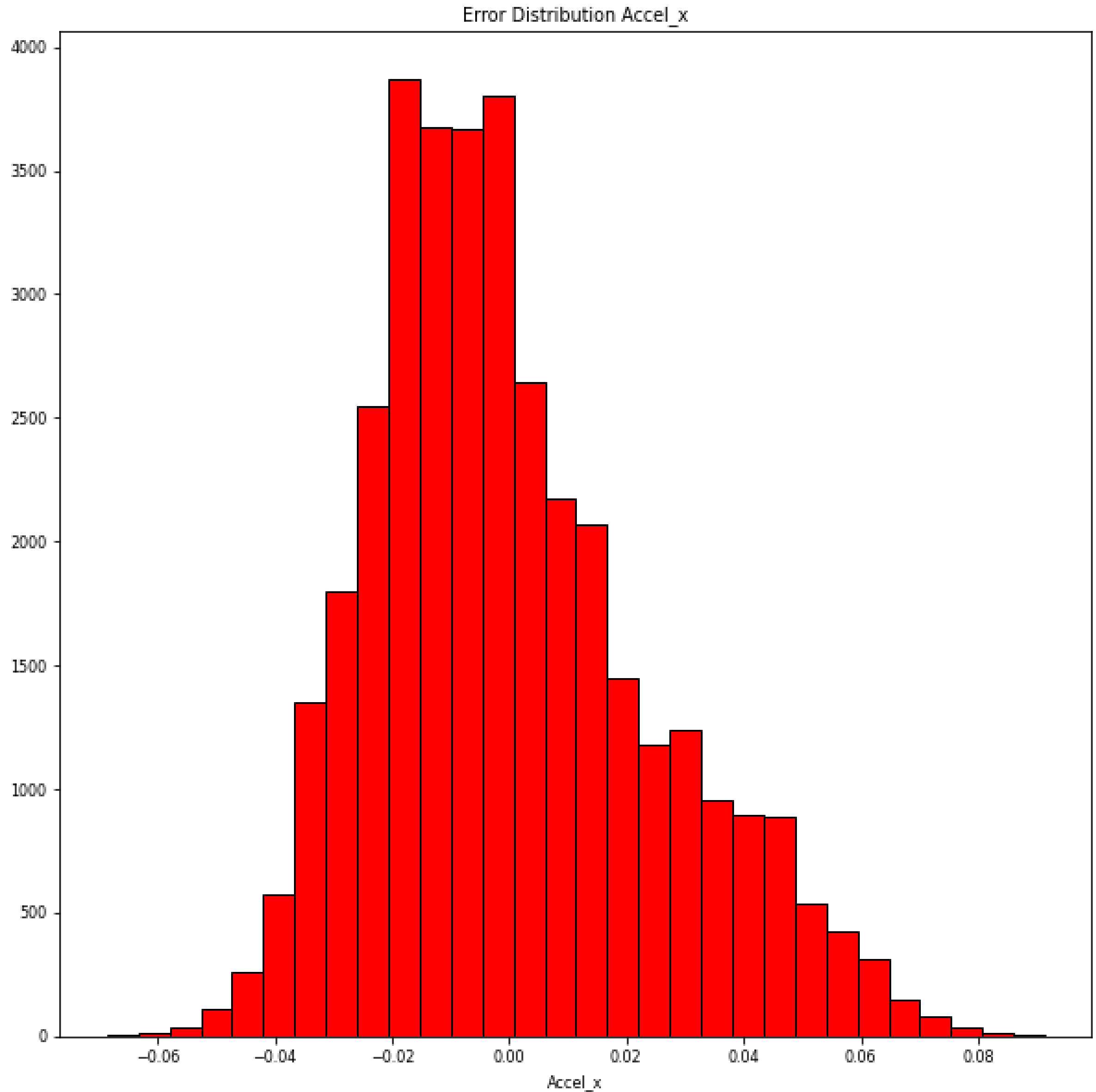

```
mn = numpy.mean(accelz)
print(mn)
sd = numpy.std(accelz)
print(sd)
```

```
-9.623409797738146
0.01858321624982835
```

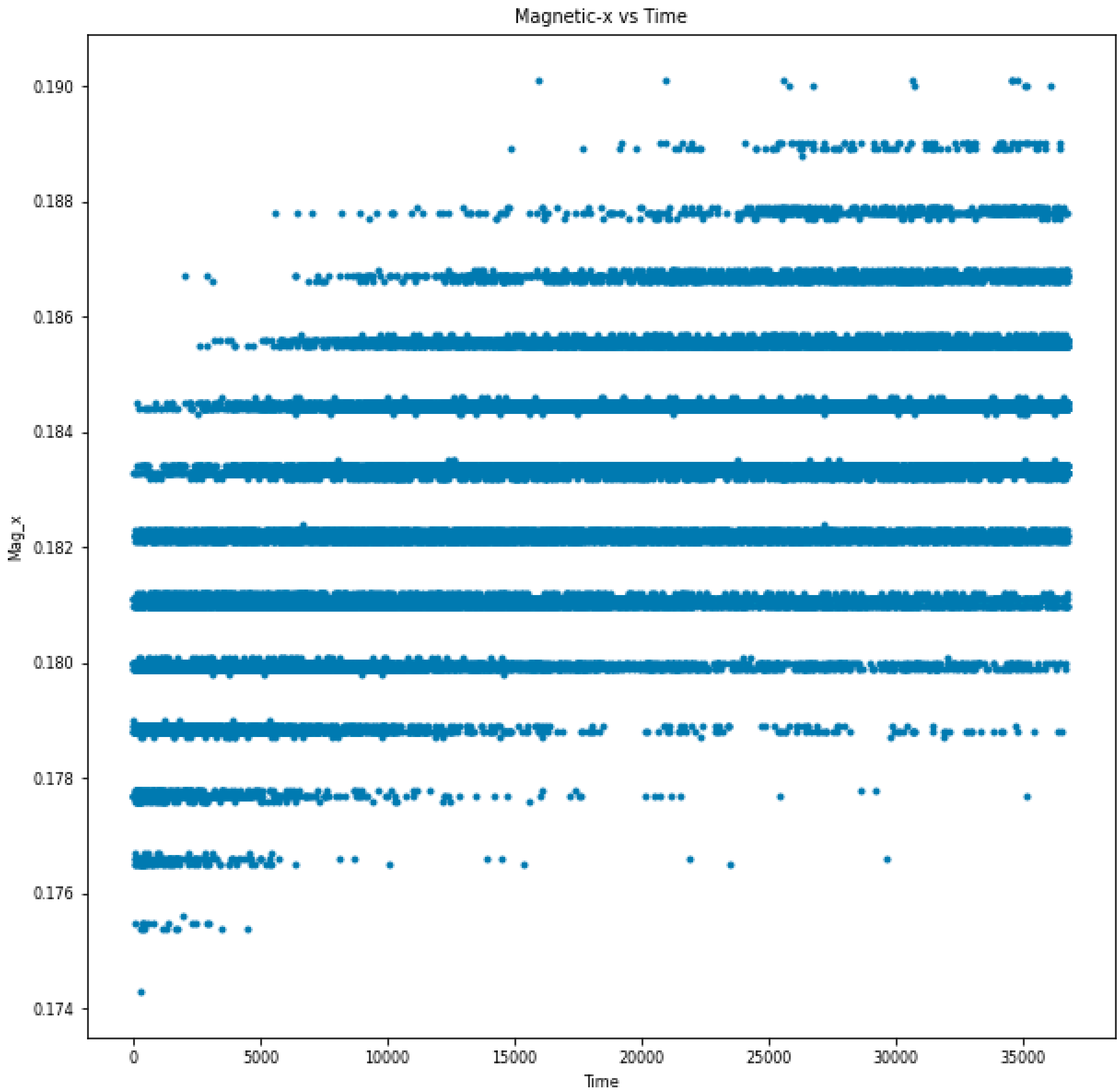

Error Distribution for Accel-z
17500
15000
12500
10000
7500
5000
2500
0
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
Accel_z



Error Distribution-Accel_y
12000
10000
8000
6000
4000
2000
0
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
Accel_y

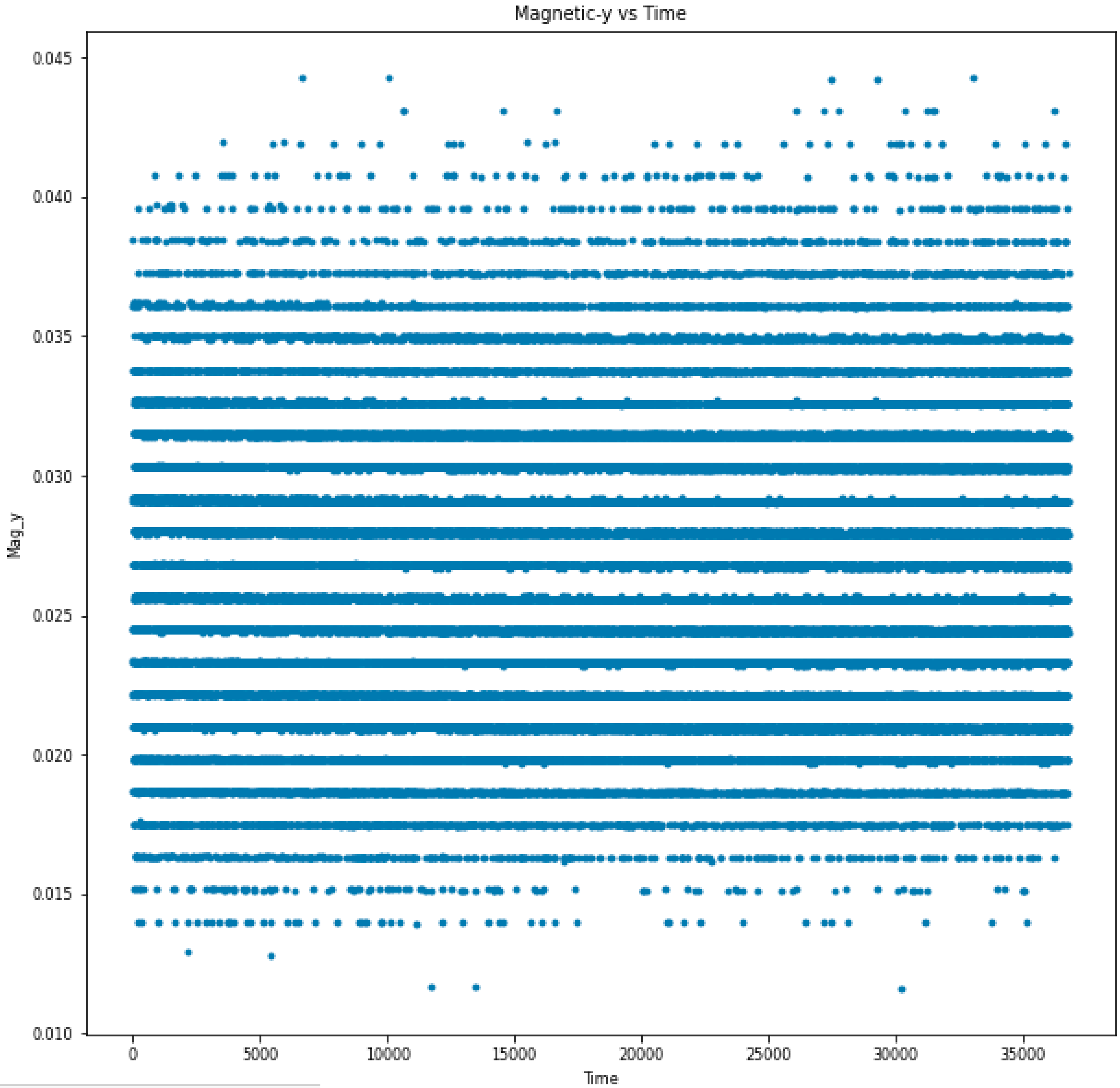


Error Distribution Accel_x
4000
3500
3000
2500
2000
1500
1000
500
0
−0.06
−0.04
−0.02
0.00
0.02
0.04
0.06
0.08
Accel_x



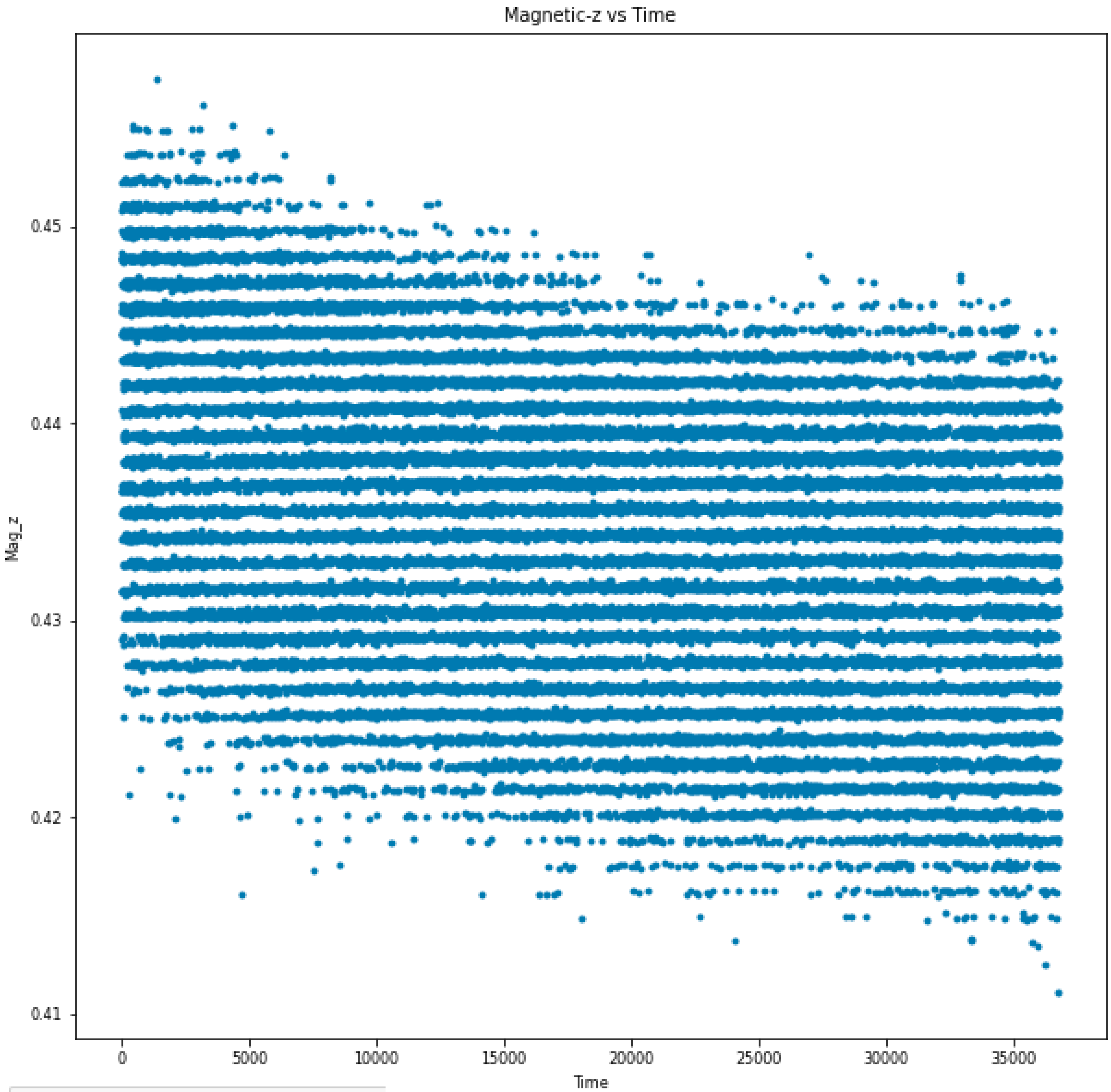

```
mn = numpy.mean(magx)
print(mn)
sd = numpy.std(magx)
print(sd)
```

```
0.1828973466724663
0.002168310340417806
```



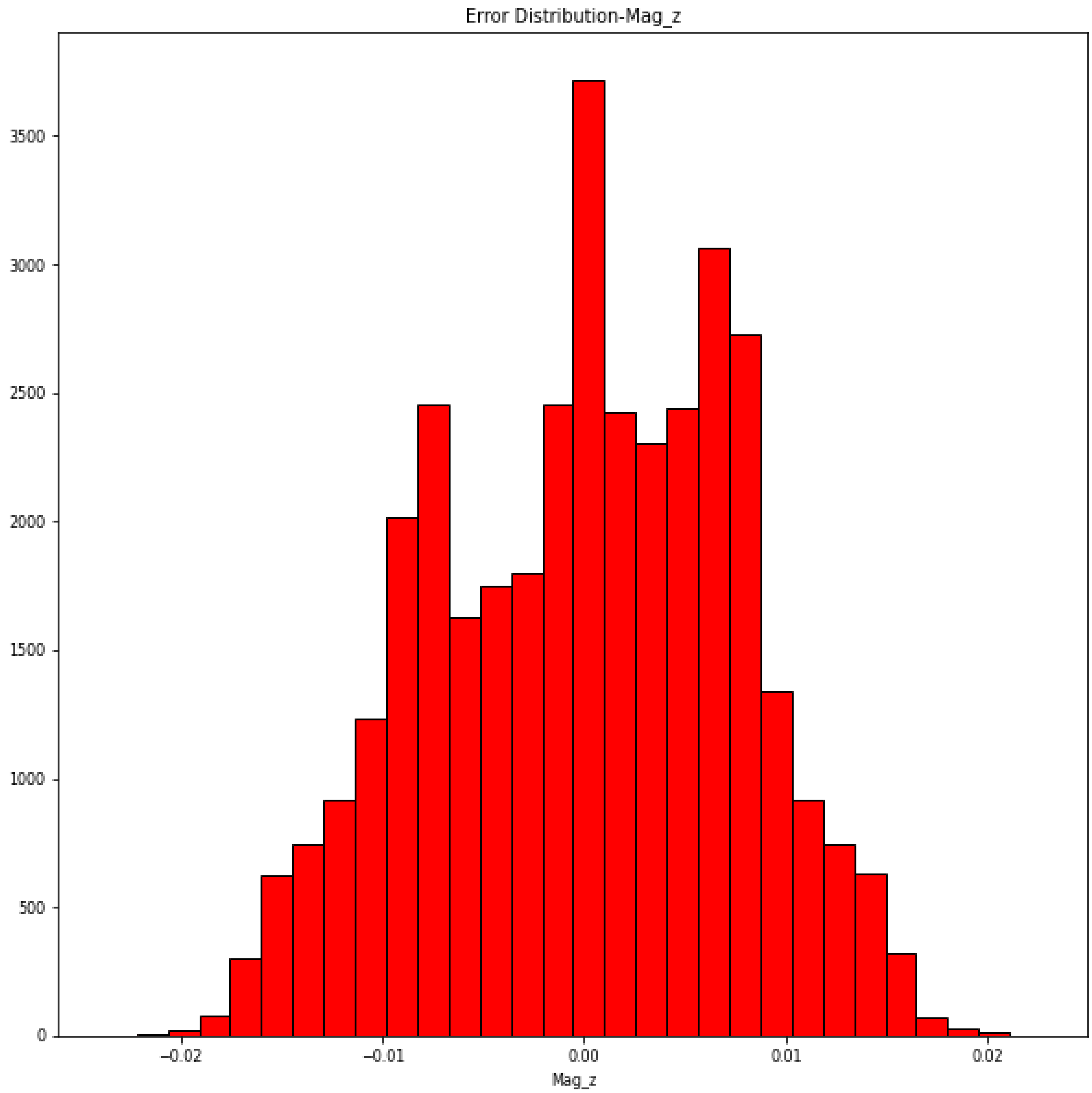

```
mn = numpy.mean(magy)
print(mn)
sd = numpy.std(magy)
print(sd)
```

```
0.02671468573292736
0.00544466781951626
```

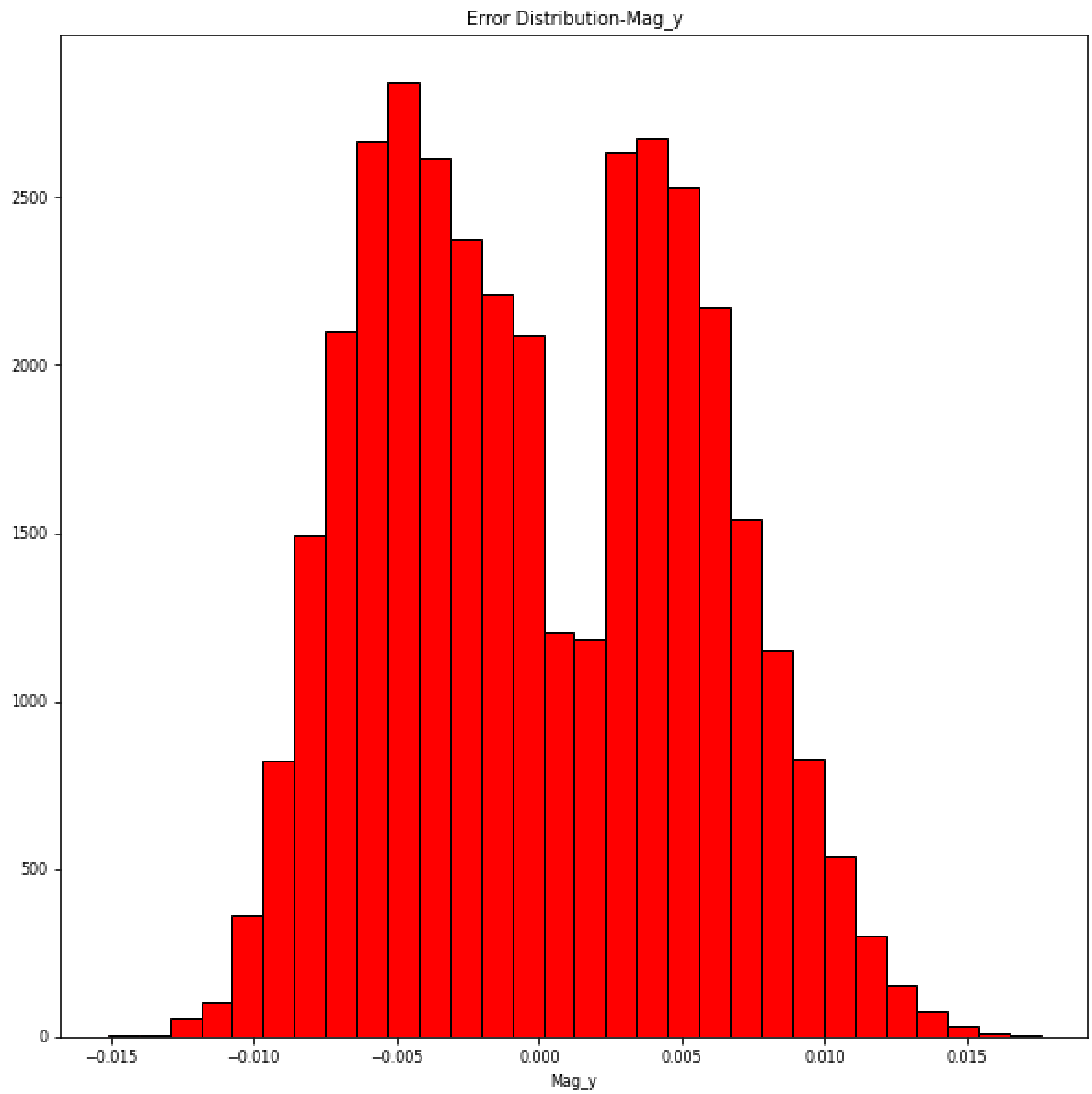


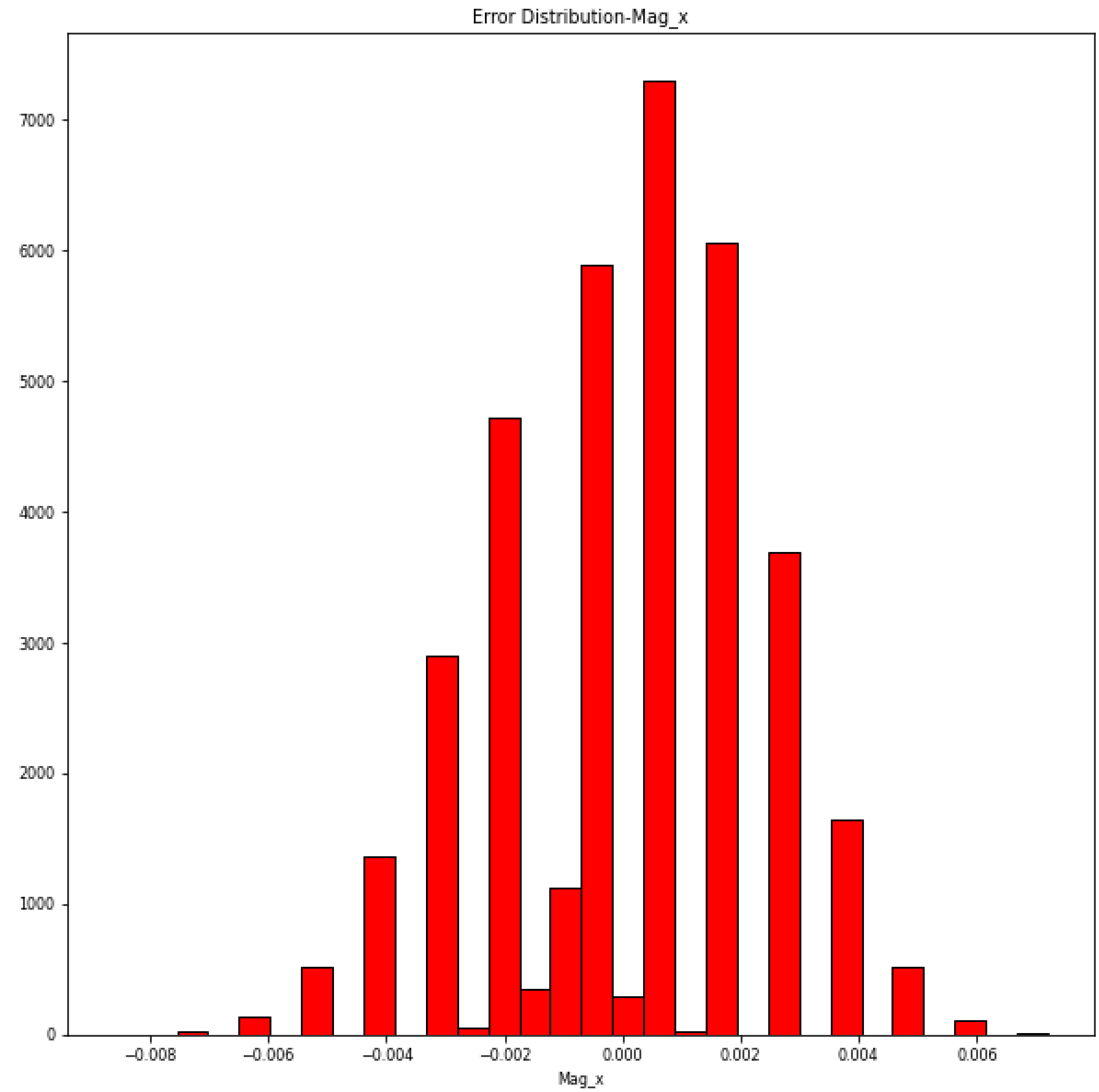

```
mn = numpy.mean(magz)
print(mn)
sd = numpy.std(magz)
print(sd)
```

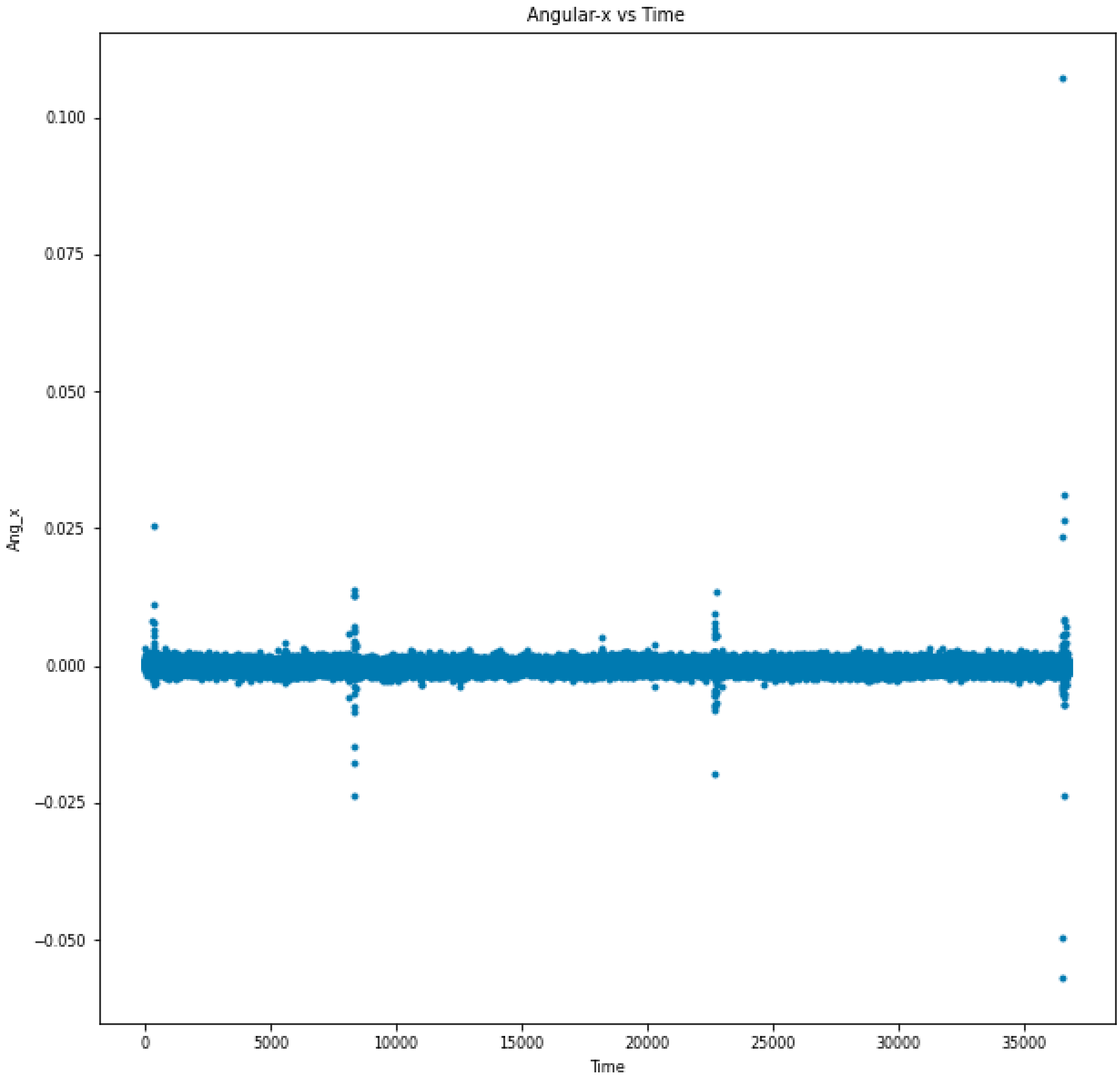
```
0.43484737929534584
0.007501904005388885
```



Error Distribution-Mag_z
3500
3000
2500
2000
1500
1000
500
0
−0.02
−0.01
0.00
0.01
0.02
Mag_z

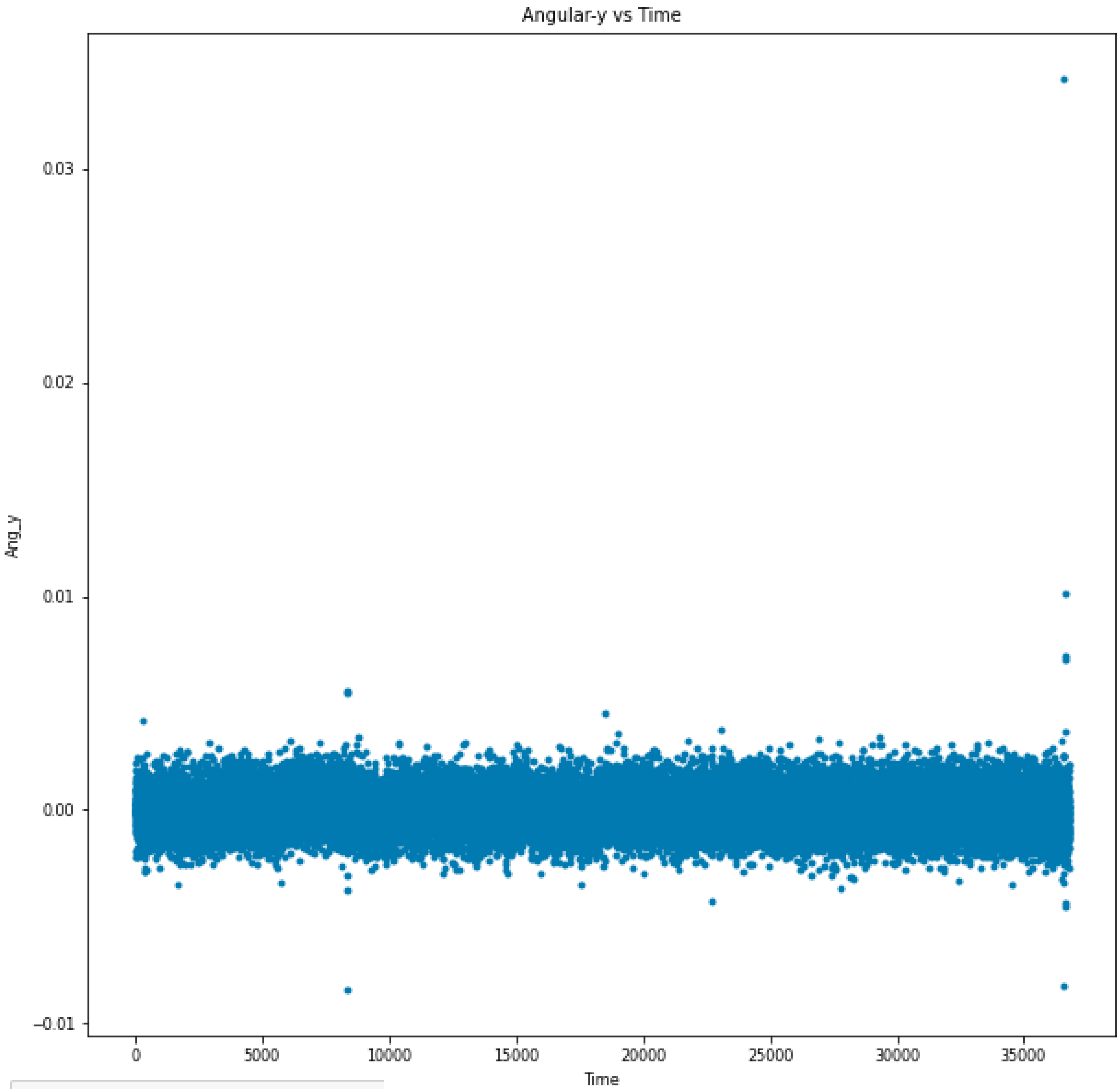

Error Distribution-Mag_y
0
500
1000
1500
2000
2500
−0.015
−0.010
−0.005
0.000
0.005
0.010
0.015
Mag_y


Error Distribution-Mag_x
7000
6000
5000
4000
3000
2000
1000
0
−0.008
−0.006
−0.004
−0.002
0.000
0.002
0.004
0.006
Mag_x



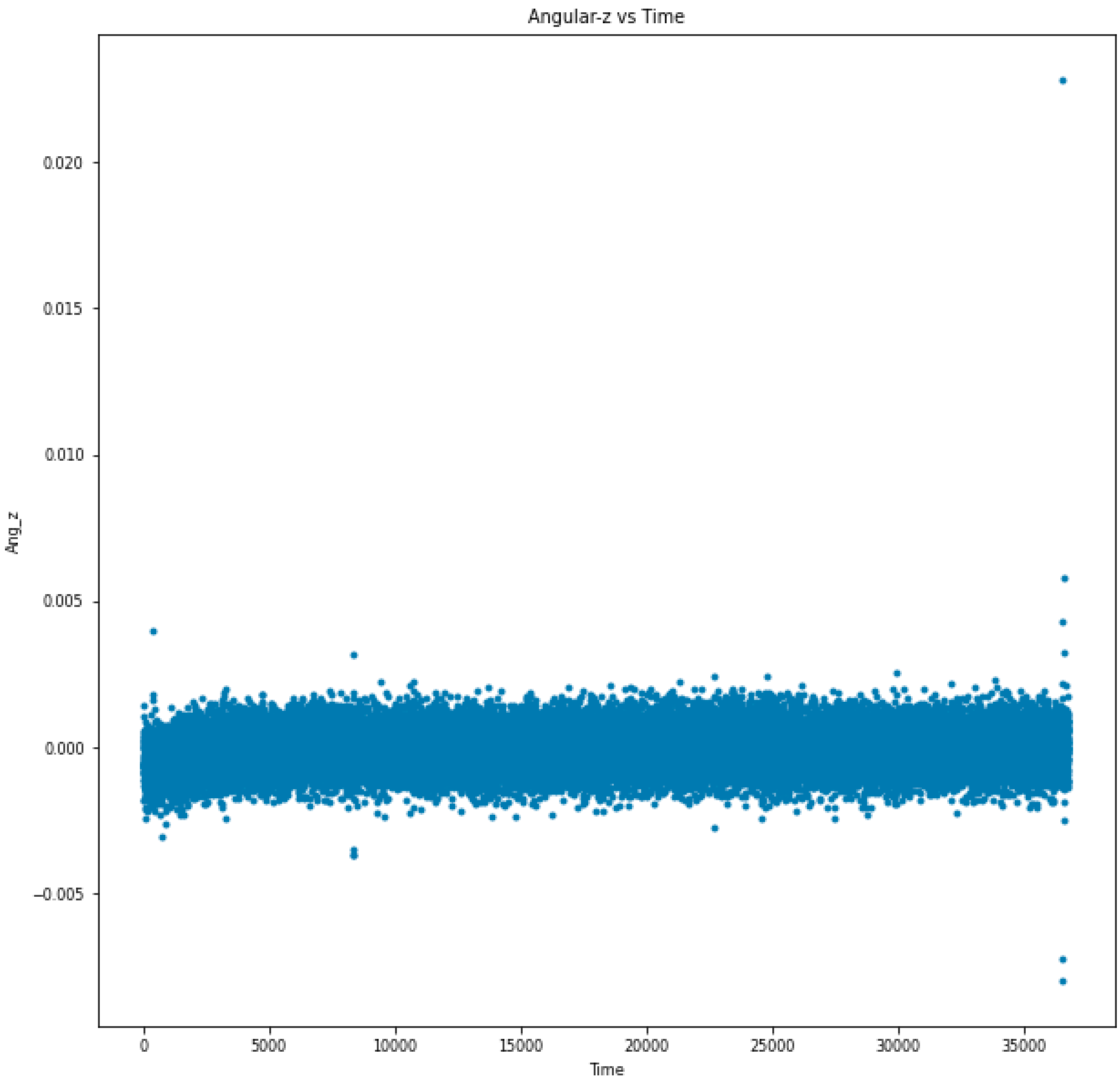

```
mn = numpy.mean(angx)
print(mn)
sd = numpy.std(angx)
print(sd)
```

```
1.7954273597216179e-06
0.0011404788799771603
```



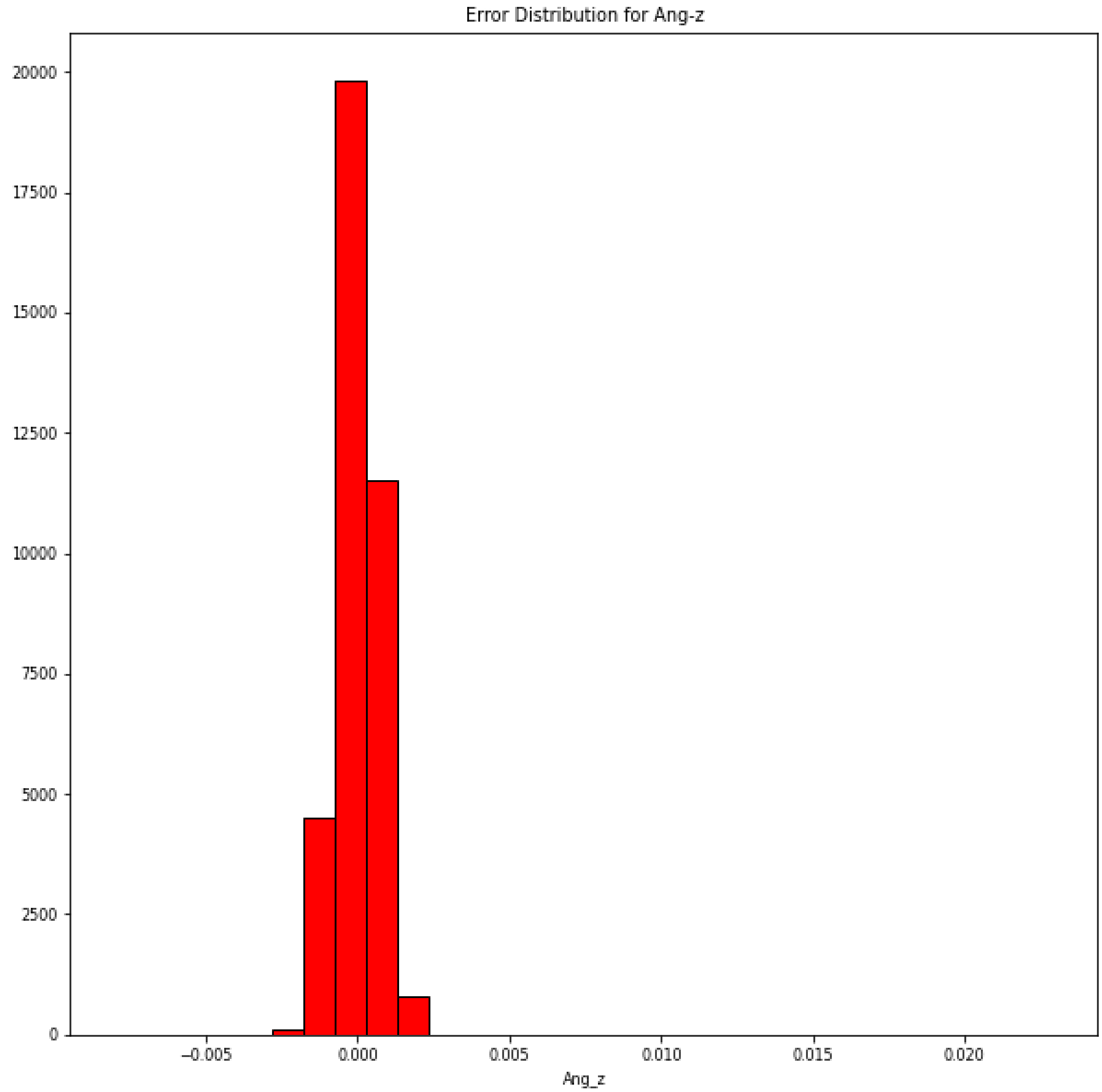

```
mn = numpy.mean(angy)
print(mn)
sd = numpy.std(angy)
print(sd)
```

```
5.941050456720313e-05
0.0009413898742092618
```

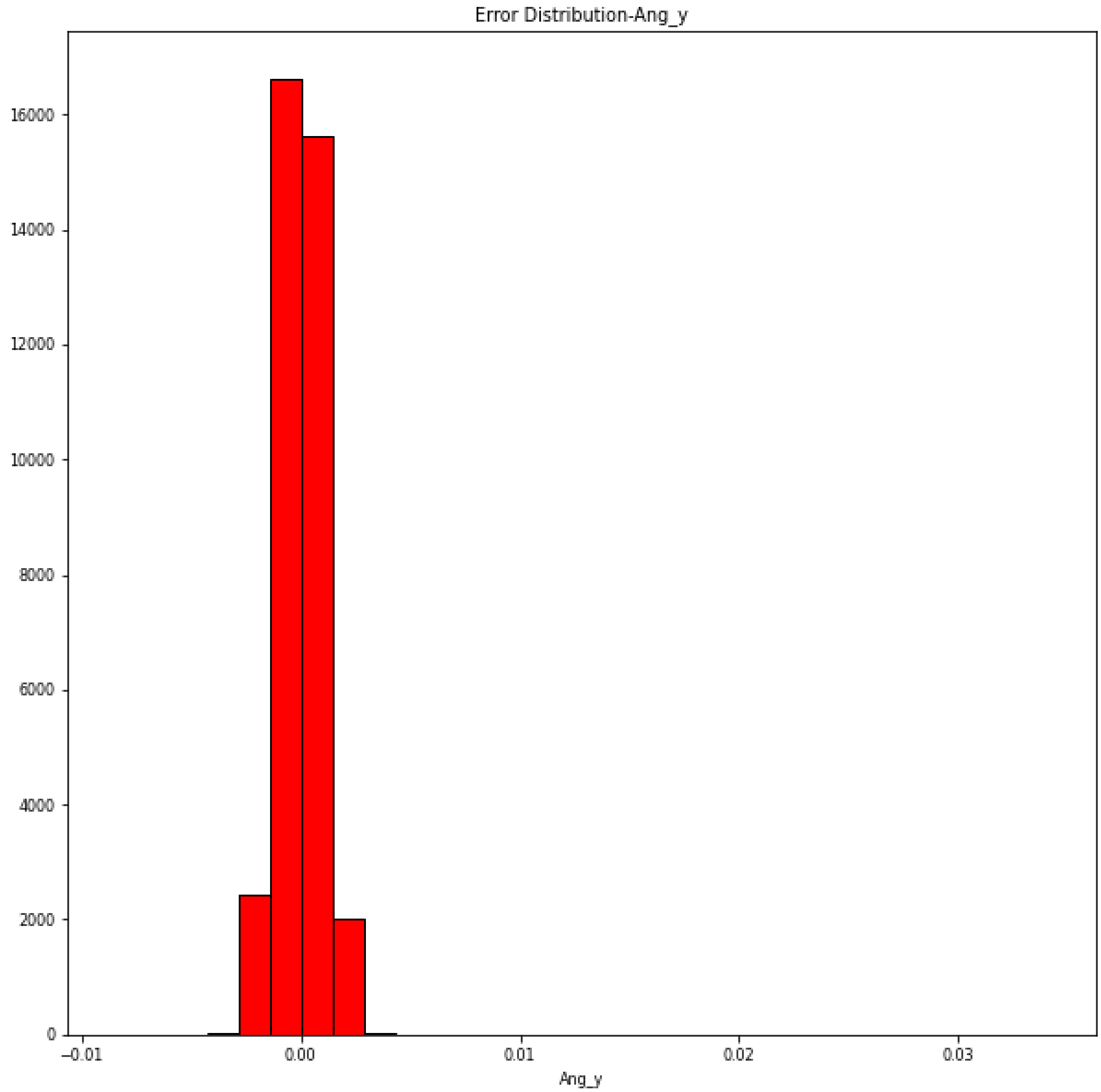


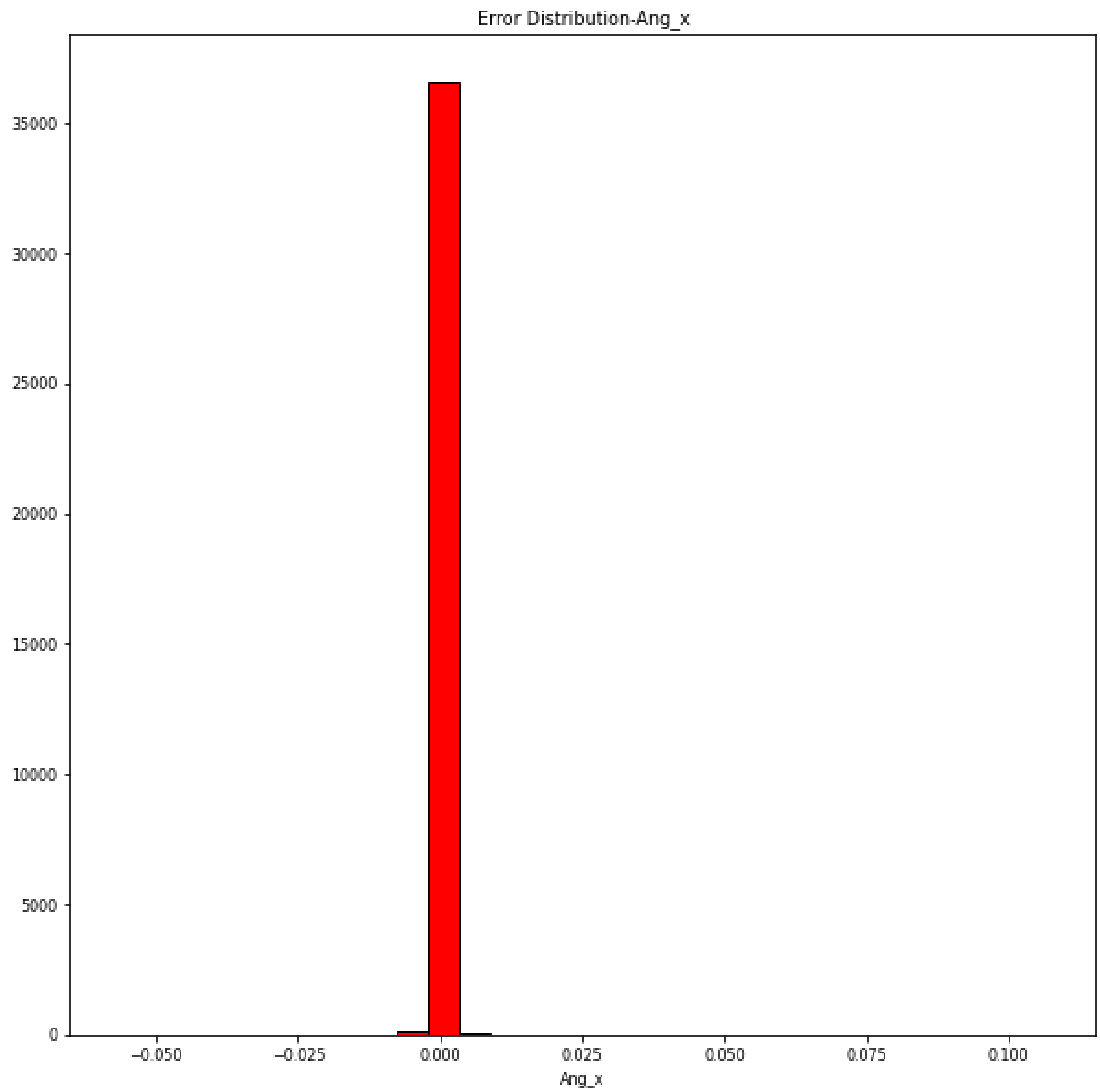

```
mn = numpy.mean(angz)
print(mn)
sd = numpy.std(angz)
print(sd)
```

```
-4.05189756415833e-05
0.0006644431459766903
```



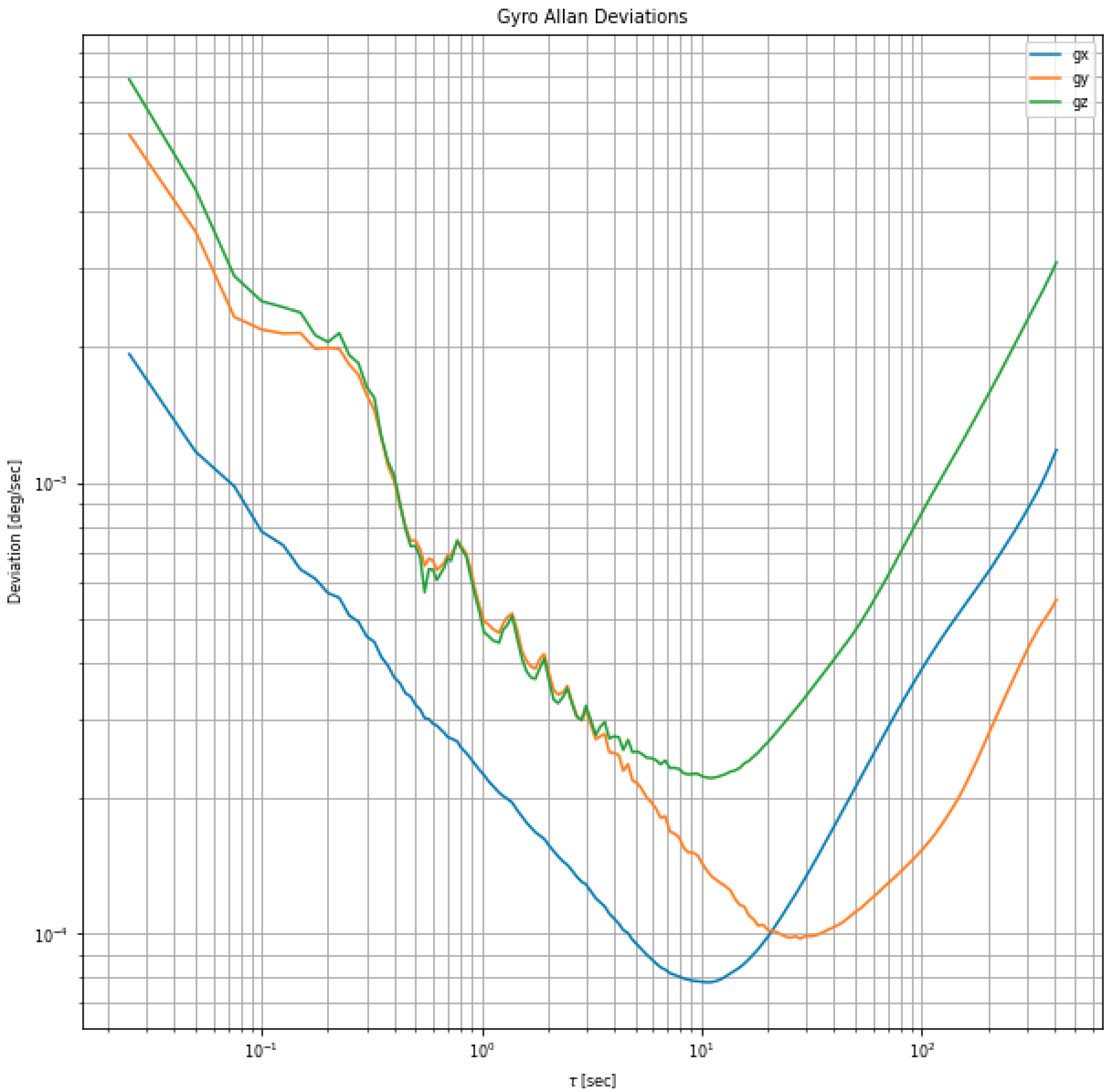
Error Distribution for Ang-z
20000
17500
15000
12500
10000
7500
5000
2500
0
−0.005
0.000
0.005
0.010
0.015
0.020
Ang_z


Error Distribution-Ang_y
16000
14000
12000
10000
8000
6000
4000
2000
0
−0.01
0.00
0.01
0.02
0.03
Ang_y



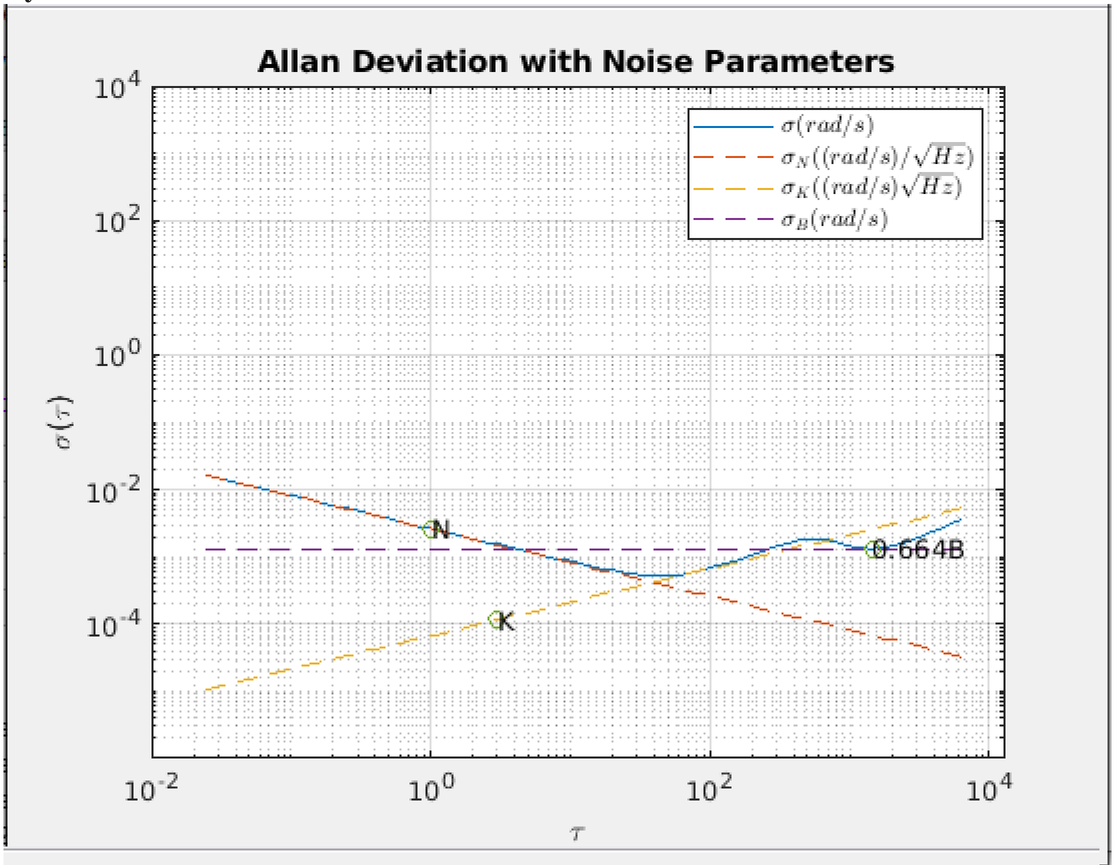
Error Distribution-Ang_x
35000
30000
25000
20000
15000
10000
5000
0
−0.050
−0.025
0.000
0.025
0.050
0.075
0.100
Ang_x

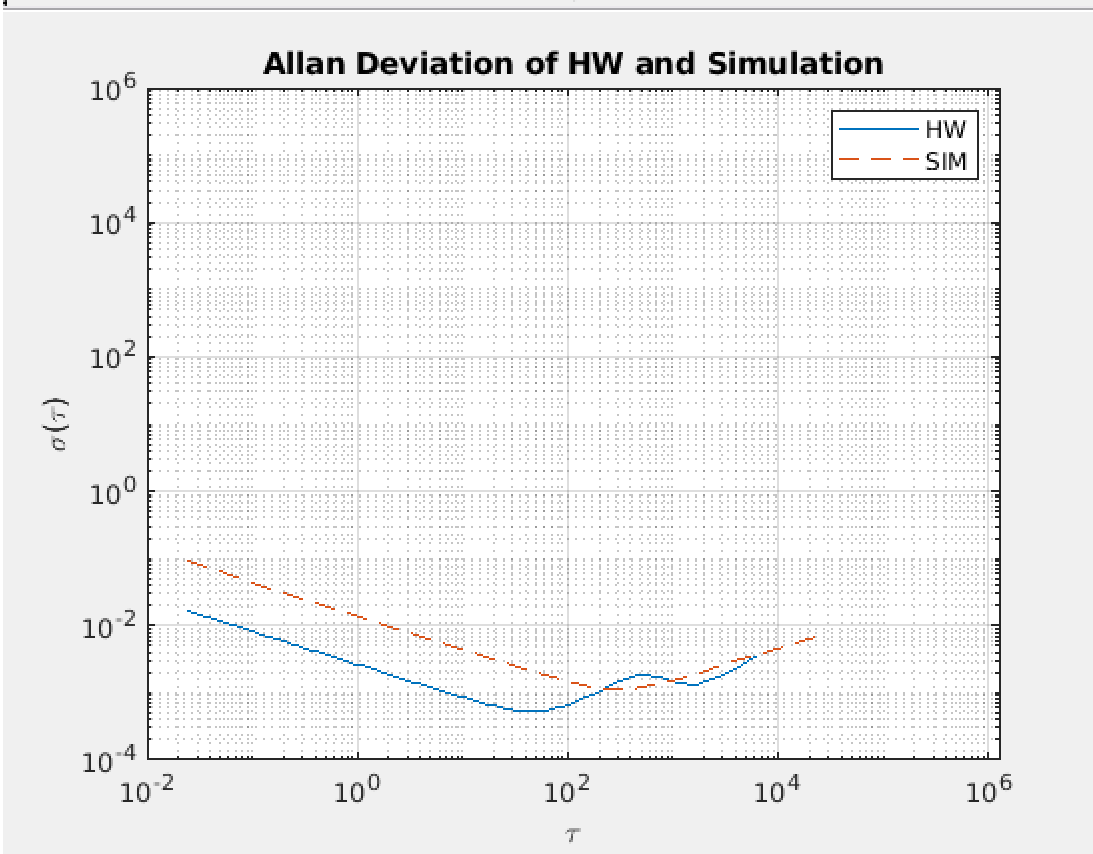
Gyro Allan Deviation for the 15 minute IMU data performed in python:



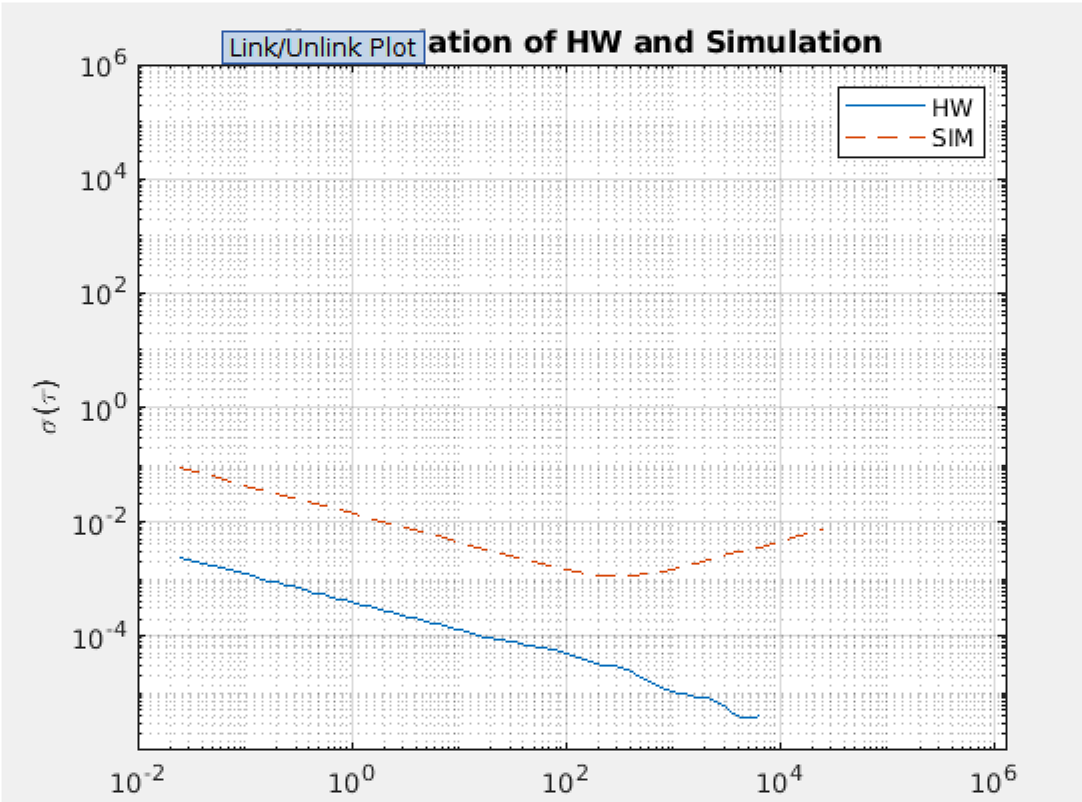
IMU Noise Characterization with Allan Variance performed in matlab for 5 hour data:

Gyro x:

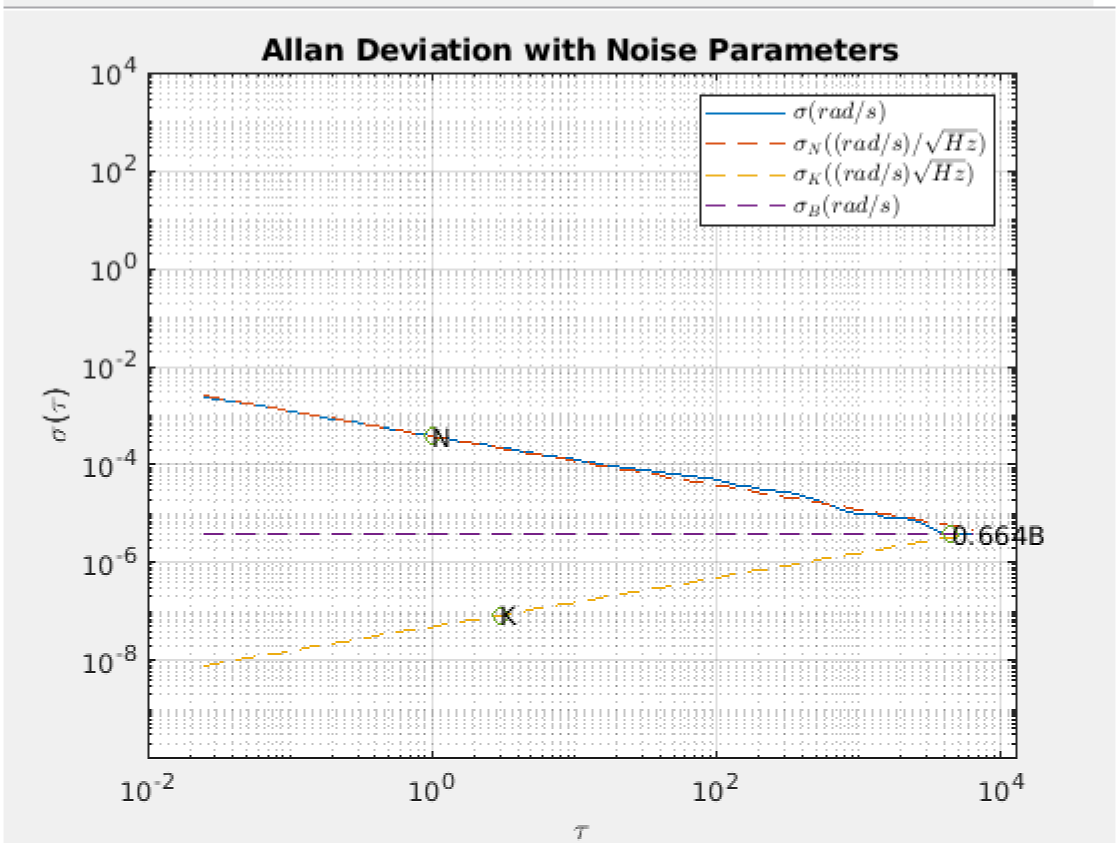

Allan Deviation with Noise Parameters
$\sigma(rad/s)$
$\sigma_N((rad/s)/\sqrt{Hz})$
$\sigma_K((rad/s)\sqrt{Hz})$
$\sigma_B(rad/s)$
N
K
0.664B
$\sigma(\tau)$
τ
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-2}
10^{0}
10^{2}
10^{4}


Allan Deviation of HW and Simulation
HW
SIM
$\sigma(\tau)$
τ
10^{6}
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-2}
10^{0}
10^{2}
10^{4}
10^{6}

Gyro y:

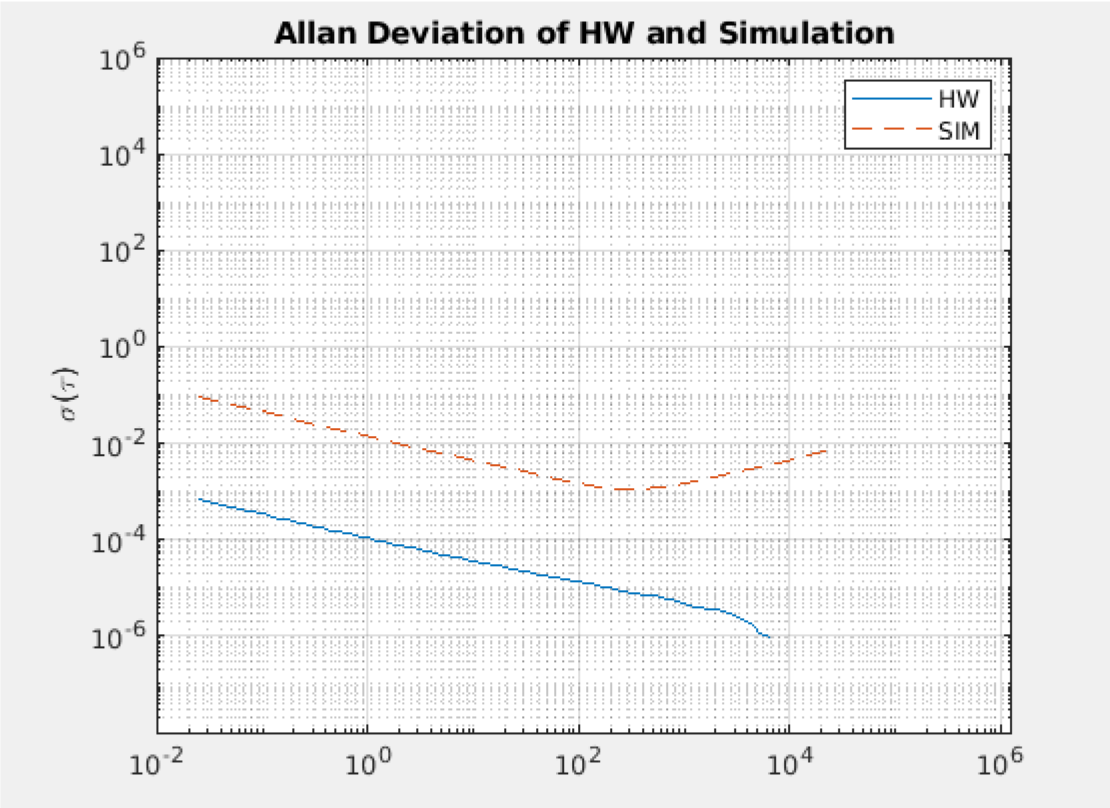


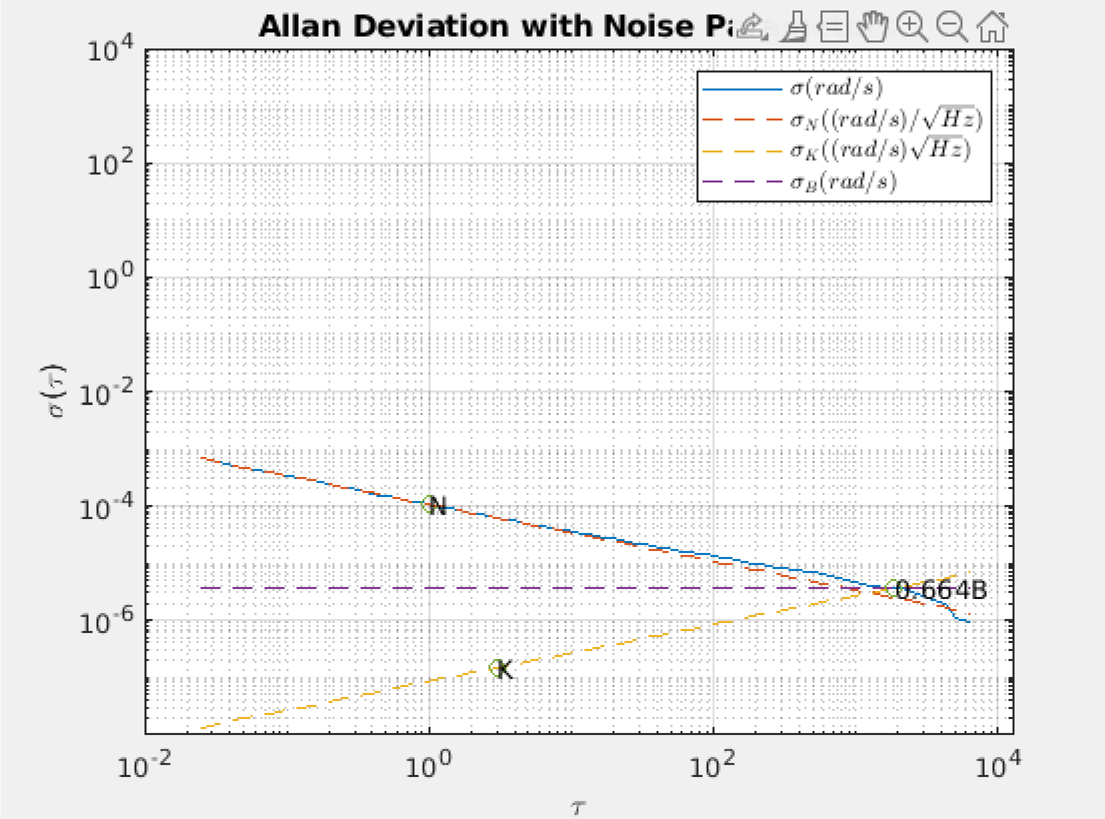
Link/Unlink Plot
ation of HW and Simulation
HW
SIM
$\sigma(\tau)$
10^{6}
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-2} 10^{0} 10^{2} 10^{4} 10^{6}



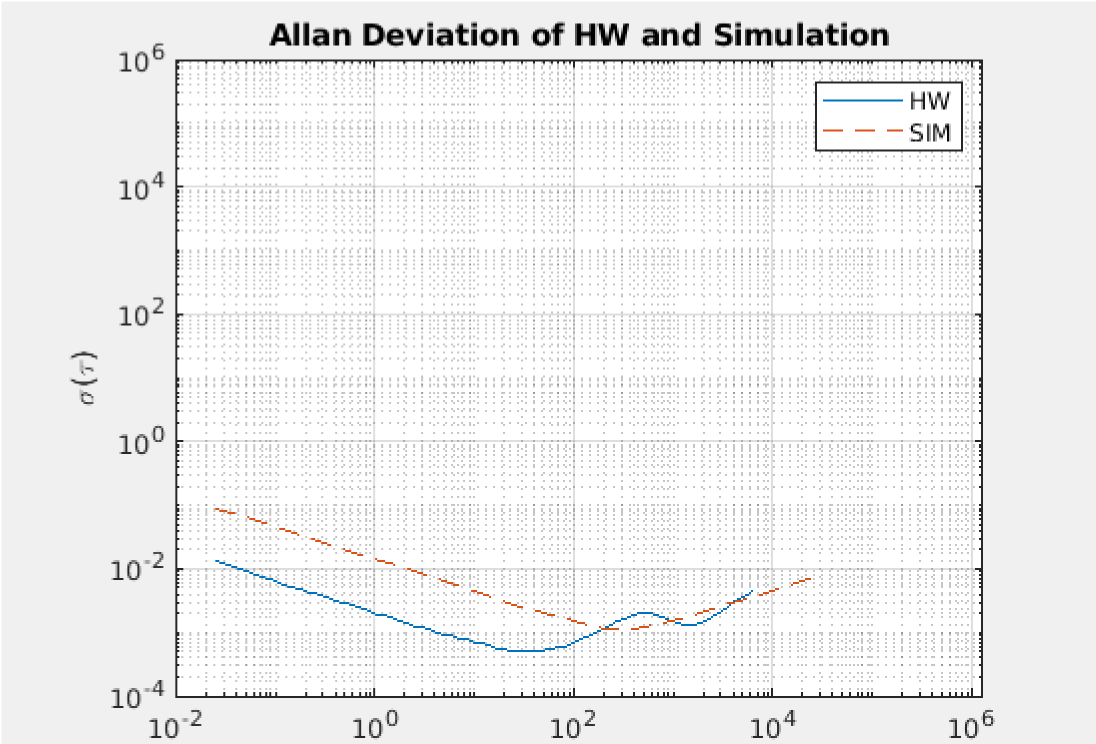
Allan Deviation with Noise Parameters
$\sigma(rad/s)$
$\sigma_N((rad/s)/\sqrt{Hz})$
$\sigma_K((rad/s)\sqrt{Hz})$
$\sigma_B(rad/s)$
N
K
0.664B
$\sigma(\tau)$
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-2} 10^{0} 10^{2} 10^{4}
τ

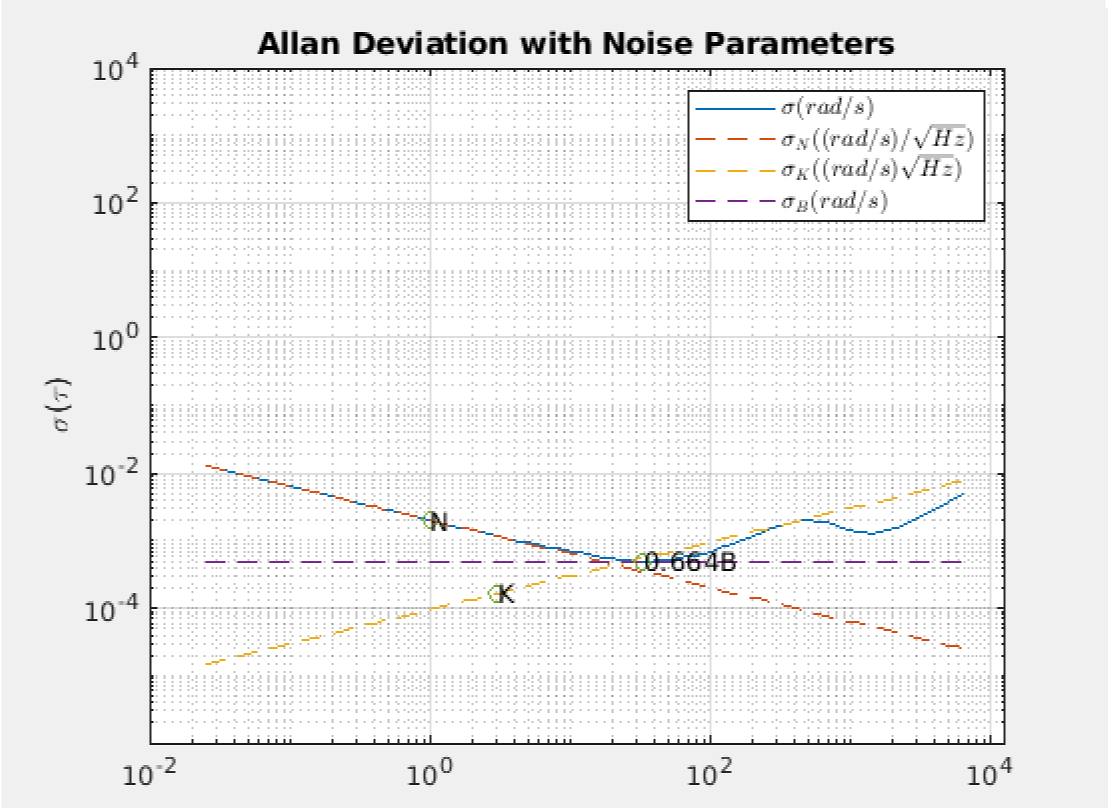
Gyro z:


Allan Deviation of HW and Simulation
HW
SIM
$\sigma(\tau)$
10^{6}
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}

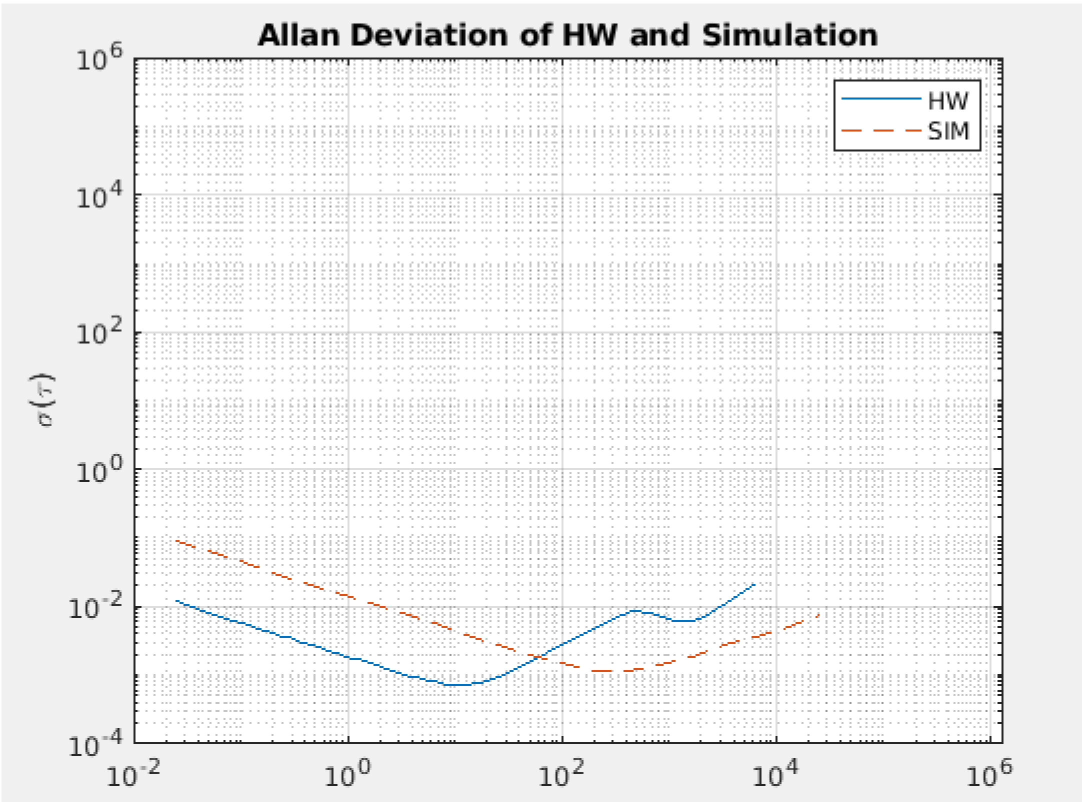

Allan Deviation with Noise P
$\sigma(rad/s)$
$\sigma_N((rad/s)/\sqrt{Hz})$
$\sigma_K((rad/s)\sqrt{Hz})$
$\sigma_B(rad/s)$
N
K
0.664B
$\sigma(\tau)$
τ
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}

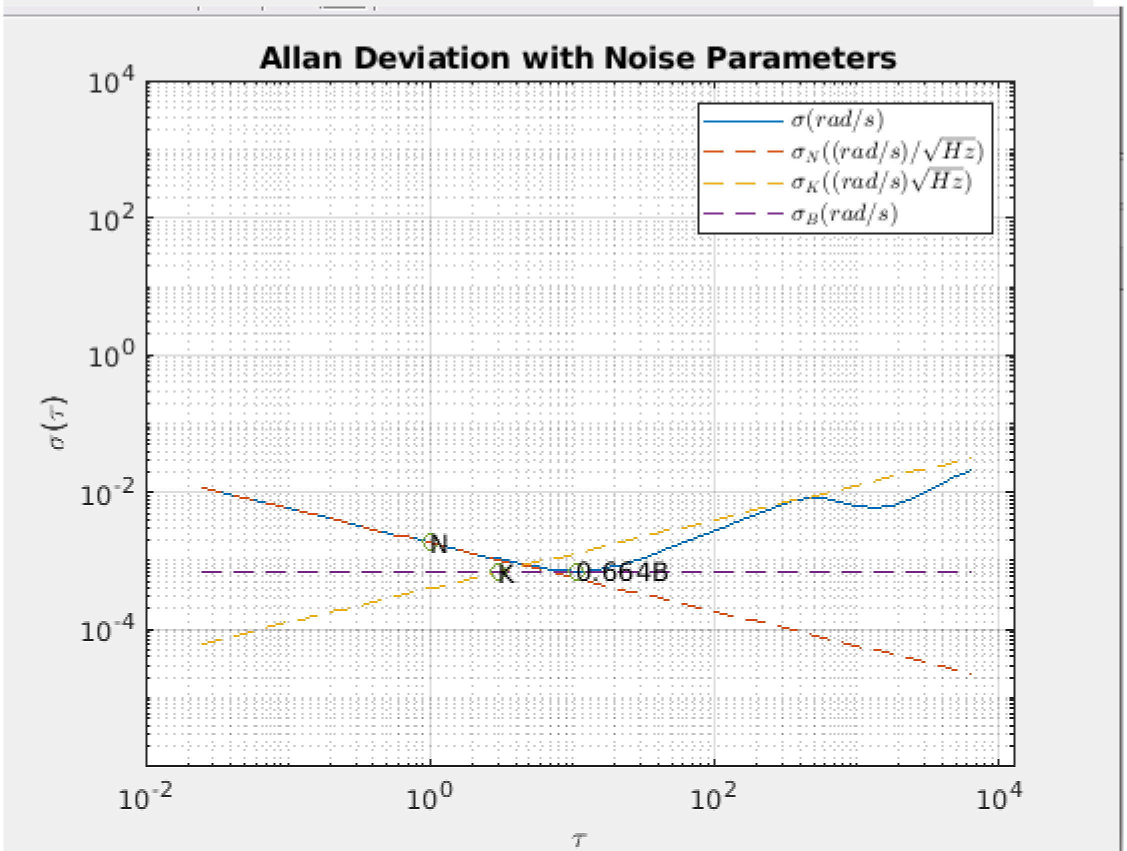
Acceleration x:





Acceleration y:





Acceleration z: