Recording Changes in Acceleration When Riding Over Various Terrains Using a Bike-Mounted Accelerometer

This is the first installment in a series of tests in which I use an accelerometer to collect data while performing various actions on my bike(s). The tests below were done to serve as a baseline for future experiments.

Experiment:

For this experiment, I will be riding my BMX bike over varying terrains (asphalt, sidewalk, and my backyard), and using Matlab Mobile to capture accelerometer data from each trial.

Setup:

For these first tests, I chose to attatch my phone to my 2018 Kink Whip, since its lack of suspension and small wheels would provide more visible feedback to the phone's accelerometer. Before securing the phone to my bike, I needed a way to make sure it was somewhat protected, as the phone by itself is pretty slippery and might scratch up the paint on my bike, so I wrapped the phone in some pipe insulation, and tightened the insulation around the phone with some wire.

[insert image of pipe insulation, wrapped phone]

After wrapping the phone, I used more wire and more pipe insulation to attatch the phone to the bike's top tube, seat tube, and down tube.

[insert image of wrapped phone attatched to bike, maybe diagram of bike tubes]

Before analyzing the data, it is important to note the accelerometer's axes, and which directions of acceleration will produce which values for each axis.

[insert image with orientation and positive/negative for phone acceleration axes]

Experiment 1: Asphalt

For the first test, I just rode my bike down the street in front of my house for about 20 yards. The road is almost perfectly smooth, so I would expect the acceleration values to have the least amount of variation out of all of the tests.

The data was stored in a file called "asphalt001_Kink40psi.mat". The first step to bein able to analyze the results is to import the data into Matlab,

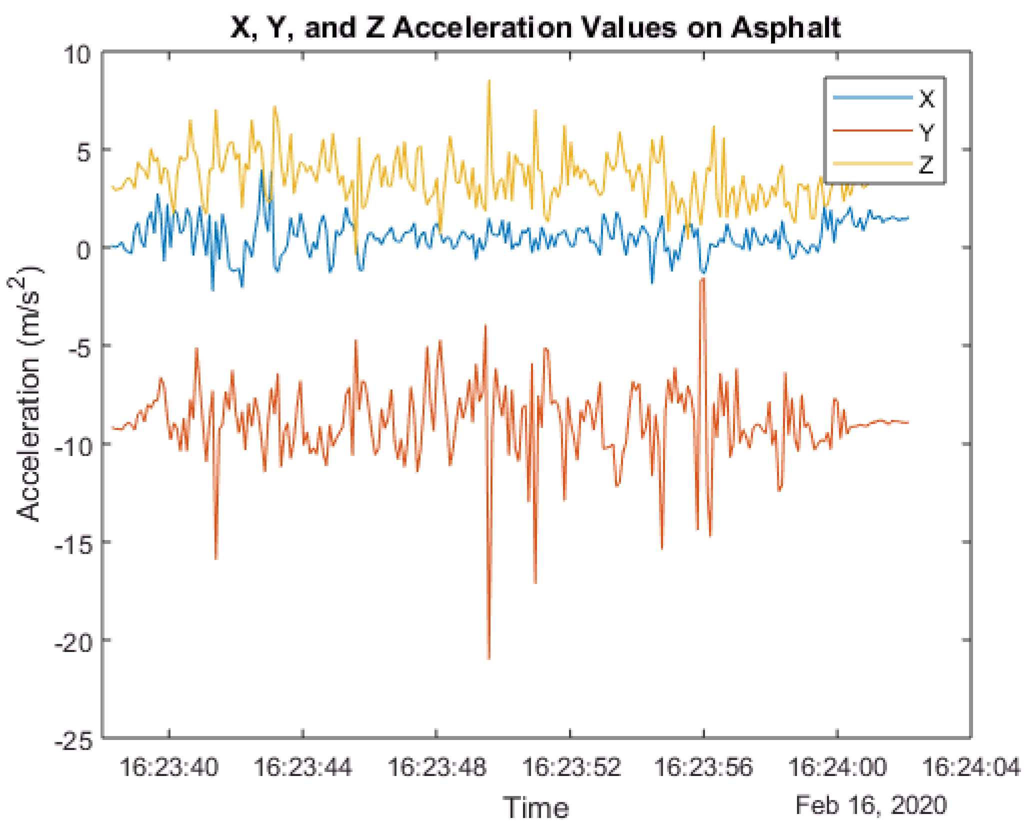
```
load asphalt001_Kink40psi.mat
```

This .mat file contains all of the variables recorded during the trial, in the form of a matrix. During our analysis, I will be focusing on acceleration and time. The values for each of the individual axes, as well as time, can be imported into their own variables, so that when future experiments are loaded, the values are still saved.

```
xAsphalt = Acceleration.X;
yAsphalt = Acceleration.Y;
zAsphalt = Acceleration.Z;
tAsphalt = Acceleration.Timestamp;
```

Using just this raw data, I can already plot a figure.

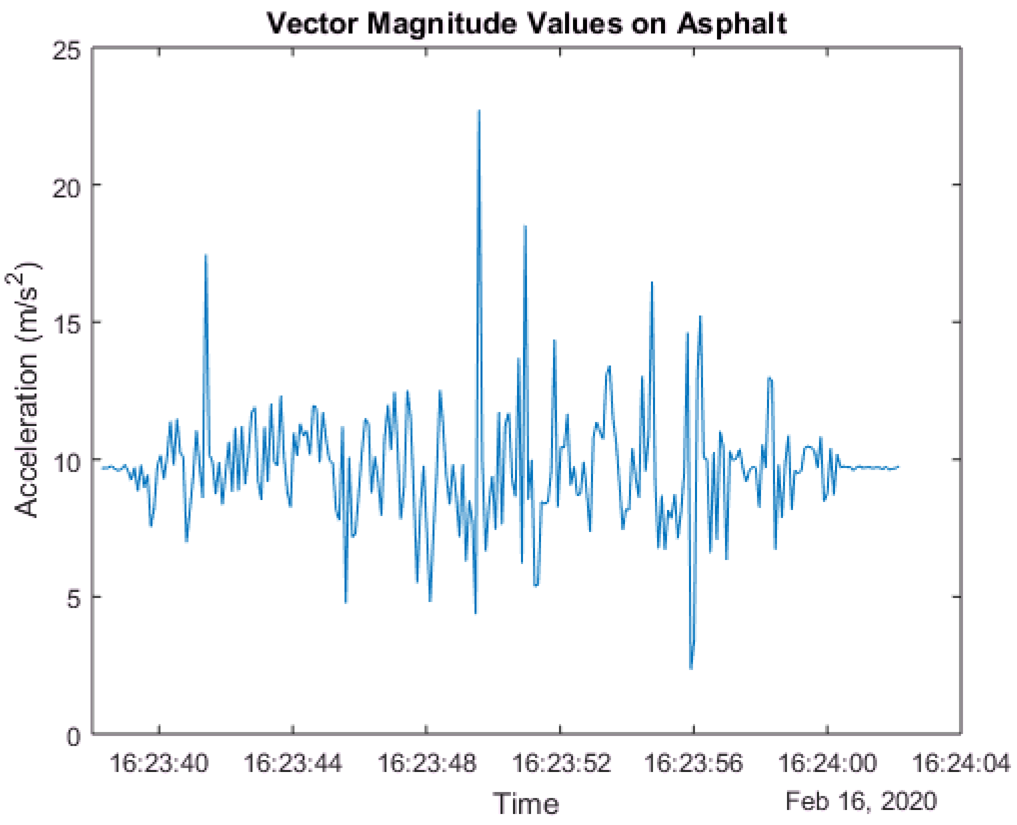
```
asphaltAllDirections = [xAsphalt, yAsphalt, zAsphalt];

figure; plot(tAsphalt, asphaltAllDirections)
title('X, Y, and Z Acceleration Values on Asphalt')
xlabel('Time')
ylabel('Acceleration (m/s^2)')
legend('X', 'Y', 'Z')
```



One unexpected thing I immediately noticed is that each of the different axes have nonzero acceleration values. While this may seem wrong at first glance, it makes sense if you consider the effect gravity has on each of the axes. Y is most strongly affected due to the phone's orientation, and z is slightly affected, since the phone is not completely perpendicular to the ground.

Using the Pythagorean Theorem, I can find the magnitude of the overall vector, and then plot it to another graph.

```
netAsphaltAllDirections = sqrt(xAsphalt.^2 + yAsphalt.^2 + zAsphalt.^2);

figure; plot(tAsphalt, netAsphaltAllDirections)
title('Vector Magnitude Values on Asphalt')
xlabel('Time')
ylabel('Acceleration (m/s^2)')
```



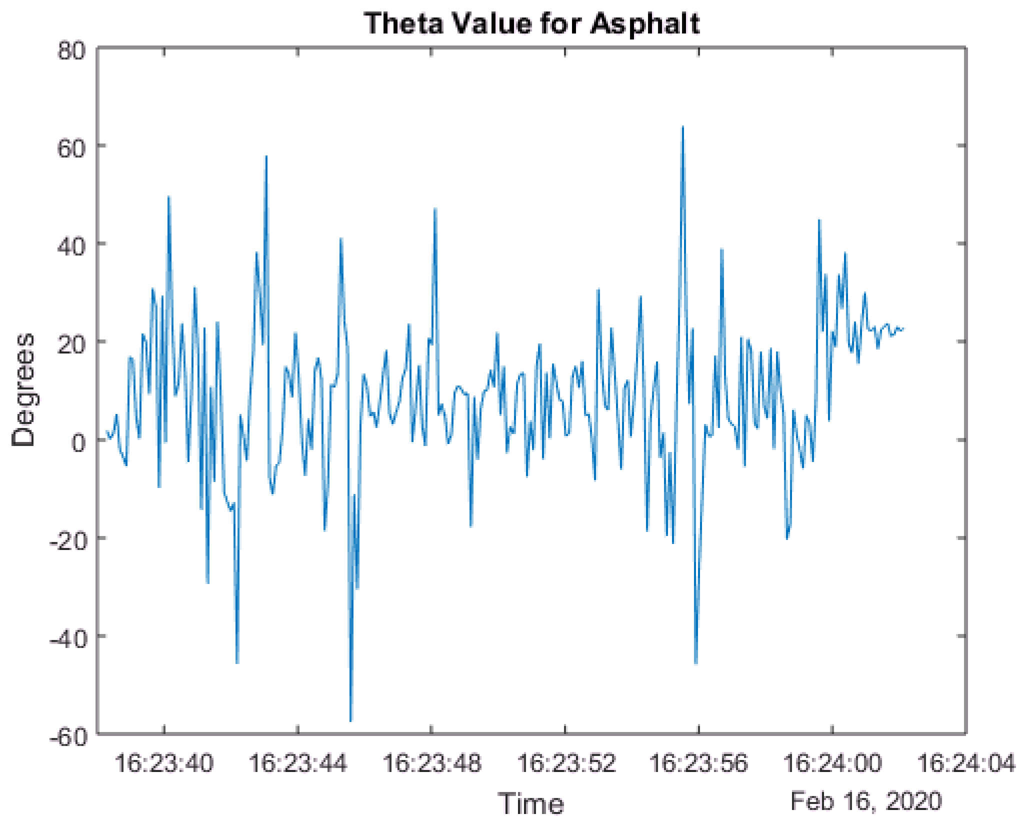
This graph shows where the major disturbances were. Overall, the vector magnitude remains relatively constant.

Using the accelerometer data, I can also find the coordinates for if I were to plot the vector in spherical polar form: (ρ, θ, ϕ)

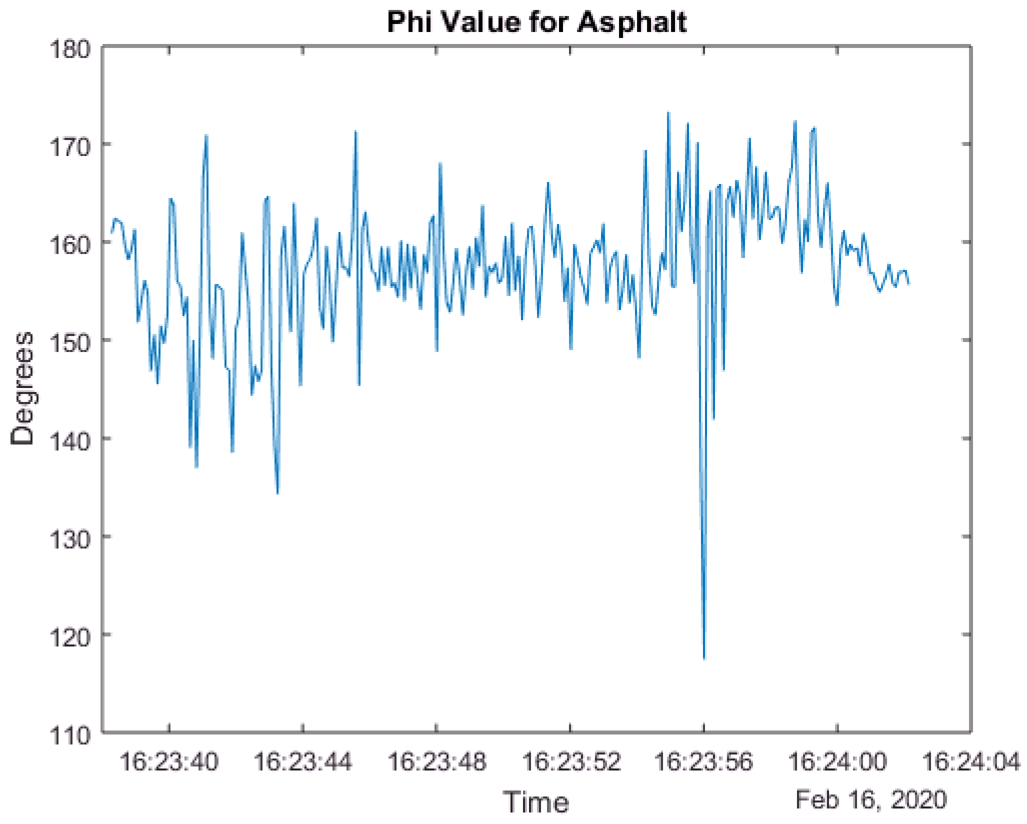
```
rhoAsphalt = sqrt(xAsphalt.^2 + yAsphalt.^2 + zAsphalt.^2);
thetaAsphalt = atand(xAsphalt./zAsphalt);
phiAsphalt = acosd(yAsphalt./rhoAsphalt);
```

I can then plot each of these as a function of time. (rhoAsphalt is the same as the vector magnitude graph, so I won't replot it)

```
figure; plot(tAsphalt, thetaAsphalt)
title('Theta Value for Asphalt')
xlabel('Time')
ylabel('Degrees')
```



```
figure; plot(tAsphalt, phiAsphalt)
title('Phi Value for Asphalt')
xlabel('Time')
ylabel('Degrees')
```



The theta graph does not show much significant information. Since the phone was somewhat perpendicular to the ground already, the theta value was all over the place.

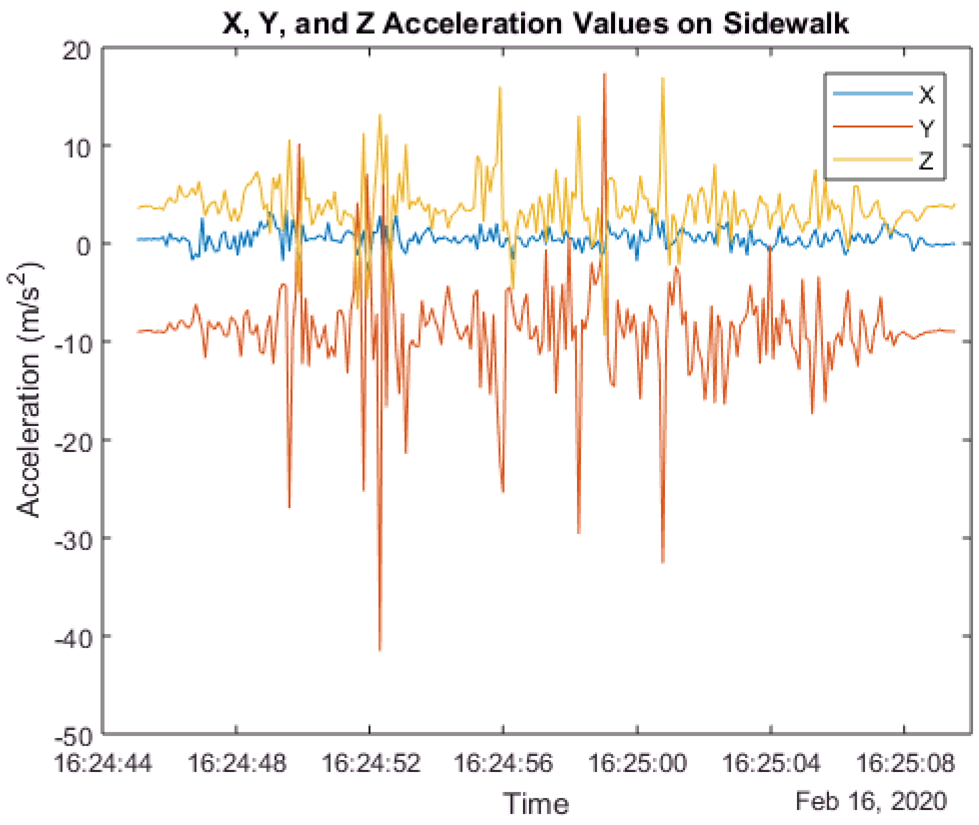
Since gravity was the major constant source of acceleration throughout the test, it makes sense that the phi graph angle points the vector downward. One thing I noticed was that the spike corresponds with when the y value on the first graph is highest, which makes sense.

Experiment 2: Sidewalk

For this test, I rode my bike down the sidewalk in front of my house for about 20 yards. The sidewalk is much less smooth than the road, so I would expect the acceleration values to have more variation than the asphalt values.

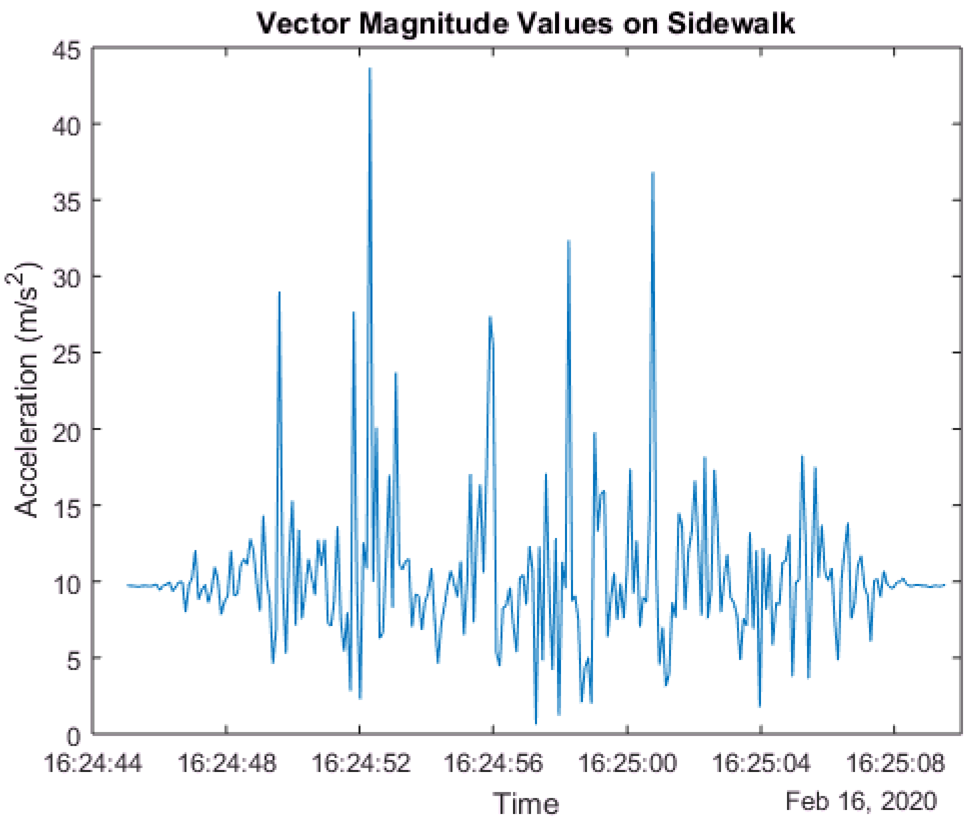
I repeated what I did for the asphalt analysis.

```
load sidewalk001_Kink40psi.mat

xSidewalk = Acceleration.X;
ySidewalk = Acceleration.Y;
zSidewalk = Acceleration.Z;
tSidewalk = Acceleration.Timestamp;


sidewalkAllDirections = [xSidewalk, ySidewalk, zSidewalk];

figure; plot(tSidewalk, sidewalkAllDirections)
title('X, Y, and Z Acceleration Values on Sidewalk')
xlabel('Time')
ylabel('Acceleration (m/s^2)')
legend('X', 'Y', 'Z')
```



I can already see that the acceleration values vary more than the asphalt values, especially for the y-axis.

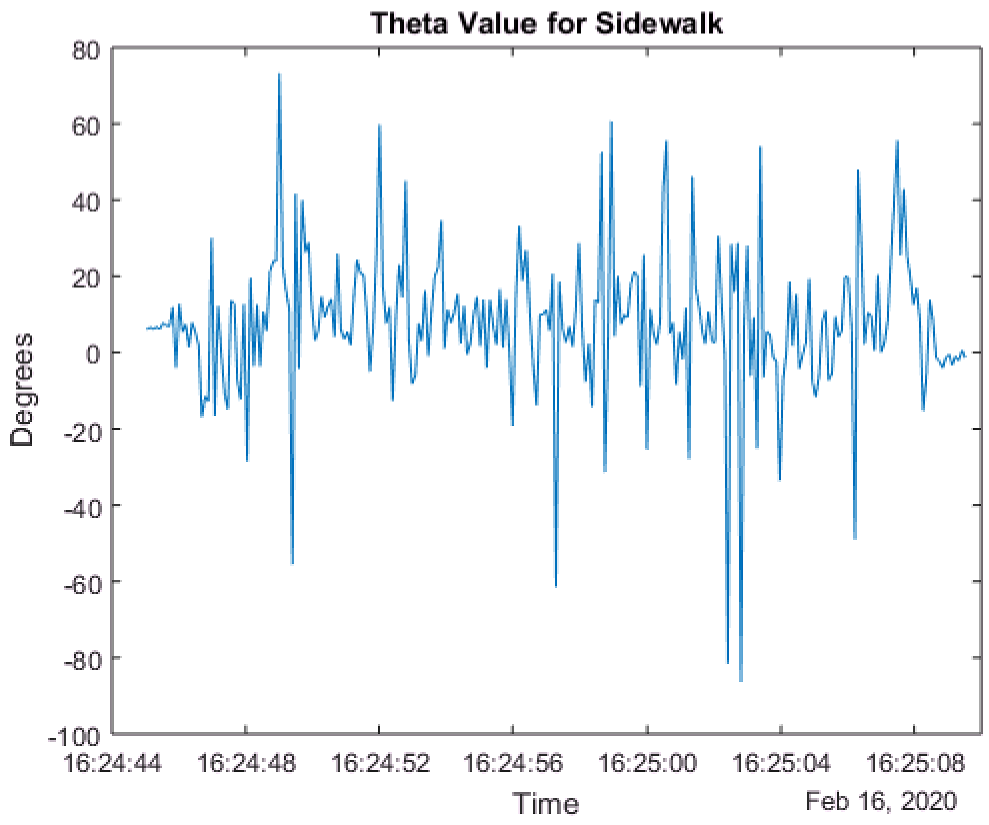
```
netSidewalkAllDirections = sqrt(xSidewalk.^2 + ySidewalk.^2 + zSidewalk.^2);

figure; plot(tSidewalk, netSidewalkAllDirections)
title('Vector Magnitude Values on Sidewalk')
xlabel('Time')
ylabel('Acceleration (m/s^2)')
```



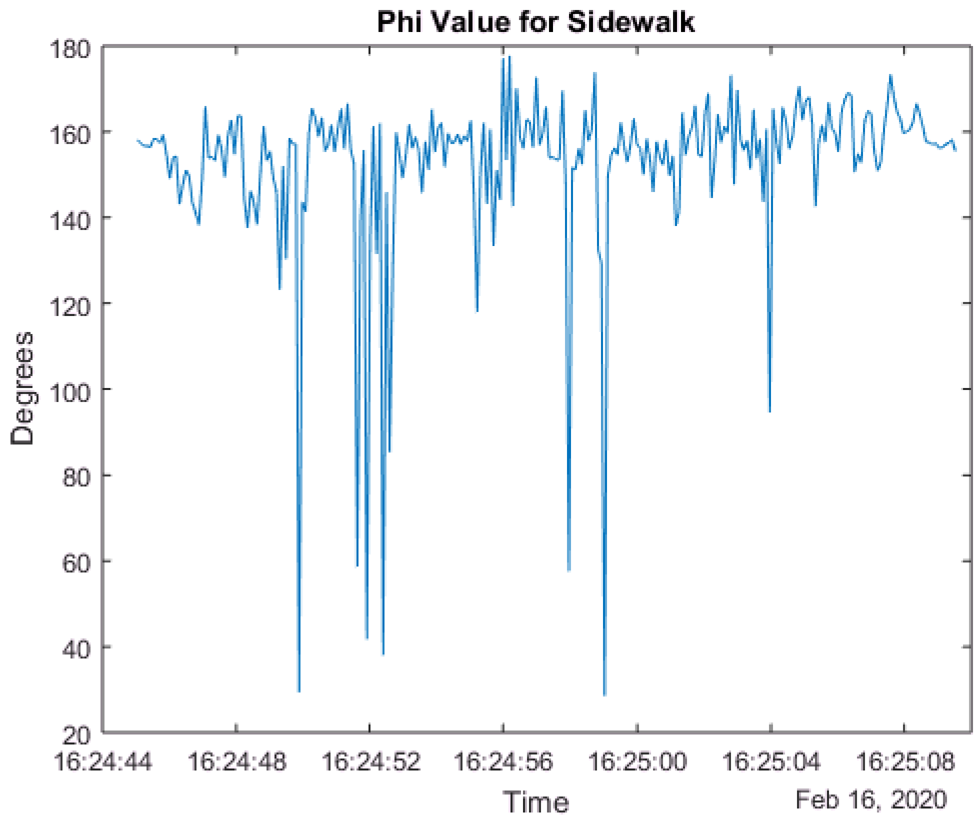
The same goes for the vector magnitude graph.

Next come the spherical polar form values:

```
rhoSidewalk = sqrt(xSidewalk.^2 + ySidewalk.^2 + zSidewalk.^2);
thetaSidewalk = atand(xSidewalk./zSidewalk);
phiSidewalk = acosd(ySidewalk./rhoSidewalk);

figure; plot(tSidewalk, thetaSidewalk)
title('Theta Value for Sidewalk')
xlabel('Time')
ylabel('Degrees')
```



```
figure; plot(tSidewalk, phiSidewalk)
title('Phi Value for Sidewalk')
xlabel('Time')
ylabel('Degrees')
```



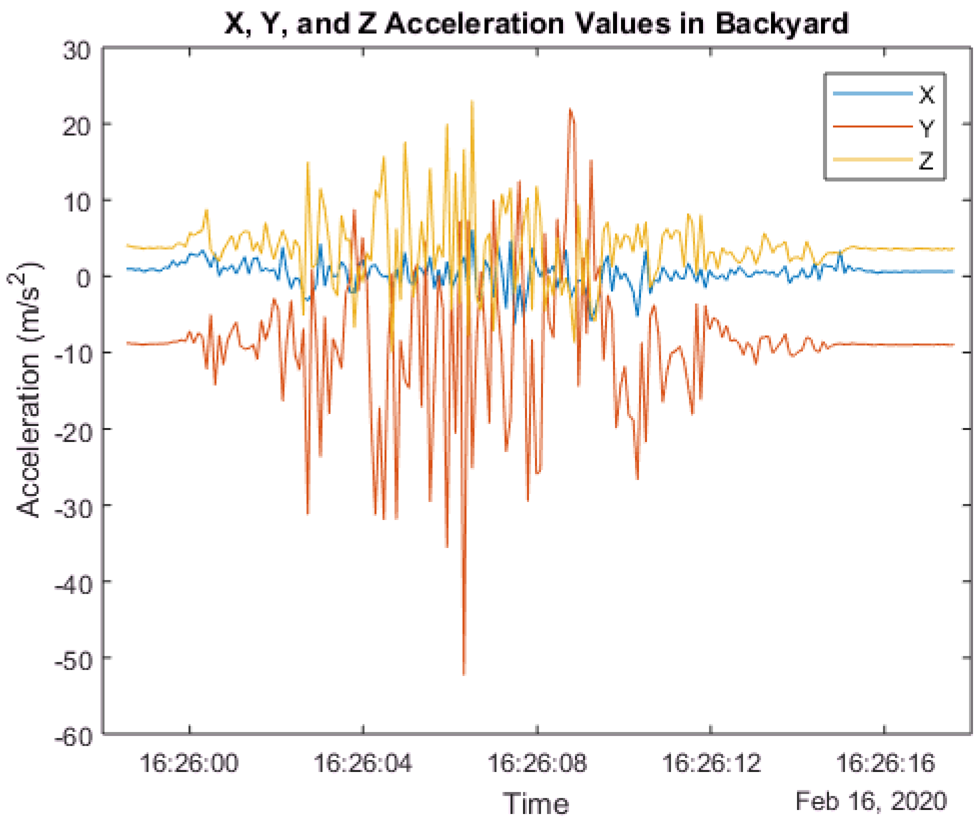
The sidewalk phi graph follows the same patterns as the asphalt phi graph. The large decreases in angle coincide with upwards spikes the y value graph.

Experiment 3: Backyard

For this test, I rode my bike down my backyard. My backyard is very bumpy and covered in grass, so I would expect the acceleration values to have the most variation out of all of the values.

I repeated what I did for the asphalt and sidewalk analyses.

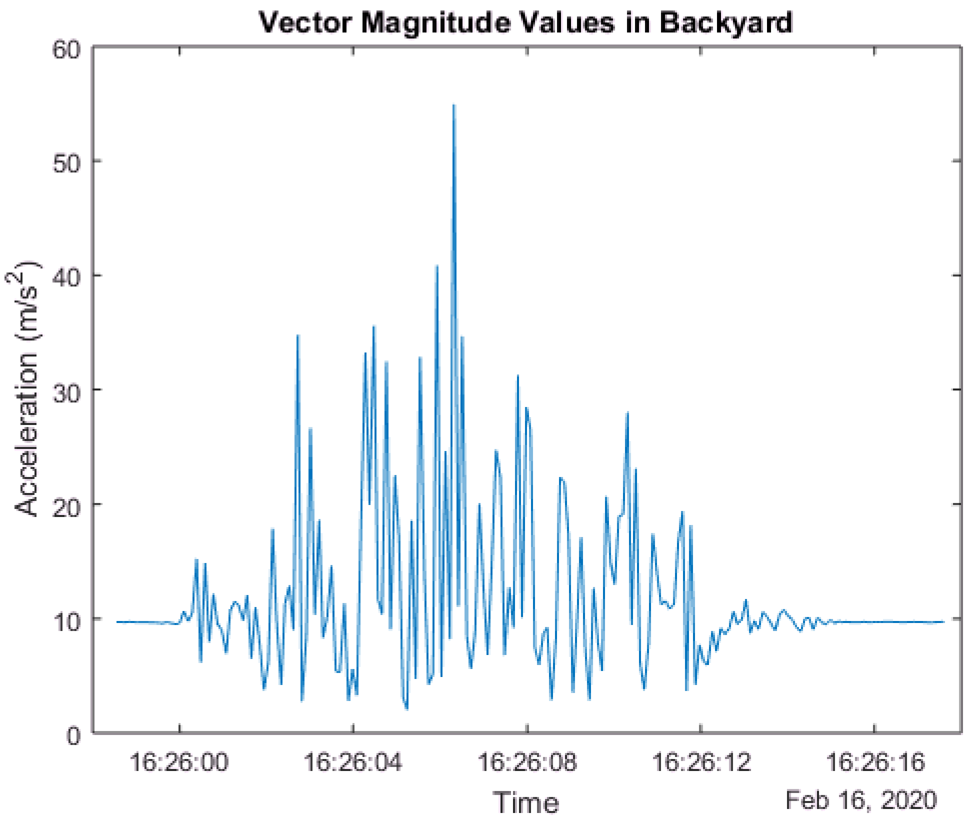
```
load backyard001_Kink40psi.mat
xBackyard = Acceleration.X;
yBackyard = Acceleration.Y;
zBackyard = Acceleration.Z;
tBackyard = Acceleration.Timestamp;


backyardAllDirections = [xBackyard, yBackyard, zBackyard];

figure; plot(tBackyard, backyardAllDirections)
title('X, Y, and Z Acceleration Values in Backyard')
xlabel('Time')
ylabel('Acceleration (m/s^2)')
legend('X', 'Y', 'Z')
```



At first glance, the variation values appear to be slightly larger than the sidewalk variation and significantly larger than the asphalt variation, on average.

```
netBackyardAllDirections = sqrt(xBackyard.^2 + yBackyard.^2 + zBackyard.^2);
```

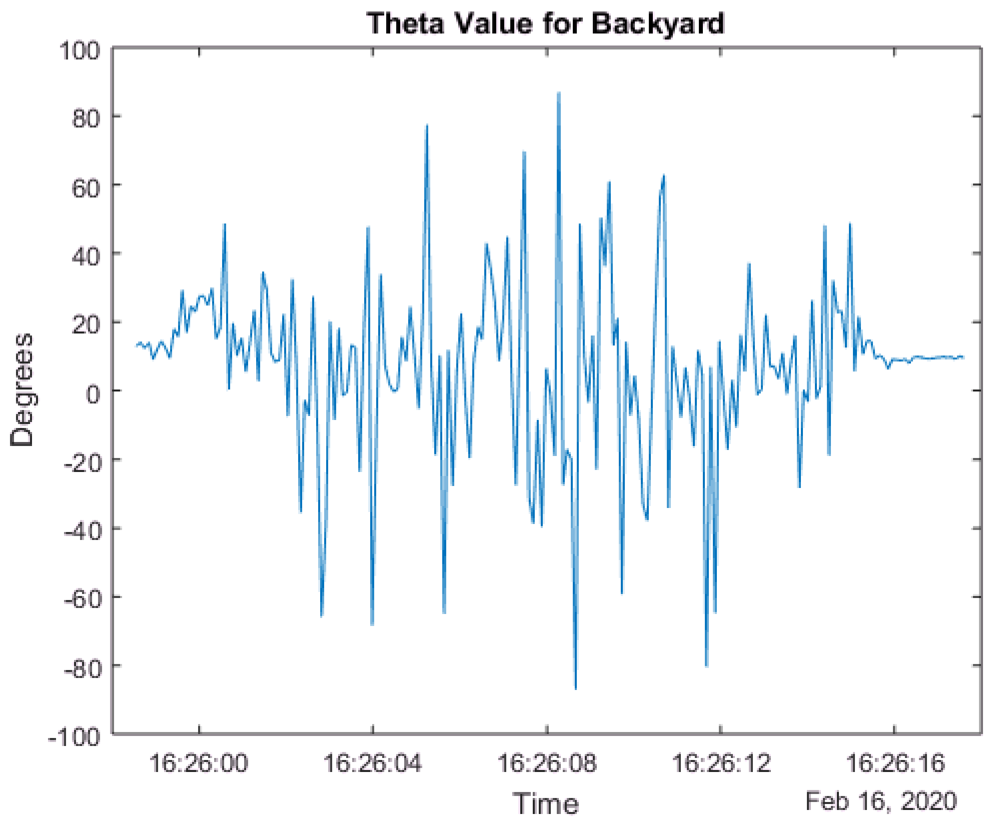
```
figure; plot(tBackyard, netBackyardAllDirections);
title('Vector Magnitude Values in Backyard')
xlabel('Time')
ylabel('Acceleration (m/s^2)')
```



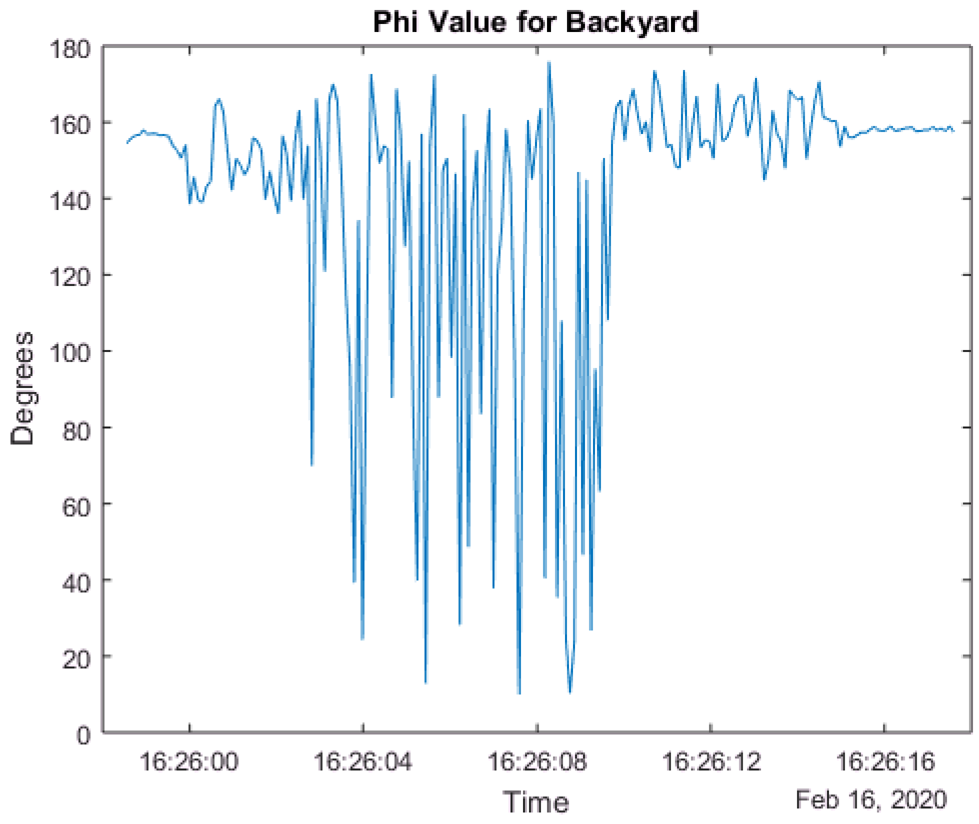
The same goes for the vector magnitude graph.

Next come the spherical polar form values:

```
figure; plot(tBackyard, thetaBackyard)
title('Theta Value for Backyard')
xlabel('Time')
ylabel('Degrees')
```



```
figure; plot(tBackyard, phiBackyard)
title('Phi Value for Backyard')
xlabel('Time')
ylabel('Degrees')
```



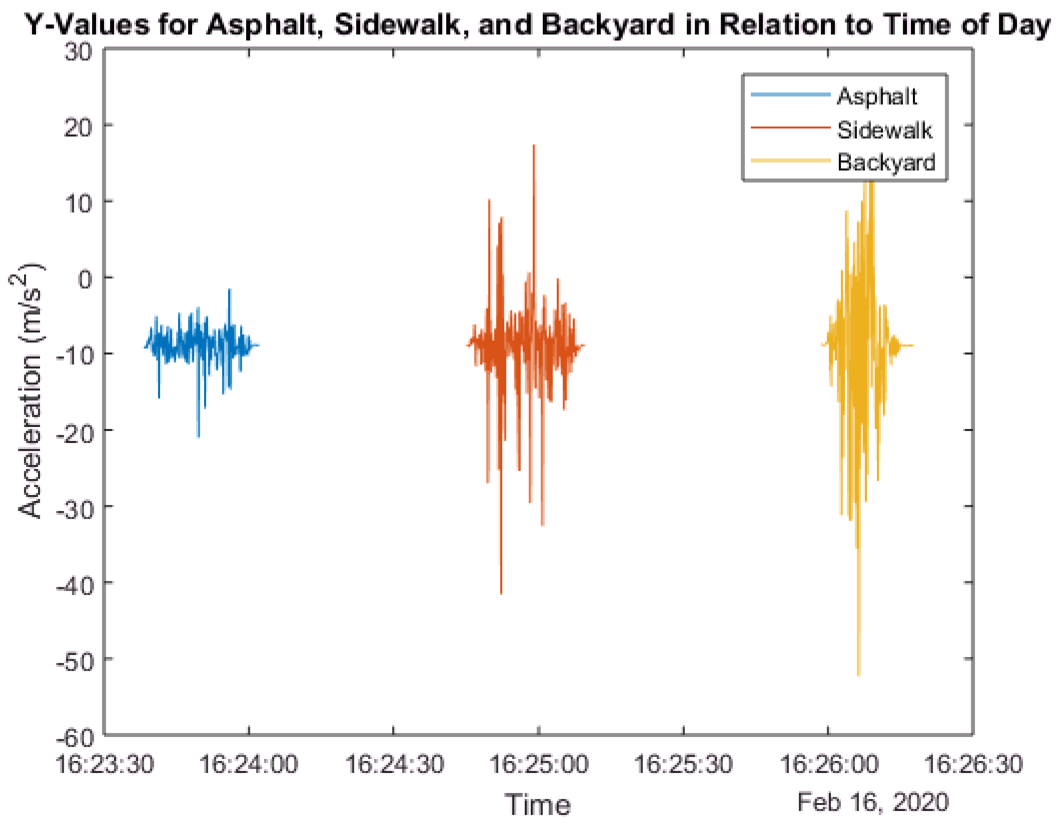
The ovservations for the asphalt and sidewalk experiments remain true for the backyard experiment

Comparing all three experiments:

While I can get a general gauge of how varied the data from each graph is by looking at each graph separately, the best way to compare the three experiments would be to plot them all on the same figure.

However, if I just try to plot all three runs as a function of time, the graph looks like this:

```
figure; plot(tAsphalt, yAsphalt)
hold on
plot(tSidewalk, ySidewalk)
plot(tBackyard, yBackyard)
hold off
title('Y-Values for Asphalt, Sidewalk, and Backyard in Relation to Time of Day')
xlabel('Time')
ylabel('Acceleration (m/s^2)')
legend('Asphalt', 'Sidewalk', 'Backyard')
```



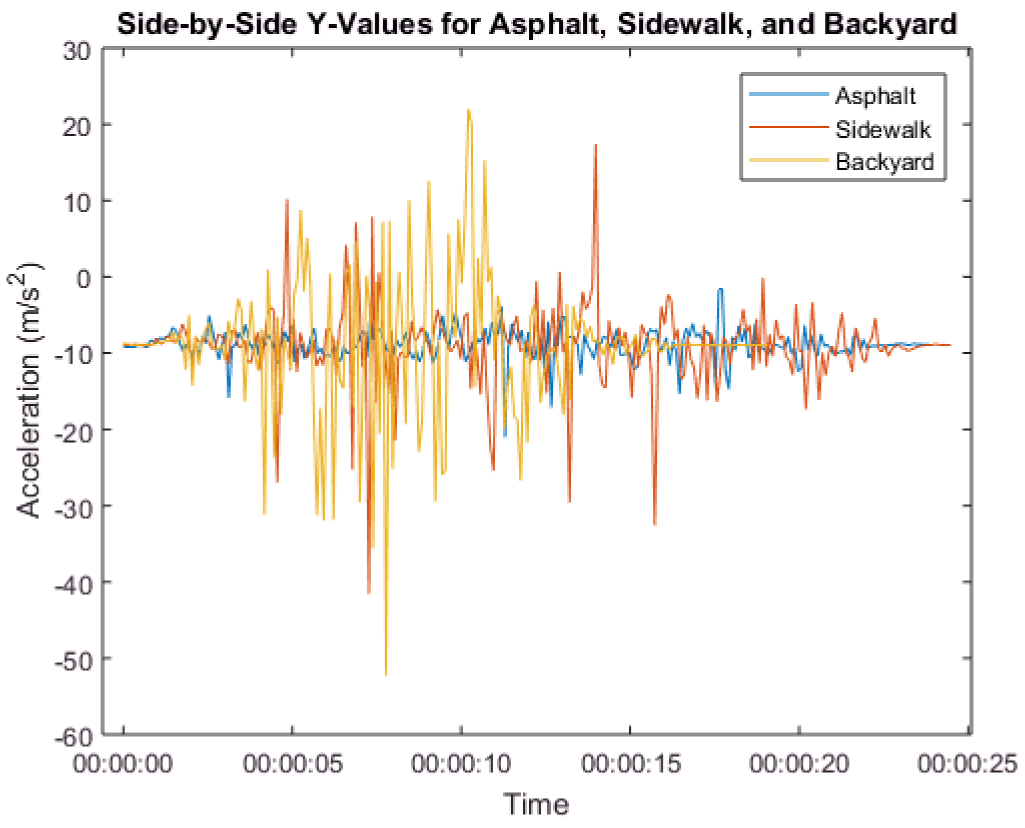
This is because the recorded times are the actual time of day that the experiments were recorded. In order to cancel out this time, I just subtracted each of the time datapoints from the initial time

```
adjustedAsphaltTime = tAsphalt - tAsphalt(1);
adjustedSidewalkTime = tSidewalk - tSidewalk(1);
adjustedBackyardTime = tBackyard - tBackyard(1);
```

With the adjusted time values, I can now plot each of the lines on top of each other.

To start off I plotted for each of the trials' y-values:

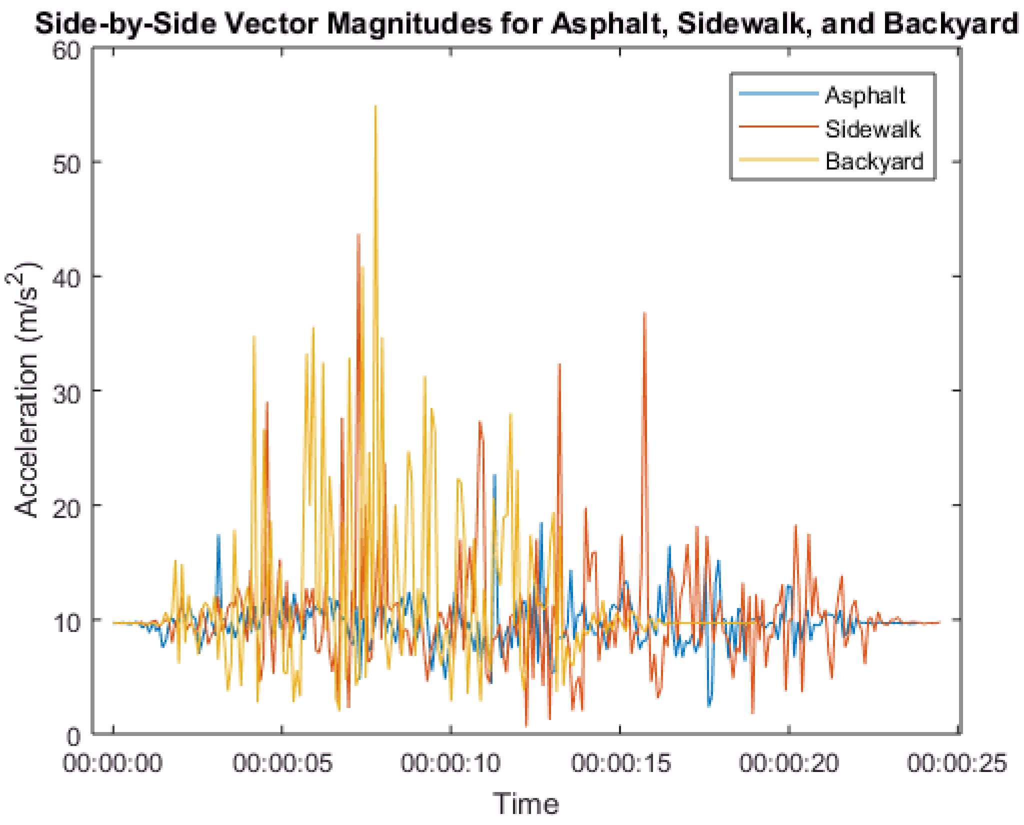
```
figure; plot(adjustedAsphaltTime, yAsphalt)
hold on
plot(adjustedSidewalkTime, ySidewalk)
plot(adjustedBackyardTime, yBackyard)
hold off
title('Side-by-Side Y-Values for Asphalt, Sidewalk, and Backyard')
xlabel('Time')
ylabel('Acceleration (m/s^2)')
legend('Asphalt', 'Sidewalk', 'Backyard')
```



While the sidewalk's values' peaks are simlar in size to the backyard peaks, on average, the backyard graph shows more variance than either of the other two graphs.

Next, I plotted each of the trials' vector magnitude values:

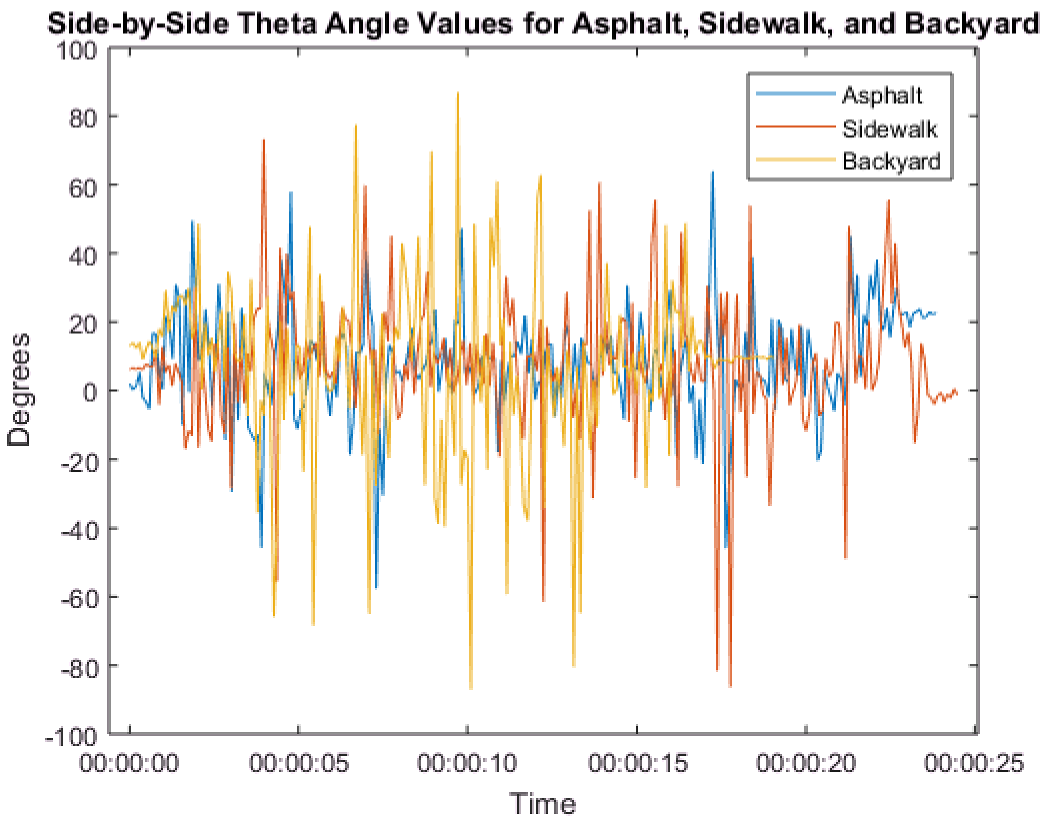
```
figure; plot(adjustedAsphaltTime, netAsphaltAllDirections)
hold on
plot(adjustedSidewalkTime, netSidewalkAllDirections)
plot(adjustedBackyardTime, netBackyardAllDirections)
hold off
title('Side-by-Side Vector Magnitudes for Asphalt, Sidewalk, and Backyard')
xlabel('Time')
ylabel('Acceleration (m/s^2)')
legend('Asphalt', 'Sidewalk', 'Backyard')
```



This figure also supports the idea that the backyard graph has the most variance, and the asphalt graph has the least.

Then, I plotted each of the trials' theta values:

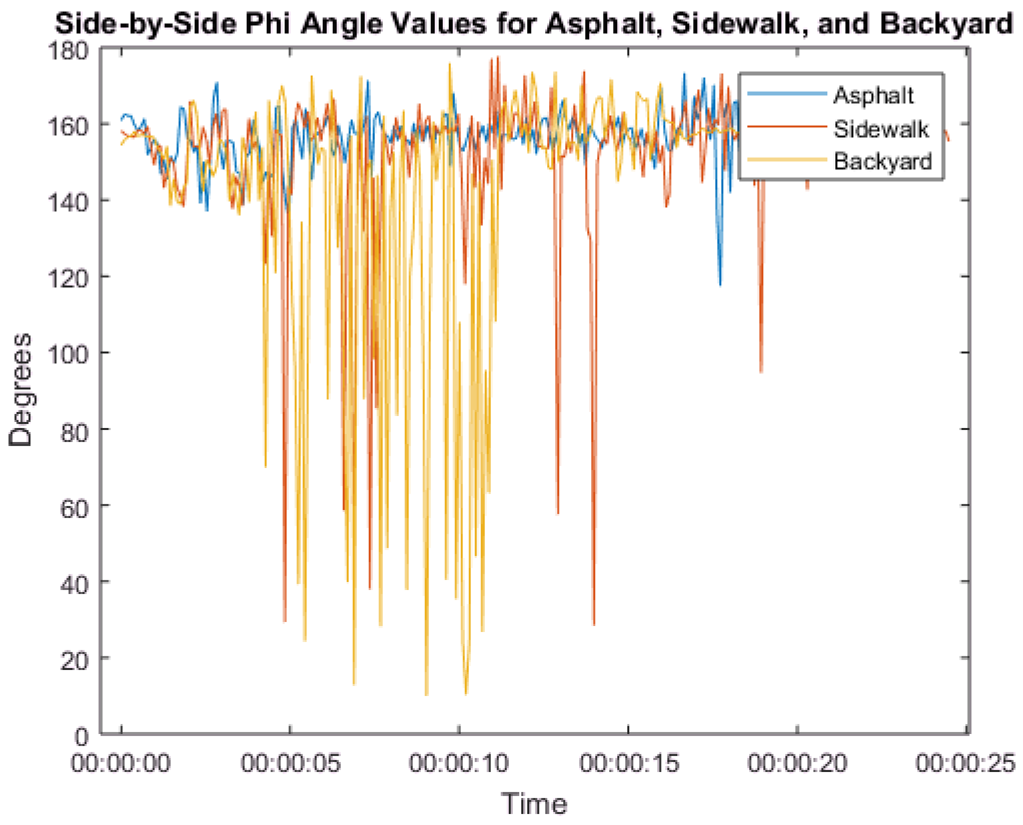
```
figure; plot(adjustedAsphaltTime, thetaAsphalt)
hold on
plot(adjustedSidewalkTime, thetaSidewalk)
plot(adjustedBackyardTime, thetaBackyard)
hold off
title('Side-by-Side Theta Angle Values for Asphalt, Sidewalk, and Backyard')
xlabel('Time')
ylabel('Degrees')
legend('Asphalt', 'Sidewalk', 'Backyard')
```



As expected the theta values were all over the place

Finally, I plotted each of the trials' phi values:

```
figure; plot(adjustedAsphaltTime, phiAsphalt)
hold on
plot(adjustedSidewalkTime, phiSidewalk)
plot(adjustedBackyardTime, phiBackyard)
hold off
title('Side-by-Side Phi Angle Values for Asphalt, Sidewalk, and Backyard')
xlabel('Time')
ylabel('Degrees')
legend('Asphalt', 'Sidewalk', 'Backyard')
```



While this graph does not give any new information in relation to the variance in the terrain, I noticed another cool thing about it: When they are not peaking, (in other words, when the main force affecting the acceleration vector is just gravity) the three lines seem to hover around 160 degrees. Assuming that the acceleration vector at these points is following gravity, I can determine that in the position the phone is tied to my bike, the phone is about 20 degrees off of being perpendicular.

[insert drawing of phone with 160 degree angle measure]

Conclusion:

As expected, my backyard had the roughest terrain, and the asphalt road had the smoothest terrain, but this was not my primary focus when doing these experiments. Having now done the experiments and analyzing the data, I now have a general idea of how to work with Matlab, a solid baseline set of tests to perform on future data, and a good idea on how the results for each of the tests should behave. In the future, I hope to expand on these tests by introducing a variety of new variables, such as different tire pressures, different bikes, rougher terrain, or jumps (the zero-gravity conditions would probably look cool on a graph)