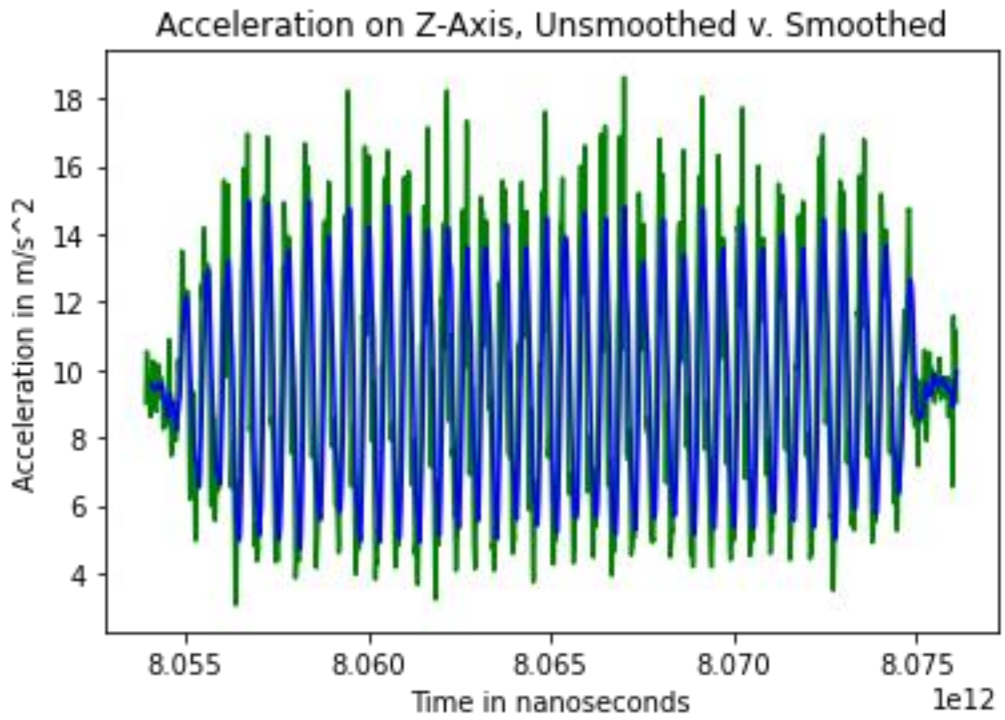
Simple Step Counter:

To appropriately process the provided data, the first step in all the following algorithms is to smooth the data that has been collected by the sensors. All the necessary data is imported into the Jupyter notebook, and then added to the data frame for manipulation. For the first segment only the acceleration data is needed, along with the timestamp. This data then needs to be smoothed out in order to use it properly. To do that we used a simple moving average. Python and Jupyter notebooks have plenty of data analysis tools that make this very easy. The new average is calculated based on the last average and the new value for the timestamp. This will reduce the noise that is included in the dataset and will allow for easier data processing.

We then need to calculate the overall magnitude of the acceleration in all three dimensions. This is done by squaring the value of all 3 acceleration readings, summing them, and then preforming a square root on the sum of those 3 values. The result is the total acceleration of the phone or the person moving. We then need to go about removing the constant effect of gravity on the person or device. This will be a constant acceleration on the z-axis of roughly 9.81 m/s^2, the result will be that the overall acceleration will be averaged at 0. We can then remove the average value of the magnitude and create a new value for the magnitude without the force of gravity. This would be the circumstance where the person or device is not moving, however this is not what we are looking for. From here we can go about finding the number of steps easily. Python yet again has a function for this in the signal processing tools, where you can find the number of peaks in a provided dataset, there needs to be a provided threshold in order to determine how many peaks there are. In this dataset, we will set the threshold to be equal to the standard deviation of series of magnitudes without the force of gravity. Thus, any local maximum that is detected in the dataset that is greater than the value of one standard deviation will be determined to be a “step”. As a result of the data smoothing this is determined to be 38 steps in the WALKING.csv. All of the necessary data will be plotted in the provided Jupyter notebook.



Simple Heading Detection:

While this is more of a simple implementation that does not account for the more minute deviations in changes to the user or the devices' heading. We know that changes along the z-axis are a result of movement vertically. Changes along the x-axis, represent changes horizontally along an East-West axis, and changes along the y-axis represent changes horizontally along the North-South axis. As such any reading of the magnetometer on the y-axis that is roughly 0 is aligned to due North. Any significant changes in the magnetic field strength can be observed to be a turn by a change of roughly 45 degrees. As such any instance where the filed strength changes by 10-15 micro-Teslas, would indicate a turn by roughly 90 degrees in either direction. By basically charting the collected dataset we can see that there are a total of 8 turns, two of which are conducted by turning a total of 180 from the previous heading. Starting at a heading of due West and proceeding by a 90 degree turn to the North, followed by the East and then a 180 degree turn from the South to the North.

Walking and Turning:

Using the previous algorithm to determine the number of steps taken by the person / device. By subsequently importing and smoothing the data by running a simple moving average across the data set. We then isolate and remove the constant effect of the force of gravity. Allowing us to run Python's data analysis tools over the dataset and find that the total number of steps taken in the WALKING_AND_TURNING.csv to be 87 steps. Once again assuming the threshold to register a step to be greater than one standard deviation from the mean of the total acceleration magnitude.