Ayuj Prasad – CS407 Lab 9 PDF

1. Smooth the data with a technique you learned in lecture

I smoothed the data using the Exponential Weighted Moving Average (EWMA) with an alpha value of 0.5.
Each field's values were smoothed into a new column: fieldname -> smooth_fieldname.

2. Write an algorithm to count the number of steps in WALKING.csv and WALKING_AND_TURNING.csv

The algorithm is in the RMD file in the zip folder.

3. Write an algorithm to detect the 90 degree turns in TURNING.csv and WALKING_AND_TURNING.csv

The algorithm is in the RMD file in the zip folder.

4. Combine your algorithms from step 2 and 3 to plot the path of the user in WALKING_AND_TURNING.csv

The algorithm is in the RMD file in the zip folder.

PDF SUMMARY:

1. **An explanation of the different algorithms you developed. Talk about some of the problems you had, and solutions to those problems.**

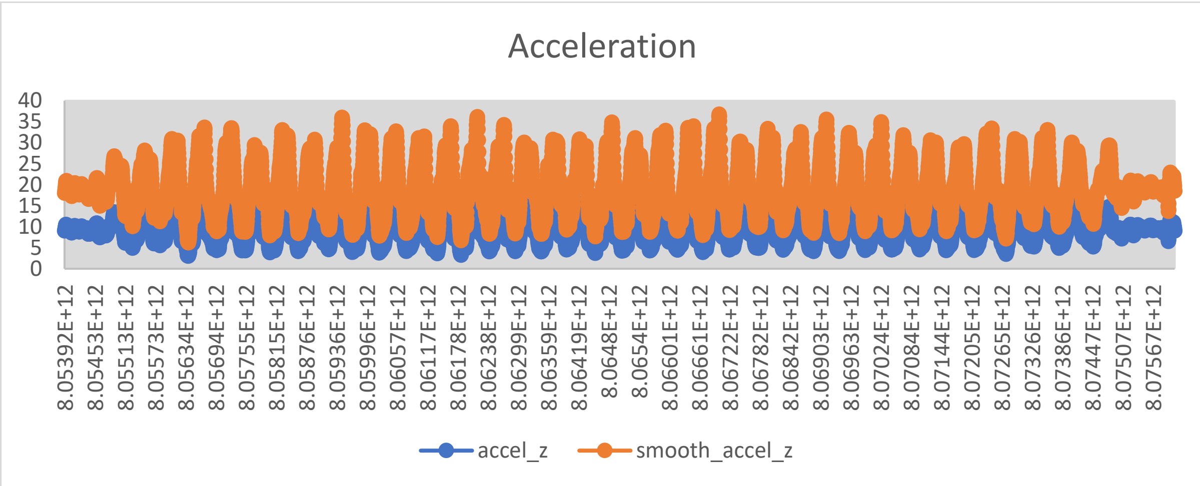
   One of the problems I ran into was deciding which package to use for reading my data. With R/RStudio, I required packages that would count the number of peaks for a given column – I had the options of using IDPMisc (peaks method), pracma (findpeaks method), quantmod (peak method), or code to find the differences and count the peaks (long method but no packages required). There were probably other packages that I did not find, but I had limited my search to these algorithms as they seemed to have the methods that would help me.

   Finally, I decided to go with the IDPMisc package and use the peaks method in my algorithms.

   Step Counting Algorithm: First the data is read, and the peaks from the smoothed values of accel_z are found. As this was counting all peaks (occurring peaks inside the bigger peak and smaller peaks that do not indicate a step taken), we put in further restraints as to the minimum peak width and threshold to eliminate peaks that do not count the steps. Through trial and error, we were able to get "mean – 0.64*sd" as the accurate minimum width value to calibrate to 38 steps in WALKING.csv.

   Turn Counting Algorithm: First the data is read, and the peaks from the smoothed values of gyro_z are found. Likewise with steps, smaller peaks are also read here, so we put a threshold of the average value of all peaks to eliminate the smaller ones (that would not be contributing to a turn smaller than 45 degrees). Another problem here was that only the peaks were counted and not the troughs (for counter-clockwise rotations). To count these, I negated all the values in this column and ran the same peak functions again (this will count the troughs for CCW turns >= 45 degrees). Then these two are added and we get total number of turns by the user.

2. **A plot of 1 second of 1 axis of your sensor data from WALKING.csv along with a smoothed version of the same data. The scale should be such that we can see how the smoothing worked.**



This graph is from WALKING.csv, and plots the accel_z and smooth_accel_z field values on the y-axis, with the timestamp values on the x-axis. A larger, clearer graph can be found in the csv file.

3. **The number of steps for WALKING.csv and WALKING_AND_TURNING.csv.**

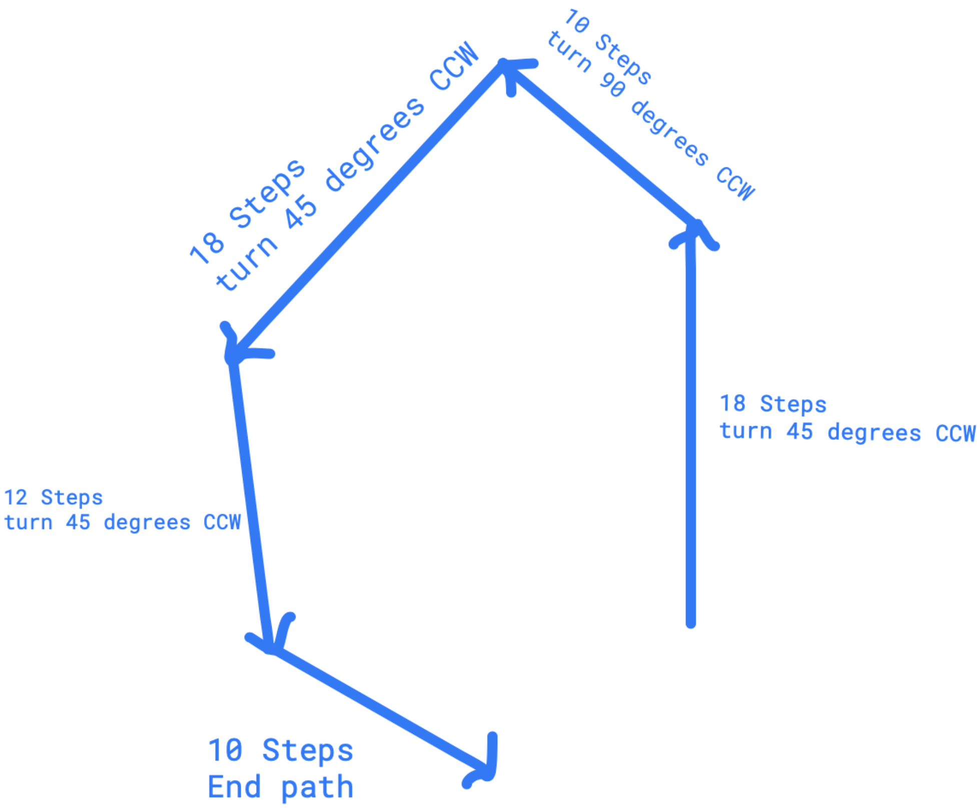
Steps in Walking.csv: 38
Steps in Walking_And_Turning.csv: 68

4. **The number and direction of all turns in WALKING_AND_TURNING.csv.**

Turns in Walking_And_Turning.csv: 5 (for some reason it counts a turn initially even though the excel graph does not show a spike over there, this is ignored in my final plot).

5. **An x,y plot of the route walked in WALKING_AND_TURNING.csv The plot does not need to be drawn to scale, but you have to indicate the distances and angles of the route walked so that it is clear what the route is.**

Using the timestamps and peak values of the gyro graph in WALKING_AND_TURNING.csv, this is the approximate route I came up with:



<u>My inference below:</u>

Although I feel that my algorithm did not read the correct number of steps, this is what I believe the steps would be based on the graphs:

(20 -> 15 -> 20 -> 15 -> 10) with the same rotations.