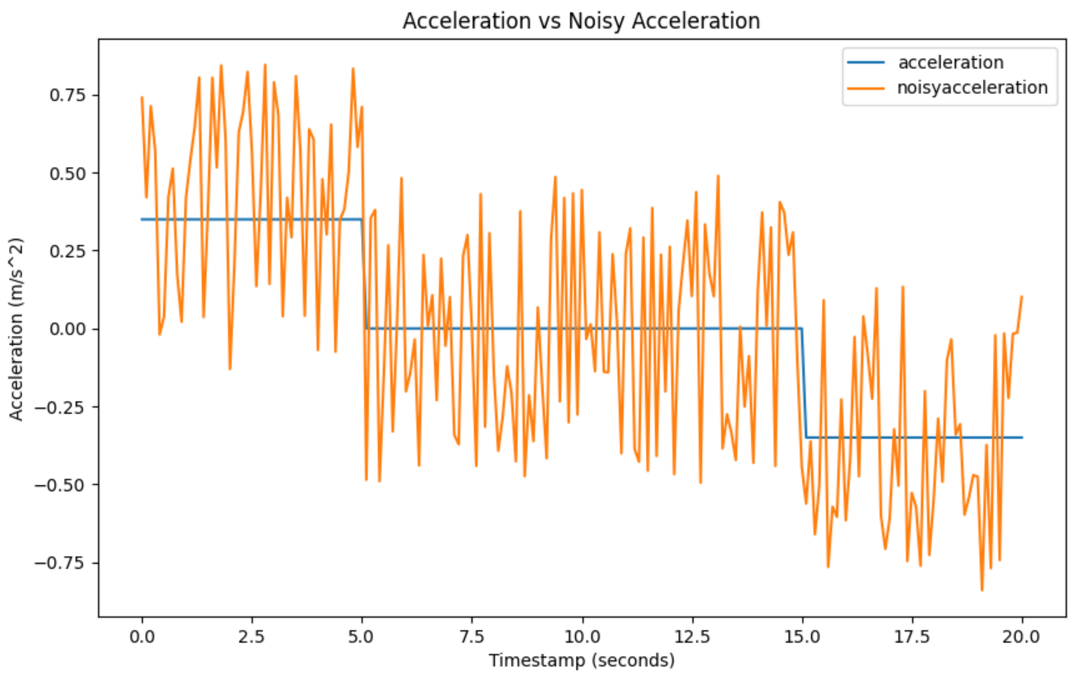
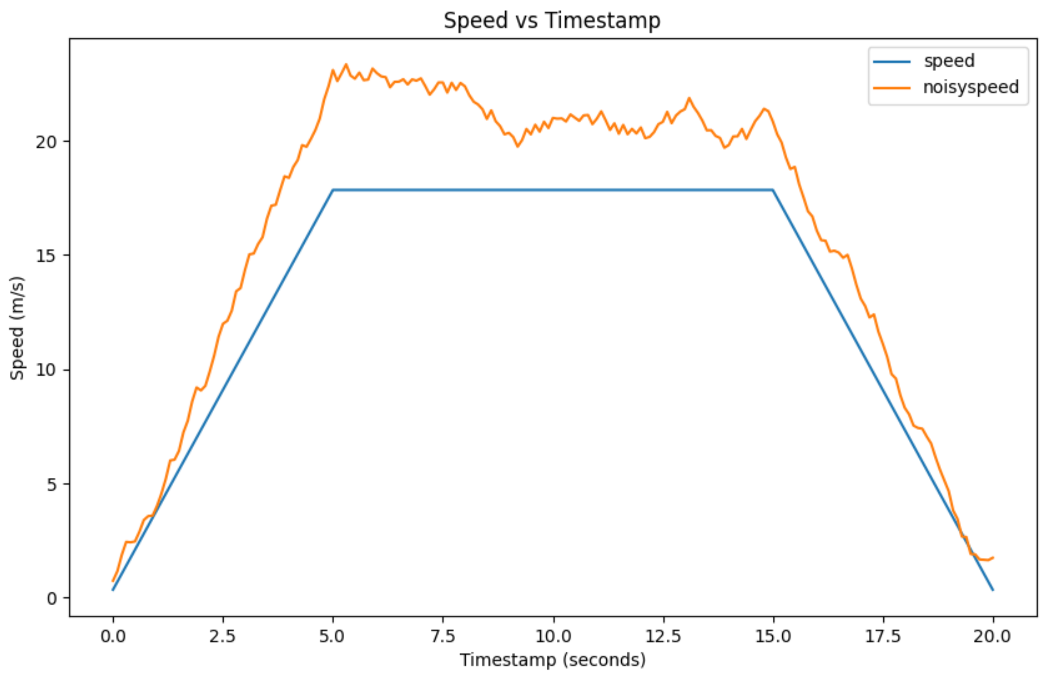
1 Milestone 1: Understanding Sensor Data Errors

- A single plot showing the acceleration and the noisy acceleration.



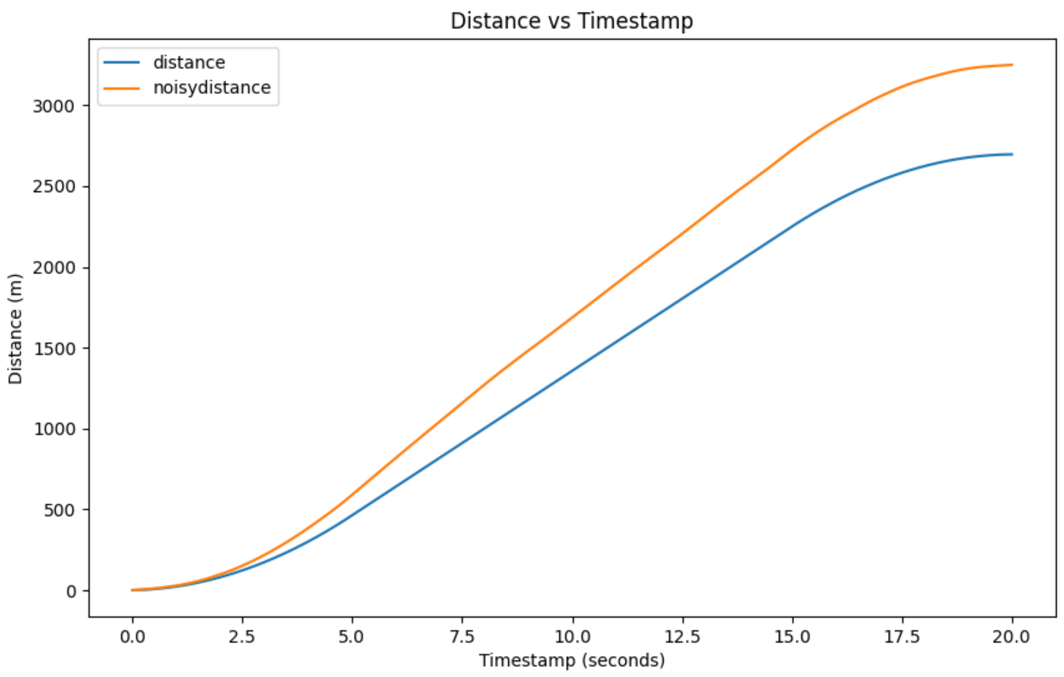
  -

- A single plot showing the speeds that would be obtained using the actual acceleration and the noisy acceleration values respectively.

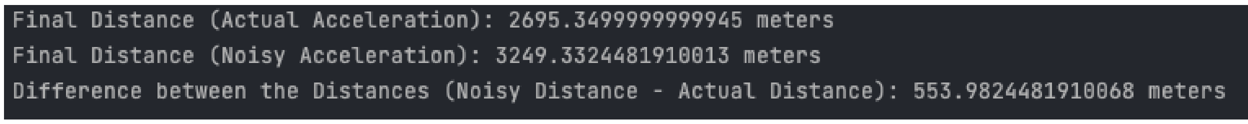


  -

- A single plot showing the distance traveled that would be obtained using the actual acceleration and the noisy acceleration values respectively.



- Report the final distances calculated using the two different accelerations. What is the difference between the two estimates?



```
Final Distance (Actual Acceleration): 2695.3499999999945 meters
Final Distance (Noisy Acceleration): 3249.3324481910013 meters
Difference between the Distances (Noisy Distance - Actual Distance): 553.9824481910068 meters
```

2 Milestone 2: Step Detection

2.1 Data Preparation

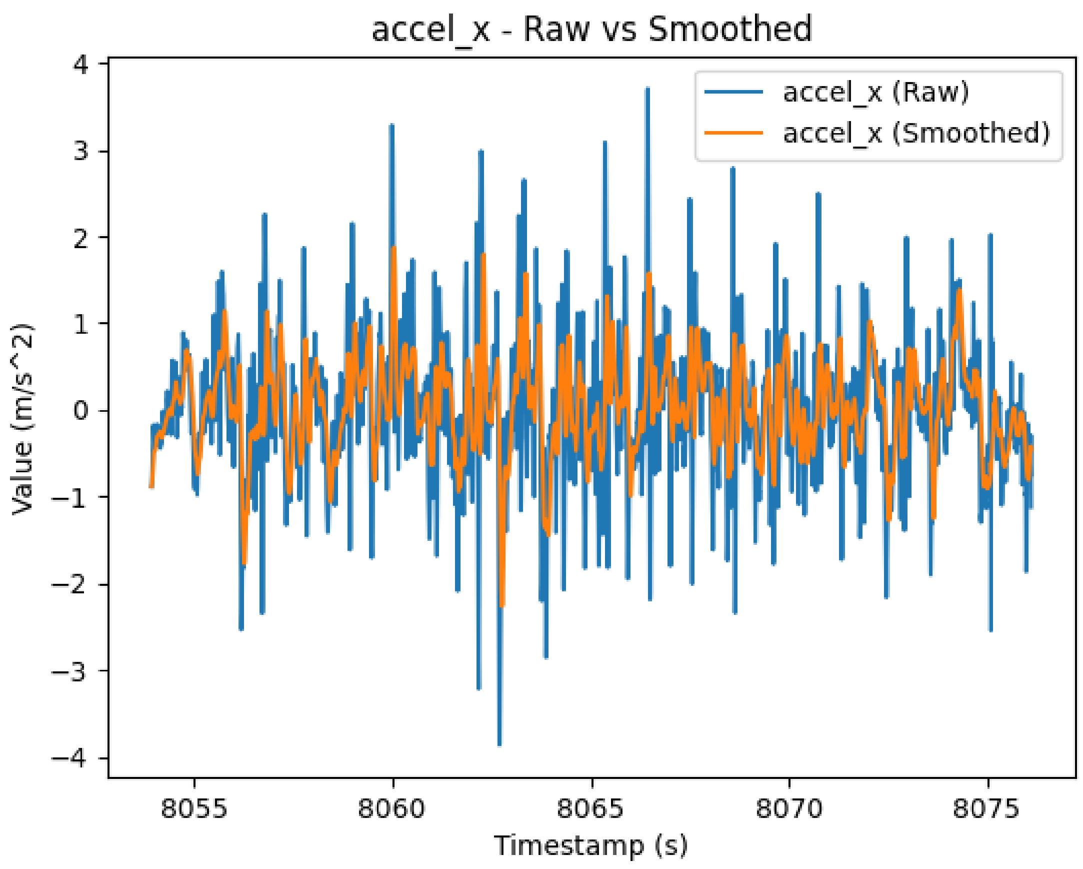
In this initial phase, your objective is to identify the most appropriate time-series dataset for step detection. Specifically, determine which column(s) from WALKING.csv will be utilized. For each column selected, show a graph depicting the raw data and the data after applying a smoothing technique. The graphs must have the timestamp on the x-axis and the respective values on the y-axis. This will visually articulate the effectiveness of your smoothing algorithms.

Detail the smoothing process by providing the relevant code snippets and an accompanying explanation. Ensure that the code is identical to the version submitted on GitHub. If an interactive tool was employed to process the data, include screenshots to facilitate reproducibility.

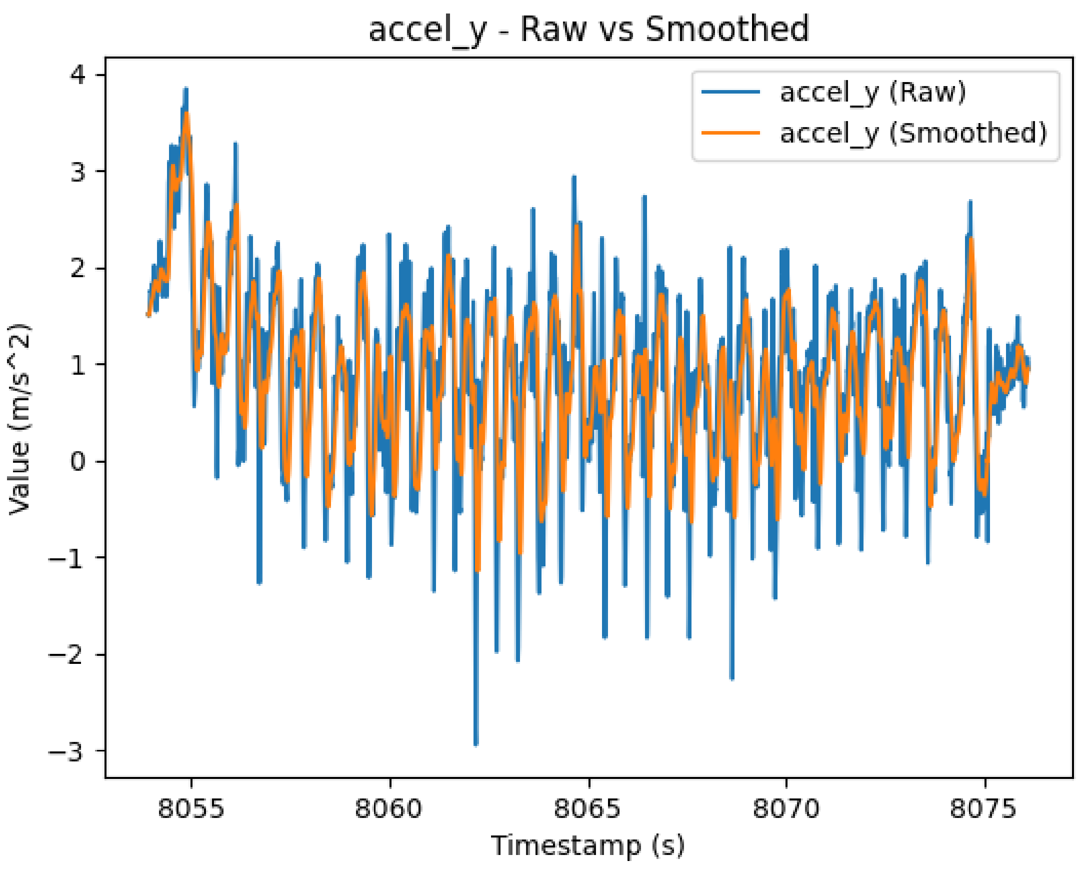
In summary, you will show

- A single figure for each time series you choose, showing the raw data and the smoothed data.

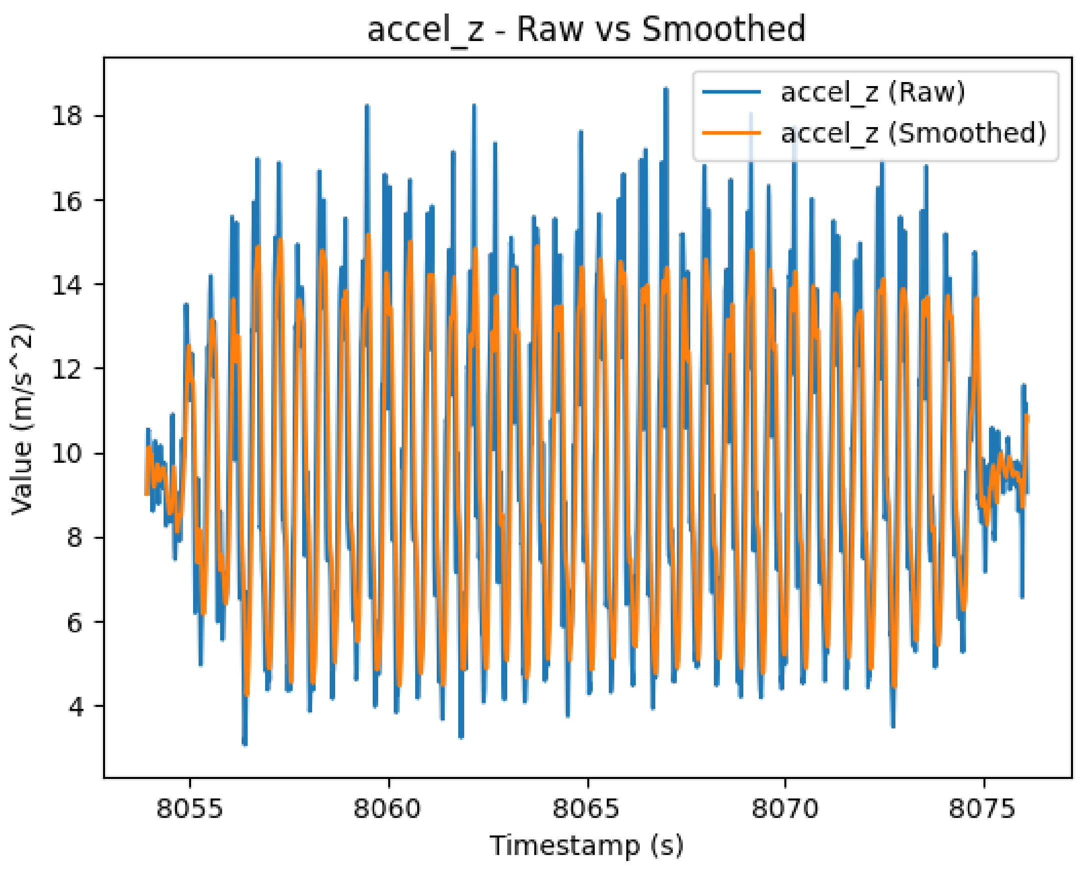
- ○ .


accel_x - Raw vs Smoothed
accel_x (Raw)
accel_x (Smoothed)
Value (m/s^2)
Timestamp (s)
4
3
2
1
0
−1
−2
−3
−4
8055
8060
8065
8070
8075

○ .


accel_y - Raw vs Smoothed
accel_y (Raw)
accel_y (Smoothed)
Value (m/s^2)
Timestamp (s)
4
3
2
1
0
−1
−2
−3
8055
8060
8065
8070
8075

    - .



- An explanation of how you smooth the data, including necessary parameters.
    - This data was smoothed using a moving average. The parameter we used was a window size of 20 as it offered a nice result when observing the magnitude of the 3 acceleration graphs. This moving average was calculated through a built-in method in pandas.

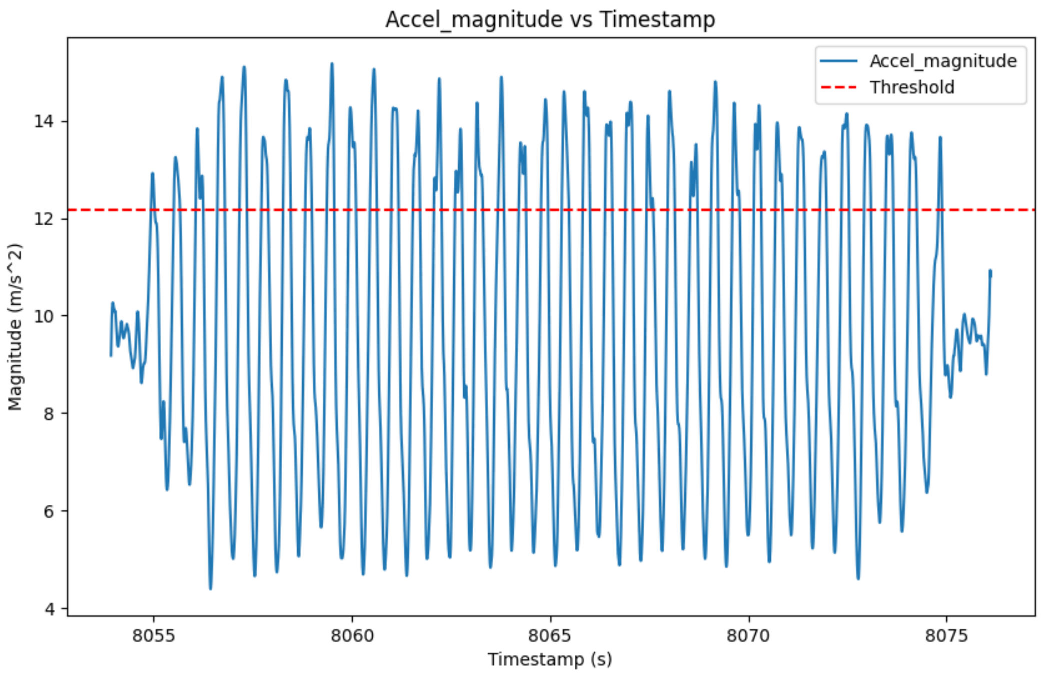
2.2 Step Detection Algorithm

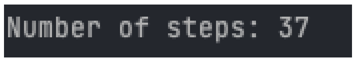
Here, you will delineate the methodology for detecting steps. Present critical information about your algorithm, which could be in the form of annotated code snippets, clear screenshots, or structured pseudocode—these should align with your GitHub submission.
Merely presenting code will not suffice. You must elucidate the logic behind the code segments you provide. Should you define any variables as thresholds within your step detection algorithm, offer a rationale for the selected values and justify why they are deemed optimal.
Your description of the implementation must provide sufficient detail to ensure the reproducibility of your GitHub submission.
Finally, report the step count results prominently in this segment.

In summary, you will show

- An explanation of how you detect steps.

- The steps were detected by counting the peaks in the acceleration magnitude graph that were above the threshold that we picked. The threshold was chosen to be the 75th quantile of data because we wanted the threshold to be higher than the peaks that would be likely noise (and would be around the mean).



- The result of step counting.
  - 

```
Number of steps: 37
```

  - Through running the script we were able to detect a total of 37 steps

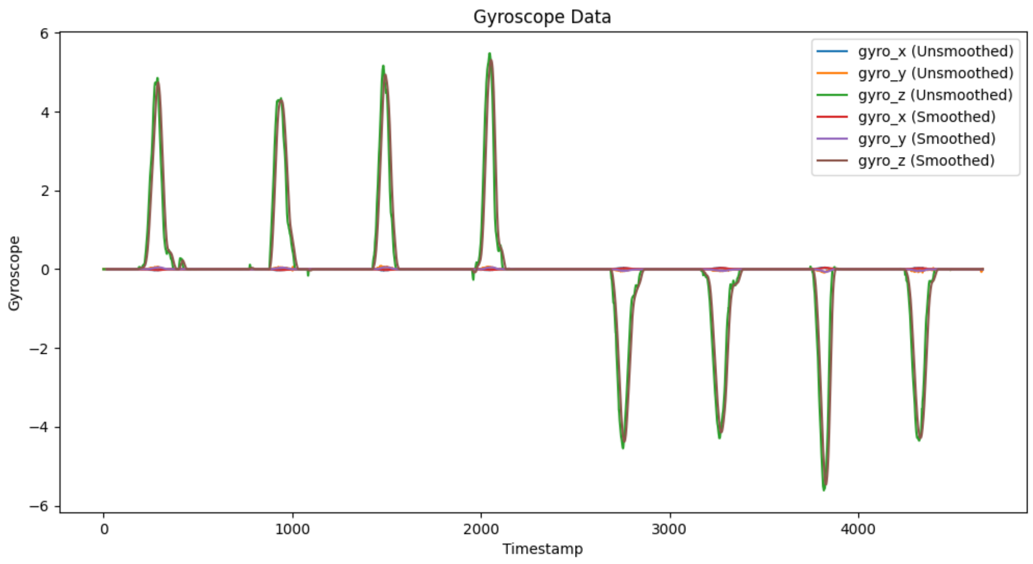
Milestone 3: Direction Detection

3.1 Data Preparation

This task mirrors the prerequisites of Section 2.1, but focuses on the TURNING.csv dataset. Display the methods used to smooth the dataset and present the corresponding visuals.

In this subsection, you will show

- A single figure for each time series you choose, showing the raw data and the smoothed data.



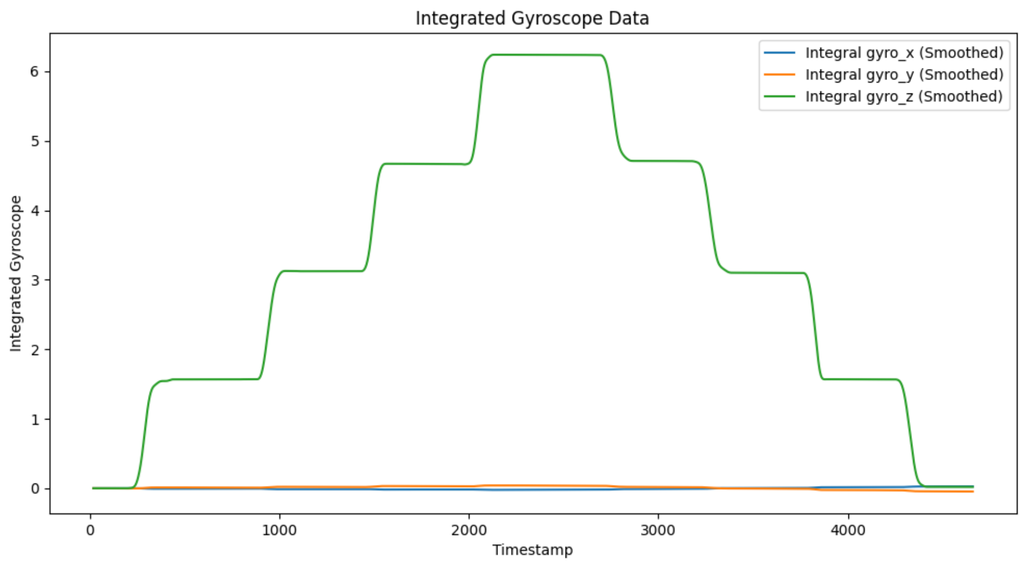
- An explanation of how you smooth the data, including necessary parameters.
    - This data was smoothed using a moving average. The parameter we used was a window size of 20 as it offered a nice result when observing the magnitude of the 3 gyro graphs. This moving average was calculated through a built-in method in pandas.

3.2 Direction Detection Algorithm

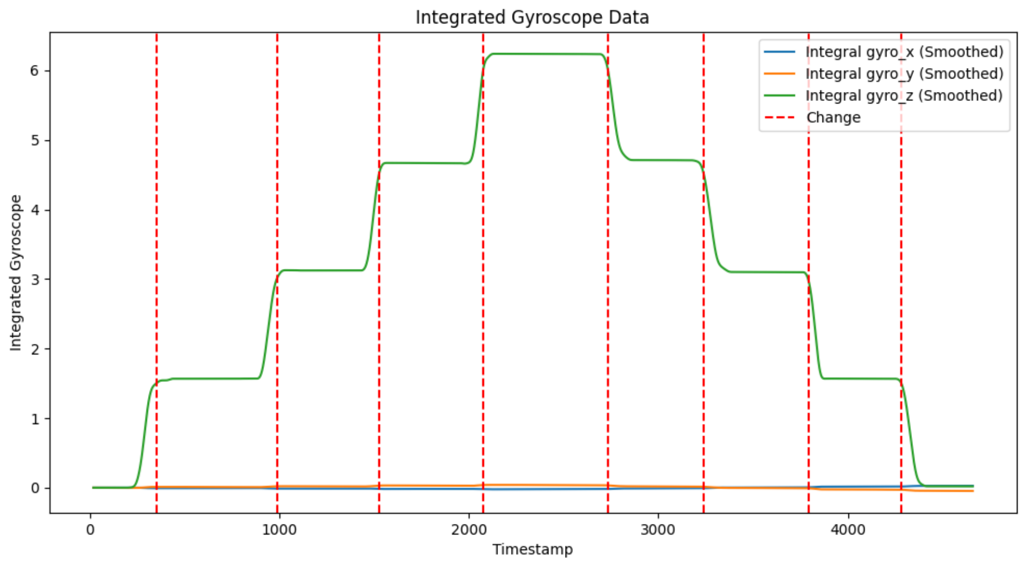
This subsection is dedicated to illustrating your approach to determining the direction of turning. As with step detection, detail the employed code and supplement it with explanations. Make sure to convey the logical framework of the code in a manner that is comprehensible and well-defined.

In this subsection, you will show,

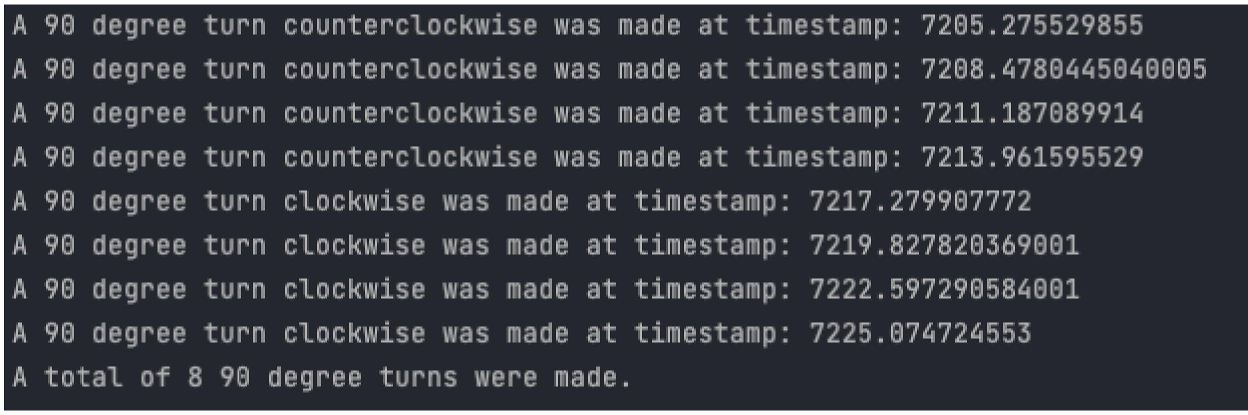
- An explanation of how you detect the direction of turning.
    - The direction of each turn, inferred from the gyroscope data, is presented as clockwise or anti-clockwise based on the sign of the changes in gyroscopic readings along the respective axes in our case mostly the z-axis. In the code if the angular displacement decreased by about 1.5 radians that meant the turn was clockwise and vice versa.



○



- The result of direction turning detection, such as the number of turns detected and the angle of each turn.

- ◦ 

```
A 90 degree turn counterclockwise was made at timestamp: 7205.275529855
A 90 degree turn counterclockwise was made at timestamp: 7208.4780445040005
A 90 degree turn counterclockwise was made at timestamp: 7211.187089914
A 90 degree turn counterclockwise was made at timestamp: 7213.961595529
A 90 degree turn clockwise was made at timestamp: 7217.279907772
A 90 degree turn clockwise was made at timestamp: 7219.827820369001
A 90 degree turn clockwise was made at timestamp: 7222.597290584001
A 90 degree turn clockwise was made at timestamp: 7225.074724553
A total of 8 90 degree turns were made.
```

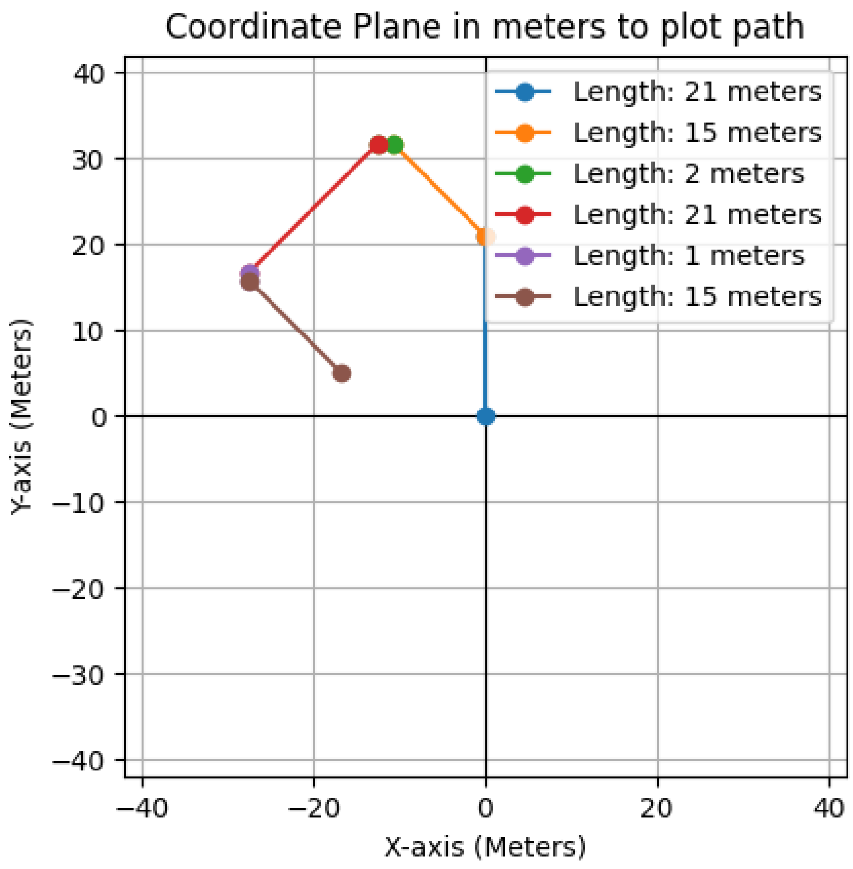
4 Milestone 4: Trajectory Plotting

The culmination of this project involves plotting the trajectory of the walking path. Provide a graphical representation that outlines the trajectory, assuming an initial northward orientation. Ensure that the diagram encapsulates (1) The path traversed by the individual; (2) Accurate depictions of the distances covered and turning angles; (3) You can annotate these directly on the figure or describe them in the accompanying text.

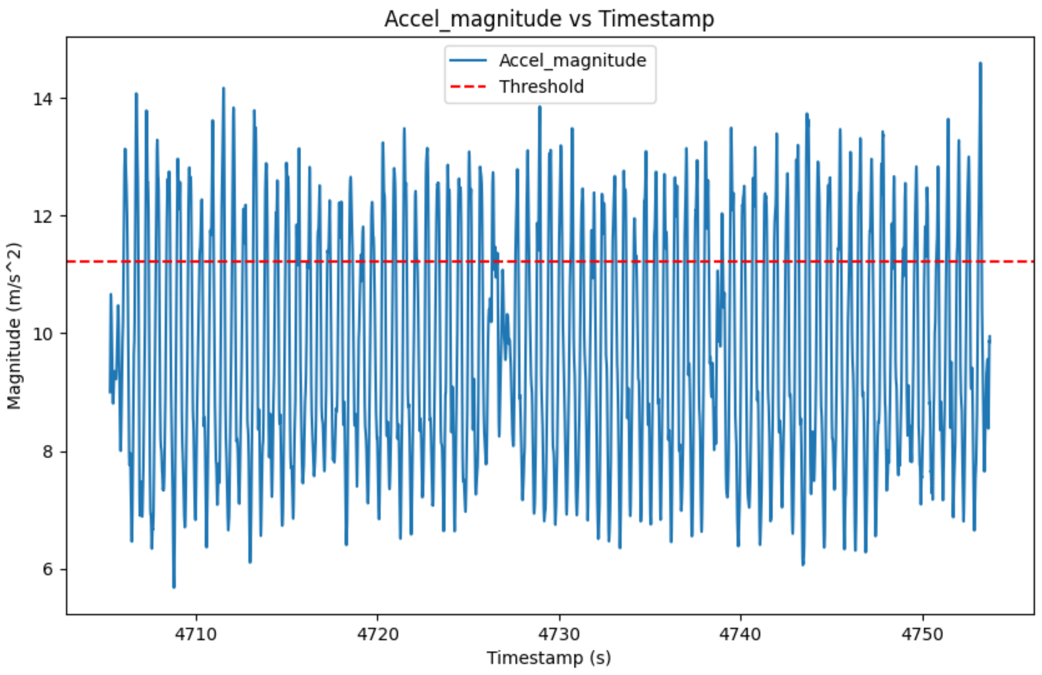
Include the pertinent code and a thorough explanation of how you accomplished the trajectory plotting task. Ensure the final plot can be obtained by following the steps you took to finish this task.

In summary, in this subsection you will show:

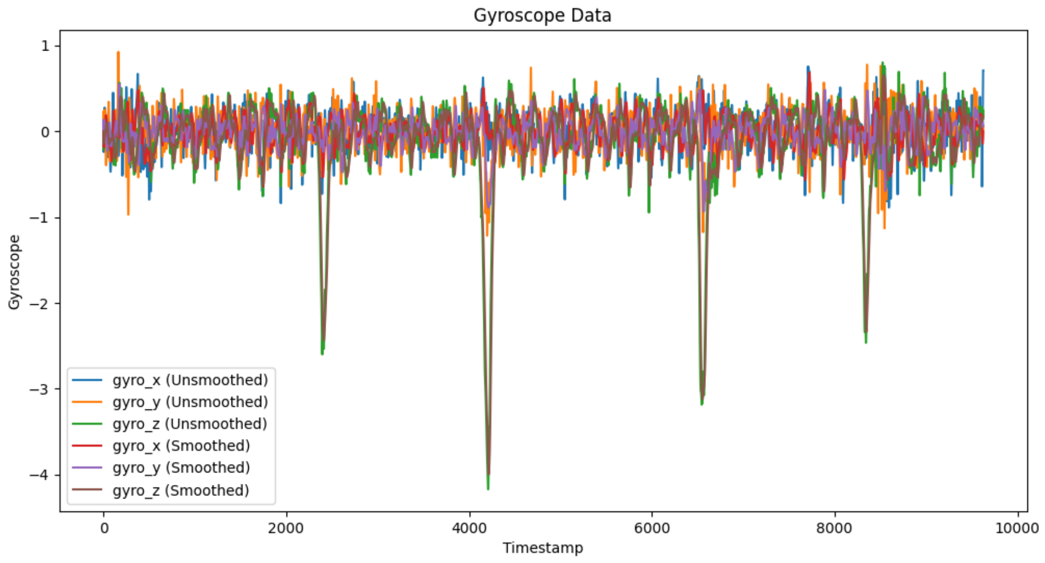
- A diagram showing the trajectory of the path.



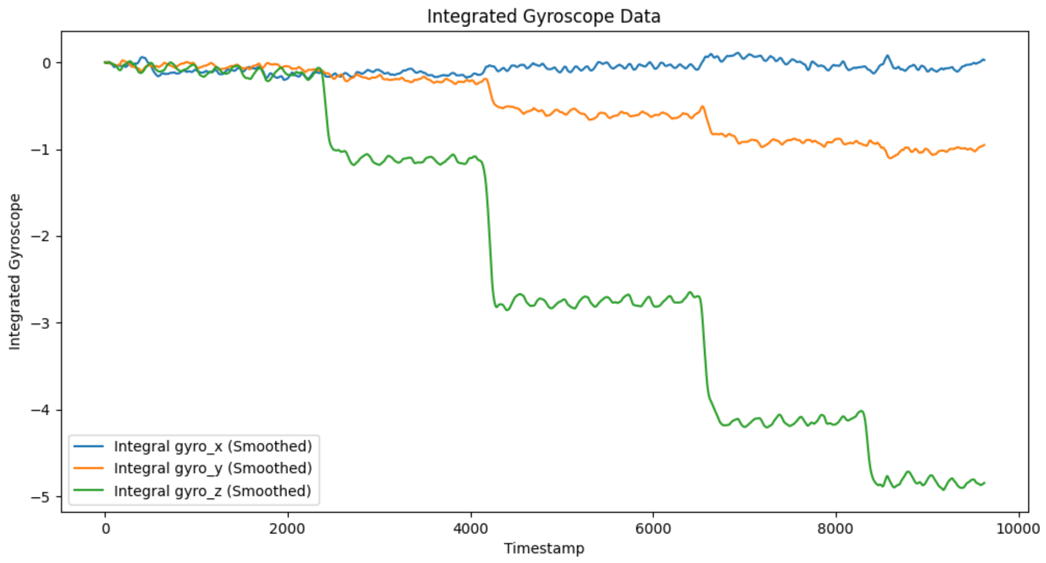
- A comprehensive explanation of the methods used to achieve the final plot.
  - We started off by using what we learned in milestone 2 to determine the total number of steps to be 87. From there we went ahead and utilized milestone 3 to help with determining the number of changes in angles we had (there were a total of 6 45-degree turns). For each turn that was calculated, we also summed the number of steps from the last turn to be able to accurately depict the path on the graph. We then assembled the code to showcase the graph in a coordinate plane where the positive y-axis is north and the path starts at 0,0. Below are the images depicting the steps we took to achieve this final answer.


Accel_magnitude vs Timestamp
Accel_magnitude
Threshold
Magnitude (m/s^2)
Timestamp (s)
6
8
10
12
14
4710
4720
4730
4740
4750

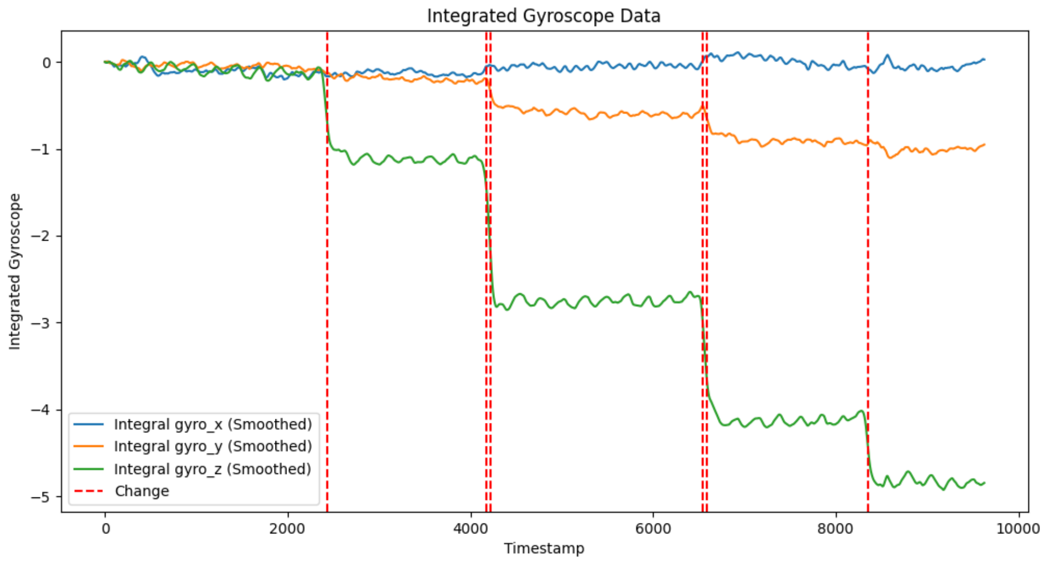
-
- .


Gyroscope Data
gyro_x (Unsmoothed)
gyro_y (Unsmoothed)
gyro_z (Unsmoothed)
gyro_x (Smoothed)
gyro_y (Smoothed)
gyro_z (Smoothed)
Gyroscope
Timestamp
1
0
−1
−2
−3
−4
0
2000
4000
6000
8000
10000

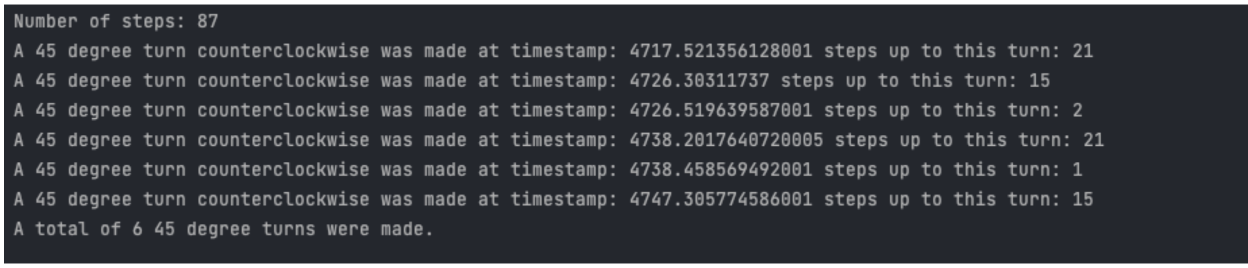
- .



- .



- .



```
Number of steps: 87
A 45 degree turn counterclockwise was made at timestamp: 4717.521356128001 steps up to this turn: 21
A 45 degree turn counterclockwise was made at timestamp: 4726.30311737 steps up to this turn: 15
A 45 degree turn counterclockwise was made at timestamp: 4726.519639587001 steps up to this turn: 2
A 45 degree turn counterclockwise was made at timestamp: 4738.2017640720005 steps up to this turn: 21
A 45 degree turn counterclockwise was made at timestamp: 4738.458569492001 steps up to this turn: 1
A 45 degree turn counterclockwise was made at timestamp: 4747.305774586001 steps up to this turn: 15
A total of 6 45 degree turns were made.
```