CS58007 IoT Homework 1 Report

Group	Student	UG/G
1	Nisa Defne Aksu	G
	Pelin Karadal	G
	Shahd Shehab Hassan Abdelfattah Şerif	G
	Barkın Var	G

The data files in this homework are generated by the **Pyphox** app. To generate it we used "Acceleration with g" functionality and for Part 3 created a new experiment with accelerometer and gyroscope together. You can see which data files are used for which parts of the homework below. The codes can be found in the **main.py** file. The code was run on Barkın's Raspberry Pi.

DATA PREPARATION: Although this is not necessary for Parts 1 and 3 due to the homework requirements, we have still included it.

Part 1: Create four CSV files with the names standing.csv, sitting.csv, walking.csv and running.csv in the same directory as main.py, then replace the existing files with these new ones.

Part 2: Place your walking CSV file in the same directory as main.py. When prompted, enter the name of the file (e.g. walking.csv).

Part 3: Create files with the same names for each activity (accelerometer.csv and gyroscope.csv). Replace them with existing CSV files in the activity folders. The code will automatically load these folders and save the results as 'pose_estimation.csv' in each folder.

Part 1

The function plot_activities(), visualizes and compares accelerometer readings for four physical activities: standing, sitting, walking, and running. For each activity, it loads the corresponding CSV file, calculates the absolute acceleration magnitude from the x, y, and z acceleration components, and plots it against time. It also prints basic statistical features—mean, standard deviation, minimum, and maximum acceleration—for each activity, showing how motion intensity changes across activities.

The function expects CSV files named standing.csv, sitting.csv, walking.csv, and running.csv in the same directory. Each file should include columns like Time (s), Acceleration x (m/s²), Acceleration y (m/s²), and Acceleration z (m/s²). The function displays a line plot comparing acceleration patterns for all activities, helping visualize differences in motion intensity (e.g., running shows higher, more variable acceleration than sitting).

Terminal Output:

=== Part 1: Activity Visualization & Feature Inspection ===

standing: mean=9.85 m/s², std=0.08 m/s², min=9.30, max=10.64

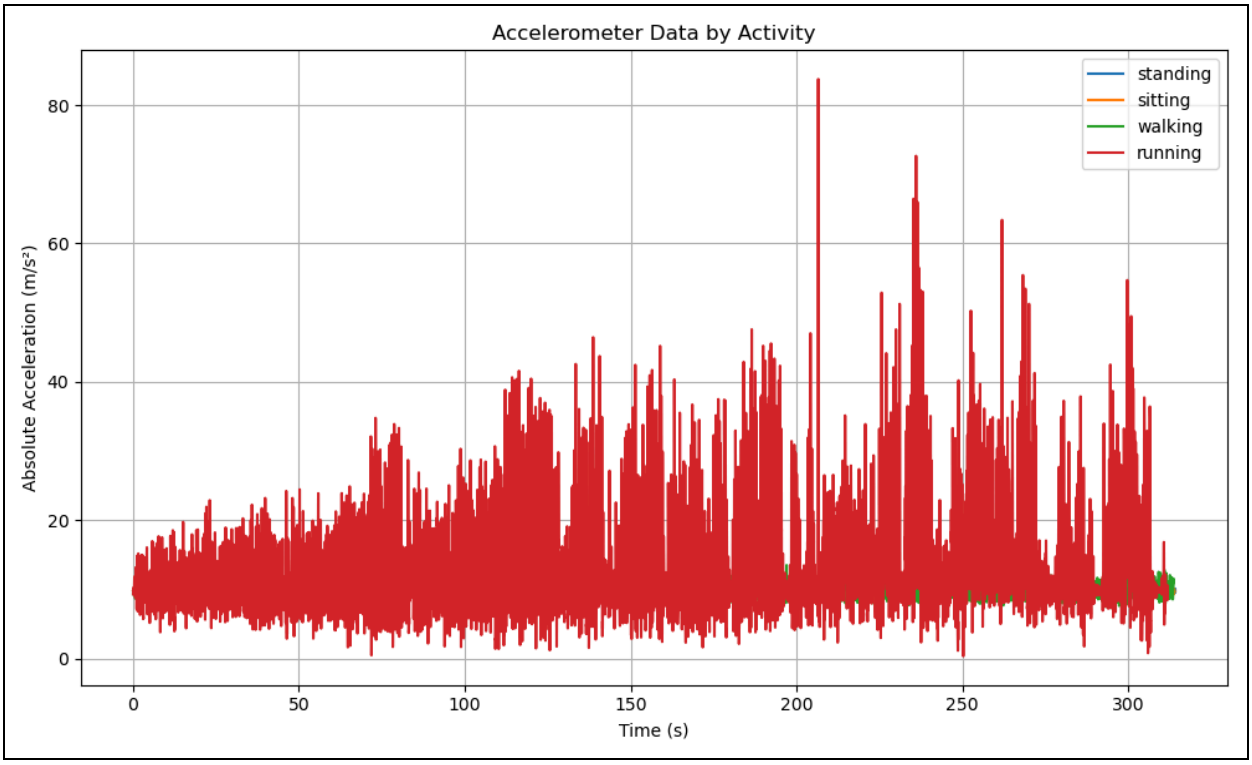
sitting: mean=9.84 m/s², std=0.06 m/s², min=9.32, max=10.26

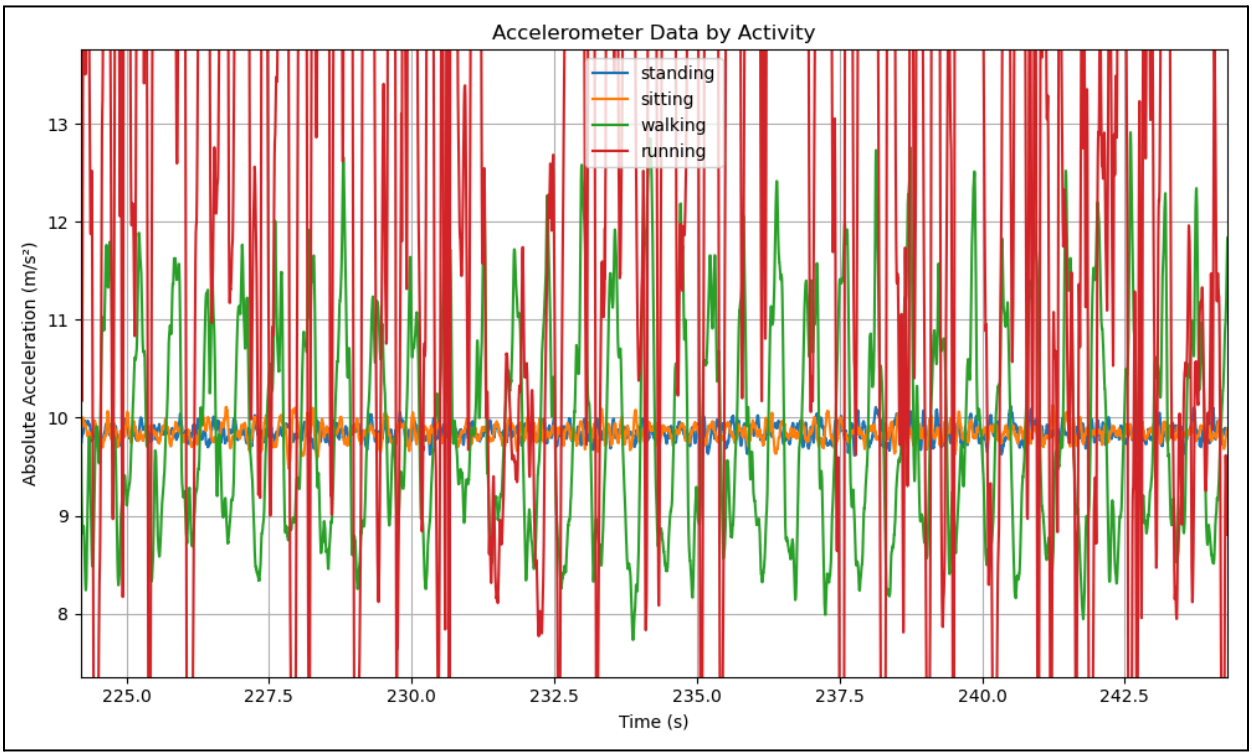
walking: mean=9.90 m/s², std=1.00 m/s², min=7.49, max=13.72

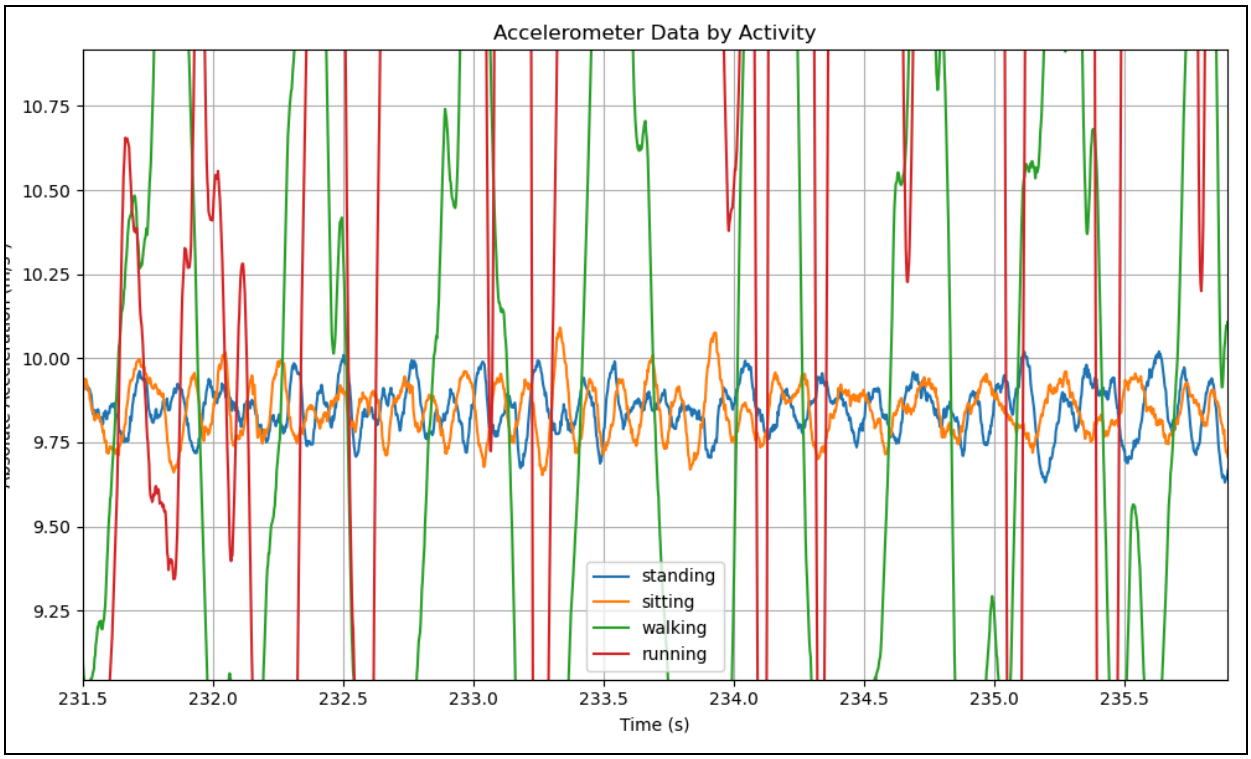
running: mean=13.08 m/s², std=6.77 m/s², min=0.38, max=83.75

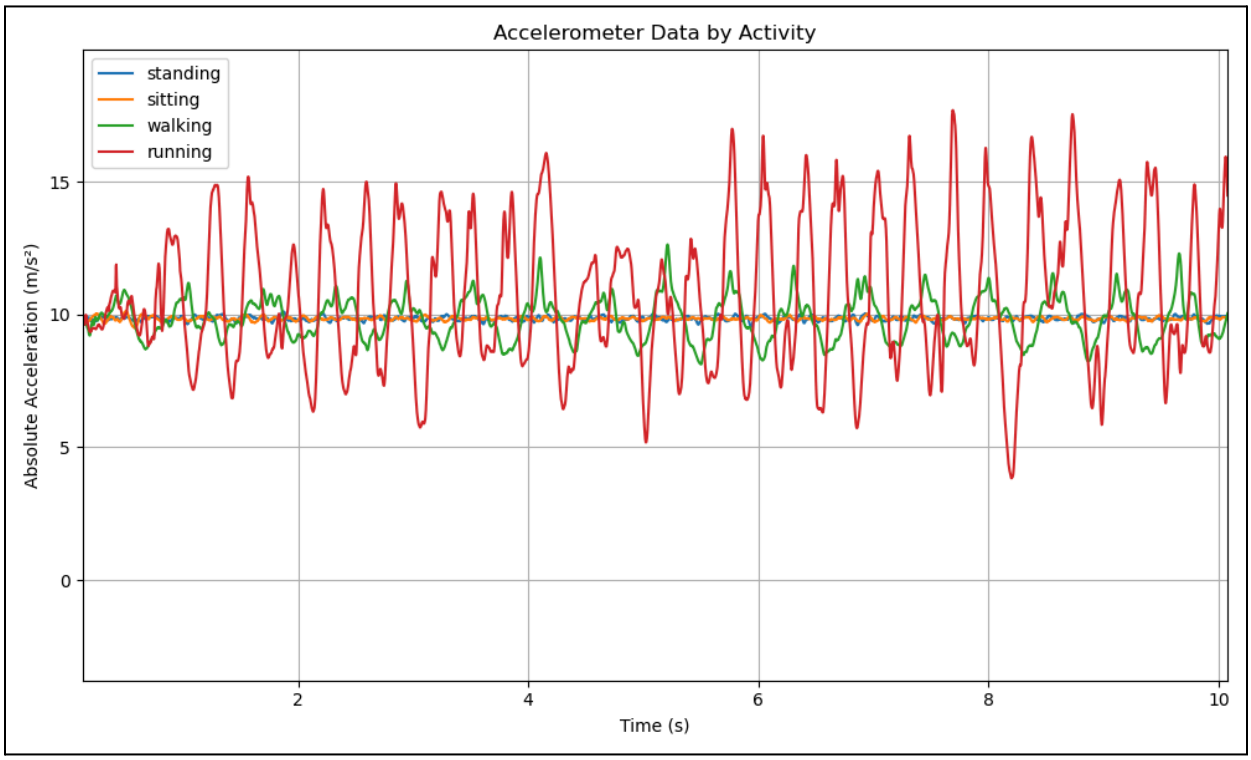
The figures below show the absolute acceleration magnitude for four activities — standing, sitting, walking, and running — recorded using a smartphone accelerometer. The first figure provides an overview, while the other three figures offer close-up views.

Standing and sitting produce nearly constant acceleration around 9.8 m/s² with minimal variation, indicating little to no movement. Walking shows moderate, periodic oscillations, reflecting regular steps, while running exhibits large, high-frequency peaks due to stronger and faster body movements. These differences in variance, amplitude, and frequency of acceleration make it possible to distinguish between the activities.


Accelerometer Data by Activity
standing
sitting
walking
running
Absolute Acceleration (m/s²)
Time (s)
0
20
40
60
80
0
50
100
150
200
250
300


Accelerometer Data by Activity
standing
sitting
walking
running
Absolute Acceleration (m/s²)
Time (s)
8
9
10
11
12
13
225.0
227.5
230.0
232.5
235.0
237.5
240.0
242.5


Accelerometer Data by Activity
10.75
10.50
10.25
10.00
9.75
9.50
9.25
231.5
232.0
232.5
233.0
233.5
234.0
234.5
235.0
235.5
Time (s)
standing
sitting
walking
running

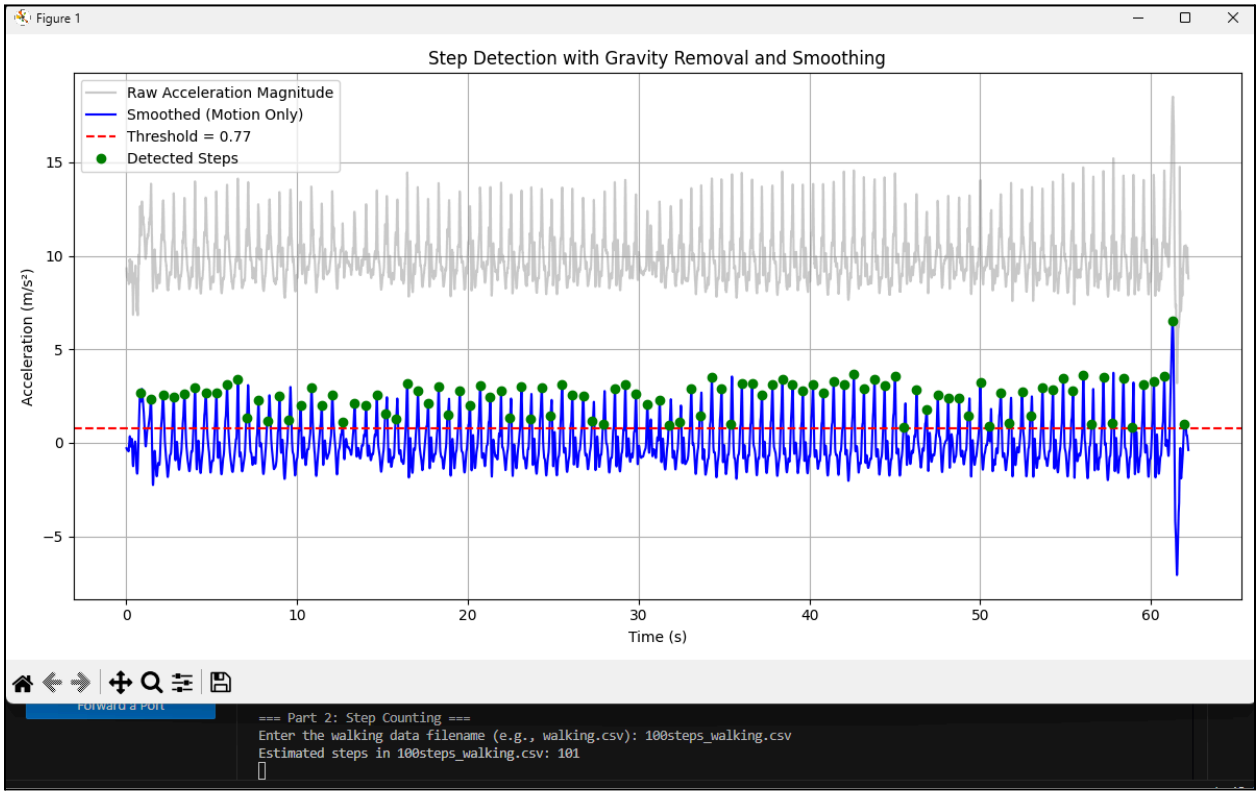

Accelerometer Data by Activity
standing
sitting
walking
running
15
10
5
0
Absolute Acceleration (m/s²)
2
4
6
8
10
Time (s)

Part 2

The function count_steps() accepts a CSV filename as input, uses manually implemented filtering and sliding window techniques, avoids shortcut functions, and does not rely on any third-party libraries for peak detection.

It estimates the number of walking steps from accelerometer data of the given CSV file. It first calculates the total acceleration magnitude from the x, y, and z axes, then uses a low-pass filter to estimate the gravity component. By subtracting this gravity signal, it isolates motion-related acceleration. A manual sliding window smoothing technique is then applied to reduce noise and highlight periodic motion patterns corresponding to steps. The function detects steps by identifying local peaks in the smoothed signal that exceed an adaptive threshold based on the signal's standard deviation, while enforcing a minimum time interval between consecutive steps to prevent double-counting. It outputs the estimated step count and generates a visualization showing the raw, smoothed, and thresholded acceleration with detected steps marked.

The figures below show the results of our trial, based on the data we collected. We also counted our steps to verify the accuracy of our function. You can see the estimated number of steps in the terminal below the images, and the actual number of steps taken in the file names.



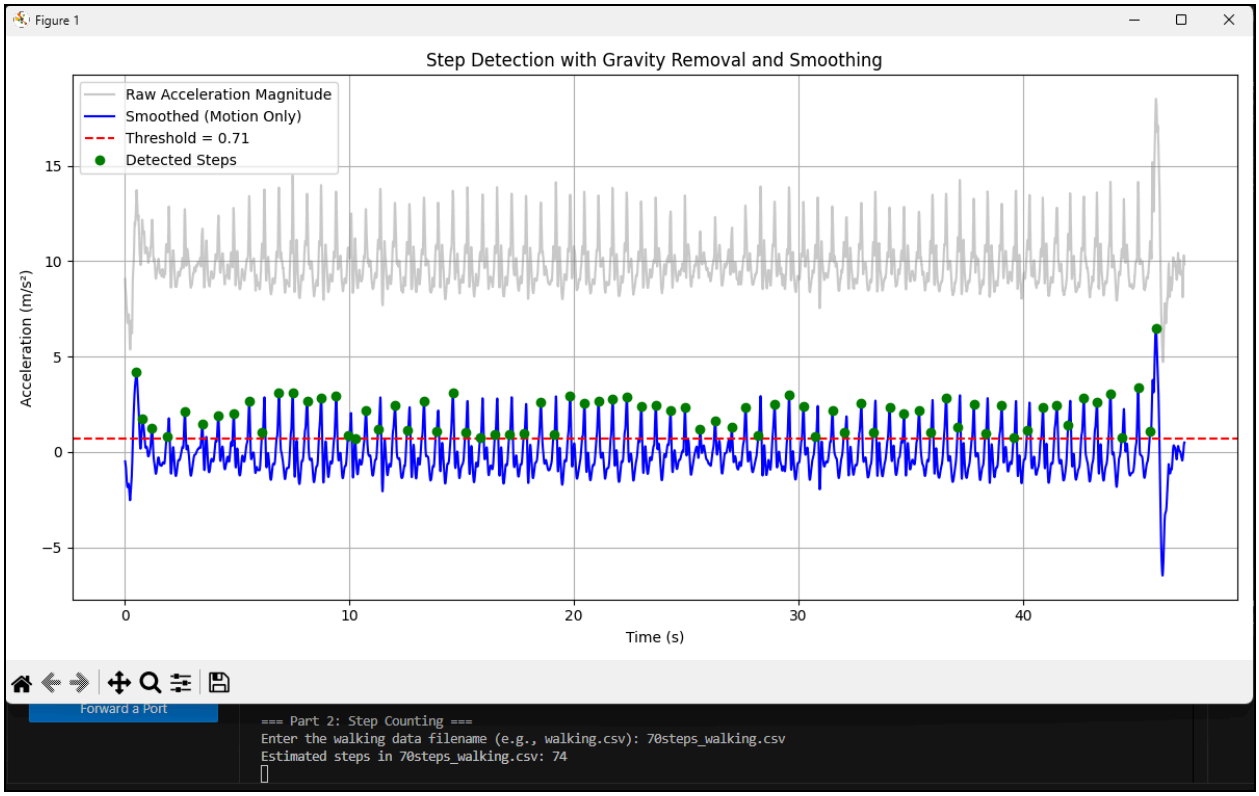


Figure 1
Step Detection with Gravity Removal and Smoothing
Raw Acceleration Magnitude
Smoothed (Motion Only)
Threshold = 0.71
Detected Steps
Acceleration (m/s²)
Time (s)
Forward a Port
=== Part 2: Step Counting ===
Enter the walking data filename (e.g., walking.csv): 70steps_walking.csv
Estimated steps in 70steps_walking.csv: 74

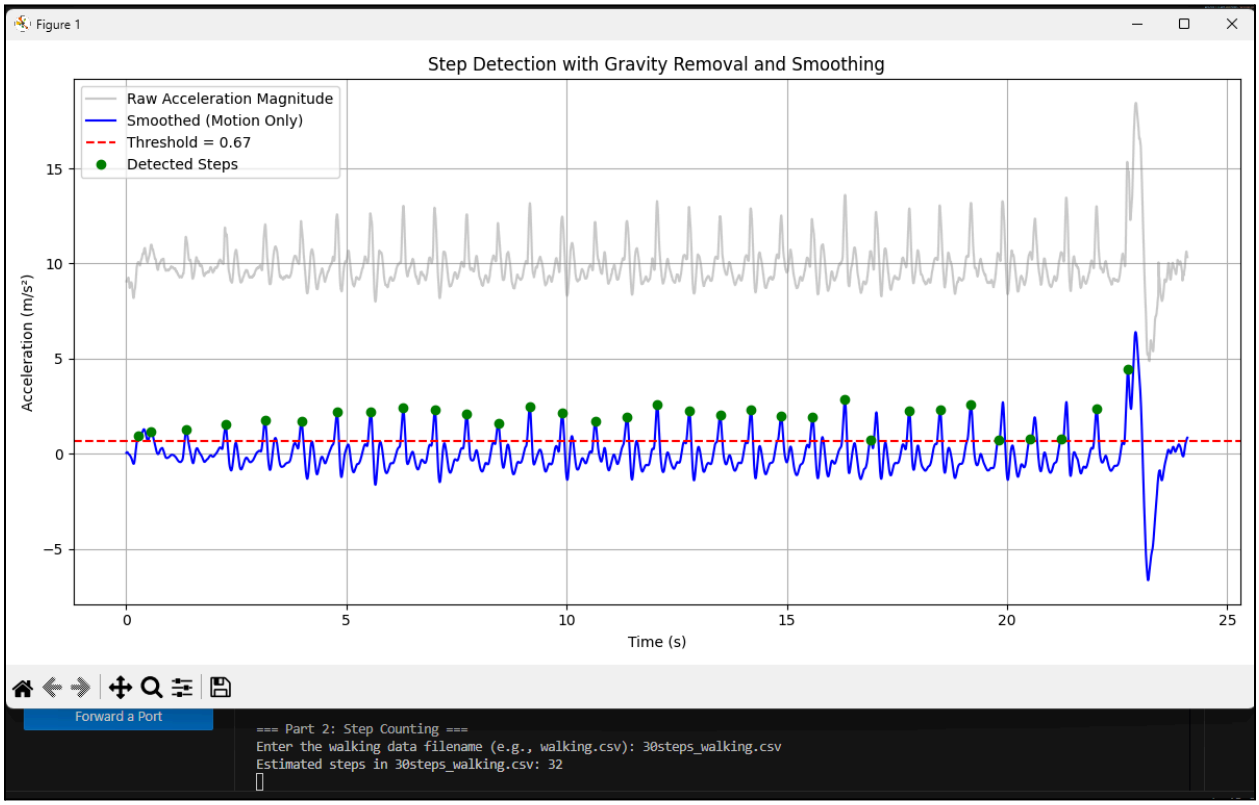


Figure 1
Step Detection with Gravity Removal and Smoothing
Raw Acceleration Magnitude
Smoothed (Motion Only)
Threshold = 0.67
Detected Steps
Acceleration (m/s²)
Time (s)
Forward a Port
=== Part 2: Step Counting ===
Enter the walking data filename (e.g., walking.csv): 30steps_walking.csv
Estimated steps in 30steps_walking.csv: 32

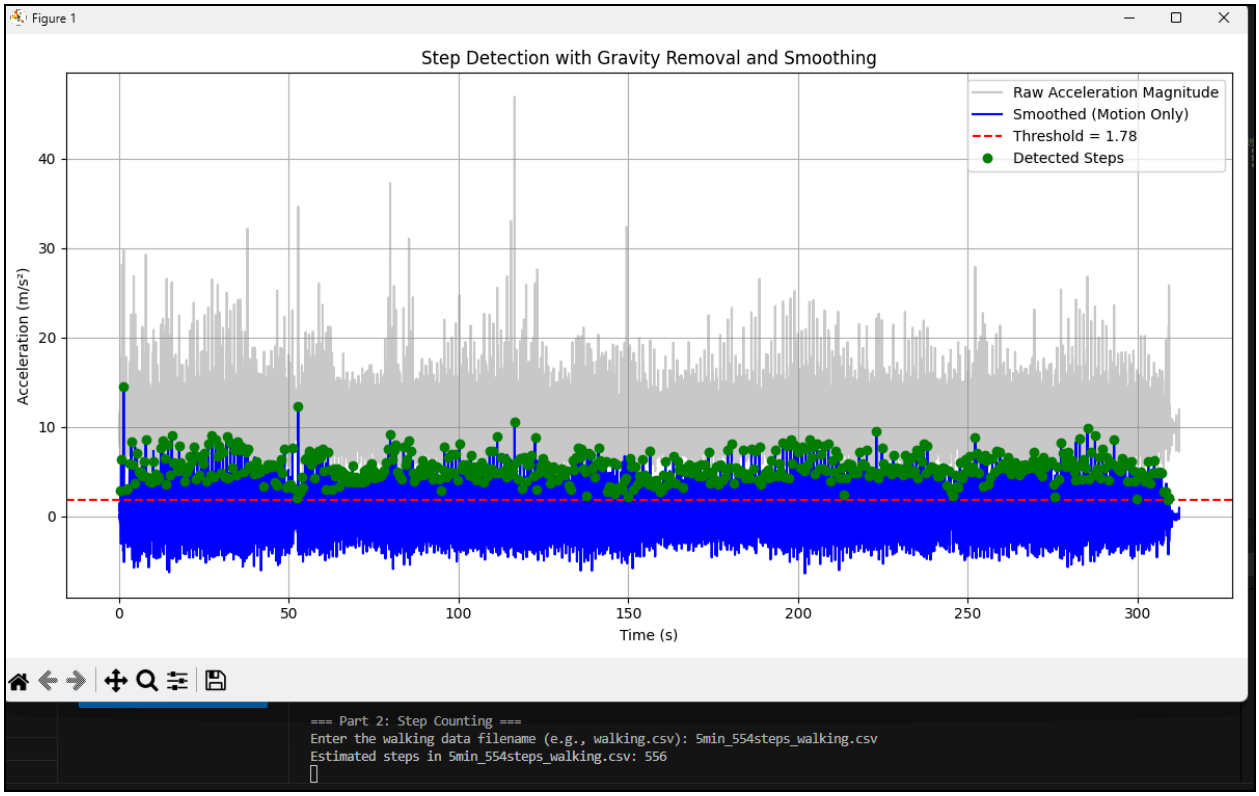


Figure 1
Step Detection with Gravity Removal and Smoothing
Raw Acceleration Magnitude
Smoothed (Motion Only)
Threshold = 1.78
Detected Steps
Acceleration (m/s²)
Time (s)
0
50
100
150
200
250
300
0
10
20
30
40
=== Part 2: Step Counting ===
Enter the walking data filename (e.g., walking.csv): 5min_554steps_walking.csv
Estimated steps in 5min_554steps_walking.csv: 556

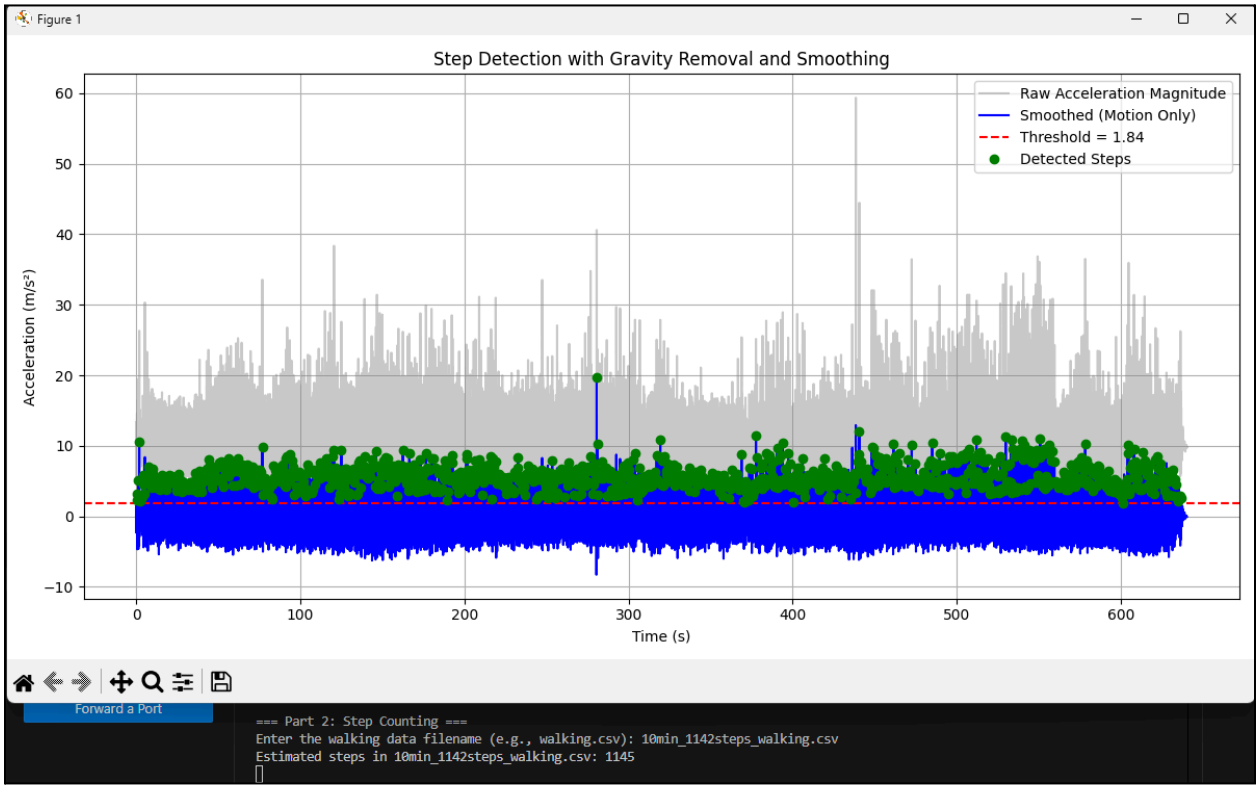


Figure 1
Step Detection with Gravity Removal and Smoothing
Raw Acceleration Magnitude
Smoothed (Motion Only)
Threshold = 1.84
Detected Steps
Acceleration (m/s²)
Time (s)
0
100
200
300
400
500
600
−10
0
10
20
30
40
50
60
Forward a Port
=== Part 2: Step Counting ===
Enter the walking data filename (e.g., walking.csv): 10min_1142steps_walking.csv
Estimated steps in 10min_1142steps_walking.csv: 1145

Part 3

The function estimate_pose_all() estimates the device's orientation—represented by roll, pitch, and yaw angles—for four activities: sitting, standing, walking, and running. It reads accelerometer and gyroscope data from corresponding folders, synchronizes their timestamps, and applies a complementary filter to fuse the two signals. The accelerometer provides long-term stable orientation estimates, while the gyroscope provides short-term accurate rotational changes. The complementary filter balances both using a blending factor ($\alpha = 0.98$), where higher α values give more weight to the gyroscope. For each activity, the code computes the orientation over time, plots the roll, pitch, and yaw angles, and saves them into a pose_estimation.csv file. This approach effectively tracks device motion and tilt during various activities, visualizing how body or phone orientation changes over time.

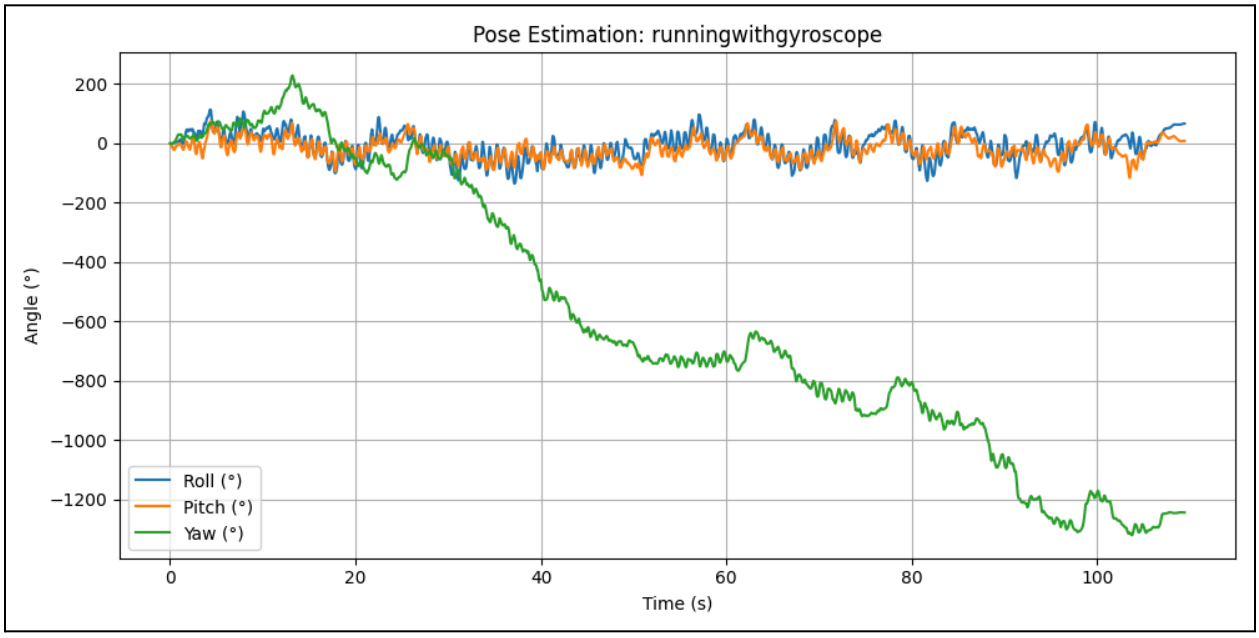
Terminal Output:

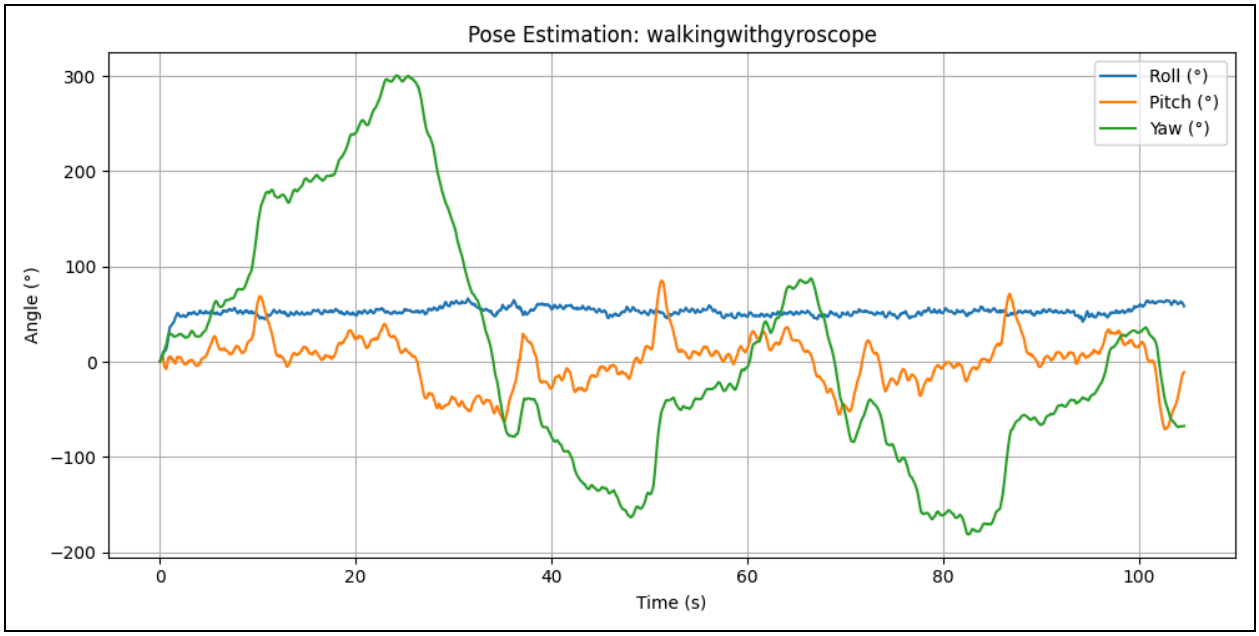
=== Part 3: Pose Estimation for All Activities ===
Pose estimation saved for sittingwithgyroscope -> sittingwithgyroscope\pose_estimation.csv

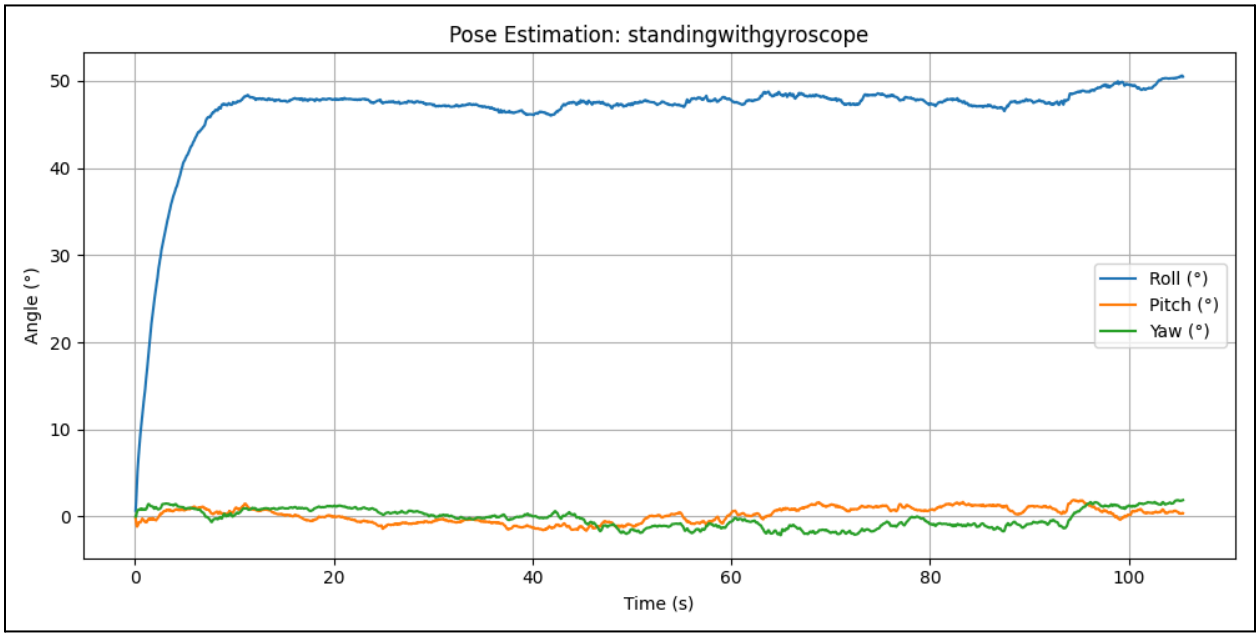
Pose estimation saved for standingwithgyroscope -> standingwithgyroscope\pose_estimation.csv

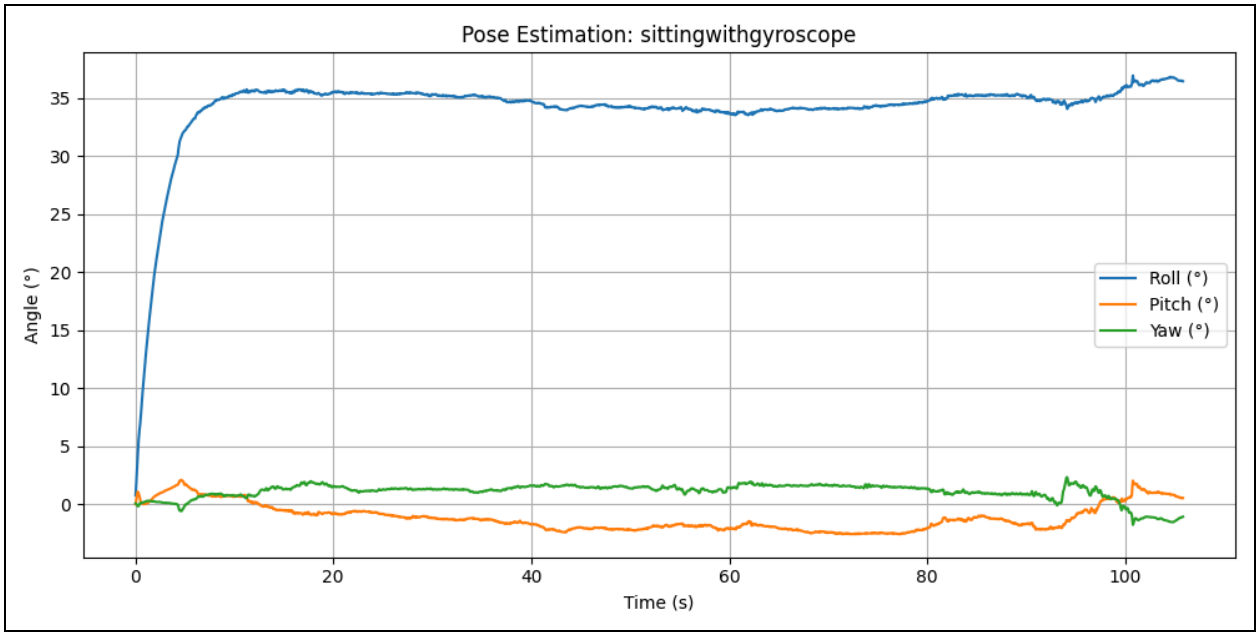
Pose estimation saved for walkingwithgyroscope -> walkingwithgyroscope\pose_estimation.csv

Pose estimation saved for runningwithgyroscope -> runningwithgyroscope\pose_estimation.csv




Pose Estimation: walkingwithgyroscope
Roll (°)
Pitch (°)
Yaw (°)
Angle (°)
Time (s)
300
200
100
0
−100
−200
0
20
40
60
80
100


Pose Estimation: standingwithgyroscope
Roll (°)
Pitch (°)
Yaw (°)
Angle (°)
Time (s)
50
40
30
20
10
0
0
20
40
60
80
100


Pose Estimation: sittingwithgyroscope
Angle (°)
Time (s)
Roll (°)
Pitch (°)
Yaw (°)
35
30
25
20
15
10
5
0
0
20
40
60
80
100