Project 2 - Human Activity Monitor

Name: Gabriel Visorde	ID: 8134974
Name: Aidan Roberts	ID: 8150847
Name: Panatda Long	ID: 8130139
Name: Sharurfti Zahin	ID: 8147538

Date: 17/5/2025

Project-Based Learning Assignment: Accelerometer Data Analysis for Activity Monitoring

Background: Due to increasing concerns regarding sedentary lifestyles and their impact on public health, coupled with the growing potential of wearable technology for remote health monitoring, the Department of Health and Human Services (DHHS) has commissioned the development of a health monitoring tool capable of tracking physical activity levels in individuals. This project focuses on a core component of such a tool: utilizing accelerometer data to accurately classify periods of movement and rest, providing valuable insights into activity patterns for public health initiatives and personal fitness tracking. The management of a person's activity when engaged in sporting or normal daily activities is also becoming increasingly important. Elite athletes need to monitor the proportion of time they spend on running, walking, or standing still during training sessions or actual match play. Amateur athletes or the average person may also be interested in their own activity data. Monitoring idleness/walking/running patterns is also important in the study of some animals. This project aims to contribute to the foundational technology required for such monitoring systems.

1. Aim (Written by: Aidan)

The primary aim of this project is to develop and implement a method for analyzing accelerometer data to accurately classify periods of human activity as either 'moving' or 'stationary'. This involved collecting accelerometer data during controlled periods of movement and stillness, processing the raw data to reduce noise, and applying a gradient-based thresholding technique to distinguish between the two activity states. A secondary objective was to evaluate the effectiveness of different filtering criteria based on the number of axes exceeding the movement threshold.

Whilst also comprehensively analyzing accelerometer data to assess activity at any given time, and provide a report on daily activity. To further inform the user on habits and times of interest or when activity could be done, or improved upon.

2. Initial Investigations (Written by Sharufti)

Before embarking on the practical implementation, initial investigations were conducted to understand the fundamental principles of accelerometry and its application in activity monitoring. This involved researching:

- **Accelerometer Technology:** How tri-axial accelerometers measure acceleration along three axes (X, Y, and Z). Understanding that acceleration includes both gravitational acceleration and motion-induced acceleration is crucial.
- **Activity Monitoring with Accelerometers:** Exploring existing methods and algorithms used for activity recognition (e.g., simple thresholding, machine learning approaches). Recognizing that simple thresholding on raw or smoothed data can be effective for basic moving/stationary detection.
- **Signal Processing Techniques:** Investigating methods to preprocess accelerometer data, particularly noise reduction techniques like moving averages or low-pass filters, which are essential for improving the reliability of threshold-based classification.
- **Data Acquisition:** Identifying suitable tools or applications for recording accelerometer data from a mobile device, such as the Sensor Logger app, which allows for easy data export in formats like CSV.
- **Data Analysis Environment:** Confirming the suitability of MATLAB as a programming environment for importing, processing, analyzing, and visualizing the accelerometer data due to its strong capabilities in numerical computation and plotting.

These initial investigations provided the theoretical foundation and practical considerations necessary to design the data acquisition and analysis methodology for the project.

3. Equipment List (Written by Aidan)

The following equipment was used for this project:

- **Smartphone with built-in tri-axial accelerometer:** A device capable of running a sensor logging application. (e.g., Android phone used with Sensor Logger app).
- **Sensor Logger:** A mobile application installed on the smartphone to record accelerometer data over time.
- **Computer:** A desktop or laptop computer for data transfer, processing, and analysis.
- **MATLAB Software:** The programming environment used for writing and executing the data analysis script.
- **Accelerometer.csv Data File:** The exported data file containing time-stamped accelerometer readings (X, Y, Z acceleration and time).

4. Setup Procedures and Measurements (Written by Sharufti)

The setup and measurement procedures involved the following steps:

1. **Sensor Logger Setup:**
   - The Sensor Logger application was installed on a smartphone.
   - The application was configured to record data from the accelerometer sensor.
   - The default sampling rate of $100\ Hz$ was used.
   - The required data streams (T, X, Y, Z acceleration) were ensured to be selected for recording.
2. **Data Recording:**
   - The smartphone was held securely by the person whose activity was being monitored. For this project, the phone was held upright parallel to the torso, with the z-axis directly parallel to the subject's direction.
      i. This was done to human error and interference during movement.
   - The recording was started in the Sensor Logger app.
   - A sequence of activities, including periods of being stationary (standing still) and periods of movement (walking/running), was performed for a total duration of approximately 71 seconds.
   - The recording was stopped in the Sensor Logger app.
3. **Data Export:**
   - Within the Sensor Logger app, the recorded data session was located.
   - The data was exported in CSV format, named `Accelerometer.csv`.
4. **Data Transfer:**
   - The `Accelerometer.csv` file was transferred from the smartphone to the computer using a USB cable.
5. **MATLAB Setup:**
   - MATLAB was ensured to be installed and functional on the computer.
   - The `Accelerometer.csv` file was placed in the MATLAB working directory or the correct file path was specified in the script.

These steps ensured that the raw accelerometer data, necessary for the subsequent analysis, was successfully acquired and made available in the appropriate format for processing in MATLAB.

5. Data Analysis (Written by Gabriel)

The analysis of the collected accelerometer data was performed using a custom MATLAB script. The process involved several key steps:

1. **Data Loading and Preparation:** The 'Accelerometer.csv' file was loaded into MATLAB. The raw data columns corresponding to total time, time, z-axis acceleration, y-axis acceleration, and x-axis acceleration were extracted and assigned to respective variables (`t_total`, `t`, `z`, `y`, `x`).
2. **Moving Average Filtering:** To reduce noise and smooth the data, a moving average filter was applied to the acceleration data for each axis (x, y, and z). A window size of 34 was chosen for this filter. The filtered data for each axis was stored in new variables (`x_ma`, `y_ma`, `z_ma`).
3. **Gradient Calculation:** The instantaneous gradient of the smoothed acceleration data for each axis was calculated. This was done by finding the difference in acceleration between consecutive time points and dividing by the difference in time. This gradient represents the rate of change of acceleration.
4. **Movement Thresholding:** A gradient threshold of 2 was defined. This threshold was used to determine whether the change in acceleration at any given point indicated movement. If the absolute value of the gradient for an axis exceeded this threshold, that time point was initially classified as 'moving' for that specific axis.
5. **Activity Classification (Individual Axes):** For each axis (x, y, and z), the time points were classified as either 'moving' or 'stationary' based on whether the individual axis's gradient exceeded the defined threshold. These classifications were plotted against time to visualize the activity state as determined by each axis independently.
6. **Activity Classification (Combined Axes - Two Methods):** Two different methods were employed to classify the overall activity state (moving or stationary) based on the combined information from all three axes. These two methods were used to compare how different criteria for combining the axis data impacted the sensitivity of the classification to movement.
    - **Method 1 (All Axes Threshold):** A time point was classified as 'moving' only if the absolute gradient for *all three* axes (x, y, and z) simultaneously exceeded the gradient threshold. If this condition was not met, the time point was classified as 'stationary'.
    - **Method 2 (At Least One Axis Threshold):** A time point was classified as 'moving' if the absolute gradient for *at least one* of the three axes (x, y, or z) exceeded the gradient threshold. If this condition was not met for any axis, the time point was classified as 'stationary'.

7. **Interval-Based Majority Rule Filtering:** To refine the classifications from the combined axes methods and mitigate the impact of short-duration fluctuations, a majority rule was applied over intervals of 100 time steps (10 ms for every step). Within each interval, if the majority of time points were initially classified as 'moving' (based on the combined axes method), all points within that interval were reclassified as 'moving'. Conversely, if the majority were classified as 'stationary', all points in the interval were reclassified as 'stationary'. This is done to reduce noise in that second of time in the interval.
8. **Time Calculation:** The total time spent in the 'moving' and 'stationary' states was calculated based on the final classifications from each of the combined axes methods.
9. **Data Visualisation:** Plots were generated to visualise the raw 3D accelerometer data, the smoothed acceleration data for each individual axis with the initial moving/stationary classifications, and the final moving/stationary classifications based on the two combined axes methods

The algorithm used above is expressed visually using the flowchart below:

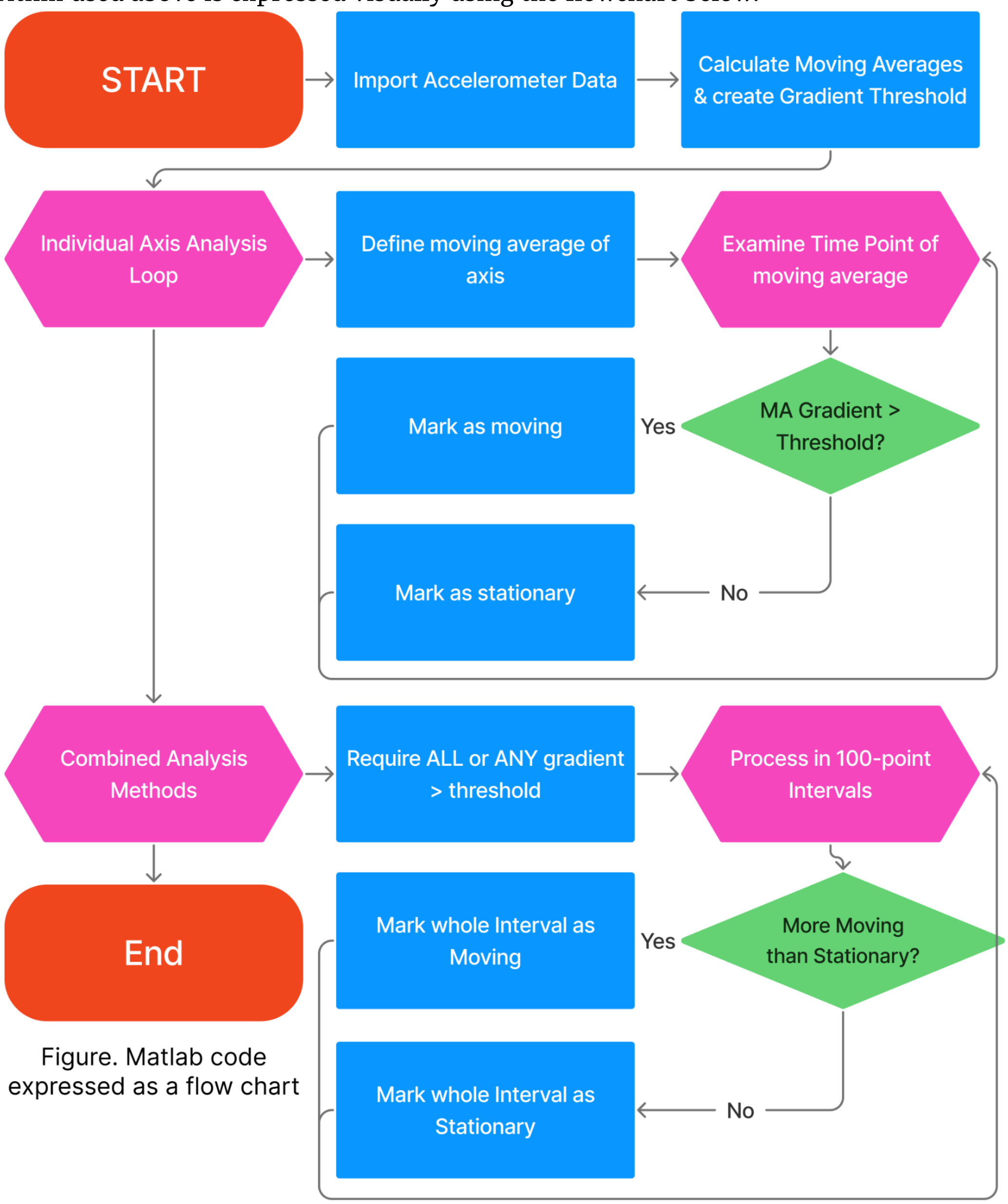

Figure. Matlab code expressed as a flow chart

6. Results (Written by Gabriel)

The analysis produced several plots and numerical outputs detailing the activity classification based on the accelerometer data.

Figure 1: 3D Accelerometer Data: This plot provided a visual representation of the path traced by the accelerometer in 3D space over the recording period. Periods of significant movement would typically show larger excursions and more complex trajectories compared to stationary periods.

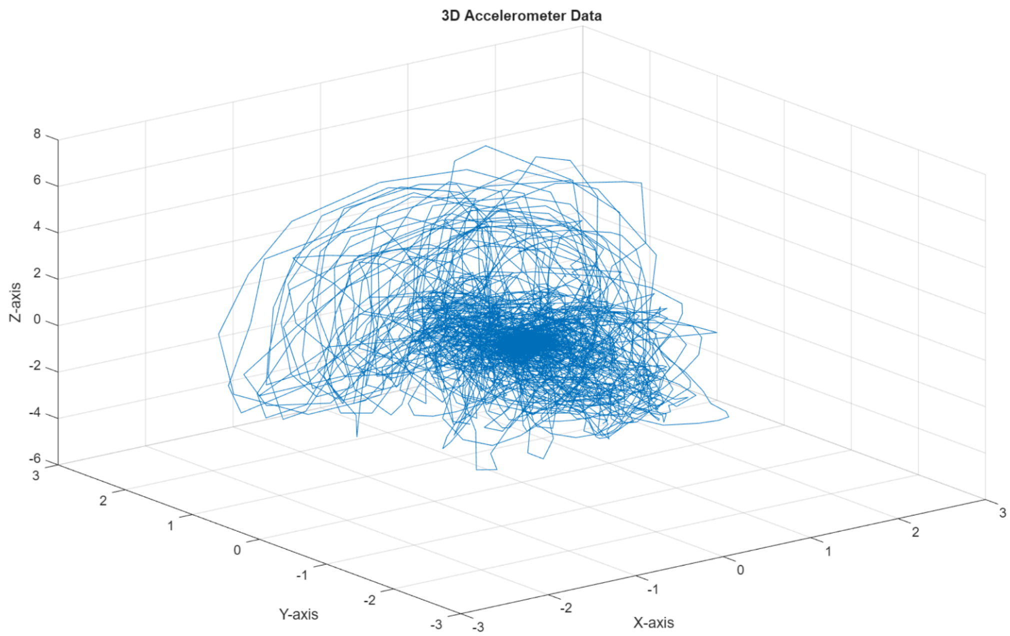

Figure 1. 3D Accelerometer Data

Figures 2-4: Individual Axis Moving Average and Classification: These plots (one for each of the X, Y & Z axes, corresponding to Figures 2, 3, and 4, respectively) showed the smoothed acceleration values over time. The overlaid markers clearly indicated which time points were classified as 'moving' (magenta '+') or 'stationary' (cyan '*') based solely on the gradient of that specific axis's moving average exceeding the threshold. Observing these plots individually allowed for an understanding of how movement affected each axis and how the thresholding performed on a single-axis basis. It was expected that during movement, the moving average plots would show larger fluctuations and steeper gradients, leading to more 'moving' classifications, while stationary periods would exhibit relatively flat lines and primarily 'stationary' classifications.

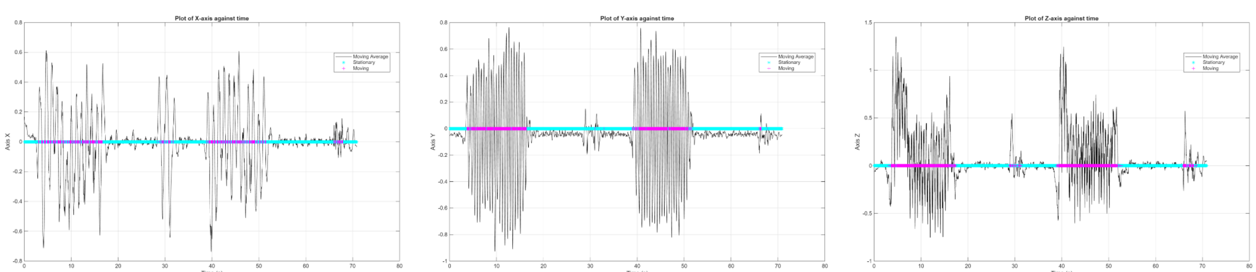

Figure 2-4. Smoothed Individual axes with movement classification overlays

Figures 5-6: Combined Axes Classification: These plots presented the time points classified as 'moving' or 'stationary' based on the two different combined-axis criteria.

- **Figure 5 (Criterion 1 - All Axes > Threshold):** This plot showed time points where the absolute gradient for all three axes simultaneously exceeded the threshold. This criterion was expected to be more stringent and likely classify only periods of significant, multi-directional movement as 'moving'.

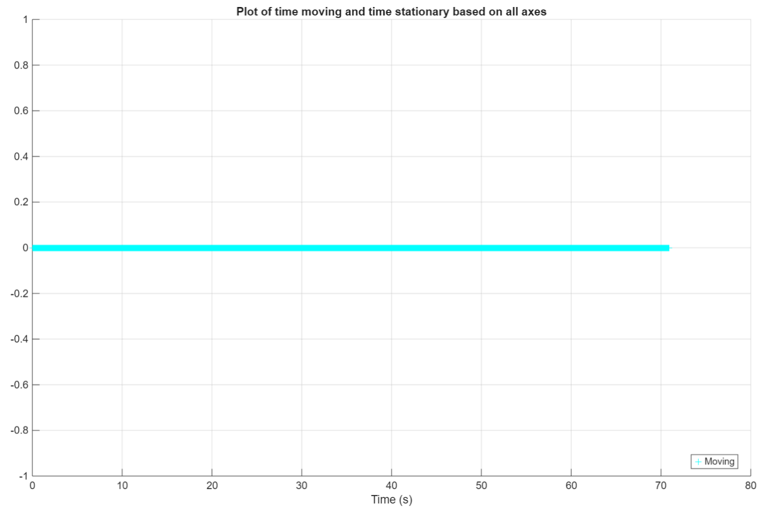

Figure 5. Plot of time moving and time stationary based on all axes

- **Figure 6 (Criterion 2 - At Least 1 Axis > Threshold):** This plot showed time points where the absolute gradient for at least one of the three axes exceeded the threshold. This criterion was expected to be less stringent than Criterion 1 and likely classify a broader range of movements as 'moving', including movements that might primarily involve one or more axes.

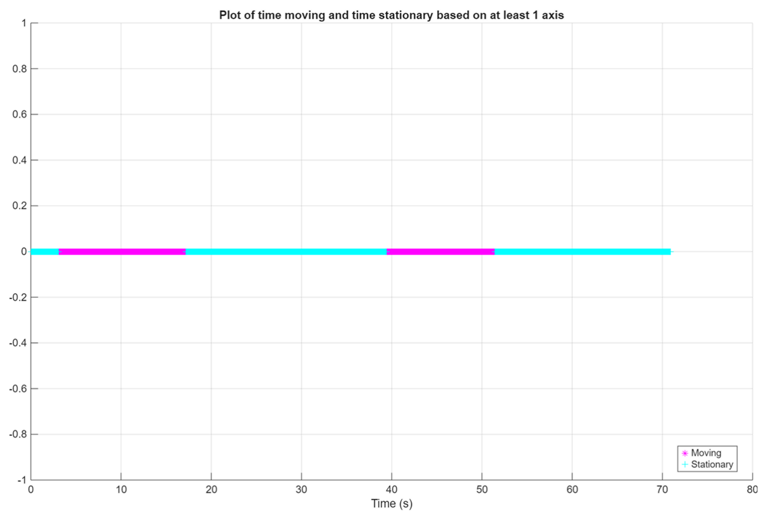

Figure 6. Plot of time moving and time stationary based on at least one axis

Both figures were generated to provide a visual summary of the final activity classifications based on the two different combined criteria after the interval-based majority rule filtering was applied.

Numerical Output: The MATLAB command window displayed the calculated total time spent in 'moving' and 'stationary' states for both combined-axis criteria, with a total time of 70.8924 seconds.

Based on filtering if all axes are above the gradient threshold:	Based on filtering if at least 1 axis is above the gradient threshold:
Time moving: 0 seconds Time stationary: 71 seconds	Time moving: 26 seconds Time stationary: 45 seconds

7. Discussion of Results and Observations (Written by Gabriel)

The results obtained from the accelerometer data analysis provide insights into the feasibility of using this method for activity classification. The 3D plot of the raw data visually demonstrated the path traced by the accelerometer during the recording period, offering a qualitative sense of the movement involved. However, while the 3D plot was useful for visualization, directly using the raw 3D data for automated classification proved challenging. Defining clear and consistent thresholds or patterns in three-dimensional space that reliably differentiate between subtle stationary movements and the beginning of intentional motion is significantly more complex than working with the gradients of individual axes.

The plots of the individual axes, along with their initial moving/stationary classifications, highlighted how movement impacts the acceleration readings on each axis differently. It was observed that periods of deliberate movement, such as walking or running, generally corresponded to larger fluctuations and steeper gradients in the acceleration data compared to periods of stillness.

The application of the moving average filter was crucial in smoothing the data and reducing the impact of noise, which could otherwise lead to spurious classifications of movement during stationary periods or vice versa. The gradient threshold served as a simple yet effective criterion for identifying significant changes in acceleration indicative of motion.

The comparison between the two combined axes methods for activity classification revealed differences in the calculated time spent moving and stationary. Method 1, which required all three axes to exceed the gradient threshold, is a more stringent criterion for classifying movement. This approach is likely to be less sensitive to minor movements or vibrations that might only significantly affect one or two axes. Consequently, it reported a lower total time spent moving and a higher total time spent stationary compared to Method 2. The calculated time moving for Method 1 was 0 seconds, while the time stationary was 71 seconds.
Method 2, which classifies a time point as moving if at least one axis exceeds the threshold, is a more lenient criterion. This approach is more sensitive to movement and is likely to capture a wider range of activities, including those that might primarily involve motion along a single axis. As a result, it reported a higher total time spent moving and a lower total time spent stationary. The calculated time moving for Method 2 was 26 seconds, while the time stationary was 45 seconds.
The total recording time was 70.8924 seconds, with the sum of the moving and stationary times for each method equalling exactly.
As the stationary time for Method 1 accounted for the total recording time, the filtering was evidently too stringent; requiring all three axes to be above the gradient threshold. Conversely, the filtering of Method 2 was more precise, effectively differentiating between motion and rest.

The interval-based majority rule filtering applied after the initial combined axes classification helped to refine the results by smoothing out short-term misclassifications. This process amalgamated every hundred time intervals into one second and effectively reclassified the data to reduce noise.

Due to experimental limitations, movement classification relied solely on acceleration magnitude. This can be misleading because acceleration measures the rate of change of velocity, not velocity or displacement. For instance, an object moving at a constant velocity has zero acceleration, which would be incorrectly classified as stationary. This limitation is inherent when using only accelerometer data from smartphones. Therefore, analyzing the gradient of acceleration and combining information across axes offered a more robust approach.
It is also important to note that the accelerometer utilized in this project was pre-calibrated, and consequently, the influence of gravity on the readings was accounted for and excluded from the analytical data. Given this calibration, distinguishing between periods of sitting and standing is not feasible due to the inability to detect changes in acceleration across axes during transitions between these two states. Thus any points that are considered 'stationary' are an amalgamation of either sitting or standing.

Despite its shortcomings, this project demonstrates a basic but functional approach to activity monitoring using readily available technology. The ability to differentiate between moving and stationary periods, even with simple signal processing, suggests that smartphone accelerometers have potential for use in public health initiatives aimed at tracking activity levels. Further refinement of the algorithms, potentially incorporating machine learning techniques or more sophisticated signal processing, could improve accuracy and allow for the classification of different types of movement (e.g., walking vs. running). The cost-effectiveness and widespread availability of smartphones make this a promising avenue for accessible activity monitoring solutions that could support the DHHS's goals.

8. Difficulties and Sources of Error (Written by Panatda, Edited by Aidan)

Several difficulties and sources of error could have influenced the results of this project:

- **Sensor Placement and Orientation:** The position and orientation of the smartphone on the person's body can significantly affect the accelerometer readings. As the position is largely dependent on human motion, there will be error in the position.
- **Calibration of the Accelerometer:** Slight inaccuracies in the accelerometer's calibration could lead to systematic errors in the acceleration measurements.
- **Noise in Accelerometer Data:** Despite using a moving average, raw accelerometer data can contain noise from various sources, including vibrations, sensor limitations, and environmental factors. This noise can affect the accuracy of the gradient calculation and thresholding.
- **Choice of Moving Average Window Size:** The selected window size for the moving average filter is a trade-off between noise reduction and preserving the details of rapid movements. An inappropriate window size could either fail to adequately smooth the data or over-smooth it, obscuring actual movement.
- **Selection of the Gradient Threshold:** The chosen gradient threshold is subjective and highly dependent on the specific activities being monitored and the characteristics of the accelerometer. An improperly set threshold is a significant source of error, leading to misclassification of moving or stationary periods.
- **Definition of "Moving" vs. "Stationary":** Given the limitations of recording instruments in a smartphone, the "moving" and "stationary" points are purely based on acceleration gradients exceeding a threshold. They can not be based off velocity itself, which is more accurate in detecting whether the subject is moving.

Acknowledging these potential difficulties and sources of error is crucial for interpreting the results and planning future improvements to the activity monitoring system.

9. Conclusion (Written by Panatda, Edited by Aidan)

In conclusion, this project successfully developed and implemented a MATLAB-based system for analyzing accelerometer data to classify human activity as either moving or stationary. The aim of processing raw data, applying a moving average filter for noise reduction, and utilizing a gradient-based thresholding method was achieved.

The analysis demonstrated that the gradient of the smoothed accelerometer data is a viable metric for distinguishing between periods of rest and movement. The project explored two different criteria for combining the information from the three accelerometer axes: requiring all axes to exceed the gradient threshold and requiring at least one axis to exceed the threshold. As hypothesized, the criterion requiring at least one axis to exceed the threshold proved to be more sensitive in detecting movement, classifying a greater proportion of the total time as 'moving' compared to the stricter criterion requiring all three axes.

While the system provides a basic classification of activity, the results highlight the importance of carefully selecting parameters such as the moving average window size and the gradient threshold, as these significantly impact the classification outcome. The project also underscored the potential sources of error, including sensor placement, noise, and the inherent simplification of defining activity solely based on acceleration gradients.

For applications requiring higher accuracy or more nuanced activity recognition (e.g., distinguishing between walking and running), more sophisticated techniques such as machine learning algorithms trained on labeled data would likely be necessary. However, for a basic moving/stationary classification, the implemented gradient-based thresholding approach, particularly when considering movement across at least two axes, provides a functional and interpretable solution, fulfilling the primary objective of the project. The project serves as a foundational step in understanding how accelerometer data can be leveraged for activity monitoring, with potential for further development and refinement.

10. References

11. Appendix

```
clear
close all
clc

data=xlsread('Accelerometer.csv'); % Written by Sharufti
variable={'t_total', 't', 'z', 'y', 'x'};

window_size = 34;     % Define window size for the moving average

for i=1:length(variable)
    varName=variable{i};
    eval([varName '=data(:,i);']);
     eval([varName '_ma = movmean(' varName ', window_size);']); % Moving averages
end

gradient=2; % Gradient threshold

figure (1)
plot3(x, y, z)
xlabel('X-axis')
ylabel('Y-axis')
zlabel('Z-axis')
title('3D Accelerometer Data')
grid on

for i = 1:length(variable)-2 % Written by Aidan
    varName=variable{length(variable)+1-i};
     ma_data = eval([varName '_ma']);   % Get moving average data outside the loop
    figure(i + 1)
    h_ma = plot(t, eval([varName '_ma']), 'w'); % Plot moving average

    moving_points = []; % Initialize empty arrays to collect points
    stationary_points = [];

     for j=1:length(t)-1   % Changed to length(t)-1 to avoid index out of bounds
```

```
                if  (ma_data(j+1)-ma_data(j))/(t(j+1)-t(j))>gradient  ||
(ma_data(j+1)-ma_data(j))/(t(j+1)-t(j))<-gradient
            moving_points = [moving_points, t(j)]; % Store moving times
        else
                stationary_points = [stationary_points, t(j)]; % Store
stationary times
        end
    end
    hold on
    h_stationary = plot(stationary_points, zeros(size(stationary_points)),
'c*'); % Plot non-moving points
     h_moving = plot(moving_points, zeros(size(moving_points)), 'm+'); %
Plot moving points

    hold off
    title(['Plot of ', char('X' + i - 1), '-axis against time'])
    xlabel('Time (s)')
    ylabel(['Axis ', char('X' + i - 1)])
    grid on

    % Hardcoded legend using handles
    legend([h_ma, h_stationary, h_moving], 'Moving Average', 'Stationary',
'Moving', 'Location', 'best');
end

for d=1:2 % Written by Gabriel
    total_time_moving = 0; % Initialize time moving variable
    total_time_stationary = 0; % Initialize time stationary variable
    figure(4+d)
    % Initialize empty arrays to collect points
    moving_times = [];
    stationary_times = [];

     for j=1:length(t)-1   % Changed to length(t)-1 to avoid index out of
bounds
        % Calculate gradients for all coordinates
        coords_ma = {x_ma, y_ma, z_ma};
        gradients = zeros(1, length(coords_ma));

        for i = 1:length(coords_ma)
```

```
            coord_ma = coords_ma{i};
                gradients(i) = (coord_ma(j+1) - coord_ma(j)) / (t(j+1) -
t(j));
        end
        if d==1
               if all(abs(gradients) > gradient) % Check if all gradients
exceed the threshold
                moving_times = [moving_times, t(j)];
            else
                stationary_times = [stationary_times, t(j)];
            end
        else
              if sum(abs(gradients) > gradient) >= 1 % Check if at least 2
gradients exceed the threshold
                moving_times = [moving_times, t(j)];
            else
                stationary_times = [stationary_times, t(j)];
            end
        end
          if j == length(t)-1   % Process intervals after all points are
classified
            temp_moving = moving_times;
            temp_stationary = stationary_times;

            % Process in intervals of 100 time steps
            for k = 1:100:length(t)-1
                 end_idx = min(k+99, length(t)-1); % Find ending index for
this interval (avoid going out of bounds)

                interval_times = t(k:end_idx);

                moving_count = sum(ismember(temp_moving, interval_times));
% Count how many moving and stationary points in this interval
                         stationary_count = sum(ismember(temp_stationary,
interval_times));

                   if moving_count > stationary_count % Apply the majority
rule
                       % More moving points - convert all stationary points
in interval to moving
```

```
                    moving_times = union(moving_times,
intersect(stationary_times, interval_times));
                    stationary_times = setdiff(stationary_times,
interval_times);
                    total_time_moving=total_time_moving+1;

                else
                    % More stationary points - convert all moving points
in interval to stationary
                    stationary_times = union(stationary_times,
intersect(moving_times, interval_times));
                    moving_times = setdiff(moving_times, interval_times);
                    total_time_stationary=total_time_stationary+1;
                end
            end
        end
    end
    % Plot all points at once with handles
    hold on
    h_moving = plot(moving_times, zeros(size(moving_times)), 'm*');
    h_stationary = plot(stationary_times, zeros(size(stationary_times)),
'c+');
    hold off
    % Add hardcoded legend to prevent overlap
    legend([h_moving, h_stationary], 'Moving', 'Stationary', 'Location',
'best');
    if d==1
        title('Plot of time moving and time stationary based on all axes')
    else
        title('Plot of time moving and time stationary based on at least 1
axis')
    end
    xlabel('Time (s)')
    grid on
    if d==1
        disp('Based on filtering if all axes are above the gradient
threshold:');
    else
        disp('Based on filtering if at least 1 axes are above the gradient
threshold:');
```

```
    end
    disp(['Time moving: ' num2str(total_time_moving) ' seconds']);
    disp(['Time stationary: ' num2str(total_time_stationary) ' seconds']);
    disp(['Total time: ' num2str(t(end)) ' seconds']);
end
```