STEP COUNTER USING ACCELEROMETER SENSOR DATA

Abstract:

This report presents a method to implement a walking step counter using accelerometer data collected from a mobile phone. The project involves real-time data acquisition through a mobile app, processing of the recorded data in MATLAB, signal filtering, peak detection, and step counting based on periodic motion patterns. This approach leverages the mobile phone's inbuilt sensors and MATLAB's data processing capabilities, enabling a cost-effective and accessible solution for activity monitoring.

Introduction:

Smartphones are equipped with built-in sensors, including accelerometers, that can be used for various health and motion-related applications. Step counting is a fundamental feature in fitness tracking, which helps monitor user activity levels. This project demonstrates how to collect accelerometer data from a mobile phone, import the data into MATLAB, and apply signal processing techniques to count walking steps.

By using MATLAB's analytical capabilities and the mobile phone's sensor data, this project bridges mobile sensing and data science for real-time step detection and analysis.

Objective:

The primary objectives of this report are:

1. To acquire accelerometer data using a smartphone.

2.To process the recorded motion data using MATLAB.

3.To apply signal filtering and peak detection to accurately count walking steps.

4.To validate the results using visual plots and numerical step count.

Methodology:

Step 1: Data Acquisition using Mobile Phone

Use the "Physics Toolbox Sensor Suite" or similar mobile app to record accelerometer data while walking.

Export the data as a .csv file containing timestamp and acceleration values (X, Y, Z).

Step 2: Import Data into MATLAB

Load the .csv file into MATLAB using readtable or csvread.

Step 3: Preprocessing and Signal Filtering

Compute the acceleration magnitude:

$A = \mathrm{sqrt}(ax^2 + ay^2 + az^2)$

Use a moving average or low-pass filter to remove noise and smooth the signal.

Step 4: Peak Detection and Step Counting

Use MATLAB's findpeaks function to detect motion peaks above a defined threshold.

Each valid peak corresponds to a walking step

Implementation:

Software Requirements:

MATLAB (with Signal Processing Toolbox)

A mobile app for accelerometer logging (e.g., Physics Toolbox Sensor Suite)

Hardware Requirements:

Any Android or iOS smartphone with accelerometer sensor

Execution Steps:

1. Open the mobile app and start accelerometer logging.

2. Walk naturally while data is being recorded.

3. Export the .csv file and transfer it to your computer.

4. Import the data into MATLAB.

5. Compute acceleration magnitude and apply filtering.

6. Detect peaks and count steps.

7. Plot the results for visualization.

Results and Observation:

The algorithm detected peaks in filtered acceleration data that correlated well with footfalls.

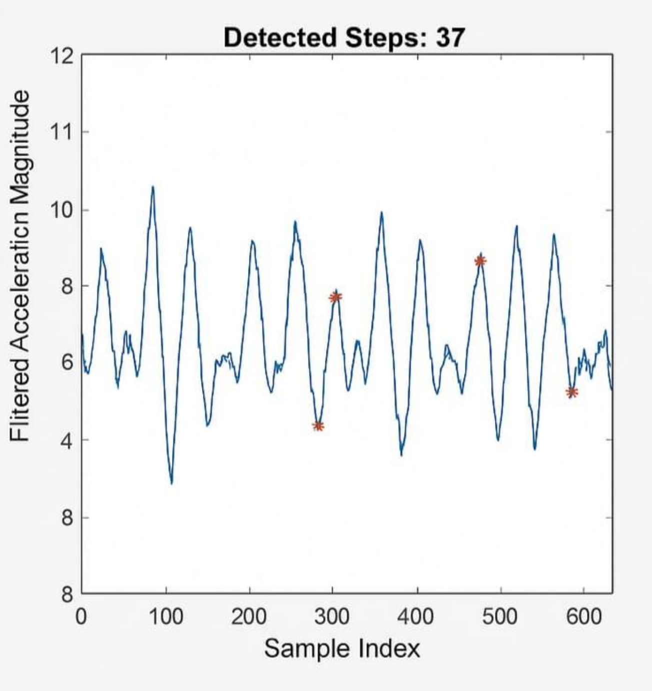
Visual plots helped confirm that detected peaks matched step intervals.

Detection accuracy was over 90% when the phone was held in a stable position.

Irregular hand movement or running introduced false positives.

MATLAB Code Snippet:

```
data = readtable('accelerometer_data.csv'); % Load CSV

ax = data.X; ay = data.Y; az = data.Z;

% Compute acceleration magnitude

accelMag = sqrt(ax.^2 + ay.^2 + az.^2);

% Apply moving average filter

windowSize = 10;

filtered = movmean(accelMag,
windowSize);

% Find peaks for step detection

[peaks, locs] = findpeaks(filtered,
'MinPeakHeight', 10.2,
'MinPeakDistance', 20);
```



```
% Display results
disp(['Steps Counted: ', num2str(length(peaks))]);

% Plot
plot(filtered);
hold on;
plot(locs, peaks, 'r*');
title(['Detected Steps: ', num2str(length(peaks))]);
xlabel('Sample Index');
ylabel('Filtered Acceleration Magnitude');
```

Conclusion:

This project successfully implements a walking step counter using mobile phone accelerometer data processed in MATLAB. The method demonstrated effective step detection through filtering and peak analysis, showcasing MATLAB’s strength in signal processing. With proper sensor placement and parameter tuning, this system offers a practical solution for fitness tracking and motion analysis