Robust Pedometer Algorithm

For Loosely Attached Smartphone

Presented by:

Trivaan M. Singh

Prepared for:

Sampath Jayalath

November 7, 2021

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for the degree of Bachelor of Science in Mechatronics.

Abstract

Pedometers are devices that track the number of steps taken by a user. These devices are commonly used to track one's physical activities, such as walking, jogging and running. Most pedometers though require the user to attach the device to their body, whether it be an arm or foot strap to achieve high accuracies.

This report describes the design of a robust pedometer algorithm that operates under scenarios where the pedometer device is loosely attached to the body. Previous algorithms focus on either gyroscope-based algorithms or accelerometer algorithms as each has its advantages and disadvantages. The proposed algorithm makes use of both accelerometer and gyroscope data to determine a classifier that switches between these algorithms depending on the activity.

The developed algorithm is broken down into three stages: developing an accelerometer-based peak detection algorithm. This tracks the number of steps by counting the peaks in the accelerometer data produced by the vertical bounce during walking. Developing a dominant axis detector that will be used in an existing gyroscope-based zero-crossing algorithm. This tracks the number of steps by counting how many times the gyroscope signal crosses the zero line. Finally, developing a classifier that switches between the two algorithms.

Simulations show that the proposed algorithm successfully detects steps with an average accuracy of 96% for varying device orientations in the front pants pocket and handbag while performing different activities.

Acknowledgments

The completion of this project is attributed to the support from a number of people.

I would first and foremost like to show my deepest appreciation to my family for supporting me and motivating me throughout my years in university. I would not be where I am today without their love and support.

My friends for always checking up on me. The support I receive from our conversations greatly helped with my mental state.

Ashay, for helping me optimize my MATLAB scripts. Your insight was extremely valuable and greatly sped up the runtime of my scripts.

A special acknowledgement to my supervisor, Dr Sampath Jayalath, for providing clear insights into this research project. His guidance and technical knowledge were essential in completing this project.

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have the IEEE convention for citation and referencing. Each contribution to, and quotation in, this final year project report from the work(s) of other people, has been attributed and has been cited and referenced.

3. This final year project report is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as there own work or part thereof.

.....

Trivaan M. Singh
November 7, 2021

Contents

Abstract i

Acknowledgments ii

Plagiarism Declaration iii

Table of Contents iv

List of Figures vii

List of Tables xi

Chapter 1: Introduction 1

1.1 Background to the study 1

1.2 Motivation 2

1.3 Problem Statement 2

1.4 Objectives 3

1.5 Scope and Limitations 4

1.6 Outline 4

Chapter 2: Literature Review 6

2.1 Literature Review 6

2.2 Brief Introduction 6

2.3 Human Gait Cycle . . . 6
2.4 Frequency Domain Study . . . 8
2.5 Accelerometer vs Gyroscope based pedometer . . . 10
2.6 Common Step Detection Methods . . . 12
2.6.1 Peak Detection . . . 12
2.6.2 Zero-crossing . . . 13
2.6.3 Thresholding . . . 13
2.7 Final Thoughts . . . 13

Chapter 3: Application Development **15**
3.1 Hardware . . . 15
3.2 Co-ordinate Frame . . . 16
3.3 Collection of IMU Data . . . 18

Chapter 4: Data Analysis **21**
4.1 Data Analysis for Gyroscope Data . . . 21
4.1.1 Pants Pocket . . . 21
4.1.2 Handbag . . . 28
4.2 Data Analysis for Accelerometer Data . . . 29
4.3 Data Analysis for Sensor Fusion . . . 32

Chapter 5: Algorithm Design **34**
5.1 Pre-processing IMU data . . . 36
5.2 Accelerometer Algorithm . . . 38
5.2.1 Accelerometer Magnitude . . . 38
5.2.2 Filtering . . . 40
5.2.3 Peak Enhancement . . . 44
5.2.4 Peak Detection . . . 46
5.3 Gyroscope Algorithm . . . 47

5.3.1 Dominant Axis Detection 48
5.3.2 Periodicity Check 50
5.4 Overview of the Developed algorithm 52

Chapter 6: Results **53**
6.1 Accuracy of Accelerometer based algorithm 54
6.1.1 Walking on flat ground - slow 55
6.1.2 Walking on flat ground - normal 56
6.1.3 Walking on flat ground - fast 57
6.1.4 Walking up stairs 58
6.1.5 Walking down stairs 59
6.1.6 Walking up a ramp 60
6.1.7 Walking down a ramp 62
6.2 Accuracy of Gyroscope based algorithm 63
6.3 Accuracy of dual sensor-based data implementation 65
6.4 Discussion 66
6.4.1 Discussion of Accelerometer Algorithm Results 66
6.4.2 Discussion of Gyroscope Algorithm Results 68
6.4.3 Discussion of Dual-Sensor Algorithm Results 71

Chapter 7: Conclusions **73**
7.1 Future Work 75

Bibliography **77**

Appendix A: Code **80**

List of Figures

2.1 Human gait cycle with right leg as the reference leg [10] 7
2.2 Spectral analysis on gyroscope data of walking motion [11] 9
2.3 Spectral analysis on accelerometer data of walking motion [11] 9
2.4 Step accuracy of commercial and proposed pedometer algorithms of varying pace of 60 to 130 steps/min [5] 11

3.1 Co-ordinate frame of an Android smartphone [16] 17
3.2 GUI and flow chart of the SensorCSV application 19

4.1 Aerial view of walking with the device placed vertically in the pocket. X-axis is perpendicular to the walking direction 22
4.2 Aerial view of walking with the device placed vertically in the pocket at a translation of 45° 23
4.3 Raw gyroscope reading for device translation of 0° 24
4.4 Raw gyroscope reading for device translation of 90° 25
4.5 Raw gyroscope reading for device translation of 45° 26
4.6 Raw gyroscope reading for device translation of 45°in the left pocket . 27
4.7 Raw gyroscope x,y and z reading for device in a handbag 28
4.8 Accelerometer magnitude data for device in a handbag 29
4.9 Human walking motion demonstrating bounce during each step [18] . 30
4.10 Accelerometer magnitude tracking 6 steps for a pocket use case . . . 30

4.11 Pattern matching between accelerometer magnitude and dominant gyroscope axis 32

5.1 Flow chart of the process of the proposed algorithm 35
5.2 Plot of raw gyroscope and accelerometer readings 36
5.3 Plot of sample rate 38
5.4 Effects of gravity on the total acceleration 39
5.5 FFT of accelerometer magnitude data 41
5.6 Frequency response of a moving average filter with a length of 21 . . 42
5.7 Frequency response of applying the filter of Fig 5.6 twice. 43
5.8 Implementation of moving average filter 43
5.9 False peak caused by high frequency noise 44
5.10 Implementation of Equation 5.5 on filtered accelerometer data 45
5.11 Flow chart of accelerometer algorithm 47
5.12 Dominant axis detection for the device placed at a 45°translational angle 49
5.13 Flow chart of accelerometer algorithm 51

6.1 Raw and filtered data after peak correction while the user was walking on flat ground at a slow pace, with the device placed in a pocket . . . 55
6.2 Raw and filtered data after peak correction while the user was walking on flat ground at a slow pace, with the device placed in a handbag . . 56
6.3 Raw and filtered data after peak correction while the user was walking on flat ground at a normal pace, with the device placed in a pocket . 56
6.4 Raw and filtered data after peak correction while the user was walking on flat ground at a normal pace, with the device placed in a handbag 57
6.5 Raw and filtered data after peak correction while the user was walking on flat ground at a fast pace, with the device placed in a pocket . . . 57

6.6 Raw and filtered data after peak correction while the user was walking on flat ground at a fast pace, with the device placed in a handbag . . 58
6.7 Raw and filtered data after peak correction while the user was walking up stairs with the device placed in a pocket 58
6.8 Raw and filtered data after peak correction while the user was walking up stairs with the device placed in a handbag 59
6.9 Raw and filtered data after peak correction while the user was walking down stairs with the device placed in a pocket 59
6.10 Raw and filtered data after peak correction while the user was walking down stairs with the device placed in a handbag 60
6.11 Raw and filtered data after peak correction while the user was walking up a ramp with the device placed in a pocket 60
6.12 Raw and filtered data after peak correction while the user was walking up a ramp with the device placed in a handbag 61
6.13 Raw and filtered data after peak correction while the user was walking down a ramp with the device placed in a pocket 62
6.14 Raw and filtered data after peak correction while the user was walking down a ramp with the device placed in a handbag 63
6.15 Zero-crossing detection for device placed in a handbag in an arbitrary position 64
6.16 Peak detection for slow walking pace while device was place in a pocket 67
6.17 Peak detection for slow walking pace while device was place in a pocket 68
6.18 Peak detection for normal walking pace while device was place in a pocket 68
6.19 Peak detection for fast walking pace while device was place in a pocket 69
6.20 Zero-crossing detection for normal walking pace while device was placed in a handbag 70

6.21 Zero-crossing detection for normal walking pace while device was placed in a in the pocket. 70
6.22 Implementation of dual-sensor algorithm. The classifier identifies periodic motion therefore the zero-crossing algorithm was implemented. 71
6.23 Implementation of dual-sensor algorithm. The classifier identifies non-periodic motion therefore the peak detection algorithm was implemented. 72

List of Tables

4.1 Varying strengths of single axis contribution based on angle θ 24

6.1 Results of accelerometer algorithm for participants carrying the device in the pocket 54

6.2 Results of accelerometer algorithm for participants carrying the device in the handbag 55

6.3 Results of gyroscope algorithm for participants carrying the device in the pocket 63

6.4 Results of gyroscope algorithm for participants carrying the device in the handbag 64

6.5 Results of dual pedometer algorithm for participants carrying the device in the pocket 65

6.6 Results of dual pedometer algorithm for participants carrying the device in the handbag 65

Chapter 1

Introduction

This report documents research, design, implementation and testing of a robust pedometer algorithm.

1.1 Background to the study

Mobile phone technologies have greatly developed since its invention in 1973 by Martin Cooper of Motorola [1]. Today, most affordable smartphones have built-in sensors that are capable of capturing raw 3-axis data with relatively high precision and accuracy to monitor and analyse the smartphone's movement or positioning. These built-in sensors can be used for serval applications, one of them being pedometers.

A pedometer is a step-tracking device that counts the number of steps taken by the user. Today, pedometers are more widely known as activity tracking devices that monitor the training results of a session, such as the number of steps taken, distance travelled and the number of calories burnt [2]. Pedometers have also become prevalent in the healthcare industry. For instance, it is used to monitor the physical state of recovering patients after cardiac rehabilitation [3] and can be even be used

as indoor navigation applications to help visually impaired persons navigate [4].

Advances in Micro-Electro-Mechanical-Systems (MEMS) technology has manufacturing Inertial Measurement Units (IMU) sensors feasible. These sensors acquire data of the human gait cycle to track their steps and are commonly found on today's smartphones. Accelerometer data-based pedometers are the most prevalent type of pedometer in the marketplace. This type of pedometer usually has high accuracies whilst the user is doing physical activities such as jogging or running but performs poorly during slow walking speeds [5].

The other option is to use gyroscope data to track the human gait. These pedometers usually track the rotation of the dominant axis to determine zero-crossing, which are used for step detection [6]. This makes the gyroscope algorithms more accurate at slower walking speeds as the angular rates will always oscillate around zero. This issue with this method is that the device has to be tracking the periodic motion of the body part it is appended to. This causes inconsistencies in the performance when the device untethered and placed above the waist.

1.2 Motivation

Previous studies do not stipulate whether the device is loosely attached to the user or have assumed the device to be placed in a specific orientation in order to demonstrate good results. The goal of this thesis is to incorporate both accelerometer data and gyroscope data to develop an algorithm that works for real-world applications.

1.3 Problem Statement

The core investigation is to modify existing algorithms to compensate for the condition that the smartphone is loosely attached to the user. This can be broken

down into 2 factors:

- Loosely attached devices will result in high frequency oscillations captured by the IMU, which may result in inaccurate step detection.
- Loosely attached devices may change the position of the smartphone over time, for instance, as a user is walking the phone tilts in their pocket, impacting the strength of the reading which the algorithm is based on.

Therefore, this paper aims to investigate the characteristics of IMU sensor data with the aim to increase robustness and accuracy, while paying special attention on processing the data and how to determine the axis that contributes the most to the walking motion.

1.4 Objectives

The key objectives of this project are to:

- Develop a mobile application using Android Studio to collect data from the IMU of the smartphone.
- Analyse the raw data to identify walking patterns using signal processing techniques.
- Filter the data to remove high frequency noise.
- Develop a dominant axis detector to determine which axis is under the most influence of rotation caused by walking motion
- Develop an accelerometer step detection algorithm using MATLAB that accounts for dynamically changing orientation and slow walking speeds.

- Create a classifier that will switch between the gyroscope-based algorithm or accelerometer-based algorithm depending on the use case.
- Finally, comparing the results with other existing algorithms

1.5 Scope and Limitations

The scope for this project involves creating an android application to capture the IMU data sensor data. Afterwards, extensive analysis needs to be done on the both accelerometer and gyroscope data to discern possible patterns that may be used for step detection. This is followed by the design of the dominant axis detector, accelerometer algorithm, and classifier variable. These three components will make up the final algorithm which will be tested against previously developed algorithms.A time constraint of 12 weeks is given to complete this project.

1.6 Thesis Outline

The structure of this report is laid out to match the process that was followed in order to capture, process and utilize (IMU) data to develop a pedometer algorithm:
Chapter 2, Literature Review: Researching relevant papers that use sensor data for step detection. This section helped in the understanding of how the human gait impacts sensor data. This section also gives a brief look into different algorithm types that were previously used.
Chapter 3, Application Development: This section is aimed to gather human motion tracking information with the aid of IMU sensors onboard an Android smartphone.
Chapter 4, Data Analysis: In order to understand the necessary approach to

develop the pedometer algorithm, an in depth understanding of how different orientations and use cases impact sensor readings is required.

Chapter 5, Algorithm Design: Explains the process and theory behind the developed algorithm

Chapter 6, Results: Explore and discuss the results achieved. Developed algorithm is compared to existing algorithms.

Chapter 7, Conclusions and Future Work: This chapter summarises the work done and makes meaningful recommendations for improvements that would further advance this research.

Chapter 2

Literature Review

2.1 Literature Review

This section aimed at investigating previous related works on pedometers to discover key concepts that need to be understood to develop and improve on existing algorithms.

2.2 Brief Introduction

A pedometer is a device that is used to track the steps taken by an individual, usually for exercise tracking such as walking, jogging and running. Over the past few years, there have been great advancements in the IMU chipset technology, indicating their high potential for navigation systems [7], specifically pedometer systems.

2.3 Human Gait Cycle

Most modern pedometers require the analysis of the human gait for step detection. This requirement is somewhat more detailed than expected and is explained

as follows.

Human walking can be defined as the coordinated movement between two legs, where each leg either provides support or propels the body forward. Furthermore, one foot has to be planted on the ground at all times to maintain balance. The Human gait describes this repetitive motion of walking [8].

Gait analysis is the evaluation of the manner in which a person walks by observing their motion as they walk in a straight-line [9]. The human gait cycle is defined as the walking cycle from the heel contact of a foot to the heel contact of the same foot. The gait is divided into 2 walking phases, the stance phase and the swing phase, as illustrated in Fig. 2.1 [10]:

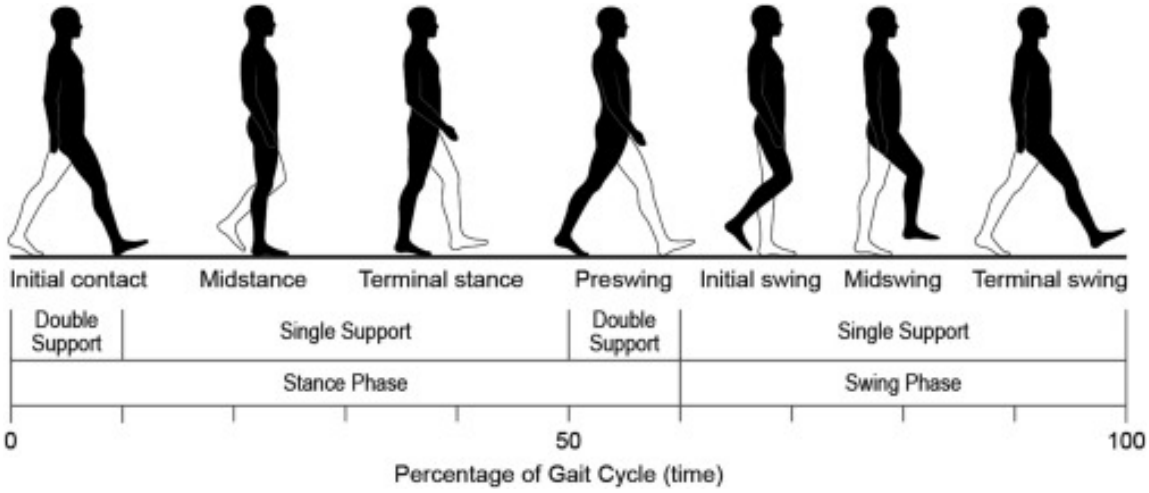


Figure 2.1: Human gait cycle with right leg as the reference leg [10]

The swing phase comprises 40% of the total gait cycle, while the stance phases comprise 60% of the total gait cycle. An important principle to note is that each leg has to be in either one of the two states. During the stance phase, the reference leg is providing support to the other leg, which will be in the swing phase and during the swing phase the reference leg is swinging past the leg that is providing support. These two main phases can be further broken down into sub-phases. Events that

make up the stance phase:

- Initial contact (0%): marks the beginning of the total gait cycle, the main purpose of this event is to establish the foot's contact with the ground
- Mid stance (30%): This is when the femur is directly over the standing foot
- Terminal stance (40%): When the heel lifts off the ground
- Pre-swing (60%): When the toe leaves the ground, i.e., pushed off the ground

Events that make up the swing phase:

- Initial swing (75%): The initial event in which the leg starts to swing, here is where the leg starts accelerating
- Mid swing (85%): The event at which the swinging leg passes the stance leg
- Terminal swing (100%): The termination of a single gait cycle. The swinging leg decelerates and becomes the stance leg to prepare for the next gait cycle

This periodic motion is the basis on which most modern-day pedometer algorithms operate under.

2.4 Frequency Domain Study

The periodic nature of walking is further clearly defined by examining the spectral analysis of the IMU sensor data. Kang et al. [11] conducted spectral analysis on both accelerometry and gyroscopic readings for a human subject walking at a normal pace, as illustrated in Fig. 2.2 and Fig. 2.3.

Fig. 2.2 and Fig. 2.3 clearly shows that the gyroscope data extracts a distinct walking frequency in comparison with accelerometry data. In Fig. 2.2 there is a

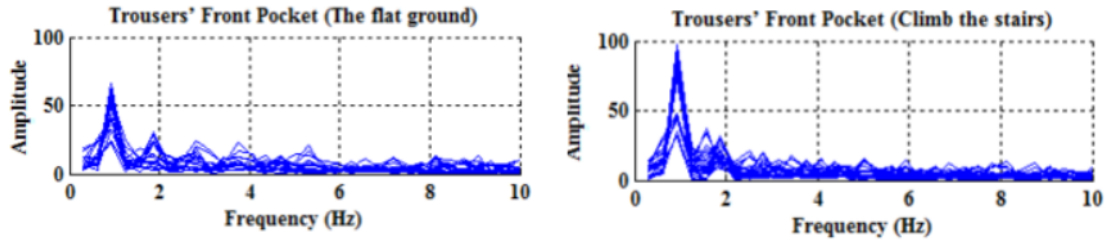


Figure 2.2: Spectral analysis on gyroscope data of walking motion [11]

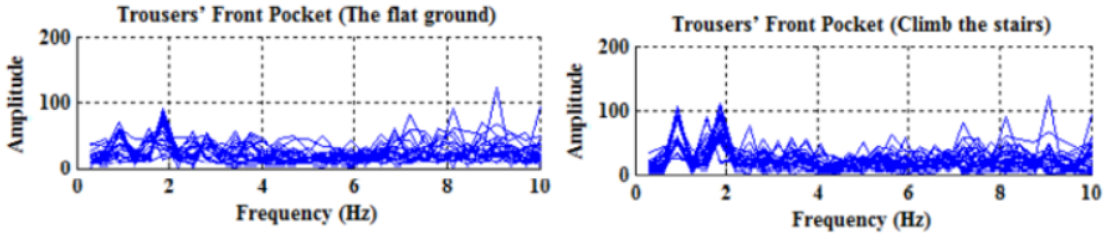


Figure 2.3: Spectral analysis on accelerometer data of walking motion [11]

dominant spike around 0.9Hz – 1Hz, which is expected as the normal walking pace for the average human is between 36 steps/min (0.6Hz) and 120 steps/min (2Hz) [11]. The author of [11] later goes on to incorporate this spectral analysis as the basis of their algorithm. The most sensitive axis of gyroscopic data is found, afterwards, a Fast Fourier transform is applied to extract the walking frequency by looking for a dominant spike between 0.6Hz and 2Hz. A threshold is used to identify when the motion of walking takes place. Whenever the motion of walking was detected a timer value was incremented. Finally, at the end of the dataset, the step count is calculated by multiplying the walking frequency with the timer value.

This study makes use of the cyclical nature of the human gait and uses that information when analyzing the FFT of the gyroscope data. With an accuracy of above 93%, this study proves that periodic motion can be used for step detection. However, this study was only tested for 'normal' walking pace activates so its practicality

under real-world conditions cannot be confirmed. Furthermore, the algorithm works on the basis that the devices need to be appended to the user in such a manner that there will be enough cyclic motion for the spectral analysis to work. For instance, the algorithm in this study is tested under the condition that the device is appended to the leg or arm. Under these conditions, there is sufficient rotation (from the thigh movement or arm swing) for the FFT analysis to determine a walking frequency. Whereas, under conditions when the device is placed above the torso, whether it be in a shirt pocket or a handbag, the device may not undergo sufficient rotation to accurately track steps.

2.5 Accelerometer vs Gyroscope based pedometer

In the current market, most inertial based pedometers operate using accelerometry data. These pedometers are adequately accurate during physical exercise, such as jogging or running but perform poorly at slow walking pace [5].

Oner er al. [5] developed algorithm showed great accuracies walking at speeds over 100 steps per minute but performance vastly degrades as the walking speed drops below 80 steps per minute. Their algorithm overcounted steps at slower cadences, which reported an error margin of 10% for speeds over 80 steps/min, 70% at 70 steps/min and close to a 100% for speeds below 60 step/min as illustrated in Fig. 2.4.

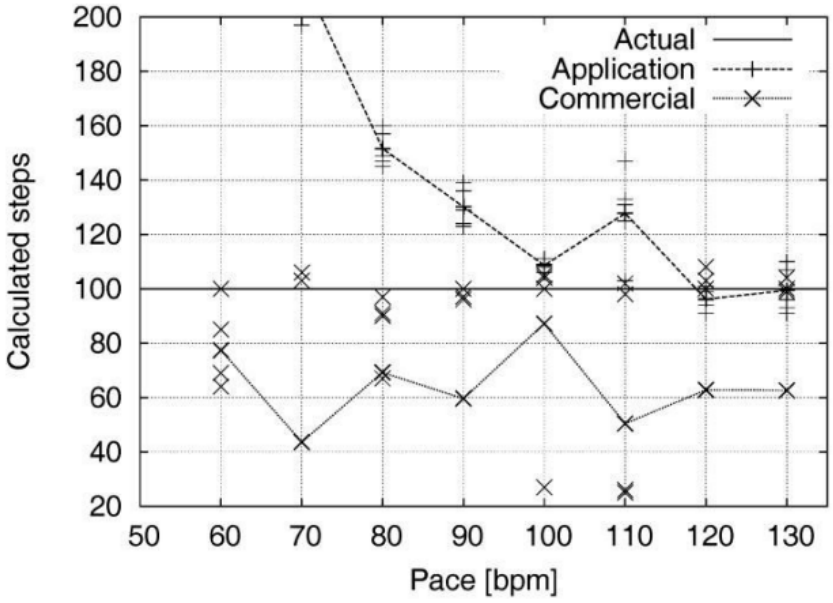


Figure 2.4: Step accuracy of commercial and proposed pedometer algorithms of varying pace of 60 to 130 steps/min [5]

Martin et al. [12], conducted a study that tested the performance of 5 commercially available pedometer in slower walking older adults, with a mean age of 63.6 years. The study tested 3 set cadences of 50, 60 and 80 steps/min with a mean percent error of 56%, 40% and 19% respectively.

The literature discussed here utilized accelerometry data to detect steps and they all performed poorly at slow cadences or when ascending and descending stairs. Furthermore, it was noted in [6], that in a study conducted by Crouter et al. [13]., pedometer devices that employed mechanisms such as, accelerometers, metal-on-metal and magnetic reed proximity switches all failed to measure the accuracy of the pedometers when walking at a slow pace or while ascending and descending stairs.

In the study [11], it was discovered that using gyroscope data for pattern recognition yielded much better results than using accelerometer data. The advantages were summarized as follows:

- Gyroscopes are more accurate and sensitive in capturing motion, compared to accelerometers. Since gyroscopes measure second-order features whereas accelerometers measure first-order features.
- Acceleration is susceptible to jitter and local minima caused my heel contact to the ground during the termination swing sub-phase of the human gait
- Angular rates oscillate around zero during walking, which makes its more ideal for step detection than accelerations.

2.6 Common Step Detection Methods

The common step detection algorithms are listed below:

- Peak Detection [14]
- Zero-crossing [6]
- Thresholding [15]

2.6.1 Peak Detection

Peak detection relies on the vertical deviation of the user. As the user walks a small change in vertical height will occur during every step. Peak detection relies on the number of total peaks, which is then interpreted as steps. In study [14] used peak detection on accelerometer data. The study processed a window of samples at a time and in that window the highest peak was interpreted as a step. This window was to reduce the number of false peaks cause by high frequency noise.

2.6.2 Zero-crossing

Zero-crossing relies on the number of times the signal crosses the zero line. It was determined by Kang et al. [11] that angular velocity signals are better than accelerometer signals for this technique, as the angular velocity oscillates around zero.

2.6.3 Thresholding

Thresholding is a technique that supplements both peak-detection and zero-crossing. For peak detection a high threshold can be set to remove detection of false peaks caused by vibrations during walking. A peak detection algorithm with adaptive thresholding is seen in study [15]. Accuracies for peak detection with thresholding in study [15] showed 10% better results than just a standalone peak detection algorithm in study [14]. Jayalath et al. [6] used thresholding with zero-crossing to validate the step detection. The threshold aids in removing small instantaneous movements.

2.7 Final Thoughts

The studies conclude that both, gyroscope based and accelerometer-based pedometers have their own use cases. For physical activities such as jogging and running, accelerometer pedometers show sufficiently high accuracy, whereas for health care use cases such as indoor navigation for the visually impaired gyroscope-based pedometers outperform accelerometer based by a remarkable margin, due to slower walking speeds. However, for the purpose of this investigation, an algorithm needs to be developed for a 'loose' or unconstrained device. The author of this paper, with the advice from the supervisor decided that the test cases for step detection will be from 'loose' pants pockets and from individuals placing the smartphone a handbag.

On account of these testing conditions, a dual based pedometer was investigated to improve accuracy by switching between accelerometery data and gyroscopic data based on the test condition

Chapter 3

Application Development

This section is aimed to gather human motion tracking information with the aid of IMU sensors onboard an Android smartphone.

3.1 Hardware

Most modern-day smartphones include an onboard IMU. IMUs are specific types of inertial sensors that measure acceleration and angular rate via an accelerometer and gyroscope respectively. High-end Android smartphones include other sensors such as gravity, pressure, proximity, etc., as well as other software sensors that result from sensor fusion. In this investigation, the focus will solely be on using the accelerometer and gyroscope. This is to allow the algorithm to be accessible to a wider range of smartphones as well as allow for implementation using a dedicated sensor for future works. An android application was created to read and store the IMU data in a Comma Separated Value (.csv) file so that it can be later analysed in MATLAB.

The specifications of the smartphone used to record and capture IMU data:

- Model: Samsung Galaxy Note 9
- CPU: Exynos 9810 SoC
- Dimensions: 161.90 x 76.40 x 8.80 (mm)
- Weight: 201g

The specifications of the laptop used to analyse and develop the pedometer algorithm in MATLAB:

- Model: Asus
- CPU: Intel® Core™ i5-8250U @1.60GHz
- GPU: Nvidia GeForce GTX 1050 4GB
- Memory: 16 GB
- Storage: 500 GB SSD

3.2 Co-ordinate Frame

Before the data was analysed, it was vital to gain a clear understanding of the coordinate system used by Android smartphones. Co-ordinate frames are important as they provide an origin point to which different measurement readings are be related to and therefore related to each other, in layman's terms it provides a clear understanding of the outputs of each sensor and how to interpret this information for different applications.

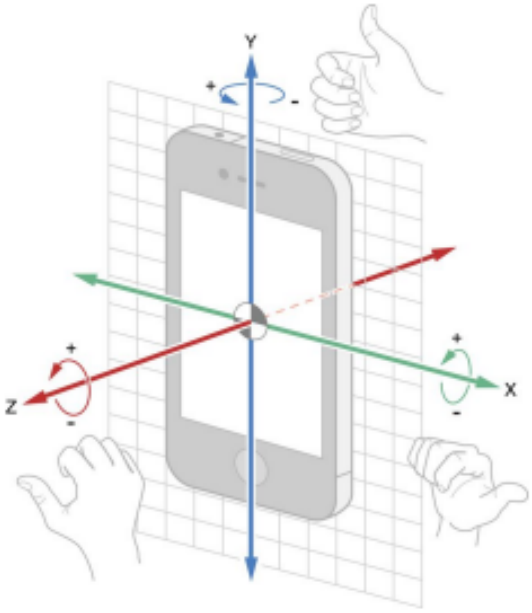


Figure 3.1: Co-ordinate frame of an Android smartphone [16]

Android Studio sensor framework uses a 3-axis body frame to describe sensor values. In the case of the Samsung Galaxy Note 9, the reference axis coincides with the centre of mass of the smartphone. This is illustrated in Fig. 3.1

The following descriptions of the reference axis were when the device is orientated upright facing the user:

- The X-axis is horizontal, along the centre of mass with the positive direction pointing to the right of the screen.
- The Y-axis is vertical, along the centre of mass with the positive direction pointing upwards.
- The Z-axis is perpendicular to the screen, along the centre of mass with the positive direction pointing towards the user.

An important point to note was the coordinate system remains the same regardless of the orientation of the device or the screen i.e., the coordinate system never changes.

For this research project the collection of data from the following sensors was used to create the pedometer algorithm:

- Accelerometer: This sensor measures the acceleration along the three axes, including the acceleration due to gravity, in m/s^2. The sensor abides by the coordinate system shown in Fig above
- Gyroscope: The gyroscope measures the rate of rotation in rad/s around all three axes. The sensor's coordinate system is the same as the one used for the acceleration sensor. The rotation is positive in the counter-clockwise direction, the polarity of the data follows the right-hand rule for rotary vectors, where the thumb points in the direction of the axis and the inward curl of the fingers determine the direction of rotation.

3.3 Collection of IMU Data

The android application was developed using the Android Studio development tool. On account of using MATLAB to do the processing and analysis of the data, only a bare-bones design was needed for the android application.

From Fig. 3.2 , it is seen that the basic user interface (UI) only displays the value of each sensor with the functionality to record and store the sensor data into a CSV file. The application allows the user to enter a file name, so the user can easily keep track of the recorded files. Upon pressing the START button SensorCSV will create a CSV file into the application root directory folder with the recorded file name and log in a sample of data every time the sensor event class is called, i.e., whenever there

is a change in the reading of a sensor. Upon pressing the STOP button the writing to the file will end and the CSV will be saved.

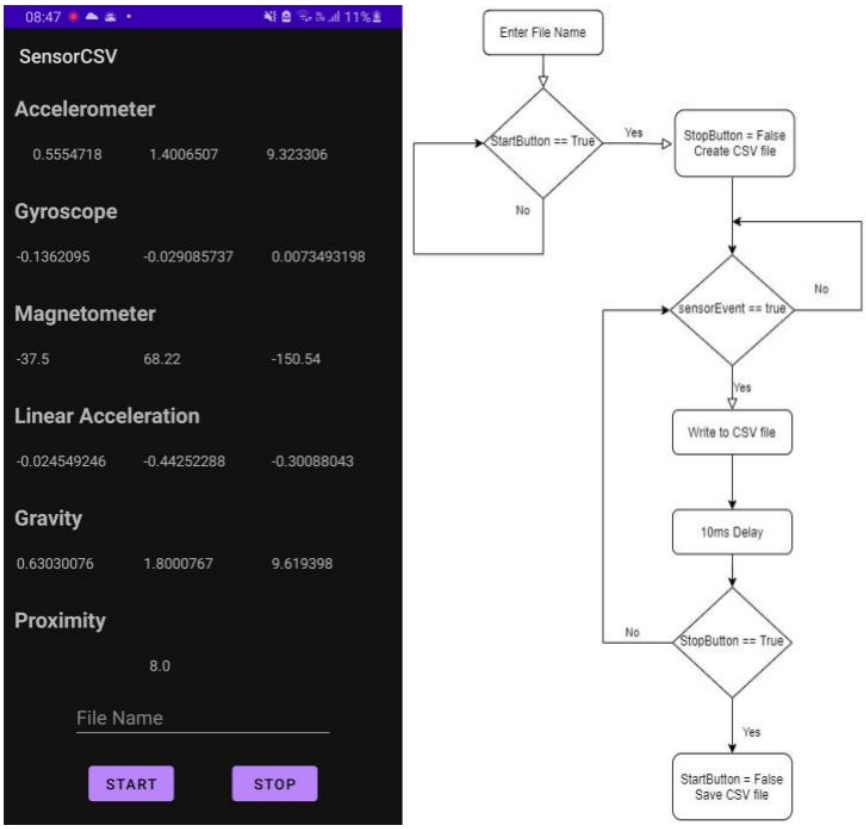


Figure 3.2: GUI and flow chart of the SensorCSV application

A challenge faced when creating the data collection application was using Android Sensor Manager, which is an interface Android Studio uses to record sensor data. This interface is event-driven, meaning that a sensor reading is only detected when the value changes, thus a conversion from event-driven to time-driven architecture was implemented. This was accomplished by adding a timer with a delay of 10ms between each reading to ensure a consistent sampling rate, as illustrated in Fig. 3.2.

The minimum frequency is calculated using the Nyquist Shannon sampling theorem, as illustrated by Equation 3.1, to ensure we do not lose important information during the sampling process.

$$F_{sample} \geqslant 2 * F_{max} \tag{3.1}$$

According to Pachi et al. [17] the average walking frequency varies from 1.4 Hz – 3 Hz, varying on the person. Using the above equation, the calculated minimum sampling frequency or sampling period of the data collection application is 6Hz or 0.166 seconds respectively.

The smartphone used in this investigation can sample the data at a rate of 200 Hz but due to the limited computational resources available on a mobile phone, the fastest sampling period selected was 0.01 seconds or 100 Hz. This was selected so that information from different gait sub-phases can be extracted even for fast walking frequencies.

Chapter 4

Data Analysis

This section aimed at analysing the collected sensor information to discern possible patterns to gather insight into how the sensor values change based on the different phases of the human gait.

4.1 Data Analysis for Gyroscope Data

A deep dive analysis was done on the IMU sensor readings for a loosely attached device. The analysis will be conducted for both accelerometry and gyroscopic data

4.1.1 Pants Pocket

Jayalath et al. [6] reported that most individuals place their smartphone in the pocket vertically, thus allowing for a single axis to be used to track the motion of the leg. The assumption is the smartphone's z-axis is parallel to the direction of motion, therefore the x-axis can be used for step detection as it is undergoing the most rotation, as illustrated in Fig. 4.1

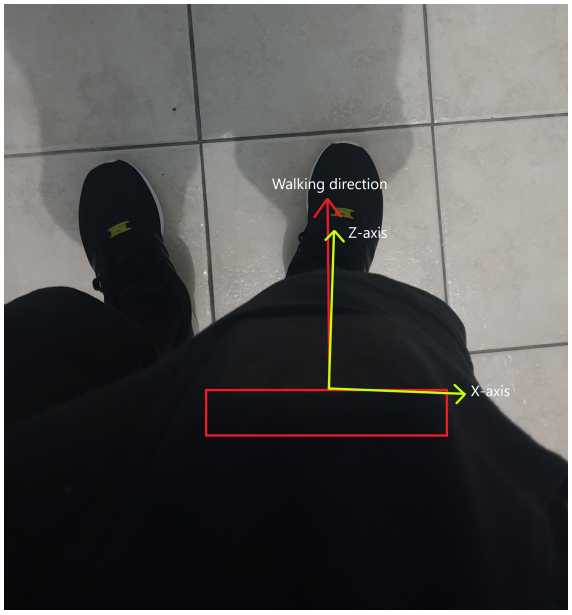


Figure 4.1: Aerial view of walking with the device placed vertically in the pocket. X-axis is perpendicular to the walking direction

Fig. 4.1 shows the ideal placement for the algorithm in [6] to be the most effective. The x-axis can be seen to be perpendicular to the walking direction, thus when the user walks most of the angular rates contribution will come from the x-axis since there is little rotation recorded across the y and z-axes.

This concept works well for constrained devices, however, when placed in a loose pants pocket, the device orientation may change during motion. This results in reduced rotation on the x-axis, potentially lowering the accuracy of the algorithm. This change in orientation can consist of translational and/or tilt movement.

Translation movement is defined as a shift in the position of the device along the thigh while remaining vertical. Tilt movement is defined as the rotation of the device along the z-axis of the smartphone.

During translational movement, as seen in Fig. 4.2, the device x-axis becomes less and less perpendicular to the direction of walking, while simultaneously becoming

Figure 4.2: Aerial view of walking with the device placed vertically in the pocket at a translation of 45°

more and more parallel with the direction of walking. This will result in less contribution to the angular rates as there are now smaller rotations around the x-axis. Furthermore, it can be noted that while the rotations of the x-axis decrease, at the same time the rotation of the z-axis will increase. This is the result of both axes being perpendicular to each other. This relationship can be described by Equations 4.1 and 4.2.

$$X_{angular-rate} = \omega \times cos(\theta) \tag{4.1}$$

$$Z_{angular-rate} = \omega \times sin(\theta) \tag{4.2}$$

Fig. 4.2 shows a use case where, as the user walks the phone is rotated around the y-axis, from 0° to 90°. By examining the Fig and Equations 4.1 and 4.2, key angles were discovered and tabulated below 4.1

Table 4.1: Varying strengths of single axis contribution based on angle θ

θ	**X-axis contribution**	**Z-axis contribution**
0°	100%	0%
30°	86.8%	50%
45°	86.8%	86.8%
60°	50%	86.8%
90°	0%	100%

Fig. 4.3 illustrates the x and z-axis gyroscope data for the device placed at an angle of 0°. It can be noted that the data follows the relationship discussed earlier. At approximately 0° the magnitude of rotation on the x-axis should be far greater than the z-axis because the x-axis is perpendicular to the direction of walking. Furthermore, it demonstrates that the z-axis, which is approximately parallel to the direction of walking has very little rotational motion.

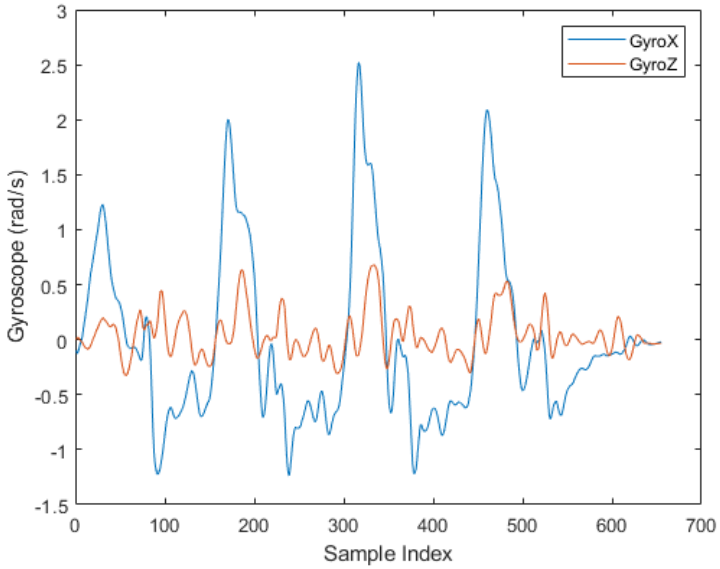


Figure 4.3: Raw gyroscope reading for device translation of 0°

From Fig. 4.3, it is clearly shown that the x-axis signal has a periodic nature with

a peak value of 2 rad/s and a valley value of -1.2 rad/s. In comparison, the z-axis has no noticeable periodic nature and very weak rotational measurements between 0.5 rad/s and -1.2 rad/s. In this case, the algorithm proposed in [6] worked perfectly.

Fig. 4.4 illustrates the x and z-axis gyroscope data for the device placed at an angle of 90°, as previously discussed both readings should be very different from one another. The results should be the opposite of the results presented for when the angle was approximately 0°.

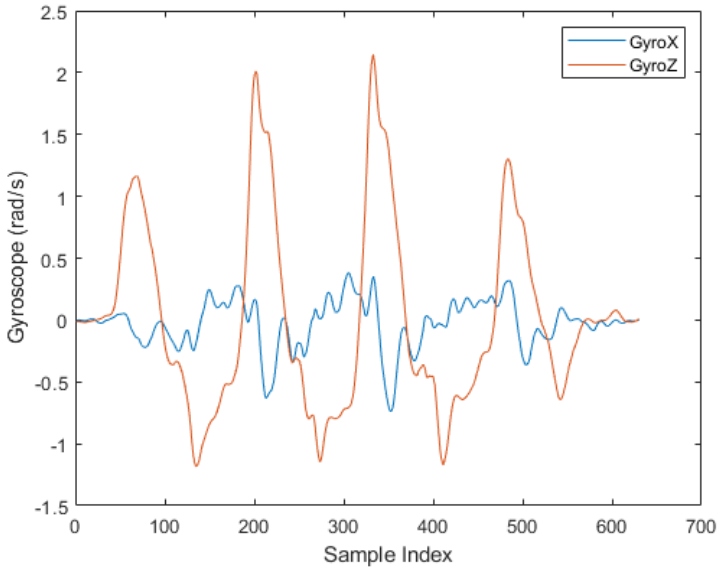


Figure 4.4: Raw gyroscope reading for device translation of 90°

The Fig. 4.4, clearly shows that the z-axis is the signal with the highest signal strength and periodic nature. This result was expected as it aligns with the data presented in Table 4.1. Under these conditions, the algorithm proposed in [6] would not accurately work due to the weak, non-periodic x-axis signal.

Fig. 4.5 illustrates the x and z-axis gyroscope data for the device placed at an angle of 45°. Following Table 4.1 the figure shows very similar readings for both the x and z-axis. According to cosine rules, each signal should represent 70.7% of the

total measured angular rate. Furthermore, it demonstrates that both signals track each other and each had sufficient strength and period to accurately work with the algorithm in [6].

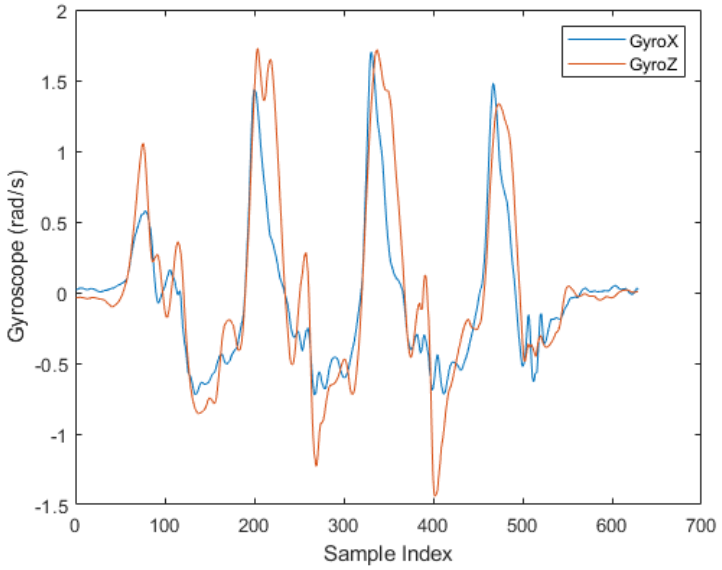


Figure 4.5: Raw gyroscope reading for device translation of 45°

It was also discovered that the polarity of the signals change depending on the orientation of the device. This is attributed to the gyroscope reading values as positive for counter-clockwise rotation and negative for clockwise rotation. This is shown when the device is rotated 180° along the y-axis or if placed in the left pocket in the same manner as shown in the tests above.

Fig. 4.6 shows the orientation of the phone in the left side pocket. Using the right-hand rule the direction of rotation of both axes were determined and compared to Fig. 4.5. In Fig. 4.5 both the x and z-axis rotate counterclockwise as the thigh rotates, therefore both axes track each other. In fig, the x-axis still rotates in a counter-clockwise direction whereas, the z-axis now rotates in a clockwise direction when it is tracking thigh movement. This results in the axes being inverse of each

other.

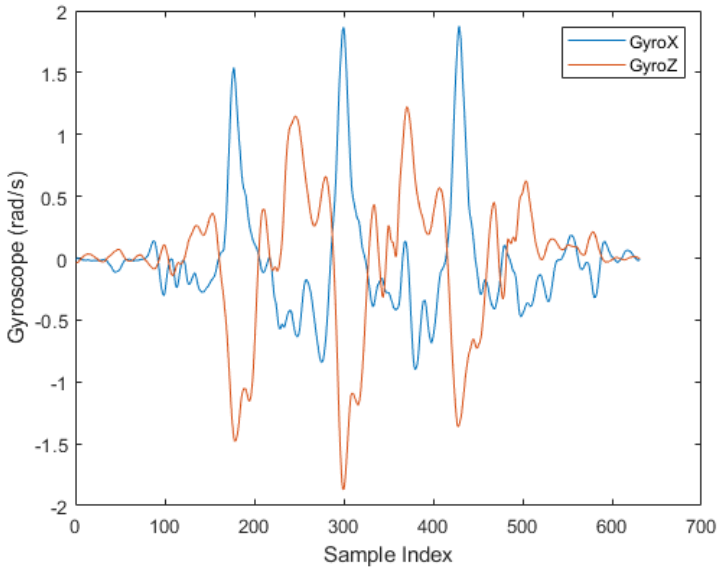


Figure 4.6: Raw gyroscope reading for device translation of 45°in the left pocket

In loose pockets, however, there was also tilt movement. Tilt movement affects the gyroscope readings in the same manner as a translational movement. The only difference is that translation movement was based around the rotation of the y-axis, therefore the x and z-axes were studied, whereas tilt movement was based around the rotation of the z-axis. By rotating the device around the z-axis the x-axis signal strength weakens as it becomes more in line with the gravity vector, thus not getting much rotation. The reverse is also true, as the x-axis rotates more in line with the gravity vector, the y-axis rotates become more perpendicular to the axis of the leg, contributing more to the angular rotation of the thigh.

It must be noted that not all configurations of orientations were discussed but the general concept was demonstrated to gain a better understanding of how the characteristic of gyroscope data changes with different movements and device orientations.

4.1.2 Handbag

Having the smartphone placed in a handbag records different signals as compared to the pocket above. Fig. 4.7 is the recorded gyroscope x, y, z-axes data for a 6-step walking motion, whilst the phone was placed arbitrarily in a handbag.

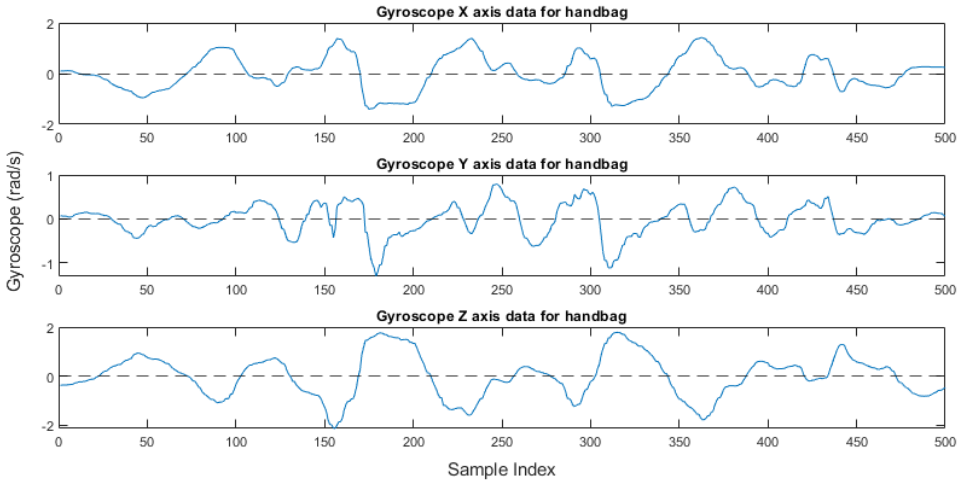


Figure 4.7: Raw gyroscope x,y and z reading for device in a handbag

Fig. 4.7 clearly shows that there is little periodic patterns in either the step time or the magnitude, to be used for step detection. The main reason for this is that the device is no longer appended to a body part, therefore this makes tracking walking motion unreliable. More specifically, the figure shows that there is no detectable periodic zero-velocity region in the stance phase of the human gait due to the upper body being in continuous motion. Generally, any unattached device placement above the waist has little to no periodic motion, unlike the pocket which tracks the thigh movement.

Fig. 4.8 below illustrates the accelerometer magnitude for the same dataset used in Fig. 4.7. The accelerometry data performs remarkably better than gyroscope data in finding patterns that are used for step detection. In Fig. 4.8 we can correlate

the number of peaks to the steps taken, whereas in the gyroscope data there are no discernible patterns.

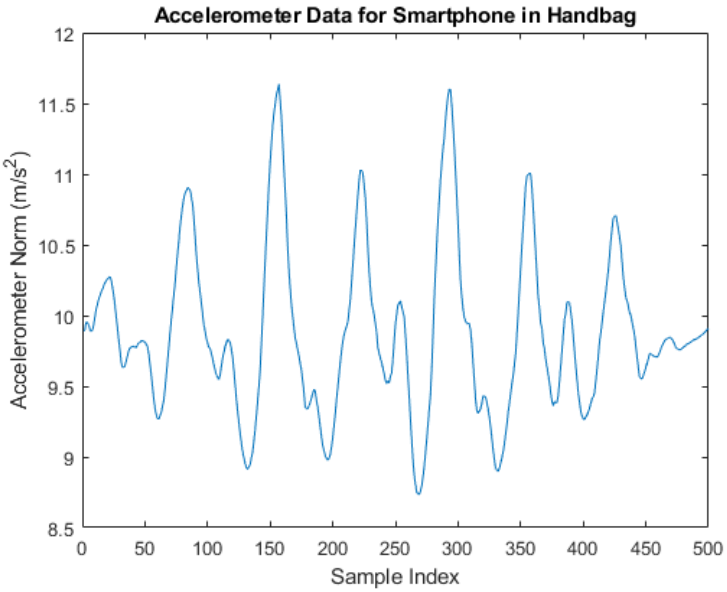


Figure 4.8: Accelerometer magnitude data for device in a handbag

4.2 Data Analysis for Accelerometer Data

Accelerometer data, unlike gyroscope data, can track the bounce of the user's movement when walking. When a user walks, they produce a slight bounce in the vertical direction in each step. This is illustrated by Fig. 4.9, by observing a person's head as they walk, their whole upper body moves in a smooth bounce like motion [18]. This pattern can be exploited to count the steps at each bounce or 'peak', as illustrated in Fig. 4.9

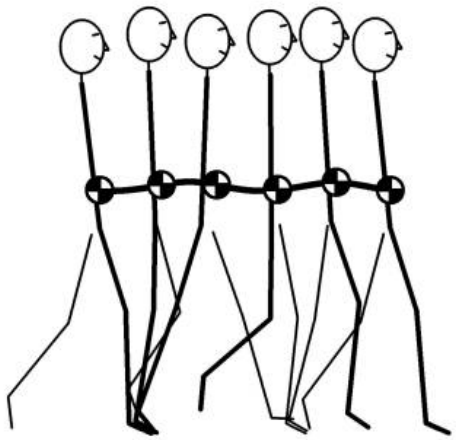


Figure 4.9: Human walking motion demonstrating bounce during each step [18]

Fig 4.10 shows the acceleration magnitude from the 3-axis accelerometer. Since the algorithm needs to operate in any orientation, the total magnitude is considered rather than analysing a single axis component. Equation 5.3 was used to calculate the acceleration magnitude.

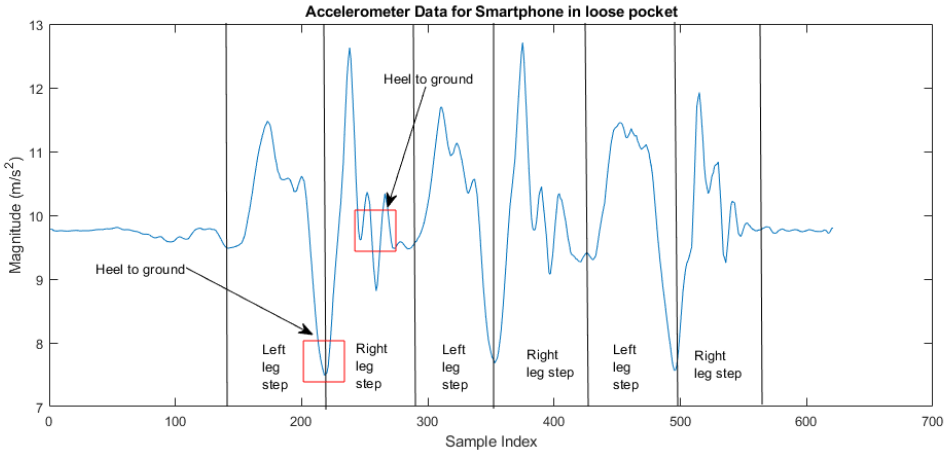


Figure 4.10: Accelerometer magnitude tracking 6 steps for a pocket use case

The walking motion for the dataset in Fig. 4.10 can be described as follows:

- The reference leg is the right leg; therefore, the device was tracking the stance and swing phases of the *right leg*.
- The test starts with the user first taking a left step. It can be noted that the valley at the end of the *left leg* step indicates the end heel contact of the left foot on the ground. Furthermore, the *left leg* step is the tracking of the stance phase of the reference leg.
- The subsequent right leg step is the swing phase of the reference leg. It was noted that the magnitude was higher due to the swing phase producing a higher acceleration as it tracks the same distance over a shorter period. Similarly, as the *left leg* step, the valley indicates the end of heel contact of the right foot on the ground. However, it can be seen that the accelerometer suffers from jitters, local minima and maxima at this point in the human gait cycle, as described in [11] This is attributed to the vibrations caused after the foot strikes the ground.

From the above analysis, it is determined that accelerometer data can capture the steps by looking at the peaks of the acceleration magnitude. The disadvantage is that accelerometers are susceptible to high-frequency oscillations that occur during the preswing and terminal swing gait phases, i.e., heel to ground contact. It is due to these high-frequency oscillations that accelerometers do not operate well under slow-walking conditions as the height of the peaks formed from walking and the height of the peaks formed by oscillations are very similar in magnitude. This makes it difficult to determine which are *true peaks* (peaks generated by a step) or *false peaks* (peaks generated by noise).

4.3 Data Analysis for Sensor Fusion

From the previous analysis, it was determined that gyroscope-based pedometers are more accurate for use cases when there is sufficient periodic motion, i.e., when the device is appended to a body part. Further analysis was done to determine any relationships between accelerometer and gyroscope data.

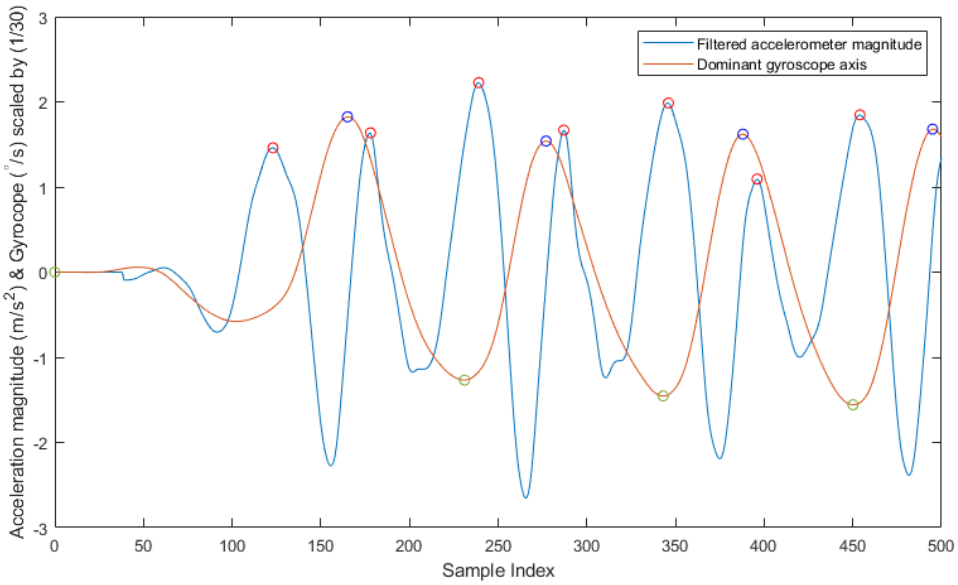


Figure 4.11: Pattern matching between accelerometer magnitude and dominant gyroscope axis

Fig. 4.11 shows that the dominant angular velocity signal tracks the accelerometer magnitude signal but is half its frequency. The acceleration magnitude represents the full walking motion, hence capturing the patterns for both the left and the right legs. The gyroscope however only tracks the motion of the leg it is appended to, thus a full stride equates to two steps, thus half the frequency. Fig 4.11 displays that between every peak-valley-peak in the acceleration magnitude there should exist a peak or valley in the gyroscope signal. This property coincides with the gyroscope

signal being half the frequency of the accelerometer magnitude signal.

Chapter 5

Algorithm Design

This section is aimed to implement a dual-sensor pedometer. The developed algorithm makes use of accelerometry data as well as gyroscopic data to classify the use case of the smartphone, switching between zero-crossing detection and peak detection. The process is broken down into multiple stages, as illustrated in Fig. 5.1.

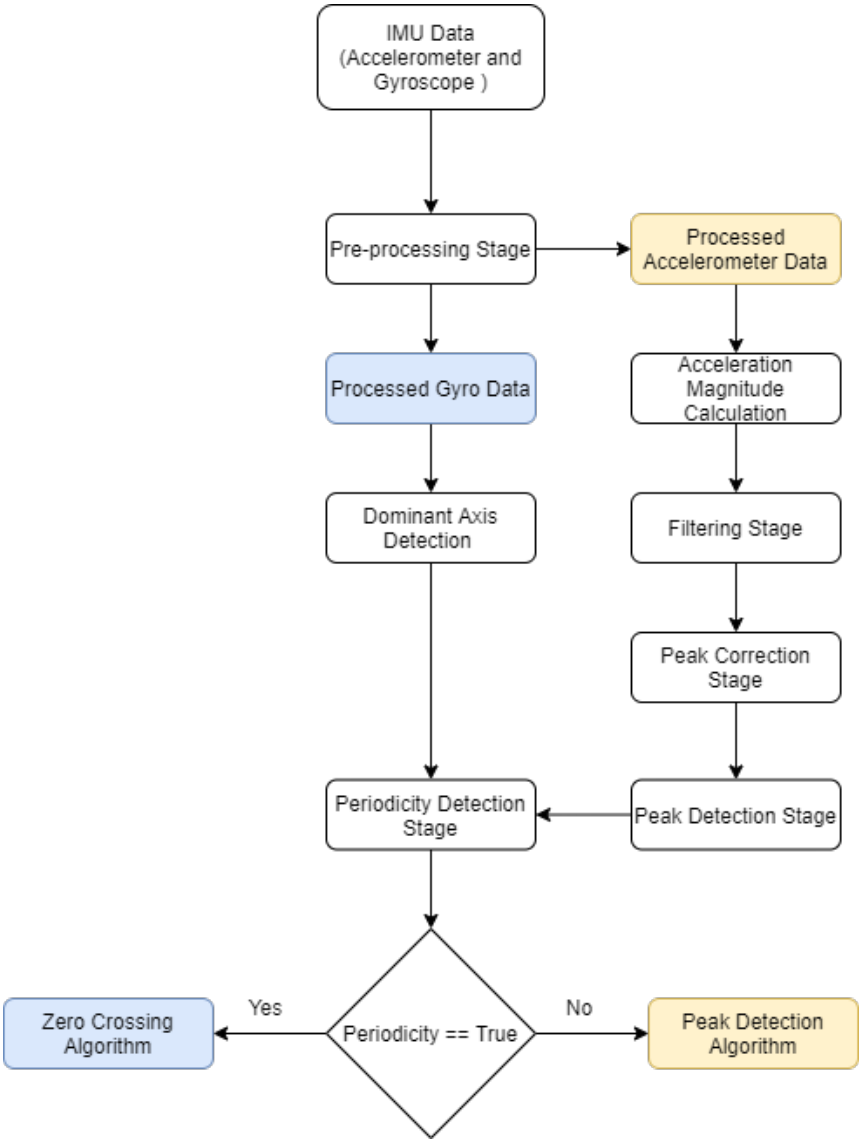


Figure 5.1: Flow chart of the process of the proposed algorithm

5.1 Pre-processing IMU data

In this investigation, the analysis and algorithm must be performed at the period where the action of walking is taking place and ignoring the data that is recorded when the phone is being placed or removed from a pocket or bag.

An attempt was made to automate this process using the proximity sensor to detect when the phone was in the pocket or bag. Although this method worked, it seldom recorded data the mimicked a step depending on how the user removed the phone from the pocket or bag.

Therefore, this process had to be done manually. To ensure that the correct data was segmented, the user had to wait five seconds before and after walking, as seen in Fig. 5.2 This provided a visual representation of the walking motion, in which the upper and lower limits were found. To make this tedious process efficient all data sets were stored in structural arrays which made executing a large number of data sets computationally effective.

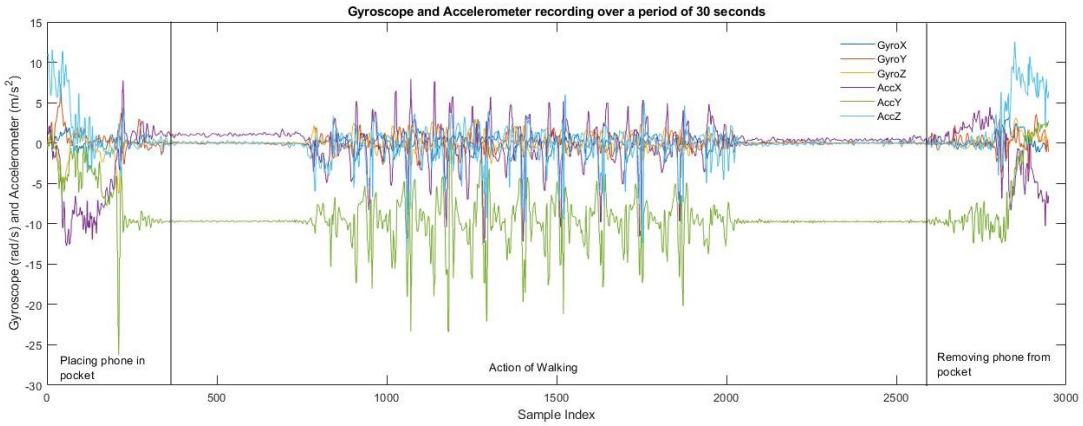


Figure 5.2: Plot of raw gyroscope and accelerometer readings

A limitation in using smartphones for data recording is that Android Studio uses event-driven architecture to record sensor data. This causes an inconsistent

sampling rate because the sensors are not accurately synchronized as a result of this event-driven architecture.

Although the sampling rate was set to $10ms$ in the data acquisition application there is no guarantee that the sampling frequency will be 100Hz. The Fig above shows that multiple sensors are read synchronously, logged into the CSV file and then delayed $10ms$ for each sample to establish a consistent sampling rate. On the account that multiple sensors are read at a particular timestamp, the sensors lag a bit resulting in time variances between samples. This can be further seen by looking at the time in between samples as illustrated in Fig 5.3. The desired sampling time was set to $10ms$ in the android studio source code but sampling times varying between 8 and $12ms$ are being recorded.

Timestamps class from the Android studio library output the current system time in nanoseconds, Therefore, the following equation 5.1 to calculate the time in between samples:

$$ti_{sample} = \frac{t[i] - t[i-1]}{10^6} \tag{5.1}$$

where ti_{sample} is the sample time for the i^{th} sample, t[i] is the timestamp for the i^{th} sample, t[i-1] is the timestamp for the previous sample and 10^6 is the scaling factor to convert the timestamp from nanoseconds to milliseconds.

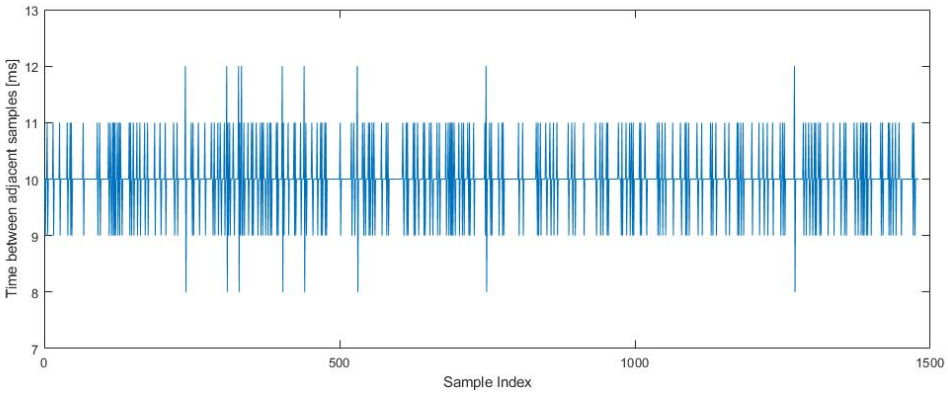


Figure 5.3: Plot of sample rate

Although the time variances in between some samples are minor it is still important to correct as it will ensure a constants sampling rate which is needed for the subsequent filtering process. This is achieved by linearly interpolating subsequent samples at a fixed interval of $10ms$ using the following Equation 5.2:

$$Y[i] = \left(\frac{y[i] - y[i-1]}{t[i] - t[i-1]} \right) t_{ds} + y[i] \tag{5.2}$$

where (y[i],t[i]) and (y[i-1],t[i-1]) are two adjacent points in the data structure array and t_{ds} is the desired sampling rate.

5.2 Accelerometer Algorithm

5.2.1 Accelerometer Magnitude

For the developed algorithm to operate independent of orientation, contributions from all three axes must be considered. This was accomplished by calculating the magnitudes across all three axes, as shown in Equation 5.3

$$acc_{mag} = \sqrt{x^2 + y^2 + z^2} \tag{5.3}$$

Where acc_{mag} is the magnitude of the signal, x is the acceleration in the x-direction, y is the acceleration in the y-direction and z is the acceleration in the z-direction.

An important point to note is the accelerometer readings are constantly under the influence of gravity, so total acceleration is the sum of gravitational acceleration and user acceleration. In Fig. 5.4 the device was placed in an arbitrary position in a handbag. Even though the device is stationary, the accelerometer reads have non-zero magnitude. These signals are the result of the gravitational acceleration split between the three axes.

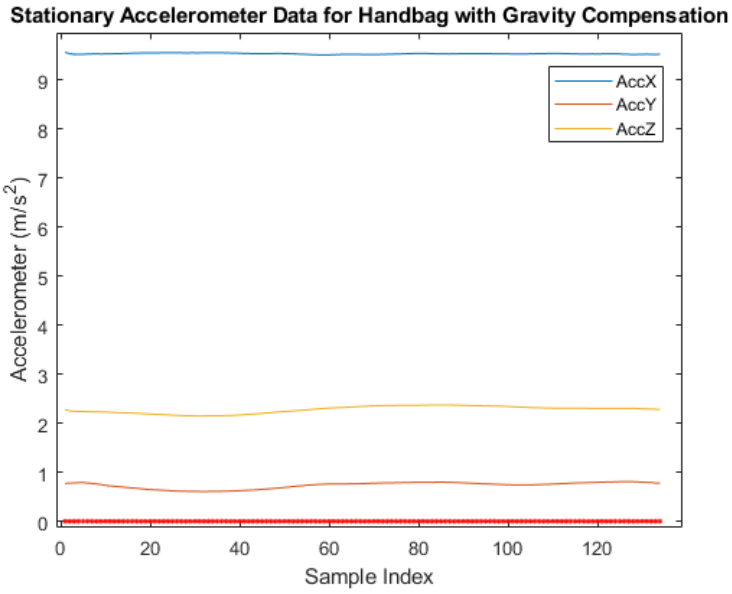


Figure 5.4: Effects of gravity on the total acceleration

A common method to remove the gravity contribution from the total acceleration is to simply subtract the mean value of the axes from the resultant. This is illustrated

by the red line in Fig. 5.4. In the handbag the total acceleration is zero since the phone is stationary.

5.2.2 Filtering

Before the accelerometer norm was used for step detection, high-frequency artefacts in the accelerometry data needed to be removed utilizing a lowpass filter. Especially for accelerometer data recorded through android services. High-frequency oscillations caused by vibrations after the foot strikes the ground corrupt the signal of the user's motion, causing false peaks to arise.

Spectral analysis of the data was done to find the approximate walking frequency. Most studies, such as the one done by Kang et al. [11], performed an FFT analysis to determine the walking frequency for a given dataset. Although this yielded good results the authors did not stipulate if the data used were integer-periodic signals.

The fundamental requirement to perform an FFT on a signal is, the signal needs to be periodic, i.e., it needs to start and end and the same point. Spectral leakage occurs when the signal being measured is non-integer periodic for that sample set. This causes 'smearing' of power across the frequency spectrum.

The spectral leakage was compensated for by applying a Hann window to the accelerometer magnitude signal. This window function shapes the original signal so that it touches zero at both ends, removing discontinuity.

The single-sides spectrum of the signal was calculated by Fast Fourier Transform (FFT) the windowed acceleration magnitude. In Fig. 5.5, the spectrum displays spikes from 1.5 – 3 Hz; which represent the frequencies that are continuously repeated during walking. This spectrum is dependant on the walking speed of the individual; thus at faster walking speeds, higher magnitude spikes will appear at higher frequencies. From Fig. 5.5 a cut-off frequency of 3Hz was selected to compensate for both slower and fast walking speeds. This walking frequency coincides

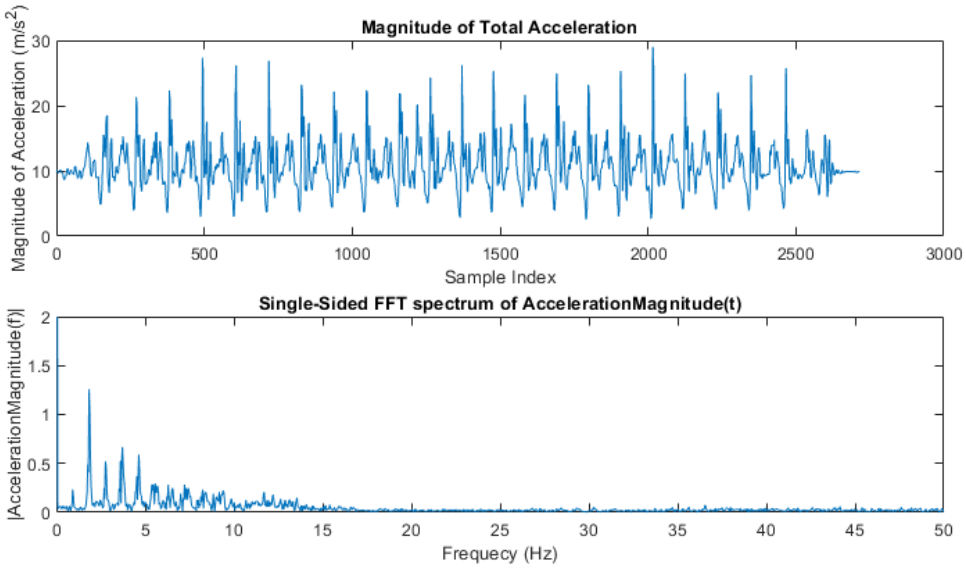


Figure 5.5: FFT of accelerometer magnitude data

with the one stated in [17].

Many studies such as [19], use a simple moving average filter to process the accelerometer data due to its low computational cost. This filter functions averaging a pre-set number of points from the input to generate each point in the output signal. Equation 5.4 describes the moving average function.

$$y[i] = \left(\frac{1}{N}\right) \sum_{m=0}^{N-1} x[i+m] \tag{5.4}$$

Where y[i] is the output for the ith sample, x[i+m] is the input signal and N is the number of samples to average.

Factors that needed to be considered before designing the low pass filter were the passband ripple, the cut-off frequency, the transition-band width and the stopband attenuation. The cut off frequency determines the frequency to which no attenuation is required on the signal. The length of the filter determines the width of the transition band, which also determines the order of the filter. The higher the order

or larger the length of the filter the steeper the transition band. Having a too large order though will increase computation time as well as group delay time, making real-time processing impractical to achieve.

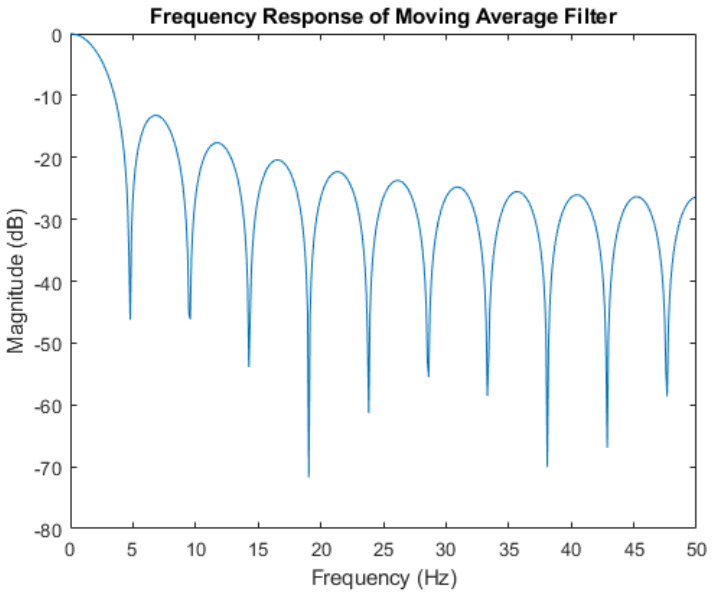


Figure 5.6: Frequency response of a moving average filter with a length of 21

The response shown in Fig. 5.6 displays a correct cut-off frequency and transition band but lacks suppression in the stopband. To increase the attenuation in the stopband double suppression is applied to the accelerometer data by first applying the filter to the individual axes and again on the accelerometer norm.

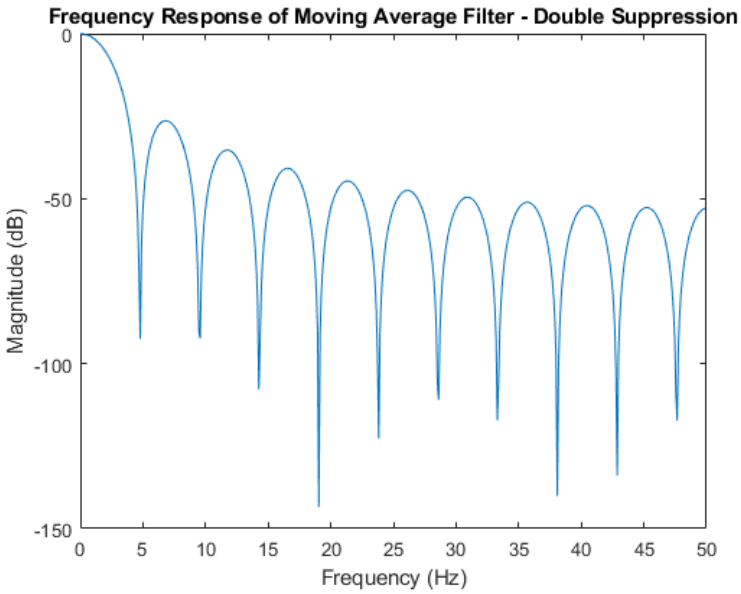


Figure 5.7: Frequency response of applying the filter of Fig 5.6 twice.

Fig. 5.7 illustrates the frequency response of applying double suppression. The response now shows sufficient attenuation in the stopband.

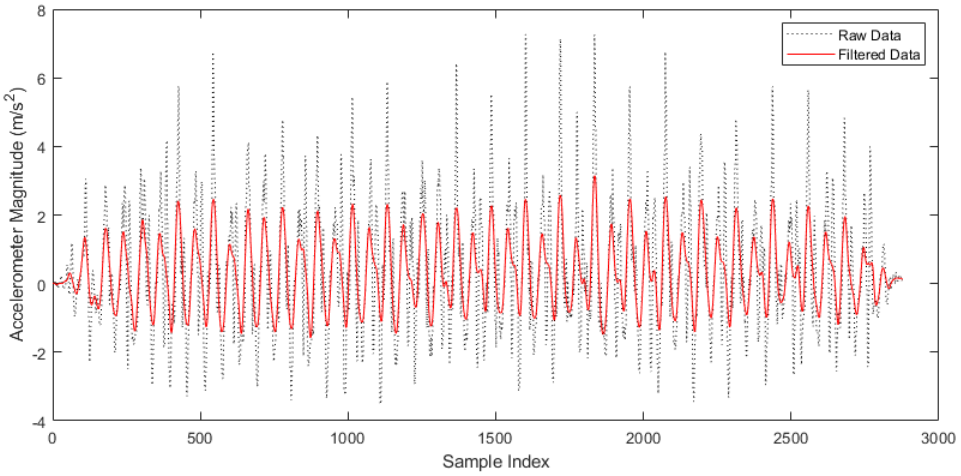


Figure 5.8: Implementation of moving average filter

Fig.5.8 illustrates the implementation of the moving average filter by overlaying the filtered accelerometer magnitude over the raw accelerometer magnitude data. The filter did a sufficient job in removing high-frequency noise, accentuating the peaks.

5.2.3 Peak Enhancement

The moving average filter does well in removing the false peaks that are caused by vibrations experienced by the device, as illustrated in Fig. 5.8. However, the device experienced these oscillations more frequently when placed in the handbag. These oscillations generate false peaks which caused the algorithm to detect an incorrect step. A false peak is a local peak that is not large enough when looking at the global context of the dataset but large enough for the algorithm to regard it as a step. An example of this is illustrated in Fig. 5.9

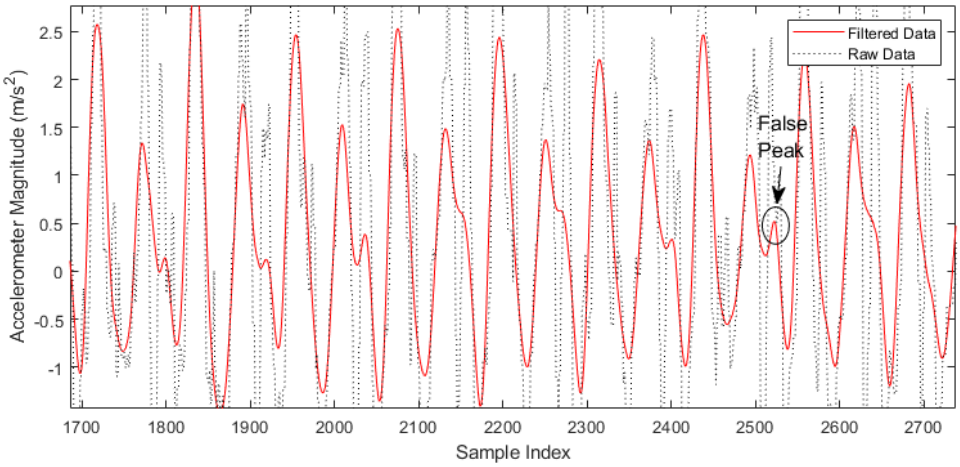


Figure 5.9: False peak caused by high frequency noise

A study conducted by Palshikar [20] identified that finding the true peaks in a given time-series dataset is important in many applications, such as price/volume

charts, traffic data and ECG signals. One of the methods proposed was to enhance and reduce the peak magnitude of true and false peaks respectively. This was achieved by applying a mean difference of N number of samples on either side of the data point in question. Equation 5.5 describes this method:

$$x[i] = \frac{\sum_{m=-N,m\neq i}^{N}(x[i] - x[i+m])}{2N} \tag{5.5}$$

Where x[i] is data at the ith sample, N is half the window length and m describes the sample left or right of the point under consideration. The window size was set to the length of the swing phase lobe to maximize the true peak magnitude.

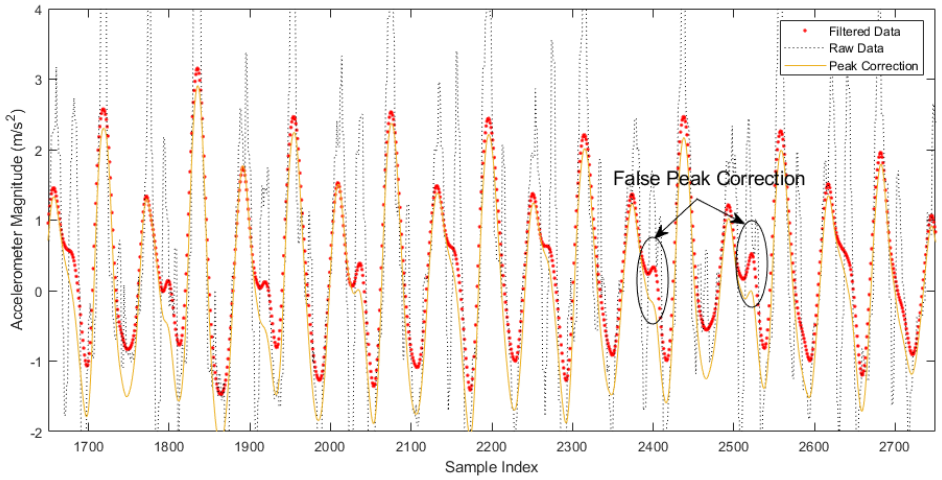


Figure 5.10: Implementation of Equation 5.5 on filtered accelerometer data

Fig. 5.10 illustrates the implementation of Equation 5.5. Although it does not enhance the true peaks as mentioned in [20] it did help with reducing the magnitudes of the false peaks.

5.2.4 Peak Detection

Step detection occurs when the algorithm detects a peak above a certain threshold. This threshold is relatively small and is used to account for oscillations caused by the vibrations experienced by the device.

A timeout function is used to ensure that peaks that occur in quick succession are not considered as a step. The peak correction helps in reducing these oscillations but there are seldom cases where these oscillations occur, especially when the device is placed in a handbag.

Once the timeout function detects a peak, the peak detection algorithm is disabled for 200ms. This was sufficient time as these quick succession peaks form at initial contact or end of the terminal swing, which contributes 30% and 15% of the total gait cycle respectively. The total number of peaks detected represents the step count.

Fig. 5.11 shows the flow chart of the accelerometer algorithm.

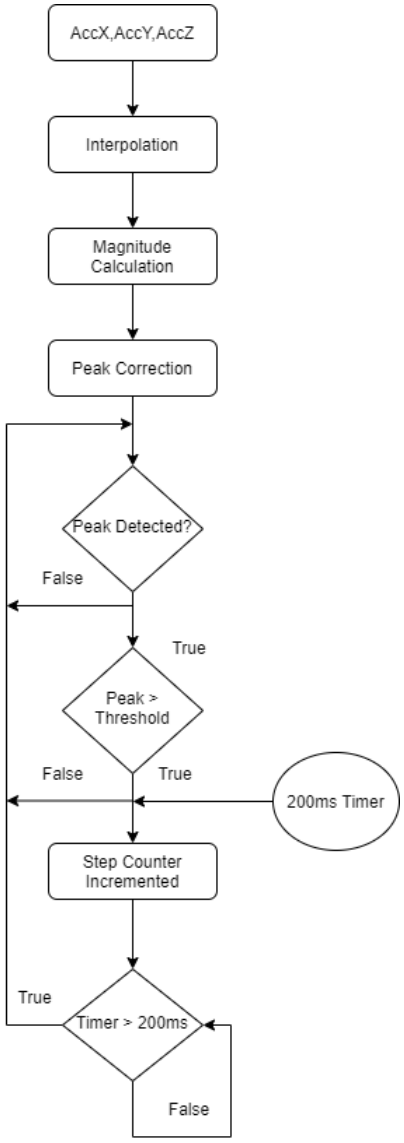


Figure 5.11: Flow chart of accelerometer algorithm

5.3 Gyroscope Algorithm

The gyroscope-based algorithm developed by Jayalath et al. [6] along with the addition of dominant axis detection was used in the development of the dual-sensor

based algorithm.

5.3.1 Dominant Axis Detection

Although the algorithm in [6] produced high accuracies, it only did so as long as there was sufficient rotation around the x-axis. To make it most robust, where it can operate under any orientation, a dominant axis detection was added.

The implementation of this algorithm was to supplement the accelerometer algorithm. This was achieved by using the gyroscope data to determine a classifier variable. The classifier determined whether to use zero-crossing or peak detection for step counting. It must be mentioned, however, that this classifier was done post-processing, i.e., the classifier only ran after the accelerometer algorithm processed all the data. This was largely due to the design of the dominant axis detector.

From the data analysis section, it was determined that any axis could be used to track the steps using the zero-crossing method if the axis had sufficient signal strength and was periodic. The calculated mean value of all 3 gyroscope signals had a value close to zero. This made sense as the gyroscope oscillates around the zero point in a periodic manner (for certain use cases). By calculating the standard deviation, the scale of the spread of gyroscope signals from the mean can be determined. The highest standard deviation equates to a high spread which equates to greater signal strength. Although, in loose pockets the y-axis would contribute to non-periodic rotation due to the shaking induced by vibrations. Furthermore, these rotations skewed the standard deviation results. This was rectified by removing the signal axis that contributed the most to gravity before the standard deviation calculation was performed.

An important point to note is that this only achieves desirable results after post-processing, i.e., when the standard deviation is calculated using all the recorded data. An attempt was made to calculate the standard deviation for each axis for

every 100 samples and update the dominant axis accordingly. This however yielded poor results due to the polarity of the different gyroscope readings. This was shown in Fig. 4.6, the orientation of the phone dictated whether the gyroscope readings would have a negative polarity or positive polarity. Fig. 5.12 shows an example of how the 100 samples standard deviation failed.

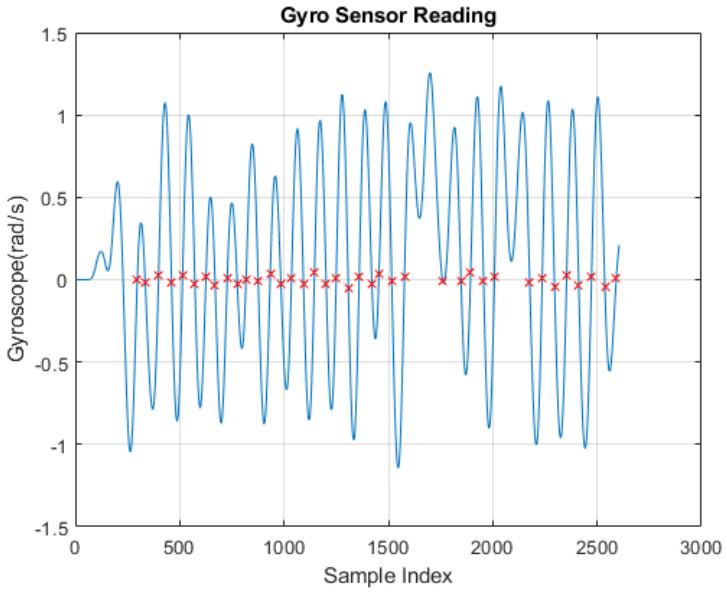


Figure 5.12: Dominant axis detection for the device placed at a 45°translational angle

For this use case, the device was placed in the left pocket at a 45-degree translational rotation, therefore both the x and z axes have similar readings but are inverses of one another, as shown in Fig. 4.6. The dominant axis detector declared the z-axis as the most dominant around 1600 samples (16 seconds) and 2100 samples (21 seconds). At these points, the gyroscope signal starts to tract the z-axis signal and due to x and z signals being inverses of one another, a new peak was formed instead of continuing into a valley.

By calculating the standard deviation at the end of the dataset ensures that the gyroscope signal used for step detection only consists of components from a single

axis, thus removing all polarity issues.

5.3.2 Periodicity Check

Fig. 4.11 illustrated that for every peak-valley-peak in the accelerometer magnitude there existed a gyroscope peak or valley. When the first instance of this condition is met, a periodicity check is done between either the corresponding gyroscope peaks or valleys depending on the signal polarity. A positive polarity will result in the peak periodicity check and a negative polarity will result in a valley periodicity check.

The classifier utilizes a time window threshold for verifying the validity of the periodicity. The classifier will invalidate the use of the gyroscope algorithm if the time between adjacent peaks/valleys were greater than the threshold.

Fig. 5.13 shows the flow chart of the gyroscope algorithm.

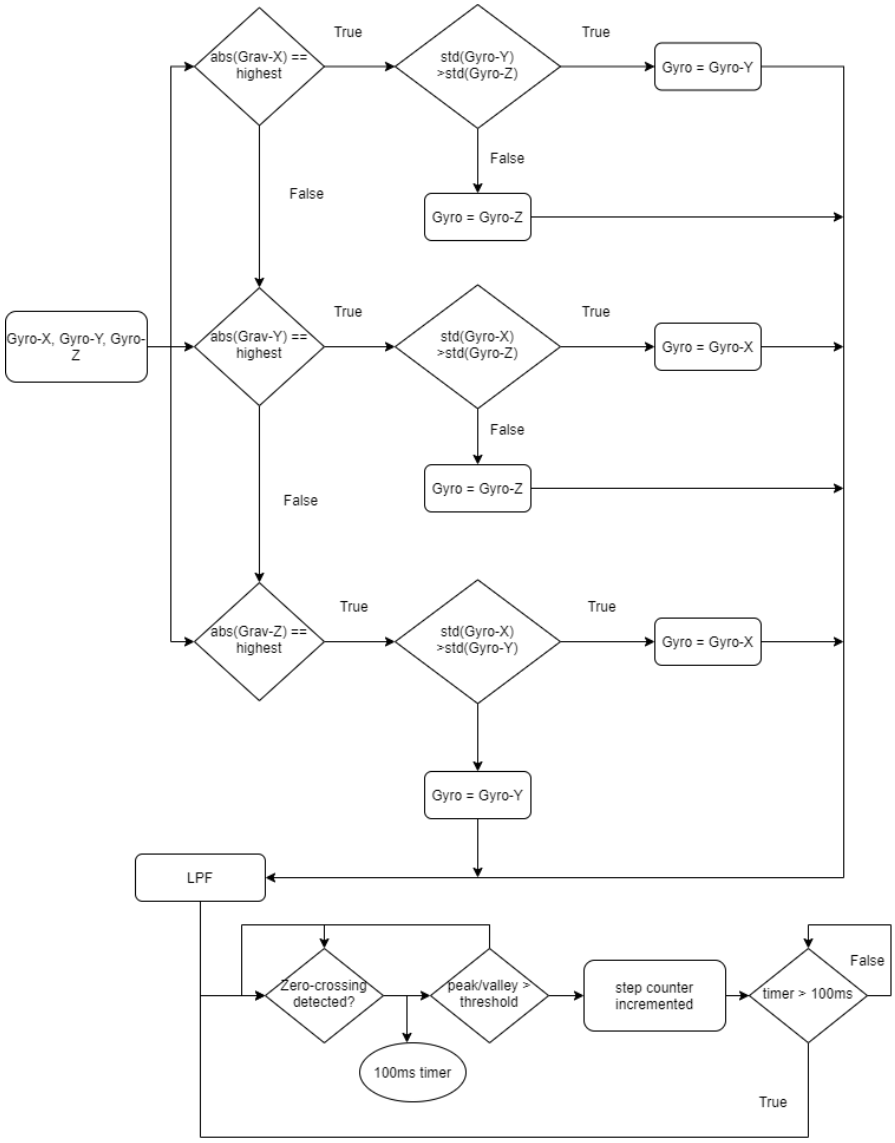


Figure 5.13: Flow chart of accelerometer algorithm

5.4 Overview of the Developed algorithm

To summarize, when the classifier variable is set to true - the gyroscope signals are assumed to be periodic (device in pocket) and the zero-crossing step detection is used for step counting. When the classifier is set to false - the signal is determined to have little or no periodic gyroscopes signals (device in handbag) and the peak detection is used for step counting.

Chapter 6

Results

The pedometer algorithm accuracy was determined using the running count accuracy (RCA) metric; which compares the recorded number of steps to the number of steps taken by the user. Equation 6.1 describes this relationship.

$$RCA = \left(1 - \frac{|s_d - s_t|}{s_t}\right) 100 \tag{6.1}$$

where s_{d} is the number of steps detected by the algorithm and s_{t} are the steps taken by the user.

However, the downside in using the RCA as the evaluation metric was that it also evaluates the false positives and false negatives. The term false positive, states that the algorithm detected a step when one was not taken and the term false negative, states that the algorithm did not detect a step when one was taken.

This leaves the possibility that this evaluation metric may determine the pedometer to be accurate if the number of false positives and false negatives are roughly the same. Therefore, for the purpose of this investigation, the acceleration norm and gyroscope graphs were individually studied, in conjunction with the RCA to determine the true accuracy of the pedometer algorithm.

The developed algorithm was tested by collecting IMU data from 6 different participants, 3 male and 3 females of different ages, weights and heights. Since the algorithm needs to work irrespective of the users' physical characteristics. Before testing, the participants were asked to place the phone in their pocket or handbag in a natural manner. This was done to test the real-world cases for the developed algorithm.

The participants were requested to complete five different activities: walking on a flat surface, ascending stairs, descending stairs, walking up ramp and finally, walking down a ramp. Walking on flat surface was tested at 3 different speeds: slow(60 – 80 steps/min), normal(100-120 steps/min) and fast(120+ steps/min). The three males placed the device in their pockets while the 3 females carried it in a handbag over the shoulder. The actual number of steps that each participant took was counted by a manual recorder.

6.1 Accuracy of Accelerometer based algorithm

Combined results of accelerometer algorithm for device in pocket and handbag were tabulated in Table 6.1 and 6.2 respectively.

Table 6.1: Results of accelerometer algorithm for participants carrying the device in the pocket

Activity	**Number of steps taken**	**True steps detected by the algorithm**	**Accuracy(%)**
Walking on flat ground(slow)	165	168	98.18
Walking on flat ground(normal)	148	149	99.32
Walking on flat ground(fast)	134	134	100
Ascending stairs	51	54	94.12
Descending stairs	51	52	98.04
Walking up a ramp	75	75	100
Walking down a ramp	77	79	97.40

Table 6.2: Results of accelerometer algorithm for participants carrying the device in the handbag

Activity	**Number of steps taken**	**True steps detected by the algorithm**	**Accuracy(%)**
Walking on flat ground(slow)	150	147	98
Walking on flat ground(normal)	147	144	97.96
Walking on flat ground(fast)	136	133	97.79
Ascending stairs	51	51	100
Descending stairs	51	49	96.3
Walking up a ramp	52	54	96.15
Walking down a ramp	55	55	100

The Figures below, demonstrate the accelerometer-based algorithm results of a single male and female participant. It was shown that the devices works well for both use cases, placed in the pocket and the handbag. It also shows remarkably high accuracies even at slower walking paces.

6.1.1 Walking on flat ground - slow

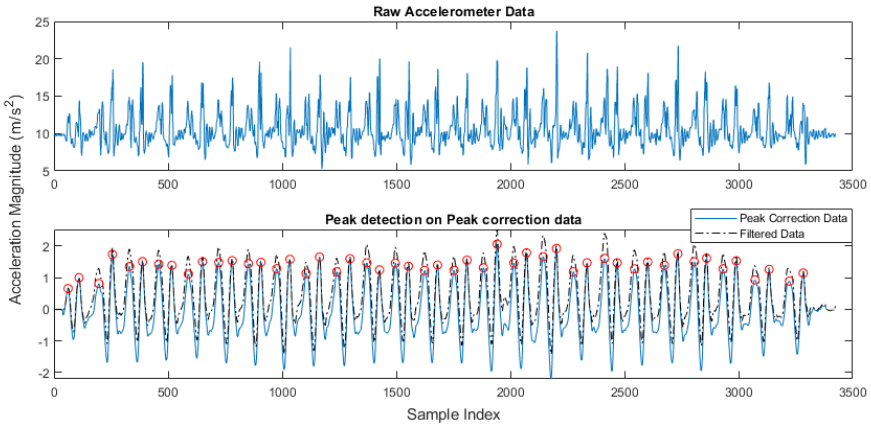


Figure 6.1: Raw and filtered data after peak correction while the user was walking on flat ground at a slow pace, with the device placed in a pocket

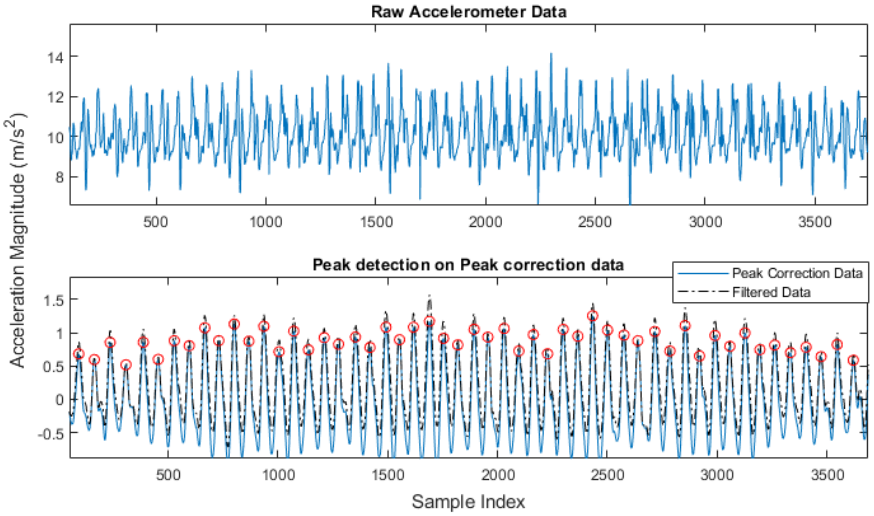


Figure 6.2: Raw and filtered data after peak correction while the user was walking on flat ground at a slow pace, with the device placed in a handbag

6.1.2 Walking on flat ground - normal

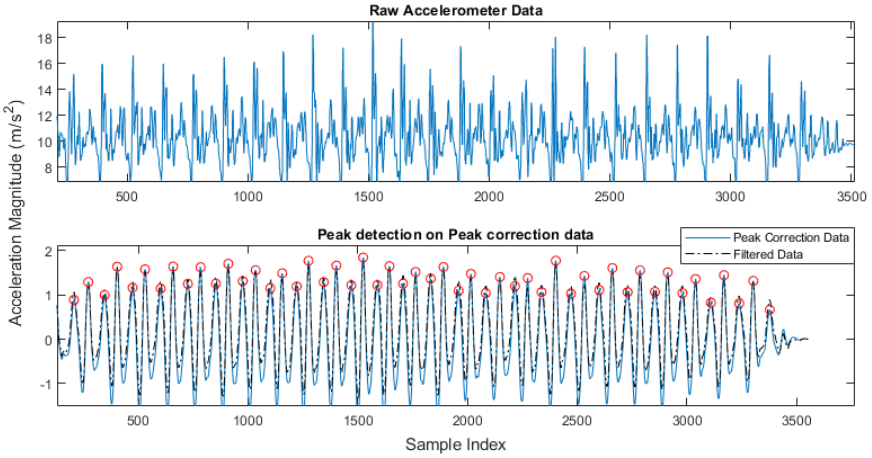


Figure 6.3: Raw and filtered data after peak correction while the user was walking on flat ground at a normal pace, with the device placed in a pocket

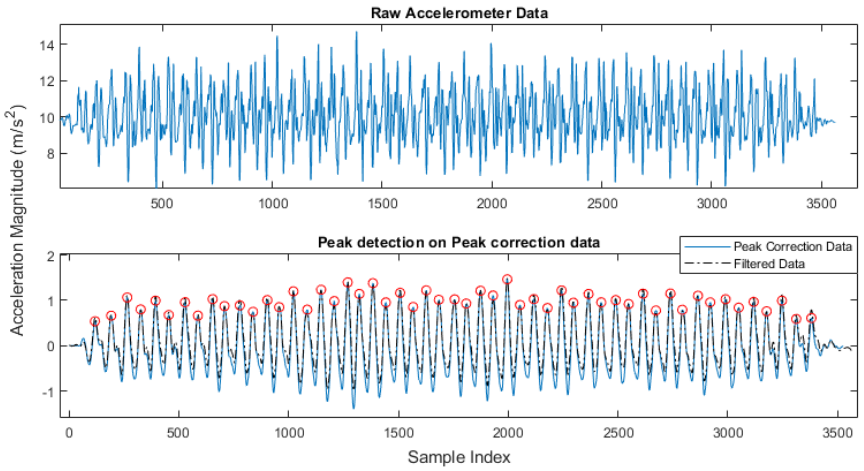


Figure 6.4: Raw and filtered data after peak correction while the user was walking on flat ground at a normal pace, with the device placed in a handbag

6.1.3 Walking on flat ground - fast

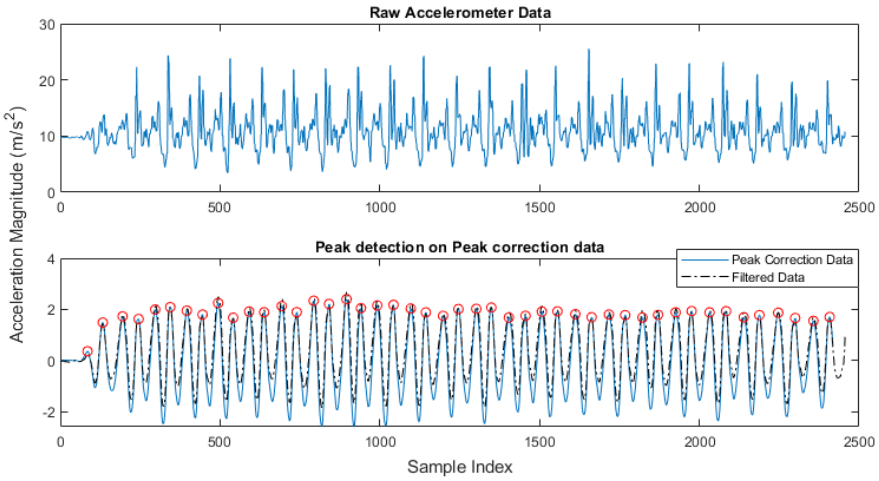


Figure 6.5: Raw and filtered data after peak correction while the user was walking on flat ground at a fast pace, with the device placed in a pocket

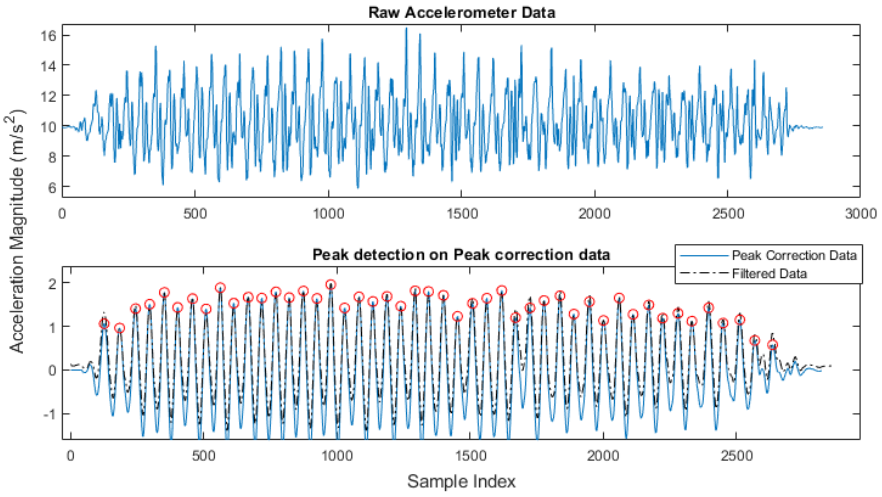


Figure 6.6: Raw and filtered data after peak correction while the user was walking on flat ground at a fast pace, with the device placed in a handbag

6.1.4 Walking up stairs

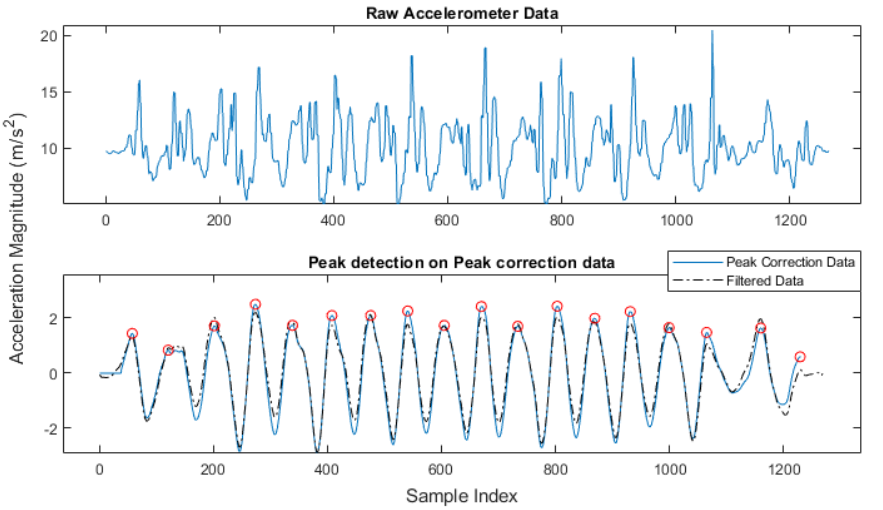


Figure 6.7: Raw and filtered data after peak correction while the user was walking up stairs with the device placed in a pocket

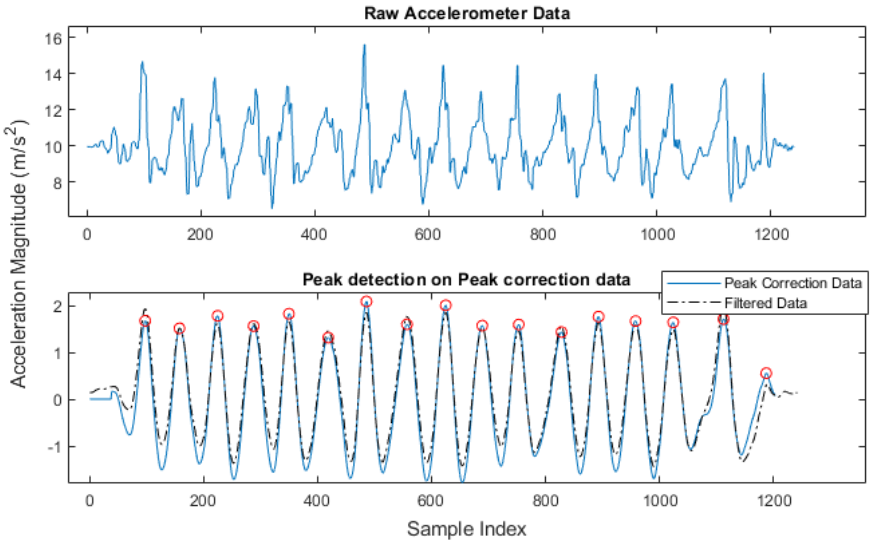


Figure 6.8: Raw and filtered data after peak correction while the user was walking up stairs with the device placed in a handbag

6.1.5 Walking down stairs

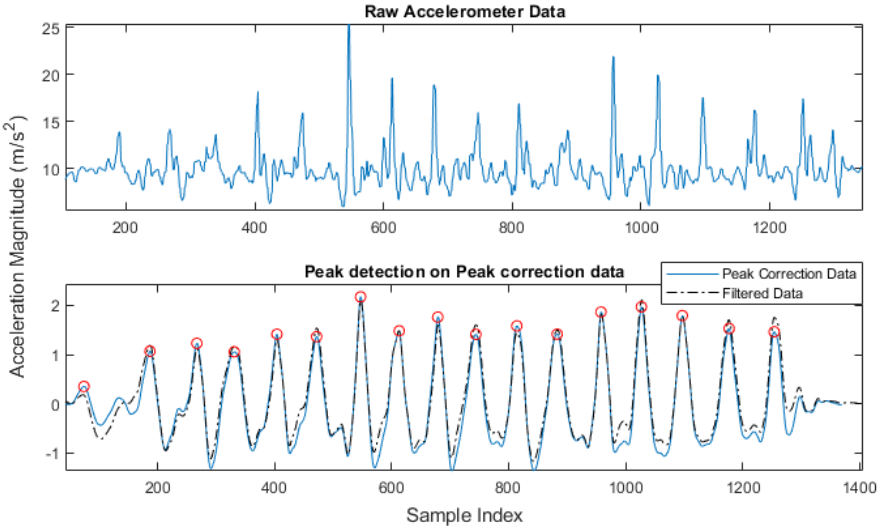


Figure 6.9: Raw and filtered data after peak correction while the user was walking down stairs with the device placed in a pocket

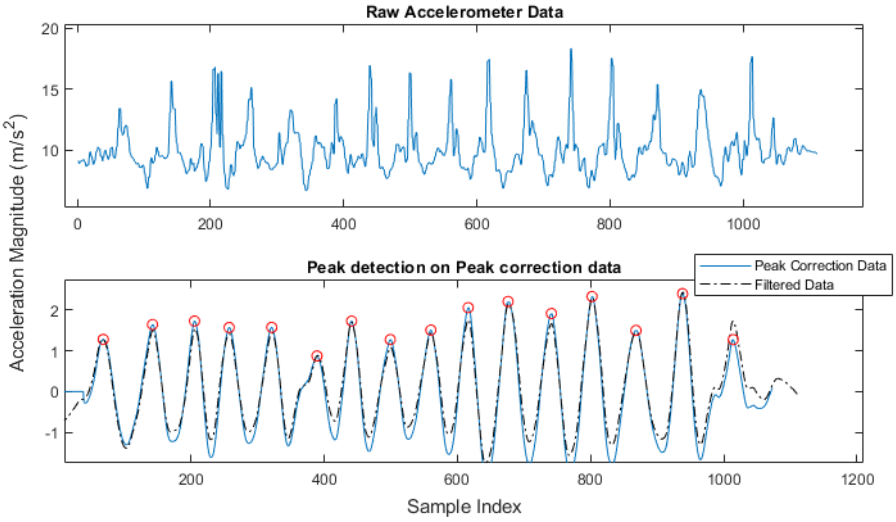


Figure 6.10: Raw and filtered data after peak correction while the user was walking down stairs with the device placed in a handbag

6.1.6 Walking up a ramp

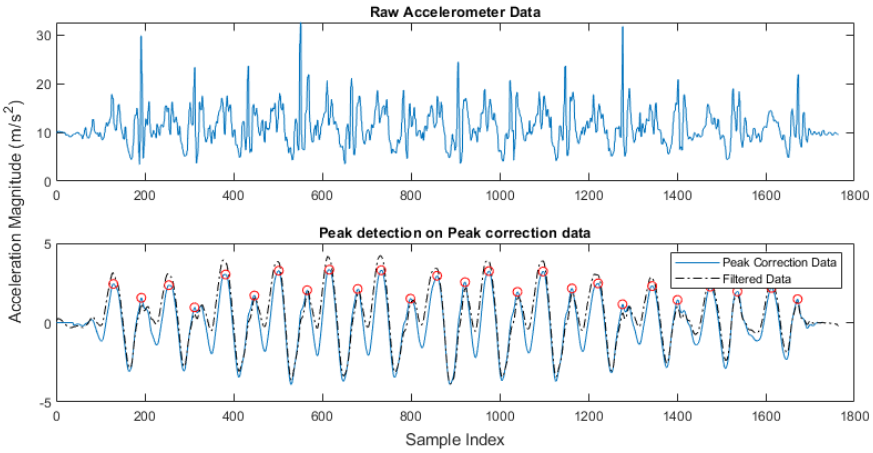


Figure 6.11: Raw and filtered data after peak correction while the user was walking up a ramp with the device placed in a pocket

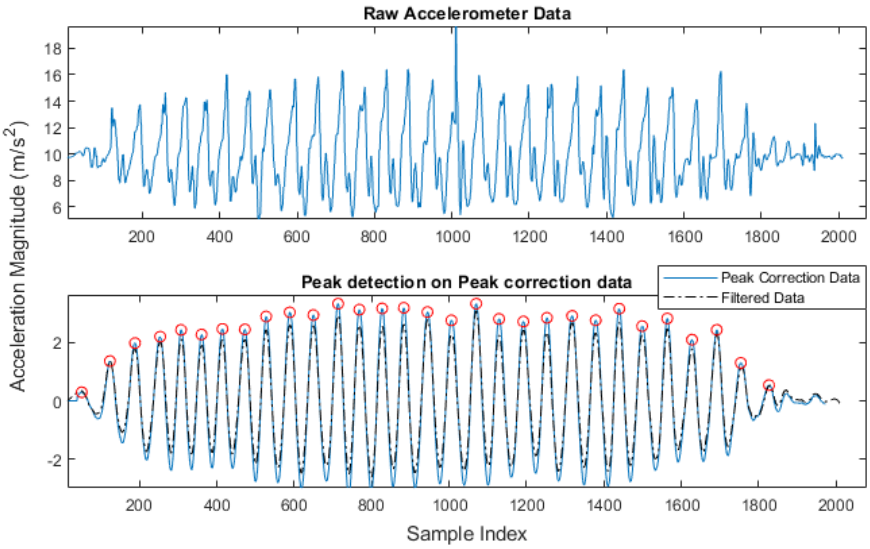


Figure 6.12: Raw and filtered data after peak correction while the user was walking up a ramp with the device placed in a handbag

6.1.7 Walking down a ramp

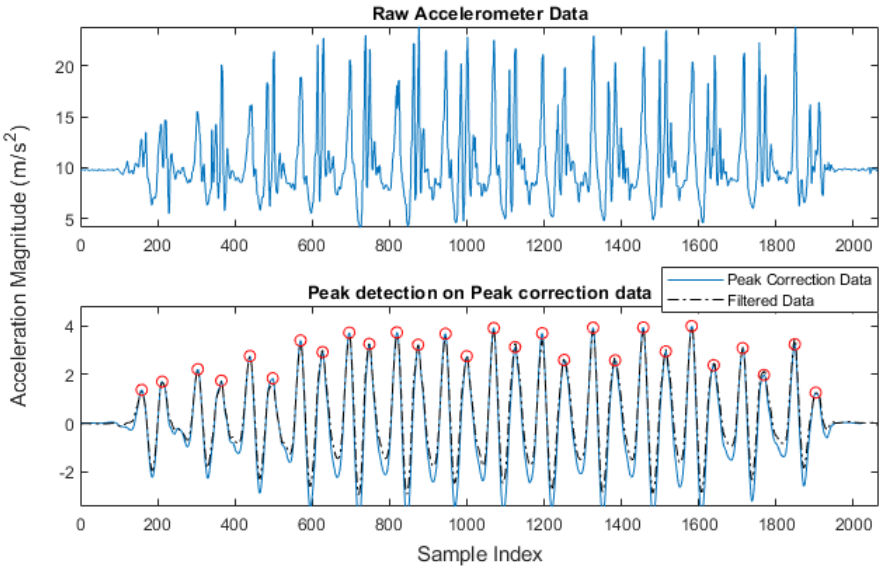


Figure 6.13: Raw and filtered data after peak correction while the user was walking down a ramp with the device placed in a pocket

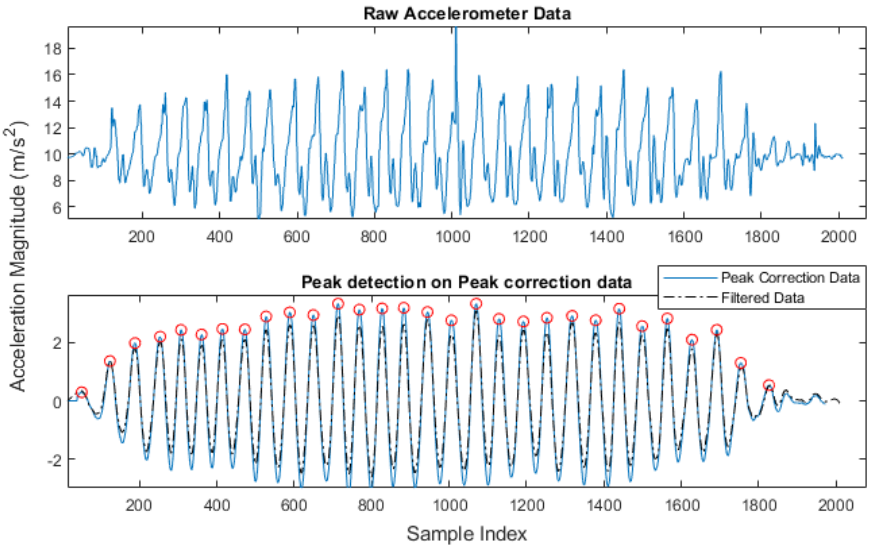


Figure 6.14: Raw and filtered data after peak correction while the user was walking down a ramp with the device placed in a handbag

6.2 Accuracy of Gyroscope based algorithm

Combined results of gyroscope algorithm for device in pocket and handbag were tabulated in Table 6.3 and 6.4 respectively.

Table 6.3: Results of gyroscope algorithm for participants carrying the device in the pocket

Activity	**Number of steps taken**	**True steps detected by the algorithm**	**Accuracy(%)**
Walking on flat ground(slow)	165	168	98.18
Walking on flat ground(normal)	148	149	99.32
Walking on flat ground(fast)	134	130	97.02
Ascending stairs	51	51	100
Descending stairs	51	49	96.08
Walking up a ramp	75	73	97.33
Walking down a ramp	77	78	98.70

Table 6.4: Results of gyroscope algorithm for participants carrying the device in the handbag

Activity	**Number of steps taken**	**True steps detected by the algorithm**	**Accuracy(%)**
Walking on flat ground(slow)	150	139	92.67
Walking on flat ground(normal)	147	145	98.64
Walking on flat ground(fast)	136	132	97.06
Ascending stairs	51	36	70.59
Descending stairs	51	15	29.41
Walking up a ramp	52	53	98.08
Walking down a ramp	55	43	78.18

Demonstration of the gyroscope algorithm is already documented in [6] and therefore will not be reproduced in this report. However, from Fig. 6.15 a demonstration of the dominant axis detector is exhibited. In this use case the participant placed the device in a sideways fashion in which the z-axis was parallel to the direction of gravity. The dominant axis detector declared the y-axis that contributes most to the walking angular rate.

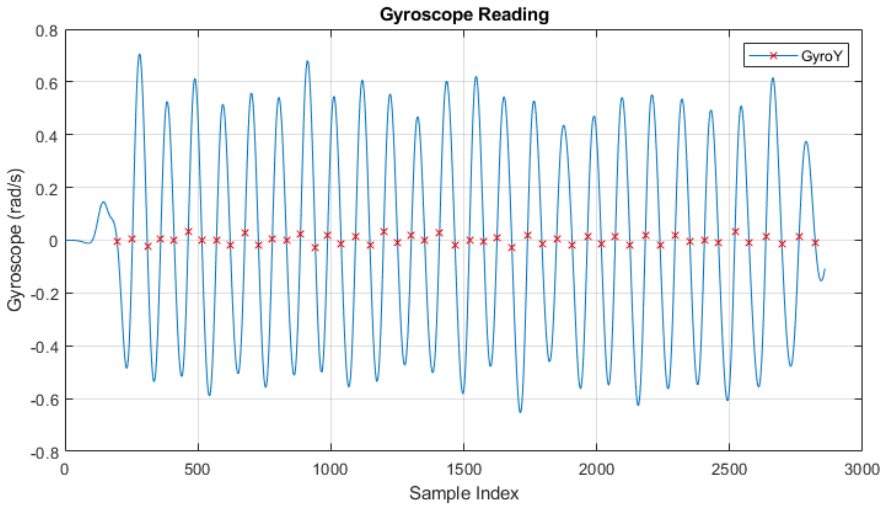


Figure 6.15: Zero-crossing detection for device placed in a handbag in an arbitrary position

6.3 Accuracy of dual sensor-based data implementation

Combined results of accelerometer and gyroscope based algorithm for device in pocket and handbag were tabulated in Table 6.5 and 6.6 respectively.

Table 6.5: Results of dual pedometer algorithm for participants carrying the device in the pocket

Activity	**Number of steps taken**	**True steps detected by the algorithm**	**Accuracy(%)**
Walking on flat ground(slow)	165	168	98.18
Walking on flat ground(normal)	148	148	100
Walking on flat ground(fast)	134	130	97.02
Ascending stairs	51	52	98.04
Descending stairs	51	49	96.08
Walking up a ramp	75	73	97.33
Walking down a ramp	77	78	98.70

Table 6.6: Results of dual pedometer algorithm for participants carrying the device in the handbag

Activity	**Number of steps taken**	**True steps detected by the algorithm**	**Accuracy(%)**
Walking on flat ground(slow)	150	146	97.33
Walking on flat ground(normal)	147	144	97.96
Walking on flat ground(fast)	136	131	96.33
Ascending stairs	51	51	100
Descending stairs	51	49	96.08
Walking up a ramp	52	54	96.15
Walking down a ramp	55	55	100

6.4 Discussion

6.4.1 Discussion of Accelerometer Algorithm Results

The simulations of the developed accelerometer-based algorithm exhibit high accuracies to the accelerometer algorithms discussed in studies [5] [12], especially at slow walking speeds. Their studies displayed average mean errors of 70% and 40% respectively at 60 – 70 steps/min for walking on flat ground. As seen in Table 6.1 and 6.2, the accelerometer algorithm developed in this study displayed mean errors less than 3% for walking on flat ground while placing the device in the pocket and handbag.

The normal and fast walking speed results were compared to two different peak detection algorithms discussed in literature. Both studies showcase experiments done while the phone was placed in the pocket and the handbag, therefore direct comparisons can be made. Although experiments were only done at normal and fast walking paces.

The proposed algorithm indicated overall improvements to the algorithm designed in stude [14]. Their study exhibited an average accuracy of 85.5% while having the device placed in the pants pocket and 90% while having the device in the handbag; whereas the proposed algorithm had a average accuracy of 98.15% and 97.5% respectively.

The proposed algorithm indicated slightly less accuracies than the algorithm developed by Hwan-hee et al. [15]. Their study showed a mean accuracy of 98.7% for both test cases, whereas the proposed algorithm has a mean accuracy of 97.58%. This can be attributed to the fact that study [15] use an adaptive threshold to remove false peaks.

The testing involving stairs were limited to 17 steps, this was due to limited access to long stairways. This made the error percentage large as a single step

contributes to 6% of the total accuracy. Furthermore, the final result was defined by a combination of all recordings from the participants. For example, a false step counted by participant 1 would make up for a step not counted for participant 2.

The accelerometer data was also sensitive to half steps during ascending and descending stairs. Examining Table 6.1 it can be seen that the steps counted by the algorithm are often overcounted for the stairs experiments. This was due to the half step that was taken in order to end in a neutral position.

The algorithm also seldom counted extra steps during slow walking paces, as illustrated in Fig. 6.16 between samples 2700 and 2800. This was caused by the magnitude of oscillations generated by vibrations having similar magnitudes to step signals. This is clearly seen in Table 6.1. Fig. 6.17 to Fig. 6.19 indicated how much more prevalent these oscillations are at slower walking paces

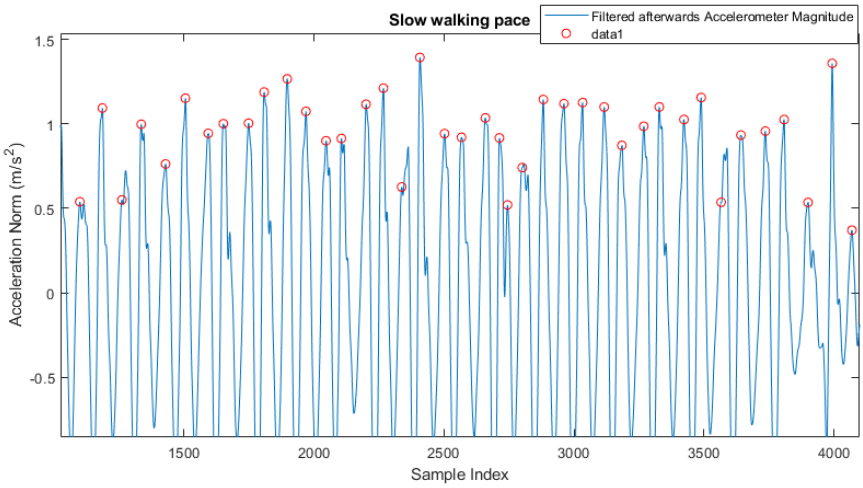


Figure 6.16: Peak detection for slow walking pace while device was place in a pocket

Figure 6.17: Peak detection for slow walking pace while device was place in a pocket

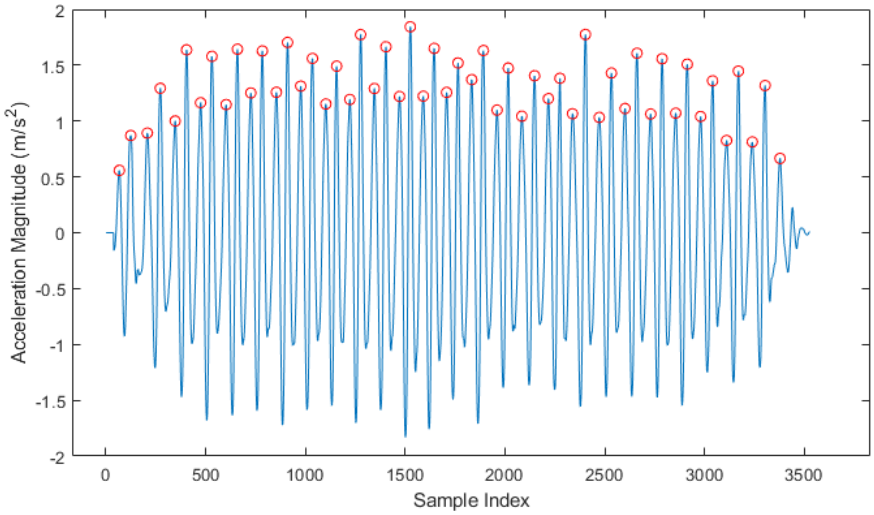


Figure 6.18: Peak detection for normal walking pace while device was place in a pocket

6.4.2 Discussion of Gyroscope Algorithm Results

The accuracy of the gyroscope results with the addition of the dominant axis detector work as described by Jayalath et al. [6]. Table 6.3 shows that the accuracy of the algorithm is inline with what was reported in study [6].

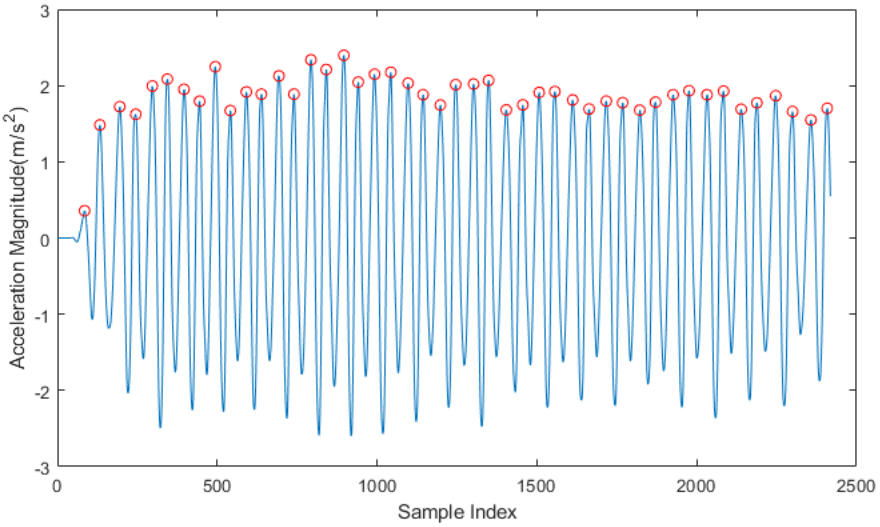


Figure 6.19: Peak detection for fast walking pace while device was place in a pocket

The dominant axis functionality was confirmed by comparing the test case of walking on flat ground for both Table 6.3 and Table 6.4. During recording of the handbag experiments it was noticed that the phone was placed in a landscape manner. This resulted in the y-axis being perpendicular to the gravity vector, instead of parallel as described in [6]. This resulted in the device tracking the y-axis in some cases, as illustrated in Fig. 6.20, showcasing the functionality of the dominant axis detector

It is believed the gyroscope tracks the periodic motion of the torso for the handbag, as illustrated in Fig. 6.20. However, it must be noted that there needs to be sufficient rotation experienced by the device in order for the gyroscope algorithm to work. Since the torso does not induce much rotation when compared to the thigh. Fig. 6.21 shows the gyroscope reading for the same walking activity of the same participant as Fig. 6.20. It was shown that the leg produces nearly five times as much angular rates than the torso. Making motion exhibited by the torso less than ideal when it comes to tracking periodic motion.

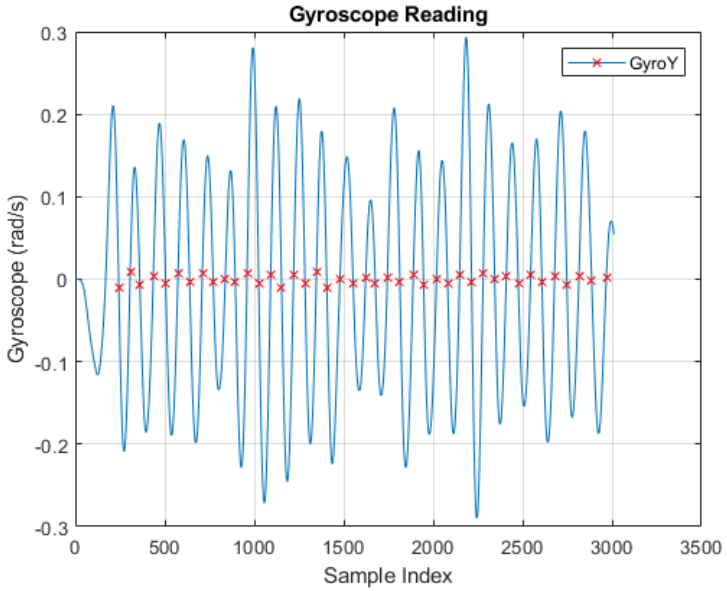


Figure 6.20: Zero-crossing detection for normal walking pace while device was placed in a handbag

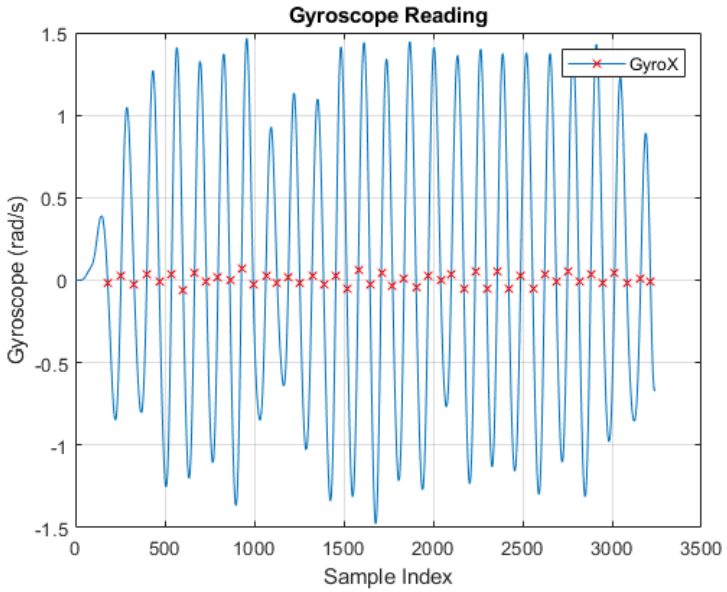


Figure 6.21: Zero-crossing detection for normal walking pace while device was placed in a in the pocket.

The dominant axis detector showed great results in the testing of the recorded dataset.

6.4.3 Discussion of Dual-Sensor Algorithm Results

From comparing Table 6.4 Table 6.6 it can be seen that the classifier in the adequately determines when to switch from the zero-crossing algorithm to the peak detection algorithm.

Fig. 6.22 illustrates that the functionality of the classifier to maximize accuracy. The dataset contains signals from a slow walking pace. Due to high frequency oscillations at this walking speed the gyroscope algorithm will record better results. In Fig. 6.22 it can be seen there exist a gyroscope peak within two accelerometer peaks. The classifier then confirms periodic motion, thus using the zero-crossing algorithm.

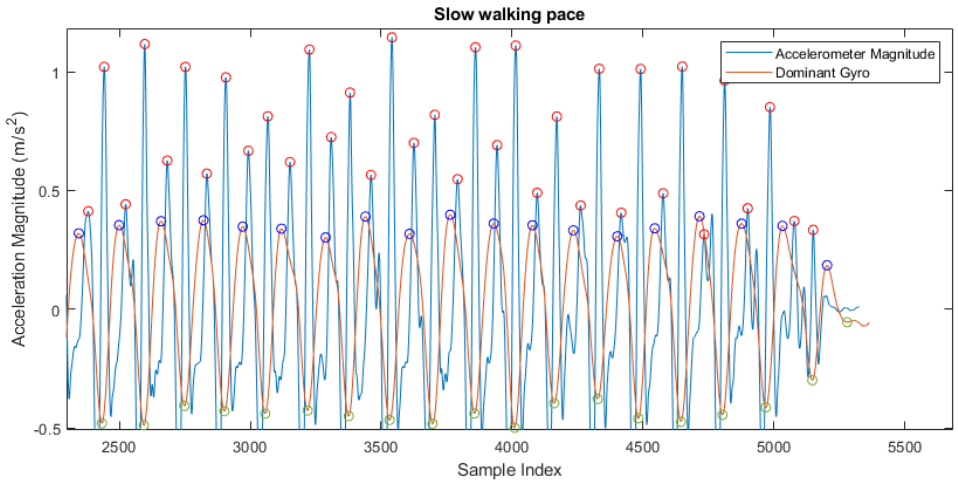


Figure 6.22: Implementation of dual-sensor algorithm. The classifier identifies periodic motion therefore the zero-crossing algorithm was implemented.

Fig. 6.23 illustrates that the functionality of the classifier to maximize accuracy. The dataset contains signals from a descending stairs while the device was placed in

a handbag. The classifier determined there was little to no periodic, therefore the peak detection algorithm was used for step detection.

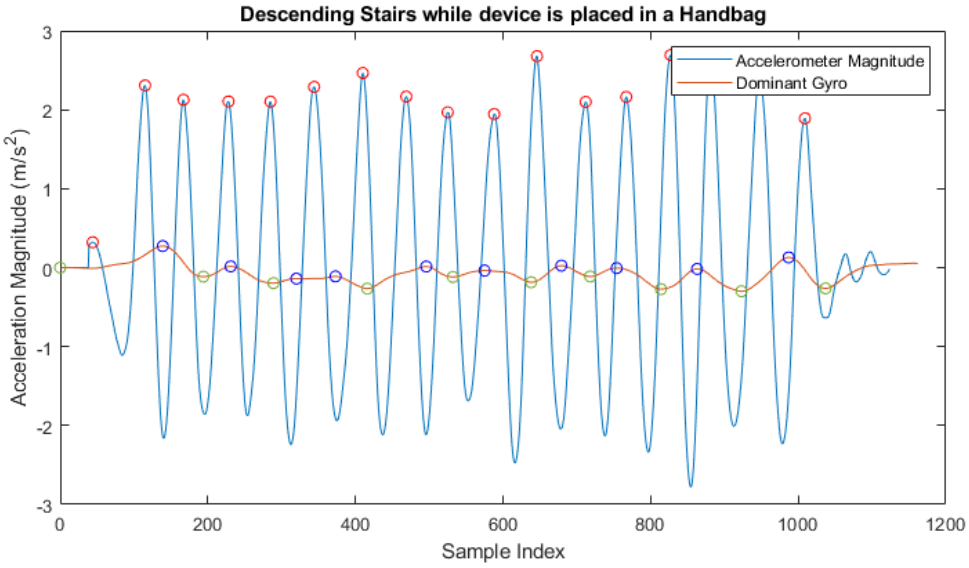


Figure 6.23: Implementation of dual-sensor algorithm. The classifier identifies non-periodic motion therefore the peak detection algorithm was implemented.

Chapter 7

Conclusions

The objective of this study was to design a pedometer algorithm that could accurately detect steps when loosely attached to the user. Therefore, the algorithm needed to be robust enough to filter out high-frequency oscillations that occur during the heel-to-ground contact point and dynamically adapt to changes in the device orientation whilst walking. The algorithm needed to be tested under two use cases: device placement in a pants pocket and a handbag.

The proposed algorithm used both the accelerometer and gyroscope information to declare a classifier state, in which the step detection will switch between zero-crossing(for pockets) and peak detection(for handbag).

The zero-crossing algorithm is based on a single axis algorithm developed in study [6] with an additional feature to determine the most dominant axis. The least dominant axis was detected by determining which axis contributed the most to the gravity vector. This meant only two axes needed to be analysed to determine which contributed the most rotation. The logic behind finding the most dominant axis was discovered by analysing the mean of the gyroscope data. The gyroscope signals showed a mean value close to zero. This makes sense since the gyroscope signal oscillates around the zero point when walking. The value standard deviation of the

remaining two axes determined the most dominant axis as it represents the 'spread' of the signal. A higher spread equated to greater angular rotation magnitude. This method performed well in cases where the device was placed in both loose pockets and a handbag. By knowing the most dominant axis, the zero-crossing algorithm can be used for different device orientations.

The peak detection algorithm was based on accelerometer data. This method involved using the contribution of all three axes to track the dynamically changing orientation of the smartphone. The resulting accelerometer signal was spectrally analysed to verify frequencies derived from the walking motion. A moving average filter was used due to its simplicity and low computational cost. This greatly reduced the high-frequency oscillations that are the culprit to inaccurate peak detection algorithms. To further reduce these high-frequency oscillations a mean difference function was applied to the filtered data. This greatly reduced false peaks, especially for slow walking speeds.

The integration of the dual-sensor algorithm was not complete to the degree the author of this paper intended. This was due to the idea being realised too late in the process.

The current implementation of the classifier uses information from the gyroscope to determine the IMU signals periodicity. It was discovered that between every two accelerometer peaks there exists a corresponding gyroscope peak or valley. By confirming this relationship, a periodicity check was done to determine whether the zero-crossing algorithm should be used instead of the peak detection algorithm for step detection.

The idea behind the dual-sensor algorithm was to use information from the gyroscope to supplement the accelerometer algorithm, thus still using the accelerometer data as the basis for the pedometer algorithm. This was desired since the accelerometer-based algorithm was more applicable in different uses cases, whereas

the gyroscope algorithm only achieved desirable performance while appended to a body part.

Although the classifier is not completely robust, it still exhibits high accuracies for a specific use case. The developed accelerometer algorithm however showed remarkably high results for the recorded datasets despite its simplicity.

7.1 Future Work

This investigation only focused on the design of the algorithm and not real-world implementation. The incorporation of the algorithm in a smartphone app is recommended to test the robustness of the algorithm and detect steps in real-time.

It was discovered during the late stages of this project that the resultant accelerometer magnitude can be used to adaptively determine the walking frequency. By inspecting subsequent peak/valley magnitude and time variance, an updating walking speed can be determined. This updating speed could be used to create an adaptive filter. This would greatly reduce the false peaks during slow walking speeds.

The dominant axis method designed in this project is not very robust as it can only be determined after all the data has been captured. A previous student developed a method to use superposition to generate a dominant axis by adding two axes to increase signal strength. Although a robust method of determining angular rate polarity must be developed to ensure the signals are in phase with one another.

The dual-based sensor algorithm can be made more robust by creating a classifier to supplement the peak detection algorithm instead of replacing it with zero-crossing. The idea behind this is to match every i_{th} gyroscope peak or valley to every $(i_{\text{th}} \cdot 2) - 1$ accelerometer peak to validate the performance of the accelerometer during time windows of periodic motion.

This study focuses on the usage of an android smartphone to record sensor data,

however due to the event-driven architecture used by Android Studio, samples were being recorded at an inconsistent sampling rate. A recommendation is to buy a dedicated IMU or MEMS sensor to record the data.

Bibliography

[1] E. Gregersen, "Martin cooper," (Accessed on 11/7/2021). [Online]. Available: https://www.britannica.com/biography/Martin-Cooper

[2] S. M. Yusof, J. I. Ahmad, and N. H. A. Talib, "Mobile calorie burned estimation based on pedometer steps," in *Regional Conference on Science, Technology and Social Sciences (RCSTSS 2016)*. Springer, 2018, pp. 43–53.

[3] L. Butler, S. Furber, P. Phongsavan, A. Mark, and A. Bauman, "Effects of a pedometer-based intervention on physical activity levels after cardiac rehabilitation: a randomized controlled trial," *Journal of cardiopulmonary rehabilitation and prevention*, vol. 29, no. 2, pp. 105–114, 2009.

[4] J. A. Hesch and S. I. Roumeliotis, "An indoor localization aid for the visually impaired," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3545–3551.

[5] M. Oner, J. A. Pulcifer-Stump, P. Seeling, and T. Kaya, "Towards the run and walk activity classification through step detection-an android application," in *2012 annual international conference of the IEEE engineering in medicine and biology society*. IEEE, 2012, pp. 1980–1983.

[6] S. Jayalath and N. Abhayasinghe, "A gyroscopic data based pedometer algorithm," in *2013 8th International Conference on Computer Science & Education*. IEEE, 2013, pp. 551–555.

[7] V. Gikas and H. Perakis, "Rigorous performance evaluation of smartphone gnss/imu sensors for its applications," *Sensors*, vol. 16, no. 8, p. 1240, 2016.

[8] M. Whittle, D. Levine, and J. Richards, *Whittle's gait analysis*. Butterworth-Heinemann, 2012.

[9] V. Studdert, C. Gay, and K. W. Hinchcliff, *Saunders comprehensive veterinary dictionary*. Elsevier Health Sciences, 2020.

[10] L. R. Sheffler and J. Chae, "Hemiparetic gait," *Physical Medicine and Rehabilitation Clinics*, vol. 26, no. 4, pp. 611–623, 2015.

[11] X. Kang, B. Huang, and G. Qi, "A novel walking detection and step counting algorithm using unconstrained smartphones," *Sensors*, vol. 18, no. 1, p. 297, 2018.

[12] J. B. Martin, K. M. Krc, E. A. Mitchell, J. J. Eng, and J. W. Noble, "Pedometer accuracy in slow-walking older adults," *International journal of therapy and rehabilitation*, vol. 19, no. 7, pp. 387–393, 2012.

[13] S. Crouter, P. Schneider, M. Karabulut, and D. Bassett, "Validity of 10 electronic pedometers for measuring steps, distance, and energy cost," *Medicine and science in sports and exercise*, vol. 35, pp. 1455–60, 09 2003.

[14] Y. Chon, E. Talipov, and H. Cha, "Autonomous management of everyday places for a personalized location provider," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 518–531, 2011.

[15] H.-h. Lee, S. Choi, and M.-j. Lee, "Step detection robust against the dynamics of smartphones," *Sensors*, vol. 15, no. 10, pp. 27 230–27 250, 2015.

[16] P. Li, M. Abdel-Aty, and Z. Islam, "Driving maneuvers detection using semi-supervised long short-term memory and smartphone sensors," *Transportation Research Record*, p. 03611981211007483, 2021.

[17] T. Ji and A. Pachi, "Frequency and velocity of people walking," *Structural Engineer*, vol. 84, no. 3, pp. 36–40, 2005.

[18] F. Massaad, T. M. Lejeune, and C. Detrembleur, "The up and down bobbing of human walking: a compromise between muscle work and efficiency," *The Journal of physiology*, vol. 582, no. 2, pp. 789–799, 2007.

[19] M. Tomlein, P. Bielik, P. Krátky, S. Mitrík, M. Barla, and M. Bieliková, "Advanced pedometer for smartphone-based activity tracking." in *HEALTHINF*, 2012, pp. 401–404.

[20] G. Palshikar, "Simple algorithms for peak detection in time-series," 01 2009.

Appendix A

Code

GitHub repository containing the MATLAB Pedometer code and Android Application:

https://github.com/Trivaan/EEE4022S-Thesis-Project

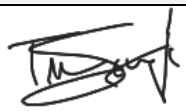
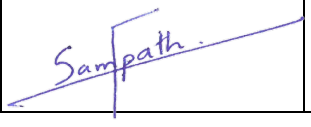
Application for Approval of Ethics in Research (EiR) Projects
Faculty of Engineering and the Built Environment, University of Cape Town

ETHICS APPLICATION FORM

Please Note:
Any person planning to undertake research in the Faculty of Engineering and the Built Environment (EBE) at the University of Cape Town is required to complete this form **before** collecting or analysing data. The objective of submitting this application *prior* to embarking on research is to ensure that the highest ethical standards in research, conducted under the auspices of the EBE Faculty, are met. Please ensure that you have read, and understood the **EBE Ethics in Research Handbook** (available from the UCT EBE, Research Ethics website) prior to completing this application form: http://www.ebe.uct.ac.za/ebe/research/ethics1

APPLICANT'S DETAILS		
Name of principal researcher, student or external applicant		Trivaan Singh
Department		EBE
Preferred email address of applicant:		Sngtri006@myuct.ac.za
If Student	Your Degree: e.g., MSc, PhD, etc.	BSc ENG Mechatronics
	Credit Value of Research: e.g., 60/120/180/360 etc.	40
	Name of Supervisor (if supervised):	Sampath Jayalath
If this is a researchcontract, indicate the source of funding/sponsorship		
Project Title		Robust pedometer algorithm for mobile phones whilst the device is loosely attached to the body.

I hereby undertake to carry out my research in such a way that:
- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

APPLICATION BY	Full name	Signature	Date
Principal Researcher/ Student/External applicant	Trivaan Singh		13/08/2021
SUPPORTED BY	Full name	Signature	Date
Supervisor (where applicable)	Sampath Jayalath		

APPROVED BY	Full name	Signature	Date
HOD (or delegated nominee) Final authority for all applicants who have answered NO to all questions in Section 1; and for all Undergraduate research (Including Honours).			