Detect Specific Movement Patterns Based on Gyro and Accelerometer Data

NIKLAS HE
ROBIN OLOFSSON

MASTER´S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Detect Specific Movement Patterns Based on Gyro and Accelerometer Data

Niklas He
dic15nhe@student.lu.se

Robin Olofsson
dic15rol@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor: Jan Eric Larsson

Assistant Supervisor: Per Eriksson

Axis Supervisors: Peter Eneroth and Johan Förberg

Examiner: Maria Kihl

June 10, 2020

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

Wearable security cameras has been used by police officers and other agents working within the field of security for many years. Some of these cameras are equipped with an accelerometer and gyroscope sensor that can be used to detect if something is happening to the wearer of the camera. Ideally, the recording should start automatically if the camera is affected by sudden movements, for example if the wearer of the camera is being attacked. If this happens, then the camera should start recording immediately to capture the situation. This masters thesis project will be about trying to develop an efficient and accurate algorithm for fall detection with the help of the accelerometer and gyroscope sensors.

Measurements and data collection was done by attaching a body worn camera to a user's chest with the help of a belt. The user performed different movements and the sensor output data was saved into separate files which was later made into a complete data set. Each file was then plotted for analysing, evaluation and trying to detect different patterns.

An iterative strategy has been taken for developing this algorithm meaning that new functions and features were continuously added to the algorithm throughout this process in order to improve its performance. This Master's thesis project will focus on how to detect a fall while also trying to filter out movements and other actions that may look similar to a fall.

The developed threshold-based algorithm was able to detect falls with a sensitivity of 100% and a specificity around 90%.

Acknowledgements

This section is dedicated to all the people that have helped and supported us throughout this entire process of completing this Master's thesis project.

First we would like to thank our two supervisors from Lund University for the help they have given us and making sure that we stayed on schedule. Jan Eric Larsson, for helping us start this project and providing important knowledge on different things to keep in mind when doing a Master's thesis project. Per Eriksson, for giving us guidance within the area of signal processing, on how to properly write this report and how to execute a Master's thesis project.

Next we would like to thank our supervisors over at Axis Communications. Johan Förberg, for giving us guidance within the area of signal processing, help us with the algorithm design and help us come up with new ideas and approach problems in a different way. Peter Eneroth, for creating this Master's thesis project in the first place and giving us permission to do it, for answering all our questions and giving us new ideas, for providing us with all the different things that we needed and for making sure the the work we produce is of high quality.

These last five months has been challenging, fun and instructive and we are truly grateful for being given the opportunity to work with all the people that was part of this. Your help has truly been essential and for that we want to express our deepest gratitude.

Popular Science Summary

Development of a fall detection algorithm using accelerometer and gyroscope sensors

Wearable cameras are used today by the police to capture footage from crime scenes. The officer may be unable to start the camera because he is attacked and falls to the ground. An accurate fall detection function will automate this process.

Security cameras are of high importance in today's modern society to keep everything safe and in order. Some of the cameras today are equipped with different sensors, such as an accelerometer and gyroscope. With the help of these sensors the camera is able to detect if it is being attacked or exposed to certain circumstances. One specific type of such cameras are wearable cameras.

For some time, police officers and other agents working within the field of public security have been using these wearable cameras to capture footage from hectic situations. Ideally, the recording should start automatically if the camera is affected by sudden movements. This could for example be if the wearer of the camera gets attacked and falls to the ground. If this were to happen the user of the camera could be unable to manually start the recording himself because he could be injured or physically restrained.

There are constantly new cameras being developed with the goal of improving functionality and performance. The goal of this Master's thesis work will be to develop a new and more efficient algorithm for fall detection with the help of a wearable cameras accelerometer and gyroscope sensors.

The first step in the development of this fall detection functionality was to first understand how the sensors worked. Sensor data was recorded by attaching the wearable camera to a user with the help of a belt. The placement of the camera was right on the chest since this is normally where a wearable camera is placed. Different falls and other daily activities, such as running, walking up and down stairs and jumping were considered. This was done by asking the user of the camera to perform these activities to the best of his abilities. The daily activities was recorded with ease and no problems. However, the falls was a different story and required great effort and determi-

nation. The data could then be plotted onto a graph to see how the sensors reacted to a specific movement. To further improve the accuracy and precision (correctness) of the algorithm, different movements, such as throwing the camera, was recorded with the goal of trying to trick the algorithm. These recordings helped to improve the performance of the algorithm.

The development of the algorithm in itself was done by gradually adding new features and functions. If a feature improved the results it was added while if the opposite happened it was removed. A lot of combinations and testing was done that lead to the final result. This project then ends with a discussion of the result produced, how the execution could have done differently and possible future work.

The developed algorithm was able to detect falls at a 100% accuracy while also avoiding false fall detections at around 90% accuracy. The fall detection algorithm produced in this paper was the result of plenty of data analysing, usage of different methods and testing. It was indeed a fun and challenging opportunity to sharpen ones mind and critical thinking.

Table of Contents

List of Figures

List of Tables

Glossary and Abbreviations

This section contains the specific words and abbreviations that have been used throughout this Master's thesis project. Below is an explanation of what each abbreviation stands for.

BWC	Body Worn Camera
SCU	System Control Unit
VMS	Video Management System
EMS	Evidence Management System
IoT	Internet of Things
LFT	Lower fall threshold
UFT	Upper fall threshold
FI	Falling index
ADL	Activities of Daily Living
DTFT	Discrete-time Fourier transform
DTFFT	Discrete-time fast Fourier transform
TP	True positive
FN	False negative
TN	True negative
FP	False positive
SMA	Simple Moving Average
GUI	Graphical User Interface

Chapter 1

Introduction

A fall is more often than not caused by an unexpected event or emergency which can result in severe injuries and damage. It could for instance be an elder person taking a bad step or a police officer getting attacked and falls to the ground. During such accidents or incidents it is very possible that the victim is in need of help and assistance.

In today's modern society with many devices having access to an internet connection and different sensors, a fall detection algorithm can be implemented and provide great support. If the user is unable to physically press the help button, an automatic fall detection algorithm can serve as a difference in these situations. A device including a fall detection algorithm gives it the possibility to detect falls. Such device can for instance be a smartphone or a wearable device on the wrist. Upon detection, appropriate and necessary action can be taken which can make a difference in many situations.

In this Master's thesis project, a fall detection algorithm is developed and the different steps leading up to the final result discussed. The sensor data recordings were done with a wearable camera provided by Axis. Over 500 cases of different movement activities in total were recorded and studied during this project. The final fall detection algorithm was then implemented into the same wearable camera. This wearable camera will mostly be used by police officers and other people working within the area of public security to capture footage.

1.1 Report Structure

This report contains the results that has been produced in this 20 week long Master's thesis project. The project has been done with the goal of developing a fall detection algorithm, doing research within this area and to give room for possible future work. The report is divided into seven different chapters.

1. Introduction - This chapter will give information about the general goal of this Master's thesis project and its delimitations. After reading this chapter the reader will have an overall understanding of the topic of this project. It will also include previous work and research that has been done within the area of fall detection.

2. System Overview - This chapter aims to give a brief overview of the hardware that has been used and its functionality.
3. Theory - This chapter will give the reader information about the accelerometer and gyroscope sensors and its different settings. It will also include other important metrics, methods and functions that has been used in the development of the fall detection algorithm.
4. Method - A description about the data collection and data validation will be given in this chapter and how it was executed. It will also include the different steps in developing the fall detection algorithm.
5. Results - The results that has been produced will be presented in this chapter. It will include the performance of the different algorithms and its accuracy.
6. Discussion - Comments and discussion about the result will be shared in this chapter. It will also include overall thoughts about this entire Master's thesis project process overall.
7. Conclusion - This chapter will wrap up this report and draw some final conclusions about this process in general.

1.2 Purpose

The purpose and aim of this Master's thesis work is to develop a fall detection algorithm for an wearable device, in this case a camera. This functionality will provide increased safety and security for the user of the camera. During emergencies and other stressful and intense situations the user may forget to start a video recording or may be unable to do so because of some physical restraint. Therefore a function that will be able to detect these situations are of interest. The possibility of automatically being able to trigger a recording upon detection of a fall is highly attractive and of high importance.

1.3 Background

There has been plenty of development and research within the area of movement detection. A couple of areas where movement detection is useful is analysing the movements of sport athletes, using it for surveillance and measuring health of elder people. One thing all these research has in common is the usage of different sensors. Two sensor that has been found particularly useful are the accelerometer, which measures acceleration and the gyroscope which measures angular velocity. More information and explanation about these two sensors will be given more in chapter 3, Theory. By reading sensor data and combining it with different signal processing methods specific movement patterns can be detected and analysed. The different sensors are usually placed on the wrist, ankle or chest which has all been found useful depending on the situation.

This Master's thesis project will focus on a more specific movement detection, namely fall detection. A lot of previous research of fall detection has been done.

These previous studies has however been more focused towards elderly care and older people. The reason for this could be that the result after a fall can be much severe and life threatening compared to if a young a healthy person falls. This Master's thesis project will instead focus on younger and, most likely, healthier adults which means that the fall may look a bit different compared to an elder person. The falls that will be recorded will mostly be performed by the authors of this report plus additional volunteers. Seniors usually move a bit slower in life which means that the fall that happens to them are often also slower.

The novelty that this project will provide is a couple of fall detection algorithms and overall more research based on younger and more healthy people falling to the ground. With the help of the collected data together with different signal processing methods and algorithm design a fall detection algorithm has been developed. It should also be possible to implement the result and apply the new research into other devices with different use cases.

1.4 Problem Formulation

Is it possible to develop an accurate and precise fall detection algorithm with the help of accelerometer and gyroscope data together with different signal processing methods?

1.5 Project Aims and Main Challenges

The goal of the Master's thesis work is to develop, evaluate and suggest algorithms used for detection of specific movement patterns of a person. The developed algorithms should be compared against each other in order to find the most suitable one. A prototype of the most promising algorithm will then be implemented into a wearable camera.

1.6 Research Questions

Fall detection is a functionality that can be implemented into many wearable devices. There has been a lot of research done within this area and there is still much room left for improvement. This Master's thesis project is based on the following research questions.

- Is it possible to detect a fall using the accelerometer and gyroscope sensors in the camera?
- Is it possible to separate a fall from other similar movements?
- How accurately can the developed algorithms recognise a fall?
- How simple and robust can the algorithms be?

1.7 Previous Work

There has been plenty of research done over the recent years in the area of fall detection. Fall detection can be described as the functionality to recognize that a person has fallen to the ground for some arbitrary reason. It is especially for elder care where the research has had its focus. The reason for why elder care has become the focus is that with age, people tend to lose their muscle strength. This also means that they start having issues with weakening of the legs and as a consequence of that, also their balance capabilities starts to decline. As the balance fail, people tend to fall more frequently when they get older. Falling in general can lead to fractures, and for the elderly people it can lead to even more severe consequences.

Most sources that have been read agree on using both accelerometers and gyroscope sensors attached to a person to feed the algorithm data that can be used to recognize the falling motion. There seems to be two main techniques for detecting falls, with the help of machine learning or a more simple threshold-based algorithms. Many of the methods claim good abilities in detecting falls, with sensitivity and specificity measurements not far from or reaching 100%, in lab environments. In short, sensitivity is a measurement on the number of true alarms detected while specificity is a measurement of how many false alarms were avoided, more of this in Section 3.11.

1.7.1 Thresholds

Huynh et al. [1]-[2] uses the total sum vector, also known as magnitude of both the accelerometer and gyroscope output data, to construct two new signals. For the signal consisting of the accelerometer magnitude samples, when a local minimum passes a certain threshold called the lower fall threshold (LFT) samples are recorded for 0.5 s, referred to as the fall window period. If both the accelerometer and gyroscope magnitude signals passes the two upper fall thresholds (UFT) within the fall window, the algorithm will alarm that a fall has been detected. Huynh et al. [2] presents results of sensitivity between 81.37% - 97.36% and specificity of 75.93% - 99.38%. It is worth noting that it is possible to maximize one of these two measurements at the cost of a lower score on the other, it depends on the use case of what is wanted and acceptable.

G. Brown and M. Eklund [3] has a similar approach to their presented algorithms as to Huynh et al. [1]-[2]. The similarity is the use of an upper thresholds for the accelerometer, while it differs in the lack of a lower threshold for the accelerometer, does not use a gyroscope sensor, looks for large changes in user angle (calculated from the accelerometer output), analyses deviations from the subjects normal posture, such as a subject lying horizontal on the floor after a fall. It must be noted that with posture analysis the algorithm becomes dependent on the mounting of the recording device, and great care must be taken to not misinterpret the output data. G. Brown and M. Eklund [3], claims a sensitivity of around 90% and a specificity of 95%.

Kangas et al. [4] compares different algorithms and parameters that can be used in threshold based algorithms, and one of them is the falling index measurement. Shortly described Falling index (FI) is a measurement on how much or how fast the axes of the signal changes over a certain time window, closely related to the concept of a derivative. More details about falling index is presented in Subsection 3.9. Placement of the sensors on the subjects and the effects on the algorithms are also examined. The placements examined are waist, head and wrist, where it is found that the wrist is not an applicable placement for the compared algorithms. Kangas et al. [4] present a sensitivity varying between 76% - 97% and a specificity of 100%.

A key difference between the previous work mentioned in this section and this Master's thesis work is the intended user group. All of the previous work research have been focused on elder care and elderly people wearing the recording devices (it is worth noting that most recordings were done by young adults up to middle aged subjects), however this is not the intended user group of this Master's thesis work.

1.7.2 Machine Learning

Another popular approach into solving this fall detection problem is with the use of machine learning. When using machine learning it is important to note that this method usually requires a much larger data set compared to a approach of not using machine learning. The reason for this is because the algorithm requires a lot of data to train on in order to make it as accurate as possible.

Yin et al. [5] published a research paper which had the goal of trying to detect different human activity, such as running and sitting, by using machine learning to analyze the data. Here they used a Android phone equipped with both accelermoter och gyroscope as the device to record sensor data. The phone was placed in the pocket during recording. Three of the different machine learning classifiers they used were: J48 which uses decision trees, support vector machine (SVM) for trying to find nearest data point and Perceptron which trains by comparing output. The results they got from the three mentioned machine learning classfiers was very promising. Each classifier had accuracy of over 99% in detecting different human activities.

Vallabah et al. [6] performed a similar study to the one in this Master's thesis project where they tried to detect falls while separating it from other daily activities, such as bending forward and sitting down. In this research project they used a prerecorded data set that included other sensors then accelerometer and gyroscope such as a magnetometer. It can also be noted that the sensors were placed in the pocket. A couple of different machine learning classifiers was used in this research including the ones Yin used in their research mentioned above. The result was measured using sensitivity and specificity to see how good each machine learning classifiers was. Here they were able to get sensitivity of between 85.11% to 90.70% and a specificity of between 72.73% to 83.78%.

The main concept of using a machine learning overall is to first train a classifier and then make it good enough so that it can detect specific features and movements such as running, jumping and falling on its own.

The reason machine learning was not used in this Master's thesis project was because it required a big data set for the machine learning classifiers to train on.

There was also a risk that the final algorithm would get too complicated since it was also going to be directly implemented into the camera. The reader might also wonder why available datasets have not been used in this Master's thesis project and the main reason for that was because the data was not compatible. A lot of the sensors was placed at the wrong location, such as on the ankle or wrist, during recording of data and the data overall was a bit unorganised and difficult to read.

1.8 Delimitations

The algorithm developed in this Master's thesis project will mainly be about detecting an arbitrary fall. The algorithm should also be able to avoid triggering on other daily activities such as sitting and walking. There has also be some research done with trying to detect different pushes but the main focus is on fall detection. The data used in this project was recorded only by young adults and the sensor was always placed on the chest. Other places to put the sensors such as wrist and ankle will not be attempted. This report will not focus on how the hardware that has been used works. It is also assumed that the final algorithm is for the camera that has been used for collecting data. Testing to see if the final algorithm works on other devices with similar sensors will not be done in this Master's thesis project.

1.9 Axis Communications

Axis Communications is a company that manufactures cameras for the physical security and video surveillance industries. The company was founded in Lund, Sweden and has today around 3000 employs in over 50 countries. New products are constantly being developed and innovated and the wearable camera used in this Master's thesis project is one of them [7].

1.10 Contributions

This report and Master's thesis project has been done by Niklas He and Robin Olofsson for the Department of Electrical and Information Technology at the Faculty of Engineering, Lund University. The work has overall been evenly distributed with a couple of distinctions:

Niklas He has written a majority of the Result chapter. He has also provided with overall necessary equipment that has been used.

Robin Olofsson has written a majority of the Theory and Discussion chapter and also creating most of the figures used in this report. He has also developed and tested different functions and methods in Python.

Both students has helped collecting data, generating ideas and writing code throughout this entire process. They have also both helped reviewing and writing this report.

Chapter 2
System Overview

The system consists of a body worn camera (BWC), a docking station, a system control unit (SCU), video and evidence management systems (VMS & EMS). Together these items form a secure system that allows the users to capture video of their everyday work and store it in the cloud in an VSM or EMS. The system fulfills the requirements that allow the captured videos to be used in legal matters, making the everyday life of the users more secure.

2.1 Body Worn Camera

The BWC is a wearable camera as indicated by the name. It is equipped with two buttons, one button for starting and stopping recording and one button for turning it on and off. It allows the wearer to start video recordings with the press of the button and the pre-buffer allows the camera to include actions that happened before the wearer actually started the recording. The BWC is displayed in Figure 2.1.

2.2 Docking Station

The docking station allows the BWC to charge up its battery and also more importantly transfer the recordings done during the session to an SCU. Figure 2.2 shows an image of the docking station together with a BWC charging in it.

2.3 System Control Unit

The SCU temporarily stores recordings until they are transferred to either a VMS or an EMS. It also provides system administrators the possibility to manage the BWC users with specific configurations that alter certain aspects of the BWC. The SCU can be seen in Figure 2.3.

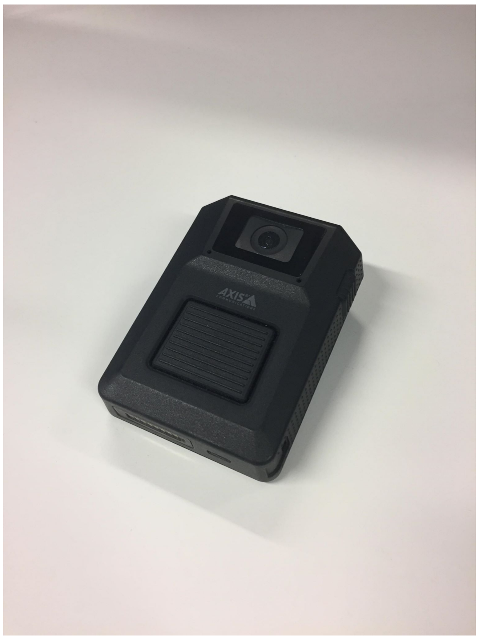


Figure 2.1: The Body Worn Camera (BWC)

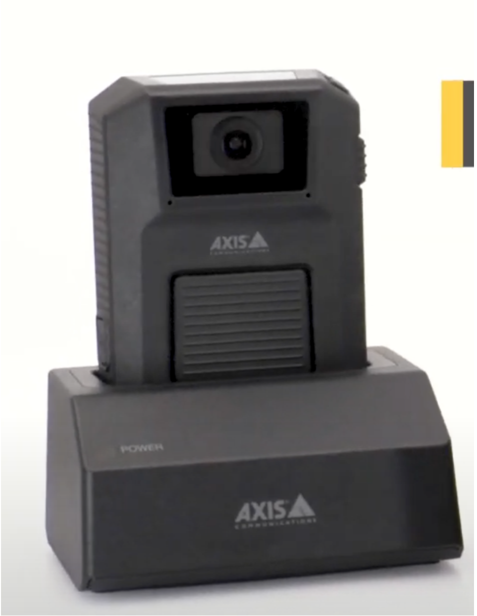


Figure 2.2: The BWC Docking station with a BWC in it

Figure 2.3: The System Control Unit (SCU)

2.4 Video Management System & Evidence Management System

This is the end destination for the video or potential evidence, which allows for secure storage in the cloud. If necessary the recordings here can be viewed by authorized personnel.

Chapter 3

Theory

3.1 Accelerometer

One of the sensors that has been used in this Master's thesis project is a triaxial accelerometer. A triaxial accelerometer returns an estimate of acceleration in the x, y and z axes from which velocity and placement of the sensor can be calculated. Accelerometers can be used as motion detectors as well as for position sensing [8]. This sensor is widely available today and are often integrated into smartphones, wearables and Internet of Things (IoT) device among others.

3.1.1 Acceleration

Acceleration is defined as the rate of which velocity changes over time [9]. It is measured in the unit m/s^2. In the real world, acceleration can be seen as a vector of three dimensions, x, y and z. As it is a vector is also has a magnitude and a direction, meaning that a change in acceleration is not limited to only the magnitude.

Gravity affects all objects on earth with a constant acceleration, pulling them towards the center of earth. The gravitational constant varies from 9.78 m/s^2 to 9.83 m/s^2 depending on where on earth it is measured [9]. See Figure 3.1 for an illustration of gravity in a 3-axis coordinate system. Throughout the rest of this report, the gravitational constant will have the value of $g = 9.82\ m/s^2$, unless stated otherwise.

Integrating acceleration over time will give the change in velocity over that time period. By further integrating the velocity will give the traveled distance or change in position.

3.1.2 Settings

Many accelerometers today have the option to change settings that will affect the sensors output data. One setting is the scale setting that affects the sensor total range and the sensor precision. The total range may start at ±2 g or lower, and increase with a factor of two, ±4 g, ±8 g and so forth. The trade-off in using a higher sensor range is that there is a compromise in the sensors precision. A reason for this is due to the fact that the same amount of bits for output data is used for all settings [10].

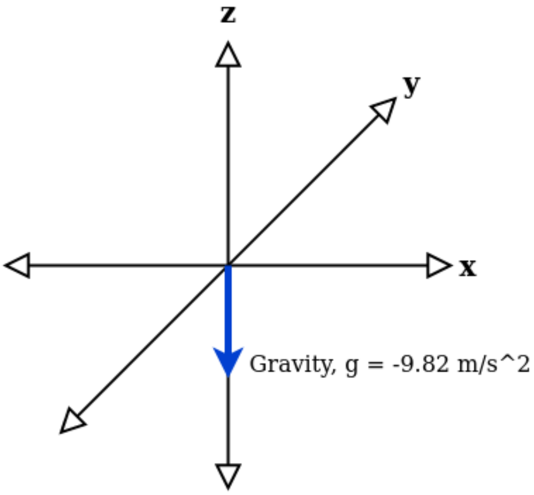


Figure 3.1: The gravity vector in a 3-axis coordinate system.

Another setting is the sensor data output rate, also known as the sample rate. The possible output data rate from the sensor is usually in fixed values. Examples are 50 Hz or lower, 100 Hz, 200 Hz or higher [10]. The impact of the sensor data output rate setting is explained more in detail in Section 3.3.

3.2 Gyroscope

The second sensor that is used in this Master's thesis project is the triaxial gyroscope. A gyroscope is a device that measures angular velocity. Just like the acceleromter, the gyroscope returns an estimate of angular velocity in the x, y and z axes. This type of sensor is also commonly seen together with an accelerometer sensor in the same type of devices as mentioned in Section 3.1. Some accelerometer and gyroscope sensors are integrated on the same chip with compact footprints as small as the ones used in some smartwatches, that is 2.5 mm x 3 mm x 0.83 mm [11].

3.2.1 Angular Velocity

Angular velocity is defined as the rate of which angular displacement or position changes over time [13]. It is measured in the unit rad/s or deg/s. In the same way as for acceleration, angular velocity can also be seen as a three dimensional vector with a magnitude and a direction. In Figure 3.2 the relation between the accelerometer axes x_acc, y_acc, z_acc and the gyroscope axes x_gyr, y_gyr, z_gyr can be seen. Using the right hand and putting the thumb in the positive direction of an accelerometer axis, the direction out from the fingers will tell the positive direction of the corresponding gyroscope axis.

Integrating angular velocity over a time period will give the change in angular distance or position. Calculating the derivative of angular velocity will result in the angular acceleration.

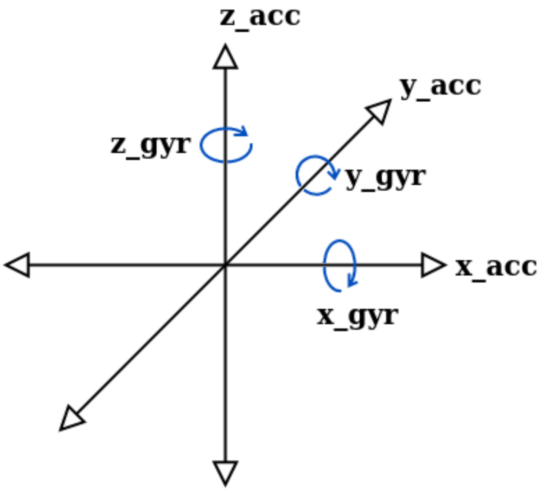


Figure 3.2: The axes of accelerometer and gyroscope combined.

3.2.2 Settings

In the same way as for the accelerometer, the gyroscope have a similar scale setting that will affect the sensors total range and precision. Example of some gyroscope settings are ±125 deg/s or lower, ±250 deg/s and ±500 deg/s or higher [12].

The gyroscope sensor also has a ability to output data at different rates. Depending on the use it can be good to use the same sample rates for both the accelerometer and the gyroscope, as then for each accelerometer sample there will be a corresponding sample for the gyroscope. However since both sensors may output timestamps for each sample they produce, the use of non matching sample rates is also possible without having a synchronization problem.

3.3 Sampling

Sampling is a term of measuring output data or some sort of values from a sensor. Using an accelerometer sample as an example, a single sample would be three decimal values, one value from each of the three axes x, y and z, e.g. x=3.22, y=1.45 and z=7.52.

3.3.1 Sampling Frequency

Sampling is done at a particular rate or a frequency, this is called sampling frequency. Sampling frequency is measured in Hz or samples/s. Example of a sampling frequency that has been used in a previous research is 120hz [14]. This means that 120 samples are generated per second.

3.3.2 Nyquist Rate

The Nyquist rate is the minimum rate that a time discrete signal $x(t)$ can be sampled at and still be reconstructed correctly [15]. This rate is 2 times or larger than the signal bandwidth. To be able to correctly reconstruct the signal of a person walking at a constant pace of 2 steps per second, i.e. 2 Hz whether that would be data from an accelerometer or gyroscope, the sample rate would need to be 4 Hz or higher. The effects of having a lower sample rate than the signal component with the highest frequency can often be observed in movies or video recordings of fast spinning objects, e.g. the helicopter rotary blades. The rotary blades will often look like they are moving very slowly, alternating spin directions or not moving at all.

It is also important to take the Nyquist rate into account when performing a Fourier transform, see Section 3.6.

3.4 Activities of Daily Living

Activities of Daily Living (ADL) is a term first coined by Dr. Sidney Katz and his team of professionals at the Benjamin Rose Hospital in Cleveland, Ohio in the late 1950s [16]. Index of ADLs is a measurement of how well a person can perform certain activities related to daily living. The ADLs may differ from person to person and the area of research. Some examples of ADLs are sitting, standing, walking, walking in stairs, laying down etc. The concept of ADL is commonly used in elderly health care and rehabilitation areas.

As will be seen in Section 3.11, this term is of high importance when it comes to preventing false alarms in regards to the different event detection algorithms that will be presented.

3.5 Vector Magnitude

The magnitude of a three dimensional vector can be calculated with the expression seen in equation 3.1. Since each value is squared to the power of two, the magnitude of a vector is always zero or positive. This also means that the directional information of a vector is lost after calculating its magnitude.

$$Magnitude = \sqrt{x^2 + y^2 + z^2} \tag{3.1}$$

3.5.1 Signal Magnitudes

Equation 3.1 can be applied to the output data of the triaxial accelerometer and triaxial gyroscope. The result of this can be seen in equation 3.2 and 3.3 respectively.

$$Magnitude_{Accelerometer} = \sqrt{x_{acc}^2 + y_{acc}^2 + z_{acc}^2} \tag{3.2}$$

$$Magnitude_{Gyroscope} = \sqrt{x_{gyr}^2 + y_{gyr}^2 + z_{gyr}^2} \tag{3.3}$$

3.6 Fourier Transform

The Fourier transform of a continuous time signal, $x(t)$ is defined as the frequency content of the signal, often denoted $X(f)$ [17]. The Fourier transform moves from the time domain to the frequency domain. The formula for calculating the Fourier transform of a time discrete signal is seen in equation 3.4. $\varphi(f)$ is the phase shift.

$$X(f) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt = |X(f)|e^{j\varphi(f)} \tag{3.4}$$

Analysing a signal in the frequency domain gives information about a signals frequency components, which can be useful for signals with repetitive patterns. ADLs, such as walking and running are examples of signals that contain repetitive patterns

To move from the frequency domain to the time domain an inverse Fourier transform may be performed, seen in equation 3.5.

$$x(t) = \mathcal{F}^{-1}\{X(f)\} = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df \tag{3.5}$$

In computer applications it is more common to work with discrete time signals. There is a discrete version of the Fourier transform, called the discrete-time Fourier transform (DTFT) but more commonly a more efficient and faster variant of this is used, called the discrete-time fast Fourier transform (DTFFT).

3.7 Filters

A filter is a tool in signal processing that can be used to separate or restore a signal. Signal separation can be used to remove some unwanted components or features of a signal. This could for example be to remove signals with some specific frequencies in the frequency domain. Signal restoration is used when a signal has been damaged in some way. The filters that have been used in this Master's thesis project are low-pass filters and high-pass filters [18] .

3.7.1 Low-Pass Filter

A low-pass filter (LPF) is used to filter out all high frequencies. A LPF allows signals to pass with a frequency lower than a chosen limit, called the cutoff frequency, and thus removes and blocks frequencies higher than this cutoff frequency. The cutoff frequency for determining if it is a high frequency signal is set by the user [19].

3.7.2 High-Pass Filter

A high-pass filter (HPF) is very similar to a LPF with the difference that it instead let all high frequencies pass and blocks all low frequencies. The cutoff frequency for determining if it is a low frequency signal is set by the user [19].

3.8 Time Shifting and subtracting

Time shifting is a method that is used in an attempt to detect repetitive patterns in a signal. An example of such repetitive signal can be a walk pattern or a run pattern. The detection is done by comparing the original signal with a time delayed version of itself. If the signal is periodic and the pattern of the signal is repetitive then the pattern will be detected by subtracting the two signals with each other. If the time Shifting works, the result can be that it is easier to identify the important part of the signal that would otherwise be hidden by the repetitive patterns.

3.9 Falling Index

Another metric that was used to improve the precision of the fall detection algorithm is the Falling Index (FI) [4]. The FI is calculated using the the formula shown in equation 3.6.

$$FI = \sqrt{\sum_{i=0}^{80}(x_i - x_{i-1})^2 + \sum_{i=0}^{80}(y_i - y_{i-1})^2 + \sum_{i=0}^{80}(z_i - z_{i-1})^2} \tag{3.6}$$

As previously mentioned, the output data from the BWC gave the accelerometer and gyroscope data in its three axis: x, y and z. These are the values that are used to calculate the FI. x in the formula stands for the x value, y stands for the y value and z stands for the z value from the sensors. The formula also includes summation signs and sums together all the samples within a fixed window. An example of such a window is 80 (corresponding to 0.4 s), meaning that 80 values of x, y and z are needed to calculate the FI. The FI can be seen as a measurement that determines how much the signal is changing during a time window. The higher the value of FI the faster the signal is changing in that time window.

3.10 Simple Moving Average

The simple moving average (SMA) is another metric that has been used to improve the fall detection algorithm. The formula for calculating a SMA can be found in equation 3.7. As can be seen, the calculation is done by adding together a certain amount of values and then dividing by the total amount of values. The total amount of values can be called the window size, n, and a large window size means a more accurate average value. An example of a window size is 100, meaning that 100 values are added together and then divided by 100 to get the SMA. The SMA is a good tool for making high spike less harmful for the signal since with a large enough window size the high spike will be evened out. The result of this is that it will give a much smoother final graph since it contains less noise disturbances.

$$X_{SMA} = \frac{1}{n} \sum_{i=0}^{n-1} x_i \tag{3.7}$$

3.11 Performance Measurements

When developing an algorithm used for binary event detection, i.e. the event happened or the event did not happen, there is a need for measurements that tell how well an algorithm performs. Some measurements that suit this area very well are the statistical terms of specificity and sensitivity, which are explained in Subsection 3.11.2 and 3.11.1.

3.11.1 Sensitivity

Sensitivity uses the terms true positives (TP) and false negatives (FN). The sensitivity measurement is calculated in according to equation 3.8 [21].

$$Sensitivity = \frac{nbr.TP}{nbr.TP + nbr.FN} \tag{3.8}$$

To explain this measurement in detail, let there be an algorithm, which is solely used to detect events of type A. When this algorithm is run on a data set containing only events of type A, the algorithm will tell how many events of type A it detected, that is the number of true positives. If the algorithm fail to detect some number of events of type A, then those are called false negatives, as they should have been classified as events of type A. The sensitivity measurement will be a number between 0 and 1, and is usually expressed in percent, where 0% means that no events were detected, and 100% means that all events were detected. 100% is the sought after result.

3.11.2 Specificity

Specificity uses the terms true negatives (TN) and false positives (FP) and is calculated in accordance with equation 3.9 [21].

$$Specificity = \frac{nbr.TN}{nbr.TN + nbr.FP} \tag{3.9}$$

Let there be the same algorithm as in Section 3.11.1. This time the algorithm is used on a data set containing only events of types that are not A, or in other words the complement of the event A, e.g. event B, C, D and so on. Events that are detected as A are considered false positives as there are no A events in the data set, whereas for data in the data set where nothing is detected is considered true negative. The specificity measurement is in the same way as the sensitivity measurement a value between 0 and 1, where 0% means that all events were falsely detected as A, whereas 100% means that no events were mistaken for the event A.

3.12 Cross-Validation

Cross-validation is a method used to validate a data set and for evaluating how well an algorithm can generalize when trained on some data set [22]. It begins by splitting the collected data set into a fix number of pieces, e.g. five pieces. For each iteration from one to five, one data set piece is selected as the validation set, and the remaining pieces are used for training the algorithm, i.e. finding good parameters, see Figure 3.3. Once the algorithm has been trained, validation is done on the validation data set piece by calculating the measurements of sensitivity and specificity. When all iterations are done, each data set piece, have taken turns to be the validation set, and five performance measurements are given, where each measurement contains sensitivity and a specificity measurements. Averages for sensitivity and specificity measurements are calculated and used to give an indication of how well a model will perform.

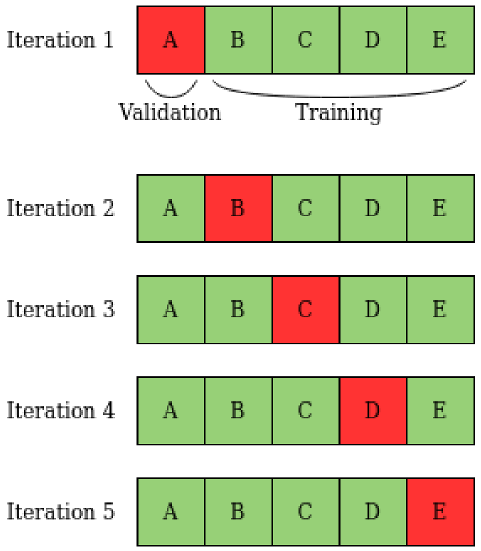


Figure 3.3: Cross validation by splitting the original data set in five equally sized pieces.

Even though five models were trained, these are not supposed to be used with new data, they are only meant to find how well the algorithm can generalize on data.

The benefits of cross-valdiation versus a regular single training data set and validation data set is that if only one split is performed, it may be an unlucky one, e.g. all the easy data is put in the training set and the difficult in the validation set or vice versa, whereas multiple splits will have a lesser risk of being affected by this.

Chapter 4

Method

This chapter will describe the different methods that has been used in this Master's thesis project. It will contain motivations and explanations on why a certain method was used. It will also include details and other decisions that was made during the development of the fall detection algorithms.

4.1 Data Collection Method

The majority of the data collection was performed by the two students doing this Master's thesis project. Each student performed an equal amount of recordings and the reason for this was so that the distribution would be even, i.e. no over-representation. Below is some general information about the two participants.

- Participant 1: Age: 23, Height: 174 cm , Weight: 68 kg, Male
- Participant 2: Age: 23, Height: 197 cm , Weight: 87 kg, Male

The difference between the two participants was quite drastic so the results offer a bit of diversification to a certain degree. Invitations were sent out to friends and other people but they were mostly declined. The reason for this was most likely because the idea of falling to the ground and risking getting hurt or injured did not seem to promising. Another reason was that social distancing was of high importance during the time this Master's thesis project took place. A higher number of participants could perhaps have offered a larger diversification and more evenly distributed data set.

The data collection or sampling was done over multiple sessions spread out over the Master's thesis project. First the sensor settings used are explained, then some comments on an Android application that was built, and finally the validation of the data collection settings and method is described.

4.1.1 Sensor settings

As mentioned in Section 3.3, the accelerometer and gyroscope can have different sensor settings that affect the sensor range, precision, and also output data at different output rates. The settings used for this Master's thesis project, are explained in Subsections Sampling Frequency, Accelerometer Scale and Gyroscope Scale.

Sampling Frequency

Looking at previous work, G. Brown and M. Eklund [3] used a sample frequency of 80 Hz, while Kangas et al. [4] started with a sample frequency of 400 Hz, and let the samples be re sampled and median filtered with a window of 3 samples, bringing the sample frequency down to 50 Hz, as a way to reduce the amount of data and the noise.

Since storage or performance was no problem it was initially decided to capture samples with a high sample frequency of 500 Hz. This would leave the possibility open to at a later stage perform down sampling or median filtering. However when analyzing the output data captured with 500 Hz it was discovered that due to technical difficulties with sample timestamps missing out, it was not appropriate to keep the sample rate at 500 Hz. As a consequence of this it was decided to lower the sampling frequency to 200 Hz to resolve this issue. This proved however to not be a major problem since close to none information was lost. The majority of the sensor recordings were therefore recorded with a sample frequency of 200 Hz.

Accelerometer Scale

Huynh et al. [1]-[2] uses a wide accelerometer scale setting of ±16 g, while G. Brown and M. Eklund [3] uses a slightly lower setting of ±10 g. Kangas et al. [4] differentiates from the other sources by sampling with the setting ±12 g and then restricting the amplitude to ±2 g and ±3 g, since ±12 g was said not to provided much more information than ±3 g. ±2 g however was found to be insufficient for the algorithms that they presented.

During the Master's thesis project, the majority of the sensor recordings were done with an accelerometer scale value corresponding to the accelerometer sensor range of ±16 g, leaving the possibility open to simulate lower levels at the cost of accuracy. The reason for this was because some of the falls and other activities gave very high values and without ±16 g the range would have been too low. The other range options were ±2 g, ±4 g and ±8 g.

Gyroscope Scale

Once again Huynh et al. [1]-[2], as with the accelerometer, use a wide gyroscope setting of ±2000 dps, indicating that information gained with the wider range setting is worth more than the loss in sensor accuracy with a lower setting.

During the Master's thesis project, the majority of the sensor recordings were done with a gyroscope scale value corresponding to the gyroscope sensor range of ±2000 dps or ±34.9 rad/s. The reason for this is the same as with the accelerometer. Some of the falls and other activities ave very high values and without ±2000 dps the range would have been too low. The other range options were ±250 dps, ±500 dps and ±1000 dps.

4.1.2 Android

At an early stage of the Master's thesis work it was decided that smartphones could be useful help. The reason for this was because the majority of smartphones today are equipped with both accelerometer and gyroscope sensors. These smartphone sensors could be used to record sensor output data and compare this with the output data from the BWC. The comparison of sensor data will serve as a method to ensure that the sampled data of the BWC is correct for the different sensor settings. It will also give an indication of the quality of the sampled data, i.e. how well does the sensors perform. Therefore a decision was made to develop a simple and usable application for this exact purpose.

Android [23], was chosen as the target platform due to the availability of Android smartphones and the students previous knowledge with developing Android applications.

Sensors Application

The application developed was given the name Sensors. It was developed using Android Studio [24], and written in Java code.

In Figure 4.1 a screenshot of the sensor application can be seen. The application consists of a single view. From top to bottom it shows, acceleration in all three axis, x, y, z from the accelerometer, angular velocity in all three axis, z, y, z from the gyroscope, a variable target sample rate from 1-500 Hz, the actual achieved sample rate for the two sensors independently, time elapsed if saving samples to a file, and the option to start a recording and save it to a local file. Finally the sensors can be stopped or started manually, which is useful if the user wants to record the sensor output data while the screen is off.

It may be noted that in Figure 4.1 the sample rate is set to 201 Hz, however in reality only 100 Hz is achieved, this is due to a limitation of running the application on an Android virtual device. When running the application on real life Android smartphones a sample rate of 500 Hz was achievable.

4.1.3 Validating the BWC Data Collection Settings

To validate that the sensors were actually using the intended settings, a few simple tests were done.

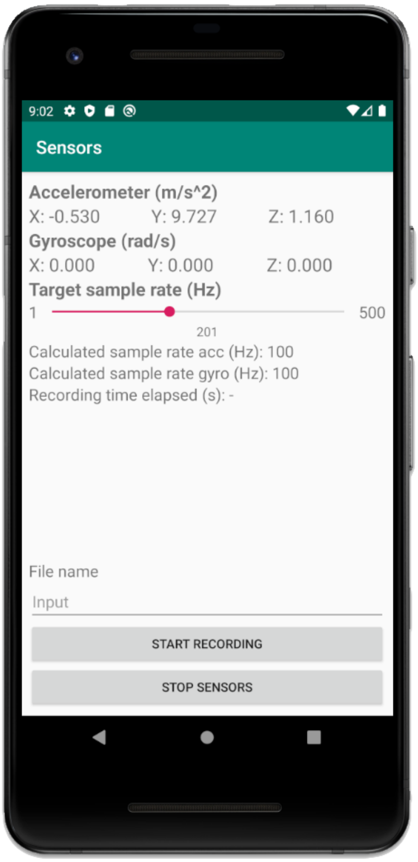
Sampling Frequency

Validation of sample frequency is done by firstly calculating the expected period time of samples, T, when sampling at F = 200 Hz, using equation 4.1.

$$T = 1/F = 1/200 = 0.005\ s \tag{4.1}$$

The period time is the time between samples. Each sample will generate a sensor output timestamp, t_i, so calculating the average period time, $T_{average}$, over all the samples, N, can be done by using equation 4.2, for a recording. This means that errors may be detected and the setting can be validated.

$$T_{average} = \frac{1}{N} \sum_{i=0}^{N-2} t_{i+1} - t_i \tag{4.2}$$



Accelerometer (m/s^2)
X: -0.530 Y: 9.727 Z: 1.160
Gyroscope (rad/s)
X: 0.000 Y: 0.000 Z: 0.000
Target sample rate (Hz)
1 500
201
Calculated sample rate acc (Hz): 100
Calculated sample rate gyro (Hz): 100
Recording time elapsed (s): -

File name
Input

Figure 4.1: The Android application Sensors, running on a virtual device, highlighting the application's capabilities.

Accelerometer Scale

The accelerometer scale setting is tested by holding the BWC with one hand and making a fast hand movement, like a throw movement, and registering the maxed out value for each axis. If the max values can reach 16 g, then this settings is validated.

Gyroscope Scale

The gyroscope scale setting is tested by holding the BWC with two hands and making fast tilting movements forwards and backwards while keeping the hands somewhat stationary and registering the maxed out values. If the max values corresponds to ±2000 dps or ±34.9 rad/s in each axis then this settings is validated.

4.1.4 Validating the Data Collection Method

When the data collection settings had been validated, the next step was to look at the data and make sure that it was correct. This was done by starting a recording to get sensor data and then plotting it onto a graph.

Shakings

Initially recordings were done by simply holding the BWC with one hand, and making some simple and recognizable movements, until it was clear that the sampling method worked as intended. Some recordings were done using a clip that would be attached to the participants clothing, e.g. at the top of a T-shirt. Figure 4.2 shows an image of the clip attachment. After beginning to analyze the data it became clear that there were issues with unintended shaking of the BWC. This was because the clip attachment was not very stable and also the participants clothing was a bit loose. Figure 4.3 shows a setup using the clip attachment. The result of this became that there was a lot of shaking and high signals when performing a ADL such as walking and running. This shaking damages the data because the data no longer only contains the real movement of the subject. Tests with different attachments were made in an attempt to deal with the shaking. A magnetic attachment, see Figure 4.4, had the same issues as the previous attachment. It was found that the best attachment was by using a regular belt by locking the BWC into place against the subject chest. This setup by using a belt to attach the BWC can be seen in Figure 4.5. A comparison of jogging forwards, turning around and jogging back to the starting position with a clip versus with a belt can be seen in Figure 4.6.

As can be seen in Figure 4.6, the difference in magnitude for the accelerometer are at peak levels 300% (3.26 g vs 9.77 g at t = 17.3 s) and for the gyroscope around 400% (2.42 rad/s vs 10.15 rad/s at t = 5.12 s), which is a significant difference.

BWC and Smartphone

To verify that there were no errors with the sensors in the BWC or the way data was retrieved from it, comparisons with the sensor outputs of a smartphone

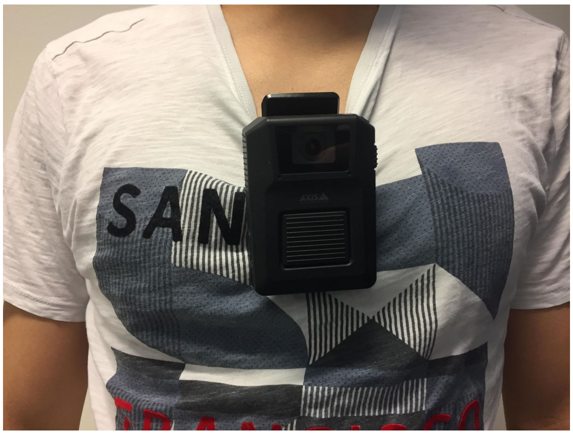


Figure 4.2: A participant is using the clip attachment for a recording.

Figure 4.3: The clip attachment used for some recordings. When used, the clip attachment is attached to the clothes of the participant together with the BWC.

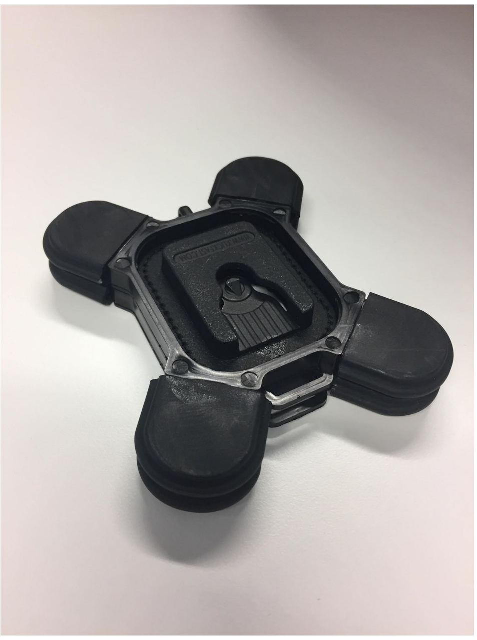


Figure 4.4: The magnet attachment used for some recordings. When used, the magnet attachment is attached to the clothes of the participant together with the BWC.

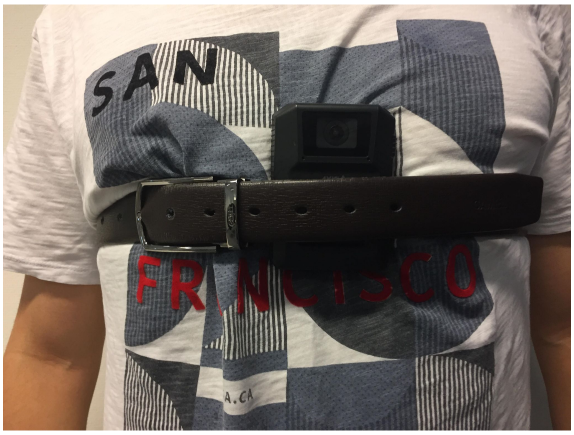
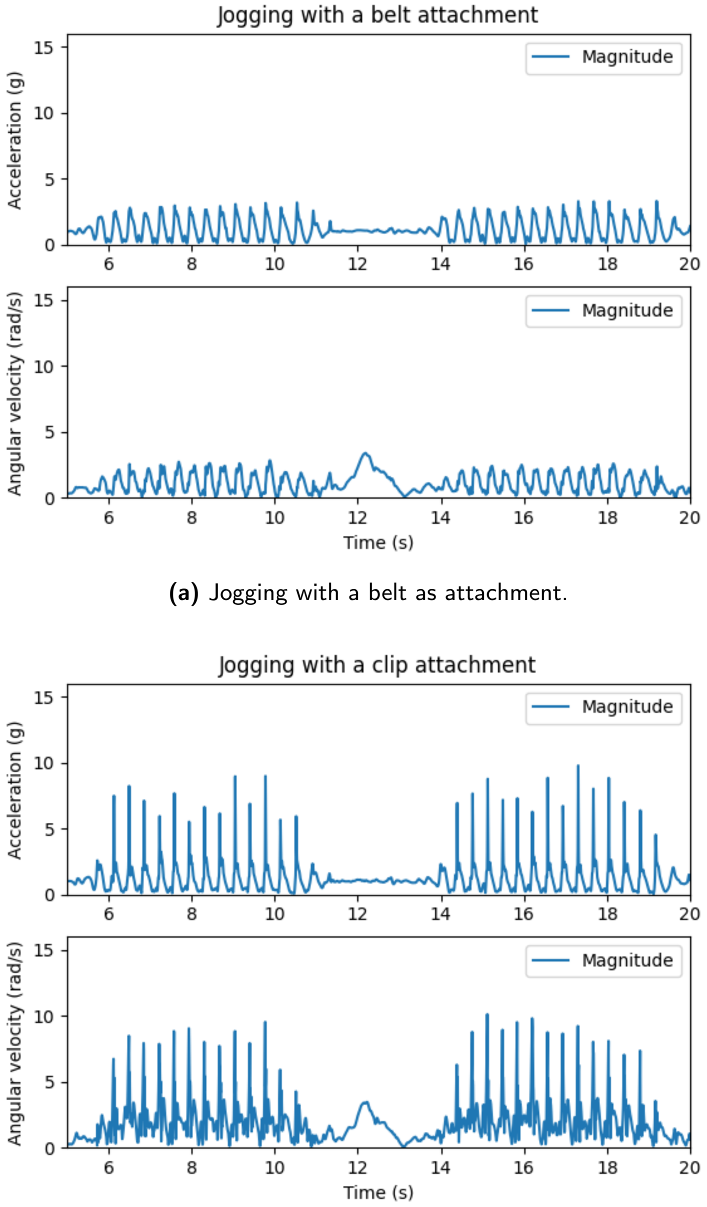


Figure 4.5: A participant is using the belt for a recording.



(a) Jogging with a belt as attachment.

(b) Jogging with a clip as attachment.

Figure 4.6: A comparison of using a belt as attachment to the (upper plot) versus using a clip (lower plot).

was done. In Figure 4.7 the comparison of the magnitude of some simple hand movements can be seen and visually comparing them shows that they perform very similarly. The recordings were made in parallel.

Two BWCs

To verify that there were no errors on the specific device that was used for data recording that could impact the collected data negatively, comparisons with another BWC was made. In Figure 4.8 the comparison of the magnitude from walking forwards, turn around and walking back to the starting position can be seen. By just visually comparing them, as with the previous comparison, shows that they perform very similarly.

4.2 Collected Data

During the Master's thesis work various interesting sensor data was recorded. The gathered data was grouped into four categories, being Activities of Daily Living, Fall, Push and External Daily BWC Movements. In total 100 ADLs, 79 falls, 46 pushes and 115 external daily BWC movements were recorded. A small database was created to store all the different files in a organized and accessible way.

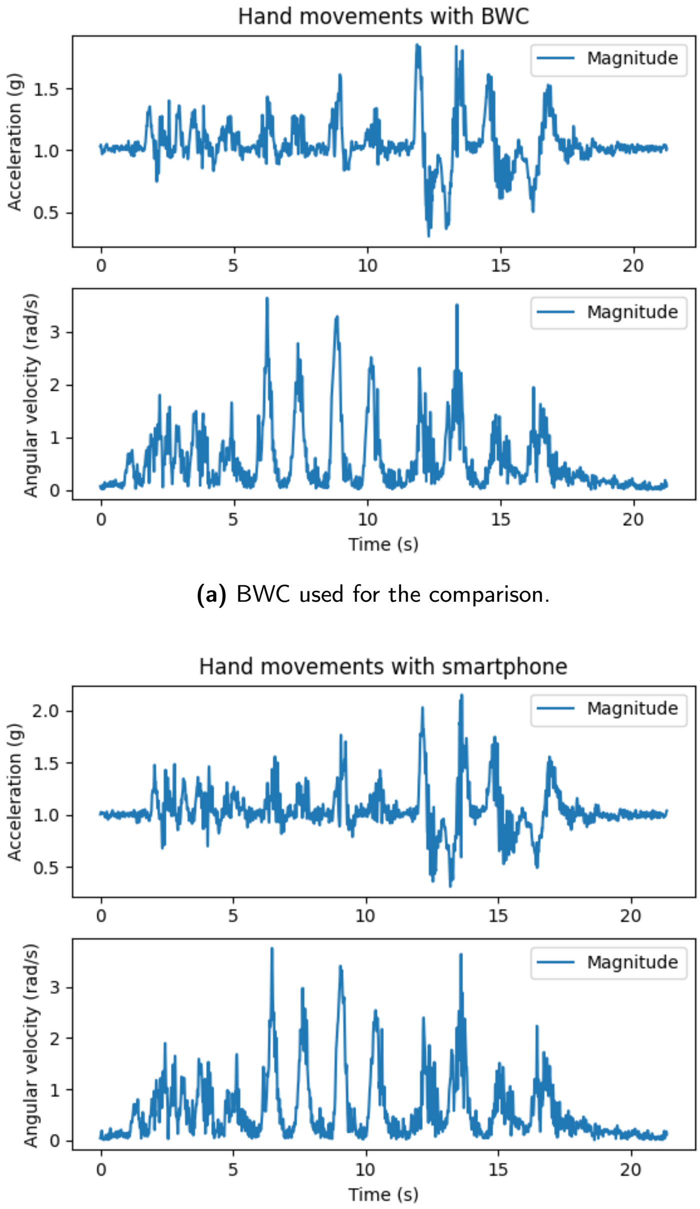
4.2.1 Activities of Daily Living

In Table 4.1, all the recordings of daily activity that was done in this Master's thesis work are listed, and a brief description for each activity is given. Some of the activities listed in the table are inspired from the sources in the previous work section 1.7, however since most of these articles were focused on elder care, it was decided to add some more scenarios that seemed relevant to the intended users of this Master's thesis work. Having a large list of daily activities allows for determining what kinds of daily activities that are possible to distinguish from falling and which ones that may sometimes lead to false alarms due to them having similar signal characteristics.

In Figure 4.9 the magnitude signals from a person walking in stairs can be seen, and in Figure 4.10 the magnitude signals of a person sitting down and standing up fast can be seen.

4.2.2 Fall

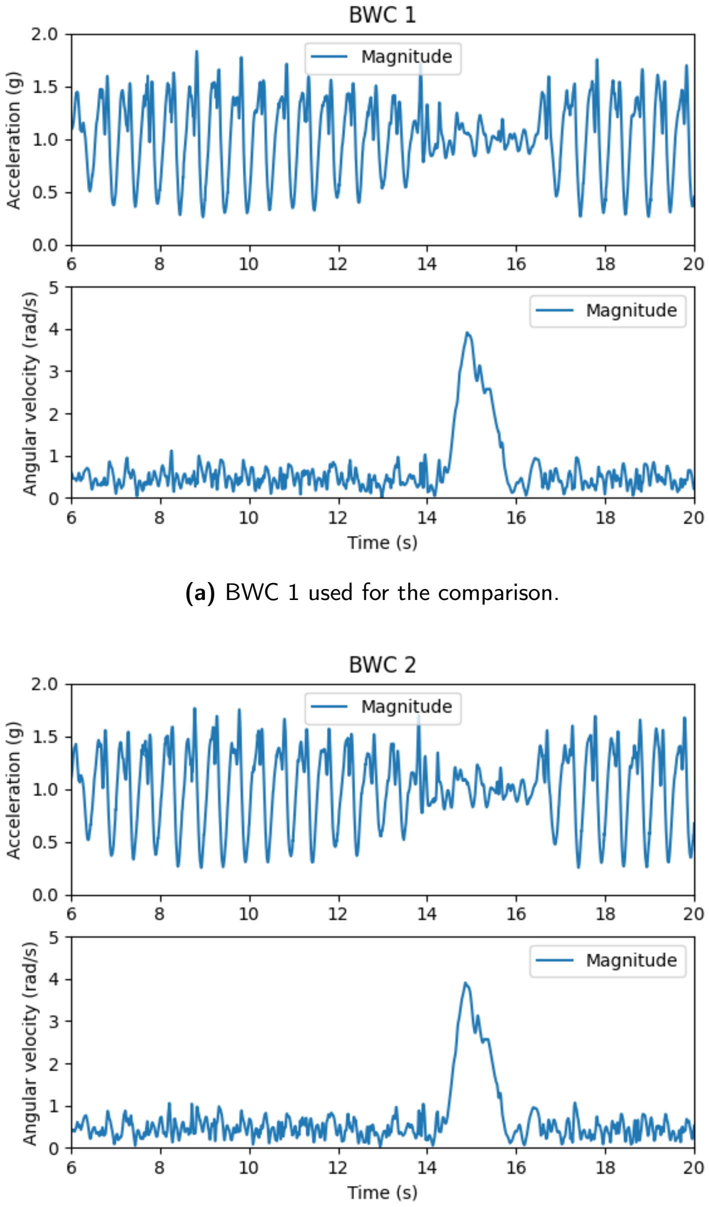
It was decided that falls would be recorded and analyzed in four different directions. These directions were backwards, forwards, fall to the right and fall to the left. A list of these fall is shown in Table 4.2. The reason for including all four directions was because this would cover a wider variety of possible ways to fall. Most of the falls recorded were from a static position meaning that there was no movement before the fall. The reason for this was because the difference between a static and a moving fall was not very big. It was also easier for the participants to make each fall more consistent.



(a) BWC used for the comparison.

(b) Smartphone used for the comparison.

Figure 4.7: A comparison between the sensors from a BWC and a smartphone showing the magnitude of the signals generated when performing some simple hand movements.



(a) BWC 1 used for the comparison.

(b) BWC 2 used for the comparison.

Figure 4.8: A comparison of the sensors from two BWCs showing the magnitude of the signals generated when walking.

Table 4.1: Summarizes the sensor data recordings done on ADL.

Activities of daily living (ADL)		
Name of recording	**Description**	**Nbr. of files**
Normal walk	Normal walking pace forwards, turn around, walk back to start.	15
Fast walk with turns	Fast walking pace around two tables in an 8-shape.	2
Jogging	Jogging pace, same route as normal walk.	4
Running	Subject max running pace, 30 m, turn around, 30 m back to start.	13
Normal walk in stairs	Up and down, stair lengths of 8 - 20 steps.	14
Fast walk in stairs	Up and down, stair lengths of 8 - 20 steps.	20
Jump down	Jump down from three stair steps.	15
Sit down & stand up	Chair seat is approximately 30 cm from ground.	6
Tie shoes	Kneel down, tie shoes, stand up (varying speed).	4
Jump	Vertical jump.	6
Elevator	6 floors up and down.	1

Table 4.2: Summarizes the sensor data recordings done on falling.

Falling		
Name of recording	**Description**	**Nbr. of files**
Fall backwards	Fall on two mattresses with a total thickness of approximately 15 cm.	19
Fall forwards	Same as above.	20
Fall to the right	Same as above.	20
Fall to the left	Same as above.	20

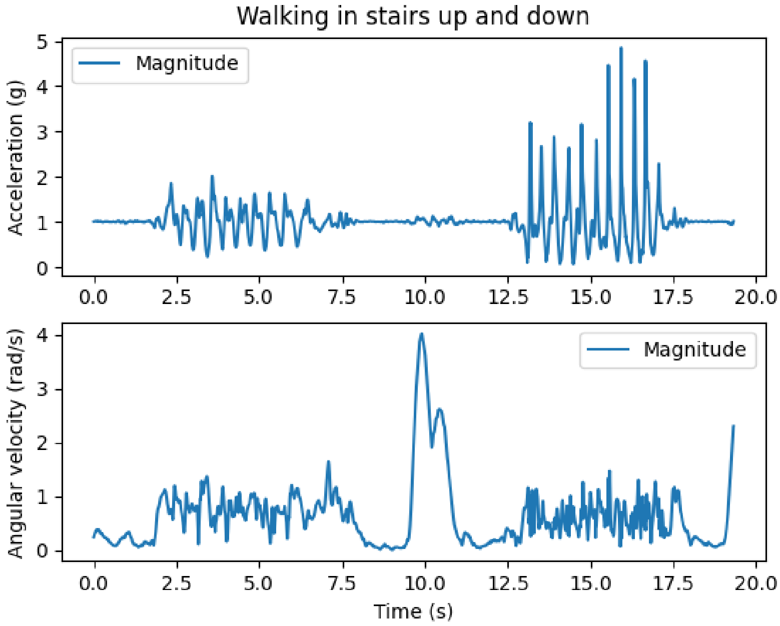


Figure 4.9: The magnitude signals of the accelerometer and gyroscope from a person walking up the stairs, turning around, and walking back down.

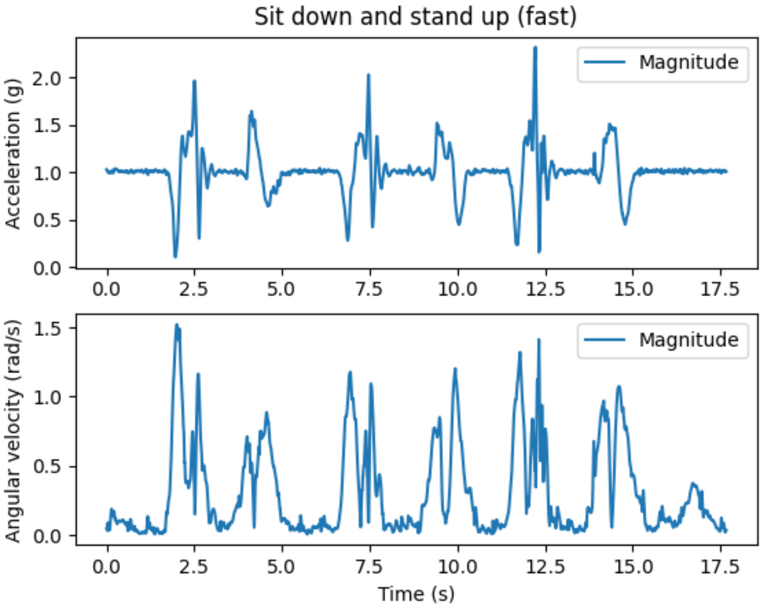


Figure 4.10: The magnitude signals of the accelerometer and gyroscope from a person sitting down and standing up fast, repeated three times consecutively.

As mentioned in the Table 4.2, the fall was done on two mattresses with a total thickness of approximately 15 cm, see Figure 4.11. This most likely made each fall a bit more soft compared to a real fall but each participant definitely felt the impact when landing on the ground.

The magnitude signals of a fall can be seen in Figure 4.12.

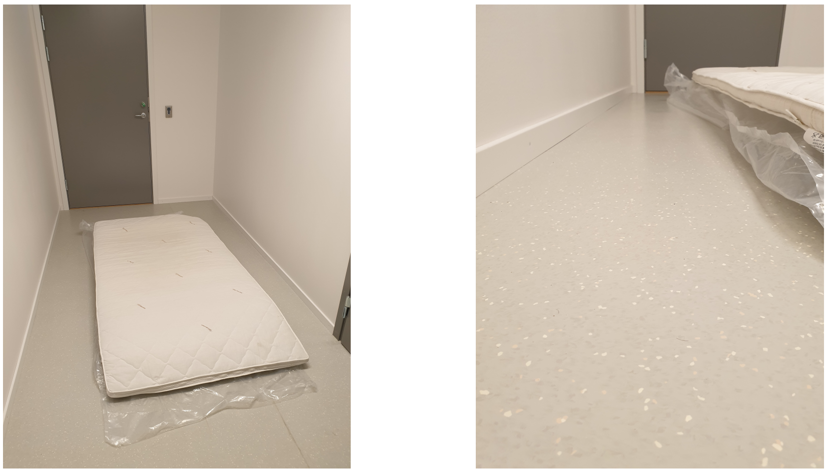
4.2.3 Push

Similarly to the recording of falling, pushes were performed in the same four directions and recorded, see Table 4.3. The reason for this was, just like above, to cover a wider range of possible pushes. Since push is a much easier case to record compared to a fall, the recordings here were most likely accurate to a push that can happen in real life.

The magnitude signals of a push can be seen in Figure 4.13.

4.2.4 External Daily BWC Movements

Two not so uncommon things that the BWC users do are detaching the camera from their clothing and putting it away, either for docking or just placing it on a table. The second thing is reattaching it again, all while the device is operating. These movements completely change the conditions of what is being recorded. Instead of recording the movements of the camera wearers body, the BWC will



(a) The top view of the two mattresses. **(b)** The side view of the two mattresses.

Figure 4.11: The setup used when recording falls.

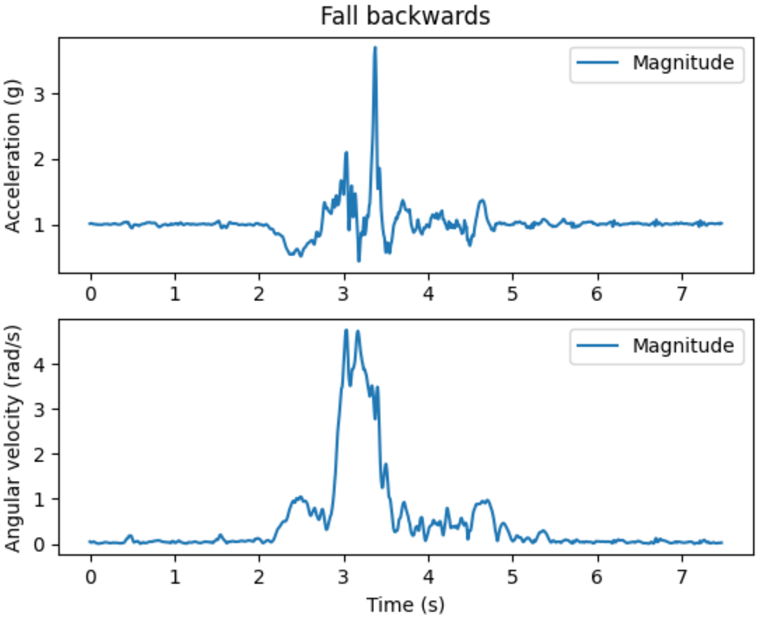


Figure 4.12: The magnitude signals of the accelerometer and gyroscope from a backwards fall.

Table 4.3: Summarizes the sensor data recordings done on pushing.

Push		
Name of recording	**Description**	**Nbr. of files**
Back push	Push the subject on the back.	20
Front push	Push the subject on the chest.	8
Push from right	Push the subject on the right shoulder.	9
Push from left	Push the subject on the left shoulder.	9

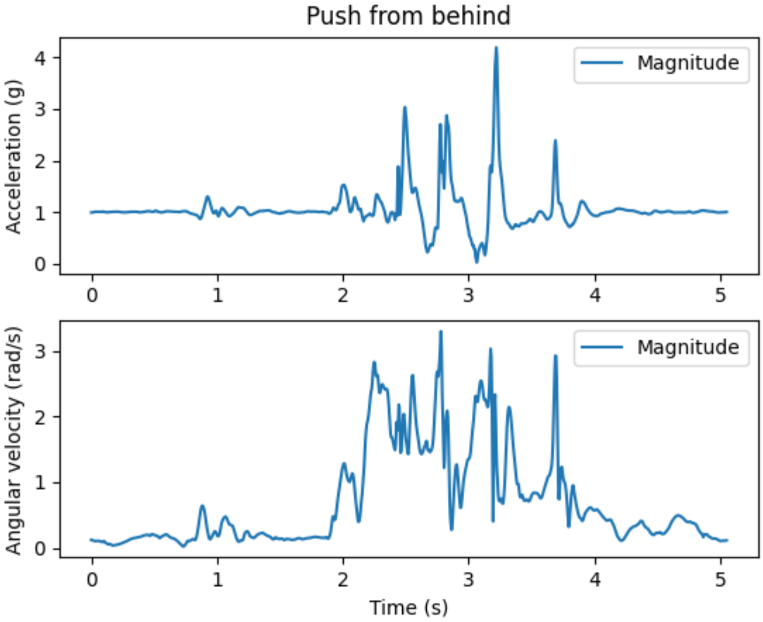


Figure 4.13: The magnitude signals of the accelerometer and gyroscope from a person being pushed from behind.

instead record some sort of hand movements. To prevent these movement patterns from falsely being classified as falls, some recordings of these movements were done, see Table 4.4. These movements were given the name external daily BWC movements. The reason for including this topic is because one of the research question in this Master's thesis project is: " How accurate can the algorithm recognise a fall". In order to answer this question, it seemed relevant to explore this kind of movements since they were a risk for triggering a false detection.

Table 4.4: Summarizes the sensor data recordings done on external daily BWC movements.

External daily BWC movements		
Name of recording	**Description**	**Nbr. of files**
Normal/fast put BWC on table	Move BWC from chest to table.	20
Soft put BWC on table	Move BWC from chest to table gently.	20
Take BWC from table and attach to clip	Clip is attached to subject.	20
Take BWC from table and attach to clothing	Clip is attached to BWC.	20
Docking the BWC	Put BWC in the docking station.	20

The magnitude signals of a external daily BWC movement can be seen in Figure 4.14.

4.3 Python

The programming language Python was used extensively throughout the project for validating collected data, analyzing collected data, implementing algorithms and evaluating the performance of the algorithms on the collected data. Python was chosen as it meets the scientific requirements of the Master's thesis work, it is open source software, freely available [25] and due to the thesis workers previous knowledge in developing Python applications. To perform all of what is enumerated above, mainly three Python packages were used, NumPy, SciPy and matplotlib, which are explained more in detail in the three subsections below.

4.3.1 NumPy

The NumPy package provides the implementation and functionality for 1- to N-dimensional arrays as well as basic mathematical operations to manipulate the data stored in these arrays [26]. The reason for using the Numpy package was because it made the code more efficient and it was also more convenient to work with 1- to N-dimensional arrays.

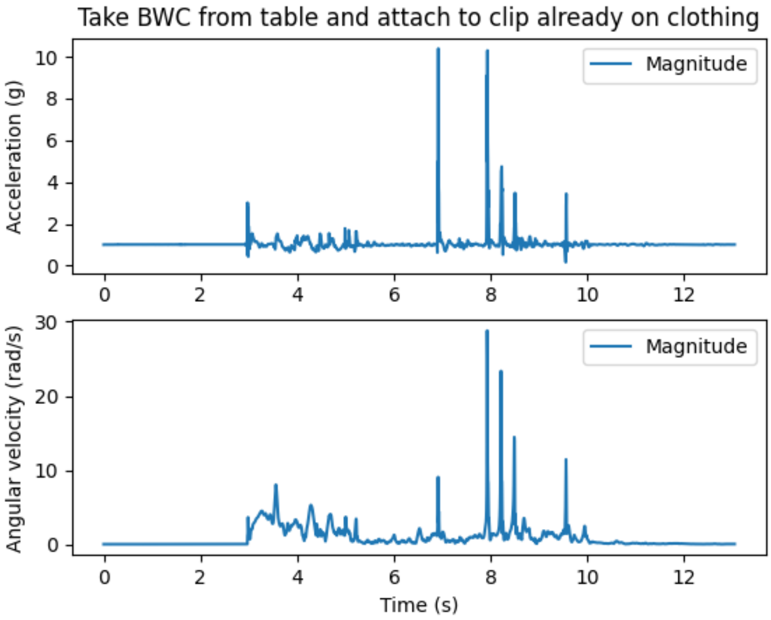


Figure 4.14: The magnitude signals of the accelerometer and gyroscope from a person taking the BWC from a table and attaching it to the clip which is already attached to the person's clothing. The gyroscope spikes around 8 s in, occurs due to the attachment has to be done upside down and then turned 180 degrees in snapping motions.

4.3.2 SciPy

The SciPy package provides among other features useful functions within the area of signal processing. Examples of this are low-pass and high-pass filters and fourier transform [27]. The reason for not creating and implementing the functions from the beginning was because it was much more convenient and easier to use the SciPy package. This also guaranteed that there were no errors in the functions which could have been the case if they were implemented from the scratch.

4.3.3 matplotlib - pyplot

The Matplotlib package is used for visualization of data through 2D and 3D graphical plots [28]. In this Master's thesis work report, the majority of the plotted data figures come from this library. Examples of usage are plotting the three different axes of the accelerometer or gyroscope, x, y and z over time in a 2D figure.

4.3.4 Testing Algorithms and Functions

The testing of each algorithm and function was done by creating a graphical user interface (GUI) and then adding a button for each algorithm or function. The GUI was created using Python and is shown in Figure 4.15. This GUI proved to be very useful since it saved a lot of time when comparing the different algorithms against each other. It was also very convenient and easy to add new functions and remove old ones.

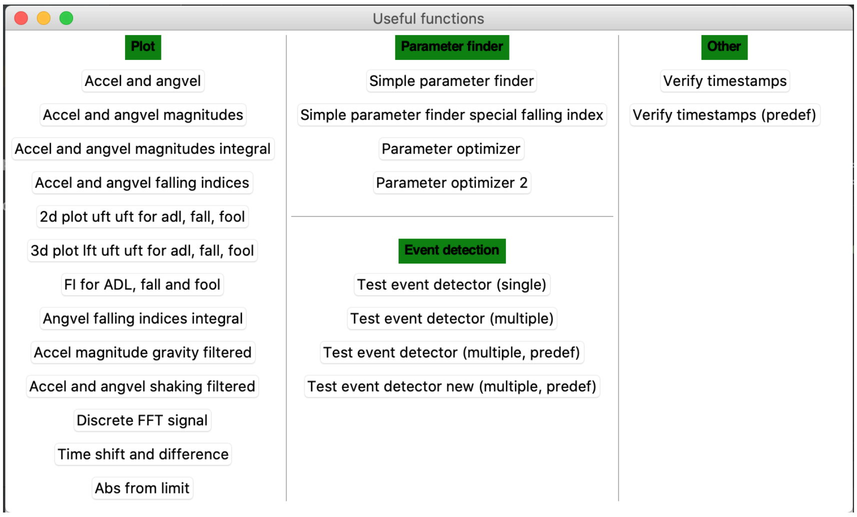

Useful functions
Plot
Accel and angvel
Accel and angvel magnitudes
Accel and angvel magnitudes integral
Accel and angvel falling indices
2d plot uft uft for adl, fall, fool
3d plot lft uft uft for adl, fall, fool
FI for ADL, fall and fool
Angvel falling indices integral
Accel magnitude gravity filtered
Accel and angvel shaking filtered
Discrete FFT signal
Time shift and difference
Abs from limit
Parameter finder
Simple parameter finder
Simple parameter finder special falling index
Parameter optimizer
Parameter optimizer 2
Event detection
Test event detector (single)
Test event detector (multiple)
Test event detector (multiple, predef)
Test event detector new (multiple, predef)
Other
Verify timestamps
Verify timestamps (predef)

Figure 4.15: Graphical user interface (GUI) used to test each function and algorithm

The GUI is divided into four different sections: plot, parameter finder, event detection and other.

The "plot" section lists all the function that are used to plot graphs. It could for instance be to plot the magnitude of a forward fall.

The "parameter finder" section contains the functions that are used to find different thresholds for the algorithms.

The "event detection" section contains functions that are used to calculate specificity and sensitivity of each algorithm.

Lastly, the "other" category contains functions that might be useful but was not part of any algorithm or function.

4.4 Testing Previous Fall Detection Algorithms

The literature study done before the start of this Master's thesis project gave a lot of knowledge about the general concept of fall detection. It gave a good idea on how the development of the fall detection algorithm could be done but it also showed the different algorithms and results of the previous studies and what they had found. This gave the opportunity to verify the precision and accuracy of the developed algorithm in this Master's thesis project. By comparing the result of this Master's thesis project against previous studies it gave a good idea on the quality of the fall detection algorithm.

A couple of different tests with previous algorithms were done to see their overall performance. It has to however be taken into account that some of these studies only focused on falls and did not focus on a wide range of ADLs. It was noticed that fast movements such as running and jumping could be detected as a fall and some of the previous researched did not test this, unlike what has been done in this Master's thesis project. It was also seen that external BWC movements had the chance of triggering the fall detection algorithm. The tests was done by using the different parameters together with the algorithm that was found in the previous research against the collected data set from this Master's thesis project.

4.5 Fall Detection Algorithm Development

The development of the fall detection algorithm has been an iterative process, meaning that new functionality was continuously added. The decision whether a new function was going to be kept or not was decided based on the sensitivity and specificity values it received upon testing. It was easy to keep track of previous result or to retest an old algorithm with the help of the GUI shown in Figure 4.15. The reason for choosing an iterative process was because the algorithm development required a lot of repetitive testing e.g. when changing parameters and adding new features, so it seemed only natural to go with this process.

4.5.1 Algorithm Development using Thresholds

The reason for exploring and testing thresholds was because a lot of previous research [1], [2], [3], [4] had used thresholds as part of their core algorithm and also received very promising results, see Section 1.7, Previous Work. Another reason was because the idea also seemed easy to understand and implement.

The main idea behind thresholds is that a value is determined to be set as a threshold. If that value is reached it will be triggered and noticed. It is also possible to have multiple thresholds and each threshold could represent different things. Example of such thresholds that have been used in this Master's thesis project are explained below. By using thresholds it opened up many opportunities for testing and experimenting. The process of finding the best mix of threshold was an iterative process as previously mentioned.

Magnitude Thresholds

The accelerometer and gyroscope produced sensor data in its three axis x, y and z which can seen as a vector as previously mentioned. The magnitude is calculated by combining each of these values using equation 3.1, i.e. equation 3.2 and 3.3 for the accelerometer and gyroscope respectively. The reason for using the magnitude is because the directional information of the vector is lost after calculating its magnitude. This is a good thing since it allows the BWC to be placed in an arbitrary position, e.g. it does not matter if the camera is turned up or down or slightly tilted since the magnitude will be the same. This means that the BWC position during the data collection did not have to be identical between each recording.

The magnitudes thresholds that have been used in this Master's thesis project is a lower fall threshold (LFT) for the accelerometer, an upper fall threshold (UFT) for the accelerometer and an UFT for the gyroscope. The reason for choosing these particular thresholds was because other previous research [1], [2], [3], [4] have used these magnitude thresholds. There was also some testing done that lead to the conclusion that these thresholds were a good fit for fall detection. The three thresholds can be seen as the red horizontal lines in Figure 4.16.

Another kind of threshold that has been used was found after studying different graphs of both falls, ADLs and external daily BWC movements. It was found that all the falls were under a certain threshold in both the accelerometer magnitude and gyroscope magnitude. With the help of this knowledge it was decided to add a maximum threshold for removing too large magnitude signals, these large signals were often present in external daily BWC movements. An example of such a signal can be found in Figure 4.14. The reason for adding these additional thresholds was because it was an easy way to remove and distinguish between falls and large magnitude signals. It was also easy to implement these additions to the algorithm. A graph including all the thresholds can be found in Figure 4.17.

A sliding time window had also to be selected for detecting a fall. If all the three thresholds (LFT, UFT and UFT) were exceeded within this time window that meant that a fall had occurred. An example of such time window could be 1.5 s, and a shorter one of 0.5 s can be seen in Figure 4.16 as the yellow area. So each time window required a separate search. A fall happens very fast but can also vary slightly in duration so it was important to make sure that the fall window was of appropriate length. The size of the time window was determined through testing and studying the results.

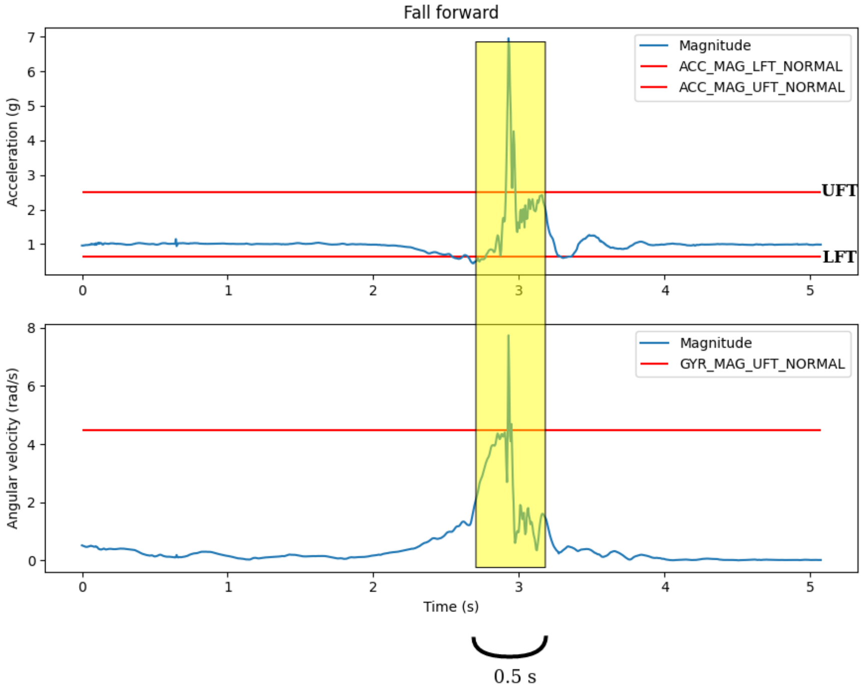


Figure 4.16: A simple threshold function using three thresholds LFT_acc, UFT_acc and UFT_gyr and a time window of 0.5 s, as presented by Huynh et al. [1]-[2], visualized on a forward fall.

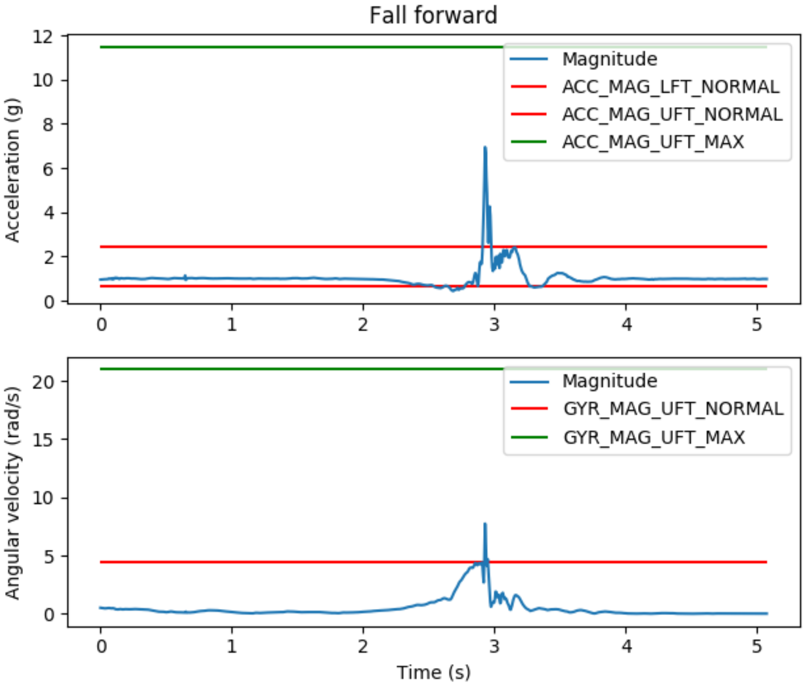


Figure 4.17: All the magnitude thresholds.

Magnitude Thresholds with Order

After plotting a couple of graphs and then studying and examining each one, it was discovered that all the falls shared a similar pattern. Figure 4.12 and 4.16 shows examples of fall patterns, when falling in two different directions. It can be seen that the plot dips down before rising up to its top. Because of this, another requirement was implemented that checked the order in which each threshold was exceeded. The order for this was that the accelerometer LFT had to first be exceeded before both UFTs for it to be classified as a fall.

A explanation behind the appearance of a fall signal is because when a person falls to the ground the person is in a state of free fall in the direction of the gravity, true free fall occurs when the accelerometer shows 0 g. The duration a person is in this state depends on the height of the fall. Upon hitting the ground, there is a quick change in the direction of the acceleration, to stop the accumulated fall velocity in a short period of time, which means that the graph will spike up. The aftermath of a fall is usually that the person is unable to move for a short amount of time and that explains the flat line. The gyroscope shows there is a lot of change in angular velocity which is another sign of a fall.

Falling Index Thresholds

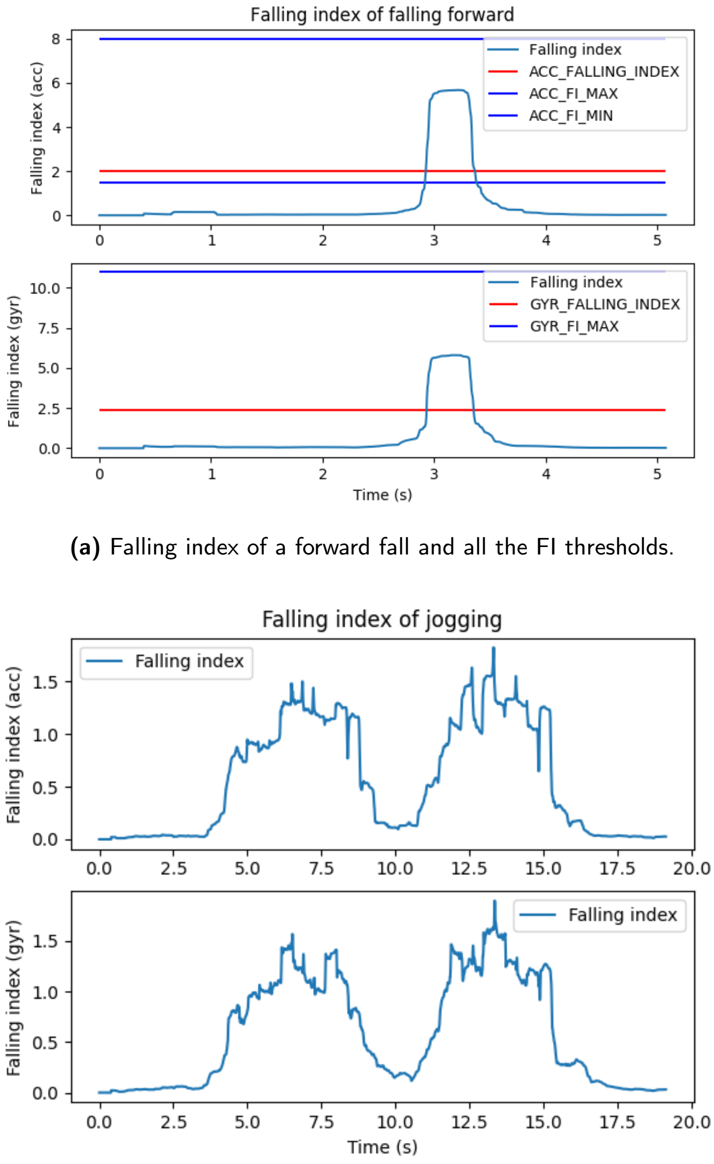
As previously mentioned in the Theory chapter, FI is a measurement that determines how much a signal is changing during a time window. Since each time window produced a single value it meant that the falling index could be plotted. Figure 4.18 shows an example of a falling index plot of a fall and a short jog. The graph showing the fall also includes all the FI thresholds.

After studying and examining different FI graphs of falls, ADLs and external BWC movements it was found that falls happened within a certain range of FI values. For this reason, just like with the maximum and minimum magnitude thresholds, a minimum FI threshold for accelerometer, a maximum FI threshold for accelerometer and a maximum FI threshold for gyroscope were introduced.

A fall is in general an action that is very fast compared to an ADL such as jogging or sitting down. This means that a high value will be obtained when calculating the FI during a fall. After inspecting different falling graphs the discovery was made that this does necessary not mean that the accelerometer and gyroscope will produce high values during a fall. This meant that these falls would not be detected if compared to the magnitude thresholds previously mentioned since their values were too low. This issue could however be solved by introducing an upper FI threshold for the accelerometer and gyroscope. This was done by if the signal exceeded the accelerometer and gyroscope FI thresholds the magnitude thresholds (both UFTs) would be lowered. By doing this, a couple of falls that had low magnitude values compared to other falls could still be detected as falls.

Simple Moving Average Thresholds

The last kind of threshold that was used in this Master's thesis project was a simple moving average threshold for the accelerometer magnitude and the gyroscope magnitude. The SMA is calculated using equation 3.7 and produces one value.



(a) Falling index of a forward fall and all the FI thresholds.

(b) Falling index of a jog.

Figure 4.18: Two figures of falling index plots for a forward fall and jogging.

As previously mentioned the SMA can be a helpful tool for filtering out noise and unwanted spikes in a signal by making the signal smoother.

The reason for introducing the SMA threshold was because it was seen that a fall often produces a much higher SMA value compared to an ADL or an external BWC movement. It was also seen, after inspecting multiple graphs of different types, that a fall produces a wider signal which was the reason why it gave a high SMA value. A wide signal signifies that an event happened over a longer time period compared to an event with a narrow signal. This means that a lot ADLs and external BWC movements could avoid fall detection by setting a sufficiently large SMA threshold. Just like with the magnitude thresholds, a maximum SMA threshold for accelerometer and a maximum SMA threshold for gyroscope was added. A SMA plot including all the thresholds for a fall is shown in Figure 4.19. The zero-padding in the end of the SMA signal is present since SMA produces fewer values than the length of the original magnitude signal.

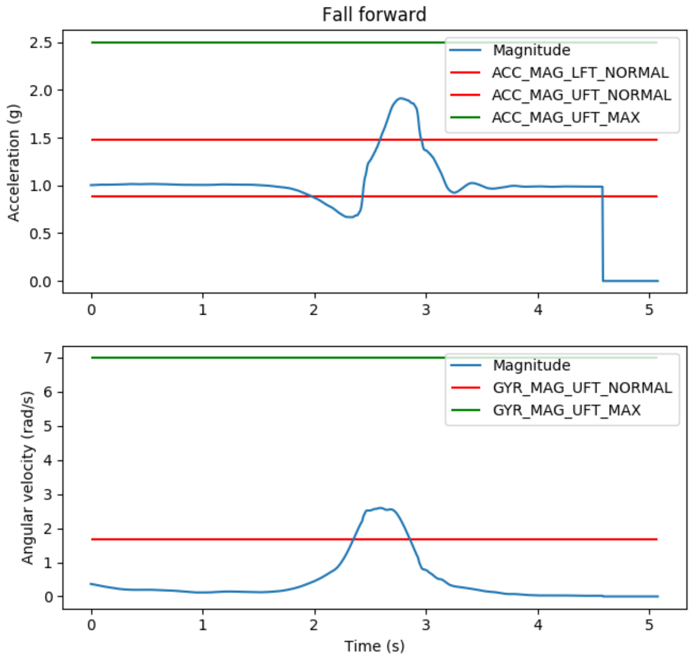


Figure 4.19: The simple moving average signal of a fall forward.

A SMA plot for jogging is shown in Figure 4.20

4.5.2 Signal Shifting to Detect Repetitive Patterns

This idea was tested and explored with the help of the method mentioned in Section 3.8. The goal behind this method was that if a signal was very long and

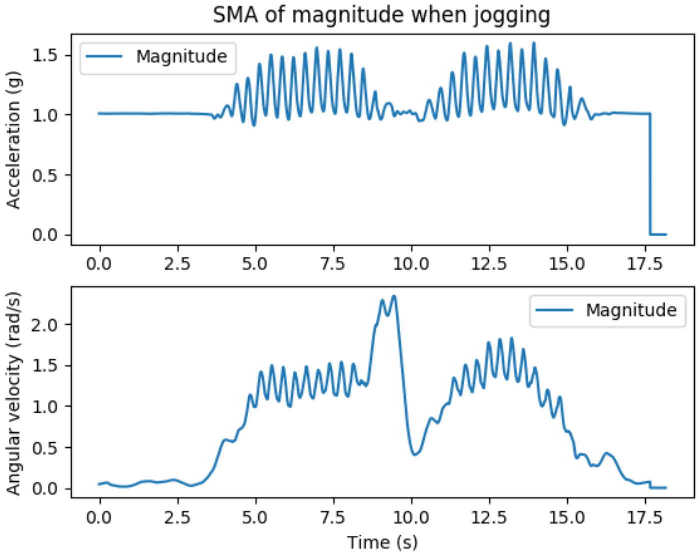


Figure 4.20: The simple moving average signal of a jog.

contained a lot of repetitive patterns it would be easier to identify which part of the signal contained an event, such as a fall. A signal containing repetitive patterns, such as walking or running, could be evened out to a certain degree with the help of this method. This meant that the resulting signal would be somewhat close to a flat line during the repetitive part of the signal. This method had not been tried in previous research but seemed easy to implement and test and for that reason it was proceeded. It also seemed to have the potential to give good results.

4.5.3 Filtering to Remove Shaking

This method was used with the goal of trying to filter and remove unwanted shakings and disturbances that occurred in the signals. The result of this would be that each signal would be easier to read and do calculations on. The first step in this process was to find the frequency range of which shakings and disturbances occurred. This was done with the help of Fourier Transform that was briefly explained in Section 3.4. After finding the frequency range of shakings, a high pass filter could then be applied to get rid of these frequencies. The filter functions used were from the SciPy Python library.

4.5.4 Modifications to Falling Index

The initial falling index was calculated with the formula shown in equation 3.6. Although the result from this equation was very promising, it could still be further improved. With the help and assistance of a supervisor, it was suggested to make the window size for each axis one step larger. The reason for this was because it was supposed to give a slightly more even signal and also to filter out a couple more small spikes and unwanted noise. The initial sample distance window size was two and the new window size became three. The equation for the new falling index formula is shown in equation 4.3.

$$FI = \sqrt{\sum_{i=0}^{80} \frac{(x_{i+2} - x_{i-1})}{3}^2 + \sum_{i=0}^{80} \frac{(y_{i+2} - y_{i-1})}{3}^2 + \sum_{i=0}^{80} \frac{(z_{i+2} - z_{i-1})}{3}^2} \tag{4.3}$$

4.5.5 Inactivity Measurement After Possible Fall

This technique was introduced in an attempt to remove external BWC movements that involved dissociation between the BWC and the user. Example of such external BWC movements is when a user is placing the BWC on a table or in the docking station.

The aftermath of a fall is usually that the person is unable to move for a short amount of time and it is easy to think that there is going to be complete inactivity during this period. This is however not the case since the person will bounce a little bit on the ground upon falling. On the other hand, when putting the BWC down on a table the sensors will show no change because in this situation the BWC is completely still. Graphs that demonstrate this is shown in Figure 4.21. The yellow windows in the figure are the inactivity measurement windows. A calculation can therefore be made to separate a fall from this kind of external BWC movement. This would then result in an overall better and more accurate fall detection algorithm.

This method is performed by doing a calculation on the accelerometer magnitude signal one second after the fall detection. This calculation is done by summarising the absolute value of the difference between the signal value and the gravitation, which is 1, for each sample over a time period of 1 s. Another threshold was then introduced and if the value is lower then the threshold it will not be classified as a fall.

4.6 Optimizing an Algorithm for Different Purposes

The main goal of the developed algorithm was to detect falls. This means that it was important that all the falls get detected and it was acceptable with a couple of false fall detections. The reason for this was because one of the end user of the BWC was police officers and it was important to capture all potential useful footage. It was decided that this trade-off between detecting all falls and avoid false fall detections was acceptable and necessary.

This however does have some drawbacks on the overall performance and accuracy since a slow and light fall will cause all the parameter values used for thresholds to be lowered. By having lower thresholds this also means that a couple of ADLs and external BWC movements maybe be falsely detected as falls.

Because of this, it was decided to create two different sets of parameters. The first set of parameters was about capturing and detecting all the falls, meaning that the sensitivity for falls was 100% and the value of the specificity for ADL

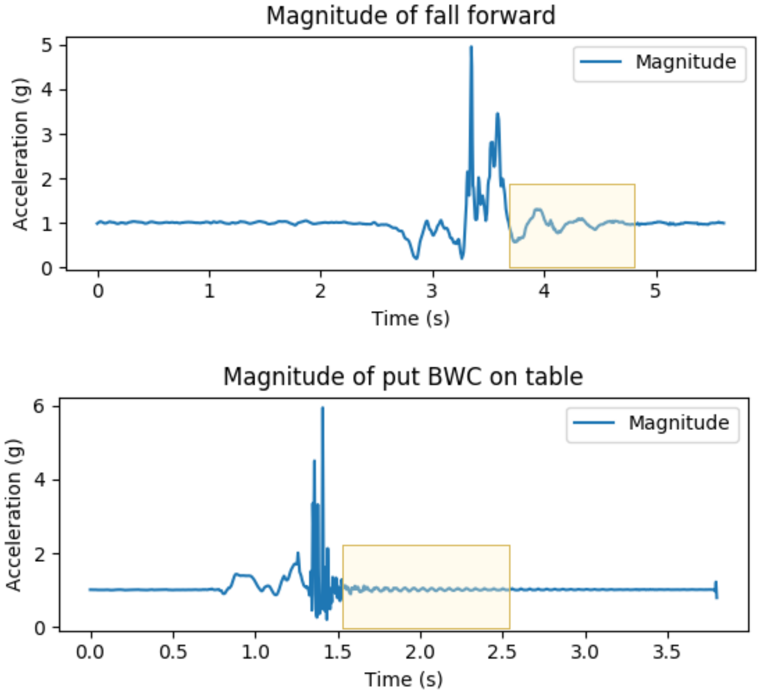


Figure 4.21: The upper graphs shows the inactivity measurement after a fall. The lower graph shows the inactivity measurement after putting the BWC on a table.

and external BWC movements was arbitrary. The measurement for performance is done with the sensitivity and specificity metric explained in Section 3.11. The second set of parameters was created with the goal of trying to get a high value on the sensitivity for falls while also prioritizing a high value on the specificity for ADLs and external BWC movements. This means that the target was to get a as high value as possible on all three metrics.

4.7 Method for Measuring Algorithm Performance

The measurement to determine how good an algorithm performs, is done with the sensitivity and specificity metric explained in Section 3.11. In this Master's thesis project a total amount of 340 data files have been collected. These data files were also used to determine how good each algorithm was. From these data files a training set and a testing set was created. The training set was used to find all the different parameters for the threshold. These parameters was then later used when testing teach algorithm on the test set. It is important to separate the training set from the test set in order to get a correct and unbiased result. To find these different sets, the cross validation method explained in Section 3.12 has been used.

The second test on the different fall detection algorithms was done by using the entire database to find two sets parameters. The first set of parameters was about trying to detect all the falls, meaning that the sensitivity for falls should be 100% and see how this affected the specificity of ADLs and external BWC movements. The second set of parameters was about trying to optimise all three metrics, sensitivity for fall and specificity for ADLs and external BWC movements. This was also mentioned in the previous section.

Chapter 5 Results

This chapter will contain all the findings and results that have been made in this Master's thesis project. The results will be presented using tables and graphs together with descriptive text. The structure of this chapter will include seven sections where each section will contain and explain its own kind of result.

The first section will contain the five different algorithms that have been developed. The second section will contain the different parameters that have been used for each algorithm. It can be noted that each algorithm will have two sets of parameters. The third section contains the performance and accuracy of each algorithm. This section will show how good each algorithm was. The fourth section will contain the result after testing algorithms from previous research on the data set that was produced in this Master's thesis project. The last section will be about the cross validation method. The decision was made to carry out this method only on the best performing algorithm. The reason for this was that it seemed futile and meaningless to do it with a worse performing algorithm since it will mostly likely not be used in the final product. The last two sections will be about the results found when time shifting and filtering a signal.

5.1 The Developed Algorithms

This section will describe the five different algorithms have been developed and tested. The description will contain all the thresholds and functions that the algorithm uses. Below is a list of the algorithms and its parameters.

- **Algorithm 1:** This algorithm used magnitude thresholds and contained a LFT for accelerometer, UFT for accelerometer, UFT for gyroscope, maximum UFT for accelerometer and maximum UFT for gyroscope. The order does not matter, meaning that the order in which each threshold gets met is unimportant.
- **Algorithm 2:** This algorithm contained the same thresholds as algorithm 1 but here the order in which each threshold was met mattered.
- **Algorithm 3:** This algorithm contained the same thresholds as Algorithm 2 and the order of meeting the thresholds was also included. Additional thresholds that was added was the inactivity measurement, a minimum FI

threshold for the accelerometer, maximum FI threshold for the accelerometer and a maximum FI threshold for the gyroscope.

- **Algorithm 4:** This algorithm used SMA thresholds and contained a LFT for accelerometer, UFT for accelerometer, UFT for gyroscope, maximum UFT for accelerometer and maximum UFT for gyroscope. The order in which each threshold was met mattered.
- **Algorithm 5:** This algorithm contained the same thresholds as algorithm 4. Additional thresholds that was added was a minimum FI threshold for the accelerometer, maximum FI threshold for the accelerometer and a maximum FI threshold for the gyroscope.

5.2 Algorithm Parameters

This section will show the different parameters that have been used to produce the result shown in Section 5.3. The different parameters used for each algorithms was brought forward, as previously mentioned, by plotting, calculating and studying the graphs. Each algorithm will have two sets of parameters. The first set will be named "Set 1" and will contain the parameters used when all the falls had to be detected, meaning that the fall sensitivity had to be 100%. The second set of parameters will be named "Set 2" and was used for getting optimal results. This means to get a as high value as possible on all three values together (sensitivity for falls, specificity for ADLs and specificity external BWC movements together).

A couple of parameters have remained the same throughout all the calculations. These parameters values are shown below and have been used if that specific parameter was part of the algorithm.

- Sample rate = 200 Hz or 200 samples/s
- Window size of fall detection = 300 samples
- FI window size = 80 samples
- SMA window size = 100 samples

The different parameter values used by each algorithm is shown in Table 5.1-5.5. The meaning behind each parameter is explained in the table description found above each table.

Table 5.1: Algorithm 1 parameters. P1 - LFT accelerometer (g), P2 - UFT accelerometer (g), P3 - UFT gyroscope (rad/s), P4 - Max accelerometer (g), P5 - Max gyroscope (rad/s)

Algorithm 1					
	P1	**P2**	**P3**	**P4**	**P5**
Set 1	0.55	1.28	1.54	13	21.3
Set 2	0.55	2.5	4.49	7.3	11

Table 5.2: Algorithm 2 parameters. P1 - LFT accelerometer (g), P2 - UFT accelerometer (g), P3 - UFT gyroscope (rad/s), P4 - Max accelerometer (g), P5 - Max gyroscope (rad/s)

Algorithm 2					
	P1	**P2**	**P3**	**P4**	**P5**
Set 1	0.55	1.28	1.54	13	21.3
Set 2	0.55	2.5	4.49	7.3	11

Table 5.3: Algorithm 3 parameters. P1 - LFT accelerometer (g), P2 - UFT accelerometer (g), P3 - UFT gyroscope (rad/s), P4 - Max accelerometer (g), P5 - Max gyroscope (rad/s) P6 - FI accelerometer, P7 - FI gyroscope, P8 - New accelerometer UFT, P9 - New gyroscope UFT, P10 Minimum FI accelerometer, P11 - Max FI accelerometer, P12 - Max FI gyroscope, P13 - Inactivity measurement

Algorithm 3							
	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**
Set 1	0.62	2.45	2.9	13	21.3	2	2.4
Set 2	0.55	2.5	4.49	7.3	11	2	2.4
	P8	**P9**	**P10**	**P11**	**P12**	**P13**	
Set 1	1	1	1.5	8.3	12	10	
Set 2	2.49	2.5	1.5	8	11	10	

Table 5.4: Algorithm 4 parameters. P1 - LFT accelerometer (g), P2 - UFT accelerometer (g), P3 - UFT gyroscope (rad/s), P4 - Max accelerometer (g), P5 - Max gyroscope (rad/s)

Algorithm 4					
	P1	**P2**	**P3**	**P4**	**P5**
Set 1	0.88	1.48	1.67	2.5	7
Set 2	0.8	1.48	1.67	2.5	5.3

Table 5.5: Algorithm 5 parameters. P1 - LFT accelerometer (g), P2 - UFT accelerometer (g), P3 - UFT gyroscope (rad/s), P4 - Max accelerometer (g), P5 - Max gyroscope (rad/s) P6 - FI accelerometer, P7 - FI gyroscope, P8 - New accelerometer UFT, P9 - New gyroscope UFT, P10 Minimum FI accelerometer, P11 - Max FI accelerometer, P12 - Max FI gyroscope

Algorithm 5						
	P1	**P2**	**P3**	**P4**	**P5**	**P6**
Set 1	0.88	1.48	1.67	2.5	7	2
Set 2	0.82	1.48	1.67	2.5	7	2
	P7	**P8**	**P9**	**P10**	**P11**	**P12**
Set 1	2.4	1	1	1.5	8.3	12
Set 2	2.4	1.4	3	1.5	8	11

5.3 Result of Developed Algorithms

This section will present the results that have been produced by the different algorithms. The result will be presented with the help of the sensitivity and specificity measurements. Table 5.6 will show the result of the case where the fall sensitivity was 100% by using the first set of parameters from Section 5.2. Table 5.7 will contain the optimised results calculated with the second set of parameters from Section 5.2. The results shown in this section was produced using all the fall, ADL and external BWC movement data files. Each value will be shown as a percent (%) value.

Table 5.6: Results with 100% fall sensitivity

Result 1			
	Sensitivity fall	**Specificity ADL**	**Specificity external BWC movement**
Algorithm 1	100	12	22.6
Algorithm 2	100	13	26
Algorithm 3	100	73	57.3
Algorithm 4	100	83	98.2
Algorithm 5	100	88	98.2

5.4 Result of Previous Research Algorithms

This section will contain the result produced by algorithms with parameters from previous research. It will be used on the same data set as in Section 5.3. Table

Table 5.7: Optimised Results

Result 2			
	Sensitivity fall	**Specificity ADL**	**Specificity external BWC movement**
Algorithm 1	87.3	85	53
Algorithm 2	79.7	87	66.9
Algorithm 3	92.4	73	70.4
Algorithm 4	88.6	96	99.1
Algorithm 5	92.4	95	99.1

5.8 contains the results of all the tested algorithms.

- Huynh et al. [1] [2] - This algorithm is similar to algorithm 1 shown in Section 5.1 except that it did not include maximum thresholds.
- Bourke et al. [29] - This is a threshold based algorithm similar to algorithm 1 but did not use a gyroscope sensor.

Table 5.8: Previous research algorithms result

Previous Research Algorithms Result			
	Sensitivity fall	**Specificity ADL**	**Specificity external BWC movement**
Huynh et al.	65.8	59	58
Bourke et al.	49.3	57	62.6

5.5 Cross Validation

The cross validation method is explained in Section 3.12 and is used to validate how well an algorithm generalizes on new data, and somewhat validate the data set. Because of this, it was decided to only perform the cross validation with one algorithm. Algorithm 4 was selected for this task because it proved to be one of the best performers by using SMA instead of the regular magnitude. The reason for choosing algorithm 4 instead of algorithm 5 was because it consisted of less parameters which made it more convenient and easier to test. The result produced by these two algorithms was also very similar. Table 5.9-5.10 shows the cross validation method used with the collected data set and algorithm 4. Table

5.9 will show the parameters generated by each training set. Table 5.10 will show the results produced by each test set.

Table 5.9: Cross validation parameters. P1 - LFT acceleromter (g), P2 - UFT acceleromter (g), P3 - UFT gyroscope (rad/s), P4 - Max accelerometer (g), P5 - Max gyroscope (rad/s)

Cross Validation Parameters					
	P1	**P2**	**P3**	**P4**	**P5**
Train 1	0.88	1.48	1.48	1.98	5.62
Train 2	0.88	1.48	1.67	2.08	5.62
Train 3	0.88	1.48	1.67	2.08	5.62
Train 4	0.88	1.48	1.67	2.08	5.5
Train 5	0.86	1.48	1.67	2.08	5.62

Table 5.10: Cross validation result

Cross Validation Results			
	Sensitivity fall	**Specificity ADL**	**Specificity external BWC movement**
Test 1	93.3	80	100
Test 2	100	85	95.6
Test 3	100	100	95.6
Test 4	93.3	75	100
Test 5	93.3	85	100
Average	95.9	85	98.2

5.6 Time Shifting

The time shifting method was first introduced in Section 3.8 with the goal of trying to detect repetitive patterns in a signal. Example of such repetitive pattern is running and walking. The resulting signal would be more even and easier to read which means it would be easier to find interesting information. Although this method has not been used too much in producing the developed fall detection algorithms, it can still be useful for possible future work.

5.7 Filtering

The filtering method is explained in Section 3.7 with the goal of removing shakings that occurred during data collection. These shakings could for example be the result of the camera being unstable during recording or some other disturbances. The frequencies of the shakings was found with the help of fourier transform. Figure 5.1 shows a graph after applying a LPF to a jogging signal. The blue line is the original signal and the orange line is the filtered signal. Just like with time shifting, this method has not been used too much when developing the fall detection algorithms. The filtering implementation was however successful so it can still be useful for possible future work.

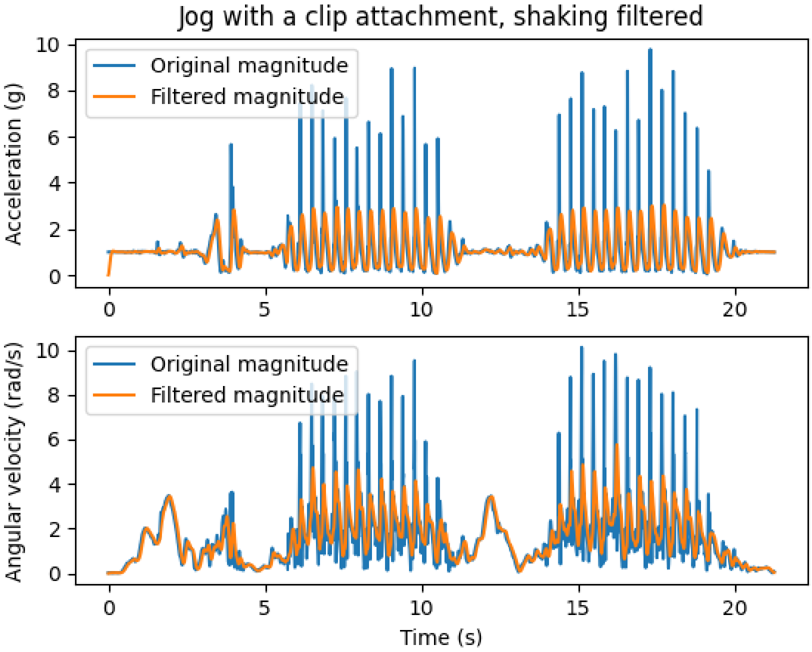


Figure 5.1: Jogging signal filtered with LPF

Chapter 6

Discussion

This chapter will show the analysis and evaluation of the produced result in detail. It will also highlight possible sources of error that could have occurred and affected the result. Lastly, it will include some possible future work and aspects of this Master thesis project that could be improved.

6.1 Performance

The main task of this Master's thesis project was to develop a simple yet robust fall detection algorithm with as high accuracy and precision as possible. A lot of previous research mentioned in chapter 2 was able to achieve a sensitivity of 80% and up and a specificity of 75% and up. These limits was set as a benchmark and the goal was to, at the very least, get above these limits. The results shown in Table 5.7 indicate that this task has been fulfilled for most of the algorithms. It can be worth mentioning that writers of this report are overall satisfied with the results.

Algorithms 4 and 5 ended up particularly good overall, as can be seen in Table 5.6 and 5.7, by calculating with SMA on the magnitude instead of just magnitude. The reason for this was because a lot of noise could be removed, or at least lowered, by using a large enough window when calculating the SMA. It was also seen that falls generally takes longer time compared to for example an external BWC movement and because of this, falls would get a higher value while calculating the SMA. This was another reason why the SMA proved to be very useful for fall detection. It can also be seen that by comparing algorithm 1-3 with algorithm 4 and 5 that it gave overall better results by using SMA. The reason why algorithm 5 performs slightly better than algorithm 4 is because algorithm 5 includes more parameters. The cost of this is that it becomes more complicated to implement algorithm 5 compared to algorithm 4 so this trade-off has to be taken into consideration when implementing the algorithm into a BWC.

The reason behind the difference between algorithm 1 and algorithm 2 is that the thresholds have to be met in a specific order. By including the order, it helped to separate between falls and ADLs plus external BWC movements because most falls followed a specific order as previously mentioned. This means that there is a trade-off between fall sensitivity and ADL and external BWC movement specificity. By including order the fall sensitivity might go down while the ADL and external

BWC movement specificity will go up. After comparing order with no order it could be identified that including the order gave better results, as can be seen in Table 5.7, and that was the reason why order was added to algorithm 3-5.

The FI is included in Algorithm 3 and Algorithm 5 and is used to separate falls from ADLs and external BWC movements. It clearly works as can be seen from Table 5.6 and improves the performance of both algorithms. It does however make the algorithm much more complicated since it includes many more parameters, as can be seen in Table 5.3 and Table 5.5.

The last thing worth mentioning is that there is a big difference in the result when aiming for a 100% fall sensitivity and when looking for the overall best parameters. This can be seen by comparing Table 5.6 and Table 5.7. The reason for this is because there can be a couple of falls that are very slow, also known at outliers, which has a very low valued signal compared to a regular fall signal. By including these outliers it means that the all the thresholds have to be adjusted drastically. This also means that specificity for ADLs and external BWC movements will be decreased since these will now be easier to be detected as falls. Algorithm 4 and 5 is however not too effected by this as can be seen by comparing their result from Table 5.6 and Table 5.7 and the reason for this is because of the SMA. If a person were to implement algorithm 1 or 2 it would probably be a wise decision to go with the optimised parameters. The reason for this is that the overall performance would then be much better since a lot of false fall detections would be avoided.

6.1.1 Previous Research Performance

As can be seen in Table 5.8 the previous research algorithms did not do too well on the data set that has been collected during this Master's thesis project. The reason for this was because these algorithm did not take into account some ADLs and external BWC movements and was therefore not trained for these cases. If they were however, there is a good chance that their performance would have increased. It can also be noted that the way they recorded their data or their lab setup was different from what has been done here. For example, as previously mentioned a belt have been used during data collection and this was not how the previous research recorded their data. This most likely also affected the overall result. It can also be worth mentioning that no previous research mentioned calculating with SMA.

6.1.2 Parameters

The parameters that have been used can be found in Section 5.2. These parameters together with the different algorithms was able to produce promising results and a couple of reliable fall detection algorithms. As mentioned earlier, the way these parameters was decided was through calculations and studying graphs. This approach might not have been the ideal way to generate these parameters and perhaps a different method could have given different parameters and a better result.

6.1.3 Cross validation

The main purpose of the cross validation method was to validate how well an algorithm would generalize on new data. It can be seen in Table 5.10 that the received average score was good which implies that the algorithm has not been over-fitted, and that it will generalize on new data. The five different training and testing sets were created manually by trying to distribute the different falls, ADLs and external BWC movements evenly. This might not have been the best option and instead a fully randomised process to generate these sets might have been better. This could be further explored in possible future work if the cross validation method is used again.

6.1.4 Time Shifting and Filtering

These two methods was both successfully implemented but was for most part unused in this Master's thesis project since they were not needed. The reason for introducing these in the first place was to remove noise and other unwanted signals. This problem was however solved to a large part by including threshold order, FI and SMA in the algorithms. The lab setup used and by recording the data with the help of a belt instead of the clip was also of assistance in solving these problems. That was the main reason for not using these methods any further. But these two methods produced seem to work nevertheless and there is a possibility that they can be useful in future work or other areas that require signal processing.

6.2 Sources of Error

Errors are part of most, if not all projects. They may vary in how much of an effect they have on their surroundings. Some errors can be small enough so that they can be neglected, while others must be considered carefully. It is not always easy to determine a measure on how much impact they have on their surroundings, but nevertheless it is always important to list sources of errors and discuss them as much as possible. In the subsections below some error sources are described that were considered relevant for this Master's thesis project.

6.2.1 Authenticity of Recording Setup

All the different sensor recordings of movement patterns have been done in controlled environments. The most realistic recordings were the ADL recordings, while the least realistic ones were the falls. The reasons for this is the use of the mattresses when performing controlled falls, see Figure 4.11, while the ADL recordings did not require any special setup. A mattress will reduce the fall impact, which is good for the subject performing the fall to avoid injuries, while as in real life the floor or ground may be e.g. concrete or some other hard material, which will have a higher impact. Performing falls on hard floor was never done since it would never be considered ethically acceptable.

The use of a belt when recording movement patterns is also of some concern. As discussed earlier in Subsection 4.1.4, Validating the Data Collection Method, it was used to get rid of shakings of the BWC that occurred when walking or jogging. The shakings would mask the true movements of the subjects body. In a way it is a good thing that only the body movements are recorded, but in real life there will always be some shakings, depending on how sturdy the BWC has been attached. It was however said at the beginning of this Master's thesis work that the end users would wear not too loose clothes which would reduce the shakings.

6.2.2 Sensor Noise and Accuracy

Sensor noise is a property that is always present and affects both the accelerometer and gyroscope. For some applications, such as communications, noise can be a real issue, it depends on how large noise is relative to the signal of interest. When modifying falling index in Subsection 4.5.4, by widening the distance between samples, it did not have too large of an effect on the measurement of falling index. This could indicate that either the sensor does not have much noise or that the levels of the noise are too low compared to the signal of interest.

As mentioned earlier in Section 3.1.2 and 3.2.2 of the Theory chapter, the setting that changes the sensor output range also affects the accuracy of the sensor. There were no apparent indications of that reduced sensor accuracy from choosing a wider sensor output range that was large enough to impact the algorithms in a negative way. For a majority of the developed algorithms it was shown that the wide sensor range was useful for separating falls from some external daily BWC movements with the use of maximum magnitude thresholds, further motivating the trade-off.

6.2.3 Analysis Tools

In Section 4.3, Python, it was mentioned that a tool for analysing recordings and developing algorithms was made. The tool combined many powerful libraries to provided useful functionality throughout the Master's thesis work. As with any software that grows in terms of lines of code, it becomes more difficult to guarantee the correctness of every module and function. The consequences of a bad or faulty function could be that e.g. data recordings are misinterpreted or results are incorrect. To avoid and to detect as many code related bugs and issues as possible, pair programming and code reviewing was done continuously.

6.3 Future Work

In this section, future work that is of interest to look further into is presented. The future work can be both in terms of new ideas or solutions to some of the error sources discussed in the previous section.

6.3.1 Different Placements of BWC

As the BWC is limited to a chest placement to be able to properly record video, it was decided not to explore any other placement of the sensors as stated in the delimitations of this Master's thesis work. However, for detecting falls in a general case it might be of interest to study different sensor locations, e.g. wrist, waist, pocket and leg in the future. Fall detection features are starting to appear in some smartwatches, indicating that even though it seems difficult to use the information from a sensor placed here as indicated by some articles, it is still possible. It is not impossible to imagine putting multiple sensors across the subjects body that work together, if it is able to improve different aspects of fall detection. In this case, of course the user convenience cost must be taken into consideration.

6.3.2 Data Collection

There are a few ways in which data collection can be improved and extended in this Master's thesis work.

More Data

Collecting more data recordings of the same movement patterns can give a better understanding of the signal variations of specific movements. Collecting more different data recordings, such as other ADLs or more falls combined with ADLs will give a better understanding of which movements are a problem, allow them to be studied and eventually make it possible to separate them from falls with great confidence, improving the algorithm.

Collect Data From More Subjects

Increasing the amount of subjects participating in the data recordings will give a better understanding of how falls vary between different people of different weights, lengths and flexibility. As of now most of the recordings have been done by the two students writing this Master's thesis work, see Section 4.1, Data Collection Method.

It could also be of interest to try and compare collected data to what is available online, while remaining cautious on the different aspects of the data collection methods that have been used.

More Realistic Recording Setup

As discussed in the sources of error section previously, the usage of mattresses when falling makes the recordings less realistic. A solution to this could be to collect data from the real users of the BWC and ask them to take notes during the day of what they are doing for both ADls and falls.

6.3.3 Machine Learning

Since there are much research on machine learning within the context of fall detection, see Section 1.7, it would be interesting to see in what different ways it could be applied to solve the problem and if it can improve the accuracy of the algorithm. It does not necessarily have to be used for detecting falls, but instead as a way to find good parameters for the threshold algorithms presented in this Master's thesis work. Although the results produced in this Master's thesis project was very promising, it could very well be further improved with the help of machine learning.

6.3.4 Push Detection

An idea to further improve the developed algorithm could be to also make it able to detect different pushes. The main focus of this Master's thesis project has been to determine if it was possible to detect falls using accelerometers and gyroscopes, and if possible how this could be done. As the threshold algorithms presented was focused on fall detection it will not perform very well for push detection since they are two distinct movements. Because of this, an idea for future research could be to further look into possible algorithms for push detection.

There has already been some data recordings of pushes done and there already been exist a framework for analyzing the data so it could help to speed up this process.

Chapter 7

Conclusions

Five different threshold based algorithms have been developed during this 20 week process. Algorithm 1 - 3 used magnitude in their calculations while algorithm 4 and 5 used SMA instead. After testing and verifying each algorithm, it was revealed that all of them gave overall good results but the algorithms using SMA performed better. The conclusion could also be drawn that the performance increased if the algorithm included FI thresholds. These claims could be further strengthened by comparing the results from this Master's thesis project with previous research algorithms since neither of these used FI or SMA.

Data from the sensors were recorded by the two writers of this report and then saved. A data set was then built up consisting of different falls, ADLs and external BWC movements. This data set was then used to find the different parameters for each algorithm which then led to the results.

Two different approaches have been taken when finding the results. The first set of results had the requirement of demanding a 100% fall sensitivity while the second set of results aimed for overall best performance. It could be seen that the optimised result was to be preferred for algorithm 1 - 3 while for algorithm 4 and 5 the difference was not too big. The trade-off between achieving 100% fall sensitivity and overall performance was very considerable for algorithm 1 - 3.

Similar to the result reported from previous research, the best algorithms developed in this Master's thesis project was able to achieve a fall sensitivity of close to 100% and an ADL specificity around 90%. None of the previous research had however tested their algorithm on external daily BWC movements which has been done here and this could be seen when comparing the results. Since these movements are done regularly it can be said that these movements should be included in order to avoid a lot of false fall detections.

To comment on the research questions stated in the beginning of the report, the high sensitivity score concludes that it is possible to detect falls using an accelerometer and a gyroscope accurately. The high specificity concludes that it is possible to distinguish between falling movements and other daily movements. The algorithms can be made very simple by using only a few parameters as has been shown, with good results, and it is therefore expected to be very robust.

Methods for removing repetitive patterns and filtering unwanted noise have also been developed. It was however shown that the sensor data produced by the BWC did not contain too much noise and because of this no attempt to modify the signals was needed.

Possible future work would be to get additional data from more participant in order to get a better variety and distribution. Different methods for finding parameters could also be further investigated, such as including machine learning into this process. It would definitely also be interesting to see how different placements of the BWC would affect the data and the entire algorithm. There is still many things that can be explored and new methods to be tested. It can however be said that a well functioning fall detection algorithm implemented into a device can definitely improve the safety of its user in today's fast developing world.

References

[1] Q. T. Huynh, U. D. Nguyen, S. V. Tran, A. Nabili and B. Q. Tran, "Fall Detection System Using Combination Accelerometer and Gyroscope", in *Proc. of the Second Intl. Conf. on Advances in Electronic Devices and Circuits*, 2013, doi: 10.3850/ 978-981-07-6261-2_56.

[2] Q. T. Huynh, U. D. Nguyen, L. B. Irazabal, N. Ghassemian and B. Q. Tran, "Optimization of an Accelerometer and Gyroscope-Based Fall Detection Algorithm", Hindawi Publishing Corporation Journal of Sensors, vol. 2015, 2015, Art. no. 452078, doi: 10.1155/2015/452078.

[3] G. Brown and M. Eklund, "An acceleration based fall detector: development, experi- mentation, and analysis", Univ. of California, Engineering Research, Berkeley, CA, USA, 2005.

[4] Maarit Kangas, Antti Konttila, Per Lindgren, Ilkka Winblad and Timo Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers", Gait Posture, vol. 28, no. 2, pp. 285-291, Aug., 2008, doi: 10.1016/j.gaitpost.2008.01.003.

[5] X. Yin, W. Shen, J. Samarabandu and X. Wang, "Human Activity Detection Based on Multiple Smart Phone Sensors and Machine Learning Algorithms", Dept. of Electrical and Computer Engineering University of West em Ontario London, Ontario, Canada, 2015.

[6] P. Vallabh, R. Malekian, N. Ye and D. C. Bogatinoska, "Fall Detection Using Machine Learning Algorithms", Department of Electrical, Electronic and Computer Engineering,University of Pretoria, Pretoria, 0002, South Africa, 2016.

[7] Axis Communications. [Online]. Available: `https://www.axis.com/sv-se`

[8] N. Ravi, N. Dandekar, P. Mysore and M. L. Littman, "Activity Recognition from Accelerometer Data", Department of Electrical,Department of Computer Science Rutgers University, 2013.

[9] P. P. Urone, R. Hinrichs, K. Dirks and M. Sharma, *College Physics.* Houston, Texas, USA:OpenStax, 2012, ch. 2, sec. 4, pp. 40-41. [Online]. Available: `https://openstax.org/details/books/College-Physics`

[10] "ADXL345", https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf (accessed May. 2, 2020).

[11] "LSM6DSO" ST.com. https://www.st.com/en/mems-and-sensors/lsm6dso.html (accessed Apr. 7, 2020).

[12] Shawn, Accelerometer vs Gyroscope sensor, and IMU, how to pick one?. [Online]. Available: https://www.seeedstudio.com/blog/2019/12/24/what-is-accelerometer-gyroscope-and-how-to-pick-one/

[13] H. A. Radi and J. O. Rasmussen, *Principles of Physics.* Germany:Springer Berlin Heidelberg, 2013, ch. 8, sec. 2, pp. 229-230. [Online]. Available: https://link-springer-com.ludwig.lub.lu.se/book/10.1007%2F978-3-642-23026-4

[14] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Bartg and J. Lach, "Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Informationl, University of Virginia, 2009.

[15] J. B. Andersson and R. Johannesson, *Understanding Information Transmission.* Hoboken, New Jersey, USA:John Wiley Sons Inc., 2005, ch. 2, sec. 4, pp. 61-63.

[16] L. S. Noelker and R. Browdie, "Sidney Katz, MD: A New Paradigm for Chronic Illness and Long-Term Care" *The Gerontologist*, vol. 54, no. 1, pp. 13-20, Feb. 2014. Accessed: Apr., 14, 2020. [Online]. Available: https://doi.org/10.1093/geront/gnt086

[17] G. Lindell, *Introduction to Digital Communications.* Lund, Sweden:Department of Electrical and Information Technology, Faculty of Engineering, LTH, Lund University, 2006, ch. 2, sec. 5, pp. 62-63.

[18] S. W. Smith , *The Scientist and Engineer's Guide to Digital Signal Processing.* California Technical Publishing, USA, 1999, ch. 14.

[19] R. Boomgaard, J. Lagerberg and T. Walstra *Digital Signal Processing.* 2016, ch. 5. [Online]. Available: https://developer.android.com/studio

[20] R.J. Hyndman and R.J. Athanasopoulos , *Forecasting: principles and practice, 2nd edition*, OTexts: Melbourne, Australia, 2018, ch. 2.

[21] J. Patterson and A. Gibson, *Deep Learning: A Practitioner's approach.* Sebastopol, California, USA:O'Reilly Media, Inc., 2017, ch. 1, pp. 36-39.

[22] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python.* Sebastopol, California, USA:O'Reilly Media, Inc., 2017, ch. 5, pp. 254-256.

[23] *Android.* Google and Open Handset Alliance. [Online]. Available: https://www.android.com/

[24] *Android Studio.* Google and JetBrains. [Online]. Available: https://developer.android.com/studio

[25] *Python.* (3.5). Python Software Foundation. [Online]. Available: https://www.python.org/

[26] *NumPy.* NumPy Developers. [Online]. Available: https://numpy.org/

[27] *SciPy.* SciPy developers. [Online]. Available: https://www.scipy.org/

[28] *Matplotlib.* The Matplotlib development team. [Online]. Available: https://matplotlib.org/

[29] A.K. Bourke, J.V. O'Brien and G.M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm", Biomedical Electronics Laboratory, Department of Electronic and Computer Engineering, University of Limerick, Limerick, Ireland Department of Physical Education and Sport Sciences, University of Limerick, Limerick, Ireland, 2007.

Printed by Tryckeriet i E-huset, Lund 2020

LUND
UNIVERSITY

Series of Master’s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2020-761
http://www.eit.lth.se