DESIGN AND VALIDATION OF AN INSTRUMENTED KNEE BRACE WITH ADJUSTABLE RESTRICTIONS FOR MACHINE LEARNING-BASED GAIT ANALYSIS

RAHIM DHUPALIA

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS
FOR THE DEGREE OF
MASTERS OF APPLIED SCIENCE

GRADUATE PROGRAM IN ELECTRICAL AND
COMPUTER ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO

JUNE 2026

ABSTRACT

Gait data can be analysed to classify which activities are performed and gauge the gait-related health of the performer. We consider health in the context of knee osteoarthritis and gait-afflicting conditions. This work captures gait from sixteen healthy participants who performed walking and sit-to-stand activities in a lab while artificially restricted at four severities, using a custom collection device and instrumented brace. A convolutional neural network classified activities with participant-averaged accuracies of 65.61% ± 8.85% and 67.21% ± 9.06% on two well-performing hyper parameter sets. Analysis found time-series features that differentiated between some restriction levels for each of the activities. Significant difference was found between swing phase of unrestricted walking activities and all other restriction levels, and for toe-off of unrestricted walking activities and moderate and severe restrictions. For sit-to-stand, significant difference was found between jerk standard deviation of negotiating sitting edge of unrestricted data and moderate and severe restrictions.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisors, Dr. Gerd Grau and Dr. Garrett Melenka, who have helped me in many ways, at many times, to complete this work. Their patience, understanding and guidance has been invaluable to me, and has allowed me to continue on when the future of my project was uncertain, as it so often felt. I am deeply thankful to them both for their time, knowledge and support. Further, I would like to thank Dr. William Gage, whose collaboration and expertise helped clarify the biomechanical side of this work.

I would like to thank Vincent Nguyen, a student of Dr. Gage and a close collaborator, who provided insights and experience into the biomechanical and human-facing parts of this work. Having a fellow student who was working on a project that was similar and facing similar challenges was very reassuring. I wish him good luck in all his future endeavours!

I would like to thank York University, whose administrative and teaching faculty and support have allowed me to complete not only an undergraduate degree, but also struggle towards success in a graduate degree. Moreover, they have granted me the opportunity to learn and grow, for which I am grateful. I hope to put what I have learned here to good use.

More personally, I would like to thank my friends and family, who have offered time and efforts, in one way or another, to help me succeed at this project. I thank my parents, who have worked long and hard, and often thanklessly, to help me reach this far into my education. I hope to be able to share the benefits of that education with them as the years go by.

To my friends, I thank you for being patient with me over the course of this project. Your friendship has kept me a little saner, and a little more able to keep pushing when it was difficult. To Edward in particular, I thank you for your tireless friendship and insights. I don’t think I’d have been able to get here without your help and support – with a special thanks to Alan.

I offer gratitude to everyone I’ve interacted with in Dr. Melenka’s lab. A conversation with fellow students going through the same things means a great deal more than it seems, and over the years, I’d been lucky enough to have many such conversations. Many thanks to all of you, with best wishes for your future goals!

Lastly, and most abstractly, I am thankful to anyone who has taught me anything useful throughout my life. It is knowledge which has granted me the ability to reach this point, and it is that knowledge that I am most thankful for, and will continue to be thankful for as it helps guide my future. Thank you all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

I2C: Inter-Integrated Circuit

SPI: Serial Peripheral Interface

LED: Light-Emitting Diode

RTC: Real-Time Clock

IDE: Integrated Development Environment

IMU: Inertial Measurement Unit

MUX: Multiplexer

RMS: Root-Mean-Squared

GPIO: General Purpose Input/Output

ICQ: Inter-Core Queue

ML: Machine Learning

GPO: General Purpose Output

PCB: Printed Circuit Board

FFT: Fast Fourier Transform

IQR: Inter-Quartile Range

DCU: Data Collection Unit

ANOVA: Analysis of Variance

EMG: Electromyography

KOA: Knee Osteoarthritis

SVM: Support Vector Machine

NN: Neural Network

CNN: Convolutional Neural Network

RNN: Recurrent Neural Network

LSTM: Long Short Term Memory

kNN: K Nearest Neighbours

HAR: Human Activity Recognition

1 INTRODUCTION AND LITERATURE REVIEW

1.1 INTRODUCTION

Human gait, particularly walking gait, is the cyclical pattern exhibited by humans as they walk[1,2]. Studying the gait of an individual can help assess conditions that affect it, such as Parkinson's[3], osteoarthritis[4,5], and various other neuromusculoskeletal conditions[6]. Further, the analysis of gait can also be used to differentiate severities between diseases, helping indicate how a disease is progressing[7,8]. Thus, having ways to measure and analyse human gait is an important field of study towards a better quality of life for afflicted individuals.

Multiple methods exist for the capture of gait data, such as optical capture techniques, both with [9,10] and without[11,12] the use of markers, as well as sensors that are mounted on the individual of interest, known as wearables[13,14]. The drawback of optical capture techniques is the necessity for there to be a camera or system to capture the movement of interest, which restricts where activities can be performed[15]. Thus a less restricting and more flexible method of data collection is found in wearables, which can capture gait information in a variety of sensor modalities, such as goniometers capturing angles, electromyography sensors capturing muscle signals, or inertial measurement units capturing acceleration, angular velocity, and magnetic field variations[2].

The option exists to use complete sensor systems which handle capture, transmission and processing of the data[16,17], or engineer custom capture systems based on needs[18,19]. Particularly in the latter case, it is a good idea to validate the custom system against a known gold-standard system to ensure that data collected are valid and represent reality.

Once the data are captured, processing and analysis can be done in a variety of ways, depending on the goal to be accomplished. At the level of a single individual, data may reasonably be parsed manually, though at scale, a framework for processing becomes necessary. There is work that classifies disease or abnormality[20–22], recognises activity[23,24], validates systems[25,26], or predicts gait parameters[27,28]. One technique of analysis that is currently very popular is machine learning, whereby a model can be trained on captured data to learn a relationship that holds for new data.

Our work produces a data collection device to capture inertial measurement unit (IMU) sensor data from the lower limb of individuals. The accelerometer results of the device are validated against a gold-standard system. Further, the device is used to capture data from individuals who are artificially inhibited by an orthosis to train a machine learning model to recognise the activities being performed. We also look for parameters in the data which can be used to differentiate between differing severities of artificial restriction. These artificial restrictions are meant to imitate gait-afflicting diseases, though the context of knee osteoarthritis (KOA) was the primary area of applicability that was considered.

Arthritis is a disease that can affect different parts of the body, and can roughly be segregated into two major types: inflammatory arthritis and osteoarthritis [29]. Inflammatory arthritis is caused by inflammation at the site of the disease causing joint damage, whereas osteoarthritis is caused by the body's inability to repair lost cartilage at the site of the disease [29,30]. Osteoarthritis can cause disability [31,32] and is of concern both in a global, and national context. Long, Liu et al. found that globally, prevalence of osteoarthritis increased by 113.25% since 1990, to 527.81 million cases [31]. In Canada, osteoarthritis the most common form of arthritis, affecting between 13.6%[33] to 14.3%[34] of Canadians. Further, of the various sites that osteoarthritis can affect, the knee is a common location [4,30,35,36], with some sources finding it to be the most common site of the disease [31,33,37]. This agrees with the Arthritis Society of Canada's claim that of those who experience pain as a symptom of osteoarthritis, the knee is most commonly reported as the site of that pain [38].

Knee osteoarthritis can cause problems performing activities of daily living due to hindering walking [39], which directly and indirectly affects the quality of life of the patient[32,40,41]. Knee osteoarthritis is linked to different symptoms, amongst which, pain, reduced joint motion and stiffness [30,33,35–38] are of interest to us. It is this limited joint motion that we will later aim to imitate by the use of a restricting knee brace. The lower leg can swing towards the thigh – flexion – or away from the thigh – extension; using an orthosis, we can limit these movements in a controlled manner.

The remaining chapter aims to establish a context by which the rest of the work should be viewed. Various areas of research are relevant, including human gait measurement and analysis, machine learning techniques, and human activity recognition, particularly in the context of gait-afflicting diseases. We describe the human gait cycle, the need to quantify it, and the approaches used to do so. We also highlight what work has been done in the literature to capture gait, specifically in clinical contexts, and home in on the area of special interest we have for our work.

1.2 *HUMAN GAIT*

1.2.1 A GAIT CYCLE

Human gait refers to how human beings walk. It is a cyclical pattern that allows us to move from one place to another, and varies across both healthy and unhealthy groups[2]. It is a cyclical behaviour, and facilitates the transfer of weight from one leg to the other in order to propel the body forward in a controlled manner. The different phases of a gait cycle are described using various, non-universal [1,2], terminology, however, the general structure is broken into two uneven phases: the swing phase and the stance phase. See Figure 1-1 for details.

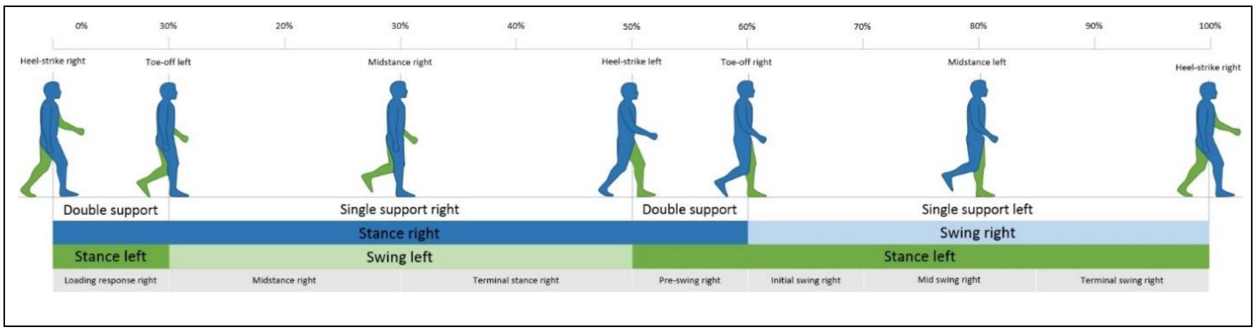


Figure 1-1: A diagram of the walking gait cycle, taken from [42], under CC BY-SA 4.0 terms. The diagram shows the stance and swing phases of both legs, as well as the activities that occur within those phases. Note that the terminology used to describe these smaller phases can vary. During stance phase, the leg accepts the weight of the body while the other leg prepares to swing forward. As that leg swings, the weight has now moved ahead of the leg in the stance phase, necessitating the other leg to enter the stance, so that the weight can be supported.

The stance phase of gait begins when one of the feet makes contact with the ground. We refer to this as heel strike, though initial contact is also a term that is used[2]. Once that heel makes contact, the body can start to apply weight to that limb, as it uses that leg as support the body. As the other leg swings forward and eventually makes contact with the ground, the leg in stance phase makes it way towards the toe-off, where it comes off the ground. This begins the swing phase for this leg, where the other leg is supporting the weight of the body, and the leg in swing phase allows the body to propel forwards. Once this leg makes contact with the ground, beginning stance phase, we have completed a single gait cycle[1,2].

1.2.2 THE NEED TO QUANTIFY

Details about a gait cycle can be quantified, such as using physical parameters like stride time, stride length, and movement speed, the gait kinematics, which describe the motion of the bodies performing gait, or the gait kinetics, which describe the forces acting during gait[2]. Though this work can be used in various fields, such as athletics and sports, and even gait forensics, the need for such quantification becomes obvious in the context of diseased gait, and disease detection.

Various conditions can cause a gait that departs from normal behaviour. One such example is Parkinson's, which may cause slowness of movement (bradykinesia[43]), and small steps [44]. Alterations to gait can be caused by underlying neuromusculoskeletal conditions[2], and thus, monitoring and quantifying gait can provide insight into an existing condition, or highlight if a condition is developing or changing.

1.2.3 GAIT MEASUREMENT APPROACHES

In the literature, various sensor-based solutions are validated against gold-standard systems to check that their data reflect reality[45–50], which then allows these systems to be used untethered and in the real world. Some researchers produce entire measurement systems which integrate existing sensors [25,26,51–53], and then validate those against gold-standard systems. We endeavoured to accomplish a similar task, by producing a custom IMU data collection device, which is meant to be portable in both indoor and outdoor environments, and allow for collection in natural walking contexts. This work lays the foundation to that goal, by producing the device, and validating the accelerometer against a known gold-standard system.

1.2.3.1 OPTOELECTRONIC MOTION CAPTURE AND THE GOLD STANDARD

Collecting gait kinematic data from individuals in daily life is the first step to making predictions about how well they are moving, however, the current gold-standard approach presents various constraints that make everyday collection difficult [54,55]. An example of a gold-standard approach to collecting gait data is an optoelectronic camera system [46,54,55] such as the VICON MX. The VICON MX system requires placement of retroreflective markers on the individual of interest, at specific positions on the body, depending on what is intended to be captured[56]. Further, the markers are ideally placed on skin-tight fabric or a Lycra suit, in order to prevent movement of the markers. Collection can only be performed within a capture volume, defined as an area around which is placed multiple VICON optoelectronic cameras, such as the MX40 cameras. These cameras are all connected to connectivity hardware which allows the system to communicate with a host computer, which captures motion. The data collected are X, Y and Z position, relative to some "home" coordinate system. A diagram derived from the VICON Nexus 1.8 Guide[57] is provided in Figure 1-2.

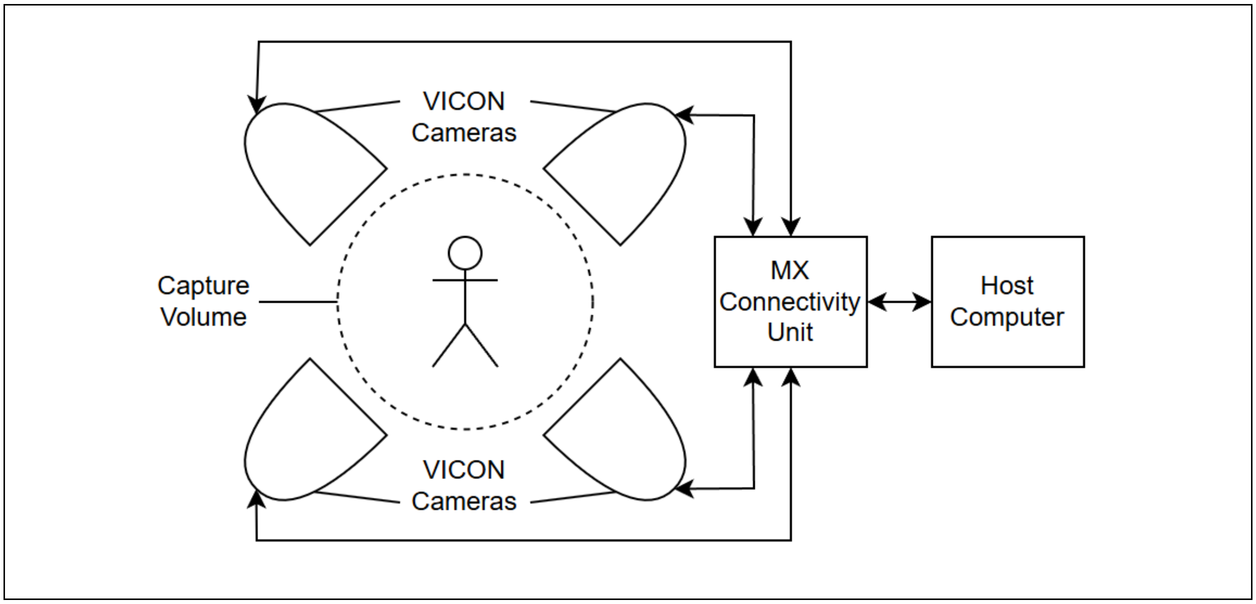


Figure 1-2: VICON MX System Layout derived from the VICON Nexus 1.8 Guide. The system uses a series of optoelectronic cameras to capture movement from retroreflective markers within the capture volume. The cameras all connect to MX Connectivity Units, which allow the whole setup to interface with a host PC with appropriate software installed. More details can be found at [57].

The inherent weakness of a camera-based motion capture is the necessity of the subject to be within a visible area relative to the placement of the camera, which is not always possible in the home, and rarely possible when outside the home. Thus, other methods to capture data must be considered if gait kinematic data are to be collected more freely, as it has been shown that lab-based and community/natural gait can vary in diseased populations [58–61], and even in non-diseased populations [62]. Wearable technology lends itself well to the purpose of natural gait monitoring; there exist sensors which are wireless[17], allowing more ease of movement in the home, whereas other sensors can stream to a mobile app[16], allowing freedom to move beyond the home.

1.2.3.2 ALTERNATE METHODS OF CAPTURE

Various types of wearables are available, such as electromyography devices, pressure-measuring insoles, digital goniometers, and inertial measurement units (IMUs)[63]. Electromyography (EMG) is used to determine the properties of electrical signals fired by motor neurons which cause muscle contractions[64]. Pressure-measuring insoles can be placed inside the shoes of individuals, and allow for the measurement of pressure distribution on the feet during gait. Goniometers can be used to measure the angle between joints during movement. IMUs can measure acceleration, rotational orientation, and magnetic field, all in X, Y and Z directions. The ability to convert from acceleration to position was of particular interest in this study when considering different wearables, since the VICON system tracks the position of its reflective markers, and so IMUs offered the possibility of position to position comparison with the VICON. This was seen as a possible verification approach for the results of the system being built, and so IMUs were chosen as the sensor to build a system around. Sections 1.2.3.2.1 to 1.2.3.2.4 describe some differing sensors and their applications to activity classification; Table 1-1 provides a summary of these considerations. Three major points of comparison were assessed, as follows:

1. **Ease of Activity Classification** encompasses the idea of how easy it is to use the sensor data to perform activity classification, which is one of the major goals of the system to be built. This category seems awash between all the different types, as varyingly complex methods can be applied for that purpose.
2. **Ease of VICON Comparison** questions how often and how easily the sensors are compared to VICON/Opto-Electronic Camera systems, as this was the means by which our system would be validated. The goniometer would be the most straightforward, since it can be used to compare joint angles to joint angles, however, the studies looked at did not underline any additional processing and analysis required in IMU and Camera comparisons. This may be due to the use of commercial systems with analysis capabilities built-in however. Physically, the IMUs can give X, Y, Z accelerations, which can be manipulated to give positions, or the VICON positions can be manipulated to give accelerations, which may give an easier arena for comparison relative to EMGs. Otherwise, further analysis would be needed to get goniometer data to compare to VICON data.

3. **Ease of Use** asks how easy it would be to work with these sensors when administering the study. This seems difficult for EMGs, relative to IMUs and Goniometers. Goniometers seem the most suitable for easy placement and comparison to VICON, however the IMU provides additional data that could be useful for other analysis, and thus, was chosen as the sensor for the system to be built.

1.2.3.2.1 EMG CONSIDERATIONS

Consider the aforementioned criteria when looking at EMG sensors:

1. **Ease of Activity Classification:** EMG data can be used alone [65], or with other data [66] for non-gait activity classification. It is also used in gait phase classification [67,68]. Fricke et al. use EMGs and various ML models for the classification of healthy and unhealthy gait [69]. Zhang et al. use SVM and EMG data to classify sitting and standing activity [70]. Al-Quraishi et al. use Linear Discriminant Analysis, kNN and Naïve Baiyes for ankle movement classification from EMG signals [71]. Phukpattaranont et al. found neural networks paired with specific feature extraction methods were best to classify finger movements [72]. Tokas et al provide various studies that use surface EMGs for the task of activity classification[73].
2. **Ease of VICON Comparison:** Difficult, since the VICON measures XYZ position data and derived data from those base data, whereas the EMGs would provide muscle activation data. Coker et al. reference a few different papers in their work that all use EMG and ML approaches to attain joint angle estimates for various joints [74]; the paper itself uses VICON data and EMG data in its model to predict future joint angle position. Zhao et al. mention various studies that have used surface EMGs with various ML techniques to predict joint angles [75]. Gautam et al. used surface EMG data to make predictions on joint angles using a neural network.[76]
3. **Ease of Use:** The Trigno Avanti Sensors (Delsys, Massachusetts USA) require cleaning of the surface of the contact point at the muscle, as well as careful placement of the sensors [77]. Donovan et al. also mentions cleaning and shaving the skin, as well as careful placement of electrodes even though they use a completely custom system [61]. Zhao et al. use Shimmer3 sensors (Dublin, Ireland) [75], whose manufacturers also recommend that electrodes are placed on cleaned and shaven skin for best results [78]. Surface EMGs can be dry or gelled[79], which can affect ease of use.

1.2.3.2.2 INSOLE CONSIDERATIONS

Consider the aforementioned criteria when looking at insole sensors:

1. **Ease of Activity Classification:** Djamaa et al. use a force sensor, bend sensor and IMU in an insole to predict activity with more than 97% classification activity [13]. They classify shuffle walking, normal walking, and toe walking, using SVM, kNN, and multi-boosting algorithms. The authors also references many works that use a variety of sensors and classification models to classify activities. For instance, [80] uses only force and bend sensors to classify between three activities using PCA (Principal Component Analysis). Note however that the form factor of an insole only allows tracking of lower leg data, which may not provide a complete picture of activity if there are other points of interest.
2. **Ease of VICON Comparison:** The insole is not a sensor in itself, but a form factor, which can contain sensors, such as pressure sensors [13,80–82] or IMUs [81–83]. Thus depending on the underlying sensor being used, VICON comparison can be made difficult or easy. Riglet et al. use VICON and insoles with IMU sensors to calculate various gait parameters and analyse the criterion validity and test-retest reliability of the systems [49].
3. **Ease of Use:** Very straightforward, for [13] simply place the insole in the shoe, and wear the shoe. Donkrajang et al. depict sensors affixed to the insole, and participants wear the shoe [80]. Chen et al. have a pressure sensor and IMU connected to an insole, with a stated goal of their device being “unobstructive” [82].

1.2.3.2.3 GONIOMETER CONSIDERATIONS

Consider the aforementioned criteria when looking at goniometers:

1. **Ease of Activity Classification:** Halim et al. use electro-goniometers (from the ankle and knee) along with IMU and EMGs for activity classification on an existing dataset [84]. The models used are XGBoost, Random Forest and SVM. Manually extracted features were very simple. Accuracy was 90%+. Javed et al. use the same existing dataset to perform activity classification with Deep Neural Networks, with 98.1% accuracy [85]. Their approach is complex relative to [84]. Since a gait activity is often mediated by movement at the knee, it follows that using goniometer angles should easily classify what activity is being

performed. Schlegel et al. show the value of keypoint-data derived joint angles for activity classification from existing datasets [86].

2. **Ease of VICON Comparison:** Ishida et al. compare derived knee flexion/extension angles from a Cortex optical system against angle measurements from a goniometer [47]. The same authors test their goniometer against the optical system after on-body recalibrations [87]. It appears that optical comparison can be made rather straightforwardly, as existing systems can typically derive joint angles. For instance, the VICON Nexus software, which is used to process captured VICON data in a life sciences setting [88], has features for joint angle calculation [89].
3. **Ease of Use:** Goniometers can be placed at the joint of interest, with the only real concern being for how to affix the sensor to the joint.

1.2.3.2.4 IMU CONSIDERATIONS

Consider the aforementioned criteria when looking at IMUs:

1. **Ease of Activity Classification:** Feldhege et al. predict simple activities like lying, sitting, walking and standing without AI modelling, using knee angle [90]. This is something that would easily be done by goniometers by definition. Pesenti et al. use LSTM to predict between walking, standing, lifting, and lowering activities in an industrial context [91]. Eyobu and Han use features derived from IMUs with LSTM for Walking, Sitting and Standing classification [92]. Müller et al. use CNNs and IMU data to predict various exercise activities [93]. Kuduz and Kaçar use IMU and goniometer data to classify fast, normal and slow walking for different participants [94]. Li et al. use IMU data to predict between walking, running, stair ascent and descent [95]. Mitchell et al. found that across various existing datasets analysed, IMU and goniometer data were found to be capable of high activity classification accuracy [96].
2. **Ease of VICON Comparison:** Yang et al. compares IMU-based system orientation against VICON calculated orientation [97]. Fusca et al. use an Elite motion capture system to validate various measured gait parameters calculated using values derived from IMU measurements [98]. Tak et al. uses an OptiTrack Flex against a novel IMU-based system to compare joint angles during a single-legged squat [48]. He et al. compare gait parameters derived from IMU sensors at the heel and pressure-measuring insoles to a Qualisys Optical

Motion Capture system [50]. Lewin et al. compare a RunScribe IMU against force plates, a Qualisys Optical system, and a Delsys Accelerometer to measure a variety of parameters [99].

3. **Ease of Use:** Sensors can be placed fairly straightforwardly, though the sensor should be affixed well; movement or rotation of the sensor can affect results.

Table 1-1: A summary of various considerations made when choosing a sensor, against multiple sensor types. The criteria considered for comparison were: 1. Ease of Activity Classification, 2. Ease of VICON Comparison, 3. Ease of Use.

Criteria	**Sensor Type**			
	EMG	**Insole**	**Goniometer**	**IMU**
Ease of Activity Classification	Low	Moderate	High	High
Ease of VICON Comparison	Low	High	High	High
Ease of Use	Low	High	Moderate	Moderate

Complete wireless IMU systems are available for use, such as the "Awinda" line of products (Movella, Nevada USA), "Movella DOT" (Movella, Nevada USA), or the "IMU Step" (I Measure U, Centennial Colorado), and are detailed in Table 1-2, including the details of our device, with technical specifications available in Table 1-3 to Table 1-6. In general, the ready-to-use solutions can capture IMU information, and sometimes additional information, as well as do on-board processing, or pass the data to a software tool to do analysis, processing, and visualisation. However, such systems can still have their limitations, such as the price, or being tethered to a wireless receiver.

Table 1-2: An overview of various complete solutions available for the capture and analysis of IMU data. Note that this table is not a comprehensive listing of all existing complete solutions. Further information can be seen in the relevant additional tables below.

Solution	Price	Features	Notes
IMU Step (I Measure U Ltd.)	$6600 USD/year to $22000 USD/year, featuring different levels of support and sensors	• Varies based on tier of subscription. • All levels feature web and mobile app for interfacing with the sensors. • Sensors have 1GB onboard storage, or can stream data wirelessly.	The complete system appears to be a combination of an app (IMU Step), and the sensors themselves (Blue Trident). Details at [16] and https://imeasureu.com/imu-step/
MTw Awinda (Movella Inc.)	Prices can vary by configuration. $715USD for a single sensor kit, and $3575USD for a 6 sensor kit.	• Can connect 34 sensors simultaneously • Real-Time visualisation of data • "XSens Kalman Filter for Human Motion"[100] • Barometer and thermometer • Can be synchronized externally.	Awinda appears to the name of a communication protocol patented by Movella, with various tiers of their products using it. The XSens and MTw Awinda are the same product, with the MVN Awinda Starter being slightly less feature-rich. The company also has other motion capture sensors available. Details at [17] and www.movella.com
Movella DOT (Movella Inc.)	$750USD for 5	• Transmission of Bluetooth data or onboard storage • On-board sensor fusion • Some on-board processing capacity	The DOT is reasonably close to the intent of our application – it is a ready-made sensor with some on-board processing capability, which can transmit data via Bluetooth. Note, these are aimed at commercial institutions. Details at [101] and www.movella.com
Delsys Trigno Systems	~$4000USD for 2 Avanti Sensors with the Trigno Lite – more for additional sensors or Trigno Centro.	• Many sensor form factors • All form factors have IMU and EMG sensing • Wireless connection to base-station or USB dongle.	The Trigno Lite and Trigno Centro systems use the Trigno Avanti Sensors, which feature EMG and IMU support. Other compatible sensor form factors exist, such as the Mini, Duo and Quattro. Details at [102,103] and www.delsys.com
DCU	~$500CAD (~$360USD)	• Variable onboard storage • Thermometer • Can be synchronized externally • 5 Hours+ battery life.	-

Table 1-3: Technical specifications of the Blue Trident sensor, part of the IMU Step system from I Measure U Ltd, derived from [104]. The details of the accelerometer, gyroscope, magnetometer and any other relevant information is provided below. Note, a 'g' is a unit of acceleration: 9.8m/s²

<table>
<tr><th rowspan="2">Blue Trident IMU Step Sensor</th><th colspan="2">Tri-Axial Accelerometer</th><th rowspan="2">Tri-Axial Gyroscope</th><th rowspan="2">Tri-Axial Magnetometer</th></tr>
<tr><th>Low</th><th>High</th></tr>
<tr><td>Range</td><td>±16g</td><td>±200g</td><td>±2000°/second</td><td>±4900µT</td></tr>
<tr><td>Resolution</td><td>16 bit</td><td>13 bit</td><td>16 bit</td><td>16 bit</td></tr>
<tr><td>Sample Rate</td><td>1125Hz</td><td>1600Hz</td><td>1125Hz</td><td>112Hz
(70Hz in some modes)</td></tr>
<tr><td>Battery</td><td colspan="4">12 Hours Depending On Usage</td></tr>
<tr><td>Notes</td><td colspan="4">This sensor has two accelerometers with a lower and higher range, which have slightly different specifications, and is meant to allow the system to capture both minor and major accelerations that may occur during a monitored activity.</td></tr>
</table>

Table 1-4: Technical specifications of the MTw Awinda sensor, part of a motion capture system from Movella Inc., derived from [17]. The details of the accelerometer, gyroscope, magnetometer, and any other relevant information is provided below. Note, a 'g' is a unit of acceleration: 9.8 m/s², and a Gauss is a unit of Magnetic Field: 1T is 10000G.

<table>
<tr><th>MTw Awinda</th><th>Tri-Axial Accelerometer</th><th>Tri-Axial Gyroscope</th><th>Tri-Axial Magnetometer</th></tr>
<tr><td>Range</td><td>±160 m/s² ≈ ±16g</td><td>±2000°/second</td><td>±1.9 Gauss ≈ ±190µT</td></tr>
<tr><td>Resolution</td><td>16 bit</td><td>16 bit</td><td>16 bit</td></tr>
<tr><td>Sample Rate</td><td colspan="3">60-120Hz Depending on number of sensors connected simultaneously</td></tr>
<tr><td>Wireless Range</td><td colspan="3">10m to 50m depending on capture environment and type of receiver</td></tr>
<tr><td>Battery</td><td colspan="3">12 Hour Continuous Use</td></tr>
<tr><td>Notes</td><td colspan="3">Sensor also contains a barometer for height estimation</td></tr>
</table>

Table 1-5: Technical specifications for the Movella DOT sensor, derived from [101,105]. The details of the accelerometer, gyroscope, magnetometer, and any other relevant information is provided below. Note, a 'g' is a unit of acceleration: 9.8 m/s².

<table>
<tr><th>Movella DOT</th><th>Tri-Axial Accelerometer</th><th>Tri-Axial Gyroscope</th><th>Tri-Axial Magnetometer</th></tr>
<tr><td>Range</td><td>±16g</td><td>±2000°/s</td><td>±8 Gauss ≈ ±800µT</td></tr>
<tr><td>Resolution</td><td>16 bit</td><td>16 bit</td><td>16 bit</td></tr>
<tr><td>Sample Rate</td><td colspan="3">1Hz, 4Hz, 10Hz, 12Hz, 15Hz, 20Hz, 30Hz, 60Hz, 120Hz for recording mode only</td></tr>
<tr><td>Wireless Range</td><td colspan="3">Bluetooth 5.0</td></tr>
<tr><td>Battery</td><td colspan="3">8 hours continuous use</td></tr>
<tr><td>Notes</td><td colspan="3">Note, unlike the other systems, the Movella DOT appears to be meant to integrate into existing systems, however it does have compatibility with a Movella DOT app, which allows for a more complete system-like use. It can store data on its on-board 64MB storage at up to 120Hz, or stream data in real-time at lower rates. The sensor has software development tools available to develop apps around the sensor. It features sensor fusion on-board for orientation calculation, but can also output accelerations and angular velocities.</td></tr>
</table>

Table 1-6: Technical specifications of Delsys series of sensors. These sensors would connect to either a Delsys Centro station or Delsys Lite receiver dongle. Delsys has a series of different sensors available, which all, in addition to IMU data, can capture EMG data. The details of the accelerometer, gyroscope, magnetometer, and any other relevant information is provided below. Note, a 'g' is a unit of acceleration: 9.8 m/s².

Delsys Trigno Avanti	**Tri-Axial Accelerometer**	**Tri-Axial Gyroscope**	**Tri-Axial Magnetometer**
Range	±2g or ±4g or ±8g or ±16g	±250°/s or ±500°/s or ±1000°/s or ±2000°/s	No Raw Output Available
Resolution	16 bits	16 bits	No Raw Output Available
Sample Rate	n*(1/0.0135s) $1 \leq n \leq 13$	n*(1/0.0135s) $1 \leq n \leq 10$	~74Hz
Wireless Range	20+ metres, depending on capture environment		
Battery	4 to 8 hours, depending on use		
Delsys Quattro	**Tri-Axial Accelerometer**	**Tri-Axial Gyroscope**	**Tri-Axial Magnetometer**
Range	±2g or ±4g or ±8g or ±16g	±250°/s or ±500°/s or ±1000°/s or ±2000°/s	±4900 µT
Resolution	10 or 16 bits	10 or 16 bits	No Raw Output Available
Sample Rate	74Hz or 148Hz or 222Hz or 296Hz or 741Hz	74Hz or 148Hz or 370Hz or 741Hz	~74Hz
Wireless Range	20+ metres, depending on capture environment		
Battery	2 to 6 hours, depending on use		
Delsys Duo	**Tri-Axial Accelerometer**	**Tri-Axial Gyroscope**	**Tri-Axial Magnetometer**
Range	±2g or ±4g or ±8g or ±16g	±250°/s or ±500°/s or ±1000°/s or ±2000°/s	±4900 µT
Resolution	16 bits	16 bits	No Raw Output Available
Sample Rate	222Hz or 74Hz or 963Hz	119Hz or 24Hz or 361Hz	~74Hz
Wireless Range	20+ metres, depending on capture environment		
Battery	2 to 6 hours, depending on use		
Delsys Mini	**Tri-Axial Accelerometer**	**Tri-Axial Gyroscope**	**Tri-Axial Magnetometer**
Range	±2g or ±4g or ±8g or ±16g	±250°/s or ±500°/s or ±1000°/s or ±2000°/s	±4900 µT
Resolution	10 or 16 bits	10 or 16 bits	No Raw Output Available
Sample Rate	296Hz or 148Hz or 519Hz or 963Hz or 74Hz or 370Hz	296Hz or 148Hz or 519Hz or 74Hz or 741Hz or 370Hz	~74Hz
Wireless Range	20+ metres, depending on capture environment		
Battery	2 to 6 hours, depending on use		
Notes			
The Mini, Duo and Quattro are a series of sensors that have one, two or four detection heads respectively that extend out from a main sensor body. The Trigno Avanti sensors are a single sensor body. All the sensors are wireless, and have varying technical features, such as onboard calculation capabilities. All the sensors can communicate wirelessly with either a Trigno Centro station, which supports up to 32 sensors, or the Trigno Lite dongle, which supports up to 4 sensors. More details on the Mini, Duo and Quattro can be found at [102], and details on the Avanti and overall system can be found at [103]. Note that according to Delsys, the magnetometer cannot be polled for raw data, so resolution of those data is not provided. Further, the Avanti sensor manual makes no mention of the range of the magnetometer – likely for the same reason.			

1.2.4 GAIT MEASUREMENT IN CLINICAL CONTEXTS

The literature applies gait measurements and analysis to various pathological and clinical contexts. McGrath et al. highlight the value of gait analysis through three-dimensional gait analysis, which includes joint kinetics, kinematics, muscle activation information, and a physiotherapist examination [106]. This information has proven to be useful in providing better plans for patients, and better outcomes for pathological child patients, and McGrath underlines the potential for better outcomes in pathological adult patients as a finding from their work.

Romijnders et al. claim that gait events can be used for monitoring disease progression in Parkinson's. They compare IMUs measurements against optical motion capture measurements during gait of healthy older adults, stroke patients, and Parkinson's patients, across various types of walking, in conditions that imitate real-life walking. They found that a shank-mounted IMU could be used for gait event detection in straight-line and curved walking [107]. Lanotte et al. used lower leg mounted IMUs on patients with strokes to validate modifications to an algorithm which was used to segment gait events in Parkinson's patients (Salarian gait segmentation algorithm) [108]. Wu et al. uses two heel-mounted IMU to collect kinematic data on post-stroke patients, and produce an algorithm that produces a gait quality metric which is compared against doctor produced scores[109].

Other less obvious conditions can also benefit from gait analysis and measurement, such as diabetes. Inacio et al. state that diabetes can cause gait deterioration over time, and by using spectral parameters of gait data collected from a back-mounted IMU, they were able to differentiate diabetic participants from healthy ones[110].

There are limitations to gait measurements, especially in how reflective in-lab measurements can be of in-community activities. Hillel et al. found that for elderly fallers, using a back-mounted accelerometer, their in-community gait was closer to dual-task walking than normal walking[111]. Further still, they found that the features they compared were worse performing in the real world than in the lab. Similarly, Shema-Shiratzky, Hillel et al. found that for multiple sclerosis patients, their in-community walking resembled dual-task walking from the lab[59].

Our focus in the clinical gait analysis literature was knee osteoarthritis (KOA), in keeping with the direction of the rest of the work. We found a variety of work in the space, which differed in number and positioning of IMU sensors used, the purposes to which the collected data were applied, as well as whether the data were collected from different populations at the same time, or the same populations at different times. We begin by considering literature in the area which uses IMUs as their method of capture of data, and tune our discussion from there to address the niche that we aim to target.

1.2.4.1 GAIT MEASUREMENT IN KOA CONTEXTS

According to a literature survey conducted by Pollet and Buraschi[4], various gait parameters were affected by knee osteoarthritis. The most pronounced change was in gait speed, where afflicted individuals would walk slower. Changes also occur throughout the gait cycle, such as stance phase showing different muscle activations, and greater ipsilateral trunk-lean angle. Changes also occur in a kinetics context, such as a reduced moments due to altered position of the leg relative to healthy individuals. They also mention a reduction in knee flexion angle prior to swing phase, amongst other alterations. Favre and Jolles claim that knee flexion angle at heel-strike is an indicator of disease progression, with higher flexion angles showing greater disease progression, and that knee flexion angle at terminal stance is greater for patients with more severe KOA, but that the latter has not been reported to be associated with disease progression[112].

A survey from Kour, Gupta and Arora which assessed literature up till 2018, presents three general categories of collection techniques of data in the KOA space, which are the vision-based, sensor-based (which we have also elaborated upon in 1.2.3.2), and hybrid-based approaches[113]. A survey from Choursiya et al. found that IMUs, which fall into the wearable sensor-based approaches, were useful in capturing kinematic data at a similar accuracy to 3D motion capture systems[114].

Due to the flexibility of IMUs, various mounting positions can be chosen for collection of data. A survey from Prisco et al., which was not specific to KOA, looks at IMU-based gait analysis systems which were compared against optical motion capture systems over 32 papers. The found the lower body to be the most common point of single IMU placement, and multi IMU setups used low back, feet, shank and thighs [115].

The data collected from IMUs can be used for various purposes, such as Almuhammadi et al. using accelerometer data to predict patient pain levels[116]. Raimondo et al. used kinematic data captured from an inertial capture system (consisting of IMUs) as a component of a model which could noninvasively estimate knee contact forces in vivo[117]. Priso et al. found in the works they surveyed that some authors focused on spatio-temporal parameters, others on joint kinematic, and two on gait events [115].

Our specific interests were in disease detection, and finding parameters from gait measurement which could be used to differentiate between healthy or unhealthy individuals. Similar work has been done in the space, such as by Pan et al. showed various gait differences across mild KOA and healthy participants[118], as well as mild KOA and moderate KOA severity groups using kinetic and kinematic data collected via motion capture[119]. There are studies that use gait analysis with optical motion capture to collect information the knee adduction moment [120–123], which is a kinetic variable[120] that relates to disease severity[124]; Sakamoto et al used a single IMU mounted to the tibial tuberosity to estimate the knee adduction moment and found satisfactory results[125]. These results encouraged us to consider IMUs as a means to collect gait data. This is also a present effort in the space, such as work by Xia et al. who used thirteen IMUs on healthy and diseased gait individuals, collected acceleration and gyroscope data, and used machine learning to predict whether data originated from diseased or non-diseased gaits[126]. Nishizawa et al used helical axis variability as a derived measure from IMUs, and used it to differentiate between KOA and non-KOA participants, finding that KOA patients exhibit lower helical axis variability[22]. Tsurumiya et al. used five IMU sensors and collected stance phase varus/valgus angular velocity data and mediolateral acceleration data, and determined differences between healthy and KOA groups, but were not able to significantly differentiate between different KOA groups[7]. Vangeneugden et al. monitored healthy, KOA and obese KOA groups over a week to ten days in the real-world using an accelerometer, and found differences between healthy and KOA groups, but did not appear to find differences between KOA groups[127]. The inherent design of many of these studies is that the healthy and unhealthy groups are not the same individuals, and differences are being found over different people.

Alternatively, there are longitudinal studies, wherein participants are assessed over time, to consider changes in condition. Sun et al. extracted various features from 2 year longitudinal knee osteoarthritis data from the Osteoarthritis Initiative, including hip-mounted accelerometer data, and found which features best predicted gait decline [128]. Mahmoudian et al. did a 2 year longitudinal study on women with early or established KOA and found kinematic and kinetic changes that occurred only in the established KOA group over time[129]. Hensor et al. found over a longitudinal study of up to 7 years on subjective pain score data from the Osteoarthritis Initiative that knee pain is likely to first occur in weight-bearing knee-bending activities[130]. Hubley-Kozey et al. did a longitudinal study from data after 8 years, and found that participants that went on to get total knee anthroplasty had higher knee adduction moment at baseline than the groups than did not[131]. Naturally the issue in these studies is that changes must be assessed over a long period of time.

If we did not want to compare different groups of healthy and unhealthy people, nor assess the same people over a long period of time, what option would we have to see how the same person reacts under different gait conditions?

A possible choice here would be to simulate impaired gait. A non-rigorous search was completed on the terms "(synthetic OR simulated OR simulation OR restricted OR restriction) AND (impairment OR disease) AND gait knee" and relevant literature was found. Simulated restrictions appear in various contexts, such as Alyami and Nessler who simulated asymmetric gait for single limb amputation and neurological impairment conditions and collected kinematic data with optical motion capture on a vertically oscillating treadmill[132]. McMain et al restricted healthy participants at knees and ankles, unilaterally and bilaterally, and sought to find metabolic cost changes by these impaired gaits[133]. This space has been active for a long time - for instance, in 1994, Cerny et al applied knee flexion contracture of 20° and 40° degrees, as well as a knee flexion contracture of 40° with a plantar flexion angle of 0°, and studied the adaptation effects during stance phase[134]. Similarly, in 2008, Harato et al applied unilateral knee flexion contractures of 0°, 15° and 30°, and studied how the impairment affected both limbs[135]. Xingchen et al. used hip and knee IMUs and the dynamic time-warping method to identify differences in time series data between healthy and simulated diseased gaits by applying contractures of 40° and 60° at the knee[136].

More closely related to our intended area is Hwang et al., who used an IMU system with four IMUs placed along the sagittal plane, on the lateral side of both legs at the thighs and shanks. They collected data from these sensors and an instrumented walkway for comparison, while the participants were unrestricted, restricted at the knee, and restricted at the ankle. They then developed various models to classify features from the collected gait data in a variety of ways. They found a 91% accuracy with their model[137].

1.3 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

In recent years, artificial intelligence (AI) and machine learning (ML) have become very popular fields of endeavour. AI is the general area of study, which features narrow, general and super intelligence, with ML being a subset of narrow intelligence[138]. In ML, a computer program is used to create statistical models which can be used for a wide assortment of applications, and performs various tasks, typically classification or regression, through supervised or non-supervised learning.

In unsupervised learning, the model does not know how the inputs given to it relate to each other, and it must find an underlying pattern to group inputs together. There are a variety of strategies to apply for this task, such as clustering, apriori algorithm, and Gaussian mixed models[139]. In supervised learning, the model is given the input data with expected output values, which the model is striving towards. As the model trains on those supervised data, it adjusts its weights so that the expected output value for a given input is the one that the model predicts. There is also semi-supervised learning, which uses both labelled and unlabelled data in its training[140]. We focus our discussion on supervised learning.

For our purposes, a model can either learn to predict discrete classes, known as a classification problem, or continuous values, known as a regression problem. For instance, a model trying to predict what gait phase is associated to which data would be solving a classification problem, whereas a model trying to predict knee joint angles associated to data would be solving a regression problem. Our work attempts to solve a classification problem, and offers preliminary efforts towards solving a regression problem in future work.

Different approaches to solving these types of problem exist, such as support vector machines (SVM) or logisitic regression for classification problems, or linear regression and decision trees for regression[141,142]. Both of these problems can also be tackled using a neural network (NN). Neural networks cascade linear combinations, activation functions, which provide non-linearity[143,144], and other computational layers to predict output on labelled input.

The general flow of a supervised neural network is to first take the dataset and split it into three pieces: training, validation, testing, where the training and validation splits account for most of the data, and testing accounts for a minority. The model would be provided the labelled training data, from which it would attempt to learn an underlying pattern. The process of training entails comparing the model's predictions against the labels, and measure how "badly" it is doing. This measure of badness, the loss function, is a function of parameters that the model can adjust, and thus, it continually adjusts these weights through training to minimise the value it calculates for its loss. Once all of its training data are exhausted, the model is validated on the unseen validation data. This process repeats for multiple units called epochs, after which the parameters which provided the best validation score are used to test the model. The model is tested on the hereto unseen testing data, which gives us a sense of how well the model will work on completely new data, known as generalisation[141].

Of particular use to us was the convolutional neural network (CNN), which is used typically in computer vision contexts, but is not restricted to only that use case. It allows us to learn weights that consider multi-dimensional data concurrently. Consider Figure 1-3 which shows the general idea, which can be extended out to multiple dimensions. The kernel moves by a user-selected amount every time, and has a user-selected size. Notice that the convolution allows multiple dimensions of data to be "seen" by the kernel window at the same time. The weights that the model would adjust here are the weights of the kernel, also known as a filter[145].

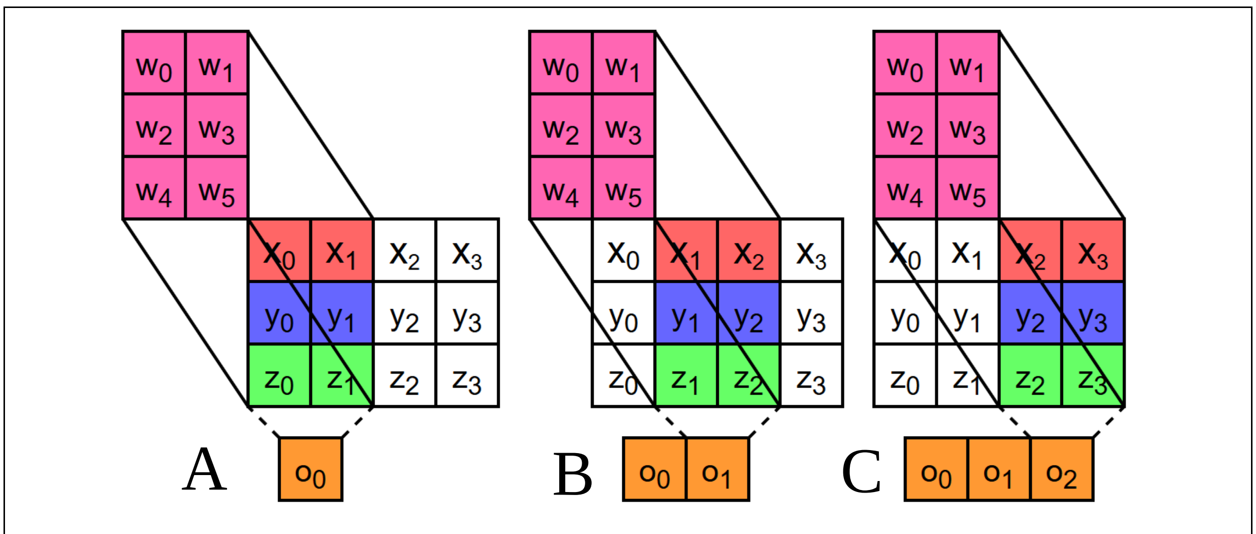


Figure 1-3: The process of 2D convolution for some example data. Assume x, y and z represent different dimensions of data. The kernel of weights is passed along the data by a user-selected amount each step. At each step, a dot product is computed on the corresponding elements of the kernel and the elements of the data. At A, the kernel convolves the first two instances of all the dimensions of the data. At B, the kernel convolves the first and second instances of all the dimensions of the data. At C, the kernel convolves last two instances of all the dimensions of the data.

1.3.1 GAIT AND MACHINE LEARNING

In the context of gait, machine learning is used in a variety of ways, such as for gait event classification, activity classification and disease classification, amongst other uses. The types of machine learning applied, the data they are applied to, and the outcomes that are being predicted can vary depending on the use case. We offer a brief look at various works in the activity and gait event classification area, and in a more clinical space, to provide the reader with a context for the sort of research that occurs.

1.3.1.1 MACHINE LEARNING IN ACTIVITY AND GAIT EVENT CLASSIFICATION

Javed et al. use the pre-existing ENABL3S dataset to train and test machine learning models to predict gait events and performed activity from given input data[85]. The dataset contains data from multiple IMU, surface EMG, and goniometers, as well and a variety of activity, such as walking, stair ascent, standing, etc. The authors produced a deep NN to predict activity performed, with 98.1% accuracy, as well as gait events during a particular type of activity, with 99.96% accuracy.

Similar work was performed by Mitchell et al. who performed activity classification on walking, standing, stair-ascent and stair-descent, over four different datasets with different sensor types and configurations, with different cross-validation approaches[96]. They used K Nearest Neighbour, SVM, decision trees, random forests and artificial NNs, and found the NNs and SVMs to be highest performing, with accuracies dependent on configuration of window size, dataset, and cross-validation method.

Romijnders et al used IMU data, collected from left and right ankles and the proximal end of the left and right shank, on healthy and unhealthy (multiple sclerosis, Parkinson's, recent stroke, chronic lower back pain, and others) participants[146]. The goal was gait event detection, particularly initial contact and final contact. They used a temporal convolutional network to classify the 3D accelerometer and 3D gyroscope input from an individual IMU as initial contact and final contact, and found high F1 scores. They also calculated errors in event timing between the predicted events and a simultaneously captured reference system timing results and found small differences between the two, as well for other derived gait parameters.

Pérez-Ibarra et al. used a single foot-mounted IMU to predict gait events against force sensor data, for healthy and Parkinson's populations[147]. They used a heuristic model and multiple linear classifiers to make predictions in an online fashion, and used SVM to learn the parameters for each linear classifier. They found F1 scores greater than 0.90 on both of their models, predicting the occurrences of heel off, toe off, heel strike, and toe strike.

Mazon et al. took data from ENABL3S and MyLeg datasets, particularly from upper and lower leg IMUs, and aimed to classify activity performed and transitions between them [148]. They used variations of convolutional networks to perform the classification task on frequency-transformed data. They have a cascading architecture that first classifies the input activity, and then a second level that determines transition and gait phase. They found that the convolutional NN with long short-term memory performed well on intra healthy participant trials (~0.90 F1 Score), and on intra unhealthy trials (~0.90 F1 Score). It did not perform as well on inter healthy participant trials, and on mixed healthy and unhealthy trials.

For activity classification, there is also the research area of human activity recognition (HAR), using IMU-based strategies. A brief search on Google Scholar with the terms "human activity (classification OR recognition OR detection) IMU gait knee (diseased OR disease OR restricted OR restriction OR impaired OR impairment OR abnormal)" identified various works in the space, as well as other searches on related terms in the YorkU library.

Ayman et al. used wrist-mounted IMU data from the PAMAP2 dataset[149] to train SVM, Random Forest, and Bagged Decision Tree models to classify human activities, with accuracies of greater than 90% based on the choice of sensors used as input, across all their tested models[150]. They extracted features from the time domain on the X, Y, Z and magnitudes of the sensors.

Some authors intend their work to be used in a real-time classification capacity, such as Li et al. who produced a model which classified data from a single foot-mounted IMU[95]. They found 99% accuracy using a triplet Markov chain model. Their model determines gait activities and gait phases, with an accuracy of greater than 93% on walking, running, stair ascent or descent, in an online context, and lower accuracies in an offline context. Another work, Eyobu and Han, tested a feature extraction and representation process in an online context, and tested their process on a deep long short term memory (LSTM) network, with 80%-90% accuracy in classifying walking, sitting, and standing on healthy participants, depending on parameter configuration[92].

More relevant to us is HAR work done with NNs, as this is the area we are interested in. Plenty of work exists in this space as well. We highlight work that relates to our methodology and approach, and extract relevant information into Table 1-7. Note that these works may have had better results using different models, but we specifically highlight their convolutional NN results and related context where possible.

Tseng and Wen produced a real-time IMU-based activity recognition system, whose results were tested against multiple model types, including various NNs[151]. They built a custom embedded system, collected IMU-based features, and processed them to derive some additional data, such as Euler angles, and linear-angular velocity. They converted their input data into an input image, which various models were trained on, with the best performances being found by random forest and extreme gradient boosting methods. These models found the best accuracy of around 90% or higher, depending on model parameter configurations.

Sherratt et al. collect data from 5 IMU sensors, placed on both hips, the inner ankles, and the chest, and study how a simplified LSTM understands the data, as well as how a full-scale LSTM performs on the data[152]. Their data include both the activities themselves, as well as the transitions between activities. Some light pre-processing is done on the data, and they were fed to the model in various lengths of time, to see how the model performed. Various model configuration parameters were iterated over. The best results they found without a transition state, which is most similar to our case, was 84.7% at test time, for a model with 128 timesteps, and 6 unit single layer LSTM. The exact input configuration to the model is unclear, though it is believed to be 3D accelerometer and gyroscope data from the shank IMU.

Bijalan et al. used a single IMU mounted at the center of mass to collect data from 30 participants, with ages ranging from 10 to 45 years[153]. They claim to minimally process their data, via normalisation and scaling, and turn it into a matrix of 80 timestamps by 3 rows, with 3D accelerometer data being used. They pass the data to various neural networks, such as deep NN, bidirectional LSTM, convolutional NN, and a convolutional NN combined with an LSTM. They found the best accuracy on a convolutional NN with LSTM, at 90%. There are some details that are unclear in their work, such as the number of participants used in their models, whether joint angles were input to the model, or whether the data were collected in a laboratory or natural setting. Thus, we provide information in Table 1-7 based on our best understanding of the work.

Tao et al. produced a novel architecture to classify activities on UCI-HAR[154] and MHealth[155] datasets, which contain data from IMU sensors while basic movements are performed[156]. Their model contains multiple "feature mixer" units, which themselves contain a convolutional window, self-attention block, and a direct-parallel fully connected layer. They connected four of these units together, and then produce output from a final fully connected layer. We specifically highlight their convolutional window approach in their feature mixers, which passes a 2D array of IMU sensor data to a window that is passed over the data. The data are sent in with 10 timesteps at a time, with multiple IMUs horizontally stacked. Overall, they found 95.83% accuracy on HCI-HAR data, and 96.01% accuracy on mHealth data with their approach. It is unclear what their training and testing regiment is.

Tan et al. investigate human activity classification in a knee osteoarthritis context, asking whether or not models would succeed on diseased data[157]. They collected IMU data from 18 KOA patients, and produced a model which classifies at multiple levels of granularity. They classified the activity being performed, the direction the activity is being performed in, if relevant, and the phase of the activity, with varying results. Four IMUs are used, and placed on the outside of the thigh, and inside of the shank. They used a convolutional NN, which took the accelerometer and gyroscope data as an input image, which were first segmented by fixed window sizes, depending on which level they wanted to classify. They found 85% accuracy on classifying activity, approximately 90% or higher accuracy at direction classification, and weak results for phase classification.

Tokas et al. produce a model meant to use surface EMG data to classify activities, but they test their model on IMU data from the mHealth and WISDM[158] dataset to prove the flexibility of their approach[73]. They devised significant pre-processing methodology which was applied to surface EMG data, but it is unclear if it was applied to the IMU data. The IMU data are segmented into 256ms windows (2 seconds for WISDM), and passed to the model with each IMU channel being passed in parallel to the model. The model performs 3 parallel 1D convolutions using multiple kernel sizes to extract information at different timescales, then combines the results, passes it further through the model, and produces an output prediction. The accuracies were very strong, with 99.2% for mHealth, and 98.6% for WISDM. They do not state their validation strategy for WISDM; we assume it be the same as for mHealth. The WISDM dataset used does not provide IMU positioning or activity collection environment information.

Slemenšek et al. built an embedded system to collect IMU and strain gauge data from both legs to perform activity recognition, on five participants, on five classes[159]. The sensors were placed below the knee, anterior to the shin bone, and data were collected indoors and outdoors. They tested many models with significantly and lightly pre-processed data from one participant to try and decide which best model to use to classify on all the participants. It is unclear whether the light pre-processing involved the normalisation that the significantly pre-processed data did. It is also unclear if the significantly pre-processed data had a time axis or not when providing input to the model. Raw data were provided to various models as a 2D array of 20 feature columns by 3 minutes of timesteps. The best models were used to independently test all the participants, which produced accuracies of ~99% for all of them, on raw data. The best performing models were convolutional NNs with and without attention, combined with a recurrent NN, and they found that training and testing on raw data did better than results from their processed input data. The authors also did freezing of gait detection on a Parkinson's patient's gait, and found best results with pre-processed data and the convolutional NN with attention combined with the recurrent NN (98.8%).

Table 1-7: A compilation of details from literature. We particularly focus on activity recognition using IMUs, via convolutional neural networks. In most cases, the data are also provided to the models in a similar fashion to how we intend to. Accel is short for accelerometer, Gyro is short for gyroscope, and Mag is short for magnetometer.

Work	Training Participants	Sensor Locations	Data Given to Model	Format Given to Model	Accuracy	Activities Classified
Tseng and Wen [151]	N = 1	Waist	3D Accel, 3D Gyro, 3D Mag, Euler Angles, Quaternions, 3D Linear Angular Velocities	Grayscale Image, with feature reduction using multiple methods	CNN: ~80%	Stand-Up, Sit-Down, Run, Walk, Continuous Actions
Sherratt et al.[152]	N = 17 or 18	Inside Ankles, Chest, Both Hips, (Best Results with only shank)	3D Accel, 3D Gyro	Various length time series of data, with limited pre-processing	LSTM, 76.0% - 84.7%, depending on model parameter	Walking, Stopped, Ascent/Descent of Stairs, Ascent/Descent of Ramps and Transitions, in Natural Environments
Bijalwan et al.[153]	N = 25	Centre of Mass	3D Accel, Timestamps, Subject ID	80 Timestamps by 3 Rows	CNN: 84%	Normal Walk, Jogging, Walking on Toe, Walking on Heel, Upstairs, Downstairs, Sit-Ups, in Natural and Laboratory environments
Tao et al. [156]	N = 30; 10 (UCI-HAR; mHealth)	Waist; Chest, Right Wrist, Left Ankle	3D Accel, 3D Gyro	10 timesteps, by 6*#IMUs columns	Convolved Model: 95.83%; 96.01%	6 Activities in Lab; 12 Activities Out-of-Lab
Tan et al. [157]	N = 17	Outer thigh, inner shank	3D Accel, 3D Gyro	Various timesteps by 6*4IMU columns	CNN Model: 85%	Walk, Chair, Stairs, in lab
Tokas et al. [73]	N = 9, 51 (mHealth; WISDM)	Chest, Right Wrist, Left Ankle; Smartwatch, Smartphone	3D Accel, 3D Gyro	256 ms; 2 s time series in parallel for each channel.	CNN Model: 99.2%; 98.6%	12 Activities Out-of-Lab; 18 Activities
Slemenšek et al.[159]	N = 1 (Five Independent Participants)	Left and Right leg, below knee.	3D Accel, 3D Gyro, strain gauge, complimentary filtered 3D Accel and 3D Gyro	3 minutes by 20 features (if raw), matrix of 540 variables (if processed).	CNN Model with Attention and Recurrent NN, on raw data: 99.2%	Standing still, Walking, Running, Up/Down stairs, indoors and outdoors

1.3.1.2 MACHINE LEARNING IN CLINICAL CONTEXTS

In a review of ML techniques in marker-based clinical gait analysis, Dibbern et al. found common use of SVMs, clustering post-principal component analysis reduction, neural networks, and logistic regression[160]. They found the most common application areas to be Cerebral Palsy, Parkinson's and post-stroke conditions.

There are also markerless applications, such as the work done by Iseki et al, who used a markerless pose capture technique, and applied a deep learning light gradient boosting machine to analyse the captured data[161]. Their goal was disease detection - whether or not their model could detect pathological gait (from a population of idiopathic normal pressure hydrocephalus, Parkinson's, and other neuromuscular pathologies) or normal gait - from the three dimensional pose estimates of 27 points along the body. They found an accuracy of 70% in their work. Further, the technique they used to capture the markerless data, the TDPT-GT app on iPhone, was also built using a machine learning approach[162].

Ali et al. used gait videos from an existing dataset, performed silhouette segmentation from those inputs by applying a convolutional neural network[163]. They then used a separate variety of convolutional neural networks with gated recurrent units to extract spatiotemporal features. The models then classified abnormal and normal gait, as well as KOA, Parkinson and normal gait; the former problem saw better accuracies than the latter.

Chavez et al. used OpenPose (itself a convolutional neural network based method[164]) and OpenCV (a computer vision library[165]) tools to extract pose information from videos of healthy and Parkinson's gait[166]. They linearly combine the unhealthy and healthy pose information to form synthetic varyingly diseased pose information, to address the paucity of gait data they found in the space, and trained SVMs, K-Nearest-Neighbour, and Gradient Boosting algorithms. The found acceptable accuracy from their results; their K Nearest Neighbour model had 97% accuracy.

Mahum et al. used X-ray images of healthy and diseased knees to detect the severity of KOA, by using convolutional neural networks and other models to extract features from the images, and then using SVM, random forest, and K Nearest Neighbours, they classified the data as having particular severities of KOA[167]. They found that using the K Nearest Neighbour as their classifier, with a convolutional neural network and histogram of gradients to extract features, that they had accuracy in the high 90 percents.

Of course, machine learning can be applied to different sensors. For instance Moghadam et al. sought to create a convolutional neural network and random forest model to predict gait kinematics and kinetics in typically developed children[27]. They collected kinematic and kinetic information to compare against using force plates and optical capture, and extracted out features from the time series data, and produced general and personal models and compared the results. They found that personal models did better than general models, and that the random forest and convolutional neural network classified with similar performance. Moghadam et al. also did similar work on adults, attempting to find good parameters for window selection, IMU number and placement, for use with a random forest to predict kinetics and kinematics[168].

Xu et al produced a pressure-sensor based low-cost device for the use of monitoring the gait of children, with mild deviations[169]. They produced neural network and SVM models and tested them in combination with various reduction techniques of data, to find if their model could classify what type of mild gait deviation the children had (toe-in, toe-out, flat-foot, normal).

Sarmah et al. used an IMU to capture pelvis and trunk motion, along with separately collected pedobarographic data, on KOA patients to produce a machine learning classifier which predicts whether a given input has KOA or not[14]. They also do some clustering work on processed EMG signals to see if the algorithm can separate out different severities of KOA data.

Raza et al. used IMUs embedded in shoes to collect walking data from KOA and healthy participants, and sought to classify three different scenarios using SVMs[83]. They found 71% accuracy on classifying healthy and diseased gait, 83% accuracy on classifying surgery-scheduled KOA participants from non-surgery-scheduled KOA participants, and 81% accuracy on classifying unilateral and bilateral KOA.

Kaya et al. used video data captured from KOA patients and healthy participants, provided from an existing dataset, extracted out 3D pose data, and performed a classification task to determine the radiographic severity of KOA[170]. They first took the video data and applied existing pose-estimation techniques to extract 3D pose information from multiple key body locations, then identified gait cycles and cleaned the data, keeping only lower leg pose estimates, which would be used for machine learning. They used a long short-term memory fully convolutional network for classification of KOA severity, and found good performance (0.91 F1 Score) when testing and training on the same participants, but lower performance (0.76 F1 Score) when training and testing on different groups of participants. The severe and healthy groups had good performance, but confusion existed between early and moderate severity.

1.4 RESEARCH GAP

In the works we have surveyed, we found research on gait in clinical contexts generally, as well as specifically in KOA applications. In KOA work, there are cross-sectional studies, which can compare diseased and non-diseased groups, or longitudinal studies, which would compare the same people, over time, to measure disease progression or other factors. We wanted to look at how the same person would react to restriction, which guided us towards simulated restricted gait.

In this area, we previously mentioned the work of Hwang et al., who classified different simulated gait restrictions classes using machine learning [137] as similar to ours. Although the goal of their work is different than ours, there are enough similarities that their work is relevant in the context of the sensors used, and the collection procedure of data followed. The major differences between our work and theirs is the number of sensors (we use one sensor at the thigh and shank, along the sagittal plane), the multiple levels of restrictions, and the multiple types of activities performed. Also, we seek to use machine learning to classify the activities performed, as opposed to the restriction affecting the performance.

At the intersection of activity classification via machine learning and gait, we found a plethora of work, which tackle related problems in a variety of ways. We were particularly interested in IMU-based collection of gait data, and offline activity classification using a convolutional neural network. We identified many contributions in that area in 1.3.1.1, though in general, we found that the methodologies differed between our work and theirs, for example, number of sensors and sensor placements used, or activities performed.

Xia et al. collect IMU data from 13 sites on healthy and KOA participants, and attempt to differentiate between the healthy and unhealthy data[126]. They used a convolutional network as one of their models of choice, and they also provide the model input similarly to how we do it, but they convert their input to a grayscale image. It is also unclear if they used their handcrafted features, which are similar to the ones we use, to their convolutional neural network. Again, we are in a similar space, but we use our neural network to classify activities instead of gait condition.

Halim et al. used the data from the five IMUs in the ENABL3S dataset to try and predict the activity being performed by the participants[84]. Their feature extraction appears to have been simple, and when classifying on data without transition states between activities being included, they found accuracies of greater than 90% on SVM, random forest, and extreme gradient boosting models. This is closer to our use case, however, their models of choice are different than ours, as well as their feature extraction methodology, which is not completely detailed in their work.

To the best of our knowledge, we see two gaps that we attempted to fill with our work.

1. There is a lack of work done on simulated knee restrictions on gait over multiple activity types, and multiple restriction levels, in the context of gait analysis.
2. Can we minimally pre-process and handcraft features, and classify activities with a simpler convolutional neural network?

1.5 THESIS ORGANIZATION

The organization of the thesis is as follows:

- Chapter 2: Details the built data collection device, and how the accelerometer data from the device were validated against a gold-standard system.
- Chapter 3: Details the participant collection that used the device to capture various artificially restricted data.
- Chapter 4: Details the analysis of data, including the produced model, its results, and the parameters found to differentiate between severities of restriction.
- Chapter 5: Concludes the thesis and provides suggestions for future work, highlighting the limitations of our efforts.

1.6 CONCLUSION

This chapter has focused on offering the reader a context of the space in which we conduct our work, and highlighted relevant topics of research. Gait measurement and analysis are tools whereby we can objectively assess the quality of movement of individuals, and possibly detect and measure severity of disease. Although various diseases can affect gait, of particular interest to us was knee osteoarthritis. Relevant work in the field focuses on classifying the presence of the disease based on measured gait information, as well as recognising activities performed by diseased individuals. There is also work that simulates similar diseases artificially, through the use of orthoses or training, though we have found such work to be limited. Further, we find that work in the human activity recognition space that uses neural networks typically uses complex pre-processing, or complex models. Thus, we aim to address this gap, by producing the following work, which collects artificially restricted data at multiple levels from healthy participants, using a custom data collection device, and aims to find measures that detect the severity of restriction, as well as classify the activities being performed. The next chapter discusses the design and development of the data collection system and validates its accelerometer data against a gold-standard system.

2 EMBEDDED SYSTEM

2.1 INTRODUCTION

The intent of this project is to lay the groundwork for a system to help aid in disease detection, specifically gait-altering conditions such as arthritis, by collecting gait data from instrumented individuals[38]. By restricting the maximum flexion angle of the right leg during everyday activities using an orthosis, we hope to crudely recreate the gait of someone afflicted by knee osteoarthritis (or other gait-altering diseases), in one leg. A diagram of the restricted setup is shown in Figure 2-1.

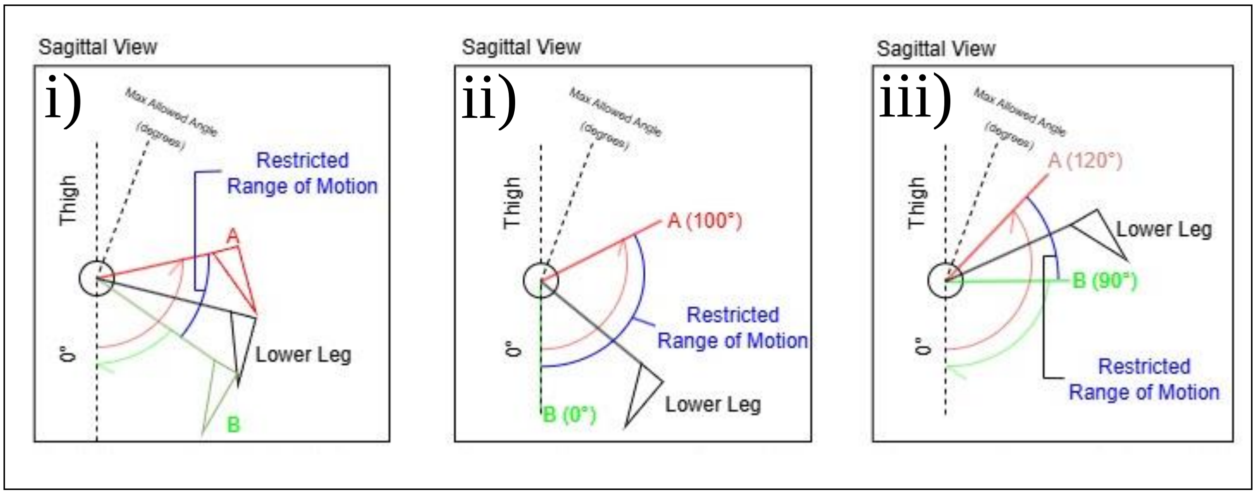


Figure 2-1: Diagram detailing the types of restrictions of which the system is capable. i) shows the general purpose of the system, restricting both flexion (moving lower leg towards the thigh) and extension (moving lower leg away from thigh) between angles A for flexion and B for extension. This situation is the general use of the system, and does not apply to our collection procedure. ii) offers an example of restricted flexion, and unrestricted extension (under the assumption that the leg will swing back to 0° at neutral). This is the general situation that will occur in our collection of data. iii) shows an example of both restricted flexion and extension, though this example would not offer any meaningful physical restriction in an ambulatory context.

Before we can create and apply a device directly to this application area, we first must collect data in an unrestricted gait setup, validate them against a gold-standard system, and then adapt the device to a restricted gait collection. Thus, we produce a system that can collect gait data from individuals, and easily move between an unrestricted and restricted gait collection.

Using off-the-shelf components, we devise a system that can collect movement information from sensors, and save them to on-board storage, which can then be extracted and analysed externally. The device is made to be small and untethered to external systems, to allow it to be used generally around the home.

To create a device that could be used both inside and outside the house, and be more affordable to implement, a custom solution was engineered. The system is outlined at a high level in Figure 2-2. The goal was to create a system that could be produced at a cheaper cost than the existing complete commercial systems, and store the results collected from the IMU sensors locally to the device itself. A time-tracking module would also be needed to timestamp data, in order to better organise and analyse the results. This system is a combination of hardware and software, and like other sensor systems, can be affixed to the body in different configurations depending on user goals, which is an important point that is elaborated upon in 2.2.3. We use this flexibility to collect data in both a completely unrestricted setup, and a varyingly restricted setup.

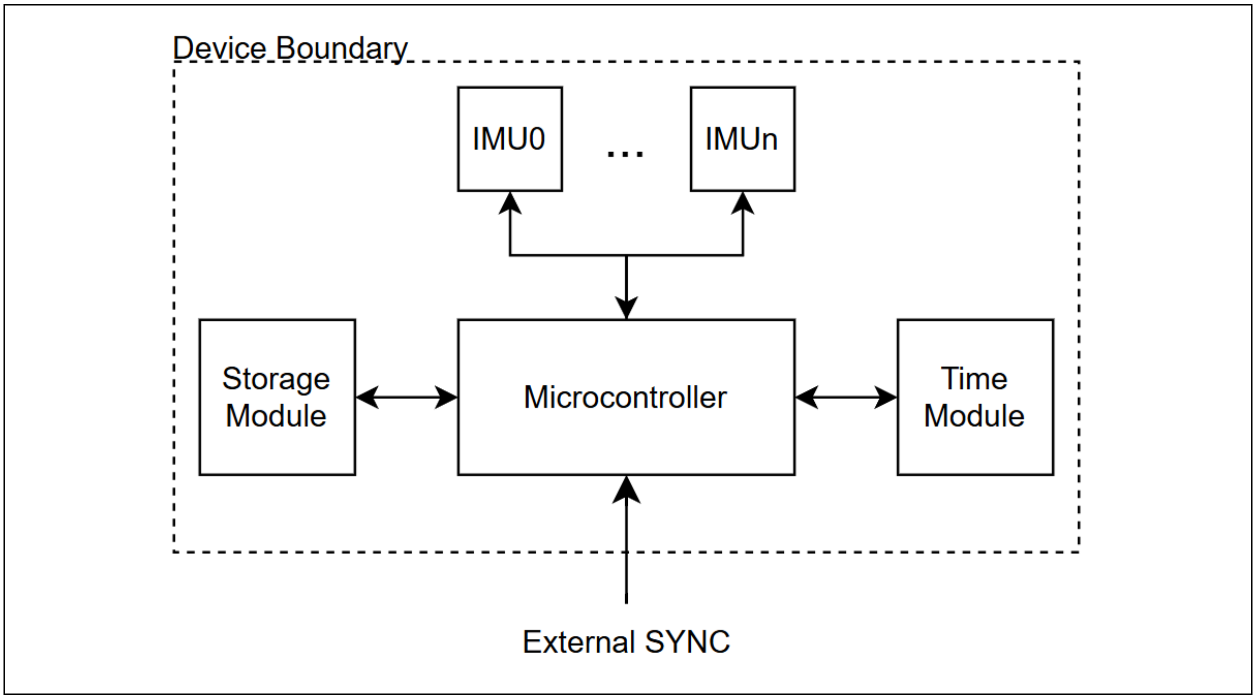


Figure 2-2: A high-level block diagram of the DCU to be built. The system must be able to communicate with IMU sensors, keep track of time, synchronise with external devices, and store the collected data. All of these requirements were not apparent from the initial stages, rather, they were introduced as they became necessary as the system was developed and refined.

2.2 DATA COLLECTION METHODOLOGY

2.2.1 EMBEDDED SYSTEM DEVELOPMENT

A data collection device ("DCU") needed to be produced in order to allow for collection of gait data from participants. The microcontroller for the device was the ESP32 SparkFun Thing Plus ("Thing Plus") (WRL-15663, SparkFun, Colorado USA), chosen for its ability to interface with the SparkFun series of I2C-based peripheral devices and components ("Qwiic Modules"), as well as the ease of programming provided via the Arduino IDE (ver.1.8.13) and compatible libraries. IMU sensors (SEN-15335, SparkFun, CO USA) compatible with the Thing Plus were chosen to allow for the collection of multiple streams of data, with the expectation that the machine learning model could use all these streams to aid in producing accurate predictions. The sensors were used to collect three dimensional accelerometer, gyroscope and magnetometer data. See Table 2-1 for details on the sensor. A compatible multiplexer (BOB-16784, SparkFun, CO USA) was needed to allow multiple sensors to communicate with the microcontroller.

Table 2-1: Technical specifications for the SEN-15335 from Sparkfun. This module uses the underlying ICM-20948 IMU. The details of the accelerometer, gyroscope, magnetometer, and any other relevant information is provided below, derived from [171,172]. The range of the accelerometer and gyroscope are independently configurable. Note, a 'g' is a unit of acceleration: 9.8 m/s^2.

SEN-15335	Tri-Axial Accelerometer	Tri-Axial Gyroscope	Tri-Axial Magnetometer
Range	±2g, ±4g, ±8g, ±16g	±250 °/s, ±500°/s, ±1000°/s, ±2000°/s	±4900μT
Resolution	16 bits	16 bits	16 bits
Sample Rate	0.27Hz to 4.5kHz	4.4Hz to 9kHz	100Hz
Wireless Range	Not Applicable		
Battery	Not Applicable		
Notes	Sensor also contains a temperature sensor. Sample rate will vary in practice based on how the sensors are configured and polled for data by the program that interfaces with them.		

Initially, the sensor output was printed directly to the computer, meaning that the device was tethered to a computer at all times, restricting use cases. Since the DCU was designed with collection of data of users in motion in mind, untethering it, and reducing its reliance on external devices was important. Thus, once reliable sensor functionality was achieved, local storage of the sensor data was explored.

At first, due to ease of implementation, a Qwiic module (DEV-15164, SparkFun, CO USA) was used to interface with the Thing Plus and a microSD card (SDSQUNB-032G-GN3MN, SanDisk, CA USA) however, this module proved to be unreliable and resulted in losses of data. A reliable solution was found in a SPI-based microSD card module (4682, Adafruit, NY USA) which was compatible with the 3.3V logic of the Thing Plus. It is important to highlight that the microSD card provided in Appendix A-1 (LMS0633032G-B2ANU, Lexar, CA USA) is experimentally known to be compatible with this particular module, and that other cards may not behave as expected, which was experienced with another microSD card.

To aid in processing of data, organization, and analysis, timestamping the collected data was considered a useful feature, and thus, an external real-time clock module was included, which interfaced with the Thing Plus via the I2C protocol. The clock was powered by its own 3V coin cell battery, to allow time to continue to be kept after the Thing Plus was powered off. The clock is initially set manually when a corresponding flag is set in the DCU's software.

The DCU was designed with the intent of collecting data simultaneously with an external VICON MX40 Optoelectronic Camera system (MX40, VICON, UK). The VICON system is connected to various other interfaces, including the MX Sync (MXSync, VICON, UK), which allows the system to easily communicate with external devices, via RCA ports. For our purposes, the VICON system allowed for a 0V to ~5V signal to programmed and transmitted to the DCU when the VICON system started collection of data. The signal definition is provided in Appendix B-1. The necessary circuit components were added to allow the 3.3V Logic DCU to process the 5V signal, and to accept a positive and negative wire from the RCA output of the VICON. Effectively, during its high pulse, the VICON synchronization signal will send a 3.3V signal to an input on the DCU, which will trigger collection on it. Once the collection starts, the sync cable is disconnected from the DCU. Although this was tested with a VICON in particular, any device able to replicate the sync signal and send it out via a two-wire (Power and Ground) signal should be able to interface with the DCU and cause it to start collecting. The DCU provides a rough estimate of time elapsed from the receipt of the of the sync signal to the acquisition of the first datum, which is intended to aid in aligning data from two separate sources.

Appendix A-1 and Appendix A-2 lists all the parts used in the creation of the DCU, with prices. In some cases, solder, extra wire, or other miscellaneous components may have been needed, but are not included in this list. Further, the cost of manufacturing a PCB or printing an enclosure are not considered here, though all the parts used for mounting the various boards to an enclosure, and a belt to tie to the wearer are listed. If a PCB is outside the scope of need, a perfboard can easily be used and drilled for mounting.

Additional miscellaneous circuit components such as a button, switch, and LED were added, to allow the user to interface with the system, and glean the status of the DCU during use. The button is connected such that it sends a 3.3V signal to the same input that expects a sync signal, thus it provides a manual way to start and stop the collection of data, without the need for an external system to sync with. At this point, a prototype system similar to the one shown in Figure 2-4 was produced using an electronic perfboard mounted to a thin, flexible, plastic sheet. The prototype is worn on the thigh, using the string that is threaded through the plastic sheet. To power the device, a 3.7V, 1000mAh battery (PRT-13813-B, Elmwood Electronics, ON Canada) was used. A major detriment of this version is the sync port, whose polarity can easily be flipped by accident, causing a short. Special care was taken to avoid this situation by colour-coding the wires.

The IMU sensors are attached to a flexible fabric sleeve (Cambivo Inc., California, USA) that was acquired online, via Velcro pads. The sleeve was used to allow for attachment points for the sensors, which was more convenient than attaching the sensors to a pant that the participant would need to wear, or onto the pants of each individual participant; the sleeve itself does not constrain the wearer enough to impact movement. The battery was placed in a fabric pocket that was also attached to the sleeve via a piece of Velcro. The pocket was made by folding a piece of fabric in half, and stitching the left and right edges closed. See Figure 2-4 for details. If the specific sleeve is not available at the time of reading, a similar sleeve can be found on Amazon with keywords like “knee sleeve”.

The IMU sensors are connected by wires, specifically SparkFun’s Qwiic Cables (PRT-17257, SparkFun, CO USA), with one end connected to the multiplexer, and the other end connected to the sensor. The sensors are placed laterally on the sleeve, as parallel to the sagittal plane (see Figure 2-3) as possible, as collinearly as possible on the thigh and shank, as shown in Figure 2-4. Note the sleeve is worn on the right leg. The positioning of the sensors is meant to capture both upper and lower leg segment data. Although other parts of the body can be involved in the activities of daily living[41], our concern was with ambulation and so, monitoring only the leg should be sufficient to differentiate between most activities. Further, the collected data were validated against the MX40 system, using a methodology that required joint angle calculation between the two segments, which was dependent on the placement of the sensors being placed on each of the segments. Further details of this validation work can be found in [173].

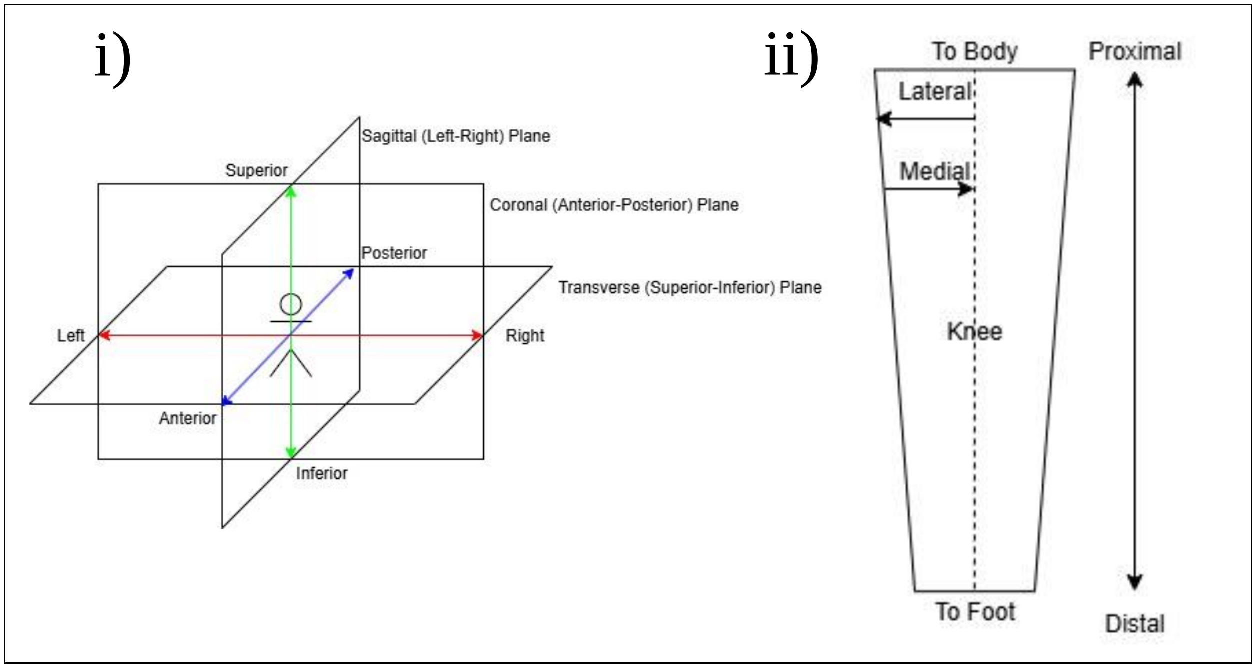


Figure 2-3: Physical orientation and direction terminology. i) Anatomical body planes, derived from [174,175]. The sagittal plane divides the body into left-right halves, the coronal plane divides the body into anterior-posterior halves, and the transverse plane divides the body into superior-inferior halves. ii) Description of the leg positions, derived from [174]. Moving towards the top of the leg is proximal, and distal is towards the foot; lateral movement is towards the side of the leg, and medial movement is towards the middle. We want to place our sensors collinearly along the proximal-distal axis, parallel to the sagittal plane.

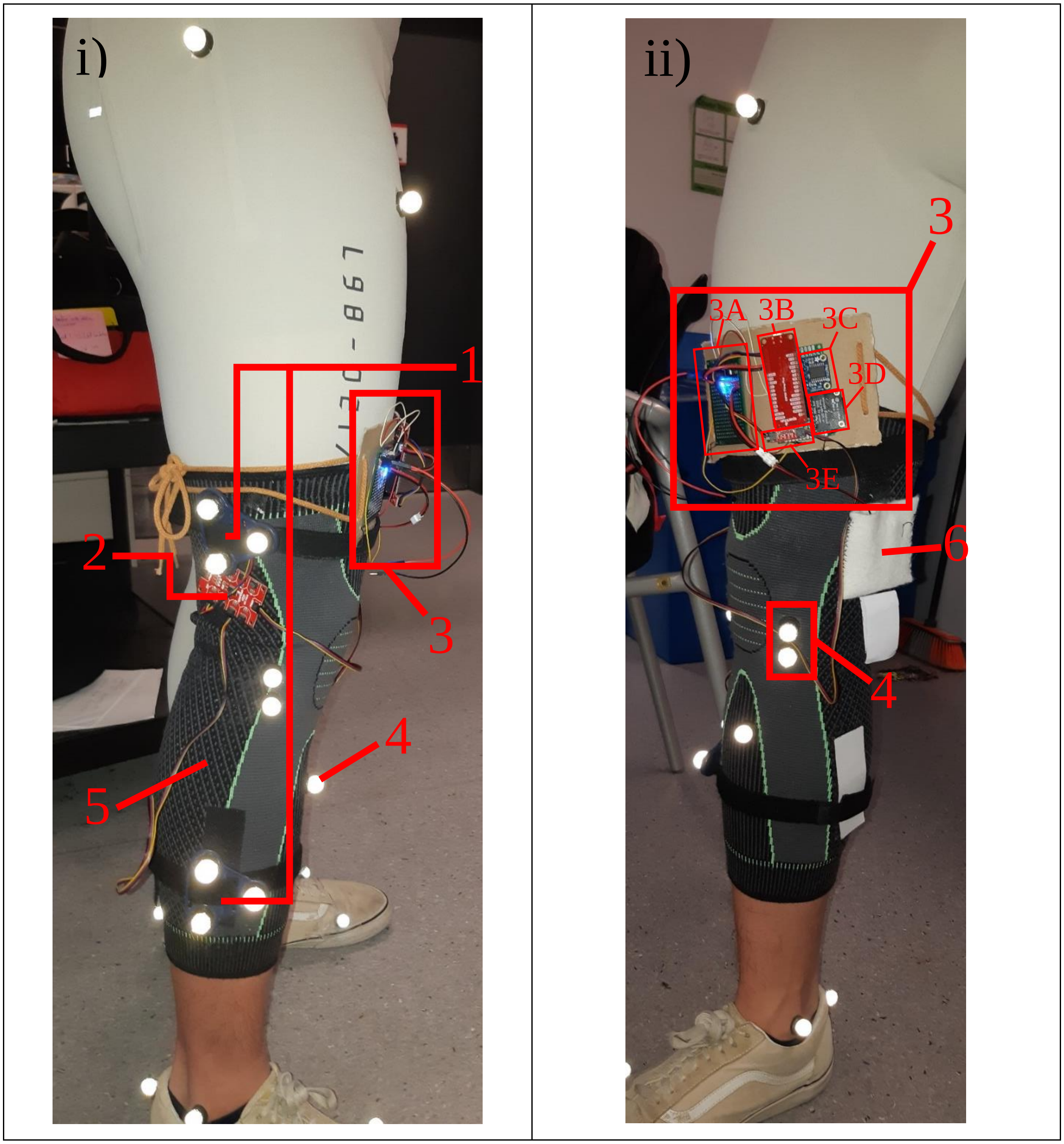


Figure 2-4: Participant wearing early prototype which was used to verify the validity of the DCU data relative to the gold standard VICON system. Each IMU is covered by a tri-marker group, so that each IMU and marker group is referencing approximately the same physical point when accelerations are derived. The participant performed various movements while instrumented to produce enough data to compare the DCU and VICON results. i) is the lateral view of the right leg, ii) is the medial view of the right leg. The labelled parts are as follows: 1. IMUs, covered by reflective marker clusters. 2. Sparkfun Qwiic MUX, allowing multiple sensors to communicate with the main microcontroller. 3. The main control boards. 3A. Board with ON/OFF switch, Start/Stop button, and synchronization port and wires. 3B. The main microcontroller, which communicates with and controls all the other components. 3C. The Real Time Clock module, powered by an external battery. 3D. MicroSD Module. 3E. Sparkfun Qwiic Adapter. 4. Reflective Markers tracked by the VICONs. Note that only a few markers are noted in the above figure, all such markers serve the same purpose. 5. Cloth sleeve. In this version of the system, the sleeve serves as little more than the mounting surface, as it does not hinder the movement of the wearer. 6. Battery and fabric pouch which contains it.

2.2.2 ENCLOSURE AND PCB DESIGN

Once the DCU was used to collect multiple trials of test data and verified, the design was translated onto a PCB, using KiCad (ver. (5.1.5)-3, release build), and sent out to be manufactured. The design of the PCB and overall circuit schematic are shown in Appendix C-1 and Appendix C-2. The PCB went through multiple iterations in order to reduce its size; the final board used two-layers, with vias to help keep the dimensions small. Minimizing size was important, as the DCU was designed with the intent of being portable and worn throughout the day.

After the physical dimensions of all the components were final, a custom enclosure, shown in Appendix D-1 and Figure 2-5, was designed and 3D-printed, to have an easy-to-use, completely encapsulated iteration of the DCU. The enclosure was modelled using Fusion 360 (ver. 2.0.16985 x86_64) and sliced using PrusaSlicer3D (ver. 2.6.1 + win64). To produce a lightweight and portable device, the dimensions of the enclosure were kept as tightly bound to the PCB dimensions as possible. Various layouts were tried, but ultimately, a layout wherein the components ended up "stacked" on top of the each other, with enough clearance to allow for assembly was finalized. The walls of the enclosure had multiple slits on all sides possible, to ensure the device stayed cool. No formal thermal analysis was conducted, so it is possible that the number of slits designed are in excess of what is required. To allow the DCU to be attached at the waist, a Velcro belt is fed through a slit designed on the underside of the enclosure.

A 3D-printed standoff was also created, shown in Appendix D-2, in order to allow the LED to be kept in a vertical position off of the PCB. This component is optional, and the LED leads can be bent to allow the component to lay flat on the board; however, if the LED is to be kept vertical, the standoff is necessary.

As the enclosure closes the PCB off from being accessed directly, some of the prototype components needed to be swapped out with panel-mounted components, so that the user could access them from the outside of the enclosure. Further, an RCA port was used for the synchronization signal, which removes the risk of making a mistake when using two wires directly. To ease assembly of the DCU, all the required screws are identical, with screws securing the PCB, the Multiplexer, and the lid of the enclosure. The battery is placed in a recessed area on the base of the enclosure, and was held in place by thread which was pulled through small holes placed near the battery area. Zip-ties may be used, but in practice, we could not secure the battery as tightly as desired with them. This iteration of the DCU is shown in Figure 2-5.

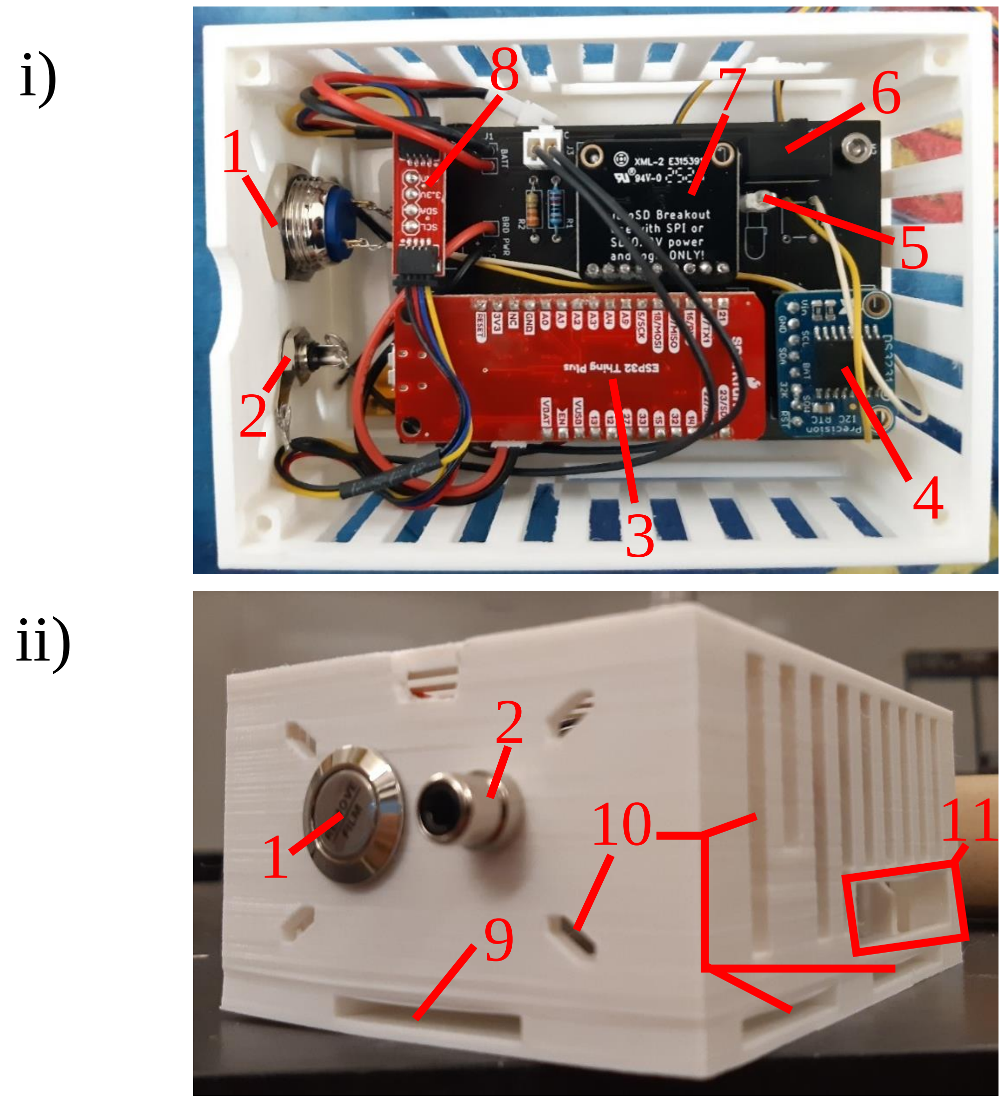


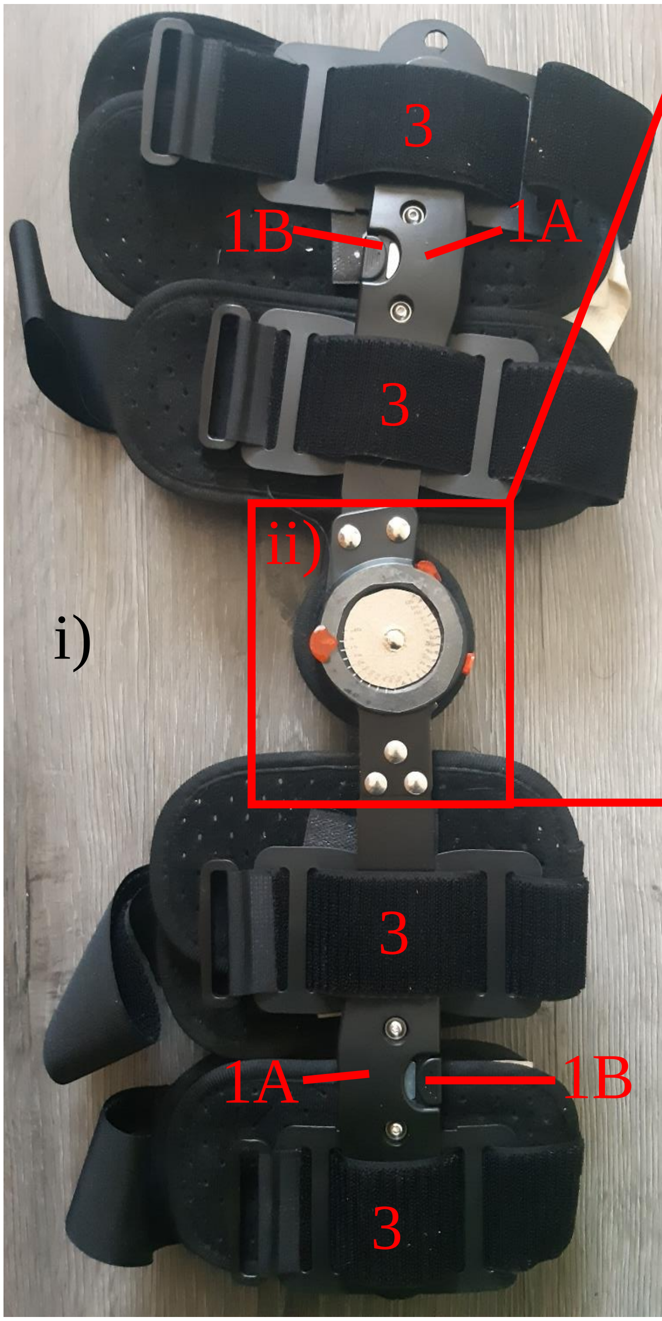
Figure 2-5: Fully assembled DCU, ready for use. i) the top view without the lid, ii) the outside of the device. The participant will attach the DCU to their waist via a Velcro strap passed through an opening (9) under the enclosure, while ensuring that the wires that connect to the IMUs run out of the floor-facing cut-out (11). The sensors should be worn on the right leg, with one on the thigh and one on the shank, with both sensors being roughly aligned along the proximal-distal axis, and kept parallel to the sagittal plane. The components annotated are as follows: 1. The panel-mounted Start/Stop button, for manually starting or stopping collection of data, if needed. 2. The panel-mounted VICON synchronisation port, for remotely starting or stopping collection of data, if needed. 3. The main microcontroller, which interfaces with all the components. 4. The Real-Time Clock Module for timestamping the data. 5. The Indicator LED, to notify user of system status. 6. The ON/OFF Power Switch. 7. The microSD card module, for writing the data. 8. The Sparkfun Qwiic Adapter, to allow the microcontroller to communicate with the IMUs. Note, the battery and Qwiic Multiplexer are not pictured, and are physically below the PCB shown above. 10. denotes the various ventilation cut-outs

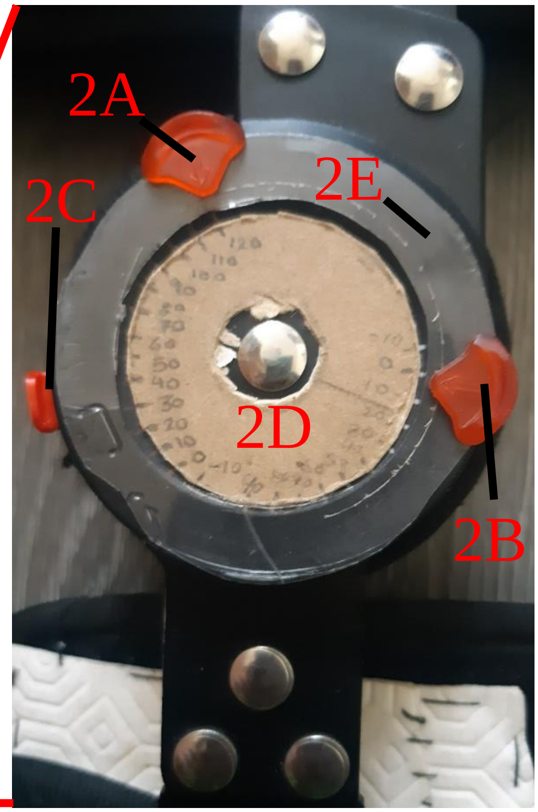
In this version of the prototype, since the DCU is worn at the waist, and not the thigh, more length of wire is needed to connect from the multiplexer to the sensors, and thus, multiple wires needed to be soldered together. In order to account for the differences in heights of possible participants, two 500mm Qwiic Cables were soldered together, which heat-shrink tubing used to cover the joint. Two of these metre-long cables were produced, for each of the two sensors used. In order to reduce movement of the excess length of wire during collection of data, an elastic band was worn on the leg, to hold back the wire. In practice, these wires appeared to be the cause of many issues, and so, two more pairs of wires were constructed, in the case that any of the wires were problematic during collection.

2.2.3 BRACE INSTRUMENTATION AND MODIFICATION

The DCU up until this point consisted of a system for collection of data with sensors that were mounted to a cloth sleeve. The sleeve was not meant to be intentionally restrictive, and allowed participants to perform movements naturally, beyond whatever limitation would be introduced by cloth bunching during flexion of the knee. The reason for this flexibility was that the purpose of the system until this point was simultaneous capture of data between the DCU and a VICON system – the underlying question being asked was "Does this system validate well against a gold-standard system?" Once the validation work, carried out by Nguyen et al. [173], was complete, the DCU could be used for the capture and analysis of further data. The specific intention of this additional capture of data was to address the real-world problem of gait-afflicting medical conditions.

Using the DCU, we aim to collect data from participants while restricting the maximum flexion angle of one of their legs. To accomplish this goal, a brace which allowed the user to restrict their maximum flexion and extension angle was purchased and instrumented (PIDAN TECH, Amazon.ca). The brace had a built-in adjustable lock which would allow the level of restriction to be selected as needed. The restricted movement data would be analysed to find if any substantial differences can be found to delineate between unrestricted and the varying levels of restricted data. The brace and its details can be seen in Figure 2-6 and Figure 2-9. In case this specific brace is not available at the time of reading this, similar braces may be found at Amazon, with keywords like "hinged knee brace immobilizer lock".





iii) A side view of the locking mechanism. 2A and 2B slide into notches (demarcated in gray) in the plastic body of 2. However, with motion, 2A and 2B can slide out of these notches and move with the swing of the leg, into other notches in the plastic, unintentionally changing the restriction of the brace. Thus, a plastic shim is used to add friction and prevent 2A and 2B from easily sliding out

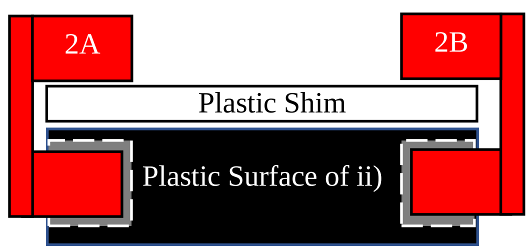


Figure 2-6: Restrictive brace, i) side view of un-instrumented side. 1A. The removable plastic brackets that hold the spring-based height adjustment mechanism (1B). To adjust the height, squeeze on 1B, and that segment of the brace will slide up or down, to adjust to other notches in the underlying metal body.1A served as the perfect site for instrumentation of the brace, shown in Figure 2-9. ii) is the restriction locking mechanism, and the location of the hinge in the brace. This is where every Participant's knee is located during every collection, so that the sensors are always at the same approximate position relative to every Participant's knee. 2A and 2B are the Flexion and Extension locks, depending on which side of the brace is being viewed. 2C is the disposable 3D-Printed lock, described in Appendix D-5. 2D is where the angular restriction values corresponding to each position of the lock is printed; this faded away, and so a replacement piece was produced and placed over top, as seen. 2E is a plastic shim, used to keep 2A and 2B in place during movement, described in iii). 3 are the Velcro straps used to tighten the brace to the Participant.

The brace is meant to be used by individuals of various heights, and thus, has a length-adjusting mechanism built-in. This mechanism is covered by a plastic bracket, which serves as a mounting point for the sensors, providing a reliable way to make sure the sensors were always at the same location on every participant, as long as the knee hinge was placed at the knee of each participant and the brace height remains constant (see PARTICIPANT STUDY). To alter the plastic piece to be the correct size to be able to accommodate the sensor, the original piece was measured, 3D modelled, and 3D printed, with the necessary screw holes built-in to mount the sensor. This part is shown in Appendix D-3. Further, a plastic cover for the IMUs was also 3D printed, in order to protect the sensors during use, also shown in Appendix D-4.

After instrumenting the brace, it was worn around the home, and on a treadmill at varying speeds, to determine if the brace would reliably remain in place. Even with the Velcro straps cinched as tight as possible without affecting the gait, the brace would eventually start slipping down the leg, meaning that the sensors would slowly have their position along the leg change as trials went on. To mitigate this issue, a non-slip material was stitched to the inside of the brace, along the pads that contacted the leg (see Figure 2-7), which, in combination with the Velcro straps, greatly helped the brace stay in position throughout the activities performed. The grip was best when the non-slip material made contact with skin directly; the sweat formed during activity helped to adhere the brace to the leg.

Figure 2-7: Anti-slip padding stitched onto inside of brace. This was made using non-slip shelf liner, purchased from a dollar store. Pieces of the liner were cut to fit the existing padding on the brace, especially in areas where the pads would make contact with the skin of the user, and stitched onto the padding. The knee pads were left without anti-slip padding to prevent any extra force being applied to the knee. The presence of this anti-slip padding made a major difference in ensuring the brace would hold to the leg and not slip during movements.

Once the brace was instrumented and stayed reliably on the leg, the built-in restricting mechanism was tested, to see if it could endure the forces applied during everyday activities, wherein a significant amount of weight would be applied at the knees. During testing, the adjustable locks broke, which necessitated the creation of new locks. To produce them, the locking mechanism was measured, and new locks were 3D modelled and printed (see Appendix D-5), with the intention of being disposable, so that if one lock broke, another one could be slotted in and the activity could resume, without needing the entire collection of data to be postponed until new locks are acquired. The locks have two grooves which serve as blocks to prevent the metal frame of the brace from swinging past the restriction level the lock is placed at. There are two additional grooves which accept the teeth in the hinge mechanism, allowing the lock to stay in place.

With this done, the instrumentation of the restricted brace was completed, as seen in Figure 2-9. Of particular importance, the figure shows the orientation of the IMUs when they are mounted to the brace (see ii) and iii) in Figure 2-9). In the case that future versions of these IMUs have different silkscreen printings, in the shown orientation of both IMUs, the marker dot on the main ICM20948 chip is on the top left. It is important to note that in practice, it was found that the actual directions of the accelerometer in the sensor were not commensurate with the silkscreen printings on the PCB. The discrepancy is shown in Figure 2-8, with the datasheet markings derived from the ICM20948 datasheet [172]. Note that the gyroscope markings are correct on the sensor, but are adjusted for the positive direction of the acceleration axes for consistency in the below figure. The validity of the datasheet's magnetometer directions could not be tested, and thus, users are cautioned to confirm before applying negative or positive interpretation to collected magnetometer data.

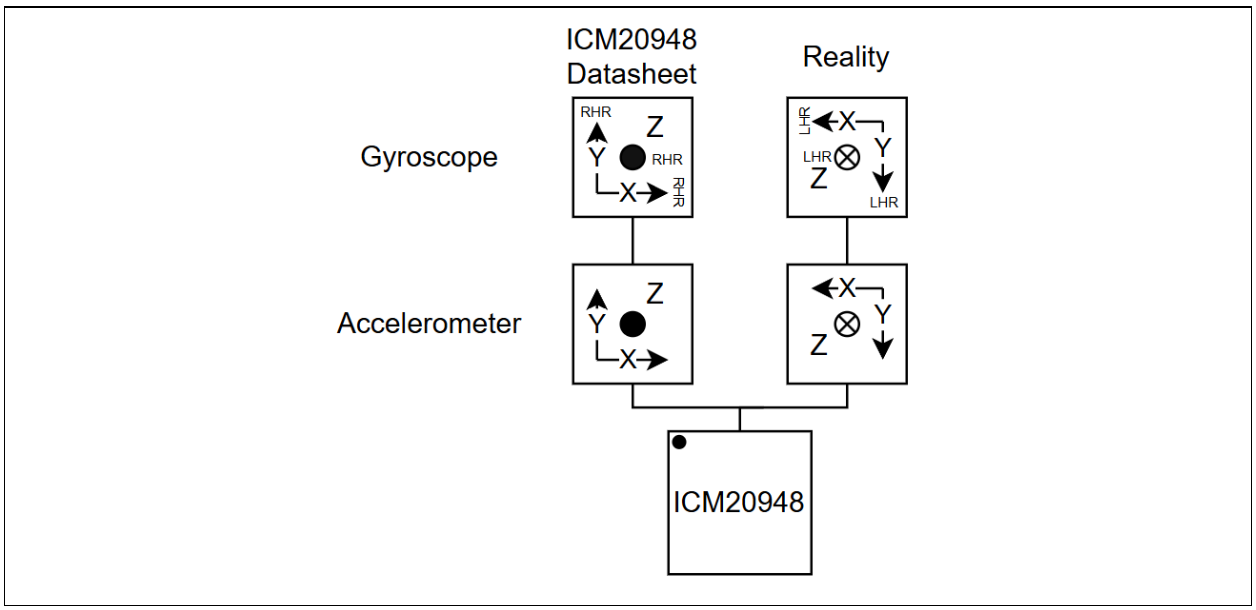


Figure 2-8: Axes orientation of the IMU sensor, relative to the marker dot of the ICM20948 chip on the sensor. Note that the silkscreen printed on the sensor (which is the same as the axes directions on the datasheet) correctly notates the gyroscope rotational axes, but not the accelerometer axes. Thus, for consistency, the correct accelerometer axes, and the positive gyroscope rotation relative to those axes are shown. Magnetometer axes were not able to be tested, and users are cautioned to confirm before assuming positive or negative values. RHR stands for right-hand rule, and LHR stands for left-hand rule.

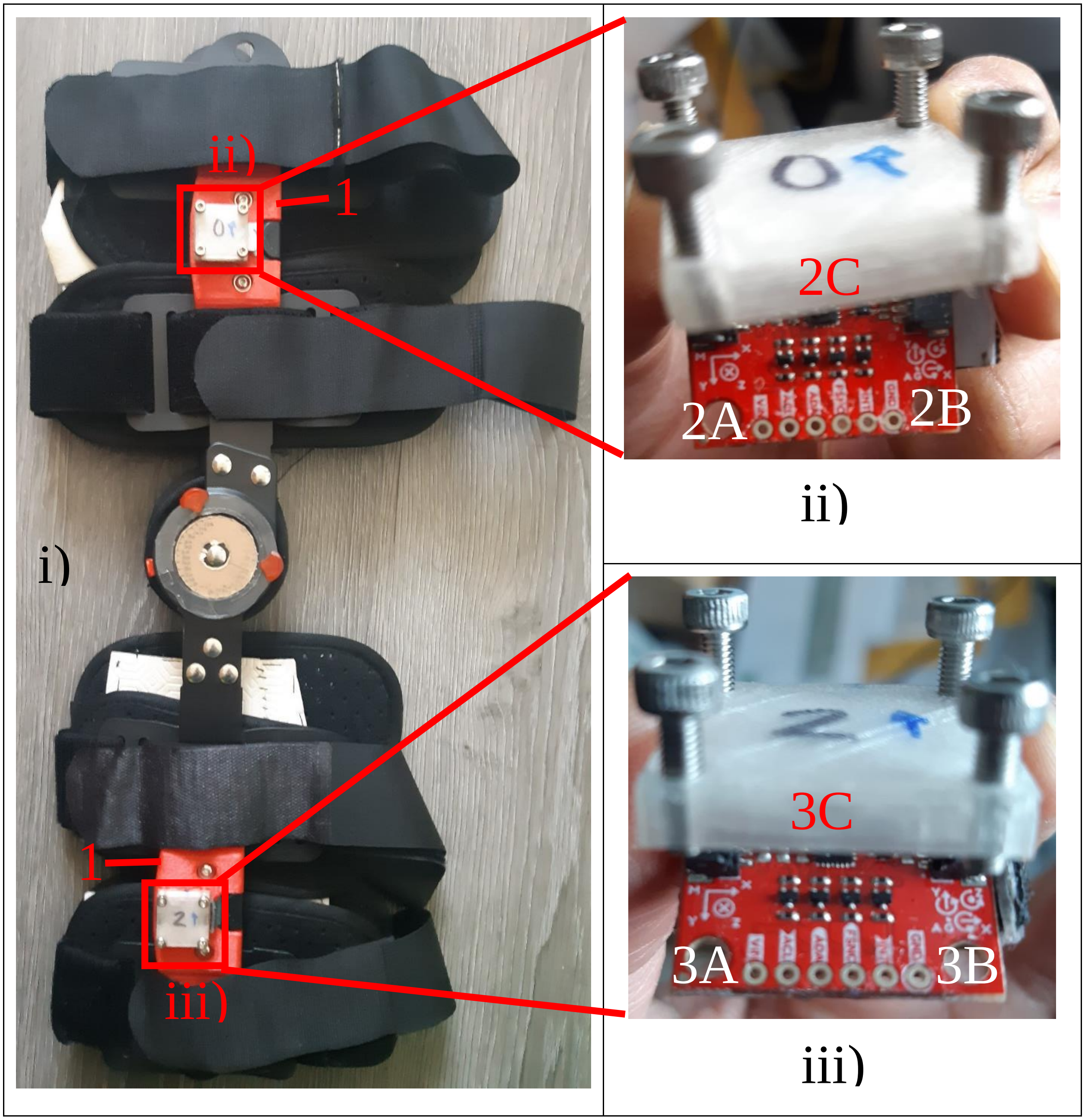


Figure 2-9: Fully instrumented restrictive brace. i) shows the entire brace, with the instrumented side in view. Only the lateral side of the brace is instrumented, and the brace is worn on the right leg. The various components of the brace are the same as the ones described in Figure 2-6, aside from the labels that are present in this figure. ii) and iii) are the thigh and shank IMUs respectively, sometimes referred to as IMU 0 and IMU 2. The close ups of ii) and iii) show the separate cover piece, 2C and 3C, which screw through the IMU mounting holes, onto the 3D-printed mounting bracket (1). Note the blue arrow pointing up on the 2C and 3C covers; this is the direction that points towards the head when the IMUs are mounted to the brace. More importantly, 2A and 3A show the coordinate system for the magnetometer, and 2B and 3B show the orientation of the accelerometer and gyroscope.

2.3 DCU ALGORITHM DEVELOPMENT

The first iteration of the algorithm was a straightforward, single-core design. During this phase of development, test collections of dummy data were used to determine if the collection was occurring at an appropriate rate (at least twice 20Hz[176]), without any regard to the actual data being collected. It is through these collections that the algorithm was determined to be performing at a collection rate that was subpar, and thus, a dual-core strategy was adopted.

The first algorithm would simply poll each sensor for data, and write those data immediately to file as they were retrieved. This method was ineffective, as the time between collecting consecutive samples from the IMUs was 200ms, and so the device would be unable to meet the required rate.

The next algorithm buffered data, and wrote them together, after a predetermined number of points had been buffered. At minimum, the algorithm would bundle together one datum from each connected sensor and write it to the file. This method was briefly explored, due to the promising reduction of delays between points, however, the amount of time taken to write a group was still large. The strategy was to attempt to increase the size of the group until the quantity of data covered 2 to 3 minutes of activity, which was the expected length of participant trials per activity. However, memory constraints prevented this strategy from working, motivating the need for a dual-core approach.

The next approach was to create an algorithm that runs on a simple dual-core architecture, wherein each core is responsible for a different task, and the cores can pass messages to each other via a queue. The maximum size of each message, and the number of messages that can be held on the queue were determined experimentally, and were chosen to be values that proved to not cause erratic software behaviour in practice.

With this algorithm, after multiple trials of walking on a treadmill and around the house, it appeared that sensors were arbitrarily failing, leading to files with only one sensor providing data. When losses occurred, they would remain until the system would be reset, meaning that all data produced after the sensor failure would be lost by that sensor. Reducing cable motion by taping them to the body was useful to help reduce some losses, but this only partially helped fix the problem. It seemed unlikely to be able to stop the sensor failures altogether, as I2C was not intended to be used for such long distance communication [177] and thus, it may have been causing the failures. However, it was possible to reset the sensors in software, thus, whenever the sensor would fail, the software could detect it, and reset the sensor. This caused approximately 100ms of loss of data in the best case, due to time needed to reset, after which the sensor would resume functioning correctly. This way, sensor failures would not result in complete losses of data and wasted time and effort from the user.

The final algorithm that runs on the DCU and allows the system to capture sensor information is outlined below. The system is able to collect data at approximately 80Hz.

The logical steps of the algorithm are broken down as follows:

1. **Startup:** The DCU initializes communication for the microSD card explicitly, and implicitly initializes communication for the RTC and I2C peripherals. It also defines the functions of various GPIO pins. During this stage, the user can:
    a. Set the RTC Time if the appropriate flag has been set in advance
    b. Configure the granularity/range of the IMU sensor. The accelerometer and gyroscope can be configured, but the magnetometer has a fixed range.

    By the end of this stage:
    - The microSD card has a newly created file ready to be written to
        - Files are named as DLX_YYYY-MM-DDTHH-MN-SC.dat, where X is an integer, starting at 0 and incrementing by 1 every new file, and HH is an hour from 0 to 23, MN and SC are minutes and seconds from 0 to 59.
    - The IMUs are set up with the newly set (or previously configured) granularity/range settings
    - The RTC is either configured with the newly set time, or continues to hold the time set prior to the DCU startup.
    - Critically, an inter-core queue (ICQ) is created, which will be used to for inter-core communication. The LED stays off during this stage.

2. **Dual Core Creation:** Two cores are initialized, with separate tasks running on each. Both cores use the queue created in the **Startup** stage to communicate with each other.

During this stage, the LED remains off. The functionality of each core is defined briefly as follows:

   a. **Core A:** This core is responsible for collecting data from the sensors, packaging them together, and once sufficient data are collected, place them on the queue.
   b. **Core B:** This core is responsible for pulling data from the queue, and writing them to the file.

3. **Standby:** The DCU is now waiting for the user to begin collection. The LED will be solidly on, to indicate this phase.

4. **Data Collection:** When the user presses the "Start/Stop" button from the **Standby** stage, the DCU begins to collect data. The LED will periodically blink to indicate that the algorithm is in this stage.
   a. **Core A:** Each of the IMUs connected to the device are polled for data.
      i. **If data are found**: they are pulled from the sensor, and placed into a buffer, with an additional timestamp field prepended to the data.
      ii. **If data are not found**: the system is attempts to reset the IMU that does not have data available. The IMUs should always have data available, and so having none implies some sort of bad state, which is fixed by a reset. The reset process is mostly invisible to the user. After resetting, the IMU is set up with the correct parameters (which were used in **1. Startup, a.**), and the next IMU is polled for data (**4. Data Collection, a.**)

   Once the buffer is filled to a predetermined size, which was determined to be 35,000 characters experimentally, the buffer is placed in the ICQ for Core B to collect from, and signals Core B that there is an item on the queue. In order to maintain a data-rate that is commensurate with at least twice the minimal data-rate required (2 x 20Hz[176]), the algorithm uses a double-buffered approach. Thus, once the core places the buffer on the ICQ, it immediately starts filling a second buffer with new data.

b. **Core B:** Once the signal from Core A is received, this core will dequeue the item on the queue, and begin to write it to the file that is opened in the **Startup** stage.

5. **Data Cessation:** When the user presses the "Start/Stop" button from the **Data Collection** stage, the DCU closes data collection and data writing operations. This is done differently for each core.
    a. **Core A:** Once this core receives the signal that the user wishes to halt data collection, it stops polling the IMUs and gathers all the data collected from the duration of the most previous data enqueue to when the core receives the halt signal. Once these data are gathered, the core places this final item on the ICQ, notifies Core B that the final item is available, and returns to stage 3 (**Standby)**.
    b. **Core B:** Once this core receives the signal that the user wishes to halt data collection, it stops checking the queue for data repeatedly, and waits to receive the final item signal from Core A. Once received, it dequeues the final item from the ICQ and writes it to the file. The core then adds a timestamp for the end of the file, as well as some metadata:
        i. The amount of time elapsed from the user pressing the "Start/Stop" button to the collection of the first data point. (This amount of time can be useful when trying to synchronize parallel data collections)
        ii. The IMU granularity/range settings used during this file.

    The core then closes the file, to allow the data to be written to the file. A timestamp is placed on the newly created file to allow for easy organization and analysis. This core will initialize a new empty file for writing data, and then flash the LED multiple times to signal the end of this stage. The core then enters into stage 3 (**Standby**).

2.4 DATA VALIDATION

2.4.1 DATA COLLECTION

Some preliminary testing was done with the DCU to ensure that data could be consistently and reliably collected in a static, resting device state. This was done by simply keeping the DCU flat on a surface, with the sensors placed on the soft-cloth sleeve, normal to the surface. The device was left on and collecting for hours at a time, sometimes overnight, to test for long term system stability. The ultimate goal of the DCU is to be used in everyday life, and thus, having it run for hours at a time, even in a static case, was a prerequisite to that goal.

After data were confirmed to be collecting reliably, the next step was to determine if the data were reflective of reality, and if the device would perform during active collection. In order to accomplish this, data collections were done simultaneously with the VICON MX40 system and the DCU with the soft-cloth sleeve, with the goal being to see the same acceleration patterns from both data sources. Using a prototype version of the soft-cloth version of the DCU, a participant was both instrumented with VICON markers and the DCU. A carefully measured out placement of the IMUs on the sleeve was not undertaken; the IMUs were placed aligned to the proximal-distal axis on the right leg, kept as parallel to the sagittal plane as possible (See Figure 2-3). The participant wore skin-tight pants in order to prevent movement of the retroreflective markers, and running shoes so that movements could be performed naturally and comfortably. The placement of the markers is described in [173]. The participant performed the activities as described in Table 2-2, on a StarTrac ∈ TRX treadmill (TREX1203-LQ1183, StarTrac). The activities chosen were based mostly on healthy ambulation, with sit-to-stand included as an activity of daily living. Squats were chosen because the activity positions the legs in a similar way to sit-to-stand, which will be of interest during data analysis. Lunges were included as an activity which had large joint angles between the upper and lower leg, which was considered possibly useful for joint angle validation work, such as that undertaken in [173]. The activities were performed initially with one participant, and the movement speeds were chosen to their comfort. All the activities were performed on the treadmill, with sit-to-stand being done by placing the chair behind the participant once they had entered the capture volume and stepped onto the treadmill. The treadmill speeds were chosen by the participant, using the treadmill's mi/h selector. They appear arbitrary in km/h, but correspond to 2.0mph (3.22km/h) and 3.7.mph (5.95km/h) for slow and fast speeds respectively. An instrumented participant can be seen in Figure 2-4. The lab space can be seen in Figure 2-10.

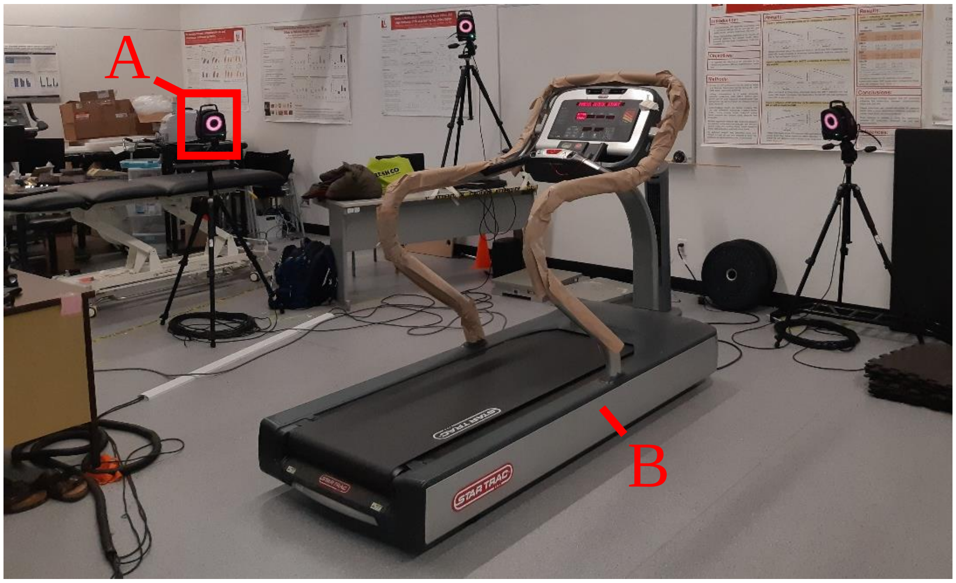


Figure 2-10: Validation collection area. A is a VICON camera, meant to capture movements of reflective markers when in the capture volume. B. is the treadmill. A chair was also intermittently placed in the capture volume when the sit-to-stand activity needed to be performed. Only the data analysed for the sake of IMU validation in this chapter were captured in this lab space.

Table 2-2: Activities performed to validate DCU data against VICON data, using the unrestricted, soft-cloth sleeve. The walk speed is 2.0mph (3.22km/h), and the run speed is 3.7mph (5.95km/h).

Activity	**Duration**	**Number of Trials**	**Notes**
Walk	2 minutes @ 3.22 km/h	3 Trials	-
Run	2 minutes @ 5.95 km/h	3 Trials	-
Sit-To-Stand	10 Repetitions	3 Trials	Participant sits in a chair, and rises from the chair, repeatedly
Squats	10 Repetitions	3 Trials	Participant squats as low as comfortable and rises, repeatedly
Lunges	10 Repetitions	3 Trials	Participant lunges with alternating legs, left leg goes forwards first.

The collection followed the steps as outlined, starting from a fully instrumented participant:

1. DCU is connected to sync wires from VICON.
2. VICON collection is started from its Host PC. Synchronisation signal is sent to the DCU, and it also starts collecting
3. The DCU is disconnected from the sync wires
4. The participant enters the capture volume, and performs three flexions of the instrumented leg (to more easily help align the VICON and DCU data later)
5. The participant performs the pre-determined activity
6. VICON collection is stopped
7. DCU collection is stopped

Initially, the activities were all completed with sensor range values of ±2G for the accelerometer, and ±250dps for the gyroscope. When the data were analysed after the collection, it was evident that the running trials exhibited clipping, and so the running trials were all redone, with the sensor range changed to ±8G for the accelerometer and ±1000dps for the gyroscope.

2.4.2 DATA COMPARISON

After the data were collected, the data were prepared for comparison. Firstly, a low-pass 4th order Butterworth Filter was applied to both datasets, with a cut-off frequency of 20Hz[176]. Once the data were cleaned, the IMU data, which were sampled at ~80Hz, were interpolated to 100Hz, in order to facilitate an easier comparison between the VICON and IMU data. Lastly, the difference between the position data of the VICON and the acceleration data of the IMU needed to be accounted for.

In order to compare the X, Y and Z positional and acceleration data, the VICON positional data were converted to acceleration, by taking their second discrete derivative. This was preferred over integrating the IMU acceleration data twice in order to avoid any error being integrated over. Additionally, due to the arbitrary alignment of the VICON and IMU coordinate systems, it was necessary to use a vector sum to compare the results between the two methods, otherwise, the component accelerations for X, Y and Z would not be relative to the same directions. This is shown in Figure 2-11. Special care was taken to apply an additional acceleration of gravity to the computed Z acceleration of the VICON, as the system is unaware of gravity – only measuring position in a defined coordinate system. Lastly, the signals are made to be the same length by trimming the end of the longer signal to the length of the shorter one.

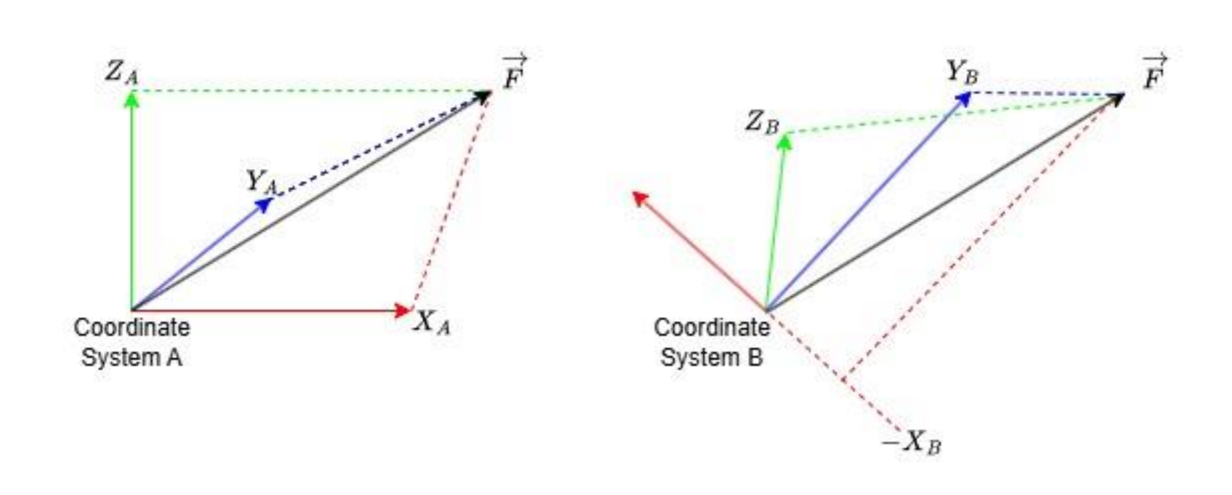


Figure 2-11: Two coordinate systems showing the same force, with different X, Y, Z component vectors. The coordinate system on the left and the coordinate system on the right have different basis vectors representing the X, Y and Z dimensions, but, for the same force applied in both systems, the vector sum is the same. Using this fact, we do not have to worry about the alignment between the VICON and IMU coordinate systems, we can simply consider their vector sum results instead of the independent forces in each dimension.

Once these steps were applied, the data were plotted in the time domain and the frequency domain, to see what the agreement was. In time, the IMU vector sum appears to lag the VICON vector sum, even after the DCU measured button-press-to-first-data delay duration is added. In most cases however, the additional lag is only a few hundredths of a second, i.e. a few frames of collection. This lag is hypothesized to be additional delay time between the VICON collection starting and the sync signal reaching the DCU. In most collections, it can be visually seen that the waveforms for both the datasets match each other across the different activities (see Figure 2-12 and Figure 2-13). To quantify this agreement, RMS and correlation coefficient values were calculated across the VICON and IMU data, for both thigh and shank. The vector sums are trimmed from the front by roughly 10 seconds (varying by activity and trial) to remove any movement prior to the three synchronisation flexion/extensions. Both signals are then independently scaled to be between 0 and 1, so that comparisons have the same sense of scale. Lastly, the IMU vector sum is shifted by multiples of 1 frame (0.01 second) to address the aforementioned lag. In general, these metrics showed mostly good agreement, with difficultly matching thigh data during two squats trials, and one poor sit trial (see Table 2-3). These issues may have been due to variances of those particular trials, as opposed to the system itself, considering the agreements across other activities. When considering the time series of these data, we see that the VICON reports consistently higher values for the squat thigh data in those trials, and that the VICON and IMU data for the poor sit trial were difficult to align. Note also that a Lunge trial has been omitted – the VICON data for that particular trial had data that were too damaged to be compared.

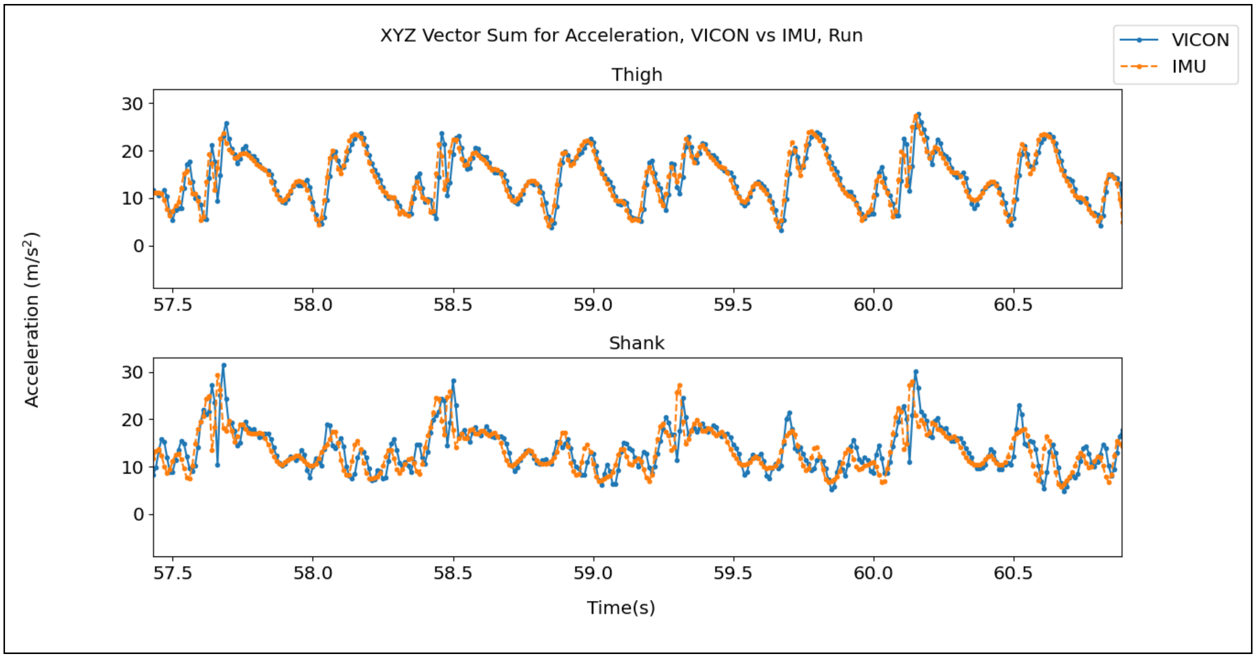


Figure 2-12: Time domain vector sum plot of thigh and shank IMU data against VICON thigh and shank data, for a trial of the "Running" activity (5.95 km/h). The top graph shows the time series thigh data, and bottom graph shows the time series shank data. Prior to vector sum calculation, the IMU and VICON data have both been 4th order Butterworth filtered, with low-pass cutoff of 20Hz. The IMU data have been interpolated up to 100Hz from their native 80Hz sample rate. Note the visual agreement between the shape of the IMU and VICON vector sums.

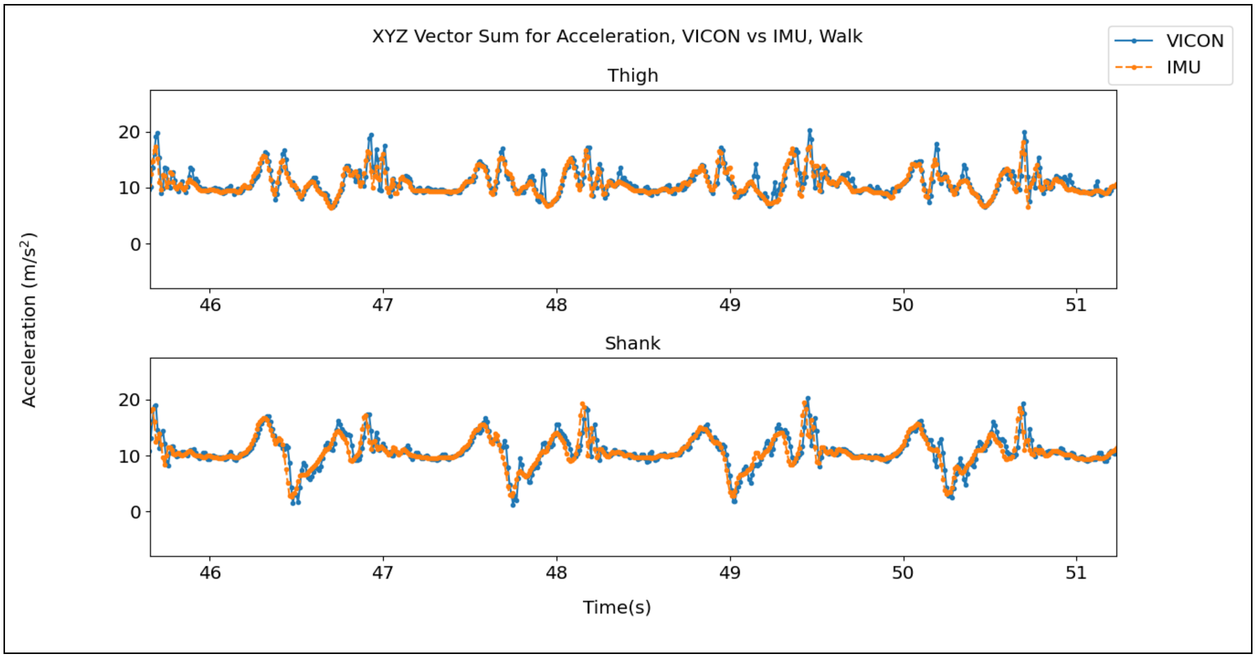


Figure 2-13: Time domain vector sum plot of thigh and shank IMU data against VICON thigh and shank data, for a trial of the "Walking" activity (3.22 km/h). Prior to vector sum calculation, the IMU and VICON data have both been 4th order Butterworth filtered, with low-pass cutoff of 20Hz. The IMU data have been interpolated up to 100Hz from their native 80Hz sample rate. The VICON vector sum appears to regularly report higher peaks than the IMU vector sum.

Table 2-3: RMS and correlation metrics for IMU and VICON time series vector sum comparisons. The front of the data are trimmed by 10 to 15 seconds, depending on activity, with the synchronisation dips untouched. Vector sums were scaled between 0 and 1 to give the RMS values a sense of scale. The IMU vector sum is shifted by steps of 0.01 second (1 VICON frame), and the metrics are calculated. The results of the best shift are shown below. Sit trial 2 was offset from the VICON data significantly and thus needed a larger shift, though the correlation is still poor. The correlations of thigh data for squat trials 2 and 3 were also relatively poor.

Activity	Trial	IMU	Trimmed (# of Points)	IMU Vector Sum Shift	RMS	Correlation
Lunge	1	Thigh IMU	1200	2	0.031	0.948
Lunge	1	Shank IMU	1200	2	0.084	0.879
Lunge	2	Thigh IMU	1200	1	0.024	0.952
Lunge	2	Shank IMU	1200	2	0.039	0.869
Run	1	Thigh IMU	1000	3	0.037	0.972
Run	1	Shank IMU	1000	3	0.065	0.901
Run	2	Thigh IMU	1000	2	0.035	0.973
Run	2	Shank IMU	1000	2	0.053	0.920
Run	3	Thigh IMU	1000	1	0.035	0.975
Run	3	Shank IMU	1000	1	0.052	0.924
Sit	1	Thigh IMU	1200	2	0.189	0.776
Sit	1	Shank IMU	1200	3	0.050	0.911
Sit	2	Thigh IMU	1200	148	0.071	0.113
Sit	2	Shank IMU	1200	171	0.069	0.386
Sit	3	Thigh IMU	1200	1	0.039	0.861
Sit	3	Shank IMU	1200	2	0.045	0.894
Squat	1	Thigh IMU	1200	2	0.095	0.786
Squat	1	Shank IMU	1200	3	0.067	0.894
Squat	2	Thigh IMU	2000	1	0.073	0.673
Squat	2	Shank IMU	2000	2	0.034	0.877
Squat	3	Thigh IMU	1000	1	0.121	0.454
Squat	3	Shank IMU	1000	2	0.023	0.885
Walk	1	Thigh IMU	1500	2	0.053	0.916
Walk	1	Shank IMU	1500	3	0.040	0.959
Walk	2	Thigh IMU	1200	1	0.078	0.919
Walk	2	Shank IMU	1200	2	0.048	0.932
Walk	3	Thigh IMU	1200	0	0.042	0.945
Walk	3	Shank IMU	1200	1	0.037	0.953

When plotting frequency data, the scipy `fft` method [178] was applied to the time series vector sum data. As was the case for the time series comparisons, here the time series is also trimmed until the start of the synchronisation movements, the IMU vector sum is shifted, and then the Fourier transforms are taken. After the transform, the absolute value is taken to incorporate the imaginary and real parts of the result together. These results have their 0Hz DC component zeroed, after which, the transform is scaled between 0 and 1. In the frequency domain, the spectra of the VICON and IMU also match quite well using the same RMS and Correlation metrics (see Table 2-4), though this was not initially the case.

As seen in Figure 2-14i), the VICON vector sum spectra appear to have a sinc-like distortion added to it. This distortion is caused by a strong series of peaks at the beginning of the vector sum for the VICON data, which are produced by the filtering process, and are not present in the raw VICON data. A filtering-produced peak is present in the IMU vector sum as well, but to a lesser extent, which, coupled with the lack of additional transformations that are needed for the VICON positional data, leads to a smaller effect in the IMU spectra. These distortions were removed by the previously mentioned front-end trimming, which removes the peaks in time, as seen in Figure 2-14ii).

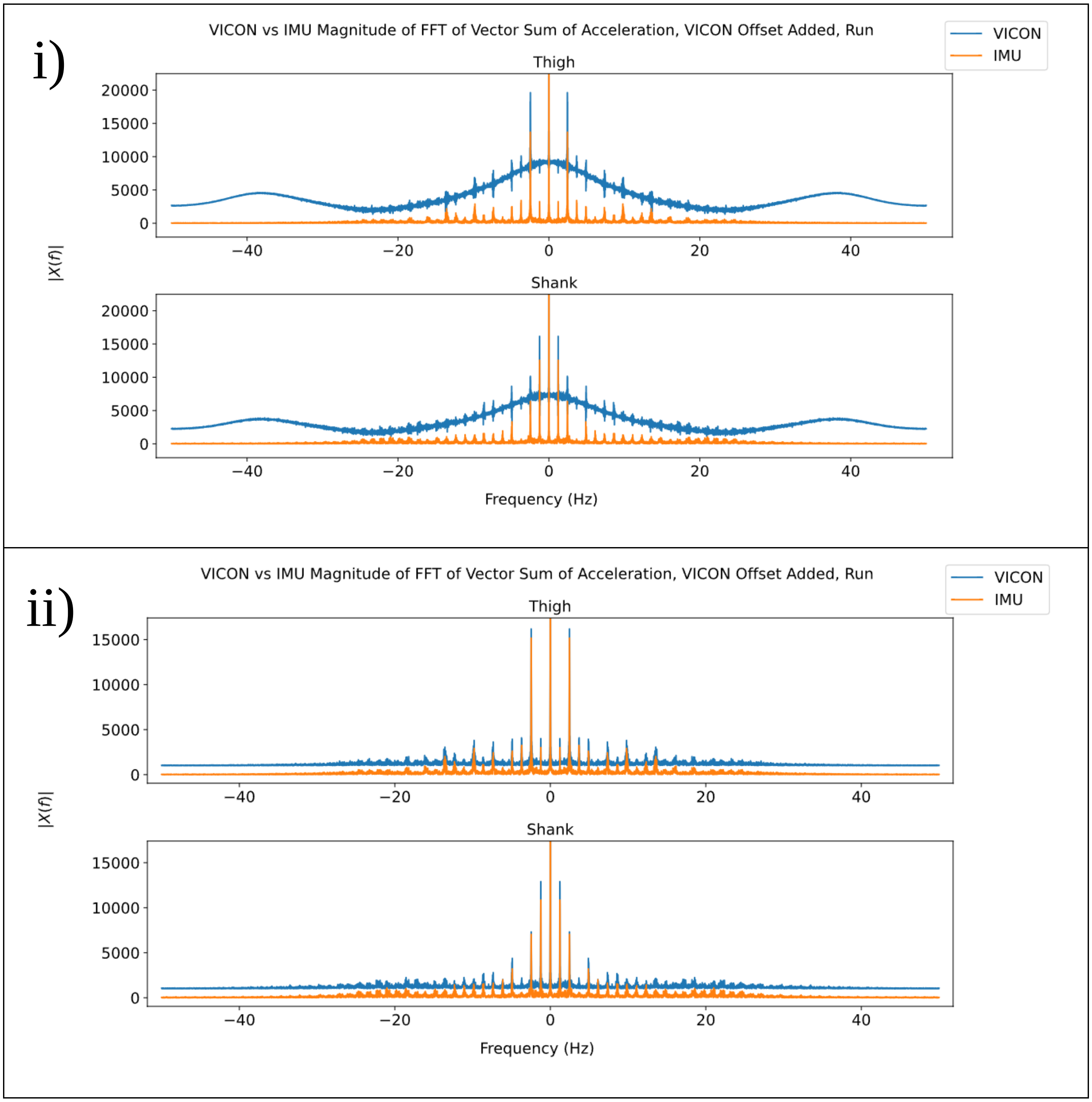


Figure 2-14: Frequency domain vector sums comparison before and after cleaning. i) The frequency domain of the vector sums of the IMU and VICON data for a trial of "Running", without the initial time series peaks being cut out. These peaks produced strong sinc-like distortions in the frequency domain, which are heightened by the processing steps taken to transform the individual X, Y, and Z positions into accelerations. ii) Frequency domain for the same data with the initial peaks removed (10 seconds of data). The IMU and VICON spectra are so similar that an offset is required to view them more clearly.

Table 2-4: RMS and correlation metrics for IMU and VICON Fourier-transformed vector sum comparisons. The front of the data were trimmed by 10 to 15 seconds depending on the activity, leaving the synchronisation dips untouched. Fourier transforms were scaled between 0 and 1 after zeroing out the 0Hz component to give the RMS values a sense of scale. The IMU transform was shifted by steps of 0.01 second (1 VICON frame), and the metrics were calculated, however, unlike in time, the shifting offered no benefit in frequency. Thus, the least shifted version of the transform was used. Sit trial 2 was offset from the VICON data significantly and thus starts at a larger minimum shift, though the correlation is still relatively poor. The correlations of thigh data for squat trials 2 and 3 were also relatively poor. These mirror the findings for the time series comparisons.

Activity	Trial	IMU	Trimmed (# of Points)	IMU FFT Shift	RMS	Correlation
Lunge	1	Thigh IMU	1200	0	0.020	0.972
Lunge	1	Shank IMU	1200	0	0.039	0.934
Lunge	2	Thigh IMU	1200	0	0.018	0.976
Lunge	2	Shank IMU	1200	0	0.029	0.929
Run	1	Thigh IMU	1000	0	0.005	0.988
Run	1	Shank IMU	1000	0	0.009	0.967
Run	2	Thigh IMU	1000	0	0.005	0.988
Run	2	Shank IMU	1000	0	0.009	0.973
Run	3	Thigh IMU	1000	0	0.004	0.989
Run	3	Shank IMU	1000	0	0.008	0.974
Sit	1	Thigh IMU	1200	0	0.064	0.865
Sit	1	Shank IMU	1200	0	0.032	0.948
Sit	2	Thigh IMU	1200	100	0.076	0.628
Sit	2	Shank IMU	1200	100	0.069	0.718
Sit	3	Thigh IMU	1200	0	0.036	0.913
Sit	3	Shank IMU	1200	0	0.036	0.944
Squat	1	Thigh IMU	1200	0	0.059	0.829
Squat	1	Shank IMU	1200	0	0.040	0.939
Squat	2	Thigh IMU	2000	0	0.075	0.761
Squat	2	Shank IMU	2000	0	0.041	0.928
Squat	3	Thigh IMU	1000	0	0.088	0.556
Squat	3	Shank IMU	1000	0	0.043	0.931
Walk	1	Thigh IMU	1500	0	0.016	0.946
Walk	1	Shank IMU	1500	0	0.006	0.986
Walk	2	Thigh IMU	1200	0	0.011	0.959
Walk	2	Shank IMU	1200	0	0.008	0.972
Walk	3	Thigh IMU	1200	0	0.014	0.966
Walk	3	Shank IMU	1200	0	0.008	0.982

2.5 CONCLUSIONS

Current motion capture based gait measurement systems, such as VICON system, are large and cumbersome, making it difficult to monitor the gait of an individual in their day to day lives, inside the home, and especially outside the home. Using portable IMU-based solutions as an alternative to these more rigid systems was of particular interest, due to the link between the acceleration provided by an IMU and the position data recorded by a VICON system, making validation straightforward. Existing wearable solutions can be used to fill in this gap, however, some of these devices are expensive, or are restricted by a distance to a receiver, making them not truly portable. As a result, a device was produced that was capable of collecting and storing IMU sensor data in a lightweight, portable, and affordable system. The device, called DCU, was validated against a gold standard system, the VICON MX, which yielded sufficient agreement to provide enough confidence to move forward with a larger collection trial.

To move the collection in the direction of machine-learning based activity classification, four classes of activity have been chosen, which are representative of ambulatory activities of daily living[41], with four differing levels of restriction to be performed for each activity. A brace was instrumented for this purpose, in addition to the DCU, which will be used to capture data in this larger trial, the details of which are mentioned in the following chapter. Using the device to capture acceleration, gyroscopic and magnetic field data, the data will be analysed to see if a statistical model can classify between the activities being performed, and whether differentiating factors can be established between differently restricted activity performances.

3 PARTICIPANT STUDY

3.1 INTRODUCTION

Having created a functional, validated data collection system and an instrumented restriction brace, it can now be used with confidence for larger-scale data collection. As mentioned in EMBEDDED SYSTEM, the intent of the DCU long term is the detection of gait-alterations which various conditions, such as knee osteoarthritis may cause. Before collecting on any unhealthy or diseased gaits, it was determined that collection of gait data from healthy individuals, whose gaits would be altered in an external and controllable way, would be a functional proof-of-concept. To control participant gait, an instrumented brace is used, featuring an adjustable locking mechanism to restrict flexion or extension of the wearer; in our use, only flexion restriction is of interest.

Some significant questions present themselves at the outset of this larger-scale collection: which activities are to be performed? How are participants to perform the activities? At which restriction levels should participants perform these activities? Further, the treadmill used in the collection also needs to be standardised, such that its speed and incline settings can be converted to real-world measures of speed and angle. Matters of participant safety must also be addressed, both in a physical and digital context.

3.2 TREADMILL STANDARDISATION

3.2.1 SPEED STANDARDISATION

Before a formal data collection could occur, the treadmill that was going to be used (Bodyguard T340, Quebec, Canada), which was different from the treadmill used in EMBEDDED SYSTEM for validation work, needed to be tested and standardised, so that its speeds could be compared against others. This was necessary because the treadmill did not provide any units with its speed values, so it was not certain if these were unit-less numbers or actual speeds with meaning. Similarly, for the incline values, the treadmill provided no units. This standardisation process did not involve the DCU in any capacity; a digital laser and contact tachometer (Powerfist, Manitoba, Canada) was used to measure speeds. The tachometer had a contact wheel, which allowed it to contact the surface being measured and capture its speed [179]. The treadmill was measured as follows:

1. Start the treadmill at some speed
2. Place the body of the tachometer against the rail of the treadmill, with the contact wheel touching the tread.
3. Allow the tachometer wheel to remain on the tread for 5 seconds, to come to a stable measurement
4. Collect 2 more measurements with the tachometer resting at the same position, to aid in accuracy
5. Repeat the process for different treadmill speeds

The tachometer was configured to produce results in meters/minute, which were then converted to more common units such as km/h or mi/h to see if the number used on the treadmill was arbitrary, or based on something physical. After conversion, it was found that the treadmill values were mi/h. The raw values measured and the converted values are provided in Table 3-1.

Table 3-1: Treadmill speed standardisation. Using a tachometer, the treadmill was set to a speed, in the range of speeds that were reasonable for the activities to be performed. Then the tachometer was used to measure these speeds three times, for consistency. The results were converted to miles per hour, and averaged, and it was found that the treadmill number approximates a speed in miles per hour. The speeds are provided in both metric and imperial units, the latter for the sake of showing the clear relationship between the treadmill's displayed values, and the real-world speeds they produce.

Treadmill Displayed Value	**Tachometer Measured (m/min)**			**Avg. Calculated Value (km/h)**	**Avg. Calculated Value (mi/h)**
	1	**2**	**3**		
3.7	97.1	97.1	97	5.824 ± 0.0035	3.619 ± 0.0022
2	52.5	52.9	52.9	3.166 ± 0.0139	1.967 ± 0.0086
0.5	13.3	13.3	13.1	0.794 ± 0.0069	0.493 ± 0.0043
4	104.9	105.4	105.1	6.308 ± 0.0151	3.920 ± 0.0094

3.2.2 INCLINE STANDARDISATION

Inclines were also anticipated to be used during collection, and thus, the treadmill's incline values needed to be tied to a real-world angle of incline. To do this, a digital protractor (Shenzhen Red Dragon Instruments Co., Ltd., Shenzhen, China) was used, as follows:

1. Ensure the protractor is in "Absolute" measurement mode
2. Ensure the treadmill is at 0 incline
3. Place the protractor on the rails of the treadmill, where it reads as close to 0° as possible.
4. Allow the protractor a few seconds to stabilise, then record the result
5. Incline the treadmill by one level. Leave the protractor in place.
6. Repeat 4 and 5 until the last incline level is reached
7. Decline the treadmill by one level
8. Allow the protractor a few seconds to stabilise, then record the result
9. Repeat 7 and 8 until the incline level of 0 is reached.

No consistent conversion between the angles in degrees recorded by the digital protractor and the treadmill incline levels was found; thus, it was concluded that the treadmill incline levels are likely arbitrary, though the incline levels offer values close to the tangent of the angle in degrees, as a percentage. For instance, for the treadmill inclination of 8, $100*\tan(4.8°) \approx 8.4$. Our measurement process produced a map that converted these likely arbitrary inclines into angles in degrees. The mapping is provided in Table 3-2.

Table 3-2: A table of the values measured from the treadmill during the incline standardisation process, described in 3.2.2. All the incline levels of the treadmill were measured, with each incline level receiving two measurements, aside from the measurement at the maximum incline. One measurement was made on the sweep to the maximum inclination, and one measurement on the way from the maximum to zero.

Treadmill Inclination	**Protractor Measurements (degrees)**		**Treadmill Inclination**	**Protractor Measurements (degrees)**	
	Measure 1	**Measure 2**		**Measure 1**	**Measure 2**
0	0.4	0.5	8	4.8	4.8
0.5	1.2	1.2	8.5	5.1	5
1	1.5	1.5	9	5.3	5.2
1.5	1.8	1.8	9.5	5.5	5.5
2	2.1	2	10	5.7	5.7
2.5	2.3	2.2	10.5	6	5.9
3	2.6	2.5	11	6.2	6.2
3.5	2.8	2.8	11.5	6.4	6.5
4	3.1	3	12	6.6	6.7
4.5	3.3	3.3	12.5	6.9	6.9
5	3.5	3.5	13	7.2	7.1
5.5	3.7	3.7	13.5	7.4	7.3
6	3.9	3.9	14	7.7	7.6
6.5	4.2	4	14.5	8	7.9
7	4.4	4.3	15	8.4	-
7.5	4.6	4.5			

3.3 PARTICIPANT SAFETY AND CONFIDENTIALITY

Participants who met the selection criteria were healthy and capable of performing everyday activities and exercises. To prevent participants from falling during activities at the most severe restriction level, a harness vest was used, connected to a tether anchored to a ceiling rail. The harness vest was tightened onto the participant and fastened to the tether to be secure, but kept loose enough to avoid applying additional restriction to the participants, and thus, preventing any unintentional difficulty being added to the activity.

Participant movement speeds on the treadmill were self-selected (see Table 3-7), reducing exhaustion and discomfort as risks. Furthermore, participants were allowed to rest between activities if they required a break. If rest breaks of significant duration occurred, they were noted, otherwise, most breaks, if taken, were short and unrecorded. Differences between starting times of different activities are encoded by the timestamps of the file names of the data.

The gait data collected from the participants are either kept physically with the principal researcher, or placed on a system with password-protected access. The identities of the participants are stored in a separate, password protected file, which maps participant IDs to names. Access to this file is very limited and is not needed for the data analysis to be performed. The anthropometry data collected from the participants (see Table 3-5) are stored in a file mapping the data to the participant ID, i.e. coded by participant ID, and is available to access by anyone who also has access to the collected gait data. Note, use of these data does not require knowledge of the participant's identities.

3.4 ACTIVITY AND RESTRICTION SELECTION

3.4.1 ACTIVITY SELECTION

After validating the system on one participant in EMBEDDED SYSTEM, with further validation work conducted by Nguyen et al. [173], a larger scale collection was undertaken, in order to collect enough data on which to perform machine learning. As opposed to the collection done in EMBEDDED SYSTEM, the activities performed in this collection were much closer to the intended use case of the DCU – common activities required for daily living. Activities of daily living (ADL) are correlated with quality of life, and reflect an individual's autonomy and ability to live independently [40,41]. The basic ADLs are those which relate to fulfilling physical needs, amongst which, is ambulation and movement; the completion of most of the other categories of basic ADLs also necessitates ambulation. Thus, it was determined that walking and sit-to-stand activities would be relevant movements in the context of healthy and unhealthy gait.

The walking activity had three types: normal walking, fast walking, and inclined walking. These were chosen to represent a realistic set of variations that may occur in daily gait. Normal walking was performed on a treadmill with incline 0, at a self-selected speed. Fast walking was performed on a treadmill with incline 0, with a fixed addition of 0.805 km/h (0.5 mph) to the self-selected walking speed. Inclined walking was performed at the same self-selected speed as normal walking, but with the treadmill incline set to 5 (3.5°). Sit-to-stand activities were performed by starting standing, and sitting on the edge of a chair, without fully relaxing into the chair, and then rising up again. Images representing the activities to be performed at an unrestricted restriction level (discussed in 3.4.2) are shown in Figure 3-1, and Figure 3-2. These images and data are based on series of demo collections conducted on the researcher for the sake of illustration alone; these data were not analysed and did not factor into any conclusions made. To see a description of all the relevant features on the instrumented investigator in these images, see EMBEDDED SYSTEM or Figure 3-5. The demo collections were performed with the arms of the principal investigator out of the way of the brace and the DCU, which is done to prevent them from being obscured in the images, and did not occur during actual collection.

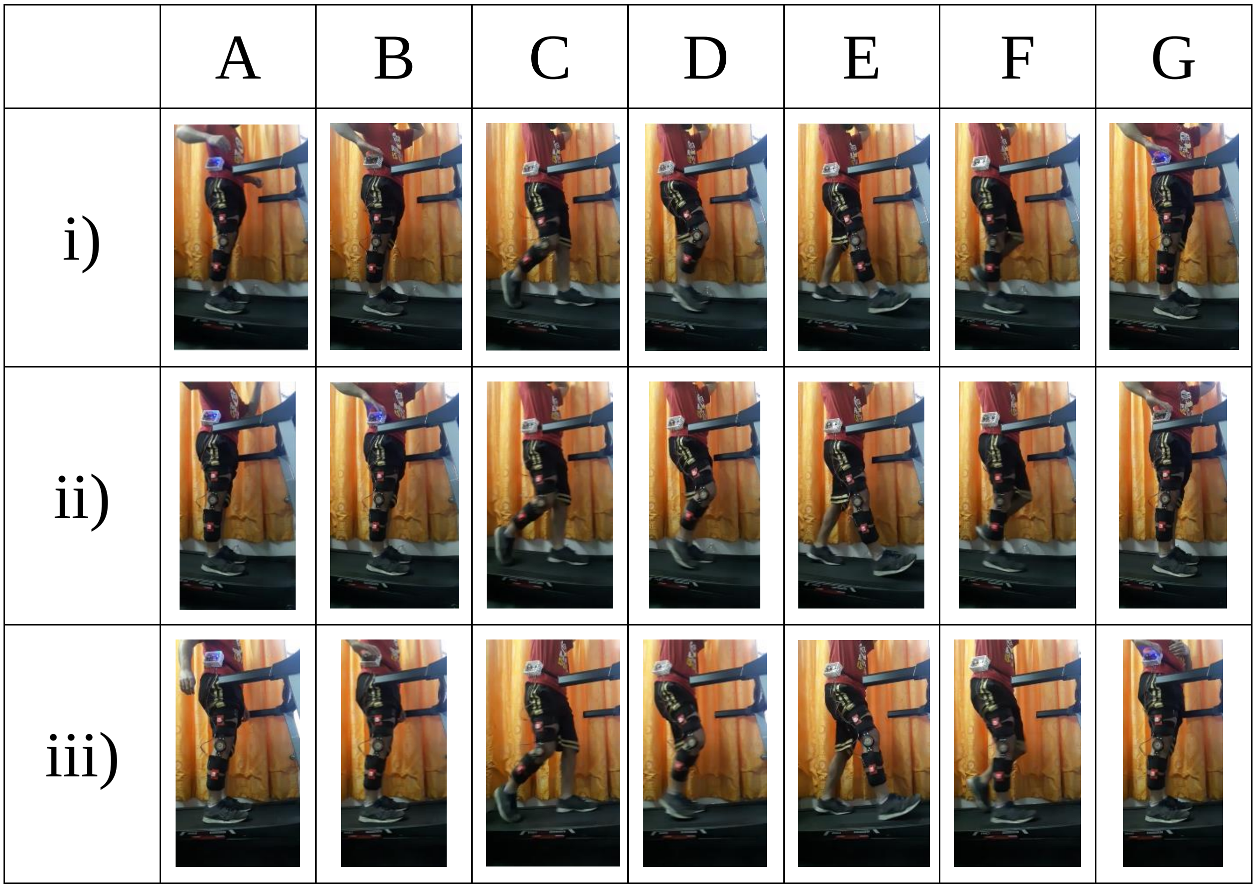


Figure 3-1: Images taken from demo collections of normal walk data (i), fast walk data (ii) and incline walk data (iii) on the researcher. The normal walk data were collected at a speed of 2mph, on a flat incline (~0.60° decline using digital protractor). The fast walk data were collected at a speed of 3mph, on a flat incline. The incline walk data were collected at a speed of 2mph on maximum incline (~4.80° incline using digital protractor) for exaggerated visual effect. All collections occurred at an unrestricted restriction level. The positions of the instrumented leg are described as follows: A shows the participant ready for collection. B shows the start button being pressed; the device is now collecting data. C shows the toe-off event of the gait cycle. D shows part of the swing phase of the gait cycle. E shows the heel strike event of the gait cycle. F shows part of the stance phase of the gait cycle. C to F are repeated while the collection continues. Once the collection is complete, and the treadmill is stopped, the collection is stopped, shown in G. Note, these data were not analysed, nor were they collected in the laboratory space.

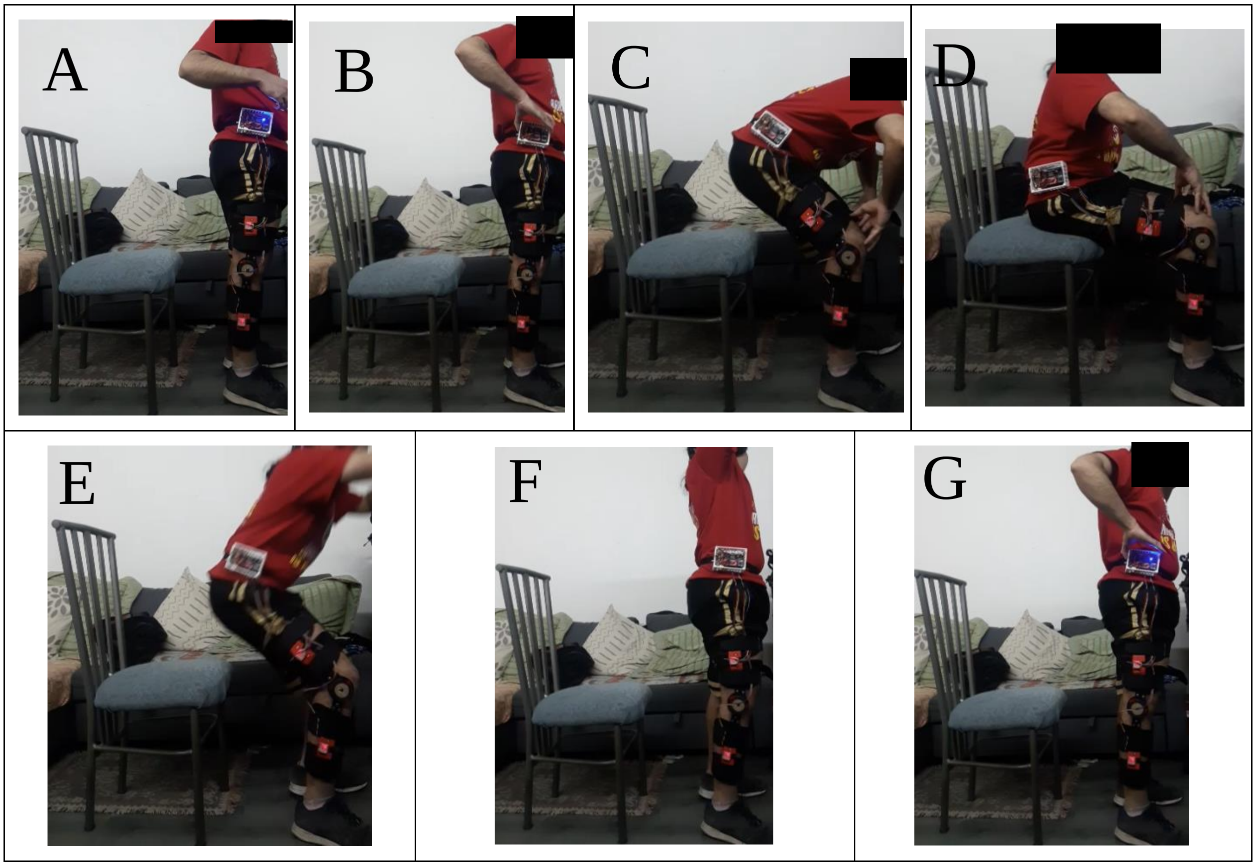


Figure 3-2: Images taken from a demo collection of sit-to-stand data at an unrestricted level, on the researcher. Like the collection on participants, no pace was set for the sit-to-stand activity here. During real collection, the chair would be propped against the treadmill body to prevent movement. A shows the participant ready for collection. B shows the start button being pressed; the device is now collecting data. C shows the participant negotiating their descent into the chair. D shows the participant seated in the chair; note they are not reclined fully into the seat, and they sit at the edge of the seat. E shows the participant negotiating their rise out of the chair. F shows the participant fully standing. C to F are repeated while the collection continues. Once the collection is complete, and the participant is standing, the collection is stopped, shown in G. Note, these demo data were not analysed, nor were they collected in the laboratory space.

To bring the collection closer to the disease-detection use case, participants' flexion was restricted on the instrumented leg. The restrictions were of varying severities, with the goal being to simulate, approximately, the gait of someone with knee osteoarthritis or another gait-altering condition. This collection would produce data that would allow comparison of the same participant’s unrestricted performance of an activity against their differently restricted performances of the same activity, to see if a distinction between restrictions can be found. This would serve as a fundamental step towards building a system that could detect disease automatically, by a change in gait.

Table 3-3: Breakdown of activities performed during formal data collection trials. Each activity was performed at unrestricted (U), light (L), moderate (M) and severe (S) restriction levels. These levels are denoted in the table below as U, L, M, and S, respectively, and detailed in 3.4.2 and Table 3-4.

Activity	Duration	Number of Trials	Restrictions	Notes
Walk	3 minutes	3 Trials	U, L, M, S	Self-selected speed; Flat treadmill
Fast Walk	3 minutes	3 Trials	U, L, M, S	Fixed offset from normal speed; Flat treadmill
Incline Walk	3 minutes	3 Trials	U, L, M, S	Self-selected speed; Treadmill inclined to 5 (3.5°)
Sit-To-Stand	15 repetitions	3 Trials	U, L, M, S	Self-selected pace; Sit and stand from edge of seat

3.4.2 FLEXION RESTRICTION SELECTION

To determine the flexion restriction for each activity, the following protocol was performed:

1. Wear the restricting brace on the right leg, with no flexion restriction applied at the knee
2. Perform the activity, to get a sense of the baseline of what the activity feels like, without restriction, but with the effect of the brace itself
3. Increase the flexion restriction, starting from the least severe level (120° on the particular brace that was used), and perform the activity again
4. Repeat the process, increasing the restriction every time, until a restriction is found at which the lock is *just* felt during the activity, without affecting the activity (e.g. for walking, this would be the maximum flexion angle at the end of the stance phase). This angle is considered to be “Light” Severity
5. The next step up from this angle (10° increments for the brace we are using) is considered “Moderate” Severity
6. The next step up from this angle is considered “Severe” Severity

This protocol was applied to the researcher, prior to collection on any of the participants, in order to determine the levels of restriction that would be used in the collection. These restrictions are shown in Table 3-4. These restriction levels were applied to all participants during collection. This should noticeably restrict the toe-off and swing phases of the gait cycle, with the possibility of affecting other points, based on the joint angles of the gait cycle [180–182]. Note that although unrestricted is assigned an angular value, in practice, the leg does not interact with the restriction, as an angle of 100° flexion was not observed during testing, nor does it appear during the gait cycle for normal walking [180–182].

Table 3-4: Angle of restriction applied to the right leg of every participant during collection. Note that the "Walk" restrictions were applied to all the walking activities, as the speed of the motion did not change the maximum flexion angle reached during the activity. The angles given are in degrees, and represent the maximum flexion angle the knee is allowed to have, with 0° being a straight leg.

Activity Type	Unrestricted	Light	Moderate	Severe
Walk	100°	40°	30°	20°
Sit-To-Stand	100°	70°	60°	50°

3.5 DATA COLLECTION

3.5.1 PREPARATION

Participants were contacted a priori by email to gauge their interest in the study and whether they would be able to participate. They were told to bring a change of clothes, especially a pair of shorts, and shoes that they deemed comfortable for everyday movements. The participants were made aware of the nature of the collection, the data to be collected, and the risks present and mitigation strategies used to reduce them. Participants were allowed to ask for any clarifications; willing participants signed the relevant paperwork on the day of collection and changed into exercise clothing.

Participants were asked for their age, sex, height and weight, and those details were recorded. Those who did not know their own approximate height were measured, and those who didn't know their approximate weight were asked to report it at their earliest convenience after the collection. Participant's right leg length was measured in two parts: the straight-line distance from the greater trochanter to the lateral femoral condyle, and the straight-line distance from the lateral femoral condyle to the lateral malleolus; these landmarks were palpated manually. The greater trochanter was identified while the participant was standing, and the lateral femoral condyle was identified while sitting. Length measurements were taken while standing. See Figure 3-3 for details. Collected anthropometric data for all participants are provided in Table 3-5. Group-averaged anthropometric data are provided in Table 3-6.

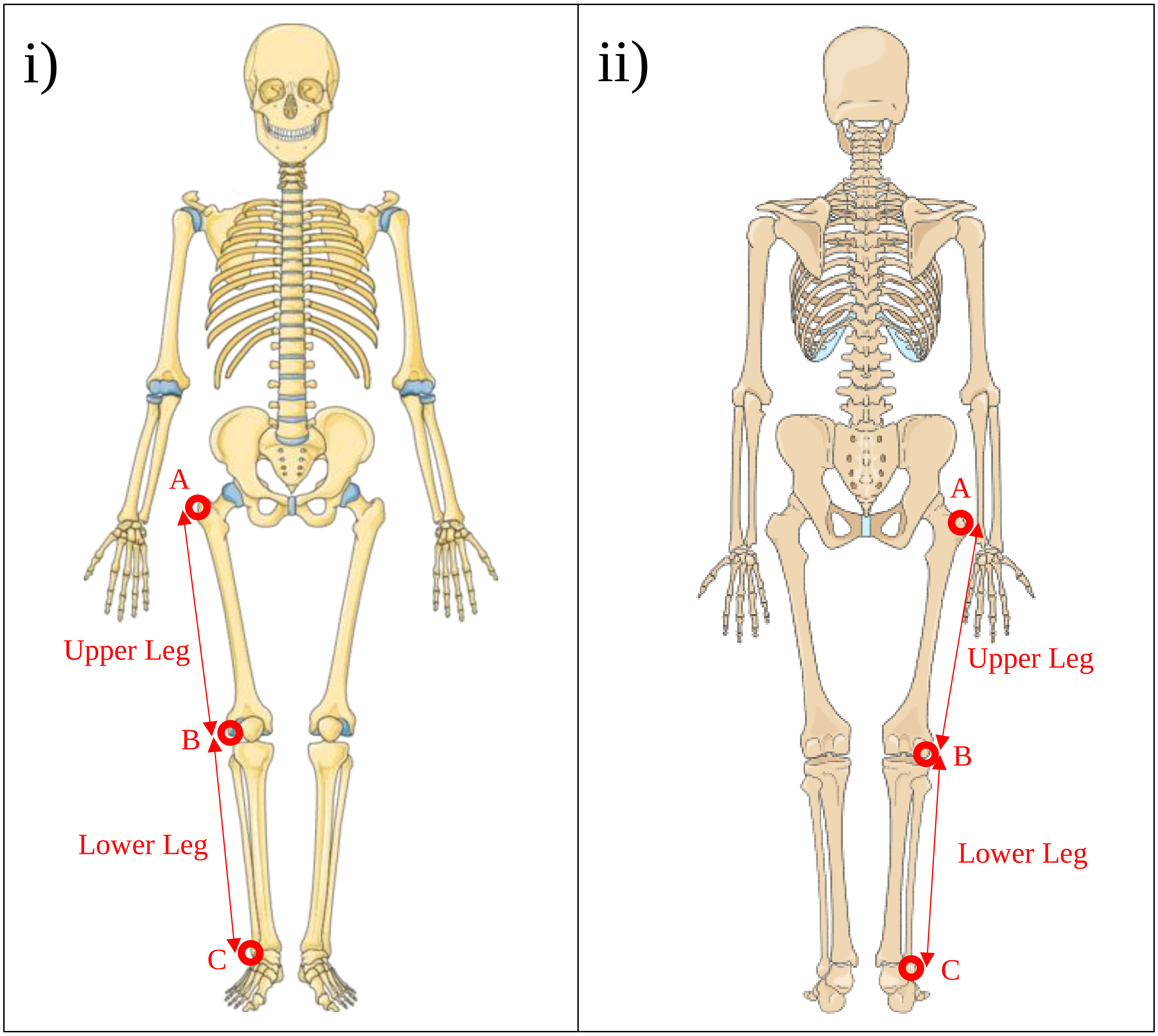


Figure 3-3: Locations of skeletal markers used to measure participant upper and lower right leg lengths, on an anterior view (i) and posterior view (ii) skeleton. Using diagrams from [183–185], the correct markers were identified and placed on the images above. A is the greater trochanter, B is the lateral femoral condyle, C is the lateral malleolus. The straight-line distance from A to B defines the upper leg, and the straight-line distance from B to C defines the lower leg. The images of the posterior and anterior skeleton[186,187] are provided by Servier Medical Art (https://smart.servier.com), licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

Table 3-5: Anthropometric data for all the participants in the collection. Collected data include the height in metres, weight in kilograms, straight-line distance from greater trochanter to lateral femoral condyle (upper leg) in metres, and straight-line distance from lateral femoral condyle to lateral malleolus (lower leg) in metres. Note the codes P02 to P04 were used for early exploratory work with unrelated data, and so those codes do not appear here.

PID	Age	Sex	Height (m)	Weight (kg)	Upper Leg (m)	Lower Leg (m)
P01	24	M	1.75	81.6	0.445	0.433
P05	25	M	1.69	72	0.375	0.42
P06	25	M	1.71	53.8	0.426	0.375
P07	20	M	1.37	49.9	0.298	0.335
P08	26	F	1.55	49.9	0.433	0.358
P09	28	F	1.85	104	0.515	0.433
P10	32	F	1.64	70	0.446	0.402
P11	32	F	1.63	58	0.465	0.383
P12	25	M	1.78	59	0.447	0.433
P13	27	F	1.71	65	0.535	0.426
P14	26	M	1.86	81	0.448	0.468
P15	26	F	1.5	48	0.42	0.35
P16	27	M	1.84	90.7	0.475	0.471
P17	25	M	1.65	76	0.415	0.41
P18	25	M	1.75	80.74	0.416	0.422
P19	27	F	1.58	60	0.395	0.38

Table 3-6: Group-averaged anthropometric data with standard deviation. Collected data include the height in metres, weight in kilograms, straight-line distance from greater trochanter to lateral femoral condyle (upper leg) in metres, and straight-line distance from lateral femoral condyle to lateral malleolus (lower leg) in metres.

Sex	Age	Height (m)	Weight (kg)	Lower Leg (m)	Upper Leg (m)
Both	26.25 ± 2.86	1.68 ± 0.13	68.74 ± 16.20	0.41 ± 0.04	0.43 ± 0.05
F	28.29 ± 2.63	1.64 ± 0.12	65.03 ± 18.98	0.39 ± 0.03	0.46 ± 0.05
M	24.67 ± 1.94	1.71 ± 0.14	71.63 ± 14.15	0.42 ± 0.04	0.42 ± 0.05

The protocol for instrumenting the participant was followed as described in these steps:

1. The participant was asked to sit on the ground, with their right leg stretched out in front of them.
2. The brace was opened flat and placed under and around their leg, such that all the gripping pads made contact with skin. If skin contact was not being made at the thigh, the participant was asked to roll up the hem of their shorts so contact could be made (see Figure 3-4 for visual).
3. The brace was tightened using the Velcro straps (see Figure 3-4 for visual). The brace needed to be comfortable but tight, and was adjusted with the participant's input until it was so. They were suggested to walk around the area to get a feel for movement with the brace on.
4. The participant was placed into the harness, with care taken not to disturb the brace too much. The harness was tightened to the participant to be secure, but not attached to the ceiling rail yet.
5. The DCU was tightened onto the participant by the Velcro belt. The DCU was placed at the lower back of the participant so it would be out of the way for the walking movements. Wires were attached to the posterior side of the sensors, so they would not hinder the collection. The wires were loosely tethered to the Velcro straps of the brace to keep them from freely swinging about during collection.

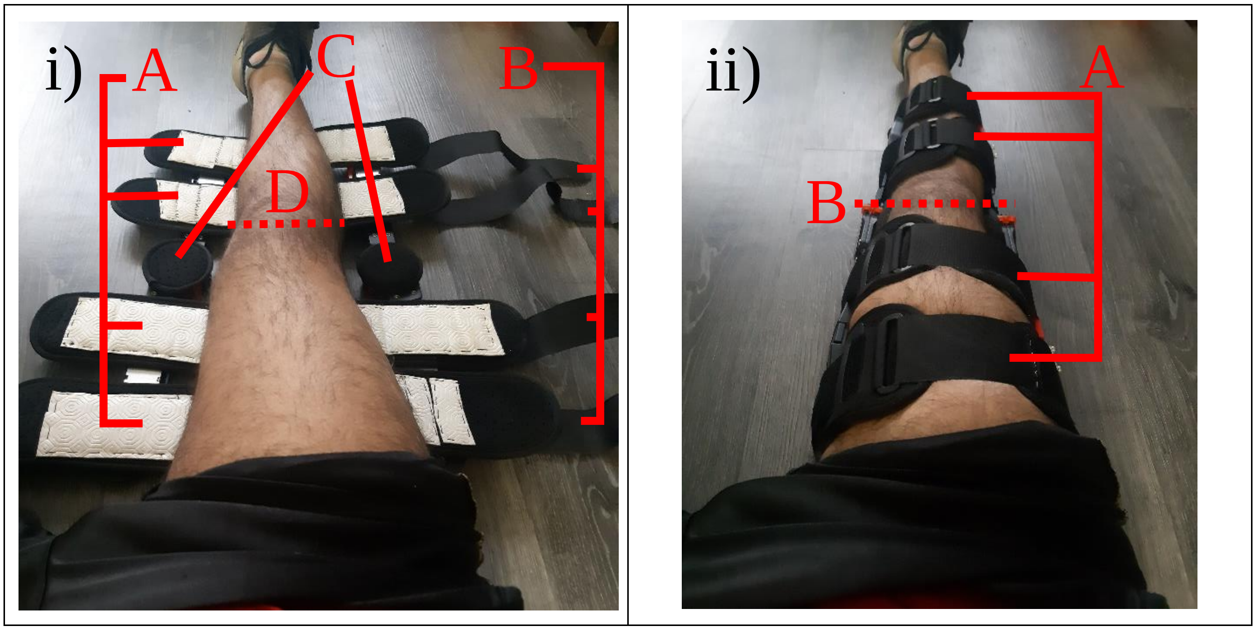


Figure 3-4: Images showing the process securing the brace to the leg of the participant. i) shows the opened brace prior to tightening it, being aligned with the leg. Enough skin needs to be exposed so the brace's gripping pads (A) can make contact and help secure the brace via the straps (B) to the participant as they move. Although no formal alignment procedure is undertaken, best efforts are made to align the knee pads (C) of the brace with the center of the knee of the participant (D). ii) shows the brace with the straps closed (A) around the leg of the participant. Once again, best efforts are made to keep the knee pads aligned to the lateral and medial sides of the knee (along B) so that the brace (and the sensors mounted to it) can be as parallel as possible to the sagittal plane.

Prior to starting collection, the participant's walking speeds had to be determined. The procedure followed was similar to a preferred walking speed test [188,189]. The participant was made to walk on the treadmill and identify a speed that they felt was most similar to an "everyday walking speed", or a "walking around the house" speed for them, while the speed of the treadmill was obscured from them. The treadmill was started at ~0.8 km/h (0.5 mi/h) and the participant was asked if the speed should be lowered or raised and was given a moment to adjust to the new speed. This was continued until the participant found a speed that they found comfortable as an everyday walking pace. This number was recorded, and then the treadmill was restarted, but at a speed of ~4.8-5.6 km/h (3-3.5 mi/h). Once again, the participant was asked if the speed should be lowered or raised until they identified a speed they felt matched everyday walking for them. This value was recorded, and averaged with the first value to give a normal walking speed. This value had an additional ~0.8 km/h (0.5 mi/h) added to it for fast walking, and both speeds were recorded by the investigator. Note, it wasn't considered important that the fast walking speed be considered "fast" by the participant, rather, the goal was to make the fast speed faster relative to the normal speed. The participants' normal and fast walking speeds are provided in Table 3-7 with group-averaged values provided in Table 3-8. An image of the overall laboratory setup can be seen in Figure 3-5.

Table 3-7: Walking speeds selected by each participant. Each participant's walking speed represents what they considered as "every day" or "around the home" walking speed. It was found by starting the treadmill at a slow speed, and raising it until the participant felt comfortable, and then starting the treadmill from a high speed, and lowering it until the participant felt comfortable, and averaging these values. The fast walk speed is a fixed ~0.8 km/h (0.5 mi/h) faster than the chosen walking speed. Speeds provided in mi/h for easy translation to treadmill settings. Note the codes P02 to P04 were used for early exploratory work with unrelated data, and so those codes do not appear here.

PID	Sex	Age	Walk Speed (km/h)	Walk Speed (mi/h)	Fast Walk Speed (km/h)	Fast Walk Speed (mi/h)
P01	M	24	3.86	2.4	4.67	2.9
P05	M	25	4.02	2.5	4.83	3
P06	M	25	2.90	1.8	3.70	2.3
P07	M	20	3.22	2	4.02	2.5
P08	F	26	2.74	1.7	3.54	2.2
P09	F	28	3.70	2.3	4.51	2.8
P10	F	32	2.90	1.8	3.70	2.3
P11	F	32	3.22	2	4.02	2.5
P12	M	25	2.57	1.6	3.38	2.1
P13	F	27	2.74	1.7	3.54	2.2
P14	M	26	4.18	2.6	4.99	3.1
P15	F	26	2.09	1.3	2.90	1.8
P16	M	27	3.54	2.2	4.35	2.7
P17	M	25	2.74	1.7	3.54	2.2
P18	M	25	4.35	2.7	5.15	3.2
P19	F	27	2.90	1.8	3.70	2.3

Table 3-8: Group-averaged walking speeds with standard deviation. The walking speed represents the average of values that each participant felt was an "every day" or "around the house" pace, with the fast walking speeds being the average of those values with a fixed offset of ~0.8 km/h (0.5 mi/h). The speeds are provided in mi/h for easy translation to treadmill settings.

Sex	**Walk Speed (km/h)**	**Walk Speed (mi/h)**	**Fast Walk Speed (km/h)**	**Fast Walk Speed (mi/h)**
Both	3.23 ± 0.65	2.01 ± 0.40	4.03 ± 0.65	2.51 ± 0.40
F	2.90 ± 0.49	1.80 ± 0.31	3.70 ± 0.49	2.30 ± 0.31
M	3.49 ± 0.66	2.17 ± 0.41	4.29 ± 0.66	2.67 ± 0.41

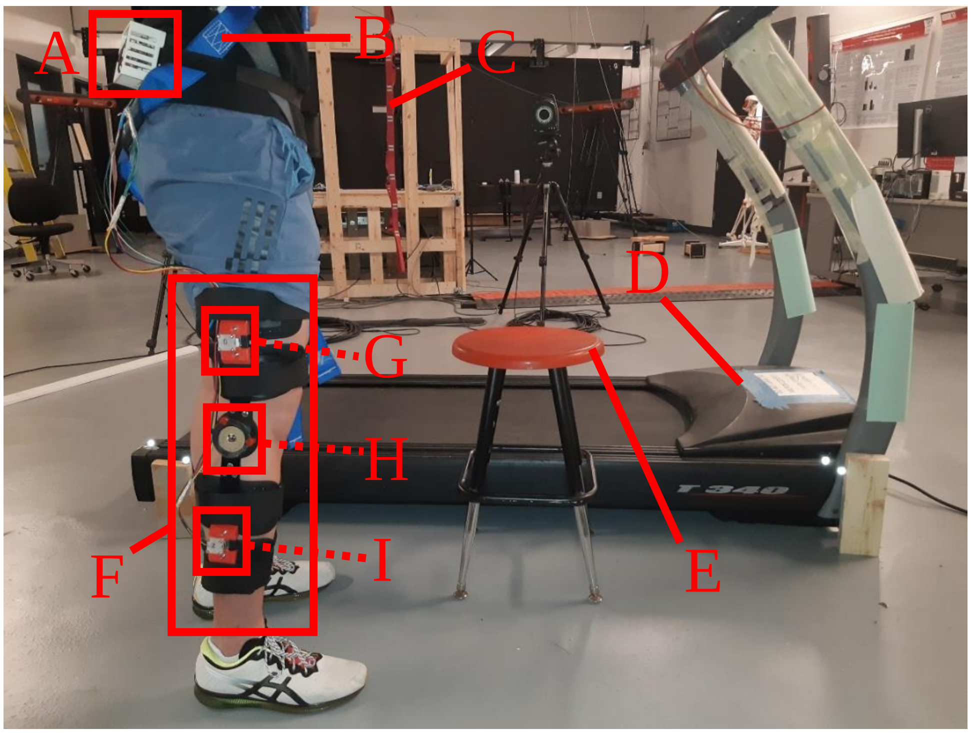


Figure 3-5: Instrumented participant in the laboratory setting, ready to collect. A is the DCU, which collects data from sensors G and I via wires and saves the captured data to an onboard microSD card. B is the harness, which will secure the participant to the tether (C) attached to a rail along the ceiling. D is the T340 BodyGuard Treadmill, which is used for the walking activities. E is the stool, which is used for the sitting activities. Note the stool was sometimes replaced with a chair, depending on what was available in the laboratory. F is the restrictive brace, with the flexion-restricting mechanism (H) which has 3D-printed locks inserted into it. The brace has two IMU sensors mounted on it, one for the upper leg (G), and one for the lower leg (I).

3.5.2 PARTICIPANT DATA COLLECTION METHODS

3.5.2.1 ACTIVITY AND RESTRICTION DETERMINATION

The ordering of the activities to be performed were semi-randomly selected. The value of restriction to be applied was randomly selected first, which informed the restriction level of the series of activities to be performed. Three consecutive trials of a randomly chosen activity were performed at this restriction level before randomly selecting the next restriction level and redoing the process at this restriction level. See Figure 3-6 for a visual explanation. Note that although both sitting and walking activities have the same categories of restriction (see Table 3-4), the *angle* at which the restriction is set is different, and needs to be considered when transitioning between activities during collection.

The process of completing three trials of a given activity at a given restriction could be made complicated by various issues that could arise during collection. These include issues with the brace, such as the restriction locks snapping, or the brace slipping significantly, or issues with the DCU, such as writing or recording errors. In these cases, the trial would be stopped, the issue would be remedied, and the trial would be restarted (see Figure 3-7 for a visual explanation). The fact that the trial had to be redone was tracked, so that at the end of the collection, files could be matched to the activities performed correctly. Due to these possible failure modes, a single collection would sometimes exceed the estimated time, necessitating that some collections be spread over two days. These splits would occur at clean breaks between completed sets of activities at a particular restriction; thus, there was never an instance of two trials of the same activity at the same restriction occurring on different days.

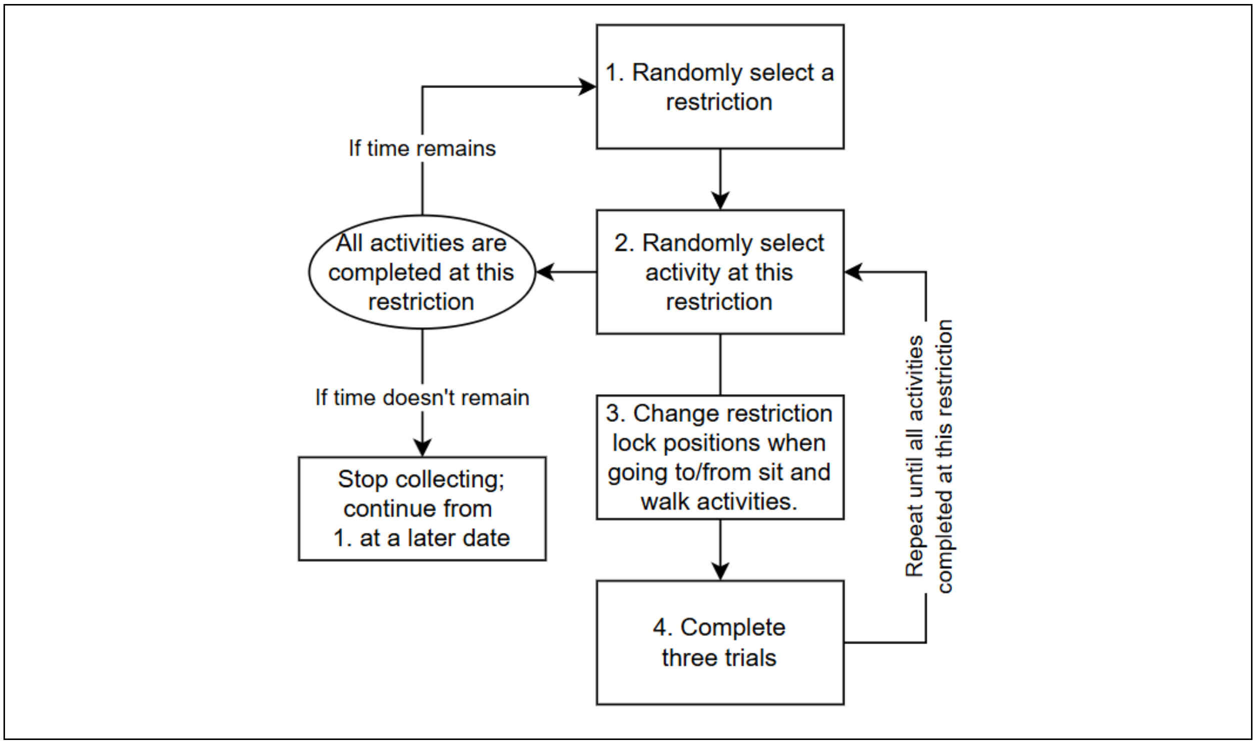


Figure 3-6: A visual explanation of activity and restriction selection for each participant. First the restriction is randomly chosen, then three consecutive trials of a randomly chosen activity are completed at that restriction. Activities continue to be randomly chosen at this restriction until all of them have been completed, at which point, a new restriction level is randomly selected. For particularly difficult collections, if time does not remain, it is at this point that the collection can be cleanly broken into a second day. This entire process is repeated until all the data have been collected. Note, the location of the restriction locks for the same restriction level of walk and sit activities differ, and need to be changed when transitioning between activities.

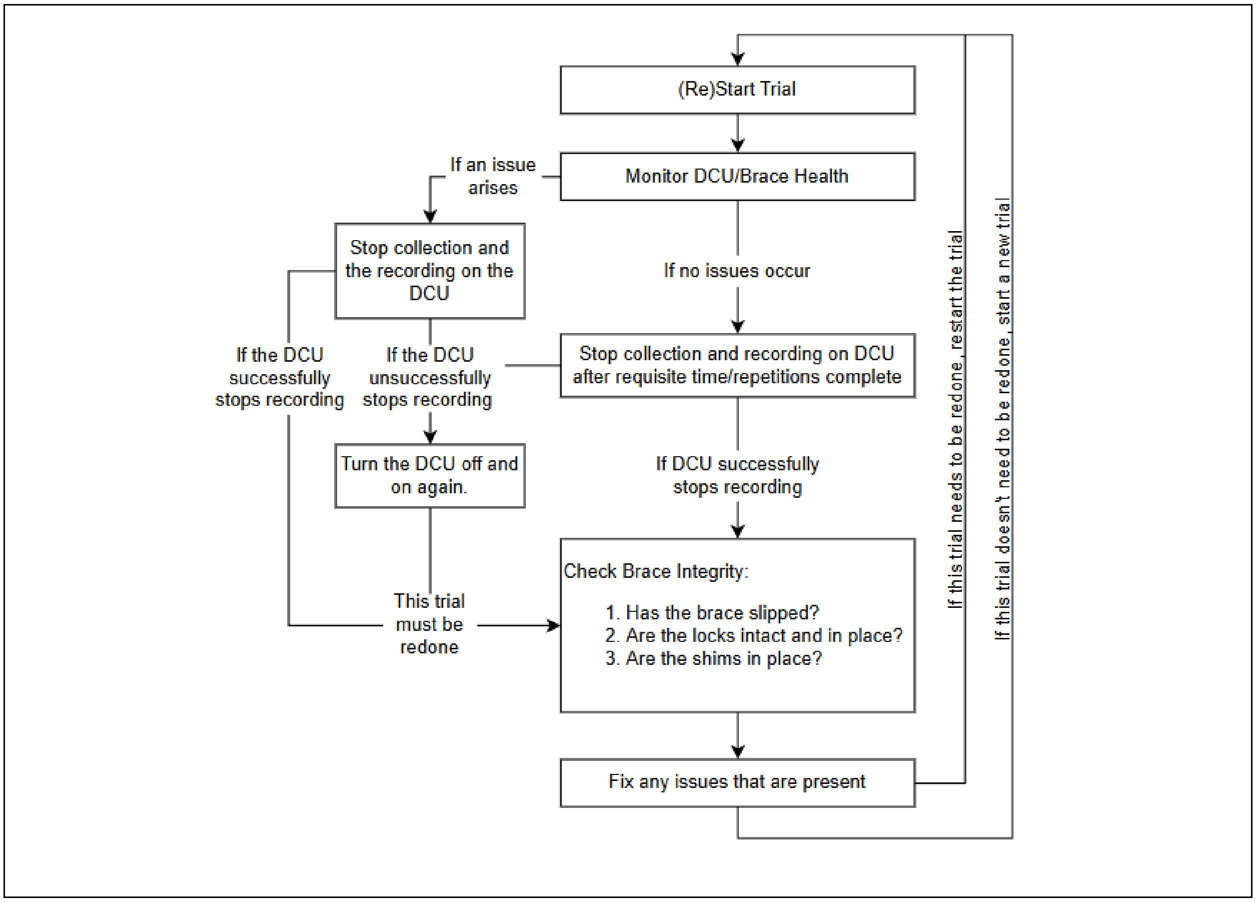


Figure 3-7: Visual explanation of the process of completing a trial for any given activity and restriction level. As a trial begins, proceeds and ends, the health of the brace and the DCU must be monitored, and if issues arise, the ongoing trial must be stopped. In some cases, the DCU may have failed to record data or to stop recording successfully, and in other cases, the brace may have slipped, or the restriction locks may have broken. In any case, the ongoing trial needs to be redone (while keep track of these extra files for later accounting).

3.5.2.2 WALKING ACTIVITIES

For any walking activity, the harness was connected to the ceiling rail, such that the tension on the tether would not hinder the activity, while being short enough to catch the participant in the case of a fall. The participant was asked to step onto the treadmill, the incline was adjusted if an Incline Walk activity was being performed, and the DCU and brace were examined. The integrity of the physical connections of the sensors and wires were double checked, the plastic shim at the restricting mechanism was repositioned if necessary, and the plastic restriction locks were checked for damage and correct and firm placement (see EMBEDDED SYSTEM for brace details). Once these details were confirmed, the DCU was adjusted to be at the lower back, so it would be out of the way of any movement. Any excess wire lengths were held back by the straps of the brace. The DCU was then powered on, and activated. After approximately 5 seconds (three flashes of the onboard LED), the treadmill was set to the correct speed, and the participant would begin walking. No directions were given to the participant on how to walk, aside from reminding them to not lean on the handrails, to produce a more natural walking cycle. The researcher was seated behind the participant, making sure the DCU was functioning correctly, wires were not coming undone, and that the brace was not significantly slipping down the leg. After the activity was complete (see Table 3-3 for times/repetitions), the treadmill was stopped, and once the participant came to a standstill, the DCU was stopped. In some cases, the DCU failed to stop correctly, which meant that trial had to be redone. Further, if the restriction lock broke mid-trial, which was a rare occurrence during walking activities, the trial would be stopped, the lock would be replaced, and the trial would be redone. This process would be repeated until three trials of the selected walking activity were completed. See Figure 3-1 for illustrative images describing this set of activities.

3.5.2.3 SITTING ACTIVITIES

For sitting activities, the participant would be asked to sit in a chair or stool which was propped against the treadmill to prevent motion; the exact dimensions of the chair or stool were not taken. If the chair had a back, it was placed so the back would be perpendicular to the treadmill, so that participants did not have the option of reclining back into it. They were specifically asked to not recline fully into the seat, and to sit at the edge of the seat. No other instructions were given to participants on how to sit in the chair, especially in terms of how to negotiate different restrictions during the activity, and no pace of sitting and standing was prescribed. Some participants would descend into the chair in a hands-free way, while others would use the back of the chair to aid their descent. Prior to starting the collection, the DCU was moved to the lateral side of the participant and wires were tucked under the straps of the brace so that the DCU would not interfere with the sitting and standing of the participant. The participant was tethered to the ceiling rail in the seated position so that tension in the tether would not hinder the activity. Much like the walking activities, the integrity of the connections, plastic shim location, and restriction lock placement and damage were checked before each trial of sitting. Once the participant was ready, the DCU was turned on, and the participant would wait for approximately 5 seconds (three flashes of the onboard LED) before commencing the trial. The number of repetitions completed was counted by the researcher while monitoring the health of the DCU during the activity. Once the requisite number of repetitions were complete, the participant would stand still, and the DCU would be stopped. If the DCU failed to record correctly, the trial was redone. If the restriction lock broke during a trial, which occurred somewhat commonly during sitting activities, that trial was stopped, the lock replaced, and the trial was redone. This process was repeated until three trials of the selected sitting activity was completed. See Figure 3-2 for illustrative images describing this activity.

3.5.3 POST COLLECTION ORGANISATION

Once the entire collection is believed to be complete, the onboard microSD card is removed from the DCU and plugged into a computer. The data files collected are compared superficially, by checking file numbers and file sizes, against a spreadsheet keeping track of all the collection details. Once it can be confirmed that the order of activities, trials, and restrictions expected line up with the file numbers and file sizes collected, the collection is complete.

3.6 CONCLUSION

The participant study was an application of the DCU and restricted brace developed, instrumented, and validated in EMBEDDED SYSTEM. Sixteen healthy participants who were capable of performing everyday activities were collected upon, with walking, fast walking, incline walking and sit-to-stand activities being performed. Four restrictions were set for each activity, each restricting the participant more and more severely, and 3 trials at each restriction level were performed. At the end of the completed collection, 48 files were produced from each participant, with 16 participants producing 767 files overall (one file is missing due to a failure to collect a second trial of moderately restricted incline walking for P05). These data would need to be labelled, sorted, and cleaned, before being analysed. Two questions served as the motivation for this collection, and will guide the analysis process:

- Could a machine learning model be produced that could distinguish between the different classes of activities?
- Could a trend be found that could distinguish different restriction levels of the same activity from each other?

The next chapter applies traditional and machine learning techniques to analyse the data in search of answers to these questions.

4 DATA ANALYSIS

4.1 INTRODUCTION

Using our DCU, 16 participants were instrumented and conducted data collection activities. Each participant was asked to conduct four types of activities (walk, fast walk, incline walk, sit-to-stand), each at four differing levels of severity (unrestricted, light, moderate, severe). Each restriction of each activity was conducted three times (to produce three trials), except in one mistaken case, where only two trials occurred.

Across all the participants, 767 files were produced, which now needed to be correctly labelled and organized; these labels would be necessary when processing the data. Once that preliminary work is complete, the driving questions for the remainder of the work are as follows:

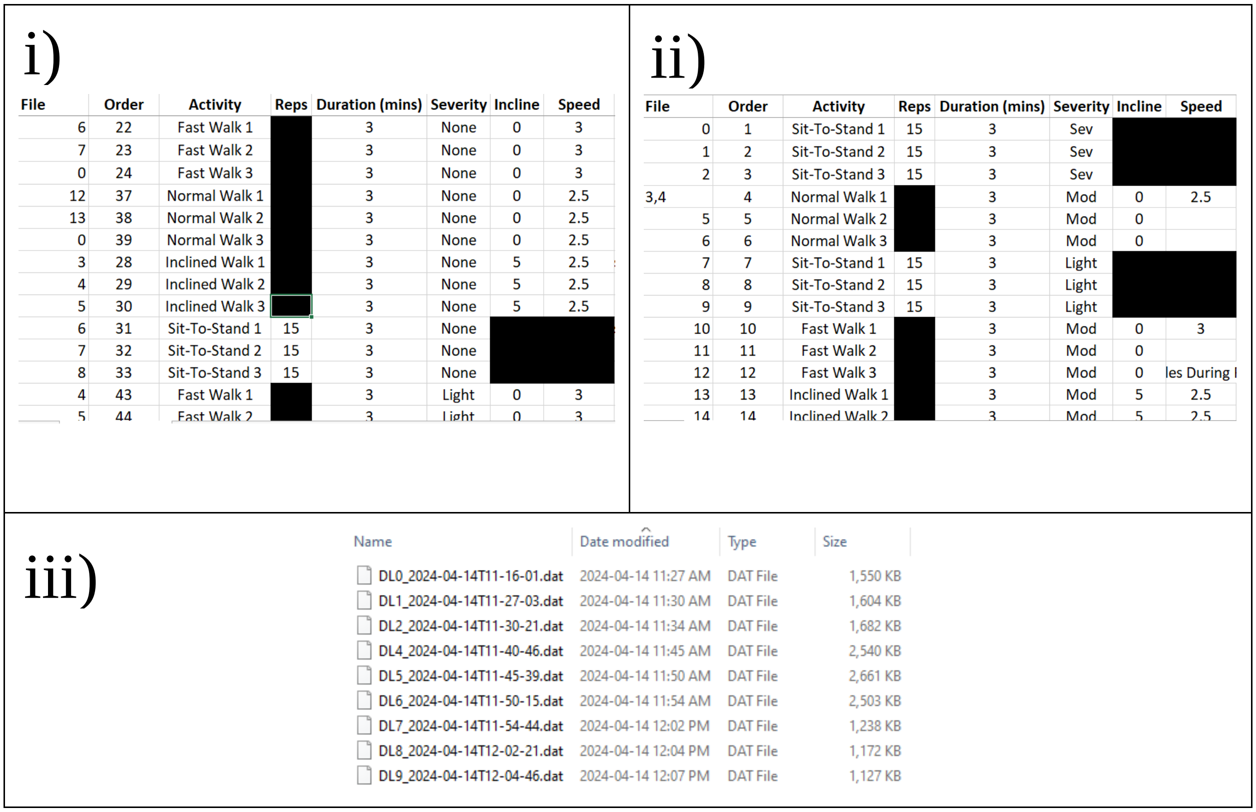
1. Can we predict which data came from which activity, irrespective of the restriction level of the activity?
2. Can we find feature(s) that differentiate differing levels of restriction for the same activity?

To attempt to find these trends in the data, traditional statistics and machine learning techniques were applied. In finding features that differentiate restriction levels, we applied ANOVA and post-hoc tests to see if there were statistically significant differences using the metrics we found relevant, as well as any necessary tests required to preclude the use of ANOVA. In the context of machine learning, we used a convolutional neural network to classify activities performed by independent participants.

4.2 DATA ORGANIZATION AND CLEANUP

4.2.1 IDENTIFYING AND LABELLING FILES

When collecting data from participants, a spreadsheet was simultaneously maintained to log details of the collection as they happened. Two such details were the data log number, and global trial number, which helped to identify files post collection. The DCU labels files starting at 0, and every consecutive file thereafter has its label increased by 1. If the system is ever shut off, the file labels are restarted at 0, though no filename conflicts occur due to additional timestamp data in the filename. These numbers are referred to as data log numbers, and during collection, the anticipated value of these numbers is kept track of, which are especially useful in cases of collection failure, where the label should be reset to 0, which is reflected by the researcher in the anticipated file label. In addition to this information, global trial number is also recorded, which is simply a number starting at 1 and incrementing to 48, to provide an ordering to the files collected. When organizing the data post-collection, these two numbers, in addition to the file sizes of the files, help to identify which files are which. See Figure 4-1 for an example of how the files are identified post collection.



i)

File	Order	Activity	Reps	Duration (mins)	Severity	Incline	Speed
6	22	Fast Walk 1	[redacted]	3	None	0	3
7	23	Fast Walk 2	[redacted]	3	None	0	3
0	24	Fast Walk 3	[redacted]	3	None	0	3
12	37	Normal Walk 1	[redacted]	3	None	0	2.5
13	38	Normal Walk 2	[redacted]	3	None	0	2.5
0	39	Normal Walk 3	[redacted]	3	None	0	2.5
3	28	Inclined Walk 1	[redacted]	3	None	5	2.5
4	29	Inclined Walk 2	[redacted]	3	None	5	2.5
5	30	Inclined Walk 3	[redacted]	3	None	5	2.5
6	31	Sit-To-Stand 1	15	3	None	[redacted]	[redacted]
7	32	Sit-To-Stand 2	15	3	None	[redacted]	[redacted]
8	33	Sit-To-Stand 3	15	3	None	[redacted]	[redacted]
4	43	Fast Walk 1	[redacted]	3	Light	0	3
5	44	Fast Walk 2	[redacted]	3	Light	0	3

ii)

File	Order	Activity	Reps	Duration (mins)	Severity	Incline	Speed
0	1	Sit-To-Stand 1	15	3	Sev	[redacted]	[redacted]
1	2	Sit-To-Stand 2	15	3	Sev	[redacted]	[redacted]
2	3	Sit-To-Stand 3	15	3	Sev	[redacted]	[redacted]
3,4	4	Normal Walk 1	[redacted]	3	Mod	0	2.5
5	5	Normal Walk 2	[redacted]	3	Mod	0	
6	6	Normal Walk 3	[redacted]	3	Mod	0	
7	7	Sit-To-Stand 1	15	3	Light	[redacted]	[redacted]
8	8	Sit-To-Stand 2	15	3	Light	[redacted]	[redacted]
9	9	Sit-To-Stand 3	15	3	Light	[redacted]	[redacted]
10	10	Fast Walk 1	[redacted]	3	Mod	0	3
11	11	Fast Walk 2	[redacted]	3	Mod	0	
12	12	Fast Walk 3	[redacted]	3	Mod	0	les During I
13	13	Inclined Walk 1	[redacted]	3	Mod	5	2.5
14	14	Inclined Walk 2	[redacted]	3	Mod	5	2.5

iii)

Name	Date modified	Type	Size
DL0_2024-04-14T11-16-01.dat	2024-04-14 11:27 AM	DAT File	1,550 KB
DL1_2024-04-14T11-27-03.dat	2024-04-14 11:30 AM	DAT File	1,604 KB
DL2_2024-04-14T11-30-21.dat	2024-04-14 11:34 AM	DAT File	1,682 KB
DL4_2024-04-14T11-40-46.dat	2024-04-14 11:45 AM	DAT File	2,540 KB
DL5_2024-04-14T11-45-39.dat	2024-04-14 11:50 AM	DAT File	2,661 KB
DL6_2024-04-14T11-50-15.dat	2024-04-14 11:54 AM	DAT File	2,503 KB
DL7_2024-04-14T11-54-44.dat	2024-04-14 12:02 PM	DAT File	1,238 KB
DL8_2024-04-14T12-02-21.dat	2024-04-14 12:04 PM	DAT File	1,172 KB
DL9_2024-04-14T12-04-46.dat	2024-04-14 12:07 PM	DAT File	1,127 KB

Figure 4-1: Visual example of identifying which files are which post collection. i) shows the spreadsheet that is maintained during collection. Note that the global trial number (Order) is tracked, which shows the order that each trial occurs over the whole collection, and the data log label (File), which is how the DCU will label each file. Some files have multiple labels, which means a file had been redone. ii) shows the sorted spreadsheet after the entire collection on Order, which shows us the order that trials occurred in over the collection. With this sorted spreadsheet in mind, we can look at all the files that we collected, and sort them by their modification date, which gives us a chronological ordering of their creation (iii)). Using the file sizes and information from ii), we can now correctly identify the files, and apply to them activity, restriction, trial number, and participant identifiers.

Once the files are correctly identified, the filenames have important metadata appended to them: the participant ID, the restriction angle, the activity performed. The data log number and timestamp information appended by the DCU are kept in the filename in the case that they become useful in the future. The definitions of the different parts of a filename are provided in Table 4-1.

Table 4-1: Breakdown of the different parts of the filename of a file produced during data collection. Data log number, year-month-day and hour-minute-second timestamps are attached by the DCU itself. All remaining information is added by the researcher post collection, once the correct files have been identified, as shown in Figure 4-1. Having these pieces of information will help during data analysis.

Filename: (ACT)(T#)DL(DL#)_F(R#)_(YYYY-MM-DD)T(HH-MM-SS)_(PID).dat **Example:** Sit1DL0_F50_2024-04-14T11-16-01_P05.dat		
Name	**Definition**	**Possible Values**
ACT	Name of Activity	Walk, FastWalk, InclineWalk, Sit
T#	Trial Number	Any positive integer
DL#	Data Log Number	Any positive integer
R#	Restriction Angle	20, 30, 40, 100 if ACT is Walk type 50, 60, 70, 100 if ACT is Sit type
YYYY-MM-DD	Year-Month-Day Timestamp	NA
HH-MM-SS	24 Hour-Minute-Second Timestamp	NA
PID	Participant ID	P0X if X is 1, 5-9 PXX if XX is 10-19

4.2.2 DATA FORMAT

The data collected by the DCU are simple text files, although they have the extension .dat to separate them easily from any .txt files that may be on the user's computer. An example row of data, and the definitions of the each of the columns as they appear is provided in Figure 4-2. These rows represent typical data, though other data do occur in the data files.

The first line of a data file is the time that the current file started collecting data, both as a real world time, and a DCU timestamp relative to the duration which the system has been powered. This line is prefixed with the letter "S" for "Start".

The last few lines of data at the end of the file are important pieces of metadata about the collection. The first of these lines is the “End” line, prefixed with an “E”, which provides a real-world and DCU-relative timestamp for the point in time when all of the data collected from the sensors had been written into the data file, prior to closing off the file. The next line is information about the duration of time that elapses from the start button being pressed to the acquisition of the first datum. This is provided in microseconds, to try and give as much resolution as possible. This information is typically only useful for synchronisation purposes when using the sync port on the DCU to trigger a remote collection simultaneously with another system. The last line of the file is the sensor resolution used when capturing the data of that file, given in Gs of acceleration for the accelerometer and degrees per second for the gyroscope.

Example Data From: Act: Walk, Trial: 3, Restriction: 100, PID: P06												
Col #	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Line No.	**Time (ms)**	**IMU ID**	**Accelerometer (mG)**			**Gyroscope (dps)**			**Magnetometer (uT)**			**Temp (°C)**
			X	**Y**	**Z**	**X**	**Y**	**Z**	**X**	**Y**	**Z**	
14015	690550	0	-605.96	1178.71	29.3	5.66	-66.53	108.02	-104.55	85.65	-15.75	28.56
14016	690557	2	-181.64	350.1	-56.15	52.99	-43.13	-48.03	6	37.35	11.7	28.6
14017	690563	0	-518.55	1141.6	130.37	8.56	-72.46	105.47	-105.9	85.65	-16.05	28.56
14018	690569	2	240.72	326.17	178.22	55.89	-36.43	4.29	4.65	37.2	10.05	28.41

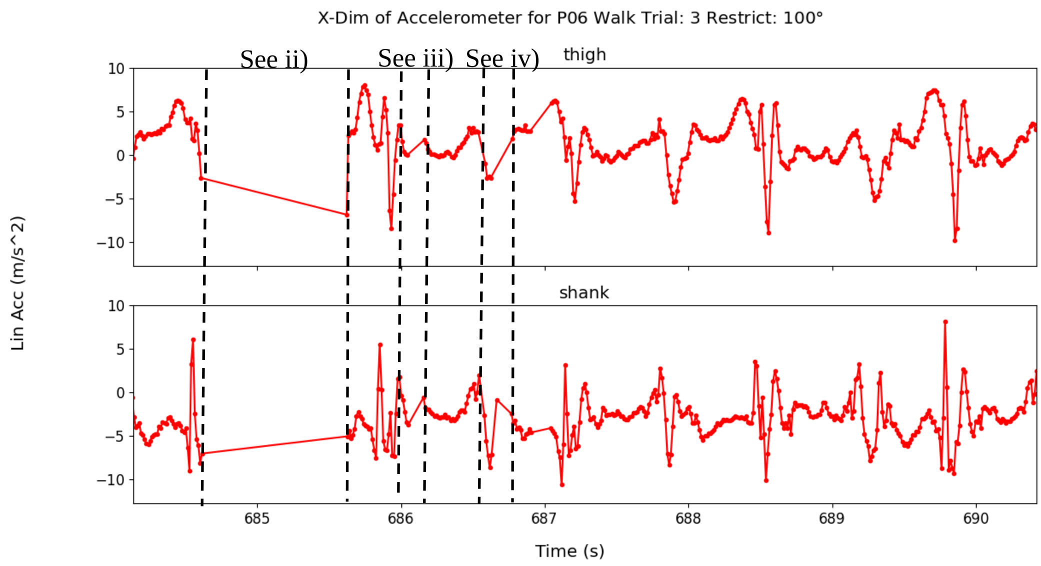
Figure 4-2: An example of rows of data taken from a file of P06 walking data. Notice the heading of each of the columns, which indicate what each column represents in the raw text data. The line number column is shaded in grey to denote its absence from the data as an actual column; it is presented only to show where the data are in the file. From left to right, data are: timestamp in milliseconds, an IMU ID, X, Y, Z accelerometer data, in milliGs, X, Y, Z gyroscope data in degrees per second, X, Y, Z magnetometer data in microteslas, and temperature in degrees Celsius.

4.2.3 DATA ERRORS

An unfortunate possibility of producing and using a custom device for data collection is experimental and buggy performance; in our case specifically, this meant the possibility of losses of data and collection failures. In some cases, the DCU will fail to communicate with a sensor, causing any incoming data from that sensor to be lost. If such a situation occurs, the system will attempt to recover by restarting the sensor, which tends to fix the issue. It was never identified why the sensors would fail in this way, but it was speculated to be either the length of the wires connecting the DCU to the sensor, or the complexity of the multi-threaded software architecture.

These losses mean that the data can often have gaps where no data occur. For an example, see Figure 4-3. In these cases, these errors serve as breaking points between otherwise continuous streams of data. Data before an error are considered a continuous block, and data after the error are considered a continuous block, with the error being captured and parsed for information such as time of occurrence, sensor that produced the error, the error produced, etc. In general, errors tend to be highlighted by the presence of identifying text when the error occurs, followed by a sensor reset. While the system resets the error-causing sensor, any activity being performed by the participant is not being captured, leading to the gap. In Figure 4-3ii), we can see an example of errorless loss, which is thought to have been due to a thread taking too long to complete a task, but without having a failure in a sensor.

i)



ii)

```
684598 0 13.18 863.28 85.94 16.32 1.97 -30.03 -120.75 72.60 -20.25 27.98
684604 2 -833.01 917.48 -767.58 40.40 5.85 -126.70 -44.40 25.95 7.35 28.70
684610 0 -271.97 1045.90 -266.11 20.95 -12.75 -37.05 -120.45 72.90 -19.50 28.37
684616 2 -716.80 892.09 -479.98 52.98 -35.07 -122.53 -43.80 27.30 6.75 28.41
685622 0 -702.15 1196.29 -450.68 63.35 -189.51 -207.56 -120.45 72.90 -19.50 28.37
685628 2 -515.14 1043.46 60.55 -35.76 34.55 242.24 -31.35 39.00 1.80 28.41
685634 0 225.59 933.11 60.06 8.60 23.98 32.26 -125.55 72.90 -27.30 28.56
```

iii)

```
686039 2 -348.14 1008.79 -111.82 3.68 47.15 -44.02 -36.00 30.90 5.85 28.41
686045 0 -2.44 1062.01 -108.89 14.44 38.73 -61.74 -115.35 80.85 -20.55 28.51
686052 2 -375.49 1008.79 -144.04 5.07 33.21 -43.04 -34.05 32.70 7.65 28.51
686053 ERR on 0 - Data Underflow
686151 RESET
686157 2 -58.59 952.64 -60.55 3.11 -3.73 -27.36 -30.15 32.40 8.10 28.60
686163 0 175.78 932.62 -30.27 1.98 3.44 -63.05 -110.10 82.05 -21.30 28.37
686169 2 -179.20 1015.63 -82.52 -0.58 7.28 -29.15 -28.50 36.00 6.45 28.84
```

iv)

```
686612 0 -255.37 1175.78 40.53 -4.75 -0.05 93.48 -102.45 84.30 -13.05 28.22
686618 2 -882.32 832.52 -444.82 19.79 -3.11 -114.78 -1.05 37.50 11.85 28.56
686624 0 -271.48 1123.54 -28.32 -2.17 -8.60 88.98 -100.95 86.10 -14.10 28.56
686630 2 -732.42 635.74 -359.37 29.19 -11.04 -100.06 -1.05 37.50 11.85 28.51
686632 ERR on 0 - Data Underflow
686660 RESET
686666 2 -84.47 375.49 -0.98 41.62 -17.18 -41.69 2.85 36.75 12.15 28.56
686668 ERR on 0 - Data Underflow
686763 RESET
686769 2 -253.42 906.25 73.73 18.60 15.71 136.35 -3.60 40.50 7.50 28.70
686775 0 187.50 1032.71 151.86 16.76 -18.34 110.35 -112.95 81.45 -22.20 28.41
686781 2 -360.84 1022.95 66.41 13.59 18.35 154.05 -4.65 40.50 6.75 28.56
686788 0 274.90 973.14 154.30 19.76 -12.46 108.00 -115.65 80.85 -22.65 28.37
686794 2 -331.05 1033.20 47.85 13.39 23.39 160.75 -8.10 40.80 7.35 28.51
```

Figure 4-3: Data for a collection showcasing some gaps in the data, both visually (i) and in the collected raw data (ii), iii), iv)), where errors have occurred. The P06 collection was particularly unhealthy from a data capture perspective, shown here. In ii), you can see the jump of ~1000 ms, without any explicit error occurring; this loss is caused by issues unrelated to sensor failure. In iii) and iv), more typical issues are seen, where a sensor has a failure and ~100-150ms are lost in the reset process. iv) shows the possibility of having multiple errors occurring in rapid succession, even after a successful reset.

4.2.4 PARSING THROUGH THE DATA

Knowing the different types of information we should expect in the data files, we parse through the data, creating a python (version 3.8.1) program to do so. We term the continuous blocks of data that are separated by errors as "fragments" of data, as the data have been fragmented from one continuous block into many continuous blocks by these errors. We keep track of relevant metadata for each file, such as participant ID, activity being performed, restriction level, and trial number, as well as the previously mentioned metadata within the file itself, in addition to the data fragments.

Various other loading-time parameters are of importance. For instance, consider Figure 4-4 which presents the beginning and ending of the same file of data for both sensors. Although efforts were made to keep the time prior to the start of an activity (start-lag) and the time prior to stopping the DCU (end-lag) consistent, some variance occurs. Furthermore, we elect to cut the beginning and end of the data, to eliminate start and end effects as the participants is getting up to speed and slowing down on the treadmill[62,190]. This is not as necessary for the sitting data, where we only want to cut off the stillness of the participant standing at the beginning and end of the data.

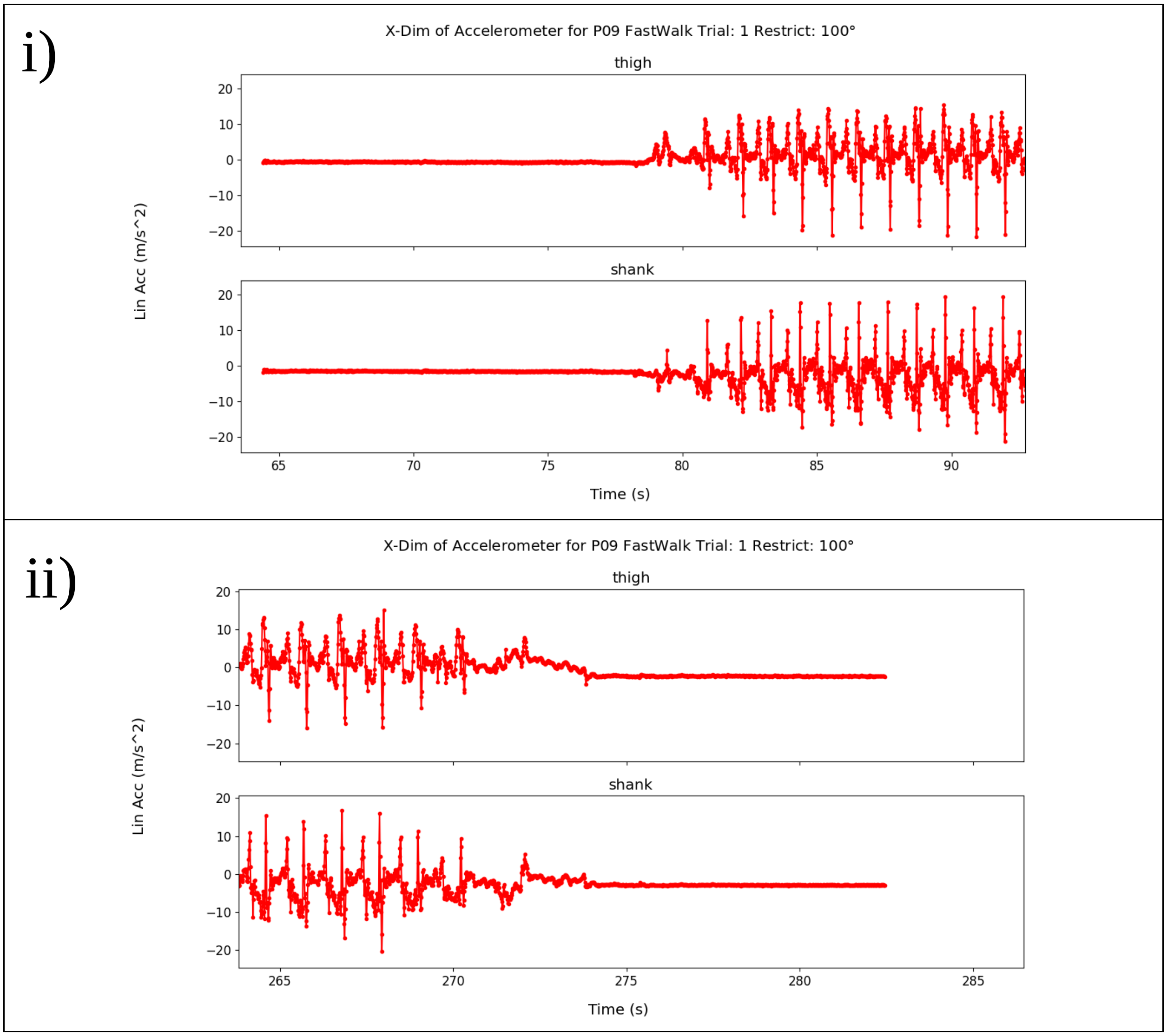


Figure 4-4: Graphs showing the beginning and end of unrestricted fast walk trial 1 for P09. Notice that for both IMUs, there is a period of stillness, associated with the participant standing still prior to the treadmill starting, and prior to the DCU being stopped after the activity. These periods of time need to be cut out of the data so they do not interfere with the results of analysis. Further, for the walking type data shown above, we want to cut further into the beginning and end of the data, to remove the effects of the treadmill ramping up to speed and ramping down from speed.

Other parameters of importance are a "short" threshold and a "deletion" threshold, which will help us decide what fragment is too short to be of importance, and how many of the data need to be lost in short fragments before we decide to delete the file altogether for being too damaged. We naively set the deletion threshold to 50%, and had a more involved process for determining the short threshold.

4.2.4.1 SHORT THRESHOLD DETERMINATION

The key idea for short threshold determination was to ask "how many data would be logically too little to be useful to us?" Upon consideration, it made sense that if a fragment was shorter than the duration of a cycle of the fastest (largest frequency) activity, it would not be useful to us. Thus, we had to determine how to find the period and frequency of the different activities, and then find the largest frequency, in order to find the shortest period. Doing this over all the participants would give us a global sense of what is the lowest amount of data we need.

We manually looked at the data and attempted to find the period of the data, across a few participants, to get a point of comparison for an automated approach. See Table 4-2 for the participant-averaged measured values. The data chosen for manual measurement came from both male and female participants, and had relatively clean (minimally fragmented) data.

Table 4-2: Averaged period lengths in milliseconds and derived frequency values in hertz, manually measured from data for P01, P05, P08 and P09, across all restriction levels. Measurements were made for each IMU and trial, and then averaged together to get a rough estimate for the activity at that restriction level, for that participant. The participants chosen were both male and female, so that these data could be used to gauge the effectiveness of automated approaches on all participants. Rst. is restriction level, IWalk is incline walk, FWalk is fast walk.

Participant-Trial-IMU-Averaged Frequency and Period Average +/- Std. Dev.								
Rst.	**Severe**		**Moderate**		**Light**		**None**	
	Freq (Hz)	**Period (ms)**	**Freq (Hz)**	**Period (ms)**	**Freq (Hz)**	**Period (ms)**	**Freq (Hz)**	**Period (ms)**
Walk	0.820 ± 0.0177	1219.6 ± 26.3	0.821 ± 0.0274	1219.7 ± 40.7	0.842 ± 0.0203	1188.4 ± 28.6	0.831 ± 0.0158	1203.3 ± 22.8
IWalk	0.805 ± 0.0381	1244.3 ± 58.9	0.815 ± 0.0543	1232.2 ± 82.1	0.818 ± 0.0384	1224.5 ± 57.4	0.800 ± 0.0267	1250.8 ± 41.7
FWalk	0.925 ± 0.0284	1081.8 ± 33.2	0.908 ± 0.0235	1101.8 ± 28.5	0.917 ± 0.0185	1090.6 ± 22.0	0.920 ± 0.0254	1087.2 ± 30.0
Sit	0.16002 ± 0.036592	6594.2 ± 1507.9	0.18939 ± 0.049393	5665.4 ± 1477.5	0.18690 ± 0.037107	5570.0 ± 1105.9	0.17978 ± 0.034242	5771.7 ± 1099.3

Once the manual results were computed, a frequency transform was done on the raw, untrimmed, data, and filtered with a 15Hz low-pass 4th Order Butterworth filter. The cut-off frequency of 15Hz was chosen due to human gait information being constrained to ~20Hz[191,192]. The goal was to approximate the frequency of the movement using easily detectable frequency spectrum landmarks. Further, the frequency information and information derived from it could be of use as discriminating factors in activity prediction[110,193,194], thus the process served useful in learning to extract spectral information from the transform.

We considered the following data points from the spectrum:

- The largest peak, and the frequency at which it occurs
- The smallest peak and the frequency at which it occurs
- The smallest frequency to have a peak, and the amplitude of that peak
- The largest frequency to have a peak, and the amplitude of that peak

These frequencies were taken using `find_peaks` from scipy with a height threshold of the 90% of the FFT being analysed and the 0Hz component removed, and compared against the manually computed frequency, using the absolute difference between the captured and observed frequency to serve as an indicator of closeness.

Ultimately, the periods given by the frequencies of the largest |X(f)| values were found to be useful, and were calculated and averaged across all the participants, giving us a short threshold of ~560ms. In practice, we used values of up to 1 second without issue.

4.2.4.2 TRIAL NUMBER NORMALISATION

During collection, trial numbers can often go higher than 3 due to DCU failures or other complications, thus the three trials that are successful can have unintuitive, non-contiguous trial numbers. Therefore, trial numbers are normalised in the program so that the three successful trials being used for analysis are labelled 1 to 3 from the smallest trial number to the largest one. This allows for straightforward comparison across all participants.

4.3 RESTRICTION TRENDS

During collection, participants performed the same activities at different restriction levels (unrestricted, light, moderate, severe). We wanted to differentiate between these restriction levels by finding factors that were different across them. We found that simple measurements across the entire time series did not yield any useful results, and so a more nuanced approach was required to finding where the data differed as a function of restriction.

4.3.1 PHYSICAL PHENOMENA

It was critical to consider the physical nature of the data we had collected to be able to extract meaningful insights from them which would separate out the restriction levels. To do so, we instrumented the researcher as all the other participants were, on the right leg, with a sensor placed on the thigh and a sensor on the shank. A video was recorded simultaneously with the data recorded by the DCU. The video was referenced against the data to be able to accurately identify where in the time series the major points along the gait cycle[1,2] occur for walking-type data, as well as for sitting type data. The major gait events are shown in Figure 4-6 and annotated in the time series in Figure 4-7 for walking, and for Figure 4-8 and Figure 4-9 for sitting. Note the frame of reference provided for the data in Figure 4-5. For walking, we pay closer attention to the shank data than the thigh, as it is not the thigh's movement that is restricted, rather, the lower leg will be affected by restrictions. For sitting, we pay closer attention to the thigh data than the shank, as the shank does not tend to engage with the activity as meaningfully.

Figure 4-5: Frame of reference for the gyroscope and accelerometer of the IMUs. This frame is correct relative to the IMUs in the images shown in Figure 4-6 and Figure 4-8, i.e. for both IMUs, X acceleration is positive to the left and Y acceleration is postive to the bottom. We do not concern ourselves with the Z axis, as most of our obvious movements occur in the XY (saggittal) plane. For the gyroscope, counterclockwise rotation in the (XY) saggittal plane is positive, and clockwise rotation is negative. For more information on the reference frame of the DCU see EMBEDDED SYSTEM.

Gait Phase	Start Collection	Toe-Off	Swing	Heel Strike	Stance
Image					

Figure 4-6: Major points along the gait phase, paired with images of the event occurring in a demo collection performed by the researcher, for the purposes of illustration. The occurrences of these events along the time axis can be seen in Figure 4-7. The simultaneous capture of video as well as DCU data allowed us to easily demarcate where the events occur in the time series, and thus, get a better sense of what we were looking for in the data. Note the gait cycle will repeat from toe-off to stance until we end the collection.

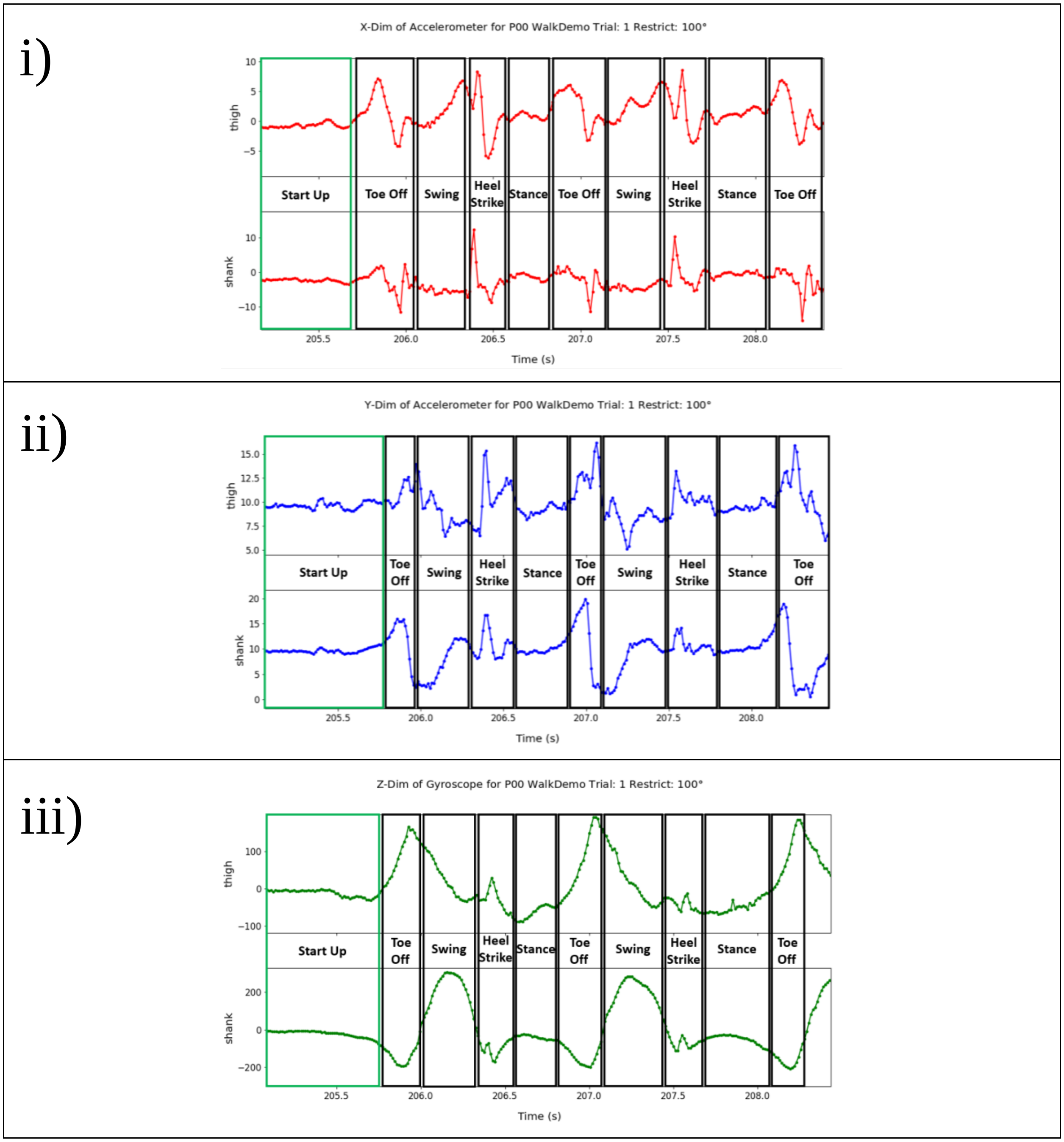


Figure 4-7: The major gait events labelled as they occurred in the time domain of the illustrative collection. In general, these patterns held for the other types of walking as well. Of particular interest to us were the XY dimensions of acceleration (see i) and ii)), and the Z dimension of angular velocity (see iii)), as these data lay in the sagittal plane, which is where the most pronounced movements occur.

As a participant moves through the gait phases of walking, the accelerations in X and Y are influenced both by the movement of the individual, as well as the rotating IMU changing how much of the movement is aligned with the axes we examine. In general, the key ideas of each of the phases relative to the data we have considered is as follows:

- For X Acceleration: During toe-off, the participant exerts an acceleration in the positive X direction, before moving the leg towards the negative X direction. This negative acceleration continues to throughout the swing phase, until we reach the heel strike, or initial contact. At this point, we have a large positive acceleration applied to the leg as we change that leg's direction. During stance, acceleration remains stagnant in the X direction, until we once again approach toe-off.
- For Y Acceleration: During toe-off, the participant exerts a force, causing an acceleration in the direction of gravity in the Y direction, thus briefly causing the acceleration to exceed gravity. During the swing phase, centripetal acceleration is opposite to the gravity direction, causing a significant reduction in positive acceleration. As the leg slow down towards the end of swing and approaches heel strike, the Y direction acceleration approach gravity again. The occurrence of heel strike causes a disturbance to this value, and once stance phase occurs, the acceleration stays close to gravity, until it once again starts increasing as we approach toe-off.
- For Z Angular Velocity: As we approach and have toe-off occur, the angular velocity of the shank is negative, as we are rotating clockwise. The point at which this velocity becomes positive, we have entered swing phase, rotating the shank in a counter-clockwise direction. Eventually, the angular velocity reduces, and becomes 0, at which point the leg has stopped swinging, and is approaching heel strike. Once heel strike occurs, the shank angular velocity decreases, as the leg straightens out in a clockwise direction. As the stance phase continues, the shank is close to still, allowing the angular velocity to reach close 0. The angular velocity starts to decrease as we approach toe-off.

Based on the physical movement of the leg, we would expect that restricting the knee should cause an effect during toe-off and swing. If the knee is restricted, is should be limited in how far back it can go during toe-off, manifesting as some sort of difference at that region of the time series.

Similarly, during swing phase, if the leg has travelled back less distance, it should also have less distance to swing forward, manifesting as some sort of difference in that region of the time series.

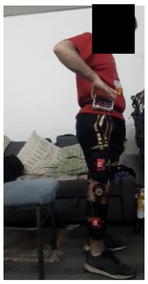
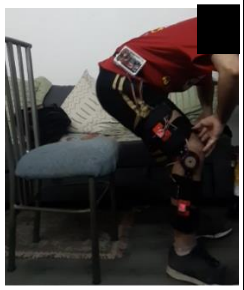
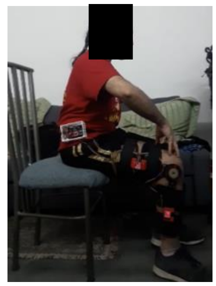
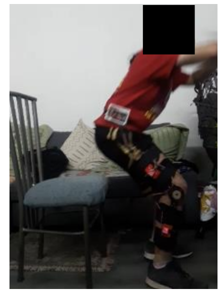
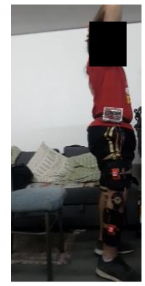
Movement Phase	Starting Collection	Negotiating Sit	Sitting in Chair	Negotiating Rise	Standing
Image					

Figure 4-8: Major points during the sit-to-stand activity, paired with images of the event occurring in a demo collection performed by the researcher, for the purposes of illustration. The occurrences of these events along the time axis can be seen in Figure 4-9. The simultaneous capture of video as well as DCU data allowed us to easily demarcate where the events occur in the time series, and thus, get a better sense of what we were looking for in the data. Note the movements will repeat from falling into chair to standing until we end the collection.

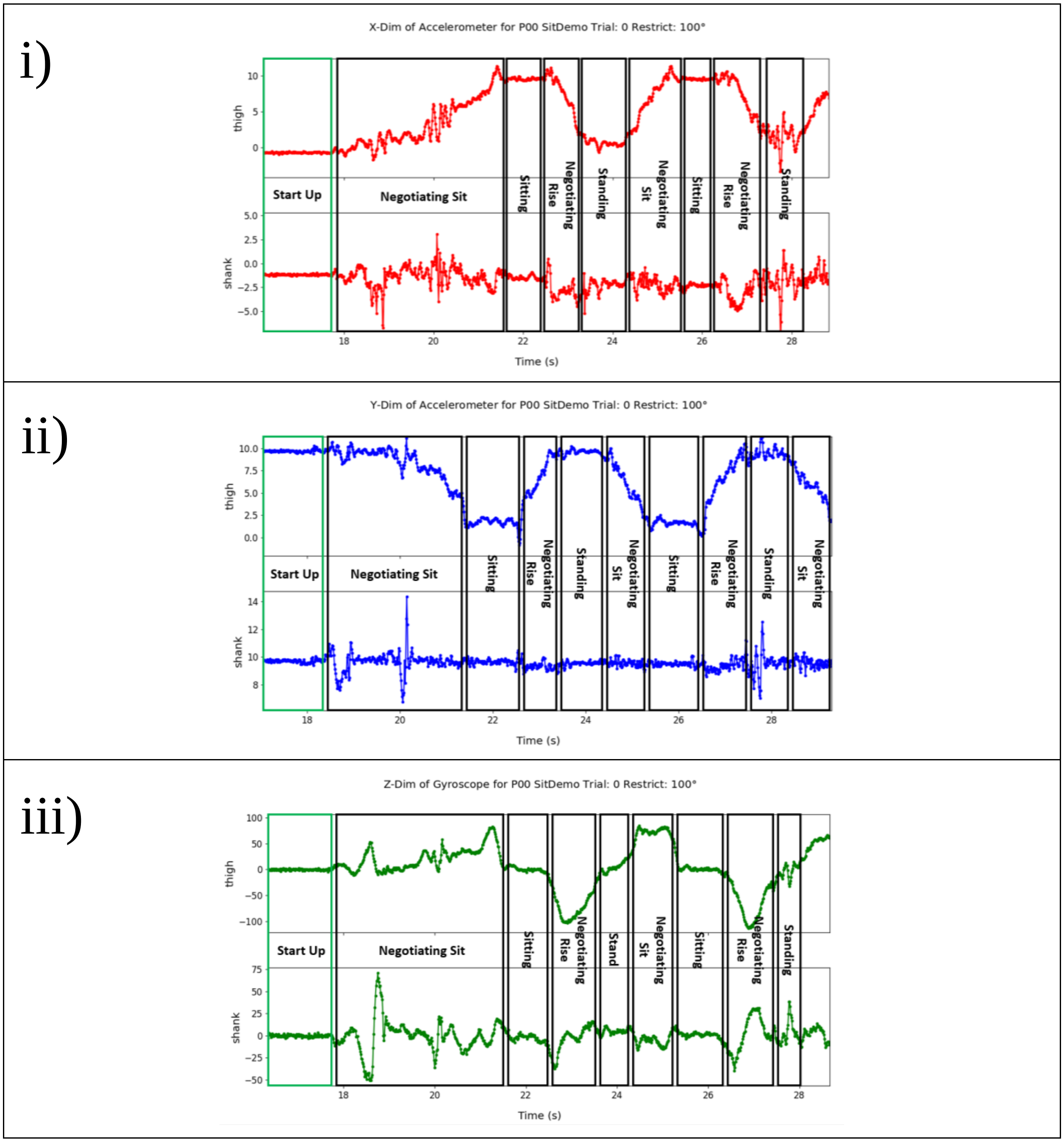


Figure 4-9: The major movement phases labelled as they occurred in the time domain of the illustrative sit-to-stand collection. Of particular interest to us were the XY dimensions of acceleration (see i) and ii)), and the Z dimension of angular velocity (see iii)), as these data lay in the sagittal plane, which is where the most pronounced movements occur.

For sitting data, the acceleration and angular velocity are greatly impacted by the rotation of the IMUs relative to gravity, as well as some effect of the forces applied by the participant to perform the motion. Notice that the shank does not engage in much interesting behaviour during sit. This is to be expected, as in most of our collection cases, once participants found a stable footing that allowed successful completion of the activity, the lower legs did not move for the rest of the activity. In the few cases where the legs did move, it was only to adjust slightly, which again, does not show up as interesting behaviour in the graphs. The general idea behind the major events of sit-to-stand relative to the data we collected is as follows:

- For Acceleration X: As the participant begins to negotiate the sit, the must control their descent into the chair. The X dimension goes from feeling no gravity, to feeling all the gravity acceleration once the descent is complete. Similarly, when negotiating the rise out of the chair, the acceleration felt in the X dimension reduces back close to 0 as the participant gets closer to standing. At standing, the X dimension acceleration is close to 0.
- For Acceleration Y: As the participant begins to negotiate the sit, the acceleration due to gravity slowly becomes perpendicular to the Y axis. Once the participant has sat in the chair, almost no acceleration due to gravity should be felt in the Y direction. As the participant begins to negotiate their rise out the chair, the Y axis begins to align back with the direction of acceleration due to gravity, so that when they stand, acceleration in the Y is almost gravity.
- For Angular Velocity Z: As the participant begins to negotiate the sit, they are engaging in the rotation counter-clockwise, and so we see a positive angular velocity. Once the participant is seated, the angular velocity is close to 0. As the participant negotiates getting out of the chair, they are rotating the thigh clockwise, causing a negative angular velocity, until the participant is standing, at which point the angular velocity is once again briefly 0.

For sitting data, we would expect to see meaningful differences in the data as the participants negotiate their way into and out of the chair, as these events are when they are engaging with the restrictions. Particularly, we'll expect to see effects in the thigh, which is performing most of the activity, as opposed to the shank, which will tend to stay still.

Having mechanically-based expectations of where to look in our time series for restriction-dependant variations, we then set out to extract out these regions of interest in an automated way, so that they could be examined and compared.

4.3.2 EXTRACTING REGIONS OF INTEREST

4.3.2.1 REGIONS OF INTEREST FOR WALKING-TYPE ACTIVITIES

When searching for which dimension of data to extract out for analysis in the walking-type activities, initially, acceleration data were considered, but extraction from the acceleration time series proved to be difficult. The data had many small peaks and valleys that made them hard to adapt an automated technique over all the participants and data. Thus, for walking-type activities, we settled on the smoother and easier to work with Z-dimension gyroscope data, specifically for the shank IMU.

One of the nice features of the Z-dimension gyroscope data was that it was very easy to separate out toe-off region from the swing phase region. Consider Figure 4-7, which shows some demo walking data. Notice that the toe-off event occurs in areas where the data are consistently less than 0, and the swing phase occurs in areas where the data are consistently greater than 0. Using this pattern, we could more easily extract out swing phase and toe-off phase from the time series. We hypothesised that there should be differences across the restrictions for the maximum swing angular velocity and the minimum toe-off angular velocity, but to check this, we needed to capture all the regions of interest from the time series.

4.3.2.1.1 SWING PHASE EXTRACTION

The swing phase could be found in the Z-dimension angular velocity data in contiguous regions of greater than 0 data. This is because a constant positive angular velocity meant a constant counter-clockwise rotation, which was emblematic of swing phase. We were interested in the maximum angular velocity of the swing phase, and we reasoned that applying restrictions would produce a change in this value. However, once we separated the swing phase from the rest of the data, we captured each swing phase region in its entirety, to analyse it later.

To separate out the swing phase regions, we found that using the 80^{th} percentile value of the Z-dimension gyroscope time series worked as an acceptable threshold value, above which, mostly only swing-phase data would occur. This served as a rough separation method. Once this was done, we had to find a way to differentiate each swing phase region of the time series from all the others, so we could capture metrics on each swing phase of a particular trial of an activity. To accomplish this, we considered that each swing phase should not connect to adjacent swing phases, so we could collect contiguous points and group them together as swing phases. Unfortunately, not all these groups were perfect, and some fine-tuning had to be done to allow for a satisfactory technique which would work over all participants at all restrictions. Parameters that needed to be adjusted were a minimum group size (for this region, a minimum group size of 0 worked fine), and a maximum allowable distance between non-adjacent groups (10 points of data ≈ 125ms of data worked well for this region), typically relevant for more erratic data. Additional care had to be taken to not consider regions that occur across a break in the data as a single group. See Figure 4-10 for details. Even still, there were malformed regions that may have been captured, so outlier removal is done on the analysis side. See Figure 4-11 for a visualisation of the technique.

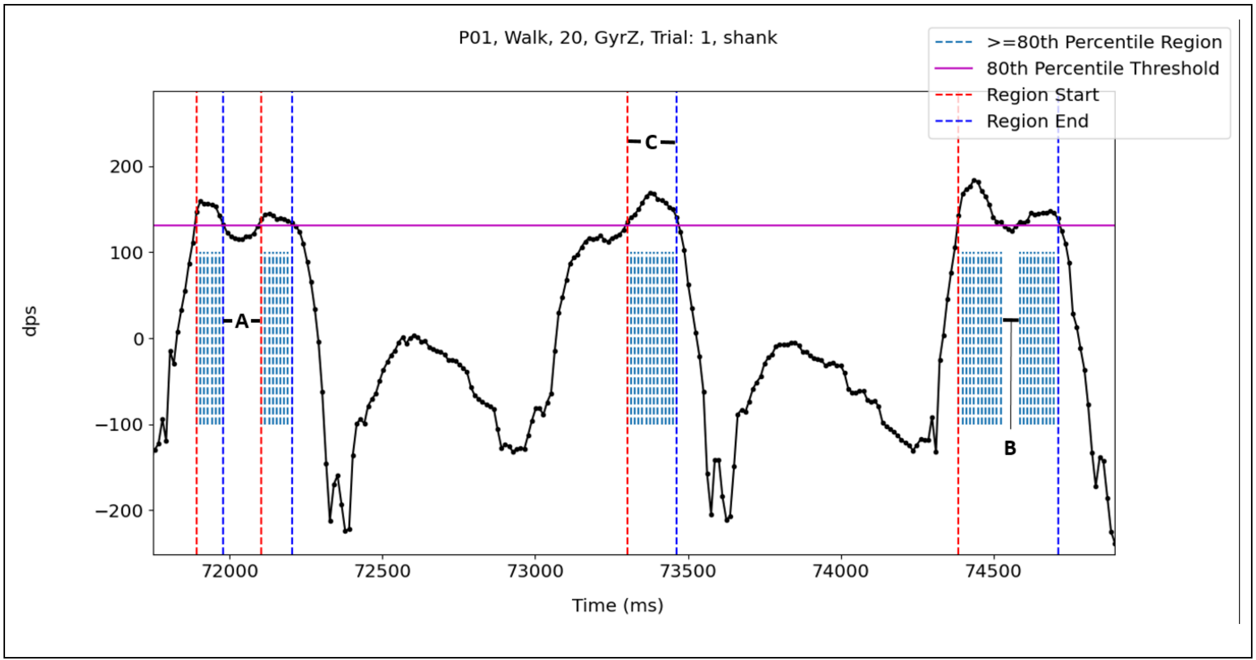


Figure 4-10: A sample from P01 data to show the purpose of various parameters used to control our swing phase capture technique. A shows the possible gap between groups, even though both captured regions capture the same swing phase. In cases like this, a maximum allowable gap parameter is kept, to allow for some tolerance if groups are "close enough" to each other, even though they may not be perfectly contiguous. B shows an instance where two non-contiguous groups are close enough to be grouped together as one group. C shows the group size of a single contiguous group. A parameter can be used to prevent groups that are captured by our technique, but would be "too small" or "too big" to be used for our purposes.

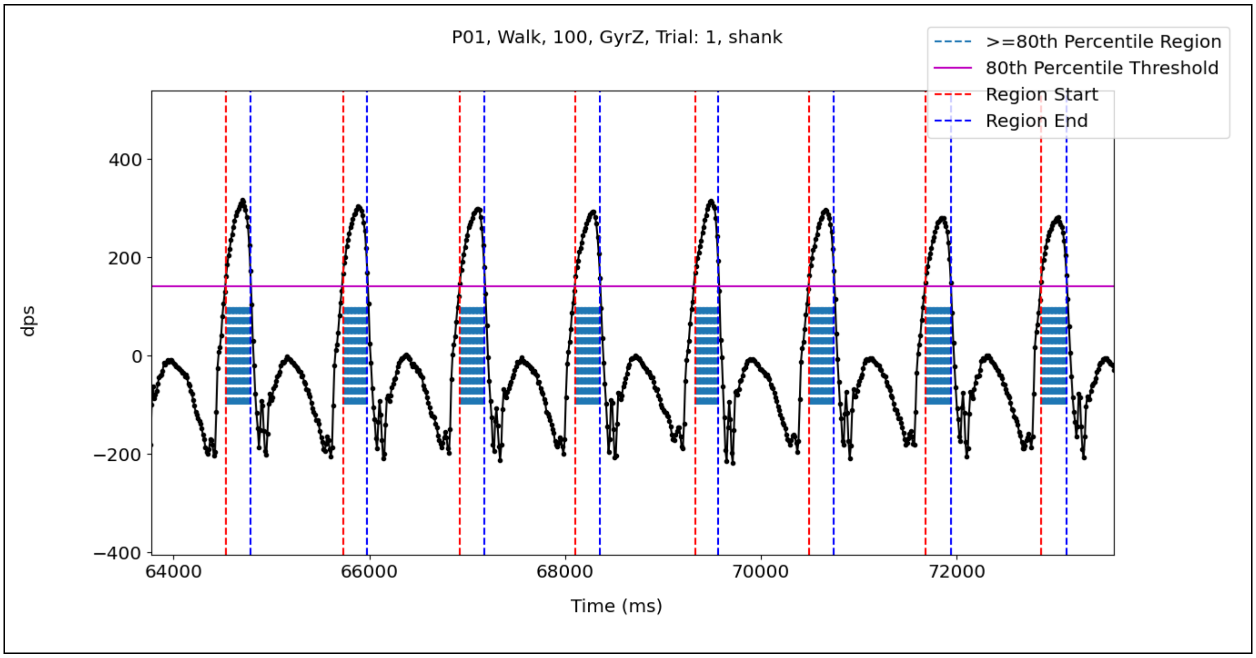


Figure 4-11: A visualisation of the areas of the time series that the swing-capture technique is trying to collect. We first find regions of the time series that are greater than our threshold of the 80th percentile of the time series. Once these areas are found, we partition them by grouping together contiguous areas of point, shown in between the red and blue dashed lines above. We collect the entire highlighted region for analysis later.

4.3.2.1.2 TOE-OFF PHASE EXTRACTION

We apply mostly the same technique as in 4.3.2.1.1, except we capture regions of angular velocity that are less than zero. Looking at Figure 4-7iii), and Figure 4-12, it is clear that areas of the time series that are less than or equal to 0 will collect both toe-off and heel strike regions. Fortunately, toe-offs tended to occur at lower angular velocities, so it was relatively simple to find the specific area within the region with which we were concerned.

Like in the swing phase case, we have to temper our selection from all regions less than zero to groups of a minimum length (half a second works well in practice), and not occurring across breaks in time series. We also found that having a maximum group length was necessary for capture in this region (1 second worked well enough in practice). We kept the same maximum allowable distance between non-adjacent groups that was used in the swing phase capture. We saved all the regions for analysis later. See Figure 4-12 for a visualisation of the technique.

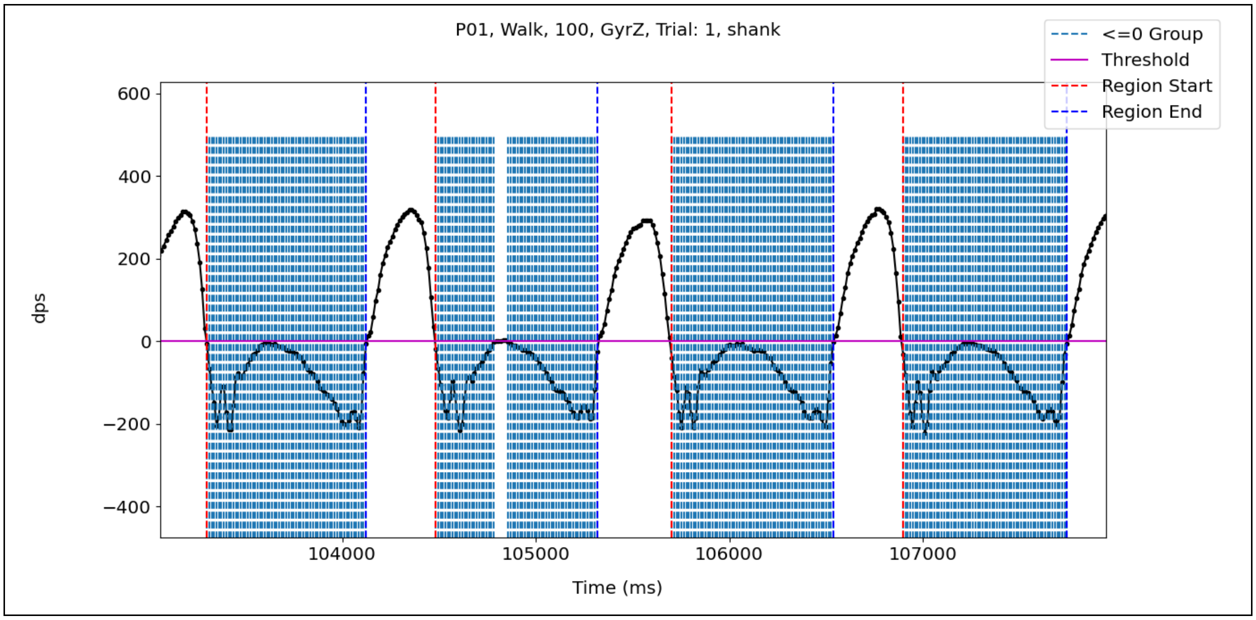


Figure 4-12: A visualisation of the areas of the time series that the toeoff-capture technique is trying to collect. We first find regions of the time series that are less than or equal to zero, exploiting the fact that toe-off occurs with negative angular velocity. Once these areas are found, we partition them by grouping together contiguous areas of point, shown in between the red and blue dashed lines above. We collect the entire highlighted region for analysis later. Note that unlike the regions in Figure 4-11, the regions here have two gait events captured – heel strike and toe-off.

4.3.2.2 REGIONS OF INTEREST FOR SIT-TO-STAND ACTIVITY

For the sit-to-stand activity, the acceleration X data were simpler to parse, and so they were used to collect the relevant regions of interest. In the X dimension of thigh acceleration, we had almost a square wave of data in the cleanest cases, where the rising edge of the waveform represented the participant negotiating their way into sitting in the chair, and the falling edge of the waveform represented the participant negotiating their way out of the chair. It was hypothesised that these areas were of particular interest when looking for differentiating factors across restriction levels. Thus, we set out to capture these regions of interest, using a technique that was different to that used in 4.3.2.1. The key observation was that the rising edge of waveform would have a roughly positive derivative and the falling edge of the waveform would have a roughly negative derivative, with sitting and standing activities having close derivatives close to 0. Thus, we built a technique around extract these regions using derivative.

Initially, we took the time derivative of the X-dimension acceleration for the thigh data, and tried to use them to extract out the rising and falling edges of the waveform. This process did not work as cleanly as we wanted, due to noise and other polluting factors in the data. To get a smoother derivative, we found that taking a windowed double average worked at smoothing out the derivative and giving us data that were useful in separating the underlying time series. Experimentally, a window of 50 points was found to be useful, and was passed along the time derivative. All the points in the window were averaged, and the window was moved forward one point in time, where the process was repeated. Once we had computed the window-averaged derivative for the entire derivative, we repeated the procedure over the window-averaged derivative. This gave us the window-averaged window-averaged (WAWA) derivative.

In practice, we found that the calculated derivative did not go to exactly zero during sitting and standing as we had anticipated, and so a threshold of the 50th percentile of the WAWA derivative with a buffer area of ± 100 m/s^3 was used to cleave the regions where the derivative was negative from where the derivative was positive, and thus, cleave the rising edge (negotiating sitting) and falling edge (negotiating standing) regions of the time series. Once we had these criteria in place, we could apply the same ideas from 4.3.2.1, namely, finding groups of contiguous points, while considering minimum group length (20 points = 250ms worked well for this region) and maximum allowable distance between groups (20 points = 250ms worked well for this region). As before, care had to be taken to prevent collecting groups across breaks in the time series a single group. We saved the rising and falling edges of the underlying time series sit-to-stand activity so that we could analyse it later. See Figure 4-13 for a visualisation of the technique used to capture the sit-to-stand edges.

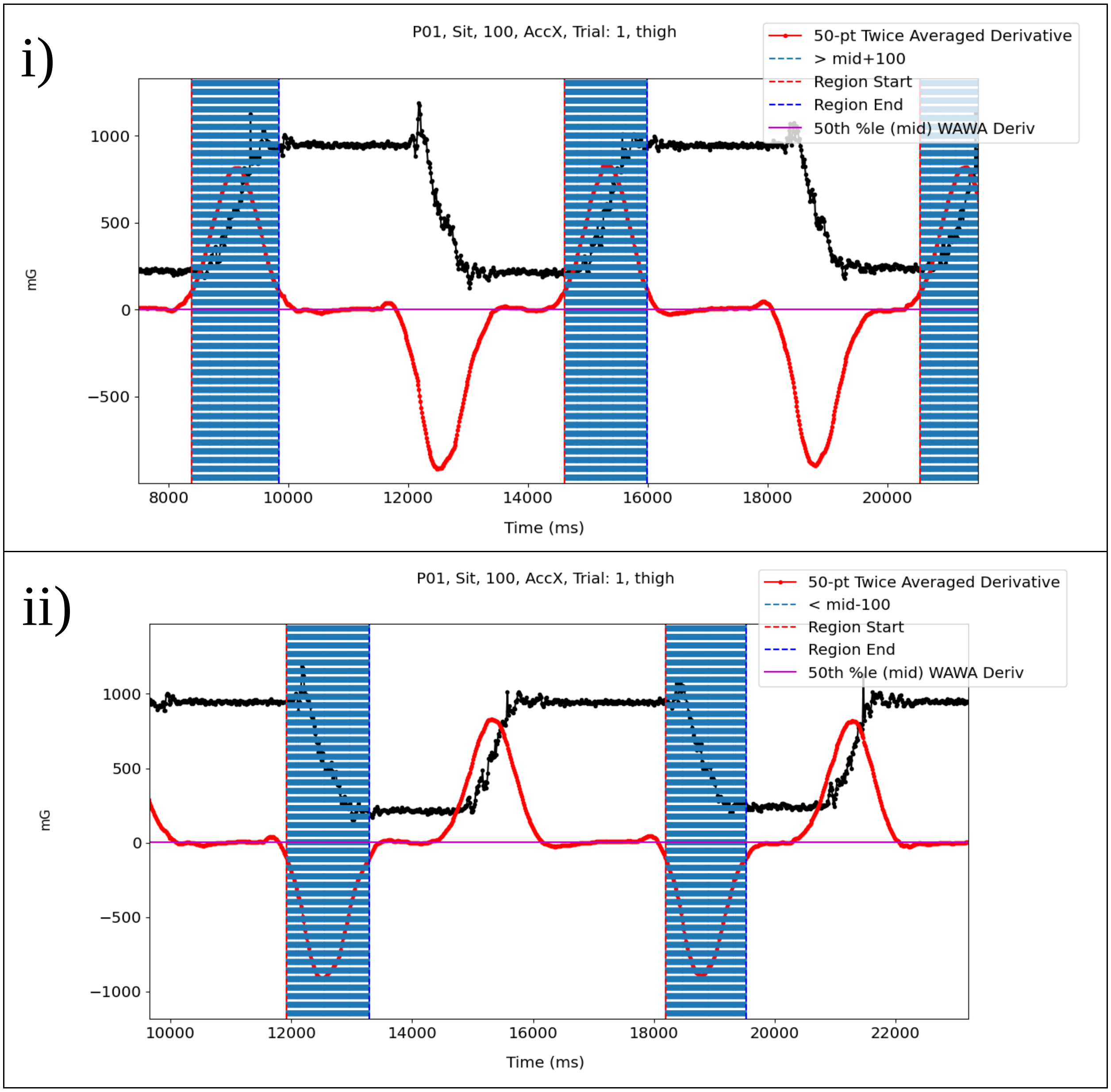


Figure 4-13: A visualisation of the areas of the time series that the edge capture technique is trying to collect. We take a windowed-average of the window-averaged derivative of the time series, and separate positive and negative derivatives using the 50th percentile of the calculated derivative. Once these areas are found, we partition them by grouping together contiguous areas of point, shown in between the red and blue dashed lines above. We collect the entire highlighted region for analysis later. i) shows the capture for rising edges (negotiating sitting), and ii) shows the capture for falling edges (negotiating standing).

4.3.3 TRENDS FOUND ACROSS RESTRICTIONS

Following the above methodologies to capture particular areas of interest for the time series data from all of our participants, across restrictions, a substantial set of data were collected. We captured the time series that occurred in the regions of interest and then extracted meaningful features from them during analysis.

4.3.3.1 WALKING TRENDS

For walking-type data, we expected that the maximum angular velocity of the swing phase, and the minimum angular velocity of the toe-off phase would be affected by restriction. This makes sense in a physical context, as the leg cannot swing back as far when restricted during toe-off, and has less distance in which to achieve its maximum angular velocity during swing phase because of it. Thus, we opted to collect these values from the captured regions of the walking data. Note, for regions containing toe-off, since we had captured both heel strike and toe-off events, we found the minimum angular velocity at toe-off by first finding the halfway point of the region, and only concerning ourselves with the minimum in the second half of that time series. Since the heel strike chronologically occurs first, even if it had produced an angular velocity that was the lowest for the overall region, it would be ignored by our approach.

Once these values were captured for each occurrence in every file, outliers were removed across each independent file. We then calculate a mean and standard deviation over these points per file, and then average these results across trials of the same PID and activity and restriction level. Lastly we remove outliers across PIDs using the IQR method[195], before normalising the data across restriction levels.

When normalising, the maximum swing angular velocity data are divided by the largest value for each PID and activity, and the minimum toe-off angular velocity data are divided by the smallest value for each PID and activity. Once normalisation is done, we take the average across PIDs for each walking-type and restriction level, and plot the results, shown in for both un-normalised and normalised data in Figure 4-14 to Figure 4-17.

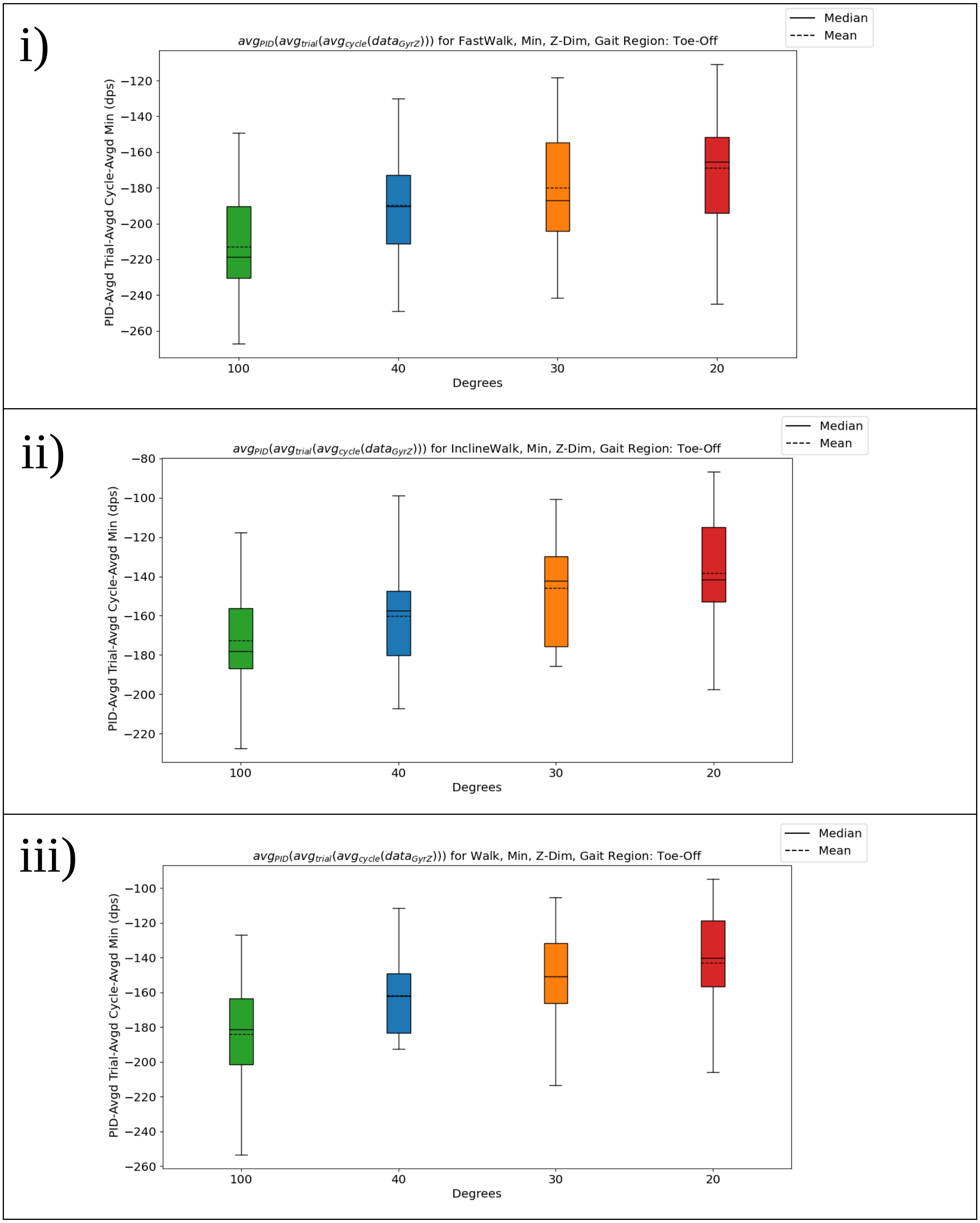


Figure 4-14: Boxplots for un-normalised data averaged out over PIDs. The data that are aggregated over are the minimum toe-off angular velocity for z-dim shank gyroscope. Results are shown for i) fast walking, ii) incline walking and iii) walking. Note that as restriction becomes more severe, the minimum toe-off angular velocity decreases.

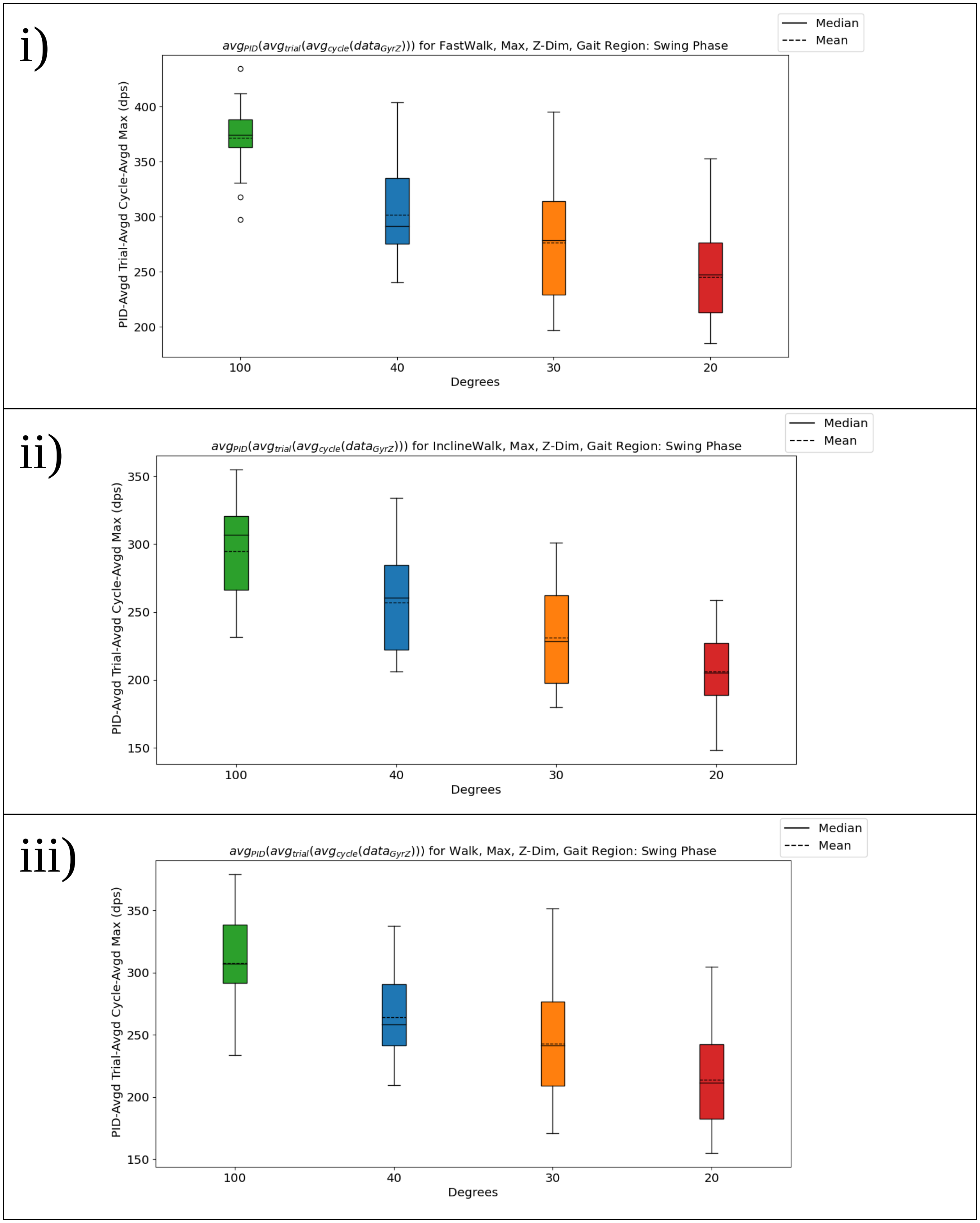


Figure 4-15: Boxplots for un-normalised data averaged out over PIDs. The data that are aggregated over are the maximum swing angular velocity for z-dim shank gyroscope. Results are shown for i) fast walking, ii) incline walking and iii) walking. Note that as restriction becomes more severe, the maximum swing angular velocity decreases.

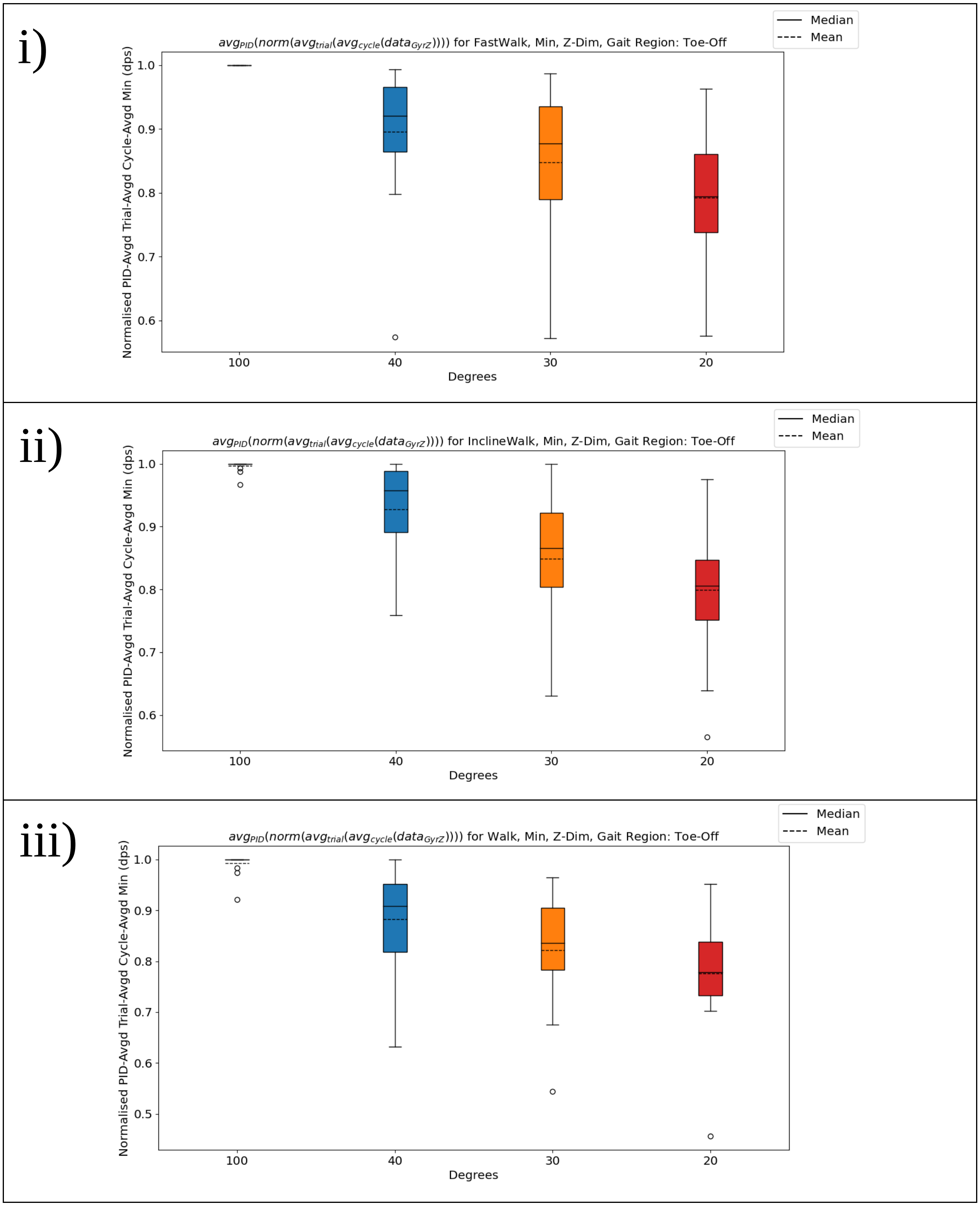


Figure 4-16: Boxplots for normalised data averaged out over PIDs. The data that are aggregated over are the minimum toe-off angular velocity for z-dim shank gyroscope. Results are shown for i) fast walking, ii) incline walking and iii) walking. Note that as restriction becomes more severe, normalised minimum toe-off angular velocity decreases.

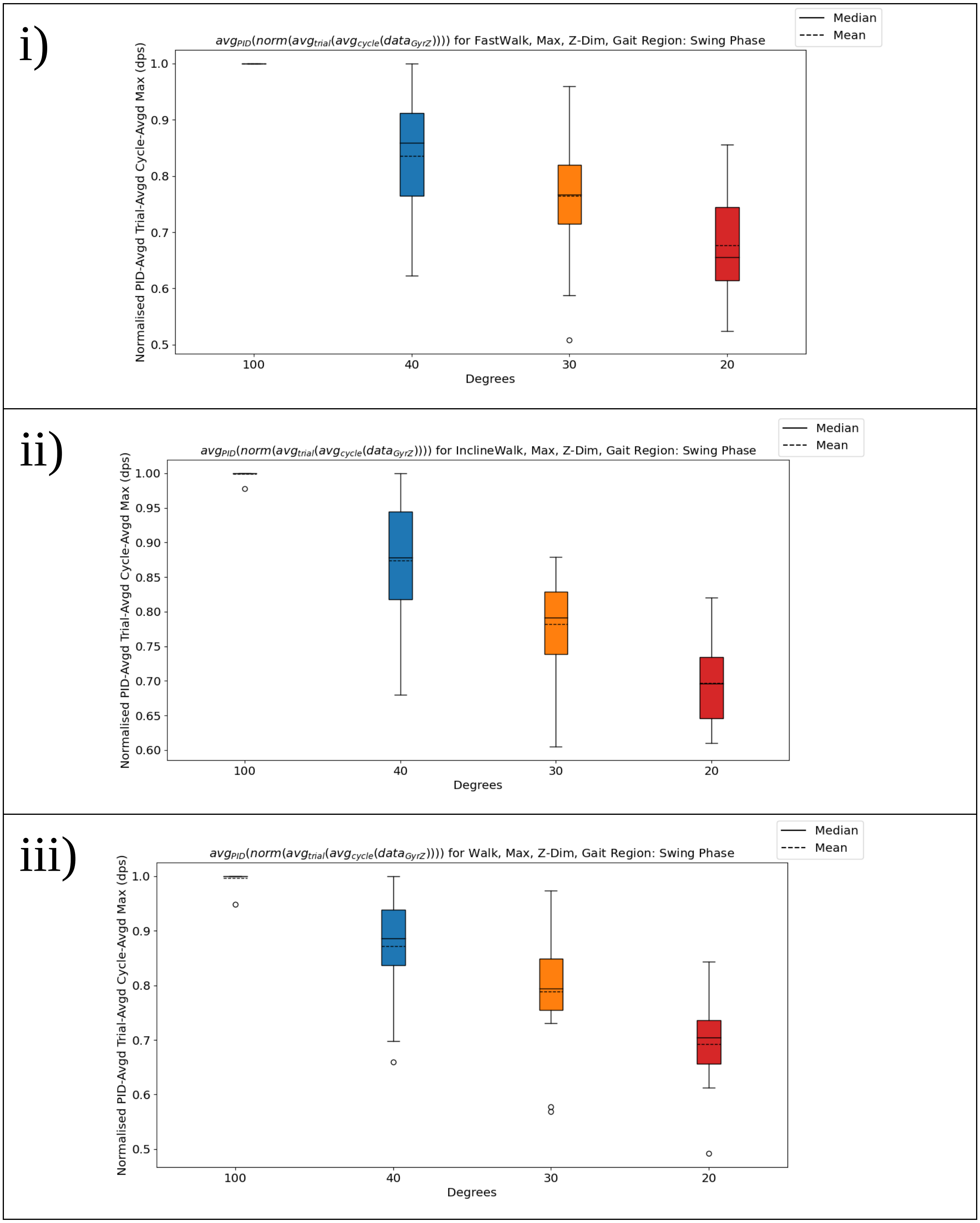


Figure 4-17: Boxplots for normalised data averaged out over PIDs. The data that are aggregated over are the maximum swing angular velocity for z-dim shank gyroscope. Results are shown for i) fast walking, ii) incline walking and iii) walking. Note that as restriction becomes more severe, normalised maximum swing angular velocity decreases.

4.3.3.2 SIT-TO-STAND TRENDS

For sit-to-stand data, we expected to see some difference in the harshness of the slope of the rising and falling edges of the time series, with the expectation that when the participant is more restricted, their descent into the chair and ascent from it become more careful. For the sit data, we separately collected averages of the derivatives over the rising and falling regions, as well as the standard deviations of the slopes, to capture the idea of "uncertainty" in the movements.

Once the chosen metrics of each of the captured regions were calculated, we did a standard inter-quartile range outlier[195] removal, to get rid of any possibly erroneous captures from our capture technique, as well as removing any outliers from the data that may be otherwise present. We now had a single point that represented each rising or falling edge of the time series for each file. We collapsed these data down by averaging the metrics independently across all the rising or falling edges for each file, getting a single value to represent rising edges and a single value to represent falling edges for each file, for each metric. We then average out the results over the trials, to get trial-averaged average slope and average standard deviation for rising and falling edges, per PID, activity, and restriction level.

At this point, we want to make comparisons across participants, so we first remove outliers, following the same IQR method[195] as before. By looking through the data manually, we know that P06 had very damaged sit trials for all non-unrestricted collections, but which did not yield values that were outliers. Thus, we manually removed those collections from the data prior to processing the data across PIDs, to avoid having the damaged data affect any aggregations.

We normalise each participant's results, independently for each participant and activity, across restriction levels. We simply divided the mean data corresponding to the rising edge by the largest value for that PID and activity, and the smallest value for mean data corresponding to the falling edge. Since we are guaranteed the rising edges should have averages above >= 0 and the falling edges should have averages <= 0, we do not have to worry about sign issues with this normalisation technique. We also normalise the standard deviation, using the corresponding max value for that PID and activity. Un-normalised and normalised results averaged over PIDs are shown as boxplots, for average jerk in Figure 4-18 and standard deviation of jerk in Figure 4-19.

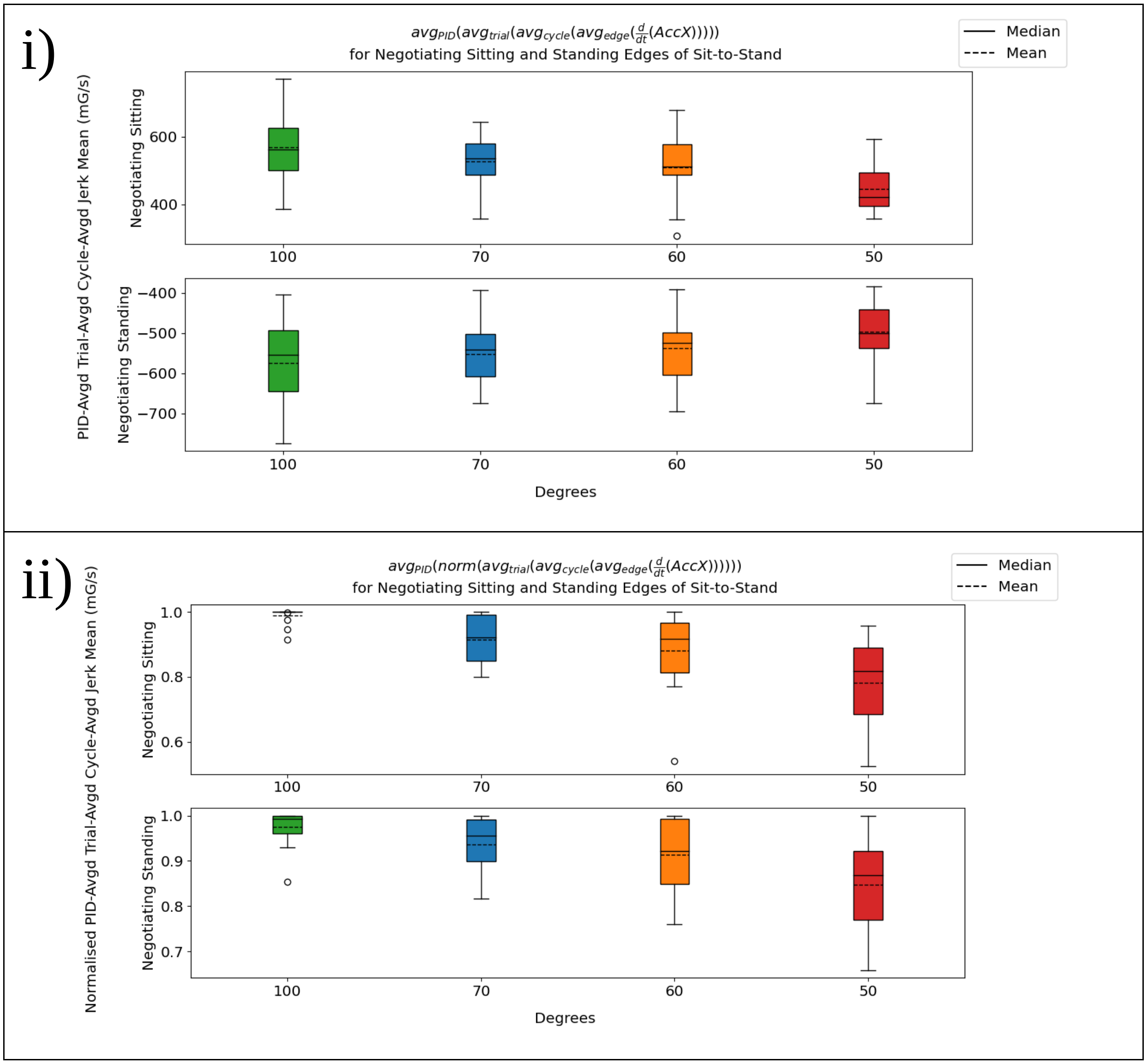


Figure 4-18: Comparison of average x-dim thigh acceleration derivative (jerk) values across restrictions, averaged over PIDs for negotiating sitting and negotiating standing regions of sit-to-stand data. i) shows un-normalised data, and ii) shows normalised data. In i), the trend for both negotiating sitting and standing shows a decreased magnitude for average jerk as restriction increases, which is the same trend shown in ii).

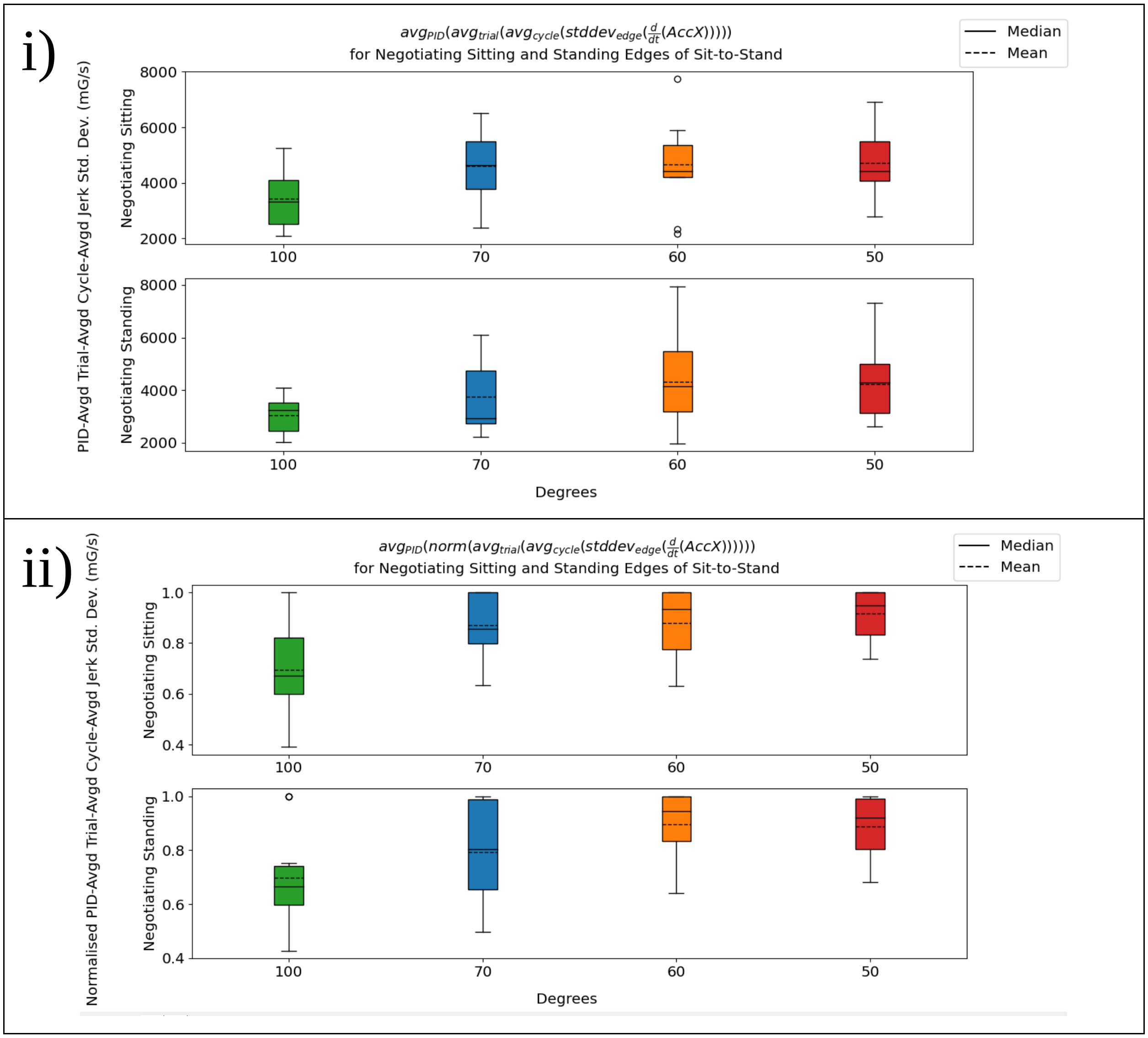


Figure 4-19: Comparison of standard deviation of x-dim thigh acceleration derivative (jerk) values across restrictions, averaged over PIDs for negotiating sitting and negotiating standing regions of sit-to-stand data. i) shows un-normalised results and ii) shows normalised results. In i) and ii), for negotiating sitting and standing, we see a general trend of increased average jerk standard deviation as restriction increases.

4.3.3.3 DISCUSSION AND STATISTICAL SIGNIFICANCE

In general, we can see that the trends that we expected for these regions of sitting and walking activities do appear to occur. During the different walking types, the minimum toe-off angular velocity and the maximum swing angular velocity are both reduced in an absolute sense by the strengthening of restriction at the knee. As we had mentioned before, this makes sense considering the physical nature of the activity, and how we would expect the restriction to affect the mechanics of the situation. We also see that during the negotiating sitting and negotiating standing phases of the sit-to-stand activity, a greater restriction reduces average jerk, and increases average jerk standard deviation, which both seem to indicate a more uncertain descent and ascent in more restricted scenarios. This makes sense naturally, as a person who feels limited in their motion at the knee will be more careful and uncertain as they negotiate in and out of a chair. We can see that there is likely a meaningful difference across completely unrestricted and severely restricted data for sit-to-stand activities, though the light and moderately restricted values tend to have statistics that are close to each other.

Once results were computed, statistical significance had to be computed on the data to see if there was any true difference across restrictions. To do so, we followed the ANOVA procedure[196–199], as follows:

1. Check that the data are normal
2. Check that the data are homoscedastic
3. Check statistical significance using ANOVA
4. Check which categories are different using post-ANOVA test

Most of the heavy lifting for this process is done by various scipy functions. Note, the post-ANOVA function is specifically from scipy 1.16.2, which also necessitated the use of python 3.13, only for that statistical test. We use a confidence interval of 95% (level of significance 0.05). We treat values from different PIDs for a particular aggregated value at a particular restriction as different samples from the same distribution for these tests. For example, the values of trial-averaged cycle-averaged mean jerk for negotiating sitting computed for each PID are considered as independent samples for an underlying distribution of values at that restriction level.

4.3.3.3.1 DATA NORMALITY

The normalcy of the data was evaluated using scipy's `shapiro`[200] function. This function uses the Shapiro-Wilk test for normality, with the null hypothesis that the data are normal. The p-values of the test computed over all the PIDs by restriction are shown in Table 4-3 for the walking activities, and Table 4-4 for the sit-to-stand activity.

Table 4-3: Shapiro-Wilk normality test p-values, for walking-type activities. We want to test if the restriction distributions for each of the types of measures we calculated are normal, so that we can apply ANOVA. The measures that we test on are the trial-averaged cycle-averaged minimum gyroscope z-dim angular velocity at toe-off ("Min") or the trial-averaged cycle-averaged maximum gyroscope z-dim angular velocity at swing ("Max").

Walking-Type Activities Shapiro-Wilk Normality P-Values						
Activity	**Fast Walk**		**Incline Walk**		**Walk**	
Gait Region	**Toe-Off**	**Swing**	**Toe-Off**	**Swing**	**Toe-Off**	**Swing**
Data	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
Restriction						
Unrestricted	0.780	0.369	0.910	0.260	0.926	0.535
Light	0.991	0.345	0.648	0.530	0.269	0.844
Moderate	0.914	0.335	0.252	0.117	0.969	0.690
Severe	0.927	0.447	0.657	0.681	0.931	0.542

At a significance level of 0.05, we do not have sufficient evidence to reject the null hypothesis for any of the walking-type activities or any of the measures we have calculated, at any of the restriction levels. We carry all of these distributions and measures to the homoscedasticity test.

Table 4-4: Shapiro-Wilk normality test p-values, for sit-to-stand activity. We want to test if the restriction distributions for each of the types of measures we calculated are normal, so that we can apply ANOVA. The measures that we test on are the trial-averaged cycle-averaged mean of the jerk ("Jerk Mean") or the trial-averaged cycle-averaged standard deviation of the jerk ("Jerk Std. Dev.")

Sit-to-Stand Shapiro-Wilk Normality P-Values				
Movement Region	**Negotiating Sitting**		**Negotiating Standing**	
Data	**Jerk Mean**	**Jerk Std. Dev.**	**Jerk Mean**	**Jerk Std. Dev.**
Restriction				
Unrestricted	0.913	0.182	0.714	0.239
Light	0.715	0.670	0.795	0.039
Moderate	0.405	0.320	0.739	0.715
Severe	0.319	0.941	0.543	0.323

We find that with a significance level of 0.05, we reject the null hypothesis that the data come from a normal distribution for negotiating standing jerk standard deviation data at a light restriction level. Thus, we remove those data from the analysis, and continue the analysis on the remaining data.

4.3.3.3.2 DATA HOMOSCEDASTICITY

To test for homoscedasticity, we use scipy's `bartlett`[201] function, which tests the null hypothesis that the samples come populations with equal variances, with an assumption of normality on the tested populations. As prior, we test with a significance level of 0.05. The p-values of the test computed over all the PIDs by restriction are shown in Table 4-5 for the walking activities and in Table 4-6 for the sit-to-stand activity.

Table 4-5: Bartlett homoscedasticity test p-values, for walking-type activities. We want to test if the restriction distributions for each of the types of measures we calculated, share the same variance, so that we can apply ANOVA. The measures that we test on are the trial-averaged cycle-averaged minimum gyroscope z-dim angular velocity at toe-off ("Min") or the trial-averaged cycle-averaged maximum gyroscope z-dim angular velocity at swing ("Max").

Walking-Type Activities Bartlett Homoscedasticity P-Values						
Activity	**Fast Walk**	**Fast Walk**	**Incline Walk**	**Incline Walk**	**Walk**	**Walk**
Gait Region	**Toe-Off**	**Swing**	**Toe-Off**	**Swing**	**Toe-Off**	**Swing**
Measure	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
Value	0.978	0.584	0.980	0.943	0.624	0.846

At a significance level of 0.05, we do not have sufficient evidence to reject the null hypothesis that the restriction distributions have the same variance.

Table 4-6: Bartlett homoscedasticity test p-values, for sit-to-stand activity. We want to test if the restriction distributions for each of the types of measures we calculated, share the same variance, so that we can apply ANOVA. The measures that we test on are the trial-averaged cycle-averaged mean of the jerk ("Jerk Mean") or the trial-averaged cycle-averaged standard deviation of the jerk ("Jerk Std. Dev.")

Sit-to-Stand Bartlett Homoscedasticity P-Values				
Movement Region	**Negotiating Sitting**		**Negotiating Standing**	
Measure	**Jerk Mean**	**Jerk Std. Dev.**	**Jerk Mean**	**Jerk Std. Dev.**
Value	0.324	0.640	0.373	0.010

At a significance level of 0.05, we reject the null hypothesis that the data for negotiating standing jerk standard deviation have equal variance with the other tested distributions. Thus, we remove those data from our analysis and perform the ANOVA and post-ANOVA on the remaining data.

4.3.3.3.3 ANOVA AND POST-ANOVA

To test that the distributions from which our samples came from have different means, we used scipy's `f_oneway`[196] which implements a one-way ANOVA. The ANOVA has the null hypothesis that the samples we are testing come from distributions with the same mean, with the underlying assumption that the distributions are normal and have equal variances. As prior, we test with a significance level of 0.05. The results for walking-type activities are shown in Table 4-7 and the results for the sit-to-stand activity are shown in Table 4-8.

Table 4-7: One-way ANOVA test p-values, for walking-type activities. We want to test if the restriction distributions for each of the types of measures we calculated share the same mean. The measures that we test on are the trial-averaged cycle-averaged minimum gyroscope z-dim angular velocity at toe-off ("Min") or the trial-averaged cycle-averaged maximum gyroscope z-dim angular velocity at swing ("Max").

Walking-Type Activities One-Way ANOVA P-Values						
Activity	**Fast Walk**	**Fast Walk**	**Incline Walk**	**Incline Walk**	**Walk**	**Walk**
Gait Region	**Toe-Off**	**Swing**	**Toe-Off**	**Swing**	**Toe-Off**	**Swing**
Measure	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
Value	0.003	0.000	0.008	0.000	0.002	0.000

At a significance level of 0.05, there is sufficient evidence to reject the null hypothesis that all the restriction distributions have the same mean.

Table 4-8: One-way ANOVA test p-values, for sit-to-stand activity. We want to test if the restriction distributions for each of the types of measures we calculated share the same mean. The measures that we test on are the trial-averaged cycle-averaged mean of the jerk ("Jerk Mean") or the trial-averaged cycle-averaged standard deviation of the jerk ("Jerk Std. Dev.")

Sit-to-Stand One-Way ANOVA P-Values			
Movement Region	**Negotiating Sitting**		**Negotiating Standing**
Measure	**Jerk Mean**	**Jerk Std. Dev.**	**Jerk Mean**
Value	0.004	0.014	0.102

At a significance level of 0.05, we reject the null hypothesis that the data come from distributions that have the same mean, that is, some of the data across different restrictions come from distributions that have different means, using the trial-averaged cycle-averaged mean of jerk and standard deviation of jerk as measures. We fail to reject the null hypothesis using the data from the negotiating standing part of the time series.

To figure out which of the restriction distributions are different from the others, we must apply a post-ANOVA test; we use scipy's `tukey_hsd`[202], whose null hypothesis is that the distributions being compared have the same mean. The results of the Tukey test are given in Table 4-9 for walking-type activities, and Table 4-10 for the sit-to-stand activity.

Table 4-9: Post-ANOVA Tukey test p-values, for walking-type activities. This test compares the different restriction distributions against each other for data of each type that showed mean differences in the ANOVA. The measures that we test on are the trial-averaged cycle-averaged minimum gyroscope z-dim angular velocity at toe-off ("Min") or the trial-averaged cycle-averaged maximum gyroscope z-dim angular velocity at swing ("Max").

Walking-Type Activities Tukey P-Values				
Fast Walk Toe-Off Min				
Restriction Angle	**100**	**40**	**30**	**20**
100	1.000	0.199	0.032	0.002
40	0.199	1.000	0.843	0.303
30	0.032	0.843	1.000	0.785
20	0.002	0.303	0.785	1.000
Fast Walk Swing Max				
Restriction Angle	**100**	**40**	**30**	**20**
100	1.000	0.000	0.000	0.000
40	0.000	1.000	0.388	0.004
30	0.000	0.388	1.000	0.207
20	0.000	0.004	0.207	1.000
Incline Walk Toe-Off Min				
Restriction Angle	**100**	**40**	**30**	**20**
100	1.000	0.633	0.060	0.008
40	0.633	1.000	0.520	0.154
30	0.060	0.520	1.000	0.872
20	0.008	0.154	0.872	1.000
Incline Walk Swing Max				
Restriction Angle	**100**	**40**	**30**	**20**
100	1.000	0.022	0.000	0.000
40	0.022	1.000	0.186	0.001
30	0.000	0.186	1.000	0.218
20	0.000	0.001	0.218	1.000
Walk Toe-Off Min				
Restriction Angle	**100**	**40**	**30**	**20**
100	1.000	0.161	0.014	0.001
40	0.161	1.000	0.744	0.297
30	0.014	0.744	1.000	0.871
20	0.001	0.297	0.871	1.000
Walk Swing Max				
Restriction Angle	**100**	**40**	**30**	**20**
100	1.000	0.031	0.000	0.000
40	0.031	1.000	0.516	0.009
30	0.000	0.516	1.000	0.229
20	0.000	0.009	0.229	1.000

Using a significance level of 0.05, across all the walking-type activities, the swing phase z-dimension maximum angular velocity measure shows sufficient evidence to reject the null hypothesis when comparing 100° restricted data (unrestricted) against all the other restricted data. Further, this measure provides sufficient evidence across all walking-type activities to reject the null hypothesis when comparing 40° (light) data against 20° (severe) data. The z-dimension toe-off minimum angular velocity measure also shows sufficient evidence across all walking-type activities to reject the null hypothesis between 100° restricted data and 30° (moderate) or 20° (severe) restricted data.

Table 4-10: Post-ANOVA Tukey test p-values, for sit-to-stand activity. This test compares the different restriction distributions against each other for data of each type that showed mean differences in the ANOVA. The measures that we test on are the trial-averaged cycle-averaged mean of the jerk ("Jerk Mean") or the trial-averaged cycle-averaged standard deviation of the jerk ("Jerk Std. Dev.").

Sit-to-Stand Negotiating Sitting Jerk Mean Tukey P-Values				
Restriction Angle	**100**	**70**	**60**	**50**
100	1.000	0.571	0.255	0.002
70	0.571	1.000	0.943	0.073
60	0.255	0.943	1.000	0.230
50	0.002	0.073	0.230	1.000

Sit-to-Stand Negotiating Sitting Jerk Std. Dev. Tukey P-Values				
Restriction Angle	**100**	**70**	**60**	**50**
100	1.000	0.051	0.048	0.030
70	0.051	1.000	0.999	0.994
60	0.048	0.999	1.000	0.999
50	0.030	0.994	0.999	1.000

From the post-ANOVA results, we have sufficient evidence to reject the null hypothesis when comparing the restriction distributions from 100° (Unrestricted) from 60° (Moderate) and 50° (Severe). We just barely fail to reject the null hypothesis between 100° and 70°, and we are unable to reject the null hypothesis between any of the non-unrestricted groups.

4.4 ACTIVITY CLASSIFICATION

Participants performed four different activities (Walk, Fast Walk, Incline Walk, Sit-to-Stand) during our collection. The question posed was whether or not we could create a model that could predict which activity was being performed by the participant in a given file of data. To test this question, a neural network was created, which could take pieces of a data file as input and produce probabilities denoting how likely the data were to be from each of the activity classes as an output. The model would train in a supervised fashion, being shown the correct answer in order to have a target to train towards. This model was tested on four participants independently, on the completely unrestricted data, using only the acceleration channel.

4.4.1 PREPROCESSING AND DATA PREPARATION

Before being fed to the model, the data were to be preprocessed. Part of this was accomplished by the trimming the data, trial number normalisation, and short threshold removal discussed in 4.2. Further, the labelling of the files served to also produce the ground truth label that the model would use as its target during training.

The data would then need to be split up into two categories: training/validation and testing; the former is used to train the model and validate how well the training is performing, the latter is used exclusively to test the performance of the model after all training is complete. We collect a random third of the data for testing and split the remaining data into training and validation sets. In the analysis and results, we refer to each random set of testing data as a “fold”, and typically run three separate training and testing trials for each parameter set, leading to three “folds” of data.

First, the testing set was created, as follows:

```
for p in Participants:
      for a in activities(p):
            for r in restrictions(a,p):
                  randomly choose a file from files(p, a, r)
                  add chosen file to test set list
```

The testing set was chosen this way for the sake of balance. If we had opted to randomly choose a third of all the files for a given participant for a given restriction level, we could have imbalanced testing sets where only one type of walking may occur, or sitting could be completely absent, for example. Following the procedure shown above ensured a balanced testing set that represented all the types of data present.

Once the set of files to be set aside for testing were chosen, the remaining files were defined as the training/validation set. From these files, the division had to be made as to which data would be training, and which data would be validation. The split would be made over each file, splitting the time series into two. 70% of each file was kept for training, and 30% of each file was kept for validation. For each file, it was randomly chosen whether the training or validation data would be taken first. This randomization prevents the training and validation data from carrying possible unintended patterns that may occur only at the beginning or end of the file, causing a 'false-training' of the model.

Once all three sets of data were separated out, the data had to be normalised. This was done following the min-max normalisation technique, particularly following the implementation from sci-kit learn[203]. We normalised the data into the range of 0 to 1, normalising each set separately, but across all the files present in the set. Each data channel was normalised separately.

Of special importance is the timestamp data. As mentioned in 4.2.3, the data can suffer from gaps in the time series, sometimes of second or longer durations. Reconstruction of the data across the gaps was not performed, instead, the data were treated as continuous, with the timestamps encoding the fact that there was a time gap between the points. For instance, if data points d_0 and d_1 are not continuous in time, then the corresponding timestamps t_0 and t_1 would have a larger difference between then compared to the timestamps t_1 and t_2 of continuous data points d_1 and d_2. The timestamp data were also normalised, however, unlike the data channels, normalising the timestamps across files made no sense, as there was no relationship between the timestamps across files. Thus, the timestamps were normalised independently for each file, on a range of 0 to 1.

Further processing of the data was required at this point, in order to provide batches of data in the correct shapes that were expected by the model. Considerations had to be made towards how to handle the multiple channels, in addition to multiple IMUs. However, as these choices were now related to the specific model we built, it is instructive to discuss the model architecture.

4.4.2 MODEL ARCHITECTURE

The model that was built was a convolutional neural network (CNN). The idea was to use the convolutional layers to allow the model to see multiple dimensions and time points and "mix" them together to try and see a pattern in the data. The results of the convolved output would be passed to a linear layer, in order to both learn from the mixed outputs, and shape the data down into the correct shape to be compared against the ground truth label.

The input to the model is a 4D Tensor, whose dimensions are Batch x IMU x time x Data Columns. This input shape keeps all the important dimensions of the data separate, and is meant to be general enough to be useful for any model, which can then shape the data however it needs for its purposes. The length of time dimension of each of these inputs was selected to be five seconds, as it offers enough data relative to our short threshold value, thus, we expect to see multiple cycles of the fastest moving data, and fewer cycles of the slower moving data. When producing this 4D tensor, some adjustments can occur to the data. Firstly, both IMU's data are cut to be same shape, trimming off any time dimension data that prevent both IMUs from aligning. Next, if the data are not divisible by the chosen length of the time dimension ("chunk"), we repeat the last point until we have reached a length that is divisible. Lastly, when we are batching the data together, if we cannot produce a whole batch, but we have half a batch or more data, instead of throwing away those data, we reuse random time-contiguous pieces from that batch to complete it. If we do not have even a half of a batch of data, we throw those data away. This is similar to pytorch's own behaviour with dropping batches. See Figure 4-20 for a visual. The order of the elements of the batch are shuffled, so that the model does not learn to expect a particular order of input.

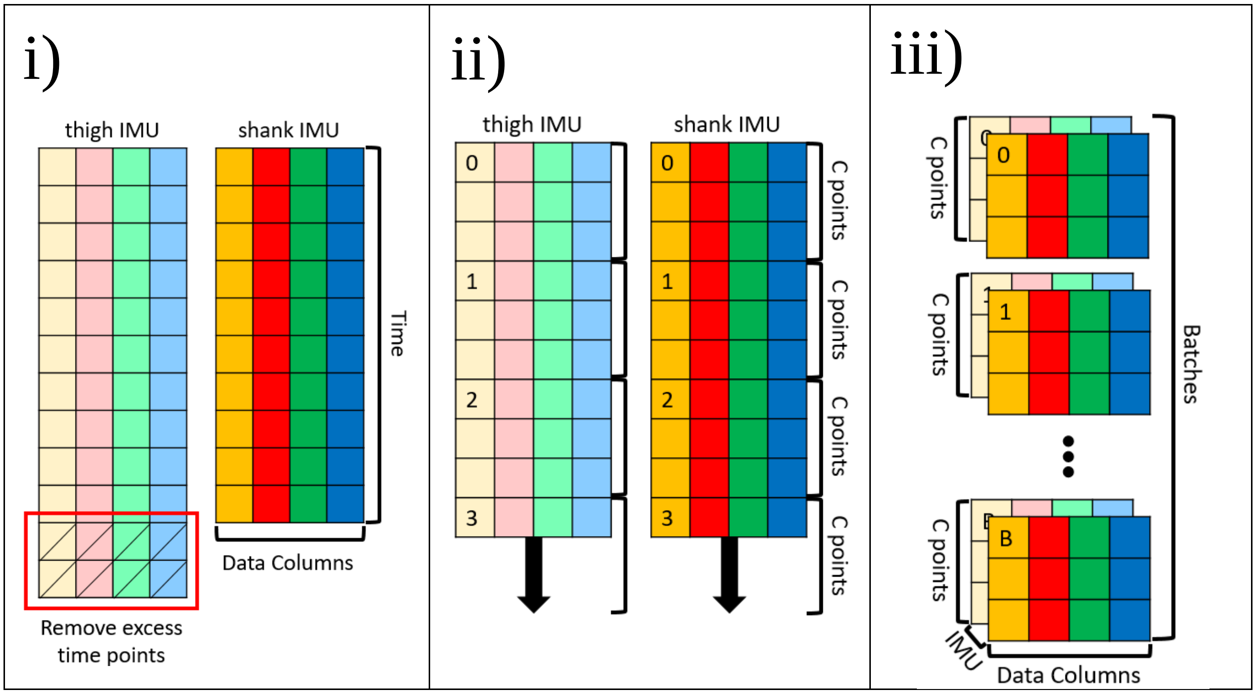


Figure 4-20: A visualisation of the data shaping that occurs prior to breaking the data into pieces to be shuffled, batched and sent to the model. i) shows the equalisation of the IMU data in time, so that the arrays can be combined together. ii) shows the extension of the last group of points to the nearest multiple of the chosen time chunk size, C. This extension is done by repeating the final point. iii) shows the batching of B-many C-time point chunks of data. If we have enough data for half or more of the last batch, we will repeat random chunks to complete the batch. The order of chunks in a batch are then shuffled before being passed to the model.

The model we have created reshapes the input into a 3D Tensor, where the IMU and data columns end up along the same dimension, stacked on top of each other, so that all the data from one IMU are together, and all the data from the other IMU are together. This gives us a shape of Batch x IMU * Data Columns x time. See Figure 4-21 for a visualisation.

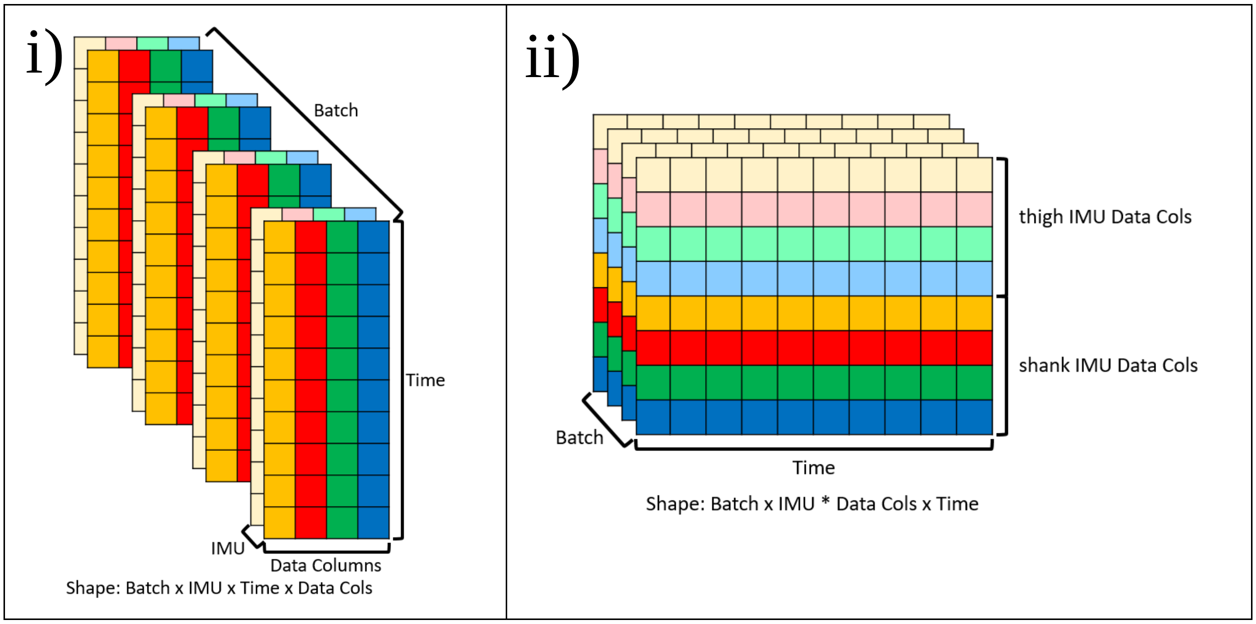


Figure 4-21: Shape of the input data, both as the shape prior to being passed to the model (i), and as the input prior to the first layer of the model (ii). The initial shape in i) is model agnostic and is kept that way so that any needed dimension can easily be separated, whereas our specific model requires data to be in the shape of ii). The batch dimension is used by the model to train on multiple data at a time, the IMU dimension separates the data by thigh or shank IMU, the time dimension separates different time points of the data, and the data column dimension separates the timestamps dimension and the X, Y, Z dimensions of acceleration.

The model itself consists of two 1D convolutional layers, whose output goes to a linear layer. The key ideas for each layer are presented as follows, and are visualised in Figure 4-22. The general structure for a logistic regression model is derived from Jurafsky and Martin[204].

- The first convolutional layer allows the model to convolve the input along the time dimension, treating each of the IMU * Data Column rows as independent channels. The kernel size (we call this the context window or context size) is chosen to be at least the short threshold size (see 4.2.4.1), as we expect only partial cycles to be below this size, and we want to allow our model to attempt to see cycles. We set the stride to be half the kernel size, so that we can see the transition from the end of one cycle to the start of another. The output of this layer is sent to a ReLU.
- The second convolutional layer allows the convolution to occur across the channels of the previous layer's output, with a kernel size encompassing the entire dimension. The idea is

to get the different dimensions to mix for each stride step of the previous convolution. The input to this layer is modified prior to convolution by adding in a "metrics" tensor, which stores information about the min, max, and mean of the data seen at that stride. These metrics are calculated on all the non-time channels of the data. The output of this layer is sent to a ReLU.

- The linear layer takes the results of the previous output, and passes it through a linear layer, in order to shape it down to the correct shape of Batch x Four, as we have four classes. The input to this linear layer has frequency data added to it, from each of the dimensions of non-time channel data. The frequency data are calculated as follows:
    - Produce the welch power spectral density of each non-time channel using scipy's `welch`[205], which computes the power spectral density (PSD) using averaging of multiple windows over the data. The default window of Hann is used, with an averaging technique of 'median', instead of 'mean', to help produce better results on discontinuous data[205].
    - Remove the PSD and frequency for the 0Hz component, as we are not interested in the DC component results.
    - Using a threshold height of the 90^{th} percentile of the calculated PSD, capture the values greater than the threshold. The choice of the 90^{th} percentile has experimentally worked well in ignoring most of the noise floor data.
    - Collect eight pieces of information from these results:
        1. The value of the smallest PSD peak
        2. The frequency at which the smallest PSD peak occurs
        3. The value of the largest PSD peak
        4. The frequency at which the largest PSD peak occurs
        5. The smallest frequency at which there is a PSD peak
        6. The value of the PSD peak at the smallest frequency where one occurs
        7. The largest frequency at which there is a PSD peak
        8. The value of the PSD peak at the largest frequency where one occurs

The output of this model, after going through a softmax function, is passed to the cross entropy loss function, and compared against a ground truth tensor. It outputs a loss result that is averaged across the entire batch. The goal of our model is to reduce this value.

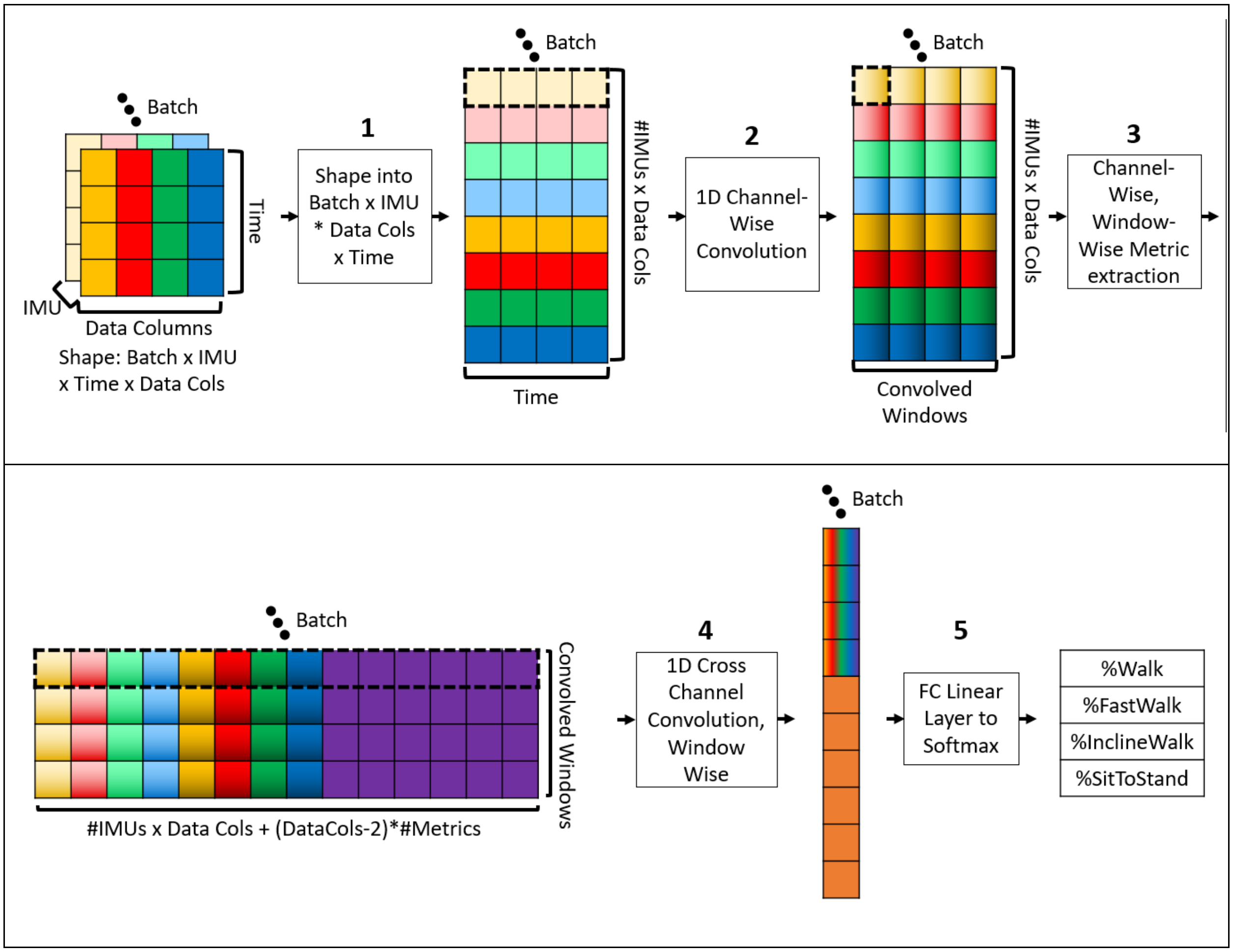


Figure 4-22: Visualisation of the model pipeline the data go through to get a prediction. We show the data in their basic shape, before they are passed to the model and internally reshaped at 1. Once reshaped, the data are passed through a 1D channel-wise convolution, with the dimension that is convolved along emphasised with a dotted-line box. Next, various metrics are extracted from each convolved window for each channel separately, excluding time channels, at 3, which are appended to the data prior to a data column-wide, window-wise 1D convolution at 4. The final column vector has additional metrics from 3 appended to it, and is then passed through a fully connected linear layer, to a softmax, for output of class probabilities at 5.

The ground truth tensor is created by keeping track of which file a tensor comes from (but not showing this information to the model), assigning a number to this activity (0 : Walk, 1: FastWalk, 2: Sit, 3: InclineWalk), and then one-hot encoding that number (each number becomes a four digit number, where only the position that corresponds to the value of that number is 1; the rest are zero). This produces a Batch x Four tensor, which we treat as the ground truth probability of the data coming from a particular activity type file; the activity that produced the data will have 100% probability, and the remaining activities will have 0% probability. This is a standard strategy to compute loss[206].

The model uses stochastic gradient descent as the optimiser. The optimiser updates the model weights if the validation set accuracy is better than the previous epoch, building towards a model that classifies the validation set as well as it can. To help aid in this process, we have a L2 regularisation added to the loss computed by the cross entropy function, in order to try to mediate weights that are too large (and thus, exhibit extreme and lopsided learning). Using pytorch's ready-made functionality, we set up the optimiser to reduce its learning rate as the epochs go on, to try and hone in on a region of the loss surface. Further, class weights are passed to the loss function. As shown in [207], the class weights can incentivise or de-incentivise decisions during training by making the loss of particular classes weigh more or less, using a multiplicative weight. The class weights are calculated on the entire training dataset, and takes into account the balance of activities in the set. It simply calculates the percentage of each activity present in the dataset, and uses those percentages as weights. Optionally, we allow the weights to be inverted by subtracting them from 1, so that the weights can either strongly emphasize inaccuracies of the more present classes (typically walking activities) and weakly emphasize inaccuracies of the less present classes (typically sit), or vice versa. In either case, we will not de-emphasize existing inaccuracies, as our weights are >=1.

4.4.3 DEFINING HYPERPARAMETER SETS

Activity classification was performed on four independent participants, on their unrestricted data. Through initial trials of the model, a set of well-learning values were chosen for the optimiser's hyper parameters and number of epochs. Variable sizes were chosen for the kernel size of the first convolutional layer, with a fixed size of 5 seconds being chosen as the time size of each initial 4D tensor. Batch sizes were not important accuracy-determining hyper parameters, but were chosen to help speed up training by allowing parallelised computing. The set of hyper parameters that were searched upon, and the possible values they were given is provided in Table 4-11.

We cleaned up the result of the Cartesian product of these possible values, as some of them were redundant, and would waste compute time if we did multiple instances of the same set. For example, it would not be useful to set class weights to False and have inversion of class weights be set to True and False, as those two hyper parameter sets would be identical, since the inversion would have no effect without the class weights. Similarly, other redundancies existed that were removed before starting the model.

Table 4-11: The hyper parameters that were used by the model, and thus likely influenced accuracy, and their possible values as we completed iterations over hyper parameters sets. We also cleaned out any redundant sets, such as not having both true and false inversion when class weights is false to begin with.

Hyper Parameter	Purpose	Values
Learning Rate	Defines how big the initial steps of the optimiser will be	0.1
Number of Epochs	Defines how many iterations the model will run for.	500
Gamma Value	Determines how much the learning rate decimates when we reach the predetermined decimation epoch	0.75
Decimation Timing	Determines how many epochs we should periodically decimate the learning rate using the gamma value	50
Dampening	Dampens the effect of the newly calculated gradient when computing new weights in SGD	0, 0.25, 0.5, 0.75, 1
Momentum	Dampens the effect of the previously computed gradients summed gradients in SGD	0, 0.25, 0.5, 0.75, 1
Inversion	Chooses whether or not the class weights to the loss function strongly emphasize less common classes (True) or more common classes (False) during training.	True, False
Regularisation	Chooses whether or not we apply L2 regularisation to our loss	True, False
Class Weights	Chooses whether or not we apply class weights to the loss function at all	True, False
Context	The value of the kernel for the first convolution of the model. It determines how much “context” the model has when it is convolving. These values are data points, at a roughly 80Hz data rate.	40, 60, 80
Chunk Size	The size of the time dimension of the tensors prior to being passed to the model. This value is in points of datum, where the data rate was 80Hz.	5*80

As mentioned in 4.4.1, we ran three trials (“folds”) for each model, and then averaged out the results over them to compare results.

4.4.4 MODEL RESULTS

Upon independently training and testing the model on four participants, P01, P05, P08, P09, over more than 350 hyper parameter sets, the following results were obtained. Results are presented in a fold-averaged, and single fold context, as in some cases, two folds may have performed very well, but a third fold may have significantly lowered the average. The results for P01 and P05 are shown in Table 4-12, and the results for P08 and P09 are shown in Table 4-13.

Table 4-12: Best fold-averaged +/- std. dev. and single fold accuracies at testing time for P01 and P05, over 371 hyper parameter sets. In general, single fold accuracies show better testing time accuracies, but do not have consistent accuracy when running that hyper parameter set on different folds. Comparatively, fold-averaged testing time accuracies show performance that works across different folds, and thus, is likely to work on other unseen data.

Best Fold-Averaged			**Best Fold-Averaged**		
P01			**P05**		
Hyper Param Set Num	**Fold**	**Accuracy**	**Hyper Param Set Num**	**Fold**	**Accuracy**
157	-	82% ± 6.8%	192	-	70% ± 6.0%
146	-	82% ± 9.9%	217	-	65% ± 5.9%
229	-	79% ± 5.4%	144	-	63% ± 4.4%
206	-	79% ± 3.8%	242	-	61% ± 5.8%
73	-	79% ± 10%	146	-	61% ± 13%
Best Single Fold			**Best Single Fold**		
P01			**P05**		
Hyper Param Set Num	**Fold**	**Accuracy**	**Hyper Param Set Num**	**Fold**	**Accuracy**
192	1	90.7%	192	2	75.3%
169	1	89.3%	146	2	75.3%
146	2	88.7%	194	2	74.7%
157	0	88.0%	72	2	74.7%
230	1	87.3%	229	2	74.0%

Table 4-13: Best fold-averaged +/- std. dev. and single fold accuracies at testing time for P08 and P09, over 371 hyper parameter sets. In general, single fold accuracies show better testing time accuracies, but do not have consistent accuracy when running that hyper parameter set on different folds. Comparatively, fold-averaged testing time accuracies show performance that works across different folds, and thus, is likely to work on other unseen data.

Best Fold-Averaged			**Best Fold-Averaged**		
P08			**P09**		
Hyper Param Set Num	**Fold**	**Accuracy**	**Hyper Param Set Num**	**Fold**	**Accuracy**
205	-	74% ± 4.0%	241	-	76% ± 3.3%
240	-	72% ± 1.5%	242	-	75% ± 6.7%
241	-	71% ± 4.1%	158	-	75% ± 6.3%
145	-	71% ± 5.2%	217	-	74% ± 2.4%
192	-	70% ± 4.2%	192	-	74% ± 7.4%
Best Single Fold			**Best Single Fold**		
P08			**P09**		
Hyper Param Set Num	**Fold**	**Accuracy**	**Hyper Param Set Num**	**Fold**	**Accuracy**
84	2	82.7%	84	1	84.0%
206	0	80.7%	205	2	83.3%
86	0	80.0%	72	0	83.3%
98	1	79.3%	146	2	82.0%
146	1	79.3%	230	1	82.0%

It is worth noting that there exist parameter sets that occur in multiple participant classifications for fold averaged or single fold results, implying that those hyper parameter sets are probably capable of classifying more participants well. For instance, hyper parameter set 192 offers some of the highest fold-averaged accuracies for P05, P08 and P09. Similarly, hyper parameter set 146 occurs across all top five single fold highest accuracies for all participants tested, which implies that that is a hyper parameter set of value.

We can visualise the performance of the model using confusion matrices, shown for the best fold-averaged accuracies for P01, P05, P08 and P09 in Figure 4-23 to Figure 4-26, along with model metrics per epoch.

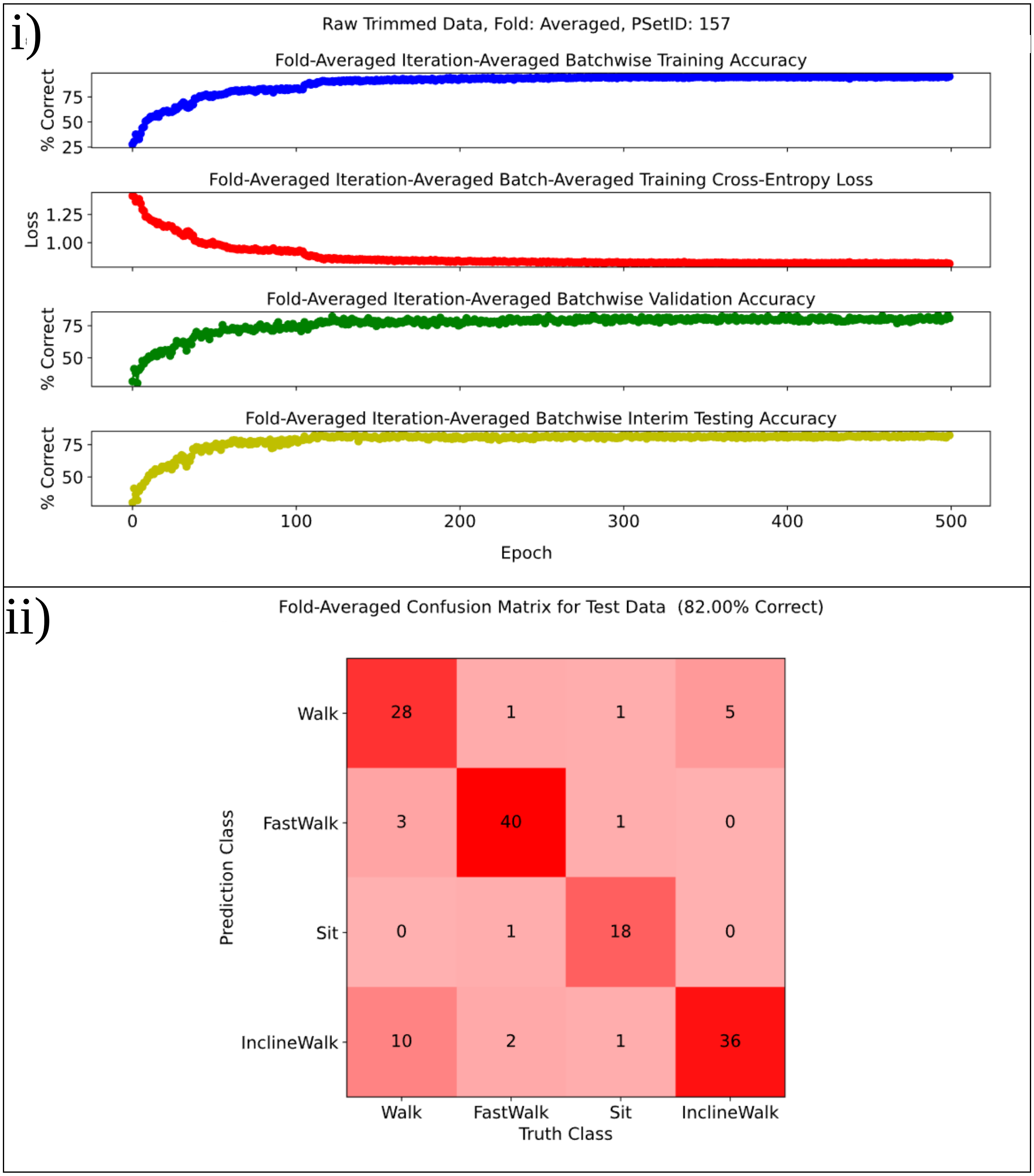


Figure 4-23: Model metrics by epoch (i) and testing time confusion matrix (ii) for P01, on best fold-averaged hyper parameter set over 371 sets. Note that the interim testing accuracy metric in i) is meant to illustrate how testing time accuracy fares over training, but was not used to influence the model.

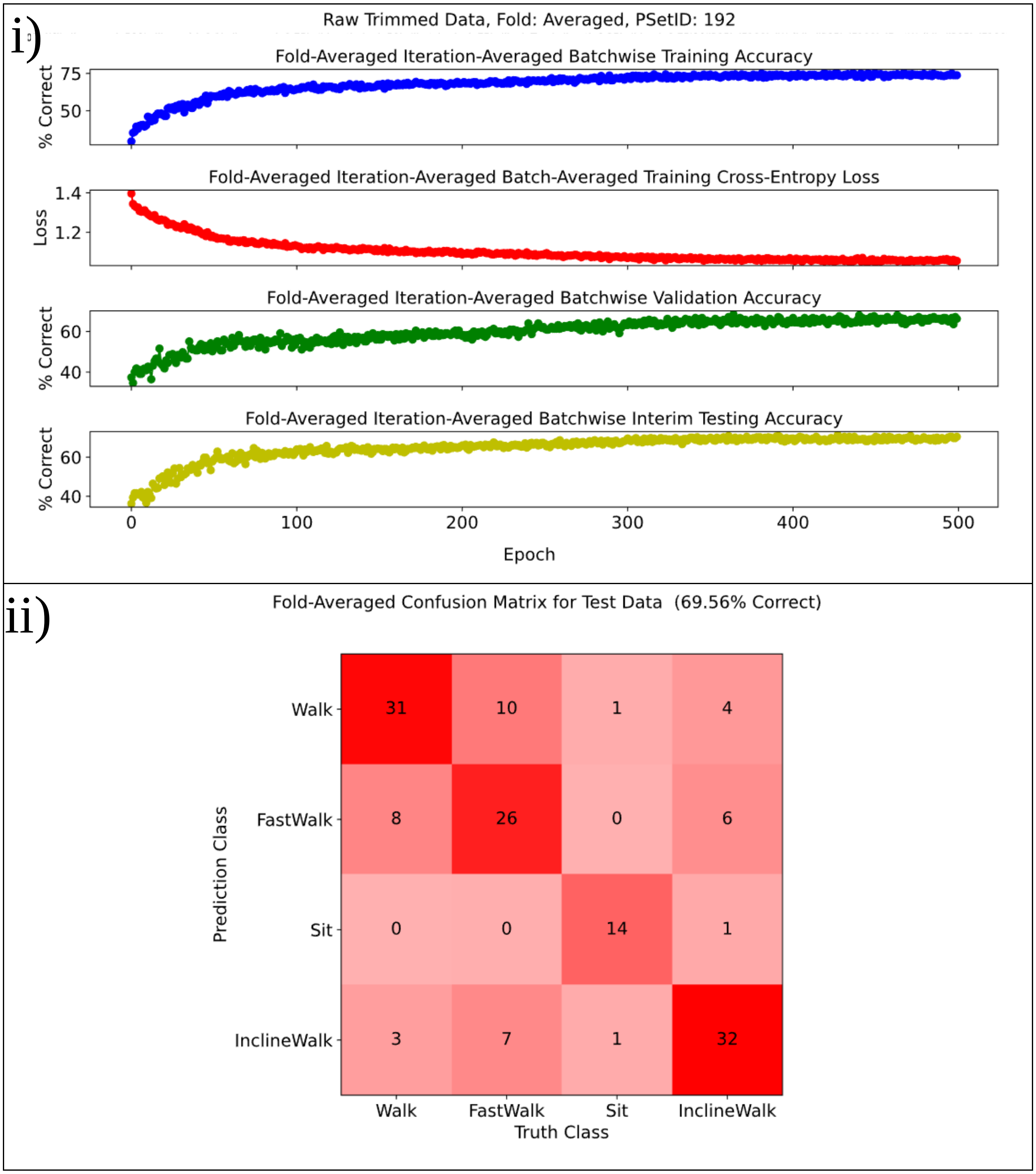
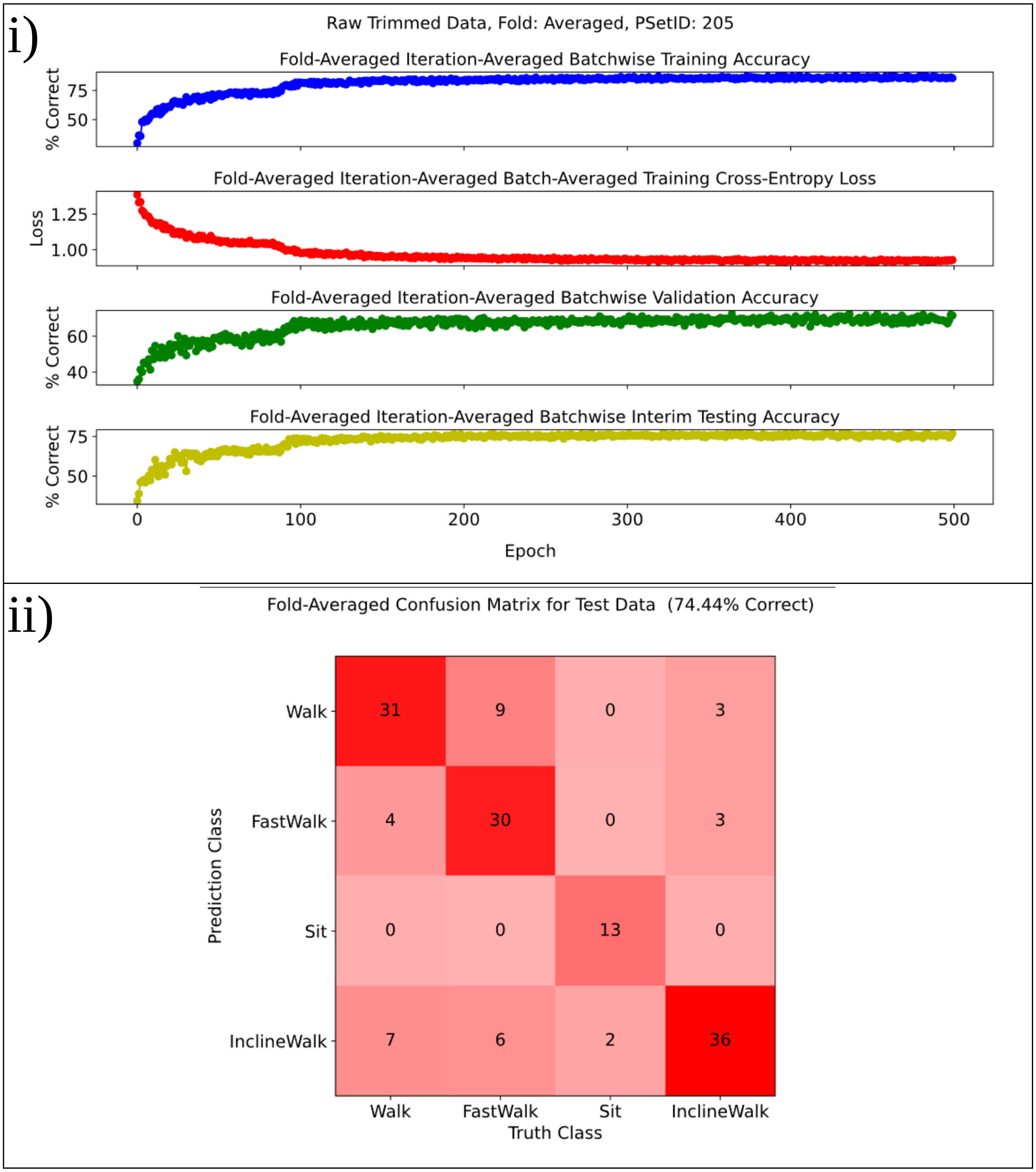


Figure 4-24: Model metrics by epoch (i) and testing time confusion matrix (ii) for P05, on best fold-averaged hyper parameter set over 371 sets. Note that the interim testing accuracy metric in i) is meant to illustrate how testing time accuracy fares over training, but was not used to influence the model.



Prediction \ Truth	Walk	FastWalk	Sit	InclineWalk
Walk	31	9	0	3
FastWalk	4	30	0	3
Sit	0	0	13	0
InclineWalk	7	6	2	36

Figure 4-25: Model metrics by epoch (i) and testing time confusion matrix (ii) for P08, on best fold-averaged hyper parameter set over 371 sets. Note that the interim testing accuracy metric in i) is meant to illustrate how testing time accuracy fares over training, but was not used to influence the model.

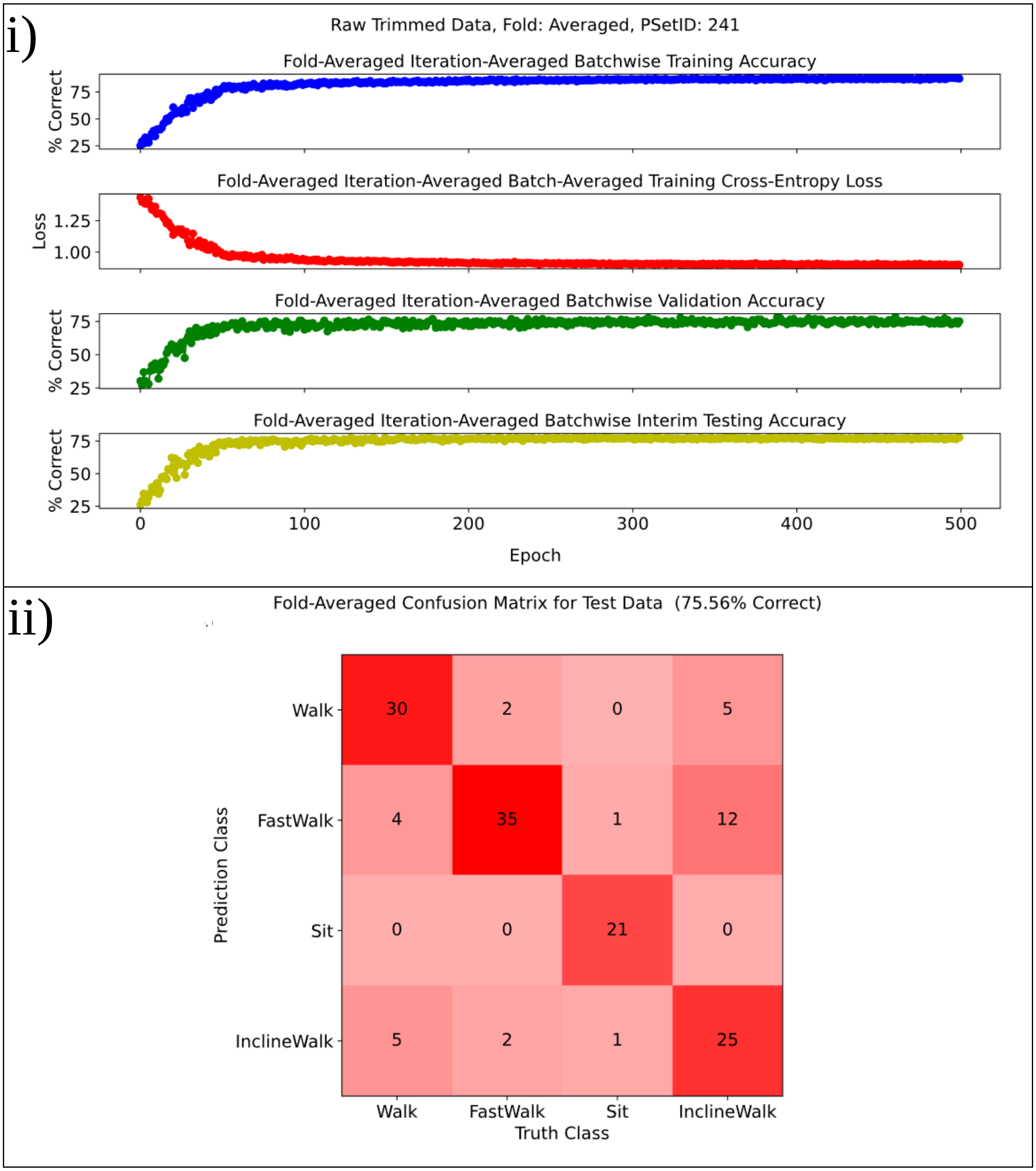


Figure 4-26: Model metrics by epoch (i) and testing time confusion matrix (ii) for P09, on best fold-averaged hyper parameter set over 371 sets. Note that the interim testing accuracy metric in i) is meant to illustrate how testing time accuracy fares over training, but was not used to influence the model.

4.5 *HYPERPARAMETERS ON REMAINING PARTICIPANTS*

4.5.1 UNRESTRICTED DATA

We took hyper parameter sets 146 and 192 and trained and tested the remaining participants on models with those hyper parameter configurations. The results are shown in Table 4-14 and Table 4-15, for both unrestricted sitting and unrestricted walking data (S100W100). We calculated overall accuracy, as well as recall and precision. Recall asks how many chunks of data belong to a class and are predicted as that class, and precision asks how many chunks predicted as a class belong to that class[208]. We find that participants P06 and P11 had the lowest accuracies when taking both parameter sets into account. We speculate that this is due to the greater occurrence of discontinuities in the time series data, relative to the other participants, making learning more difficult for the model on those data. More comprehensive pre-processing could help with this problem. Overall, hyper parameter set 192 was able to find a participant-averaged accuracy of 65.61% ± 8.85%, and hyper parameter set 146 was able to find a participant-averaged accuracy of 67.21% ± 9.06%.

We also find that the average accuracy across all the participants closely matches the precision and recall metrics for each class, meaning that on average, the model is about as accurate for each class as it is overall (see Table 4-14 and Table 4-15). Worth noting is that the sit class recall is relatively low, partially due to the inability for any correct classification in the case of some participants, such as P12 and P13. Further, because relatively fewer sit data occur, when the other classes misclassify as sit, it causes a higher penalty in recall for sit. However, a comparatively higher precision for the sit class relative to its recall means that whenever classes are predicted as sit, they are generally correct predictions. We average out the confusion matrices for each participant and produce an averaged confusion matrix over all the participants in Figure 4-27 for hyper parameter set 146 and Figure 4-28 for hyper parameter set 192.

Table 4-14: Fold-averaged metrics calculated across independent participants using unrestricted data, on hyper parameter set 146, on testing time data. FWalk means fast walk and IWalk means incline walk. Recall represents how many chunks that belong to a class were predicted that way, and precision represents how many chunks predicted to be of a class belong to that class. Where a 0/0 case occurs, NaN is used.

S100W100, Hyper Param Set 146, Testing Results		**Recall**				**Precision**			
PID	**Acc.**	**Walk**	**FWalk**	**Sit**	**IWalk**	**Walk**	**FWalk**	**Sit**	**IWalk**
P01	81.78%	82.80%	93.78%	54.81%	81.43%	72.16%	89.85%	97.50%	81.56%
P05	65.11%	60.48%	74.50%	49.02%	66.08%	68.22%	58.75%	89.58%	66.74%
P06	54.67%	41.28%	39.27%	60.78%	74.67%	50.85%	36.19%	88.93%	64.00%
P07	71.78%	76.43%	73.09%	33.33%	76.72%	64.23%	81.12%	84.62%	72.47%
P08	68.22%	62.40%	63.28%	55.12%	81.45%	66.08%	64.83%	100.00%	67.39%
P09	63.78%	67.00%	53.17%	57.95%	75.45%	68.98%	72.08%	88.99%	58.68%
P10	57.78%	35.11%	53.22%	53.22%	90.04%	63.21%	88.07%	84.93%	52.61%
P11	50.22%	74.21%	7.58%	0.00%	81.53%	57.98%	16.67%	NaN	48.01%
P12	56.89%	79.71%	28.68%	0.00%	75.02%	54.28%	94.87%	NaN	56.17%
P13	81.56%	90.16%	87.01%	0.00%	87.42%	79.65%	92.03%	NaN	75.21%
P14	75.56%	80.72%	82.61%	64.71%	65.75%	83.87%	69.33%	84.44%	75.73%
P15	69.56%	49.40%	87.25%	62.89%	82.21%	69.68%	82.79%	84.76%	56.43%
P16	71.56%	89.24%	60.74%	23.53%	83.02%	66.13%	72.38%	85.71%	79.57%
P17	72.00%	63.70%	89.08%	27.08%	78.72%	70.48%	76.79%	92.86%	68.82%
P18	64.44%	58.47%	59.82%	20.00%	84.32%	70.17%	54.89%	60.00%	71.83%
P19	70.44%	42.57%	90.61%	62.75%	81.91%	67.18%	90.11%	93.30%	54.76%
AVG	67.21%	65.85%	65.23%	39.07%	79.11%	67.07%	71.30%	70.98%	65.62%
VAR	9.06%	17.29%	24.65%	24.19%	6.68%	8.30%	21.47%	36.26%	10.21%

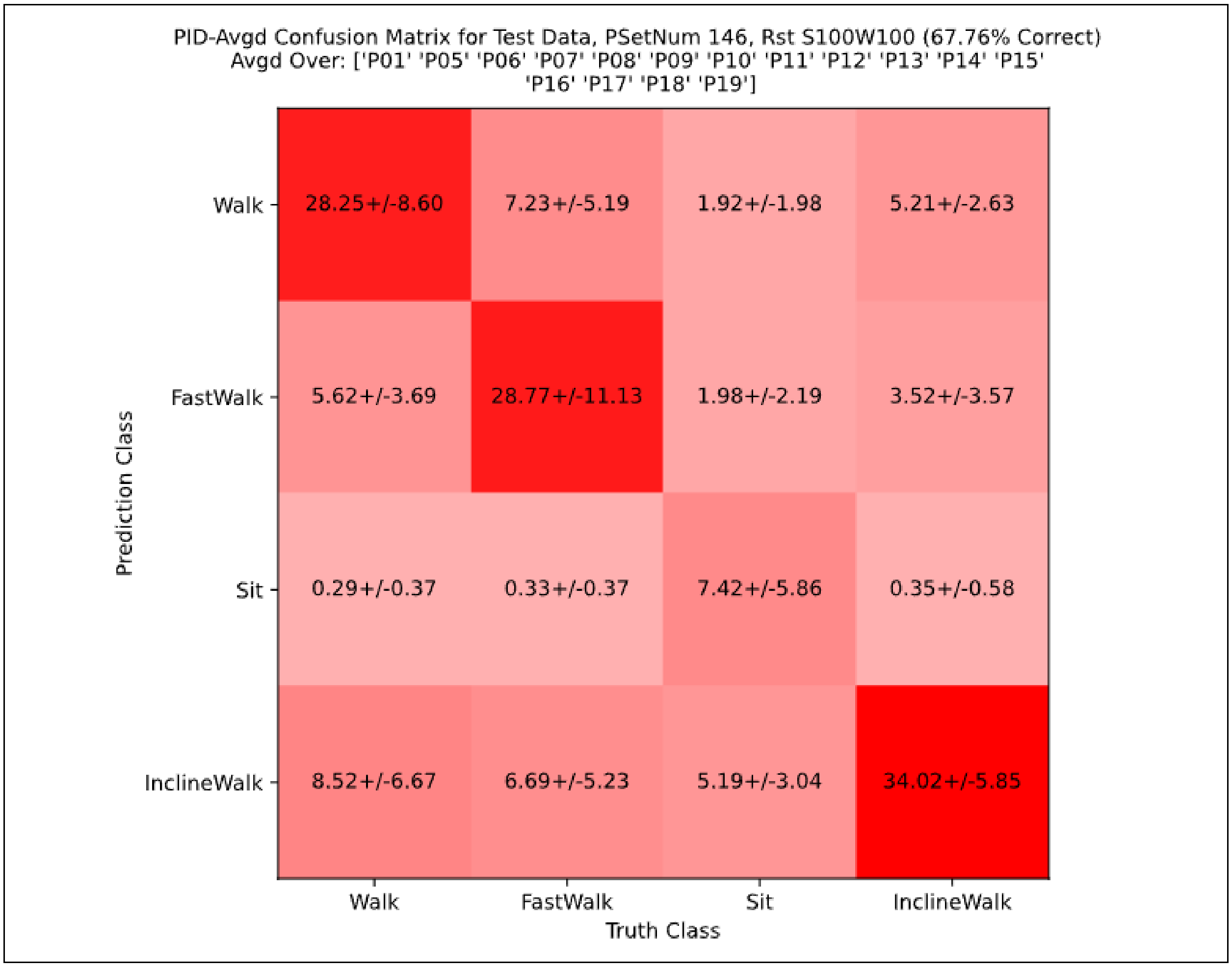


Figure 4-27: A confusion matrix generated by averaging out the confusion matrices generated for each independent participant's results on unrestricted data using hyper parameter set 146. This differs from the metrics in Table 4-14, which simply averages the computed statistics. The model is able to make predictions on the classes of the data with moderate success on average, producing a noticeable diagonal along the matrix.

.

Table 4-15: Fold-averaged metrics calculated across independent participants using unrestricted data on hyper parameter set 192, on testing time data. FWalk means fast walk and IWalk means incline walk. Recall represents how many chunks that belong to a class were predicted that way, and precision represents how many chunks predicted to be of a class belong to that class. Where a 0/0 case occurs, NaN is used.

S100W100, Hyper Param Set 192, Testing Results		**Recall**				**Precision**			
PID	**Acc.**	**Walk**	**FWalk**	**Sit**	**IWalk**	**Walk**	**FWalk**	**Sit**	**IWalk**
P01	68.22%	38.37%	92.08%	58.49%	80.40%	66.02%	84.25%	80.50%	57.45%
P05	60.89%	70.51%	38.97%	53.38%	74.68%	61.33%	55.09%	91.18%	62.61%
P06	48.44%	37.83%	30.22%	61.88%	61.71%	58.16%	37.17%	84.03%	43.13%
P07	60.44%	56.52%	49.43%	31.11%	83.79%	65.26%	77.71%	93.33%	53.77%
P08	64.67%	79.88%	26.24%	74.31%	82.54%	58.08%	80.43%	100.00%	63.93%
P09	78.44%	70.70%	85.84%	88.89%	72.66%	81.90%	72.75%	91.54%	76.38%
P10	75.56%	63.70%	84.78%	58.80%	86.98%	72.31%	86.73%	97.06%	70.33%
P11	60.89%	83.94%	0.00%	33.33%	89.64%	74.34%	0.00%	80.00%	54.80%
P12	65.78%	86.20%	56.44%	0.00%	68.15%	55.90%	100.00%	NaN	68.22%
P13	75.11%	93.15%	89.96%	0.00%	59.14%	75.30%	82.15%	NaN	74.27%
P14	81.78%	77.38%	82.24%	97.92%	79.57%	87.66%	74.53%	85.50%	85.14%
P15	63.78%	68.18%	27.93%	84.93%	73.70%	48.54%	64.58%	90.82%	66.35%
P16	62.67%	55.55%	54.19%	75.65%	74.29%	71.58%	51.77%	78.80%	65.32%
P17	68.22%	53.95%	88.90%	50.48%	68.69%	61.72%	68.93%	75.71%	71.43%
P18	53.33%	41.87%	63.81%	46.67%	56.17%	61.95%	49.99%	73.89%	72.13%
P19	61.56%	57.31%	81.67%	28.89%	58.97%	55.63%	90.71%	86.67%	44.69%
AVG	65.61%	64.69%	59.54%	52.79%	73.19%	65.98%	67.30%	75.56%	64.37%
VAR	8.85%	17.08%	28.61%	28.92%	10.40%	10.47%	24.50%	30.42%	11.35%

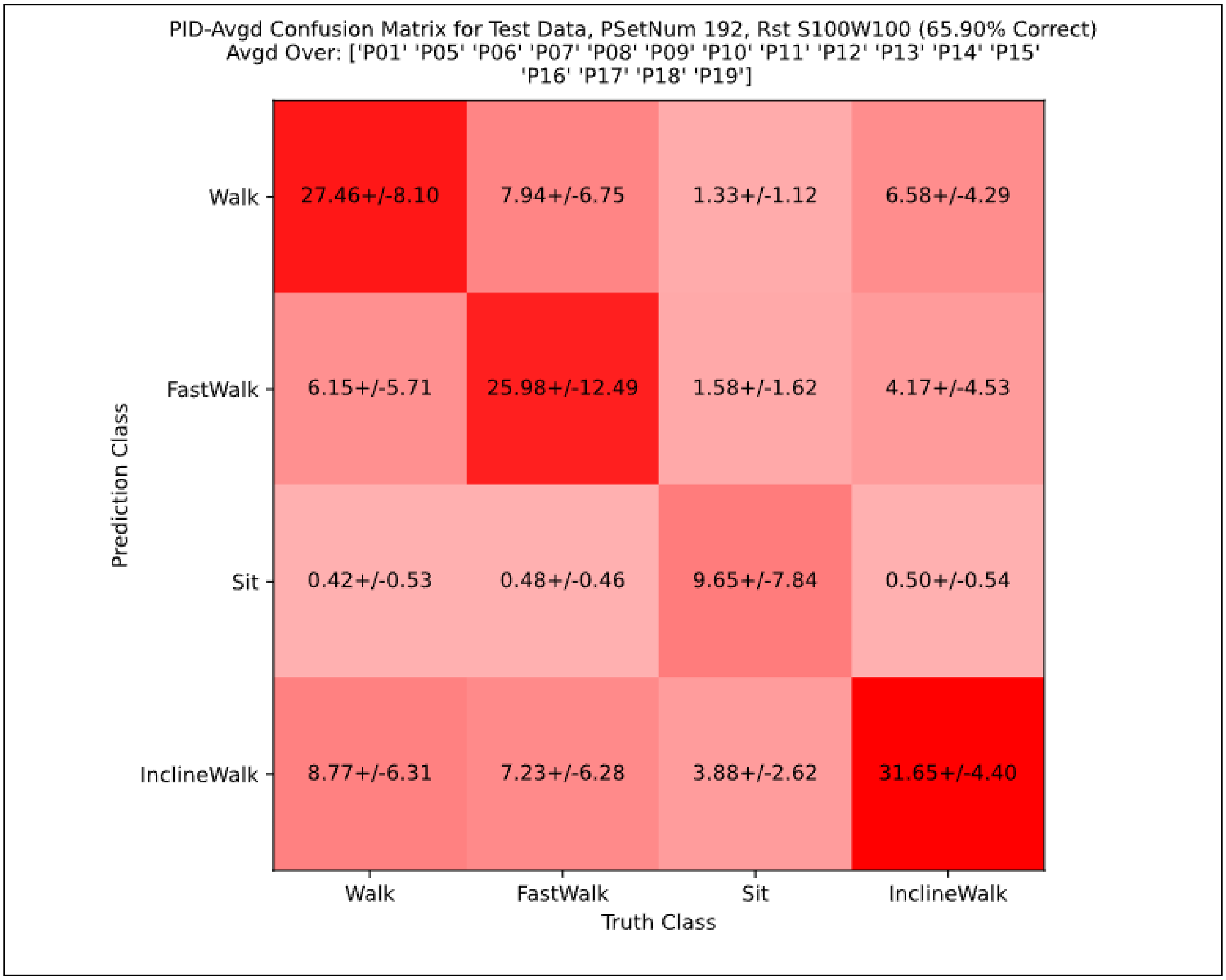


Figure 4-28: A confusion matrix generated by averaging out the confusion matrices generated for each independent participant's results on unrestricted data using hyper parameter set 192. This differs from the metrics in Table 4-15, which simply averages the computed statistics. The model is able to make predictions on the classes of the data with moderate success on average, producing a noticeable diagonal along the matrix.

4.5.2 RESTRICTED DATA

We also took the same hyper parameter sets, which were calculated on unrestricted data, and used them to train and test on the different restrictions of data independently, to see how performance varied as restriction level changed. The results are shown in Table 4-16. Note that results for P11 in the lightly restricted level could not be produced, due to damaged normal walking data files. Although there are variations in the trends across restriction level, for instance, P07 performing better with severe restriction data than moderate restriction data whereas P17 performs better with moderate restriction than unrestricted data, on average, we find that the more restricted the data are, the lower the overall accuracy is for the same hyper parameter set. This means that either different hyper parameter set searches could be needed for differing levels of restriction, or that a more complex model is needed to learn underlying pattern that can help distinguish activities better at different restriction levels.

Table 4-16: Comparison of fold-averaged overall accuracies across different restriction levels, on both hyper parameter sets. The accuracies are provided per participant, as a percentage.

Rst. Level	**S100W100 (Unrestricted)**		**S70W40 (Light)**		**S60W30 (Moderate)**		**S50W20 (Severe)**	
Hyper Param Set	**146**	**192**	**146**	**192**	**146**	**192**	**146**	**192**
PID	**Overall Accuracy**							
P01	81.78%	68.22%	49.33%	57.78%	66.22%	52.89%	42.89%	58.89%
P05	65.11%	60.89%	67.11%	67.78%	44.22%	43.11%	60.67%	57.78%
P06	54.67%	48.44%	46.22%	36.00%	46.22%	49.11%	52.22%	51.56%
P07	71.78%	60.44%	63.56%	57.11%	51.33%	57.78%	56.89%	59.78%
P08	68.22%	64.67%	58.44%	54.67%	65.56%	66.44%	70.89%	59.33%
P09	63.78%	78.44%	68.00%	74.67%	54.22%	56.22%	48.22%	66.00%
P10	57.78%	75.56%	68.44%	77.78%	55.11%	57.33%	66.22%	67.11%
P11	50.22%	60.89%	NaN	NaN	68.22%	61.11%	45.78%	58.22%
P12	56.89%	65.78%	54.44%	66.22%	69.56%	70.67%	63.11%	42.00%
P13	81.56%	75.11%	74.89%	65.78%	65.56%	64.67%	73.78%	58.67%
P14	75.56%	81.78%	55.33%	57.78%	40.67%	48.89%	53.56%	45.11%
P15	69.56%	63.78%	70.44%	72.89%	68.00%	64.67%	49.56%	72.89%
P16	71.56%	62.67%	57.33%	57.78%	56.22%	44.44%	44.22%	53.11%
P17	72.00%	68.22%	59.56%	65.56%	74.89%	72.22%	49.78%	49.33%
P18	64.44%	53.33%	51.33%	58.67%	52.89%	47.56%	45.33%	44.00%
P19	70.44%	61.56%	70.44%	63.11%	73.33%	73.56%	58.89%	66.44%
AVG	67.21%	65.61%	57.18%	58.35%	59.51%	58.17%	55.12%	56.89%
VAR	9.06%	8.85%	17.42%	18.36%	10.74%	9.93%	9.67%	8.89%

We also compare the participant-averaged fold-averaged recall and precision values over the different restriction levels, for hyper parameter set 146 in Table 4-17 and set 192 in Table 4-18. No clear trend emerges here as we go from lighter to more severe restrictions. In general, we do see that compared to completely unrestricted metrics, the severely restricted metrics are worse, except for sitting recall, which improves slightly. Much like the degradation in accuracy, these metrics also indicate that the performance of hyper parameter sets on unrestricted data does not extend to differently restricted data.

Table 4-17: Participant-averaged fold-averaged recall and precision per class, with variance, at differing levels of restriction. Data are provided on hyper parameter set 146. FWalk means fast walk, and IWalk means incline walk.

Fold-Averaged Testing Time Results, Mean and Var. over Participants		**Recall**				**Precision**			
Hyper Param Set 146		**Walk**	**FWalk**	**Sit**	**IWalk**	**Walk**	**FWalk**	**Sit**	**IWalk**
S100W100	**AVG**	65.85%	65.23%	39.07%	79.11%	67.07%	71.30%	70.98%	65.62%
	VAR	17.29%	24.65%	24.19%	6.68%	8.30%	21.47%	36.26%	10.21%
S70W40	**AVG**	66.80%	60.38%	37.70%	59.61%	58.93%	64.98%	62.96%	57.32%
	VAR	18.60%	21.56%	32.07%	24.35%	12.57%	20.54%	35.99%	17.39%
S60W30	**AVG**	57.84%	63.19%	46.47%	60.48%	56.52%	61.79%	64.31%	58.64%
	VAR	20.87%	19.87%	28.04%	21.86%	17.78%	13.77%	33.08%	9.68%
S50W20	**AVG**	58.47%	42.95%	43.66%	65.91%	55.67%	56.55%	58.26%	51.59%
	VAR	18.20%	22.75%	34.53%	13.60%	10.77%	18.90%	35.78%	12.44%

Table 4-18: Participant-averaged fold-averaged recall and precision per class, with variance, at differing levels of restriction. Data are provided on hyper parameter set 192. FWalk means fast walk, and IWalk means incline walk.

Fold-Averaged Testing Time Results, Mean and Var. over Participants		**Recall**				**Precision**			
Hyper Param Set 192		**Walk**	**FWalk**	**Sit**	**IWalk**	**Walk**	**FWalk**	**Sit**	**IWalk**
S100W100	**AVG**	64.69%	59.54%	52.79%	73.19%	65.98%	67.30%	75.56%	64.37%
	VAR	17.08%	28.61%	28.92%	10.40%	10.47%	24.50%	30.42%	11.35%
S70W40	**AVG**	62.03%	61.02%	42.09%	69.11%	62.61%	63.38%	78.32%	61.62%
	VAR	12.30%	23.81%	28.05%	14.91%	11.97%	19.70%	32.44%	13.21%
S60W30	**AVG**	57.04%	61.42%	45.40%	59.96%	58.19%	62.48%	68.59%	57.61%
	VAR	14.69%	18.97%	26.13%	24.69%	11.99%	14.67%	28.59%	9.88%
S50W20	**AVG**	58.15%	60.41%	42.04%	54.88%	55.22%	61.55%	57.28%	52.73%
	VAR	14.11%	17.75%	36.60%	15.17%	6.60%	12.11%	40.89%	15.07%

The model results for the different restriction levels on hyper parameter sets 146 and 192 are visualised as participant-averaged confusion matrices in *Figure* 4-29 and *Figure* 4-30, for ease of interpretation. The accuracies found from the matrices will vary slightly from the metrics calculated in Table 4-16 to Table 4-18, as the matrices average out the raw counts in each category across participants, whereas the metrics in the tables are averaged out across each participant's calculated metrics.

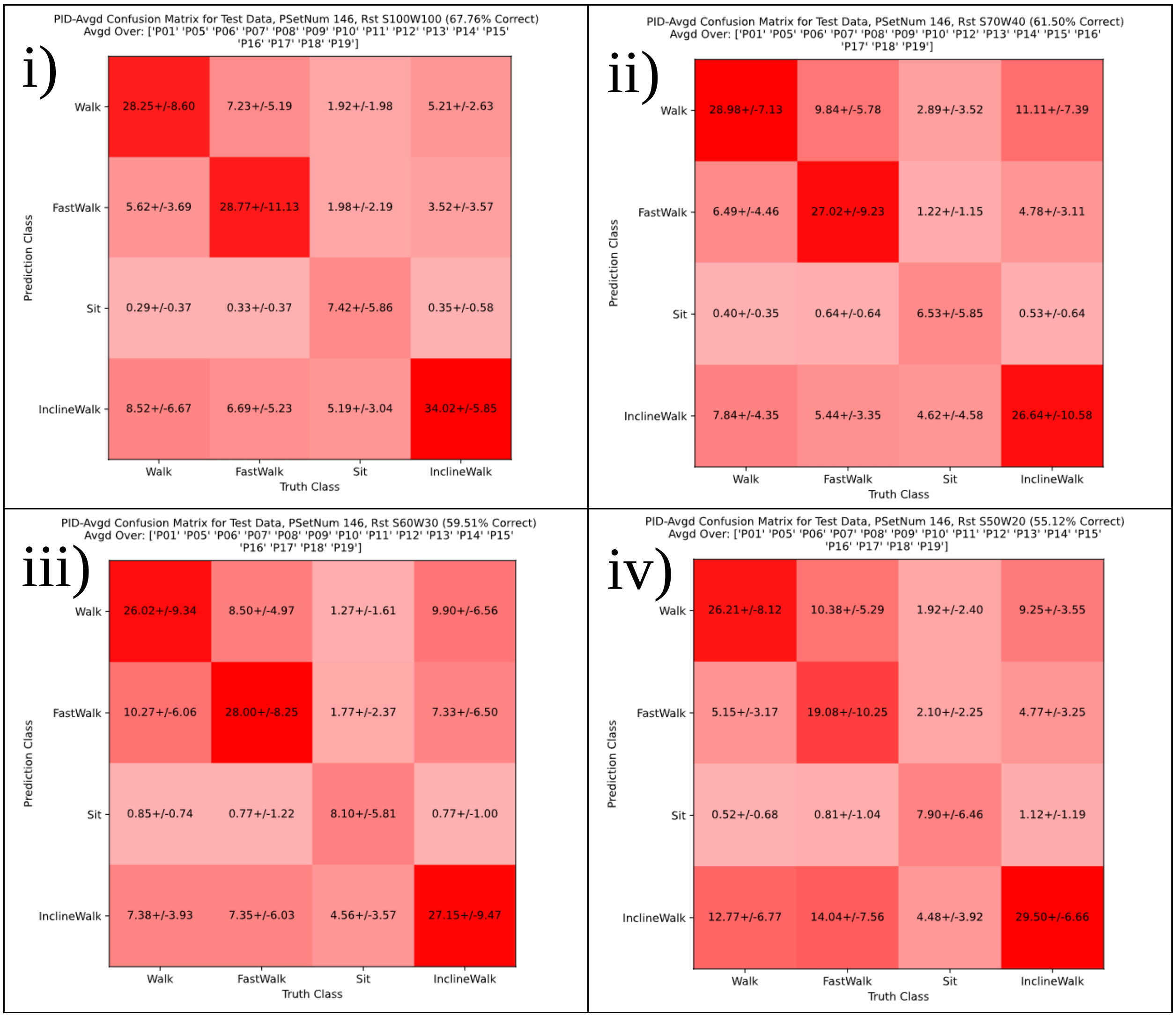


Figure 4-29: Participant-averaged confusion matrices for i) unrestricted, ii) light, iii) moderate, and iv) severe gait restrictions. These results come averaging the testing time results of each independent participant, on models produced using hyper parameter set 146. Notice that as restriction severity increases, overall accuracy decreases, and in general, recall and precision of each class decrease as well.

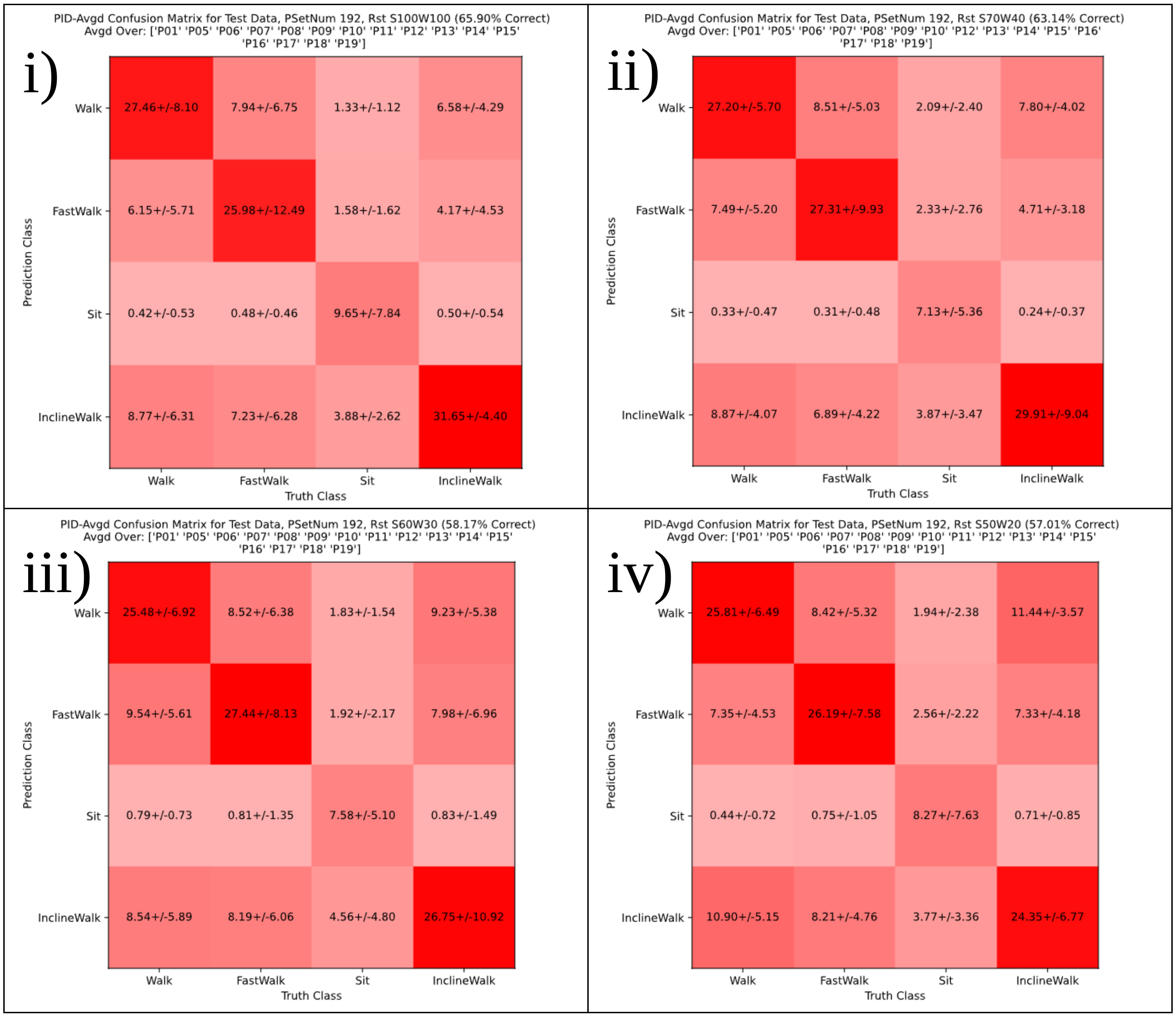


Figure 4-30: Participant-averaged confusion matrices for i) unrestricted, ii) light, iii) moderate, and iv) severe gait restrictions. These results come averaging the testing time results of each independent participant, on models produced using hyper parameter set 192. Notice that as restriction severity increases, overall accuracy decreases, and in general, recall and precision of each class decrease as well.

4.6 CONCLUSION

We performed data collection on 16 participants, producing three trials of four restrictions of four activities for each participant. These data were organised, labelled, and parsed into a form which would be useful for data analysis. Of interest to us were two questions: 1) could we find factors which differentiated the different restriction levels of the data, 2) could we use a machine learning model to classify the different activities being performed.

To answer 1), we found that the standard deviation of the derivative of the x-dimension thigh acceleration time series for the negotiating sitting portion of a sit-to-stand activity can separate unrestricted data from almost any restriction of data, just barely failing to do so for unrestricted and lightly restricted sit-to-stand data. However, we were unable to find a differentiating factor which could separate restricted data from each other. We also found that the maximum angular velocity at the swing phase of the z-dimension shank gyroscope time series for all walking-type activities could separate unrestricted data from all restricted data. It was also able to separate lightly restricted data from severely restricted data for all walking-type activities. We were unable to find a measure that could separate all restricted data from each other.

To answer 2), we used a convolutional neural network trained on independent participants, using their unrestricted data, and found that the model was capable of classifying activities with an accuracy of ~65% or higher. The model performance varies by participant, and most of the inaccuracies come from classifying different types of walking. We compared results of the best found hyper parameter sets on unrestricted and restricted data, and found a general trend of worse overall accuracy as restriction level is increased, for the same parameter set.

Although much data analysis work has been done, there are still various avenues that could be pursued, or alternate techniques that could have been implemented. Suggestions for further work are offered in the following chapter.

5 CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSION

The monitoring and measurement of human gait is a meaningful way to assess the presence of gait-afflicting conditions. Although multiple conditions can be the cause of gait alterations, of interest to our work was arthritis, specifically knee osteoarthritis.

Upon reviewing the literature, we found a gap existed in the analysis of artificially restricted gait over the same participants using simple convolutional networks. Further we sought to find features that could be used to differentiate differently restricted gaits of the same participants. Lastly, we aimed to collect these data using a custom-built data collection system, to address the higher barrier of entry to existing systems.

We built this data collection system by programming a microcontroller to read sensor information from two IMUs, whose data would be stored to an onboard microSD card, and timestamped by an onboard RTC. The IMUs could be positioned wherever analysis was desired; for our purposes, the thigh and shank were suitable. The core of the system was printed onto a circuit board and enclosed in a 3D-printed case to allow it to be used comfortably. We were able to collect data at a rate of approximately 80Hz, and were able to validate the accelerometer results against a known gold-standard system using one participant, showing small RMS values for time and frequency-spectrum comparisons, and well as correlation values close to 1 in many cases.

After validation, we were able to conduct a data collection on a population of 16 participants (9M, 6F, Ages: 26.25 ± 2.86). Each participant performed four different types of activities (walking, fast walking, incline walking, sit-to-stand) at four different restriction levels each (unrestricted, light, moderate, severe) in a laboratory environment. Data were recorded, organised, and labelled, so that they could be analysed. The analysis conducted was able to find participant-averaged differences in the collected data across restriction levels for walking and sit-to-stand activities, considering the particular mechanical details of the movements.

For walking activities, in the z-dimension of shank gyroscope, toe-off and swing gait regions showed differences across restrictions, with higher restriction levels having lower angular velocities. Statistical significance was found between swing regions of unrestricted walking activities and all other restriction levels, and for toe-off regions of unrestricted walking activities and moderately and severely restricted levels.

For sit-to-stand activities, in the x-dimension of thigh acceleration, lower average jerk for negotiating sitting and negotiating standing edges were found, as well as larger jerk standard deviations for negotiating sitting and negotiating standing edges. Statistically significant difference was found between jerk standard deviation of the negotiating sitting edge of unrestricted data and almost all restricted levels, with unrestricted and lightly restricted groups barely failing to be significantly different.

Further analysis aimed to use the unrestricted acceleration data of both thigh and shank IMUs to predict the activities that were being performed, in the context of a supervised machine learning problem. We used a convolutional neural network as our predictive model and conducted a parameter search over more than three hundred parameters on four of the participants. Using the best common parameter sets from those searches, the remaining participants were independently classified upon, leading to participant-averaged accuracies of 65.61% ± 8.85% on one set, and 67.21% ± 9.06% on the other. The restricted data were used as input to these models to see how well these parameter sets would perform on restricted data, which resulted in lower accuracy compared to unrestricted data, and worsening accuracy as restriction level is increased.

We found that for the purposes of activity classification on our data, although some participants performed better than others, on average, our machine learning pipeline struggled to perform with high accuracy on the task of the activity classification.

5.2 FUTURE WORK

Much work can be done at the conclusion of this thesis, both in the realms of the embedded system and the DCU, as well as in the machine learning side for activity classification. Suggestions for directions of improvement are provided in the following sections.

5.2.1 EMBEDDED WORK

A simple quality of life improvement to be made to the enclosure of the DCU is to add space for the USB port of the microcontroller; oftentimes software needed to be updated which necessitated the complete removal of the board. This was time consuming and wore down the plastic of the enclosure at the screw posts, thus an easy in-place method to interface with the device would be useful. There is much work still to be done with regard to the DCU algorithm. The SD module used for reading and writing the data to a microSD card was able to communicate with the main microcontroller with different protocols. The SPI protocol was used due to ease of implementation, but the module can also use the SDIO (SD Input/Output) protocol. It could be that using the SDIO protocol could lead to faster read/write times, and eliminate the need for multi-threaded approach altogether. This would simplify the algorithm for the DCU, making debugging easier. Further, the IMUs themselves have some processing and buffering facilities onboard, and could be investigated to see if some of the microcontroller processing could be offloaded to the sensors. However, if the multi-threaded approach cannot be avoided, serious effort will have to be taken using specialized debugging equipment in order to address some reliability concerns faced by the device during data collection. Another avenue for increasing reliability is to investigate wireless usage of the IMUs, by transmitting the I2C or SPI signal by Bluetooth; this may address reliability issues experienced by long lengths of cable. Alternatively, the I2C protocol can be partially augmented by including a twisted pair adapter to cover the longer distance more reliably.

5.2.2 MACHINE LEARNING WORK

One of the most straightforward additions to be made on the machine learning side of the work is to marry the activity classification with a restriction level regression model. As we have already found features in the data that can be used to separate restriction levels out, adding that functionality to the machine learning model would widen its scope and allow for activity classification with a restriction level prediction. This would bring us closer to the overall direction of the project – to gauge the quality of gait of individuals.

Improvements to the machine learning model can be made in a variety of ways. One such way would be to modify the model to be able to use both acceleration and gyroscope data as input, which may help model accuracy. Similarly, pre-processing the data more thoroughly by filtering and interpolating them may help as well. In particular, some of the data have gaps in them due to DCU issues, which may be solved by a more complex interpolation procedure, possibly using averaged gait cycles to fill gaps depending on where in the cycle the gaps occur. Another direction would be to not use the entire time series, but to break the data into exact gait cycles, and pass those into the model.

Other work could include different experiments such as inter-participant training and testing of data, as opposed to the intra-participant work that was done in this thesis. In the literature, it was found that many authors would train and validate the model on multiple participants, and then test on one participant which was left out of training. Future authors could also train the model across restriction levels to see how the model performs, instead of training on only unrestricted data. Lastly, results across different model types could be compared, or the convolutional model could be tested across a variety of model depths and architectures. Different input features could be provided to the model to aid its accuracy, such as the entirety of the frequency spectrum, instead of the peaks we found most interesting, or even wavelet transformed data.

BIBLIOGRAPHY

1. Emmett McMahon, M. S. Introduction to Human Gait. https://pressbooks.calstate.edu/humanneuromechanics/chapter/introduction-to-human-gait/.
2. *Technologies and Techniques in Gait Analysis*. (The Institution of Engineering and Technology).
3. di Biase, L., Raiano, L., Caminiti, M. L., Pecoraro, P. M. & Di Lazzaro, V. Parkinson's Disease Wearable Gait Analysis: Kinematic and Dynamic Markers for Diagnosis. *Sensors* **22**, (2022).
4. Pollet, J. & Buraschi, R. Gait Alterations in Knee Osteoarthritis: A Narrative Review on Gait Analysis Studies. *Top. Geriatr. Rehabil.* **37**, 239–243 (2021).
5. Subasinghe Arachchige, R. S. S. *et al.* Stage-specific gait deviations in individuals with hip osteoarthritis. *Gait Posture* **120**, 226–233 (2025).
6. Okada, K. *et al.* Categorizing knee hyperextension patterns in hemiparetic gait and examining associated impairments in patients with chronic stroke. *Gait Posture* **113**, 18–25 (2024).
7. Tsurumiya, K. *et al.* Quantitative Evaluation Related to Disease Progression in Knee Osteoarthritis Patients During Gait. *Adv. Biomed. Eng.* **10**, 51–57 (2021).
8. Suzuki, Y. *et al.* Assessing knee joint biomechanics and trunk posture according to medial osteoarthritis severity. *Sci. Rep.* **13**, (2023).
9. McCain, E. M. *et al.* Isolating the energetic and mechanical consequences of imposed reductions in ankle and knee flexion during gait. *J. NeuroEngineering Rehabil.* **18**, 21 (2021).
10. Halim, H. N. A., Azaman, A. & Zulkapri, I. Cluster Analysis of Biomechanical Gait Data and Pain Score as a Potential Classification of Severity in Knee Osteoarthritis. *J. Hum. Centered Technol.* **1**, 33–43 (2022).
11. Limcharoen, P., Khamsemanan, N. & Nattee, C. Gait Recognition using Double-Window and CNN Classification on Freestyle Walks. in *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)* 1231–1237 (2018). doi:10.1109/SCIS-ISIS.2018.00194.
12. Soltaninejad, S., Cheng, I. & Basu, A. Kin-FOG: Automatic Simulated Freezing of Gait (FOG) Assessment System for Parkinson's Disease. *Sensors* **19**, (2019).
13. Djamaa, B., Bessa, M. M., Diaf, B., Rouigueb, A. & Yachir, A. BoostSole: Design and Realization of a Smart Insole for Automatic Human Gait Classification. in 35–43 (2020). doi:10.15439/2020F92.
14. Sarmah, A., Boruah, L., Ito, S. & Kanagaraj, S. Integrative approach to pedobarography and pelvis-trunk motion for knee osteoarthritis detection and exploration of non-radiographic rehabilitation monitoring. *Front. Bioeng. Biotechnol.* **12**, (2024).

15. Wong, C. C. & Han, Y. C. Gait Recognition System: A Review. in *2023 International Conference on Digital Applications, Transformation & Economy (ICDATE)* 1–5 (2023). doi:10.1109/ICDATE58146.2023.10248815.
16. IMU Step - IMeasureU. https://imeasureu.com/imu-step/.
17. MTw Awinda | Movella.com. https://www.movella.com/products/wearables/xsens-mtw-awinda.
18. Bhongade, A., Gupta, R., Bhatia, M., Prathosh, A. P. & Gandhi, T. K. Classification of Gait Phases Using a Shank-Mounted Single IMU Sensor. *IEEE Sens. J.* **25**, 14183–14195 (2025).
19. Lu, Y., Zhu, J., Chen, W. & Ma, X. An IMU-Based Real-Time Gait Detection Method for Intelligent Control of Knee Assistive Devices. *IEEE Trans. Instrum. Meas.* **72**, 1–9 (2023).
20. Han, Y. C., Wong, K. I. & Murray, I. Gait Phase Detection for Normal and Abnormal Gaits Using IMU. *IEEE Sens. J.* **19**, 3439–3448 (2019).
21. Slimi, H., Balti, A., Sayadi, M. & Khelifa, M. M. B. Augmented Gait Classification: Integrating YOLO, CNN–SNN Hybridization, and GAN Synthesis for Knee Osteoarthritis and Parkinson's Disease. *Signals* **6**, (2025).
22. Nishizawa, K., Nagura, T. & Ogihara, N. Assessment of knee joint motion during gait using IMU-based helical axis analysis for early detection of knee osteoarthritis. *Biomed. Signal Process. Control* **112**, (2026).
23. Mao, Y. *et al.* A Hybrid Human Activity Recognition Method Using an MLP Neural Network and Euler Angle Extraction Based on IMU Sensors. *Appl. Sci.* **13**, (2023).
24. Nasrabadi, A. M. *et al.* A new scheme for the development of IMU-based activity recognition systems for telerehabilitation. *Med. Eng. Phys.* **108**, 103876 (2022).
25. Marinou, G., Kourouma, I. & Mombaur, K. Development and Validation of a Modular Sensor-Based System for Gait Analysis and Control in Lower-Limb Exoskeletons. *Sensors* **25**, (2025).
26. Manupibul, U. *et al.* Integration of force and IMU sensors for developing low-cost portable gait measurement system in lower extremities. *Sci. Rep.* **13**, (2023).
27. Mohammadi Moghadam, S., Ortega Auriol, P., Yeung, T. & Choisne, J. 3D gait analysis in children using wearable sensors: feasibility of predicting joint kinematics and kinetics with personalized machine learning models and inertial measurement units. *Front. Bioeng. Biotechnol.* **12**, 1372669 (2024).
28. Truong, M. T. N., Ali, A. E. A., Owaki, D. & Hayashibe, M. EMG-Based Estimation of Lower Limb Joint Angles and Moments Using Long Short-Term Memory Network. *Sensors* **23**, 3331 (2023).
29. What Is Arthritis? Types, Symptoms, Causes & Treatment Options. https://arthritis.ca/about-arthritis/what-is-arthritis.

30. Public Health Agency of Canada. Osteoarthritis in Canada. https://www.canada.ca/en/public-health/services/publications/diseases-conditions/osteoarthritis.html (2020).
31. Long, H. *et al.* Prevalence Trends of Site-Specific Osteoarthritis From 1990 to 2019: Findings From the Global Burden of Disease Study 2019. *Arthritis Rheumatol.* **74**, 1172–1183 (2022).
32. Webber, S. C., Barclay, R., Ripat, J., Nowicki, S. & Tate, R. Factors associated with social participation and community ambulation in people with osteoarthritis: Findings from the Canadian Longitudinal Study on Aging. *Int. J. Rheum. Dis.* **26**, 360–369 (2023).
33. Bentz, J. A., Horton, J. & MacDougall, D. Exercise-Based Interventions for the Presurgical Management of Knee Osteoarthritis: A Rapid Qualitative Review. *Can. J. Health Technol.* **3**, (2023).
34. Summary of Special Report: The Burden of Osteoarthritis in Canada. https://arthritis.ca/wp-content/uploads/2025/09/OAReportSummary_EN.pdf (2021).
35. Roos, E. M. & Arden, N. K. Strategies for the prevention of knee osteoarthritis. *Nat. Rev. Rheumatol.* **12**, 92–102 (2016).
36. Sharma, L. Osteoarthritis of the Knee. *N. Engl. J. Med.* **384**, 51–59 (2021).
37. Yang, G. *et al.* Burden of Knee Osteoarthritis in 204 Countries and Territories, 1990–2019: Results From the Global Burden of Disease Study 2019. *Arthritis Care Res.* **75**, 2489–2500 (2023).
38. Osteoarthritis Symptoms and Diagnosis. https://arthritis.ca/about-arthritis/arthritis-types-(a-z)/types/osteoarthritis/osteoarthritis-symptoms-and-diagnosis.
39. O'Connell, M., Farrokhi, S. & Fitzgerald, G. K. The role of knee joint moments and knee impairments on self-reported knee pain during gait in patients with knee osteoarthritis. *Clin. Biomech.* **31**, 40–46 (2016).
40. Mlinac, M. E. & Feng, M. C. Assessment of Activities of Daily Living, Self-Care, and Independence. *Arch. Clin. Neuropsychol.* **31**, 506–516 (2016).
41. Edemekong, P. F., Bomgaars, D. L., Sukumaran, S. & Schoo, C. Activities of Daily Living. in *StatPearls* (StatPearls Publishing, Treasure Island (FL), 2024).
42. BoH. *English: Walk Cycle*. (2019).
43. What Is Bradykinesia? *Cleveland Clinic* https://my.clevelandclinic.org/health/symptoms/bradykinesia.
44. Gait Abnormalities. *Stanford Medicine 25* https://med.stanford.edu/stanfordmedicine25/the25/gait.html.
45. Campbell, R., Andrew, K., L'Ange, C. & Concannon, J. Preliminary validation study of novel fabric EMG device in non-affected adults – Can it compare to the industry gold-standard, Vicon? *Gait Posture* **97**, S382–S383 (2022).

46. Benjaminse, A., Bolt, R., Gokeler, A. & Otten, B. A VALIDITY STUDY COMPARING XSENS WITH VICON. *ISBS Proc. Arch.* **38**, 752 (2020).
47. Ishida, T. & Samukawa, M. Validity and Reliability of a Wearable Goniometer Sensor Controlled by a Mobile Application for Measuring Knee Flexion/Extension Angle during the Gait Cycle. *Sensors* **23**, 3266 (2023).
48. Tak, I., Wiertz, W.-P., Barendrecht, M. & Langhout, R. Validity of a New 3-D Motion Analysis Tool for the Assessment of Knee, Hip and Spine Joint Angles during the Single Leg Squat. *Sensors* **20**, 4539 (2020).
49. Riglet, L. *et al.* The Use of Embedded IMU Insoles to Assess Gait Parameters: A Validation and Test-Retest Reliability Study. *Sensors* **23**, 8155 (2023).
50. He, Y. *et al.* Accuracy validation of a wearable IMU-based gait analysis in healthy female. *BMC Sports Sci. Med. Rehabil.* **16**, 2 (2024).
51. Wilhelm, N. *et al.* Development and Evaluation of a Cost-effective IMU System for Gait Analysis: Comparison with Vicon and VideoPose3D Algorithms. *Curr. Dir. Biomed. Eng.* **9**, 254–257 (2023).
52. Chandel, V., Singhal, S., Sharma, V., Ahmed, N. & Ghose, A. PI-Sole: A Low-Cost Solution for Gait Monitoring Using Off-The-Shelf Piezoelectric Sensors and IMU. in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* 3290–3296 (2019). doi:10.1109/EMBC.2019.8857877.
53. Setiawan, A. W. & Ananda, A. R. Development of Wireless Gait Recognition System using IMU Sensors. in *2019 International Symposium on Micro-NanoMechatronics and Human Science (MHS)* 1–6 (2019). doi:10.1109/MHS48134.2019.9249312.
54. Sivakumar, S., Agape Gopalai, A., Hann Lim, K. & Gouwanda, D. A review on gait kinematics acquisition sensors and its advancements in IoT and machine learning | Cognitive Sensing Technologies and Applications. *Books* https://digital-library.theiet.org/doi/10.1049/PBCE135E_ch9 (2024).
55. Zhou, H. & Hu, H. Human motion tracking for rehabilitation—A survey. *Biomed. Signal Process. Control* **3**, 1–18 (2008).
56. Marker Set Guidelines [HAS-Motion Software Documentation]. https://has-motion.com/wiki/doku.php?id=visual3d:documentation:modeling:marker_sets:marker_set_guidelines&s[]=marker&s[]=set&s[]=guidelines.
57. Nexus legacy PDFs - Nexus Reference Documentation - Vicon Help. https://help.vicon.com/space/NLD/12124170/Nexus+legacy+PDFs?attachment=%2Fdownload%2Fattachments%2F12124170%2FNexus1_8Guide.pdf&type=application%2Fpdf&filename=Nexus1_8Guide.pdf (2015).

58. Graham, J. E., Ostir, G. V., Fisher, S. R. & Ottenbacher, K. J. Assessing walking speed in clinical research: a systematic review. *J. Eval. Clin. Pract.* **14**, 552–562 (2008).
59. Shema-Shiratzky, S. *et al.* A wearable sensor identifies alterations in community ambulation in multiple sclerosis: contributors to real-world gait quality and physical activity. *J. Neurol.* **267**, 1912–1921 (2020).
60. Taylor, D., Stretton, C. M., Mudge, S. & Garrett, N. Does clinic-measured gait speed differ from gait speed measured in the community in people with stroke? *Clin. Rehabil.* **20**, 438–444 (2006).
61. Donovan, K., Lord, S. E., McNaughton, H. K. & Weatherall, M. Mobility beyond the clinic: the effect of environment on gait and its measurement in community-ambulant stroke survivors. *Clin. Rehabil.* **22**, 556–563 (2008).
62. Davis, J. J. *et al.* Are Gait Patterns during In-Lab Running Representative of Gait Patterns during Real-World Training? An Experimental Study. *Sensors* **24**, 2892 (2024).
63. Chockalingam, N. Technologies and Techniques in Gait Analysis: Past, present and future. https://books.scholarsportal.info/uri/ebooks/ebooks7/iet7/2022-07-05/1/PBHE031E.
64. Electromyography (EMG) - Mayo Clinic. https://www.mayoclinic.org/tests-procedures/emg/about/pac-20393913 (2019).
65. Bieck, R., Fuchs, R. & Neumuth, T. Surface EMG-based Surgical Instrument Classification for Dynamic Activity Recognition in Surgical Workflows. *Curr. Dir. Biomed. Eng.* **5**, 37–40 (2019).
66. Bangaru, S. S., Wang, C. & Aghazadeh, F. Data Quality and Reliability Assessment of Wearable EMG and IMU Sensor for Construction Activity Recognition. *Sensors* **20**, 5264 (2020).
67. Tigrini, A. *et al.* Phasor-Based Myoelectric Synergy Features: A Fast Hand-Crafted Feature Extraction Scheme for Boosting Performance in Gait Phase Recognition. *Sensors* **24**, 5828 (2024).
68. Ziegler, J., Gattringer, H. & Mueller, A. Classification of Gait Phases Based on Bilateral EMG Data Using Support Vector Machines. in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)* 978–983 (2018). doi:10.1109/BIOROB.2018.8487750.
69. Fricke, C. *et al.* Evaluation of Three Machine Learning Algorithms for the Automatic Classification of EMG Patterns in Gait Disorders. *Front. Neurol.* **12**, (2021).
70. Zhang, Y. *et al.* Extracting time-frequency feature of single-channel vastus medialis EMG signals for knee exercise pattern recognition. *PLOS ONE* **12**, e0180526 (2017).
71. AL-Quraishi, M. S. *et al.* Classification of ankle joint movements based on surface electromyography signals for rehabilitation robot applications. *Med. Biol. Eng. Comput.* **55**, 747–758 (2017).

72. Phukpattaranont, P., Thongpanja, S., Anam, K., Al-Jumaily, A. & Limsakul, C. Evaluation of feature extraction techniques and classifiers for finger movement recognition using surface electromyography signal. *Med. Biol. Eng. Comput.* **56**, 2259–2271 (2018).
73. Tokas, P., Semwal, V. B. & Jain, S. An Efficient Deep Learning Framework for Lower Limb Activity Recognition From Wearable Sensors in Knee Pathologies. *IEEE Sens. J.* **25**, 44740–44750 (2025).
74. Coker, J., Chen, H., Schall, M. C., Gallagher, S. & Zabala, M. EMG and Joint Angle-Based Machine Learning to Predict Future Joint Angles at the Knee. *Sensors* **21**, 3622 (2021).
75. Zhao, H. *et al.* Prediction of Joint Angles Based on Human Lower Limb Surface Electromyography. *Sensors* **23**, 5404 (2023).
76. Gautam, A., Panwar, M., Biswas, D. & Acharyya, A. MyoNet: A Transfer-Learning-Based LRCN for Lower Limb Movement Recognition and Knee Joint Angle Prediction for Remote Monitoring of Rehabilitation Progress From sEMG. *IEEE J. Transl. Eng. Health Med.* **8**, 1–10 (2020).
77. Delsys Technical Note 101: EMG Sensor Placement. https://delsys.com/downloads/TECHNICALNOTE/101-emg-sensor-placement.pdf.
78. EMG User Guide Rev1.12. https://shimmersensing.com/wp-content/docs/support/documentation/EMG_User_Guide_Rev1.12.pdf (2017).
79. Vijayvargiya, A., Singh, B., Kumar, R. & Tavares, J. M. R. S. Human lower limb activity recognition techniques, databases, challenges and its applications using sEMG signal: an overview. *Biomed. Eng. Lett.* **12**, 343–358 (2022).
80. Donkrajang, W., Watthanawisuth, N., Mensing, J. P. & Kerdcharoen, T. Development of a wireless electronic shoe for walking abnormalities detection. in *The 5th 2012 Biomedical Engineering International Conference* 1–5 (2012). doi:10.1109/BMEiCon.2012.6465433.
81. Chatzaki, C. *et al.* The Smart-Insole Dataset: Gait Analysis Using Wearable Sensors with a Focus on Elderly and Parkinson's Patients. *Sensors* **21**, (2021).
82. Chen, D. *et al.* Bring Gait Lab to Everyday Life: Gait Analysis in Terms of Activities of Daily Living. *IEEE Internet Things J.* **7**, 1298–1312 (2020).
83. Raza, A. *et al.* Gait classification of knee osteoarthritis patients using shoe-embedded internal measurement units sensor. *Clin. Biomech.* **117**, (2024).
84. Halim, A., Abdellatif, A., Awad, M. I. & Atia, M. R. A. Prediction of human gait activities using wearable sensors. *Proc. Inst. Mech. Eng. [H]* **235**, 676–687 (2021).
85. Javed, T. *et al.* Multi-Class Classification of Human Activity and Gait Events Using Heterogeneous Sensors. *J. Sens. Actuator Netw.* **13**, 85 (2024).

86. Schlegel, K., Jiang, L. & Ni, H. Using joint angles based on the international biomechanical standards for human action recognition and related tasks. Preprint at https://doi.org/10.48550/arXiv.2406.17443 (2024).
87. Ishida, T. & Samukawa, M. The Difference in the Assessment of Knee Extension/Flexion Angles during Gait between Two Calibration Methods for Wearable Goniometer Sensors. *Sensors* **24**, 2092 (2024).
88. Nexus | Software For Motion Capture In Life Sciences. *Vicon* https://www.vicon.com/software/nexus/.
89. Vicon Nexus User Guide. https://help.vicon.com/space/Nexus217/350752592/PDF+downloads+for+Vicon+Nexus?attachment=https://help.vicon.com/download/attachments/350752592/Vicon%2520Nexus%2520User%2520Guide.pdf&type=application/pdf&filename=Vicon+Nexus+User+Guide.pdf.
90. Feldhege, F. *et al.* Accuracy of a Custom Physical Activity and Knee Angle Measurement Sensor System for Patients with Neuromuscular Disorders and Gait Abnormalities. *Sensors* **15**, 10734–10752 (2015).
91. Pesenti, M. *et al.* IMU-based human activity recognition and payload classification for low-back exoskeletons. *Sci. Rep.* **13**, 1184 (2023).
92. Steven Eyobu, O. & Han, D. S. Feature Representation and Data Augmentation for Human Activity Classification Based on Wearable IMU Sensor Data Using a Deep LSTM Neural Network. *Sensors* **18**, 2892 (2018).
93. Müller, P. N., Müller, A. J., Achenbach, P. & Göbel, S. IMU-Based Fitness Activity Recognition Using CNNs for Time Series Classification. *Sensors* **24**, 742 (2024).
94. Kuduz, H. & Kaçar, F. Biomechanical sensor signal analysis based on machine learning for human gait classification. *J. Electr. Eng.* **75**, 513–521 (2024).
95. Li, H., Derrode, S. & Pieczynski, W. An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model. *Neurocomputing* **362**, 94–105 (2019).
96. Mitchell, J. C., Dehghani-Sanij, A. A., Xie, S. Q. & O'Connor, R. J. Analysis of Multimodal Sensor Systems for Identifying Basic Walking Activities. *Technologies* **13**, 152 (2025).
97. Yang, Z., Van Beijnum, B.-J. F., Li, B., Yan, S. & Veltink, P. H. Estimation of Relative Hand-Finger Orientation Using a Small IMU Configuration. *Sensors* **20**, 4008 (2020).
98. Fusca, M. *et al.* Validation of a Wearable IMU System for Gait Analysis: Protocol and Application to a New System. *Appl. Sci.* **8**, 1167 (2018).
99. Lewin, M., Price, C. & Nester, C. Can a shoe-mounted IMU identify the effects of orthotics in ways comparable to gait laboratory measurements? *J. Foot Ankle Res.* **16**, 54 (2023).

100. DOT Pro/MTw2 Awinda - Getting started. https://wsdocs.movella.com/introduction.

101. Movella DOT User Manual. https://www.movella.com/hubfs/Movella%20DOT%20User%20Manual.pdf (2023).

102. Trigno Wireless Biofeedback System. https://delsys.com/downloads/USERSGUIDE/MAN-039-1-2%20Trigno%20Quattro%20Duo%20Mini%20Biofeedback%20Sensor.pdf (2023).

103. Trigno Wireless Biofeedback System User's Guide. https://delsys.com/downloads/USERSGUIDE/MAN-049-1-1%20Trigno%20Centro.pdf (2024).

104. Blue Trident Inertial Measurement Unit Sensor | Vicon Motion Capture. *Vicon* https://www.vicon.com/hardware/blue-trident/.

105. Movella DOT. https://www.xsens.com/hubfs/Xsens%20DOT%20data%20sheet.pdf.

106. McGrath, M., Wood, J., Walsh, J. & Window, P. The impact of Three-Dimensional Gait Analysis in adults with pathological gait on management recommendations. *Gait Posture* **105**, 75–80 (2023).

107. Romijnders, R. *et al.* Validation of IMU-based gait event detection during curved walking and turning in older adults and Parkinson's Disease patients. *J. NeuroEngineering Rehabil.* **18**, 28 (2021).

108. Lanotte, F., Okita, S., O'Brien, M. K. & Jayaraman, A. Enhanced gait tracking measures for individuals with stroke using leg-worn inertial sensors. *J. NeuroEngineering Rehabil.* **21**, 219 (2024).

109. Wu, Y.-C., Huang, Y.-J., Han, C.-C., Cheng, Y.-Y. & Chang, C.-S. Development of an IMU-Based Post-Stroke Gait Data Acquisition and Analysis System for the Gait Assessment and Intervention Tool. *Sensors* **25**, (2025).

110. Inacio, M. *et al.* Spectral parameters of gait differentiate diabetic patients from healthy individuals. *The Foot* **56**, (2023).

111. Hillel, I. *et al.* Is every-day walking in older adults more analogous to dual-task walking or to usual walking? Elucidating the gaps between gait performance in the lab and during 24/7 monitoring. *Eur. Rev. Aging Phys. Act.* **16**, (2019).

112. Favre, J. & Jolles, B. M. Gait analysis of patients with knee osteoarthritis highlights a pathological mechanical pathway and provides a basis for therapeutic interventions. https://eor.bioscientifica.com/view/journals/eor/1/10/2058-5241.1.000051.xml (2016).

113. Kour, N., Gupta, S. & Arora, S. A Survey of Knee Osteoarthritis Assessment Based on Gait. *Arch. Comput. Methods Eng.* **28**, 345–385 (2021).

114. Choursiya, P. *et al.* Gait Analysis Technologies for Measurement of Biomechanical Parameters of Knee Osteoarthritis. *SN Compr. Clin. Med.* **6**, 6 (2024).

115. Prisco, G. *et al.* Validity of Wearable Inertial Sensors for Gait Analysis: A Systematic Review. *Diagnostics* **15**, (2024).

116. Almuhammadi, W. S., Agu, E., King, J. & Franklin, P. OA-Pain-Sense: Machine Learning Prediction of Hip and Knee Osteoarthritis Pain from IMU Data. *Informatics* **9**, 97 (2022).

117. Di Raimondo, G. *et al.* Peak Tibiofemoral Contact Forces Estimated Using IMU-Based Approaches Are Not Significantly Different from Motion Capture-Based Estimations in Patients with Knee Osteoarthritis. *Sensors* **23**, (2023).

118. Pan, J. *et al.* Three-Dimensional gait biomechanics in patients with mild knee osteoarthritis. *Sci. Rep.* **15**, (2025).

119. Pan, J. *et al.* The effect of mild to moderate knee osteoarthritis on gait and three-dimensional biomechanical alterations. *Front. Bioeng. Biotechnol.* **13**, (2025).

120. Wilson, J. L., Amiri, P. & Brandon, S. Tibiofemoral joint contact forces during gait differ with radiographic knee oa progression over 3 years. *Osteoarthritis Cartilage* **29**, S176–S178 (2021).

121. Chang, A. H. *et al.* The relationship between the external knee adduction moment during gait and progression of knee tissue damage over two years in persons with knee osteoarthritis. *Osteoarthritis Cartilage* **22**, S44–S45 (2014).

122. Uhlrich, S. D., Silder, A., Beaupre, G. S., Shull, P. B. & Delp, S. L. Subject-specific toe-in or toe-out gait modifications reduce the larger knee adduction moment peak more than a non-personalized approach. *J. Biomech.* **66**, 103–110 (2018).

123. Gholami, S., Torkaman, G., Bahrami, F. & Bayat, N. Gait modification with subject-specific foot progression angle in people with moderate knee osteoarthritis: Investigation of knee adduction moment and muscle activity. *The Knee* **35**, 124–132 (2022).

124. Hutchison, L. *et al.* Relationship Between Knee Biomechanics and Pain in People With Knee Osteoarthritis: A Systematic Review and Meta-Analysis. *Arthritis Care Res.* **75**, 1351–1361 (2023).

125. Sakamoto, H. *et al.* THE RELATIONSHIPS BETWEEN STATIC ALIGNMENT, RADIOGRAPHIC DISEASE SEVERITY OR GAIT SPEED AND KNEE ADDUCTION MOMENT ESTIMATED BY A WEARABLE SYSTEM IN PATIENTS WITH KNEE OSTEOARTHRITIS. *Osteoarthritis Cartilage* **33**, S161–S162 (2025).

126. Xia, C. *et al.* Knee Osteoarthritis Classification System Examination on Wearable Daily-Use IMU Layout. in *Proceedings of the 2022 ACM International Symposium on Wearable Computers* 74–78 (Association for Computing Machinery, New York, NY, USA, 2022). doi:10.1145/3544794.3558459.

127. Vangeneugden, J. *et al.* Signatures of knee osteoarthritis in women in the temporal and fractal dynamics of human gait. *Clin. Biomech.* **76**, (2020).

128. Sun, R., Tomkins-Lane, C., Muaremi, A., Kuwabara, A. & Smuck, M. Physical activity thresholds for predicting longitudinal gait decline in adults with knee osteoarthritis. *Osteoarthritis Cartilage* **29**, 965–972 (2021).
129. Mahmoudian, A. *et al.* Changes in gait characteristics of women with early and established medial knee osteoarthritis: Results from a 2-years longitudinal study. *Clin. Biomech.* **50**, 32–39 (2017).
130. Hensor, E. M. A., Dube, B., Kingsbury, S. R., Tennant, A. & Conaghan, P. G. Toward a Clinical Definition of Early Osteoarthritis: Onset of Patient-Reported Knee Pain Begins on Stairs. Data From the Osteoarthritis Initiative1. *Arthritis Care Res.* **67**, 40–47 (2015).
131. Hubley-Kozey, C. L., Stanish, W. D., Urquhart, N., Ikeda, D. M. & Wilson, J. L. A. Longitudinal changes in knee joint gait mechanics and muscle activation patterns in individuals with medial compartment knee osteoarthritis. *Osteoarthritis Cartilage* **28**, S58–S59 (2020).
132. Alyami, M. & Nessler, J. A. Walking on a Vertically Oscillating Platform with Simulated Gait Asymmetry. *Symmetry* **13**, (2021).
133. McCain, E. M. *et al.* Reduced joint motion supersedes asymmetry in explaining increased metabolic demand during walking with mechanical restriction. *J. Biomech.* **126**, 110621 (2021).
134. Cerny, K., Perry, J. & Walker, J. M. Adaptations during the stance phase of gait for simulated flexion contractures at the knee. *Orthopedics* **17**, 501–513 (1994).
135. Harato, K. *et al.* Knee flexion contracture will lead to mechanical overload in both limbs: A simulation study using gait analysis. *The Knee* **15**, 467–472 (2008).
136. Wang, X., Kyrarini, M., Ristić-Durrant, D., Spranger, M. & Gräser, A. Monitoring of gait performance using dynamic time warping on IMU-sensor data. in *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)* 1–6 (2016). doi:10.1109/MeMeA.2016.7533745.
137. Hwang, S. *et al.* Machine Learning Based Abnormal Gait Classification with IMU Considering Joint Impairment. *Sensors* **24**, (2024).
138. IBM Data and AI Team. AI vs. Machine Learning vs. Deep Learning vs. Neural Networks | IBM. https://www.ibm.com/think/topics/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks (2023).
139. What Is Unsupervised Learning? | IBM. https://www.ibm.com/think/topics/unsupervised-learning (2021).
140. Bergmann, D. What Is Semi-Supervised Learning? | IBM. https://www.ibm.com/think/topics/semi-supervised-learning (2023).
141. Bergmann, D. What is Machine Learning? | IBM. https://www.ibm.com/think/topics/machine-learning (2025).

142. Belcic, I. & Stryker, C. What Is Supervised Learning? | IBM. https://www.ibm.com/think/topics/supervised-learning (2025).
143. What Is Deep Learning? | IBM. https://www.ibm.com/think/topics/deep-learning (2025).
144. Naru, P. S. Understanding Activations & Optimization- Neural Networks. https://community.ibm.com/community/user/ai-datascience/blogs/pavan-saish-naru/2023/01/27/optimization-hyperparameters (<date>).
145. What are Convolutional Neural Networks? | IBM. https://www.ibm.com/think/topics/convolutional-neural-networks (2021).
146. Romijnders, R., Warmerdam, E., Hansen, C., Schmidt, G. & Maetzler, W. A Deep Learning Approach for Gait Event Detection from a Single Shank-Worn IMU: Validation in Healthy and Neurological Cohorts. *Sensors* **22**, (2022).
147. Pérez-Ibarra, J. C., Siqueira, A. A. G. & Krebs, H. I. Identification of Gait Events in Healthy and Parkinson's Disease Subjects Using Inertial Sensors: A Supervised Learning Approach. *IEEE Sens. J.* **20**, 14984–14993 (2020).
148. Marcos Mazon, D., Groefsema, M., Schomaker, L. R. B. & Carloni, R. IMU-Based Classification of Locomotion Modes, Transitions, and Gait Phases with Convolutional Recurrent Neural Networks. *Sensors* **22**, (2022).
149. Reiss, A. & Stricker, D. Introducing a New Benchmarked Dataset for Activity Monitoring. *2012 16th Int. Symp. Wearable Comput.* 108–109 (2012) doi:10.1109/ISWC.2012.13.
150. Ayman, A., Attalah, O. & Shaban, H. An Efficient Human Activity Recognition Framework Based on Wearable IMU Wrist Sensors. in *2019 IEEE International Conference on Imaging Systems and Techniques (IST)* 1–5 (2019). doi:10.1109/IST48021.2019.9010115.
151. Tseng, Y.-H. & Wen, C.-Y. Hybrid Learning Models for IMU-Based HAR with Feature Analysis and Data Correction. *Sensors* **23**, 7802 (2023).
152. Sherratt, F., Plummer, A. & Iravani, P. Understanding LSTM Network Behaviour of IMU-Based Locomotion Mode Recognition for Applications in Prostheses and Wearables. *Sensors* **21**, (2021).
153. Bijalwan, V., Semwal, V. B. & Gupta, V. Wearable sensor-based pattern mining for human activity recognition: deep learning approach. *Ind. Robot Int. J. Robot. Res. Appl.* **49**, 21–33 (2021).
154. Jorge Reyes-Ortiz, D. A. Human Activity Recognition Using Smartphones. UCI Machine Learning Repository https://doi.org/10.24432/C54S4K (2013).
155. Oresti Banos, R. G. MHEALTH. UCI Machine Learning Repository https://doi.org/10.24432/C5TW22 (2014).

156. Tao, S., Goh, W. L. & Gao, Y. A Convolved Self-Attention Model for IMU-based Gait Detection and Human Activity Recognition. in *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)* 1–5 (2023). doi:10.1109/AICAS57966.2023.10168654.
157. Tan, J.-S. *et al.* Human Activity Recognition for People with Knee Osteoarthritis—A Proof-of-Concept. *Sensors* **21**, (2021).
158. Weiss, G. WISDM Smartphone and Smartwatch Activity and Biometrics Dataset. UCI Machine Learning Repository https://doi.org/10.24432/C5HK59 (2019).
159. Slemenšek, J. *et al.* Human Gait Activity Recognition Machine Learning Methods. *Sensors* **23**, (2023).
160. Dibbern, K. N. *et al.* Scoping Review of Machine Learning Techniques in Marker-Based Clinical Gait Analysis. *Bioengineering* **12**, (2025).
161. Iseki, C. *et al.* Artificial Intelligence Distinguishes Pathological Gait: The Analysis of Markerless Motion Capture Gait Data Acquired by an iOS Application (TDPT-GT). *Sensors* **23**, 6217 (2023).
162. Aoyagi, Y. *et al.* Development of Smartphone Application for Markerless Three-Dimensional Motion Capture Based on Deep Learning Model. *Sensors* **22**, (2022).
163. Ali, Z. *et al.* Vision-based approach to knee osteoarthritis and Parkinson's disease detection utilizing human gait patterns. *PeerJ Comput. Sci.* **11**, e2857–e2857 (2025).
164. Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E. & Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 172–186 (2021).
165. Bradski, G. The OpenCV Library An open-source library for processing image data. https://jacobfilipp.com/DrDobbs/articles/DDJ/2000/0011/0011k/0011k.htm (2000).
166. Chavez, J. M. & Tang, W. A Vision-Based System for Stage Classification of Parkinsonian Gait Using Machine Learning and Synthetic Data. *Sensors* **22**, (2022).
167. Mahum, R. *et al.* A Novel Hybrid Approach Based on Deep CNN Features to Detect Knee Osteoarthritis. *Sensors* **21**, (2021).
168. Moghadam, S. M., Yeung, T. & Choisne, J. The Effect of IMU Sensor Location, Number of Features, and Window Size on a Random Forest Model's Accuracy in Predicting Joint Kinematics and Kinetics During Gait. *IEEE Sens. J.* **23**, 28328–28339 (2023).
169. Xu, L. *et al.* Machine-learning-based children's pathological gait classification with low-cost gait-recognition system. *Biomed. Eng. OnLine* **20**, 62 (2021).
170. Kaya, E., Şirzai, H., Yavuzer, G. & Argunsah, H. Classification of knee osteoarthritis severity using markerless motion capture and long short-term memory fully convolutional network. *Comput. Biol. Med.* **195**, (2025).

171. SparkFun 9DoF IMU Breakout - ICM-20948 (Qwiic) - SparkFun Electronics. https://www.sparkfun.com/sparkfun-9dof-imu-breakout-icm-20948-qwiic.html.
172. ICM-20948. https://cdn.sparkfun.com/assets/7/f/e/c/d/DS-000189-ICM-20948-v1.3.pdf.
173. Nguyen, V. IMU-INTEGRATED ‘SMART’ KNEE BRACE VALIDATION USING OPTOELECTRONIC MOTION CAPTURE DURING TREADMILL WALKING.
174. Biga, L. M. *et al.* 1.4 Anatomical Terminology. https://open.oregonstate.education/aandp/chapter/1-4-anatomical-terminology/ (2019).
175. Davidson, K. The 3 Anatomical Body Planes and The Movements In Each. *Healthline* https://www.healthline.com/health/body-planes (2023).
176. Antonsson, E. K. & Mann, R. W. The frequency content of gait. *J. Biomech.* **18**, 39–47 (1985).
177. SFUptownMaker. I2C - SparkFun Learn. https://learn.sparkfun.com/tutorials/i2c/all.
178. fft — SciPy v1.17.0 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.fft.fft.html.
179. Digital Laser and Contact Tachometer User Manual. https://www.princessauto.com/file/general/8714941_manualhb_00_01_v01_manuel_enfr.pdf.
180. Wang, S. *et al.* Estimation of knee joint angle during gait cycle using inertial measurement unit sensors: a method of sensor-to-clinical bone calibration on the lower limb skeletal model. *J. Biomech. Sci. Eng.* **17**, 21–00196 (2022).
181. Favre, J., Jolles, B. M., Aissaoui, R. & Aminian, K. Ambulatory measurement of 3D knee joint angle. *J. Biomech.* **41**, 1029–1035 (2008).
182. Gangeddula, V. Knee Joint Biomechanics in People with Medial Compartment Knee Osteoarthritis. https://qspace.library.queensu.ca/items/f8785a2b-771a-4628-8eb6-f3fb64202db8 (2009).
183. Primal’s 3D Atlas of Human Anatomy of the Knee. https://www.anatomy.tv/anatomytv/html5ui_2018/#/product/har_knee_2014/type/Views/id/39302/layer/0/angle/0/structureID/546816.
184. Primal’s 3D Atlas of Human Anatomy of the Leg, Ankle and Foot. https://www.anatomy.tv/anatomytv/html5ui_2018/#/product/har_foot_2014/type/Views/id/39343/layer/0/angle/5/structureID/567320.
185. Primal’s 3D Atlas of Human Anatomy of the Hip. https://www.anatomy.tv/anatomytv/html5ui_2018/#/product/har_hip_2014/type/Views/id/43005/layer/0/angle/34/structureID/526344.
186. Skeleton Anterior. *Servier Medical Art* https://smart.servier.com/smart_image/skeleton-face/.
187. Skeleton Posterior. *Servier Medical Art* https://smart.servier.com/smart_image/skeleton-back/.
188. Dingwell, J. B. & Marin, L. C. Kinematic variability and local dynamic stability of upper body motions when walking at different speeds. *J. Biomech.* **39**, 444–452 (2006).

189. Kiriella, J. B., Di Bacco, V. E., Hollands, K. L. & Gage, W. H. Evaluation of the Effects of Prescribing Gait Complexity Using Several Fluctuating Timing Imperatives. *J. Mot. Behav.* **52**, 570–577 (2020).
190. Zhen, T., Yan, L. & Yuan, P. Walking Gait Phase Detection Based on Acceleration Signals Using LSTM-DNN Algorithm. *Algorithms* **12**, 253 (2019).
191. Crenna, F., Rossi, G. B. & Berardengo, M. Filtering Biomechanical Signals in Movement Analysis. *Sensors* **21**, 4580 (2021).
192. Khusainov, R., Azzi, D., Achumba, I. E. & Bersch, S. D. Real-Time Human Ambulation, Activity, and Physiological Monitoring: Taxonomy of Issues, Techniques, Applications, Challenges and Limitations. *Sensors* **13**, 12852–12902 (2013).
193. Staab, W. *et al.* Accelerometer and Gyroscope Based Gait Analysis Using Spectral Analysis ofPatients with Osteoarthritis of the Knee. *J. Phys. Ther. Sci.* **26**, 997–1002 (2014).
194. Nazarzadeh, K., Arjunan, S. P., Kumar, D. K. & Das, D. P. Non-invasive detection of the freezing of gait in Parkinson's disease using spectral and wavelet features. *2016 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC* 876–879 (2016) doi:10.1109/EMBC.2016.7590840.
195. PSU Dept of Statistics. 3.2 - Identifying Outliers: IQR Method | STAT 200. https://online.stat.psu.edu/stat200/lesson/3/3.2.
196. f_oneway — SciPy v1.17.0 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.f_oneway.html#scipy.stats.f_oneway.
197. Analysis of Variance (ANOVA). https://sites.calvin.edu/scofield/courses/m143/materials/handouts/anova1And2.pdf.
198. DOUGLAS C. MONTGOMERY & GEORGE C. RUNGER. Applied Statistics and Probability for Engineers. https://people.stat.sc.edu/Tebbs/stat509/Montgomery_2018.pdf.
199. Stats. https://www.cs.columbia.edu/~julia/courses/Resources/Stats.pdf.
200. shapiro — SciPy v1.17.0 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html#scipy.stats.shapiro.
201. bartlett — SciPy v1.17.0 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.bartlett.html#scipy.stats.bartlett.
202. tukey_hsd — SciPy v1.17.0 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.tukey_hsd.html#scipy.stats.tukey_hsd.
203. MinMaxScaler. *scikit-learn* https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.

204. Daniel Jurafsky & James H. Martin. Speech and Language Processing. https://web.stanford.edu/~jurafsky/slp3/ed3book_Jan25.pdf.
205. welch — SciPy v1.17.0 Manual. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.welch.html.
206. Abhijeet, A. & Nayak, S. B. Cross-entropy loss - NISER,Bhubaneswar, Odisha. (2023).
207. CrossEntropyLoss — PyTorch 2.10 documentation. https://docs.pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.
208. Ng, A. & Katanforoosh, K. Section 8 (Week 8) Advanced Evaluation Metrics. https://cs230.stanford.edu/section/8/ (2025).

APPENDIX

A. PARTS LIST

Appendix A-1: Parts list for the DCU. Prices are in CAD at time of compiling this table, and are sourced from Digikey Canada for consistency, unless otherwise noted. Prices will vary by vendor so total cost is an estimate. Schematic ID links to schematic diagram in Appendix C-2.

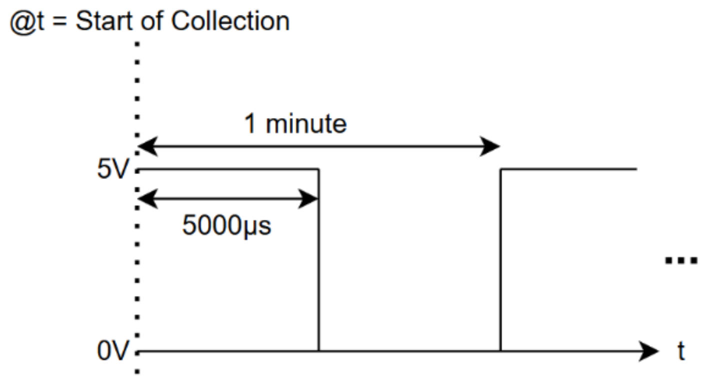
Part	Model	Qt	Purpose	Schematic ID	Cost Per Item (Digikey, $ CAD)	Total Cost (Digikey, $ CAD)	Notes
SparkFun Thing Plus - ESP32 WROOM (Micro-B)	WRL-15663	1	Microcontroller	U2	38.75	38.75	-
SparkFun 9DoF IMU Breakout - ICM-20948 (Qwiic)	SEN-15335	2	Sensor used to collect data	-	31.98	63.96	-
Flexible Qwiic Cable - 500mm	PRT-17257	4	Wires used to connect to the sensors	-	4.65	18.60	-
Flexible Qwiic Cable - 50mm	PRT-14426 (Use PRT-17260)	2	Wires used to connect the multiplexer to the Qwiic Adapter, and the adapter to the microcontroller	-	2.2	4.40	Use second part number - first part is deprecated
SparkFun Qwiic Mux Breakout - 8 Channel (TCA9548A)	BOB-16784	1	The multiplexer used to interface with the IMUs	-	21.87	21.87	-
SparkFun Qwiic Adapter	DEV-14495	1	The Qwiic adapter connects the multiplexer to the microcontroller	U5	2.33	2.33	-
Panel Mount Button	PV2F240NN	1	The Start/Stop Button	U1	8.98	8.98	-
Panel Mount RCA Jack	RCJ-031	1	The sync signal port	J3	2.75	2.75	-
Male Header	22112022	1	Header for RCA port wires to connect to	-	0.55	0.55	-
Rectangular Cable Assembly	2177961022	1	Wires to connect from RCA port to the corresponding male header	-	2.71	2.71	-
Lexar High-Performance 633x 32GB (2-Pack) microSDHC UHS-I Card	LMS0633032G-B2ANU	1	The MicroSD Card that the data will be stored on	-	25.61 for 2	25.61	Amazon, 2 pk, Lexar 32GB
WSZJINB Velcro Strap and Plastic Loop	-	50"	Strap threaded through the enclosure to attach to the participant	-	13 for 32'	13	Amazon has a 32' roll that can be cut down to size
Adafruit DS3231 Precision RTC Breakout	3013	1	Holds time while the system is not on	U3	25.5	25.5	-
3V Coin Cell Battery	CR1220	1	Allows the RTC to hold time while not powered by the system's battery	-	9.74 for 5	9.74	Amazon Panasonic CR1220 3 Volt Lithium Coin Battery (5 Batteries)
Adafruit MicroSD Card Module	4682	1	Allows the microcontroller to interface with the MicroSD card	U4	5.1	5.1	-
Partial Total	-					243.85	-

Appendix A-2: Continued parts list for the DCU. Prices are in Canadian dollars at time of compiling this table, and are sourced from Digikey Canada for consistency, unless otherwise noted. Prices will vary by vendor so total cost is an estimate. Schematic ID links to schematic diagram in Appendix C-2.

Part	**Model**	**Qt**	**Purpose**	**Schematic ID**	**Cost Per Item (Digikey, $ CAD)**	**Total Cost (Digikey, $ CAD)**	**Notes**
Kingbright LED Through Hole Blue 2000mcd 470nm 20 deg Water Clear	WP710A10 QBC/D	1	Indicator LED	D1	0.61	0.61	-
Polymer Lithium Ion Battery - 1000mAh, 3.7V	PRT-13813-B	1	Battery Power for the System	-	-	-	Unable to find similar part that is being sold in Canada
(Optional) SparkFun LiPo Charger Basic - Micro-USB	PRT-10217	1	Charger for Battery (Battery can also be charged via the microcontroller)	-	15.3	15.3	-
Male/Female JST-PH 2.0 Connectors	-	2	Connectors which can easily interface with the ports on the microcontroller and charger.	J1, J2	10 for 10	10	Be mindful of connection polarity! Wire colouring is not always indicative of which side is positive or negative
E-Switch Slide Switches SPDT ON-ON SIDE	600SP1S3 M1QE	1	Main Power Switch	SW1	3.22	3.22	-
M3x4mm Machine Screws	-	12	Fastening Hardware; PCB and MUX to Enclosure, Lid to Enclosure	H1 – H4	-		Can be acquired easily in bulk
Male Headers	-	2	Serve as Sockets, 2.54mm pitch; ensure M/F mating compatibility	-	-		
Female Headers	-	2		-			
.25W 10K Resistor	-	4	Needed for SPI/SDIO board	R3 – R6	-		
.25W 330 Resistor	-	1	Resistive Divider needed to take in 5V sync signal	R2	-		
.25W 220 Resistor	-	1		R1	-		
Partial Total	-					29.13	~$45 CAD with battery, misc components
Total over both Appendix A-1 and Appendix A-2	-					272.98	~$300 CAD with costs of resistors, 3D filament, misc

B. DCU SYNCHRONISATION SIGNAL

i)

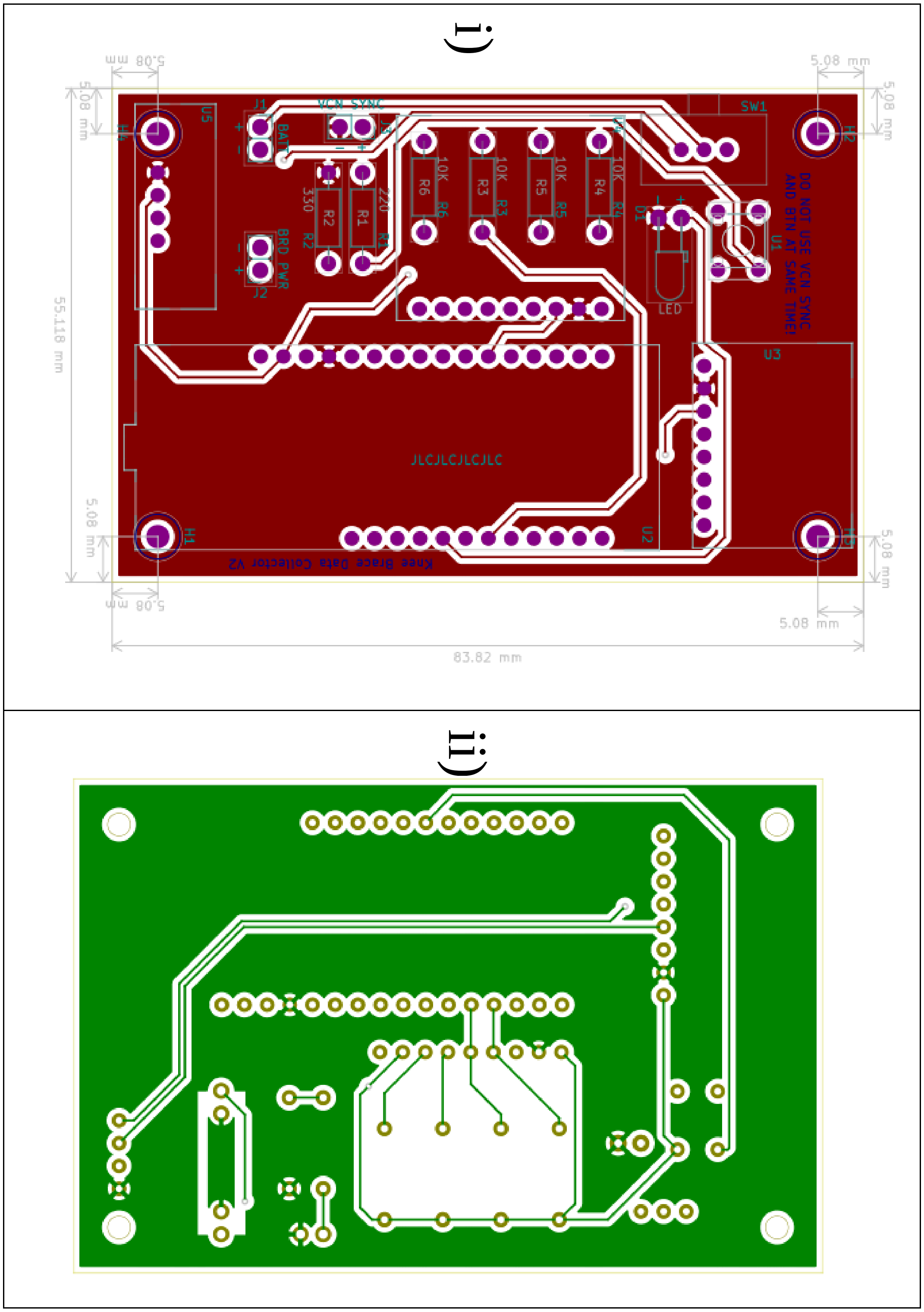


ii)

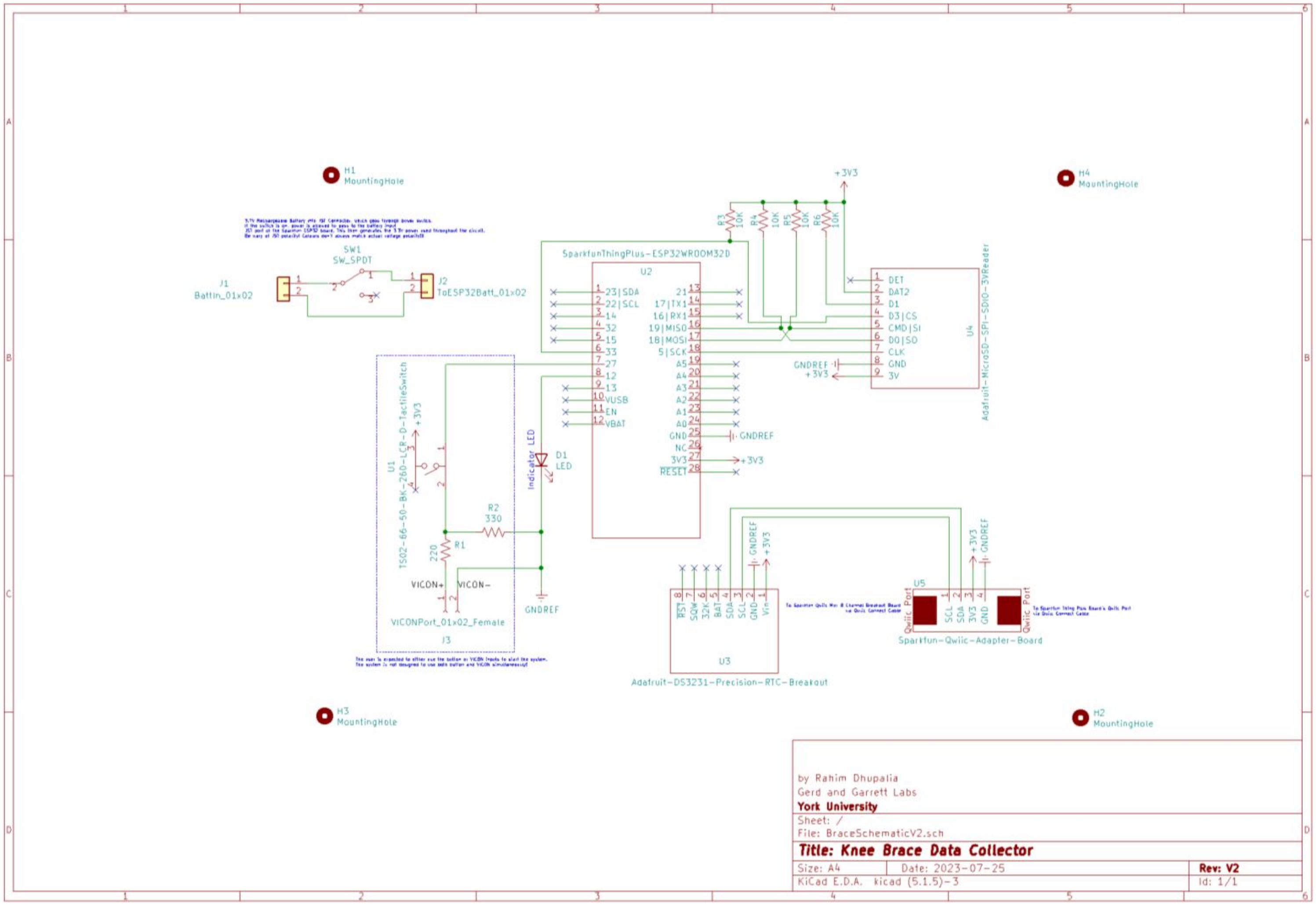
```
<?xml version="1.0" standalone="yes"?>
<AllPrograms>
<Program Name="Duration">
        <Type>Repeating</Type>
        <Polarity>High</Polarity>
        <StartEvent>StartCapture</StartEvent>
        <StopEvent>StopCapture</StopEvent>
        <StartOffset Frames="0" MicroSeconds="0"/>
        <StopOffset Frames="0" MicroSeconds="0"/>
        <PulseWidth Frames="0" MicroSeconds="5000"/>
        <PulsePeriod Frames="0" MicroSeconds="60000000" Ticks="0"/>
</Program>
</AllPrograms>
```

Appendix B-1: Sync signal, i) defined in time and ii) as a VICON GPO XML file. When the collection is started on the VICON system, it also sends out the signal shown in i) via its MX Sync module's GPO. The signal is a constant 5V output for 5 milliseconds, followed by 0V. The signal stays at 0V for almost a minute, until the signal repeats. Note the 1 minute period is to give the individual time to disconnect from the sync cable and is not a functional feature of the signal itself, similarly, the repeating nature is not a functional feature of the signal.

C. PCB AND ELECTRONIC SCHEMATIC DIAGRAM

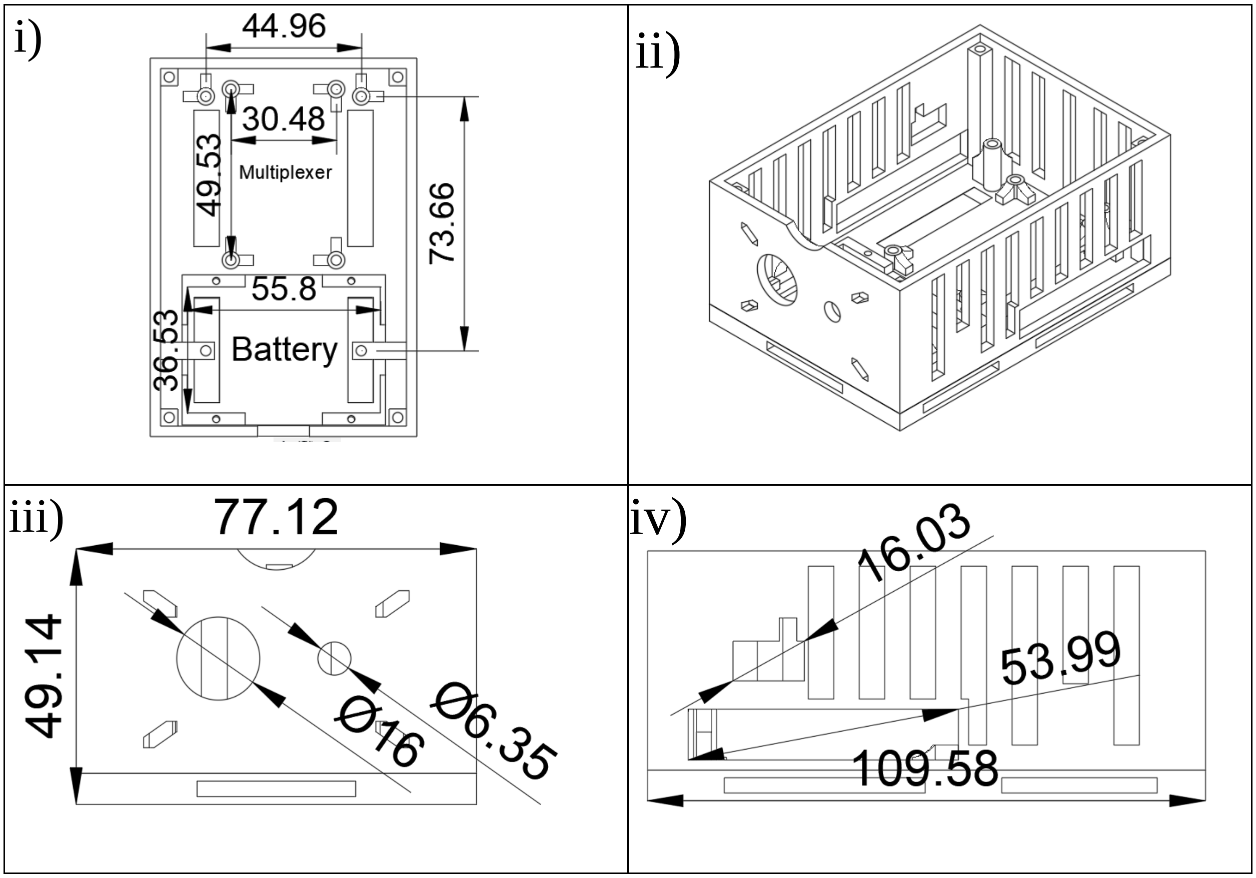


Appendix C-1: PCB designs (i) Front, ii) Back) that were printed by JLCPCB (JLCPCB, New Territories, Hong Kong, China). The PCB went through multiple iterations trying to reduce the overall size of the board. The main microcontroller, the Real-Time clock, the microSD module, the Qwiic Adapter, the indicator LED, the ON/OFF switch and resistors populated the board. The Qwiic MUX, battery, Start/Stop button, the VICON synchronization port were all placed on the enclosure and are not physically found on the board itself. The PCB is mounted to the enclosure by 4mm M3 screws, which held the board securely during activities.

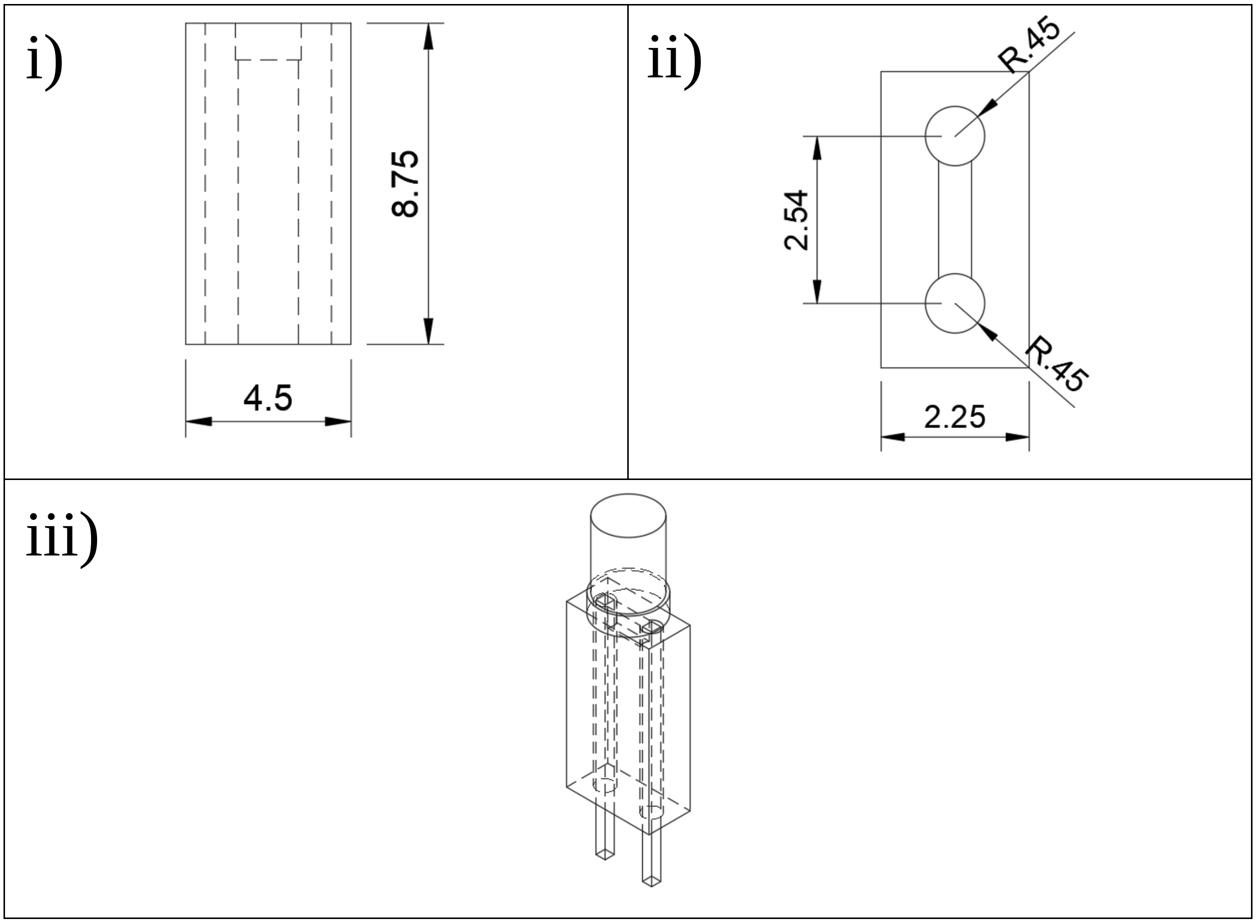


Appendix C-2: Schematic diagram of the circuit. At the heart of the circuit is the microcontroller, which communicates with the real-time clock module for timestamping data, the Qwiic Adapter Board to interface with the IMUs and collect data, and the microSD module to write the data. There is some simple circuitry for the sake of VICON synchronisation, which can remotely trigger a Start or Stop of collection. Alternatively, a push buttton can be used for starting and stopping collection as well. An ON/OFF switch is connected to the battery to control power. There is an LED that is used to indicate the status of the system to the user.

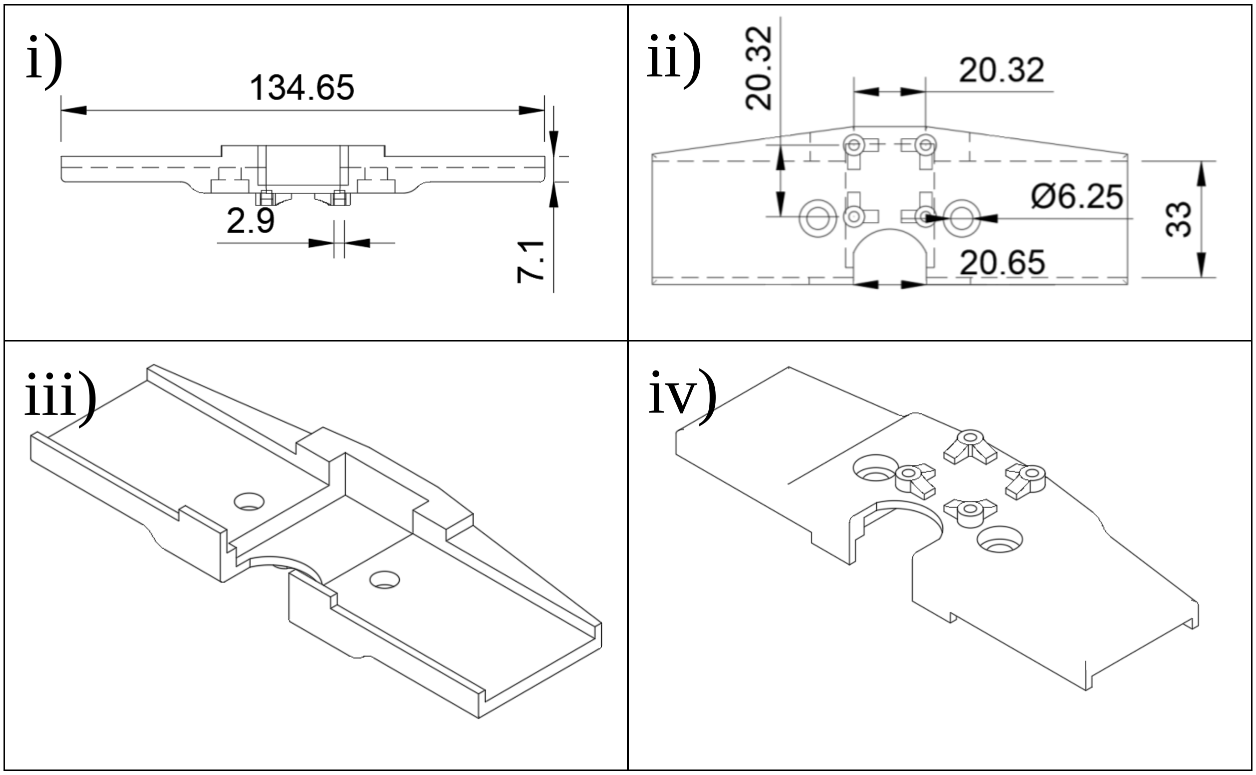
D. 3D PRINTED COMPONENTS



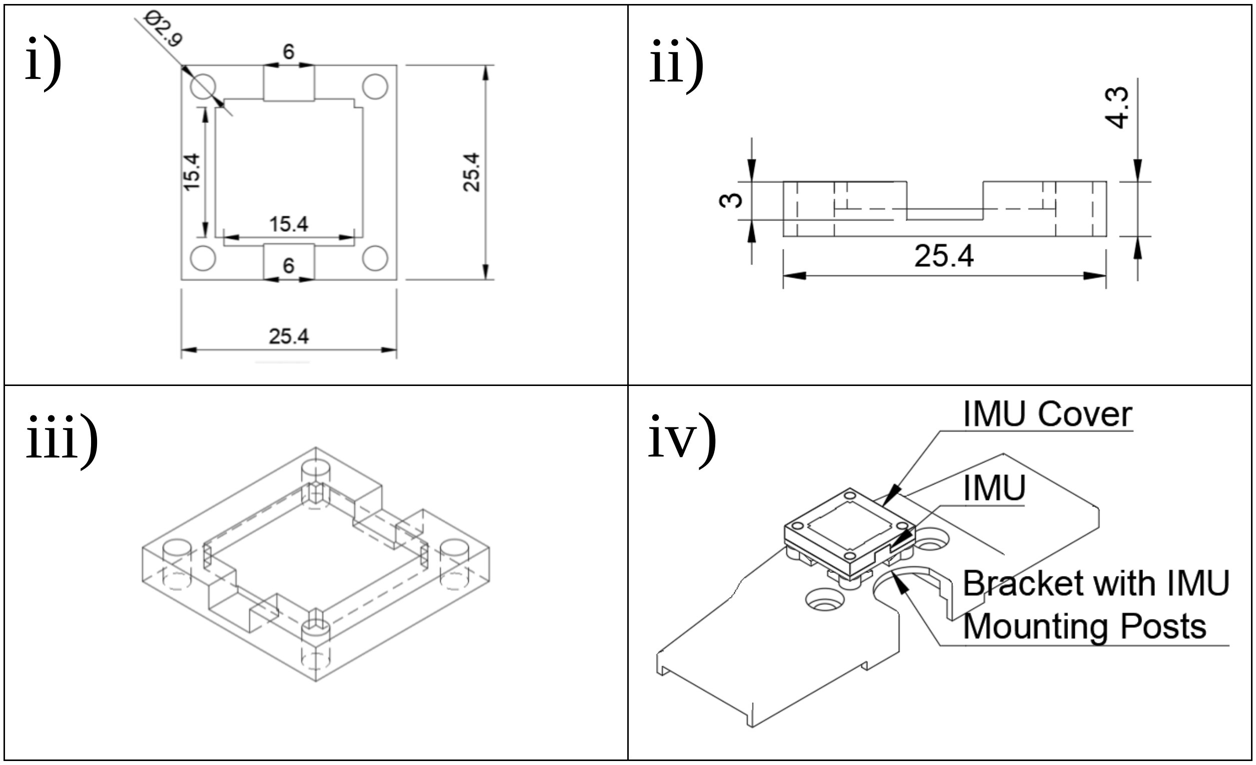
Appendix D-1: 3D Model of Enclosure, all units in mm. i) Top View, showing sizes of areas for the multiplexer, battery, and PCB. The PCB and multiplexer both screw into posts with M3x4mm screws ii) Isometric View, iii) Front View, showing length and height of the enclosure, and the cut-outs for the panel mounted components. iv) Side View, which would face up towards user when the enclosure is tied at the waist. There is a window from where the power switch is usable and the LED is viewable, and a window for access to the multiplexer's communication ports. The sensors would plug in from the ports on the opposite side to the one show.



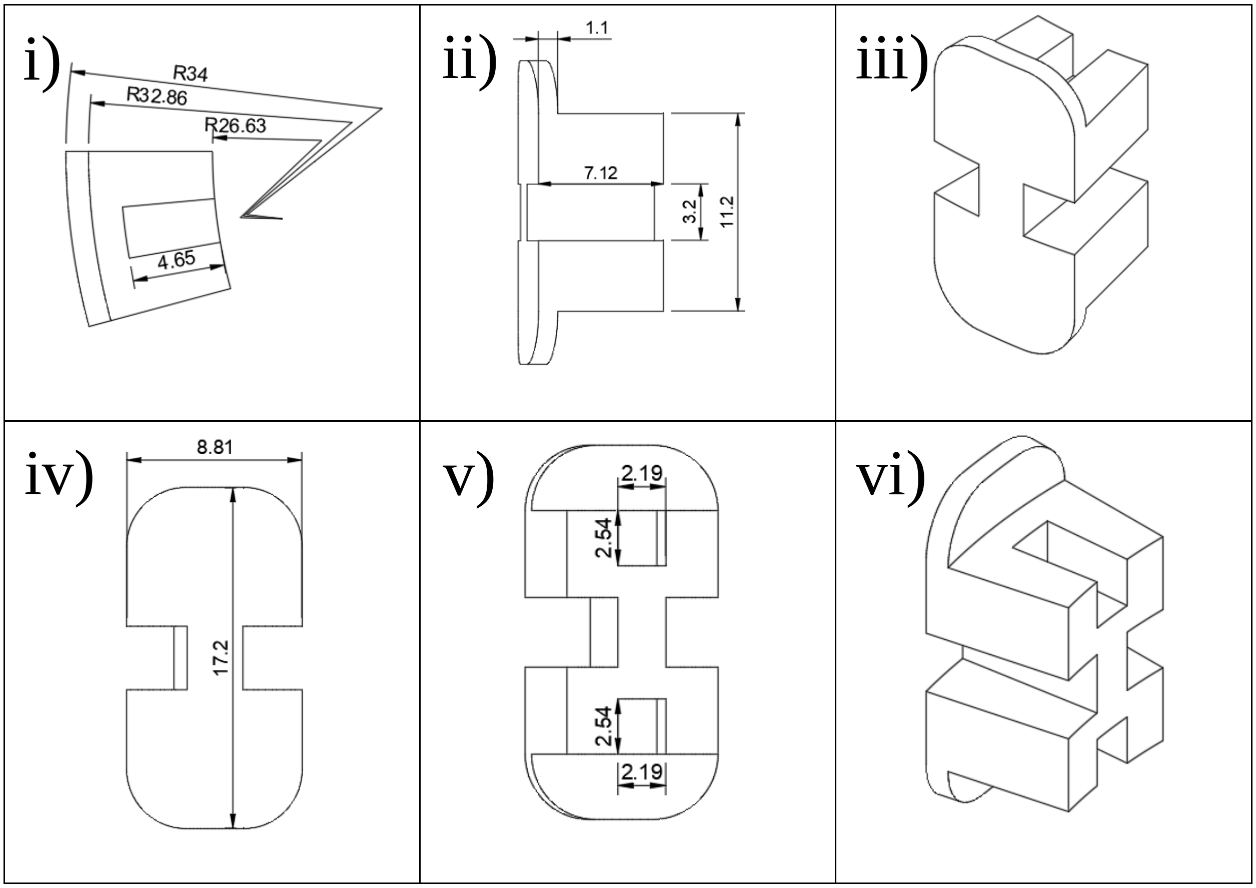
Appendix D-2: 3D Printed LED Standoff, all units in mm. This part, in accordance with manufacturer recommendation, supports the weight of the LED when it is positioned vertically off the PCB. Standoff may not be necessary for all LEDs. i) Front View showing height and width. ii) Top View showing depth of the standoff, the maximum radius of the LED leads, and their spacing. iii) Isometric View showing LED placed into standoff.



Appendix D-3: 3D-Printed Bracket, with base design taken from manufacturer provided bracket. All units in mm. i) Side View, showing length and thickness of the bracket, and the M3 screw holes for IMU mounting. ii) Top View, showing the IMU mounting posts, the gap needed to accommodate the length-adjustment button, and the 6.25mm screw holes that allow the manufacturer provided screws to hold the bracket to the brace. iii) Isometric Bottom View. iv) Isometric Top View.



Appendix D-4: 3D-Printed IMU Cover, all units in mm. The purpose is to prevent any sort of damage or debris from contacting the exposed side of the IMU PCB. i) Bottom View, showing the size of the cover, and the gaps cut out for sensor communication port, M3 screw holes, and the inset for the IMU board. ii) Front View showing cover depth and communication port depth. iii) Isometric Bottom View. iv) The Assembly. The cover goes on top of the IMUs mounted to the plastic bracket shown, and is fastened together to the bracket with M3x10mm screws.



Appendix D-5: 3D-Printed disposable lock, used to restrict movement of participants during motion. All units are in mm. i) Top View, showing the curved nature of the lock, and the length of the groove that fits into the teeth of the brace hinge. ii) Side View, showing depth and width of the lock, as well as the width of the groove that limits the swinging metal frame of the brace. iii) Isometric Front View. iv) Front View. v) Back View, showing the grooves that fit into the teeth in the brace hinge. vi) Isometric Back View.